State Feedback

Prof. Seungchul Lee
Industrial AI Lab.

State Space Representation

- Given a point mass on a line whose acceleration is directly controlled:

$$\ddot{p} = u$$

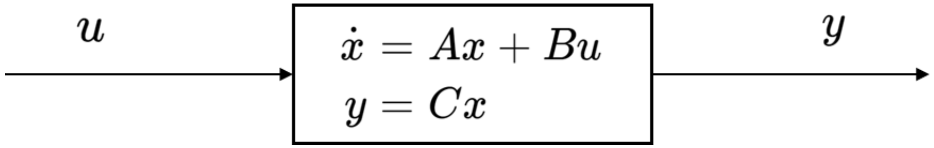
- We want to write this on a compact/general form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned}$$

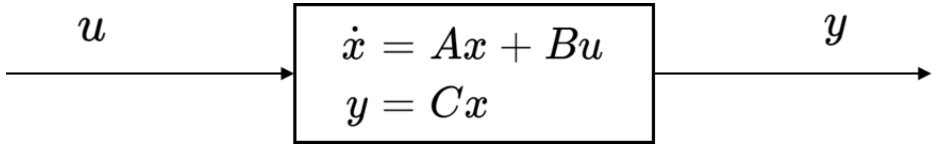
- On a state space form

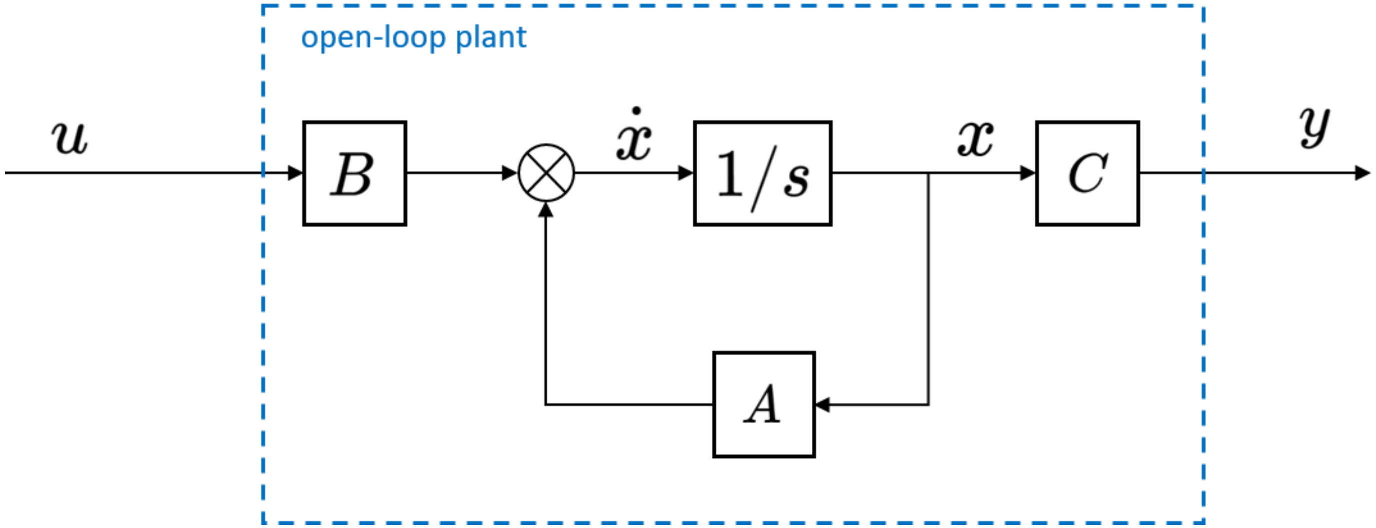
$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = p = x_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



Block Diagram





The Car Model

$$\dot{x} = \frac{c}{m}u - \gamma x$$

- If we care about/can measure the velocity:

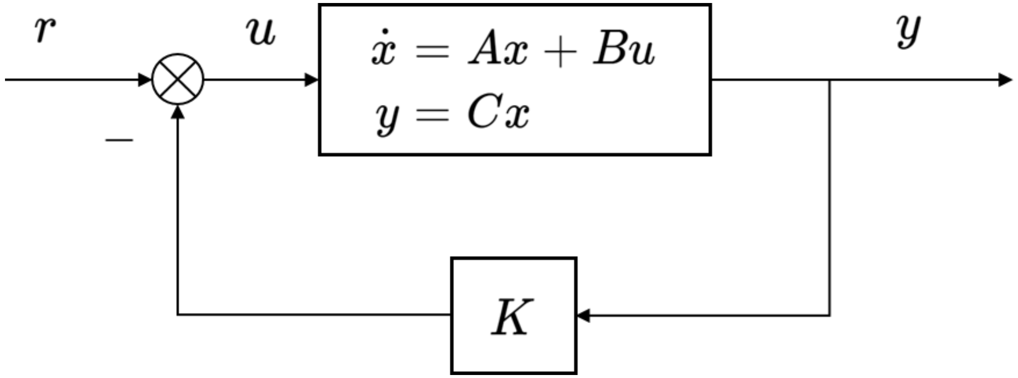
$$A = -\gamma, \qquad B = \frac{c}{m}, \qquad C = 1$$

