Quadrotor Trajectory Tracking Control

An Introduction

UAVs

MEAer - Spring Semester – 2020/2021

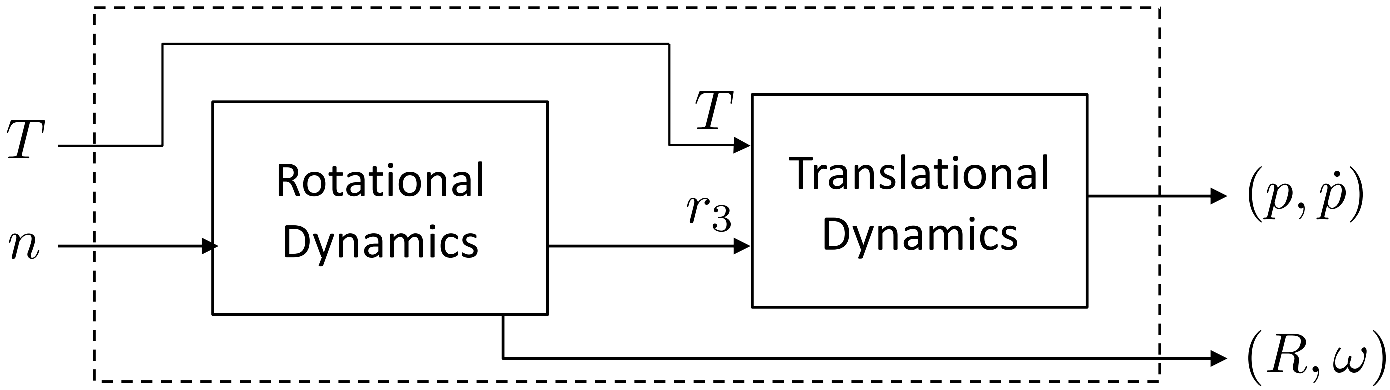
Rita Cunha

Quadrotor dynamic model

- Equations of motion

$$m\ddot{p} = -Tr_3 + mge_3,\ \ r_3 = Re_3$$
$$\dot{R} = RS(\omega)$$
$$J\dot{\omega} = -S(\omega)J\omega + n$$

- Block diagram

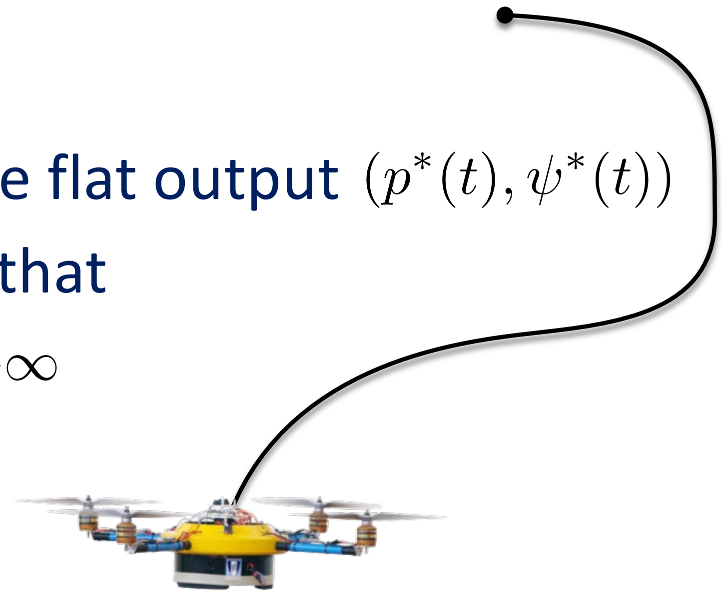


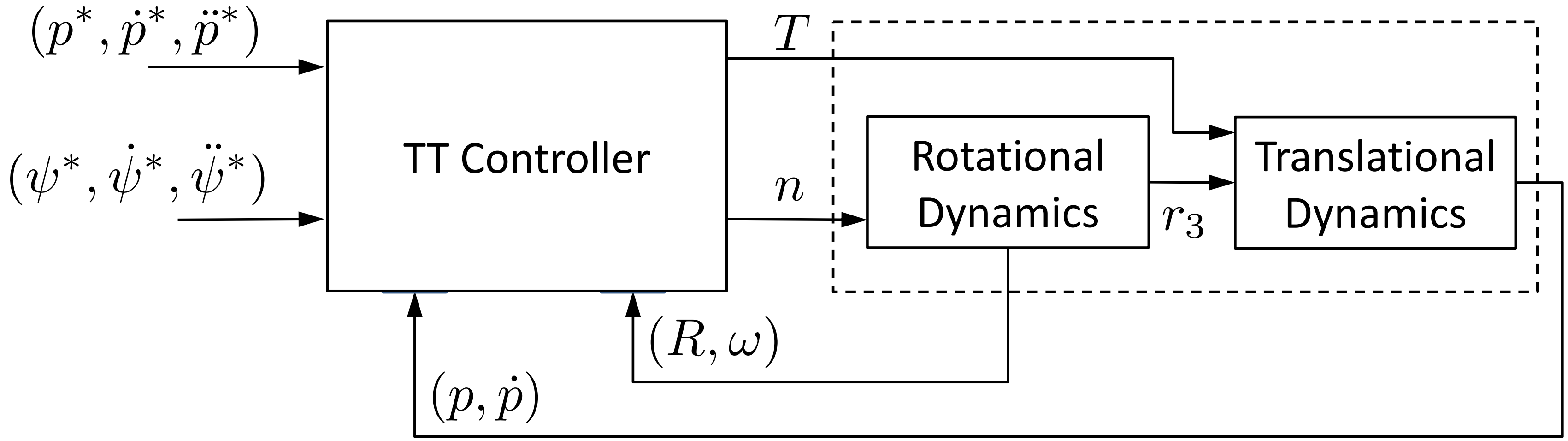
Feedback control for trajectory tracking

- Control objective:
  - Consider a desired trajectory for the flat output $(p^*(t), \psi^*(t))$

    Define a control law for (*T*, *n*) such that

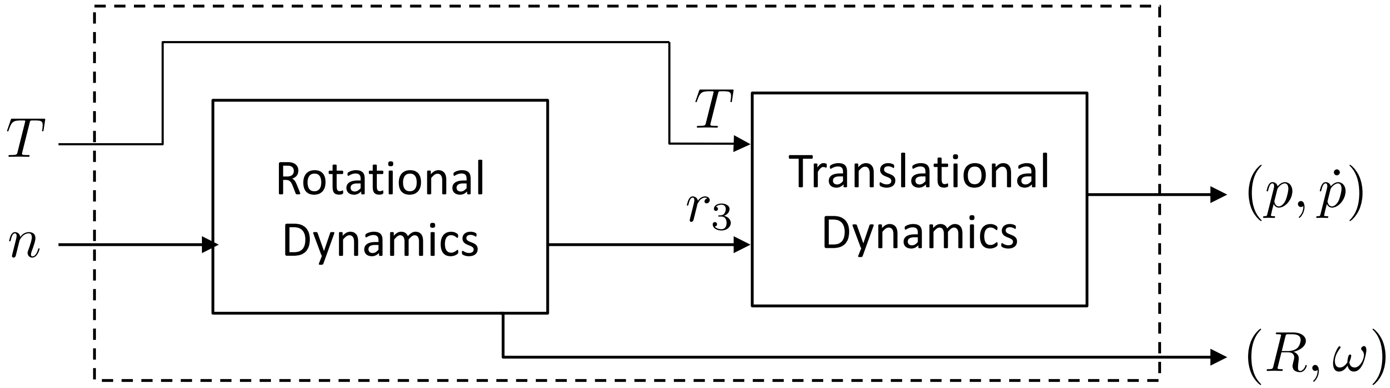
    $(p(t), \psi(t)) \to (p^*(t), \psi^*(t))$ as $t \to +\infty$





Hierarchical control

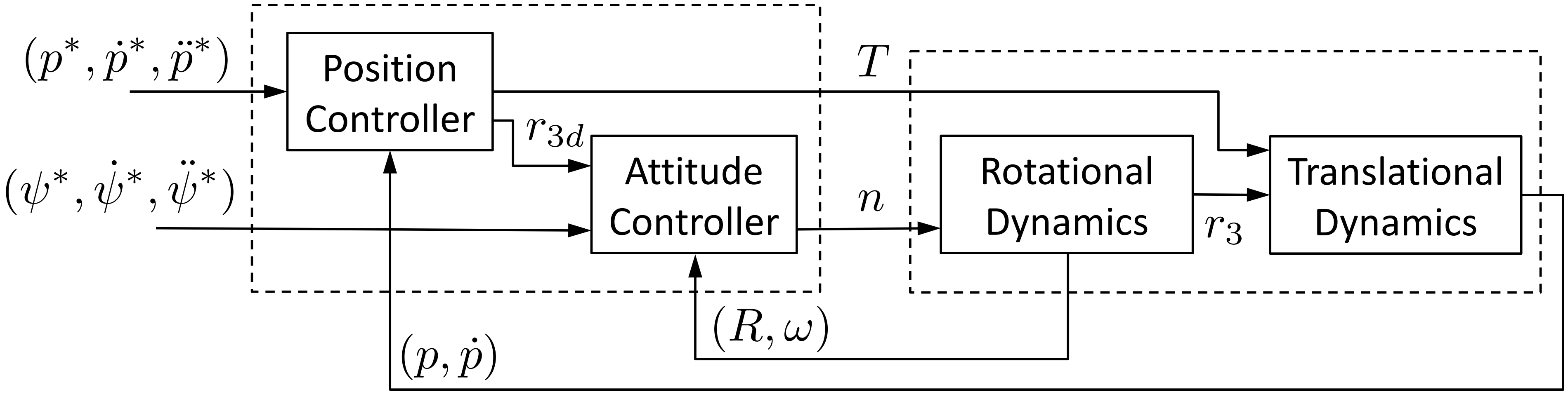
- Block dependencies suggest a hierarchical control structure



  - Outer-loop controller for translational dynamics determines the thrust T and desired orientation R_d
    - Goal: track desired position
  - Inner-loop controller for rotational dynamics determines the torque n
    - Goal: track desired orientation

