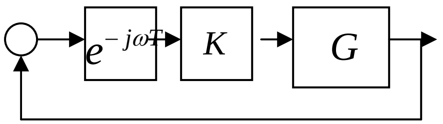
# Today (11/29/05)

- Today
  - Project/homework
  - Tracking control
  - Discretization of observer

# Homework (due 11/29/05)

- Put the 2-link system in the project in state space form.
- Linearize the system.  Find poles of the system.
- Obtain $\tau_1 \rightarrow \theta_1$ and $\tau_2 \rightarrow \theta_2$ transfer functions.
- Consider the following the design spec: closed loop stability, rise time about ½ sec, 2% settling time, less than 1 sec, no more than 5% overshoot, 6dB gain margin (gain scaling from ½ to 2) and 10ms delay tolerance.
  - Use root locus to design a controller to achieve the design spec.
  - Assume full state is available (e.g., by using the washout filter). Use state space pole place design method to achieve the design spec.
- Simulate the linear and nonlinear system responses using the above controllers.

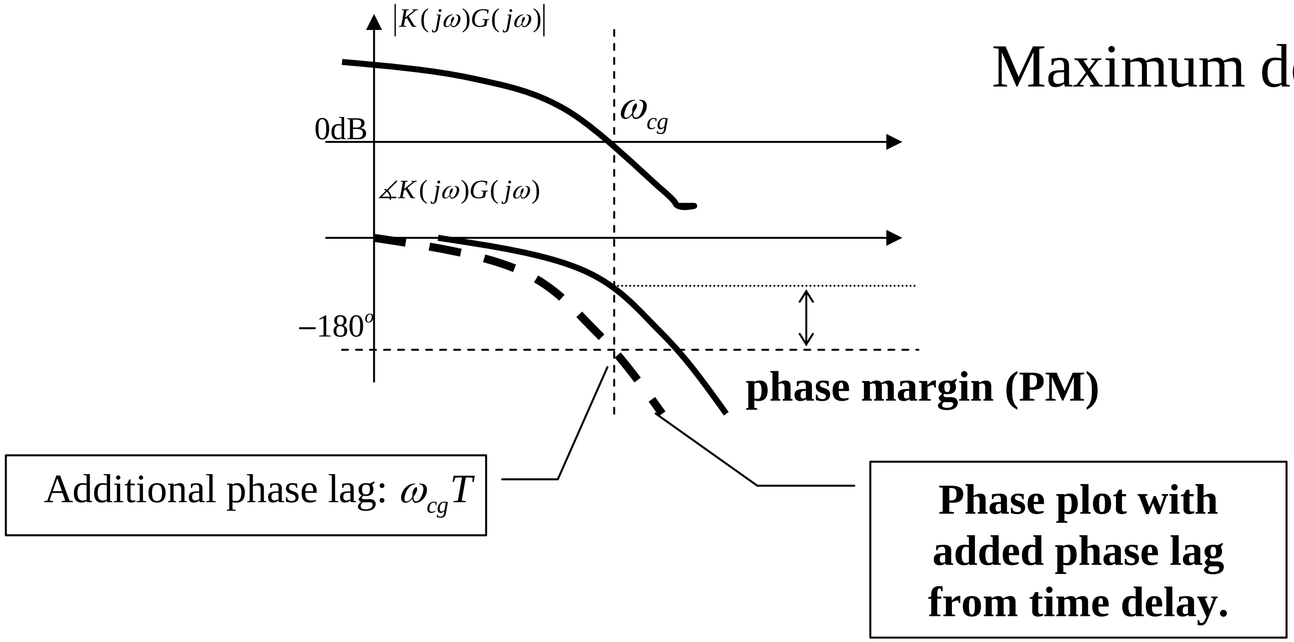
# Time Delay



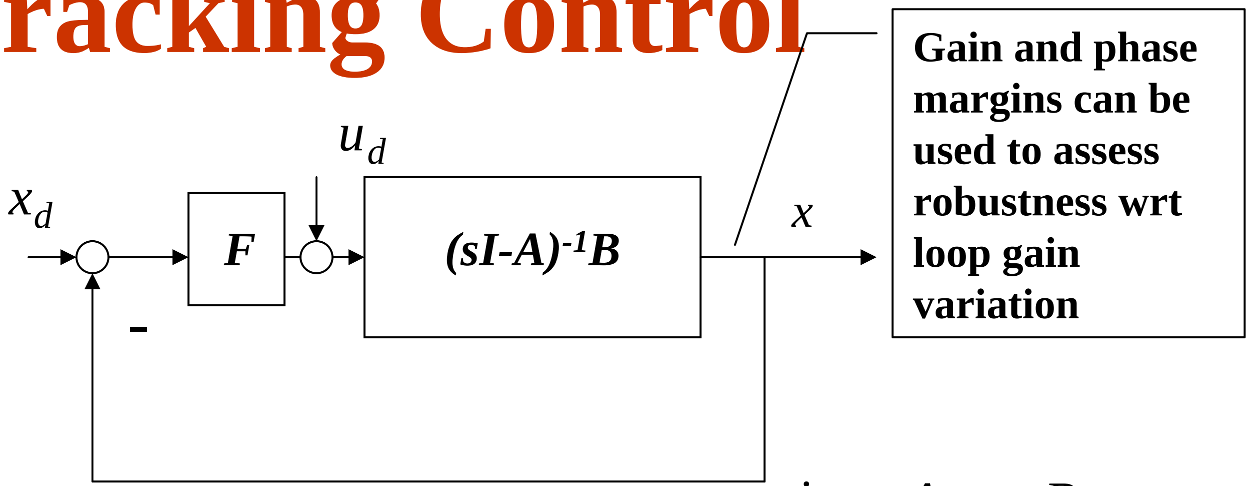

**Time delay adds a phase shift of -$\omega T$.**

Boundary of stability: PM=$\omega_{cg}T$

Maximum delay: $T_{\max} = \dfrac{\text{PM}}{\omega_{cg}}$



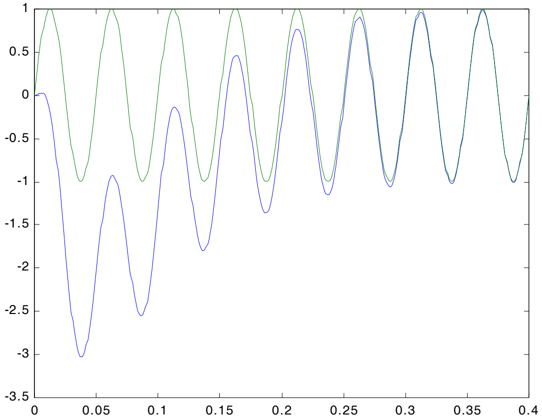
# Tracking Control

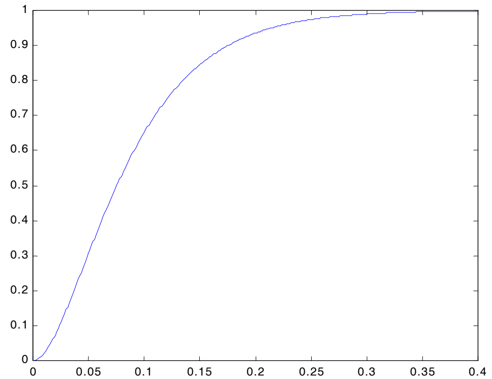



$$u = -F(x - x_d) + u_d$$
$$= -Fx + (Fx_d + u_d)$$

$$\dot{x}_d = Ax_d + Bu_d$$
$$y_d = Cx_d + Du_d$$

- Tracking: Given $y_d$, find $(x_d, u_d)$ such that