- If we care about/can measure the position we have the same general equation with different matrices:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\gamma \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \frac{c}{m} \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Output Feedback

- Control idea: move towards the origin $r = 0$



$$u = r - Ky = -KCx$$

$$\dot{x} = Ax + Bu = Ax - BKCx = (A - BKC)x$$

Output Feedback

- Assume $\gamma = 0$
- Pick, if possible, $K\ (= 1)$ such that

$$\mathrm{Re}\,(\lambda) < 0 \quad \forall\lambda \in \mathrm{eig}\,(A - BKC)$$

$$\dot{x} = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 \begin{bmatrix} 1 & 0 \end{bmatrix} \right) x$$

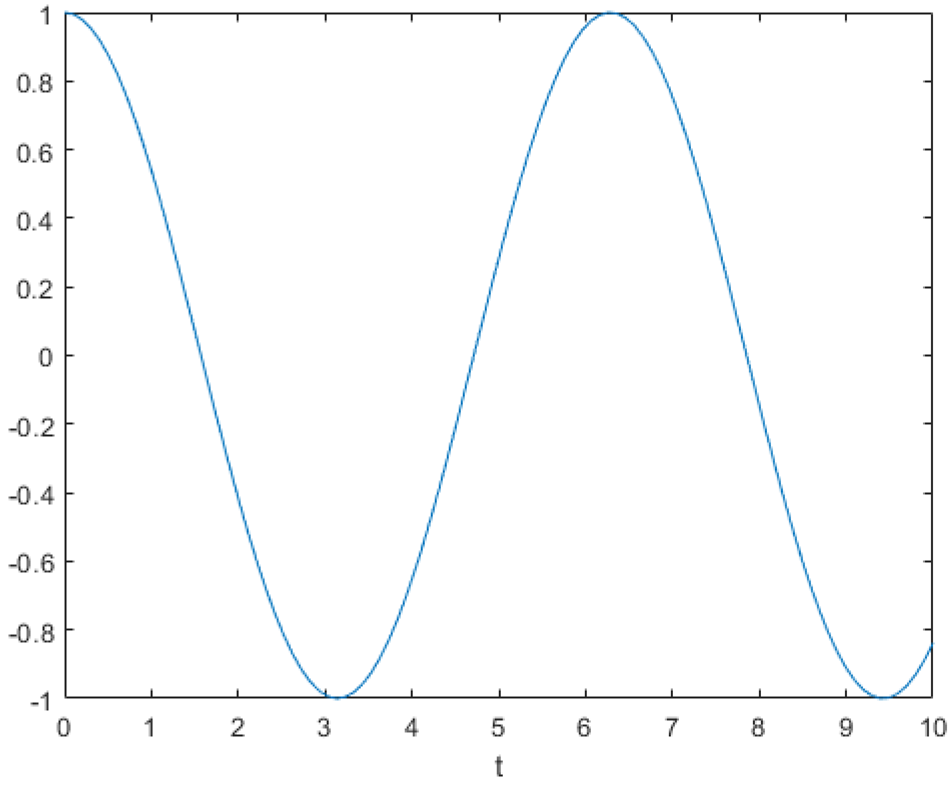
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x$$

$$\mathrm{eig}(A - BKC) = \pm j$$

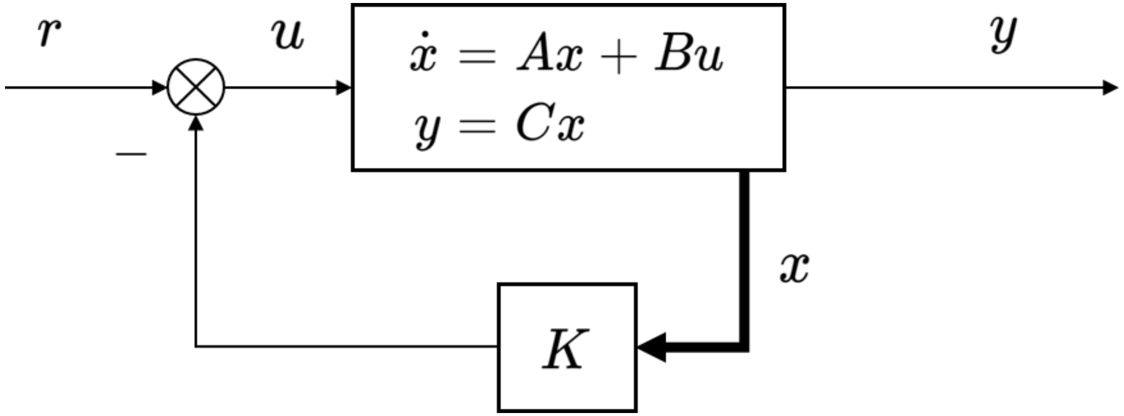
- What's the problem?
  - the problem is that we do not take the velocity into account
  - we need to use the full state information in order to stabilize this system