Hierarchical control

- Inner-outer loop control structure



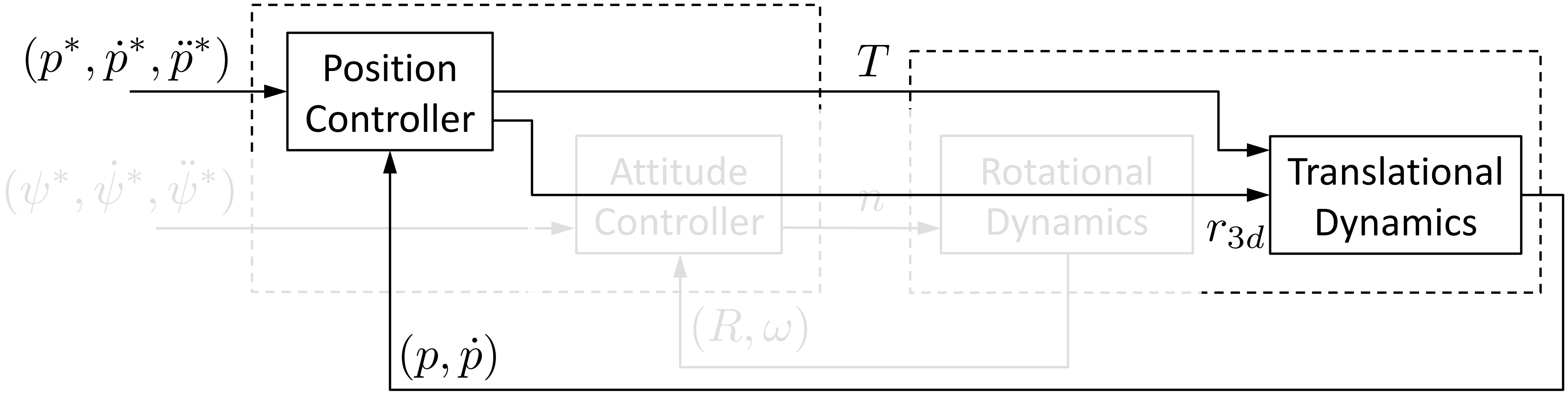
Hierarchical control

1. Neglect rotational dynamics
   - Replace r_3 by a desired value r_{3d}
   - Define a new virtual input for the translational dynamics $u_T = -\frac{T}{m} r_{3d}$
     T gives magnitude, r_{3d} gives direction

$$T = m\|u_T\|, \; r_{3d} = -\frac{u_T}{\|u_T\|}$$

2. Define a control law for u_T, assuming

$$\ddot{p} = u_T + g \qquad\qquad u_T = ?$$



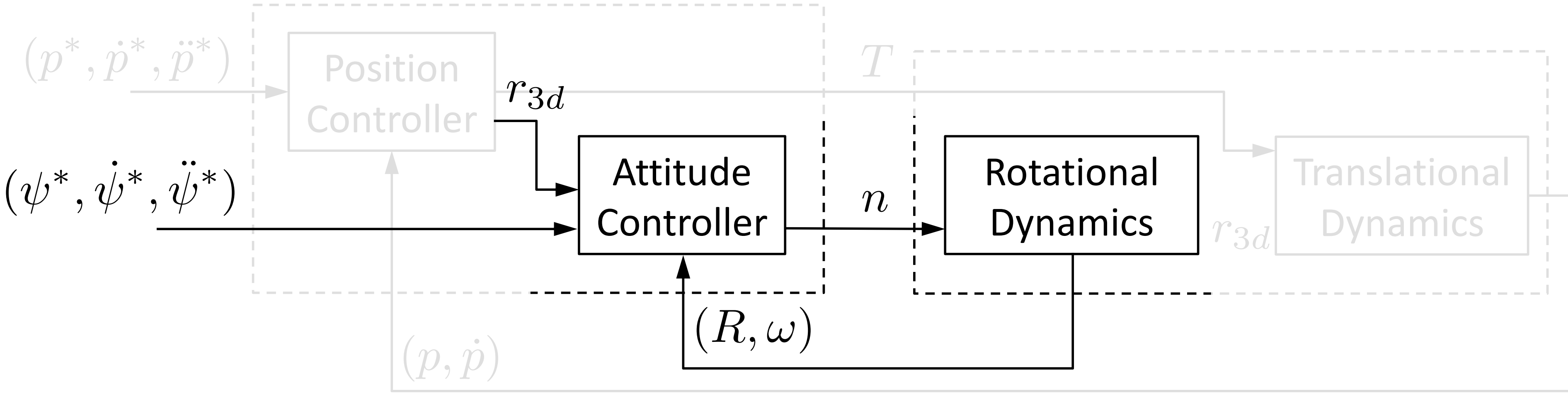
Hierarchical control

3. Combine r_{3d} with ψ^* to define a desired attitude R_d

$$R_d = f(r_{3d}, \psi^*) = ? \qquad \text{or} \qquad \begin{gathered} \lambda_d = (\phi_d, \theta_d, \psi^*) \\ (\phi_d, \theta_d) = g(r_{3d}, \psi^*) = ? \end{gathered}$$

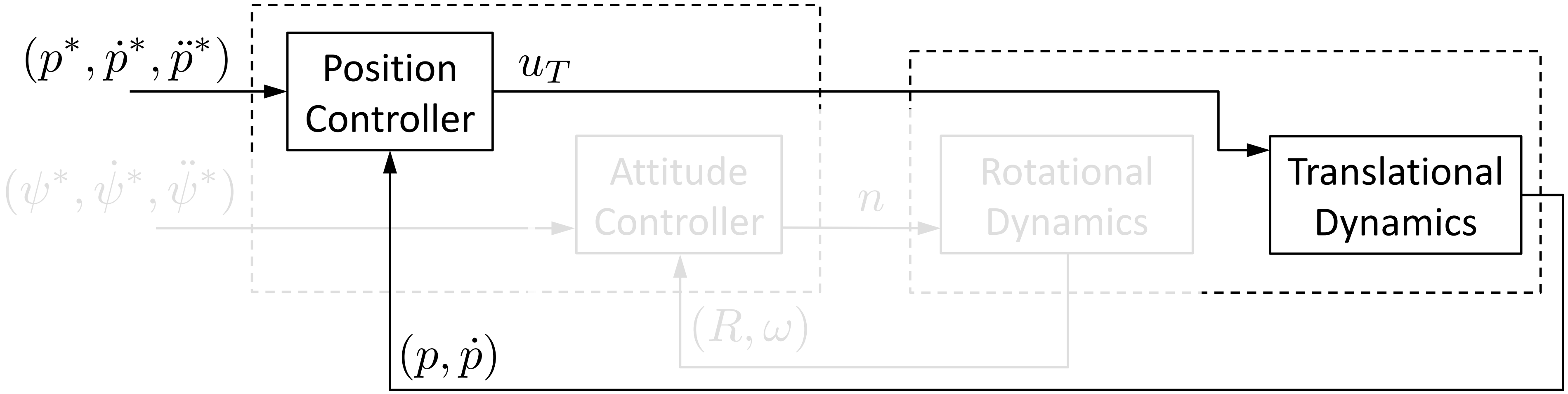
4. Define a control law for *n* to track the desired attitude R_d or λ_d

$$\begin{gathered} \dot{R} = RS(\omega) \\ J\dot{\omega} = -S(\omega)J\omega + n \end{gathered} \qquad \text{or} \qquad \begin{gathered} \dot{\lambda} = Q(\phi, \theta)\omega \\ J\dot{\omega} = -S(\omega)J\omega + n \end{gathered} \qquad n = ?$$



Trajectory Tracking Controller (position)

2. Define a control law for u_T to track a desired trajectory
   - Model $\ddot{p} = u_T + g$
   - Reference to be tracked $(p^*(t), \dot{p}^*(t), \ddot{p}^*(t))$
   - Control law $u_T + g = ?$



Trajectory Tracking Controller (position)

2. Define a control law for u_T to track a desired trajectory
   - Model

$$\ddot{p} = u_T + g$$

   - Reference to be tracked

$$(p^*(t), \dot{p}^*(t), \ddot{p}^*(t))$$

   - Error system

$$e(t) = p(t) - p^*(t), \quad \dot{e}(t) = \dot{p}(t) - \dot{p}^*(t)$$

$$\ddot{e}(t) = u_T(t) + g - \ddot{p}^*(t)$$

   - Control law

$$u_T(t) = -k_P e(t) - k_D \dot{e}(t) - g + \ddot{p}^*(t)$$

   - Closed-loop error system

$$\ddot{e}(t) = -k_P e(t) - k_D \dot{e}(t)$$

Trajectory Tracking Controller (position)

2. Define a control law for u_T to track a desired trajectory

– Closed-loop error system

$$\ddot{e}(t) = -k_P e(t) - k_D \dot{e}(t)$$

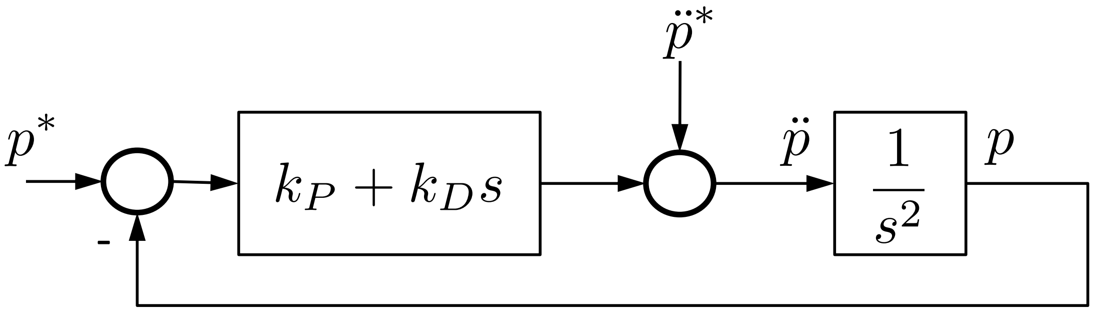
– Stability

$$k_P > 0,\ k_D > 0 \longrightarrow \text{Asymptotic Stability}$$

– Performance Specifications
  - Settling time
  - Overshoot
  - Disturbance rejection
  - Noise attenuation
  - Gain and phase margin

Trajectory Tracking Controller (position)

- Classical SISO system analysis



  - Closed-loop system in time domain

$$\ddot{p} = \underbrace{-k_P(p - p^*) - k_D(\dot{p} - \dot{p}^*)}_{\text{feedback}} + \underbrace{\ddot{p}^*}_{\text{feedforward}}$$

  - Closed-loop transfer function

$$\frac{P(s)}{P^*(s)} = \frac{k_P + k_D s}{s^2 + k_D s + k_P} \quad \text{compare with} \quad G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

Classical SISO system analysis

– Closed-loop transfer function

$$\frac{P(s)}{P^*(s)} = \frac{k_P + k_D s}{s^2 + k_D s + k_P} \quad \text{compare with} \quad G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

– Stability: poles with negative real part

$$k_P > 0, \;\; k_D > 0$$

Spring effect

Damping effect

– Decrease overshoot (increase damping coefficient)

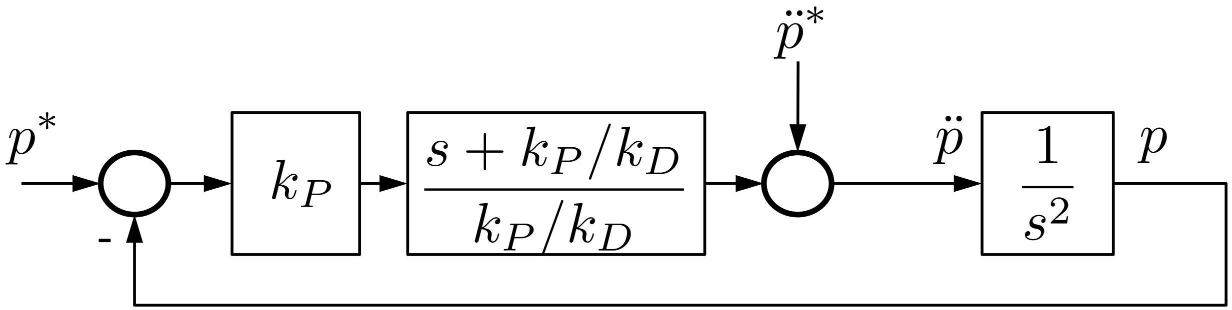
$$\xi = k_D/(2\sqrt{k_P}) \qquad k_D \uparrow \Rightarrow \xi \uparrow$$

$$(k_D \geq 2\sqrt{k_P}) \Rightarrow (\xi \geq 1) \qquad k_P \uparrow \Rightarrow \xi \downarrow$$