- Set Point: $\begin{bmatrix} x_d \\ u_d \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ y_d \end{bmatrix}$



# Tracking Control

Example: $G(s) = 1/(s^2 + 2.427s + 1)$

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2.427 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Set point control:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_d \\ u_d \end{bmatrix} = \begin{bmatrix} 0 \\ y_d \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 & 1 & 0 \\ -1 & -2.427 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{d_1} \\ x_{d_2} \\ u_d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ y_d \end{bmatrix}$$

$$x_{d_1} = y_d, \; x_{d_2} = 0, \; u_d = y_d$$

Tracking control:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_d \\ u_d \end{bmatrix} = \begin{bmatrix} \dot{x}_d \\ y_d \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 & 1 & 0 \\ -1 & -2.427 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{d_1} \\ x_{d_2} \\ u_d \end{bmatrix} = \begin{bmatrix} \dot{x}_{d1} \\ \dot{x}_{d2} \\ y_d \end{bmatrix}$$

$$y_d = A \sin \omega t$$

$$x_{d1} = A \sin \omega t$$

$$x_{d2} = \dot{x}_{d1} = A\omega \cos \omega t$$

$$u_d = x_{d1} + 2.427 x_{d2} + \dot{x}_{d2}$$

$$= A \sin \omega t + 2.427 A\omega \cos \omega t - A\omega^2 \sin \omega t$$

# What about the closed loop?

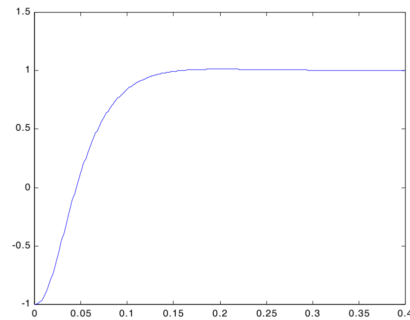
## Separation Theorem:

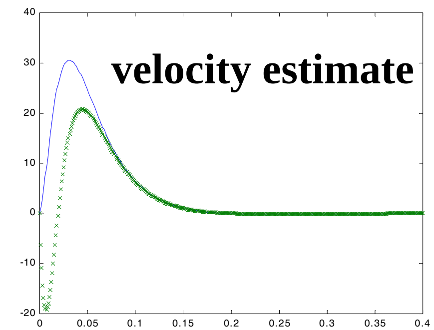
Replace *x* in full state feedback by state estimate

$$u = -F\hat{x}$$

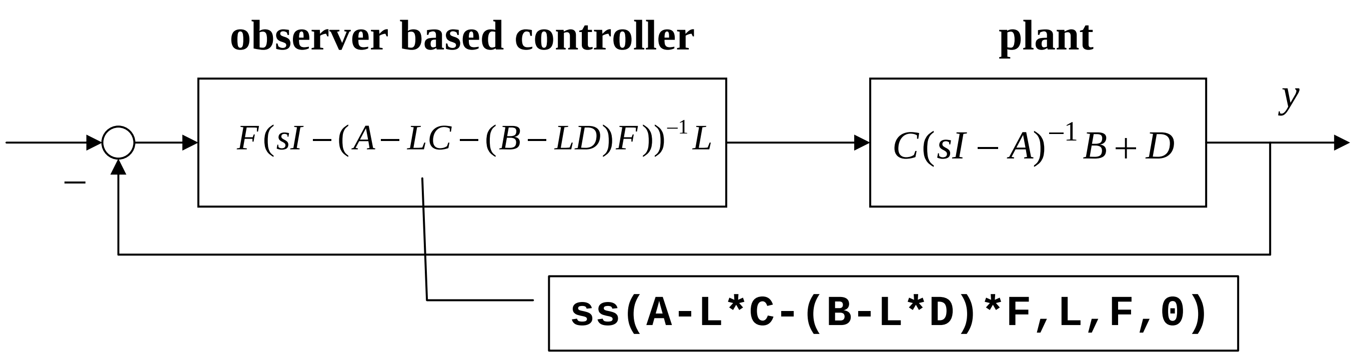
$$\dot{\hat{x}} = A\hat{x} + Bu - L(C\hat{x} + Du - y)$$

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




Closed loop poles given by full state feedback poles eig(*A*-*BF*) and observer poles eig(*A*-*LC*)



# Discretization of Observer

Used in implementation since $A_d$, $B_d$ can be precomputed.

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

$$u = -F\hat{x}$$

$$\dot{\hat{x}} = A\hat{x} + \begin{bmatrix} B & I \end{bmatrix} \begin{bmatrix} u \\ L\left(y - (C\hat{x} + Du)\right) \end{bmatrix}$$

$$u = -F\hat{x}$$

$$\dot{\hat{x}} = \left(A - LC\right)\hat{x} + \begin{bmatrix} B & L \end{bmatrix} \begin{bmatrix} u \\ \left(y - Du\right) \end{bmatrix}$$

$$u = -F\hat{x}$$

```
[Ad,Bd]=c2d(A,[B,eye(n,n)]);
```

```
[Ad,Bd]=c2d(A-L*C,[B,L]);
```

$$\hat{x}_{k+1} = A_d\hat{x}_k + B_d \begin{bmatrix} u_k \\ L\left(y_k - (C\hat{x}_k + Du_k)\right) \end{bmatrix}$$

$$u_k = -F\hat{x}_k$$

$$\hat{x}_{k+1} = A_d\hat{x}_k + B_d \begin{bmatrix} u_k \\ L\left(y_k - Du_k\right) \end{bmatrix}$$

$$u_k = -F\hat{x}_k$$

# Discretization of Observer

$$\dot{\hat{x}} = \left(A - L(C - DF) - BF\right)\hat{x} + Ly$$
$$u = -F\hat{x}$$

```
[Ad,Bd]=c2d(A-L*(C-DF)-B*F,L);
```

$$\hat{x}_{k+1} = A_d \hat{x}_k + B_d y_k$$
$$u_k = -F\hat{x}_k$$

```
obsv=c2d(ss(A-L*(C-DF)-B*F,L,-F,0));
[Ad,Bd,Cd,Dd]=ssdata(obsv);
```

$$\hat{x}_{k+1} = A_d \hat{x}_k + B_d y_k$$
$$u_k = C_d \hat{x}_k + D_d y_k$$