Output Feedback in MATLAB

```
A = [0 1;0 0];
B = [0 1]';
C = [1 0];
D = 0;
G = ss(A,B,C,D);

K = 1;
Gcl = feedback(G,K,-1);
x0 = [1 0]';
t = linspace(0,10,100);
r = zeros(size(t));
[y,tout] = lsim(Gcl,r,t,x0);
plot(tout,y), xlabel('t')
```



State Feedback



- To move forwards origin, $r = 0$

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

$$u = -Kx$$

$$\dot{x} = Ax + Bu = Ax - BKx = (A - BK)x$$

State Feedback

- Pick, if possible, K such that the closed-loop system is stabilized

$$\mathrm{Re}\left(\mathrm{eig}(A - BK)\right) < 0$$

$$K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

$$\dot{x} = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \right) x$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} x$$

State Feedback

- Let's try
  - Asymptotically stable
  - Damped oscillations

$$k_1 = k_2 = 1$$

$$A - BK = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$$

$$\text{eig}(A - BK) = -0.5 \pm\ 0.866j$$

- Let's do another attempt
  - Asymptotically stable
  - No oscillations

$$k_1 = 0.1, k_2 = 1$$

$$A - BK = \begin{bmatrix} 0 & 1 \\ -0.1 & -1 \end{bmatrix}$$

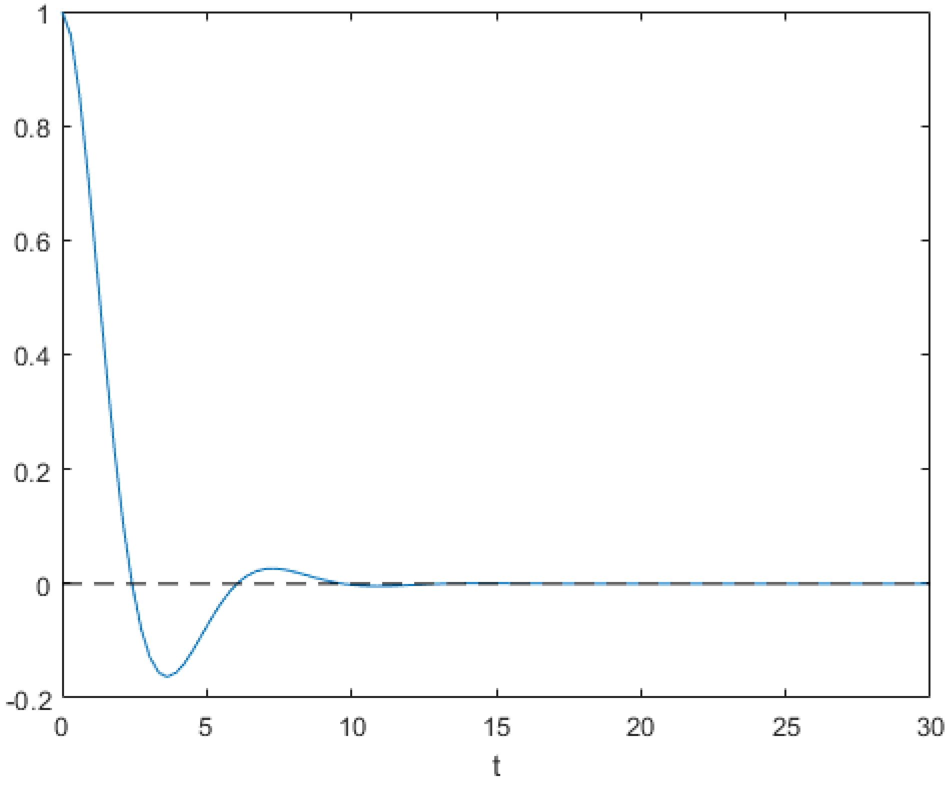
$$\text{eig}(A - BK) = -0.1127, -0.8873$$

State Feedback

- Eigenvalues Matter
  - It is clear that some eigenvalues are better than others. Some cause oscillations, some make the system respond too slowly, and so forth ...
  - We will see how to select eigenvalues and how to pick control laws based on the output rather than the state.