– Decrease settling time $\quad k_D \uparrow \Rightarrow \xi\omega_n \uparrow$

Classical SISO system analysis

- Root-locus



  - Closed-loop transfer function

$$\frac{P(s)}{P^*(s)} = \frac{k\frac{s+z}{s^2}}{1 + k\frac{s+z}{s^2}}$$

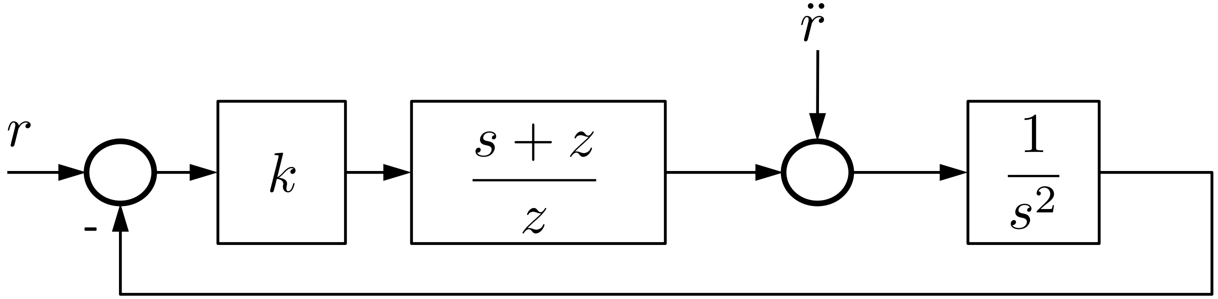
$$z = \frac{k_P}{k_D}, \quad k = k_P$$

  - Open-loop transfer function $G(s) = k\frac{s+z}{s^2}$

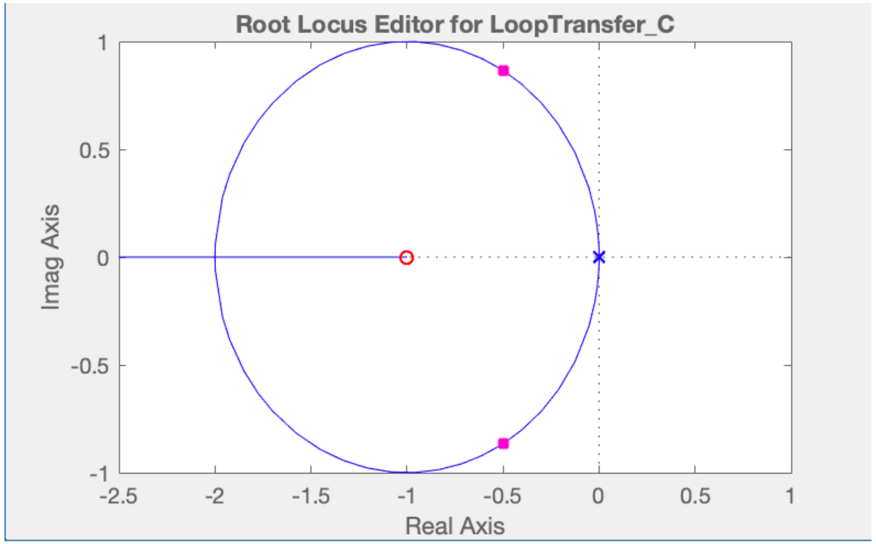
  - Characteristic equation $1 + k\frac{s+z}{s^2} = 0$

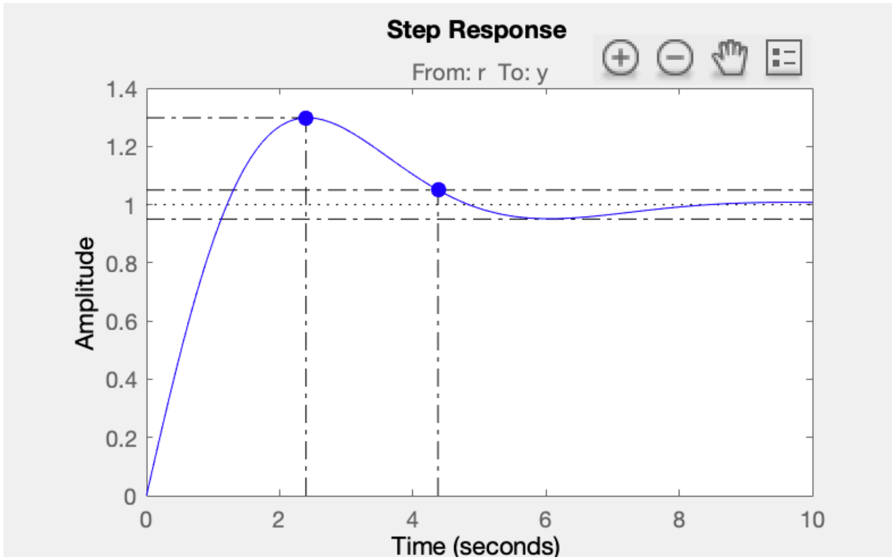
Classical SISO system analysis

- Root-locus



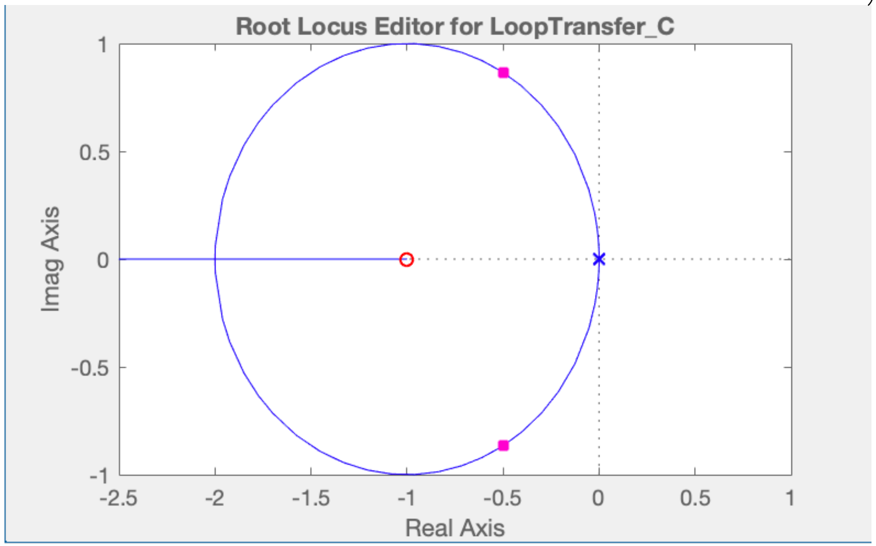
$$k = k_P = 1, \quad z = \frac{k_P}{k_D} = 1$$

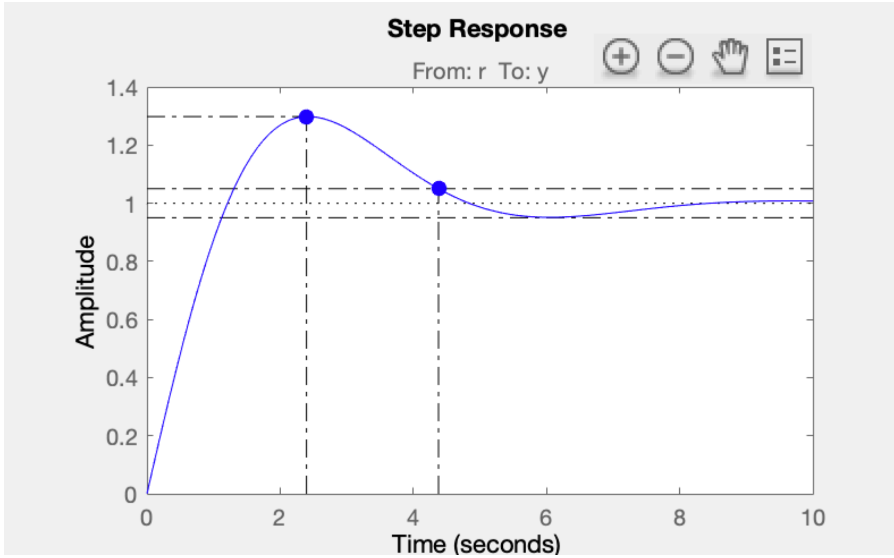




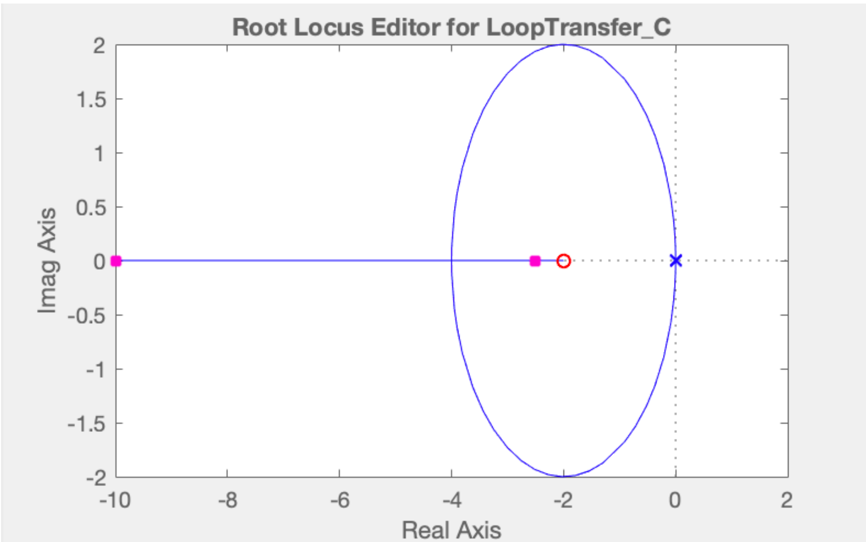
Control design based on Root-locus

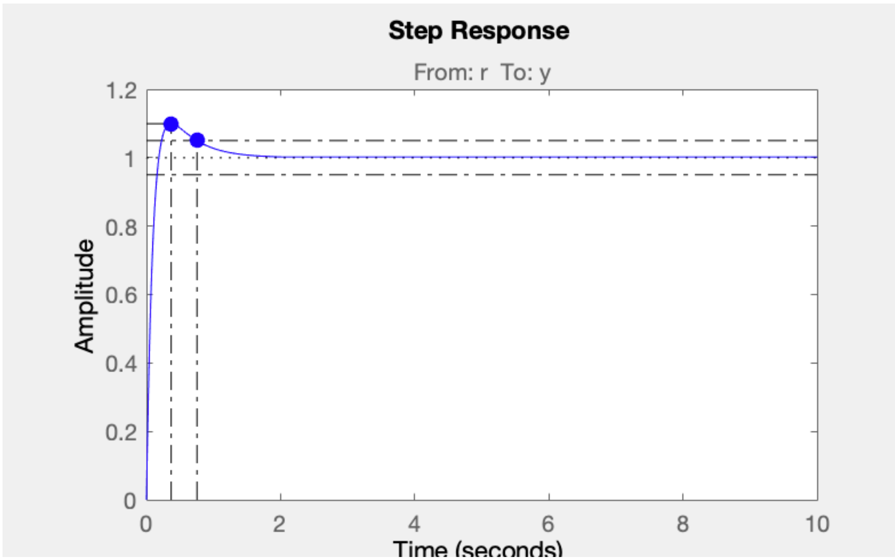
$$k = 1, \quad z = 1$$



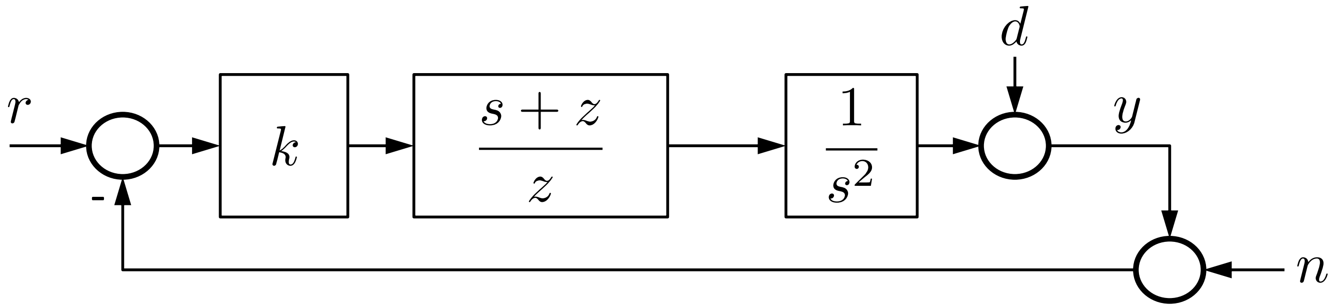


$$k = 24, \quad z = 2$$





Control design based on Loop-Shaping

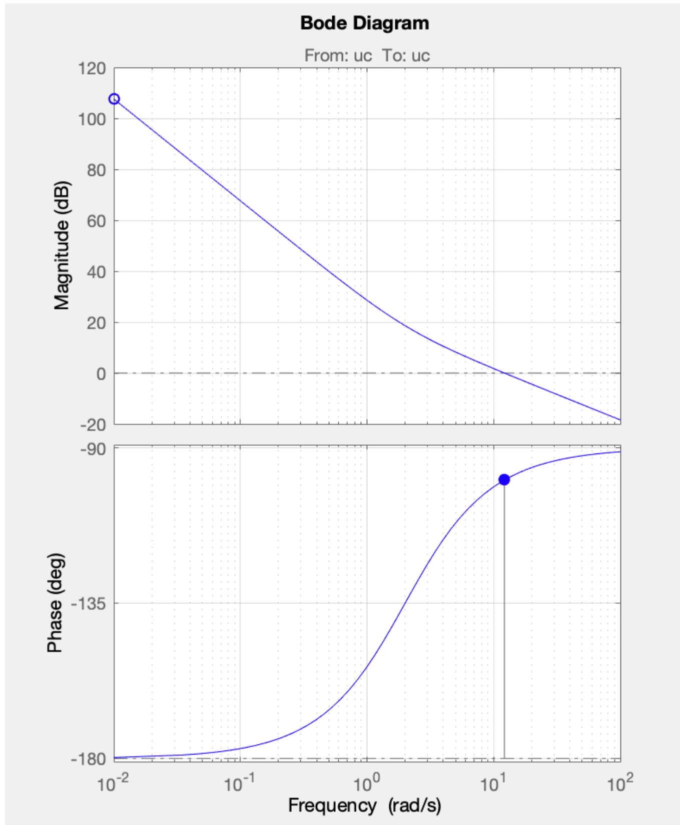


- Bode plot for open-loop transfer function
  - G_M = ?
  - P_M = ?
  - Noise and disturbance attenuation

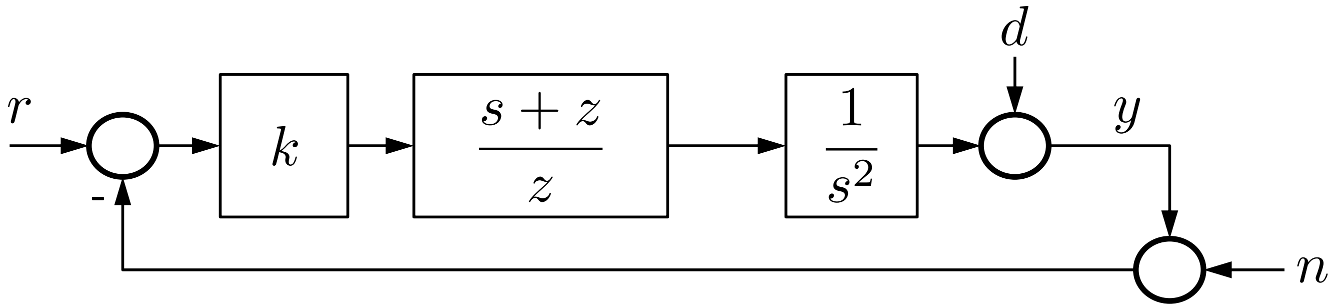
$$\left|\frac{Y(j\omega_n)}{N(j\omega_n)}\right| \leq ?,\ \omega_n \in [\bar{\omega}_n, \underline{\omega}_n]$$

$$\left|\frac{Y(j\omega_d)}{D(j\omega_d)}\right| \leq ?,\ \omega_d \in [\bar{\omega}_d, \underline{\omega}_d]$$

$$k = 24, \quad z = 2$$



Control design based on Loop-Shaping



– G_M = ?

$$G_M = +\infty$$

– P_M = ?

$$P_M = 80.7\,^{\circ}$$

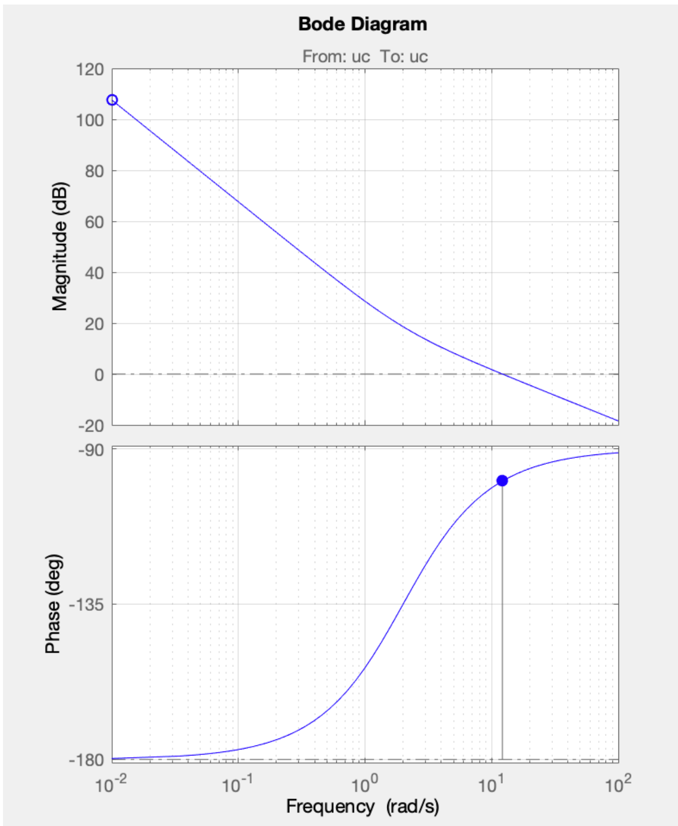
$$\omega_c = 12.2\,\mathrm{rad\ s}^{-1}$$

– Maximum tolerable time delay

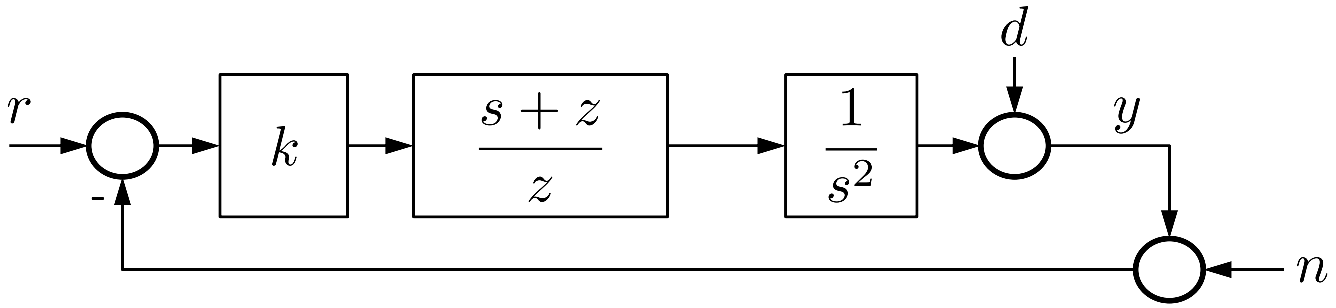
$$\tau\,\omega_c < \tau_c\,\omega_c = P_M$$

$$\tau < \tau_c = 0.116\,\mathrm{s}$$

$$k = 24, \quad z = 2$$



Control design based on Loop-Shaping

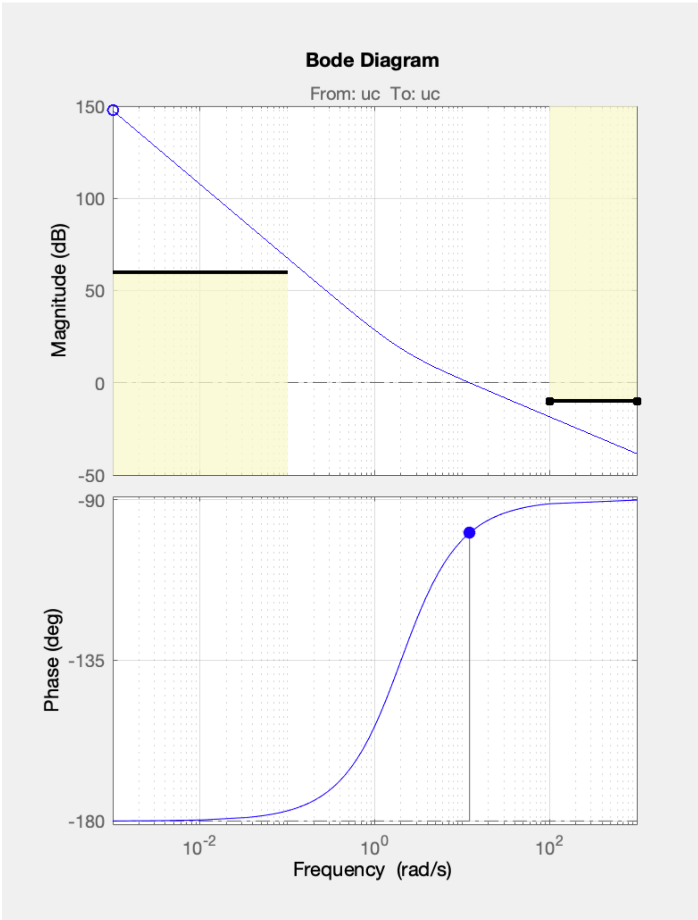


- Noise reduction

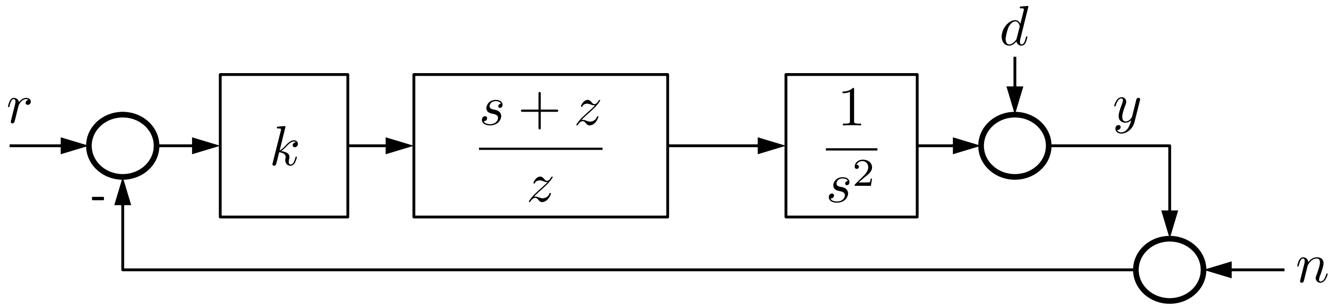
$$\left|\frac{Y(j\omega_n)}{N(j\omega_n)}\right|_{dB} \leq -10\,dB$$