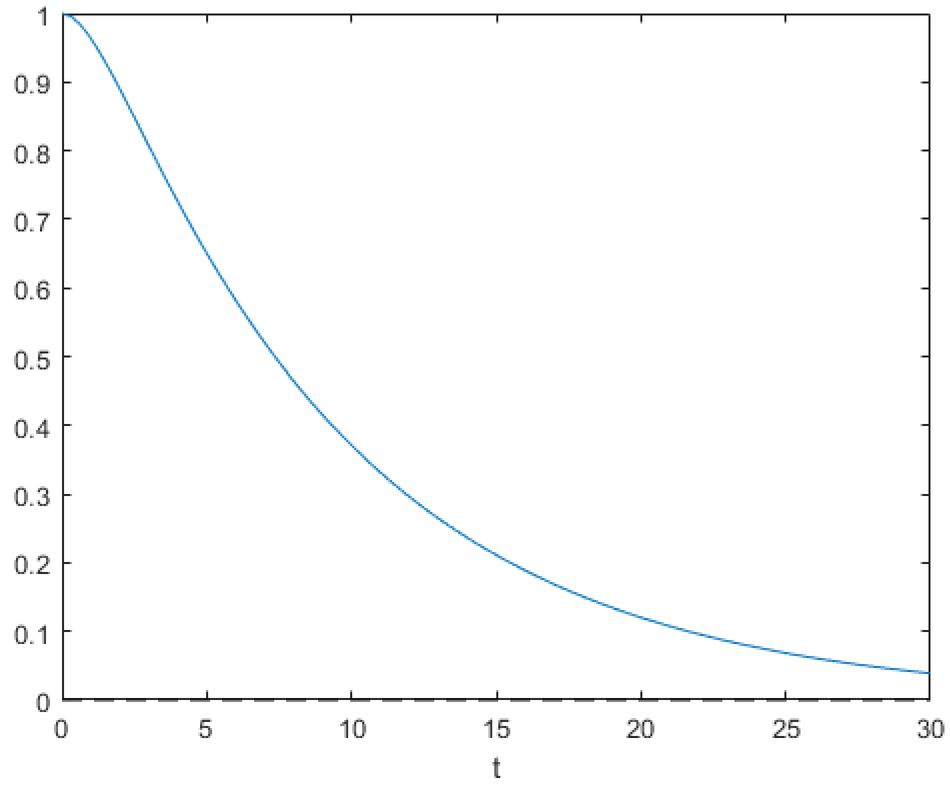
State Feedback in MATLAB

```
A = [0 1;0 0];
B = [0 1]';
C = [1 0];
D = 0;
G = ss(A,B,C,D);

k1 = 1;
k2 = 1;
K = [k1 k2];
Gcl = ss(A-B*K,B,C,D);

x0 = [1 0]';
t = linspace(0,30,100);
r = zeros(size(t));
[y,tout] = lsim(Gcl,r,t,x0);
plot(tout,y,tout,zeros(size(tout)),'k--'), xlabel('t')
```



State Feedback in MATLAB

```
A = [0 1;0 0];
B = [0 1]';
C = [1 0];
D = 0;
G = ss(A,B,C,D);

k1 = 0.1;
k2 = 1;
K = [k1 k2];
Gcl = ss(A-B*K,B,C,D);

x0 = [1 0]';
t = linspace(0,30,100);
r = zeros(size(t));
[y,tout] = lsim(Gcl,r,t,x0);
plot(tout,y,tout,zeros(size(tout)),'k--'), xlabel('t')
```



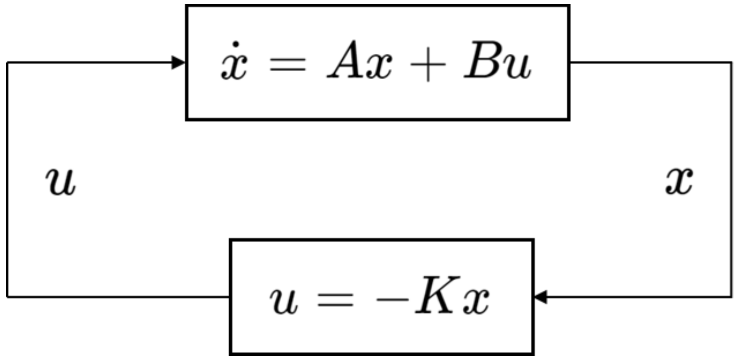
Pole Placement

- Back to the point-mass, again

$$u = -Kx \quad \rightarrow \quad \dot{x} = (A - BK)x$$

$$A - BK = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}$$

$$\begin{vmatrix} -\lambda & 1 \\ -k_1 & -k_2 - \lambda \end{vmatrix} = \lambda^2 + \lambda k_2 + k_1$$