$$|G(j\omega_n)|_{dB} \leq -10\,dB$$

$$\omega_n \in [10^2, 10^3]$$



Control design based on Loop-Shaping



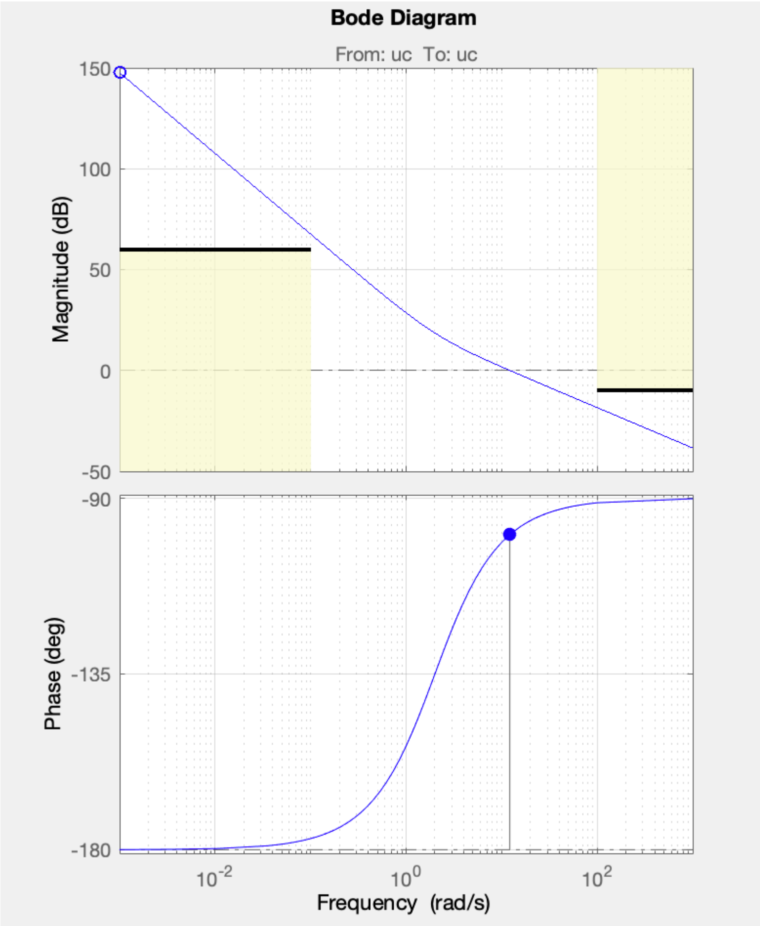
- Disturbance attenuation

$$\left|\frac{Y(j\omega_d)}{D(j\omega_d)}\right|_{dB} \leq -80\,dB$$

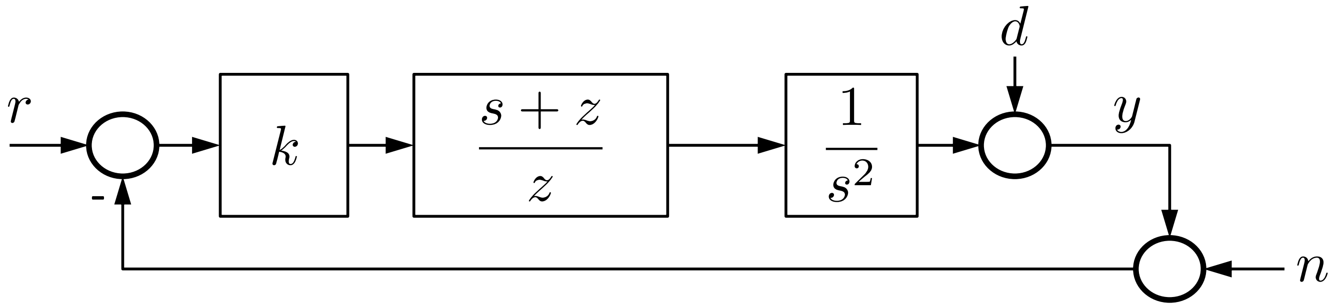
$$\downarrow$$

$$|G(j\omega_d)|_{dB} \geq 80\,dB$$

$$\omega_d \in [10^{-2}, 10^{-1}]$$



Control design based on Loop-Shaping



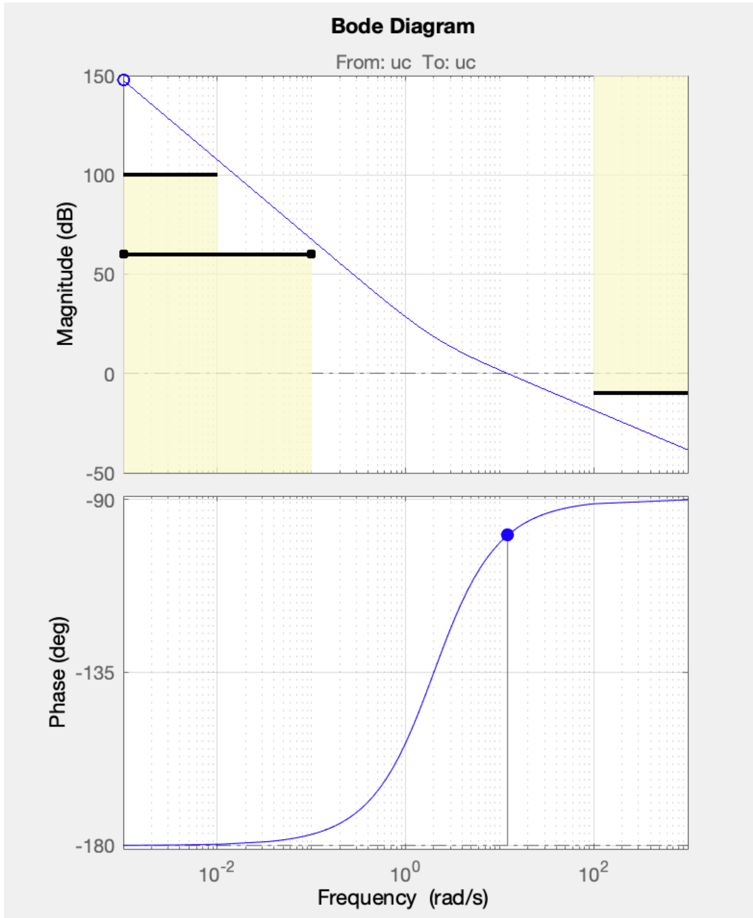
- Reference Following

$$\left|\frac{E(j\omega_d)}{R(j\omega_d)}\right|_{dB} \leq -100\,dB$$

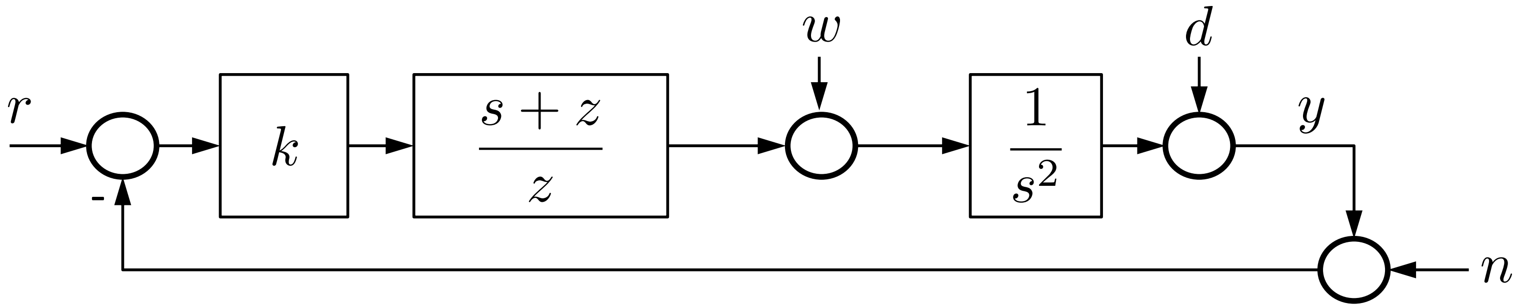
$$\downarrow$$

$$|G(j\omega_r)|_{dB} \geq 100\,dB$$

$$\omega_r \in [0, 10^{-2}]$$



Disturbance Rejection



- Wind disturbance rejection

wind disturbance

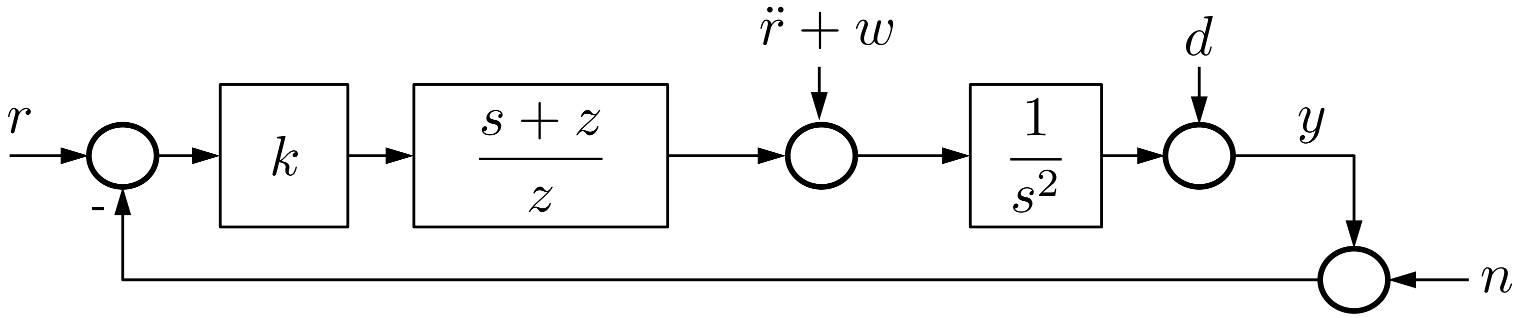
$$Y(s) = D(s) + \frac{1}{s^2}\left[W(s) + k(s+z)(R(s) - Y(s) - N(s))\right]$$

$$Y(s) = \frac{s^2}{s^2 + C(s)} D(s) + \frac{1}{s^2 + C(s)} W(s) + \frac{C(s)}{s^2 + C(s)} (R(s) - N(s))$$

$$\lim_{t \to \infty} y(t) = \lim_{s \to 0} sY(s)$$

- Does the system reject a constant wind disturbance ?

Disturbance Rejection



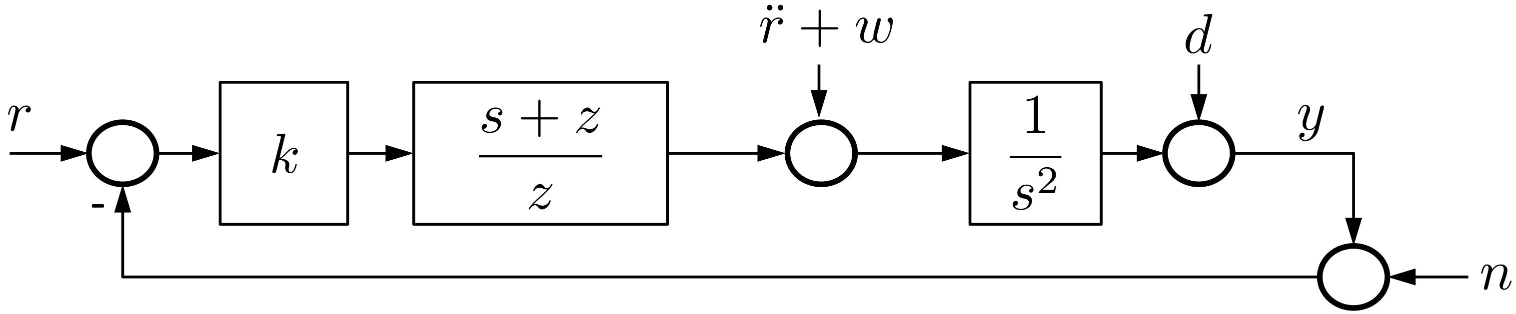
- Wind disturbance rejection
  - Does the system reject a constant wind disturbance?
  - No.

$$\frac{Y(s)}{W(s)} = \frac{1}{s^2 + C(s)}, \qquad W(s) = w_0/s$$

  - Application of the Final Value Theorem

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \frac{w_0}{C(0)} = \frac{w_0}{k}$$

Disturbance Rejection



- Wind disturbance rejection
  - Does the system reject a constant wind disturbance? No.
  - Also easy to see in the time domain

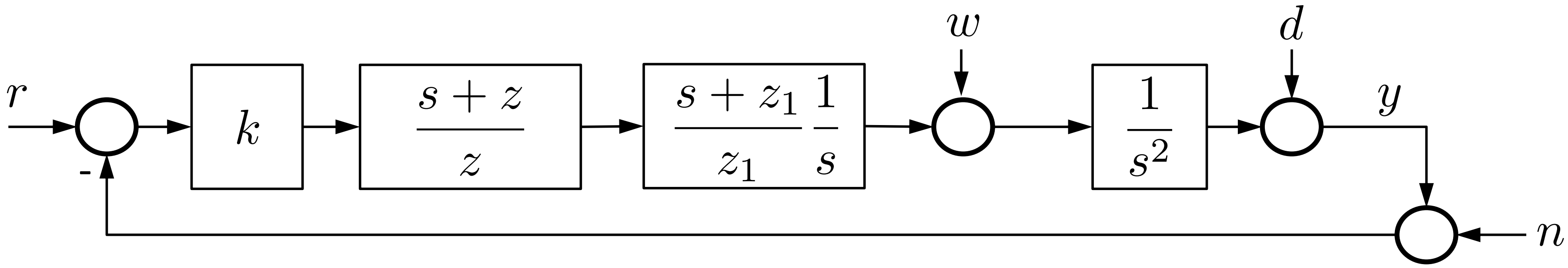
$$\ddot{y} + \frac{k}{z}(\dot{y} - \dot{r}) + k(y - r) = (\ddot{r} + w) + \ddot{d}$$

$$\ddot{e} + \frac{k}{z}\dot{e} + ke = w + \ddot{d}$$

  - Stable equilibrium point

$$e^* = \frac{w_0}{k} \Leftrightarrow y^*(t) = r(t) + \frac{w_0}{k}$$

Disturbance Rejection – Integral Action



- Wind disturbance rejection
  - Solution: add integral action -> PID controller

$$C(s) = k_P + \frac{k_I}{s} + k_D s = \frac{k_D s^2 + k_P s + k_I}{s}$$

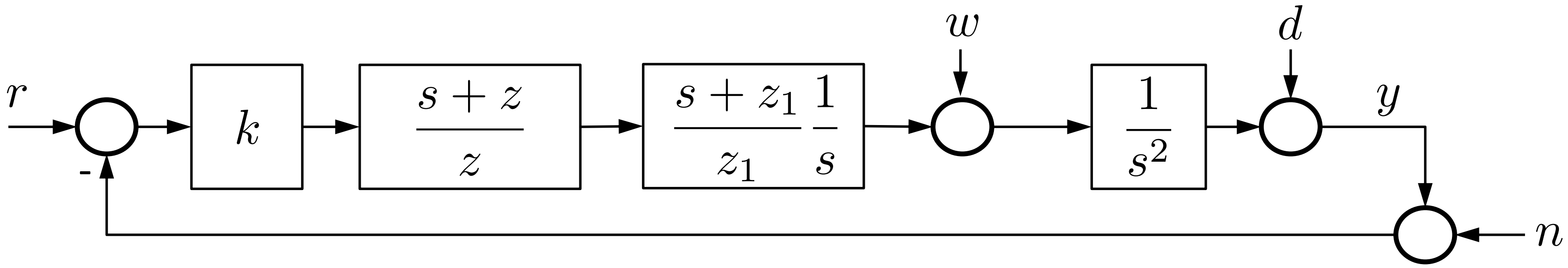
$$\frac{Y(s)}{W(s)} = \frac{1}{s^2 + C(s)}$$

$$= \frac{s}{s^2 + k_D s^2 + k_P s + k_I}$$

$$W(s) = \frac{w_0}{s}, \quad \lim_{t \to \infty} y(t) = \lim_{s \to 0} sY(s) = 0$$

Experimental results

Disturbance Rejection – Integral Action



- Strategy to go from PD to PID controller

$$C_{PD}(s) = k_P + k_D s$$

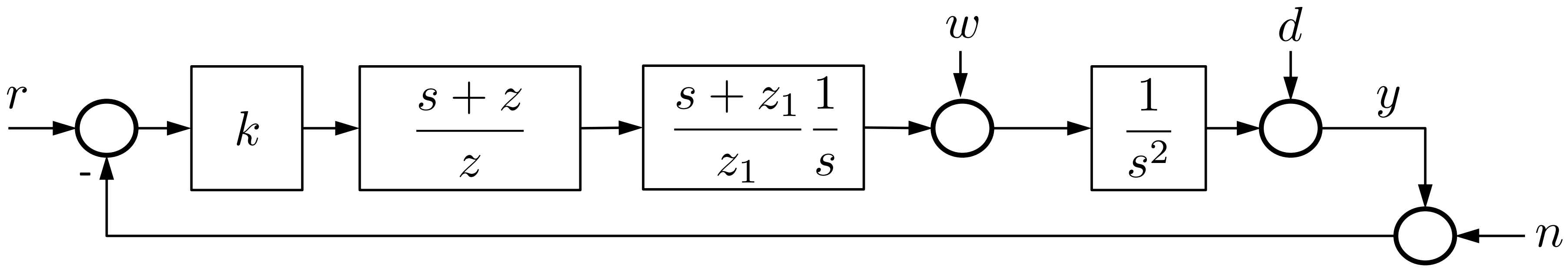
$$C_{PID}(s) = k_P + \frac{k_I}{s} + k_D s$$