Pole Placement

- Desired eigenvalues: let's pick both eigenvalues at -1

$$(\lambda + 1)(\lambda + 1) = \lambda^2 + 2\lambda + 1$$

$$k_1 = 2, k_2 = 1$$

- Pick the control gains such that the eigenvalues (poles) of the closed loop system match the desired eigenvalues
  - Questions: is this always possible? (No)

- How should we pick the eigenvalues? (Mix of art and science)
  - No clear-cut answer
  - The "smallest" eigenvalue dominates the convergence rage
  - The bigger eigenvalues, the bigger control gains/signals

Example

$$\dot{x} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$

$$A - BK = \begin{bmatrix} 2 - k_1 & -k_2 \\ 1 - k_1 & 1 - k_2 \end{bmatrix}$$

$$\varphi = \lambda^2 + \lambda(-3 + k_1 + k_2) + 2 - k_1 - k_2$$

- Let's pick both eigenvalues at -1

$$\varphi = (\lambda + 1)^2 = \lambda^2 + \lambda(-3 + k_1 + k_2) + 2 - k_1 - k_2$$

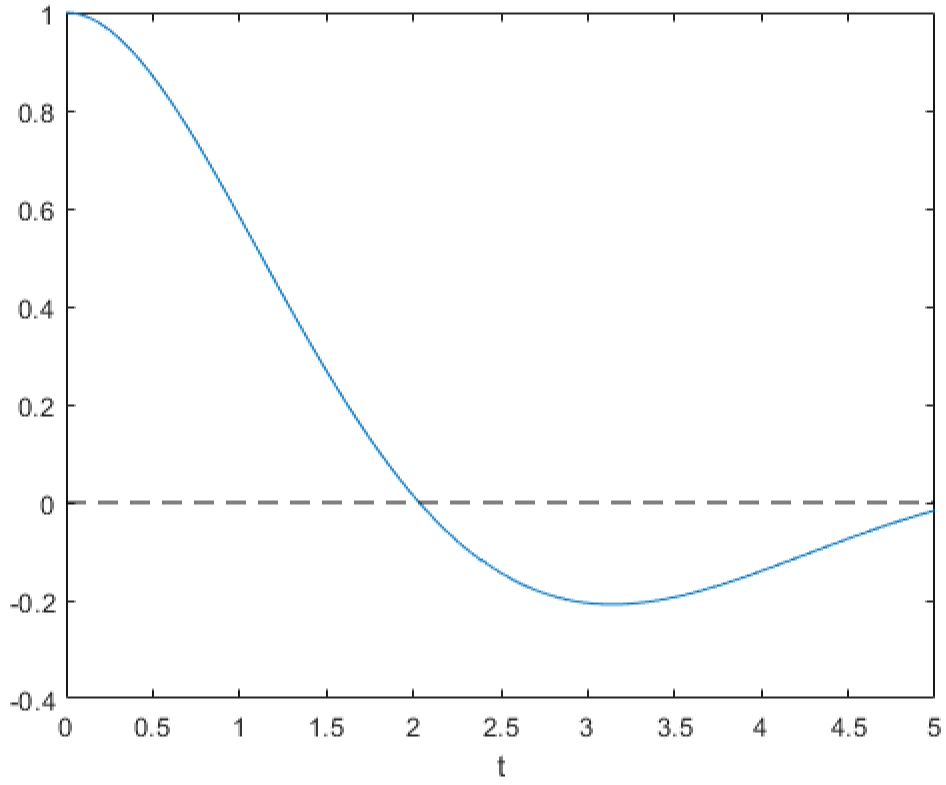
$$-3 + k_1 + k_2 = 2 \quad \text{and} \quad 2 - k_1 - k_2 = 1$$

$\rightarrow$ no k_1 and k_2 exist

- What's at play here is a lack of controllability, i.e., the effect of the input is not sufficiently rich to influence the system enough

Pole Placement in MATLAB

```
A = [2 0;
     1 -1];

B = [1 1]';
C = [1 0];

P = [-0.5 + 1j, -0.5 - 1j];
%P = [-0.1 + 1j, -0.1 - 1j];
%P = [-0.5, -1];
%P = [-5, -4];

K = place(A,B,P)
```



Controllability

- When can we place the eigenvalues using state feedback?

- When is B matrix (the actuator configuration) rich enough so that we can make the system do whatever we want it to do?
- The answer revolves around the concept of controllability

- The system $\dot{x} = Ax + Bu$ is controllable if there exists a control $u(t)$ that will take the state of the system from any initial state x_0 to any desired final state x_f in a finite time interval

- Given a discrete-time system

$$x_{k+1} = Ax_k + Bu_k$$

Controllability

- Given a discrete-time system

$$x_{k+1} = Ax_k + Bu_k$$

- We would like to drive this system in n steps to a particular target state x^*

$$\begin{cases} x_1 = Ax_0 + Bu_0 = Bu_0 \\ x_2 = Ax_1 + Bu_1 = ABu_0 + Bu_1 \\ x_3 = Ax_2 + Bu_2 = A^2Bu_0 + ABu_1 + Bu_2 \\ \quad \vdots \\ x_n = A^{n-1}Bu_0 + \cdots + Bu_{n-1} \end{cases}$$

- We want to solve

$$x^* = [\, B \quad AB \quad \cdots \quad A^{n-1}B \,] \begin{bmatrix} u_{n-1} \\ \vdots \\ u_1 \\ u_0 \end{bmatrix}$$

- The system (A, B) is controllable if and only if C has full row rank

$$\text{rank}\,([\, B \quad AB \quad \cdots \quad A^{n-1}B \,]) = n$$

Controllability

$$\operatorname{rank}\left(\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}\right) = n$$

$$\dot{x} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

Controllability in MATLAB

- ctrb(A, B) is the MATLAB function to form a controllability matrix, C

$$\dot{x} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$

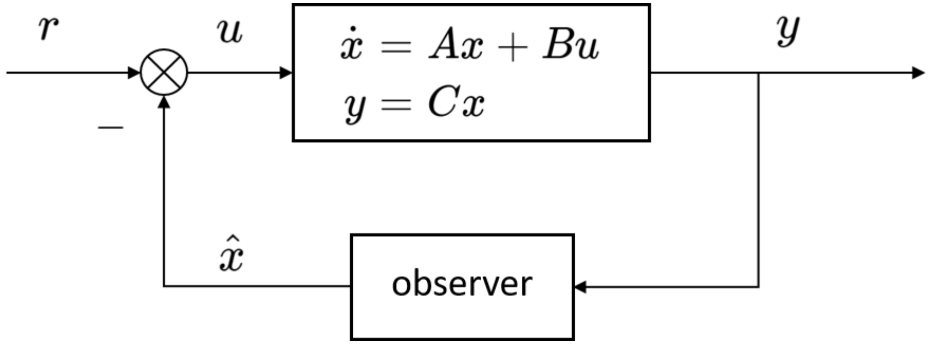
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

```
A = [2 0;
     1 1];
B = [1 1]';

G = ctrb(A,B)
rank(G)
```

```
A = [0 1;
     0 0];
B = [0 1]';

G = ctrb(A,B)
rank(G)
```

Observer

- We now know how to design rather effective controllers using state feedback.
- But what about y ?



- The predictor-corrector (observer)
  - Assume $B = 0$ or
  - Assume that we are aware of B and u

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

  - Make a copy of the system
  - Add a notion of how wrong your estimate is to the model

Observer

- The predictor-corrector (observer)

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

  – Make a copy of the system

$$\dot{\hat{x}} = A\hat{x} \quad \text{predictor}$$

  – Add a notion of how wrong your estimate is to the model

$$\dot{\hat{x}} = A\hat{x} + \underbrace{L\left(y - C\hat{x}\right)}_{\text{corrector}}$$

- What we want to stabilize (drive to zero) is the estimation error, i.e., the difference between the actual state and the estimated state $e = x - \hat{x}$

$$\dot{e} = \dot{x} - \dot{\hat{x}} = Ax - A\hat{x} - L\left(y - C\hat{x}\right)$$

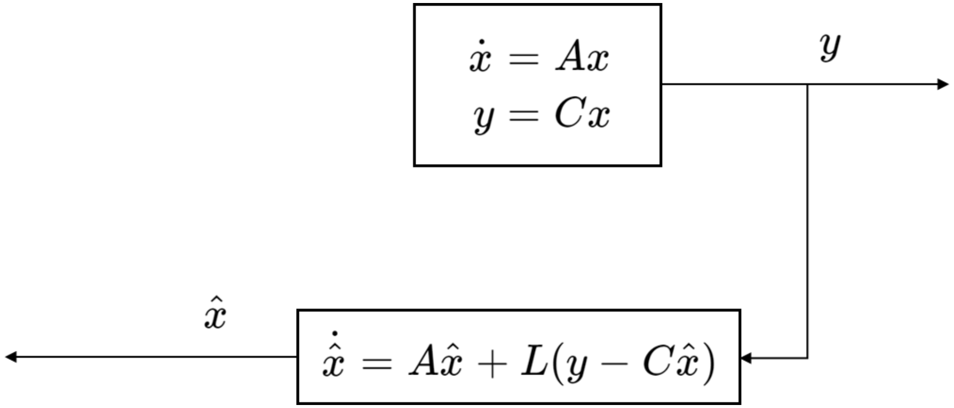
$$= A\left(x - \hat{x}\right) - LC\left(x - \hat{x}\right) = \left(A - LC\right) e$$