$$C_{PD}(s) = k\frac{s+z}{z}$$

Add zero and integrator

$$C_{PID}(s) = k\frac{(s+z)(s+z_1)}{zz_1 s}$$

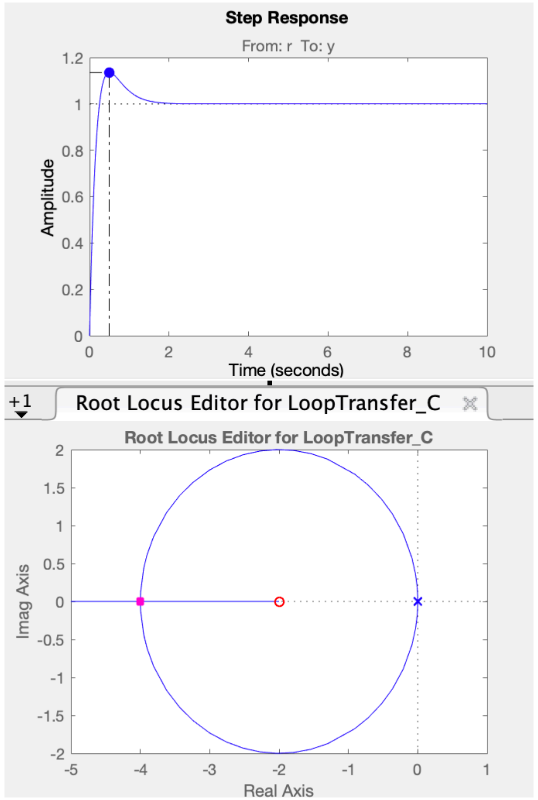
Disturbance Rejection – Integral Action



– keep zero at -z, add integrator and zero at -z1

$k = 16$

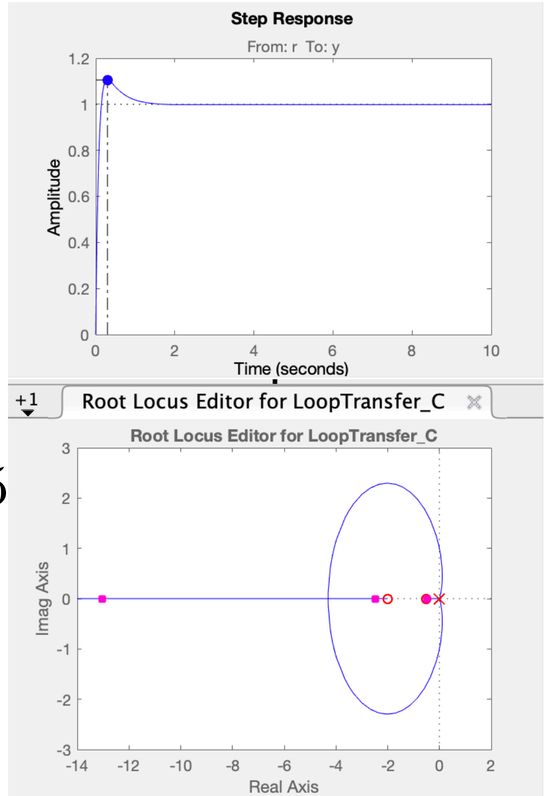
$z = 2$



$k = 16$

$z = 2$

$z_1 = 0.5$



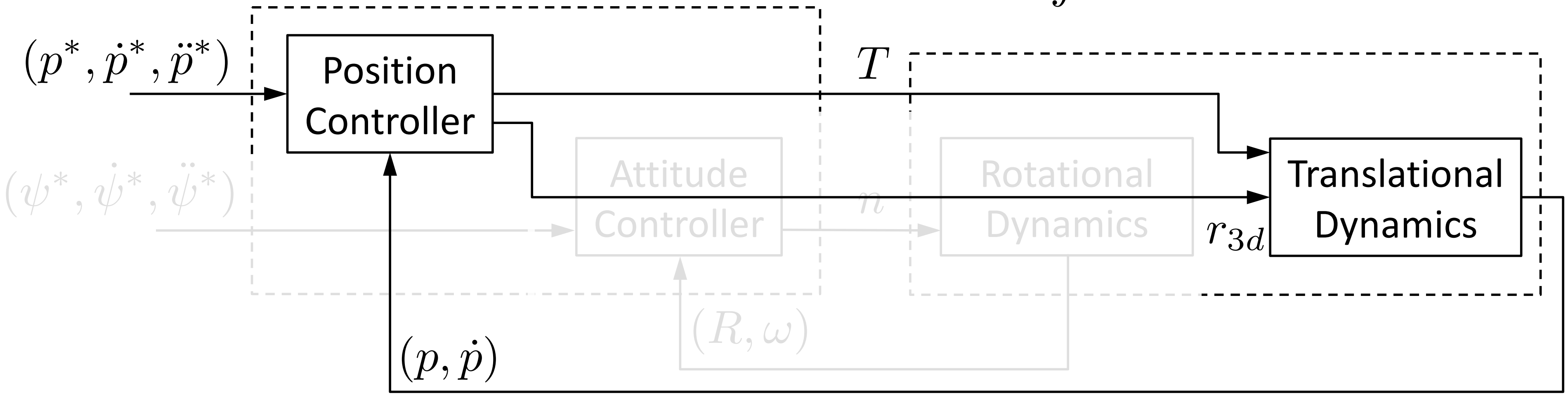
Hierarchical control

1. Neglect rotational dynamics
   - Replace r_3 by a desired value r_{3d}
   - Define a new virtual input for the translational dynamics $u_T = -\frac{T}{m} r_{3d}$
     T gives magnitude, r_{3d} gives direction

$$T = m\|u_T\|, \ r_{3d} = -\frac{u_T}{\|u_T\|}$$

2. Define a control law for u_T, assuming $\ddot{p} = u_T + g$

$$u_T = -k_P(p - p^*) - k_D(\dot{p} - \dot{p}^*) - k_I \int (p - p^*) d\tau + \ddot{p}^* - g$$



Hierarchical control

3. Combine r_{3d} with ψ^* to define a desired attitude R_d

$$R_d = f(r_{3d}, \psi^*) = ?$$

or

$$\lambda_d = (\phi_d, \theta_d, \psi^*)$$
$$(\phi_d, \theta_d) = g(r_{3d}, \psi^*) = ?$$

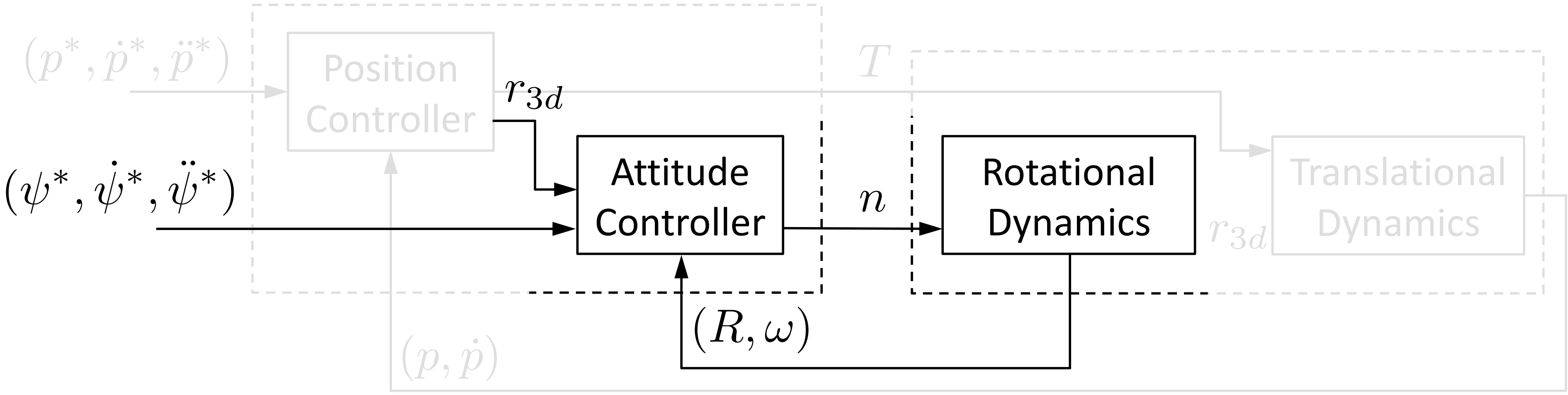
4. Define a control law for *n* to track the desired attitude R_d or λ_d

$$\dot{R} = RS(\omega)$$
$$J\dot{\omega} = -S(\omega)J\omega + n$$

or

$$\dot{\lambda} = Q(\phi, \theta)\omega$$
$$J\dot{\omega} = -S(\omega)J\omega + n$$

$$n = ?$$



Hierarchical control

3. Combine r_{3d} with ψ^* to define a desired attitude R_d

$$R_d = f(r_{3d}, \psi^*) = ?$$

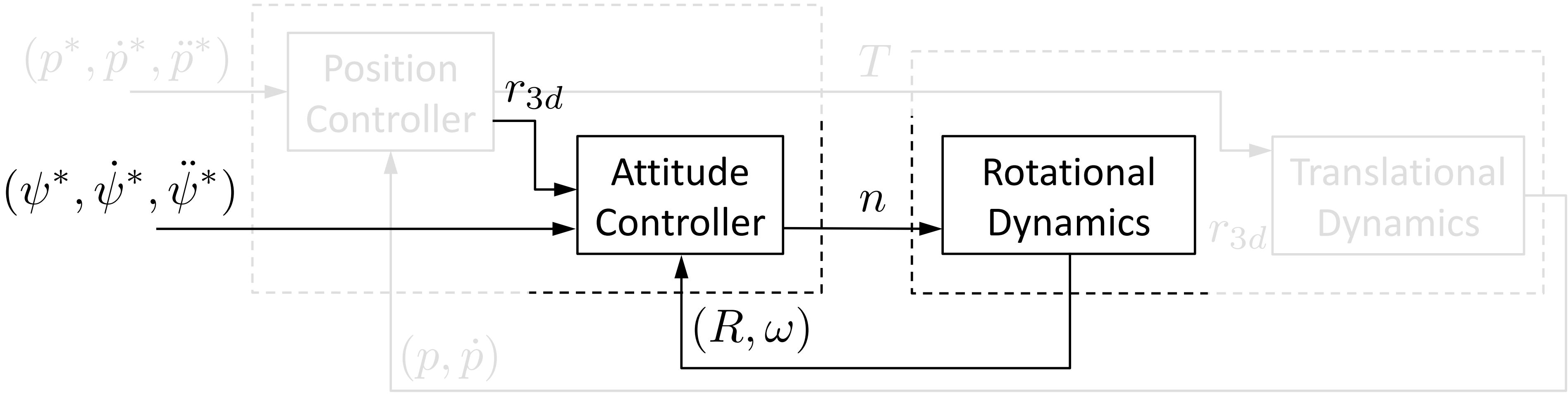
or

$$\lambda_d = (\phi_d, \theta_d, \psi^*)$$
$$(\phi_d, \theta_d) = g(r_{3d}, \psi^*) = ?$$

– References for the attitude controller $\lambda_d = (\phi_d, \theta_d, \psi^*)$

$$r_{3d} = R_z(\psi^*)R_y(\theta_d)R_x(\phi_d)e_3$$

$$\begin{bmatrix} \cos\phi_d \sin\theta_d \\ -\sin\phi_d \\ \cos\phi_d \cos\theta_d \end{bmatrix} = R_z(-\psi^*)r_{3d}$$



Attitude Controller

- Linearize model about hover condition

$$\lambda_0 = [0\ 0\ \psi^*]^T$$
$$\omega_0 = [0\ 0\ 0]^T$$
$$n_0 = [0\ 0\ 0]^T$$

$$\dot{\lambda} = Q(\phi, \theta)\omega$$
$$J\dot{\omega} = -S(\omega)J\omega + n$$

- Given a equilibrium point $f(x_0, u_0) = 0$
  recall that for $\dot{x} = f(x, u)$

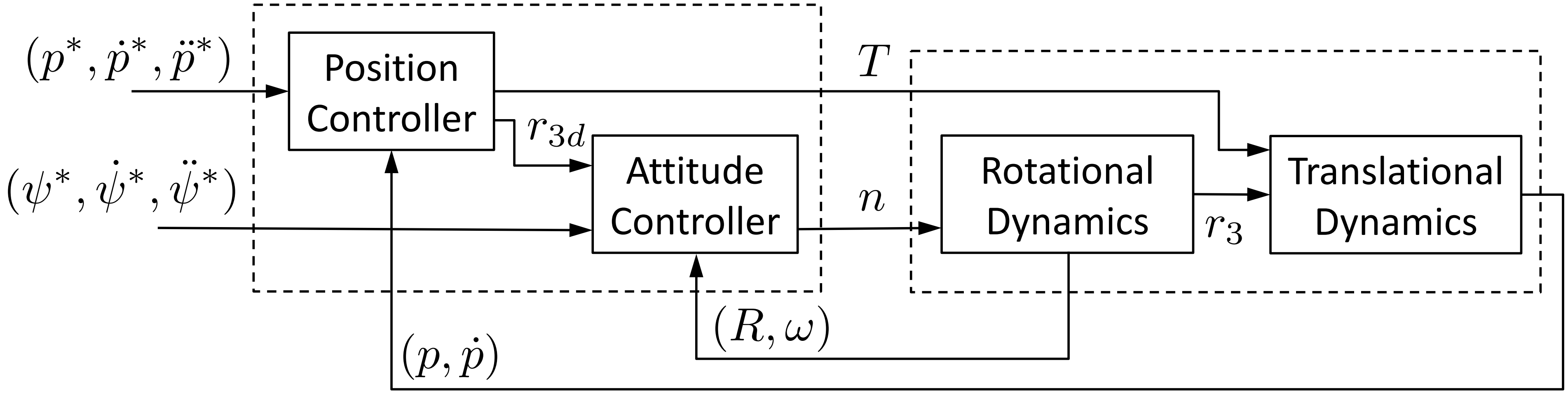
$$\delta\dot{x} = \left.\frac{\partial f}{\partial x}\right|_{eq} \delta x + \left.\frac{\partial f}{\partial u}\right|_{eq} \delta u$$