Observer

- Just pick L such that the eigenvalues to $A - LC$ have negative real part !!!

$$\mathrm{Re}\left(\mathrm{eig}(A - LC)\right) < 0$$

- We already know how to do this → Pole-placement



- Does this always work?
  – No

Observability

- Need to redo what we did for control design to understand when we can recover the state from the output

- The system is observable if, for any $x(0)$, there is a finite time τ such that $x(0)$ can be determined from $u(t)$ and $y(t)$ for $0 \leq t \leq \tau$

- Given a discrete time system without inputs

$$
\begin{aligned}
x_{k+1} &= Ax_k \\
y_k &= Cx_k
\end{aligned}
$$

Observability

- Can we recover the initial condition by collecting n output values?

$$\begin{aligned} x_{k+1} &= Ax_k \\ y_k &= Cx_k \end{aligned}$$

$$\begin{aligned} y_0 &= Cx_0 \\ y_1 &= Cx_1 = CAx_0 \\ &\vdots \\ y_{n-1} &= CA^{n-1}x_0 \end{aligned}$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}}_{} x_0$$

- The system (A, C) is observable if and only if R has full column rank

- The initial condition can be recovered from the outputs when the so-called observability matrix has full rank.

POSTECH

Observability

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

$$\dot{x} = \begin{bmatrix} 1 & 1 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x$$

Observability in MATLAB

- obsv(A, C) is the MATLAB function to form a observability matrix

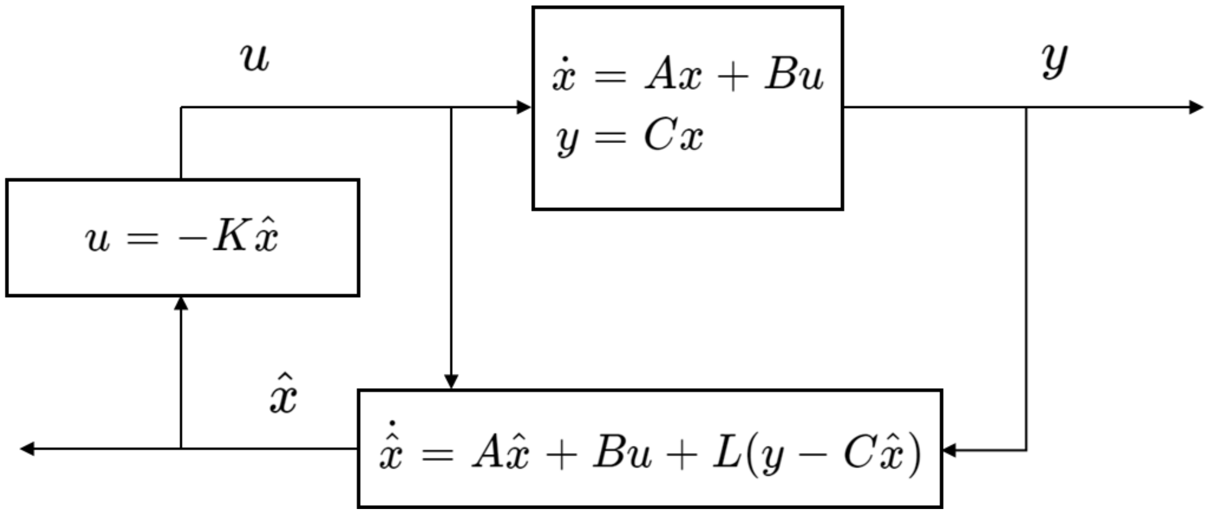
```
A = [1 1;
     4 -2];
C = [1 0;
     0 1];

ob = obsv(A,C)
rank(ob)
```

$$\dot{x} = \begin{bmatrix} 1 & 1 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x$$

Now, How Do We Put Everything Together ?

- Step 1) Design the stat feedback controller as if we had x (which we don't)
- Step 2) Estimate x using an observer (that now also contains u)



Now, How Do We Put Everything Together ?

- Step 1) Design the stat feedback controller as if we had x (which we don't)

$$u = -Kx \qquad \Longrightarrow \quad \dot{x} = (A - BK)x \quad \text{what we design for}$$

$$u = -K\hat{x} \quad \text{what we actually have}$$

- Step 2) Estimate x using an observer (that now also contains u)

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

$$\Longrightarrow \quad \dot{e} = (A - LC)\,e, \qquad (e = x - \hat{x})$$

The Separation Principle

- Want both x and e to be stabilized ($r = 0$)

$$\dot{x} = Ax - BK\hat{x} = Ax - BK(x - e) = (A - BK)x + BKe$$

$$\dot{e} = (A - LC)\, e$$

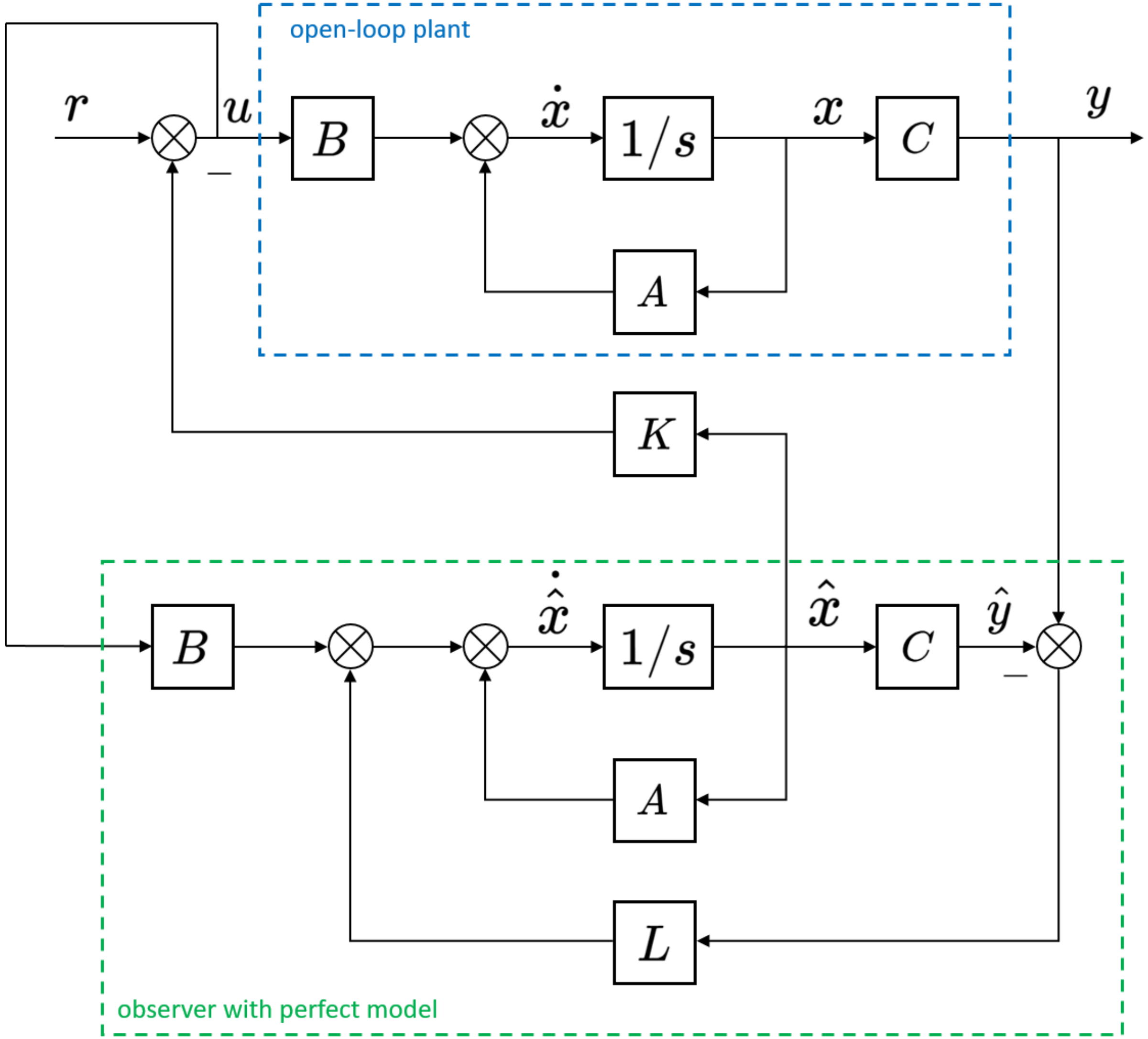
$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}}_{M} \begin{bmatrix} x \\ e \end{bmatrix}$$

- This is an (upper) triangular block matrix
  – Its eigenvalues are given by the eigenvalues of the diagonal blocks !

- (The Separation Principle) Design K and L independently to satisfy

$$\mathrm{Re}\,(\mathrm{eig}(A - BK)) < 0, \quad \mathrm{Re}\,(\mathrm{eig}(A - LC)) < 0$$

Everything in Block Diagram



$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}}_{M} \begin{bmatrix} x \\ e \end{bmatrix}$$