- Again a double integrator if J is diagonal

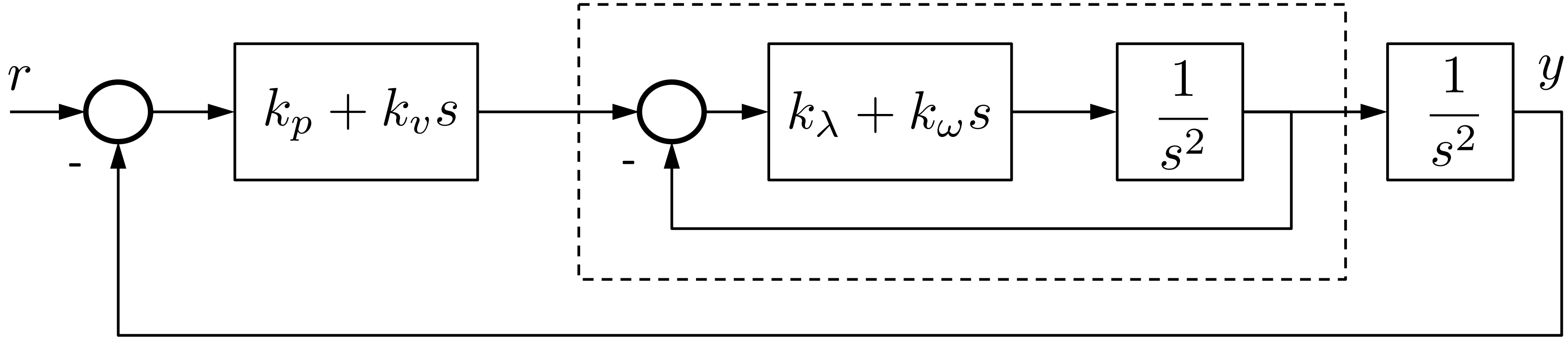
$$\delta\dot{\lambda} = \delta\omega$$
$$J\delta\dot{\omega} = \delta n$$

- PD controller $n = \delta n = -k_1(\lambda - \lambda_d) - k_2\omega$

Hierarchical control



- Simplified SISO model
  - Bandwidth of inner-loop system 5 to 10 times higher than bandwidth of outer-loop system.

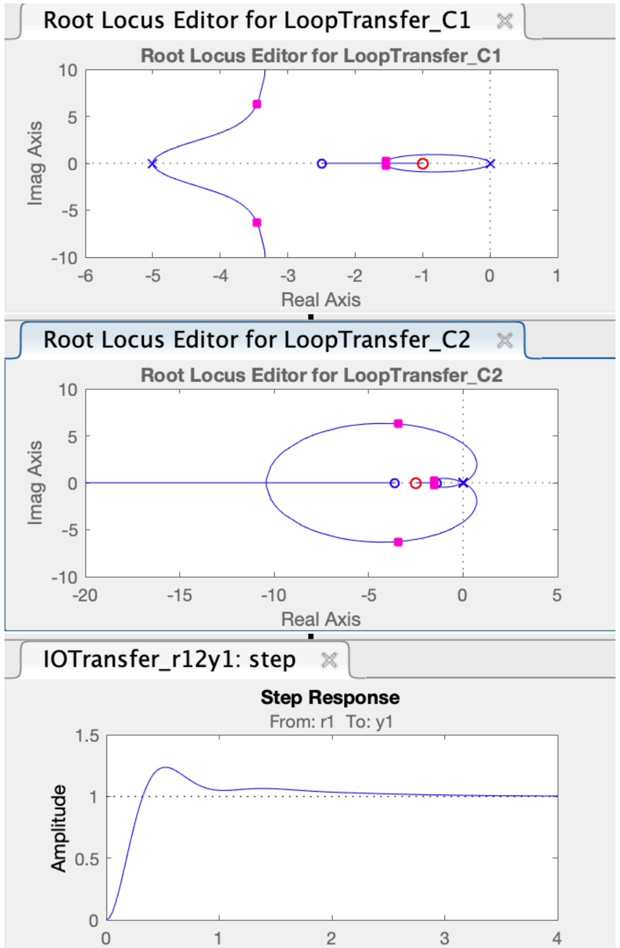


Hierarchical control

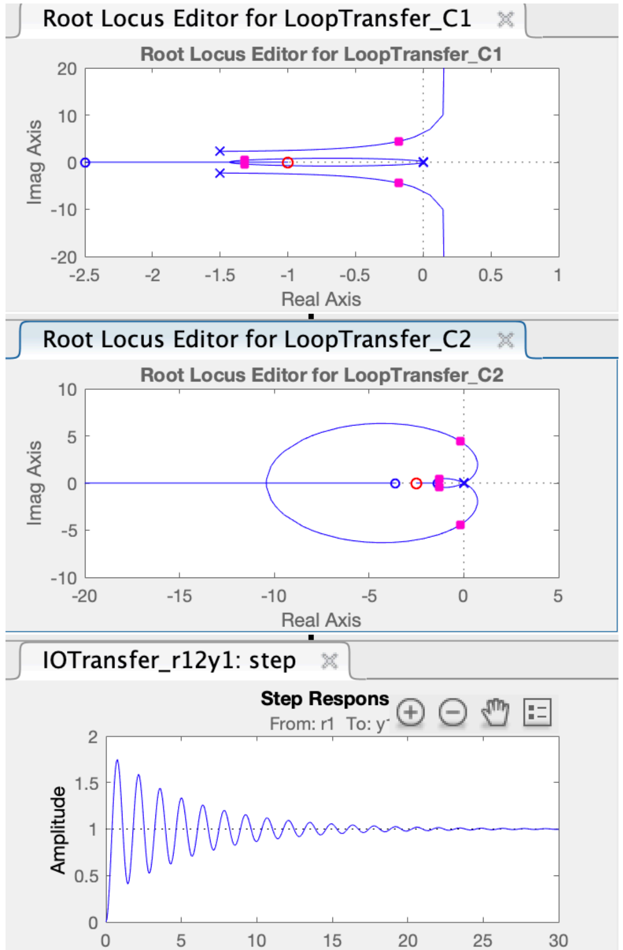
- Bandwidth of inner-loop system 5 to 10 times higher than bandwidth of outer-loop system.

$$C_1(s) = k_p + k_v s$$
$$C_2(s) = k_\lambda + k_\omega s$$

$$C_1(s) = 5(s+1)$$
$$C_2(s) = 10(s+2.5)$$



$$C_1(s) = 5(s+1)$$
$$C_2(s) = 3(s+2.5)$$



State-space representation – example

- Double integrator system

$$\ddot{y} = u$$

  – State vector

$$x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$

  – State space representation

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u$$

  – Solution of the unforced system (*u=0*)

$$x(t) = e^{At} x(0)$$

State-space representation - example

- Double integrator system
  - State space representation

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u \qquad x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$

  - Solution of the unforced system (*u=0*)

$$\begin{aligned} x(t) &= e^{At}x(0) \\ &= \left(e^{\lambda_0 t} v_1 w_1^T + e^{\lambda_0 t} v_2 w_2^T + t e^{\lambda_0 t} v_1 w_2^T\right) x(0) \end{aligned}$$

$$\lambda_0 = 0 \quad v_1 = w_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad v_2 = w_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$x(t) = y(0)v_1 + \dot{y}(0)v_2 + \dot{y}(0)tv_1 = \begin{bmatrix} y(0) + \dot{y}(0)t \\ \dot{y}(0)t \end{bmatrix}$$

Position control – Pole placement

- Double integrator system – PID Control
  - State space representation of error system

$$x_e = \begin{bmatrix} y - r \\ \dot{y} - \dot{r} \end{bmatrix} \quad \dot{x}_e = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (u - \ddot{r})$$

  - PID controller

$$u = -k_P e - k_D e - k_I \int e \, d\tau + \ddot{r}$$

  - Augment the state with x_I

$$\dot{x}_I = e$$

$$x = \begin{bmatrix} x_I \\ x_e \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (u - \ddot{r})$$

$$u = -\underbrace{\begin{bmatrix} k_I & k_P & k_D \end{bmatrix}}_{K} x + \ddot{r}$$

Position Control – Pole placement

- Double integrator system – PID Control
  - Closed-loop error system

$$\dot{x} = (A - BK)x = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_I & -k_P & -k_D \end{bmatrix} x$$

  - Characteristic equation

$$\det(sI - (A - BK)) = s^3 + k_D s^2 + k_P s + k_I = 0$$

  - Desired poles and desired characteristic equation

$$(s - p_1)(s - p_2)(s - p_3) = 0$$

$$s^3 - (p_1 + p_2 + p_3)s^2 + (p_1 p_2 + p_1 p_3 + p_2 p_3)s - p_1 p_2 p_3 = 0$$

  - Direct matching

$$k_D = -(p_1 + p_2 + p_3)$$

$$k_P = (p_1 p_2 + p_1 p_2 + p_2 p_3)$$

$$k_I = -p_1 p_2 p_3$$

Position Control – Pole placement

- Double integrator system – PID Control
  - Closed-loop error system

$$\dot{x} = (A - BK)x = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -k_I & -k_P & -k_D \end{bmatrix} x$$

  - Direct matching for pole placement

$$k_D = -(p_1 + p_2 + p_3)$$
$$k_P = (p_1 p_2 + p_1 p_3 + p_2 p_3)$$
$$k_I = -p_1 p_2 p_3$$

  - Examples

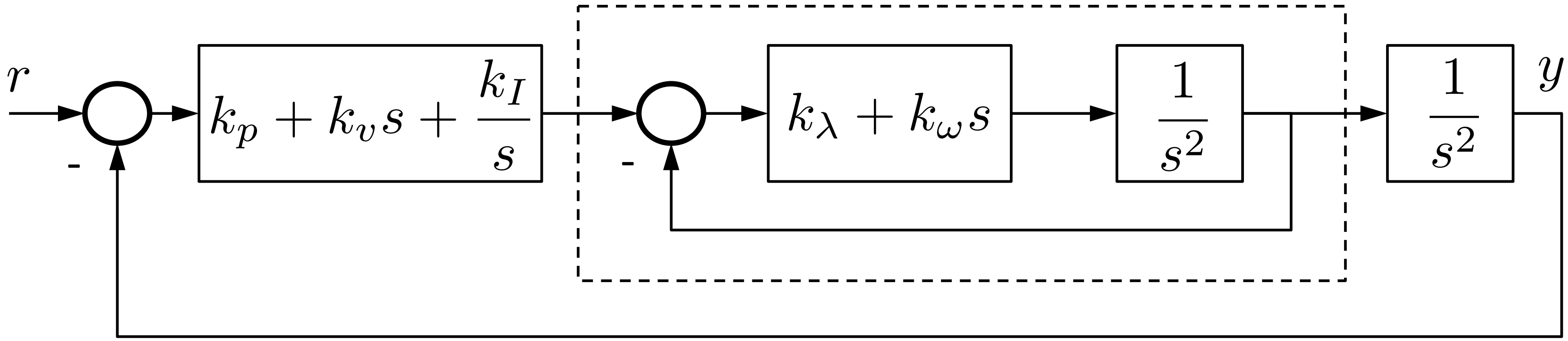
$$p_1 = -2; p_2 = -2; p_3 = 0$$
$$k_P = 4; k_D = 4; k_I = 0$$

$$p_1 = -2; p_2 = -2; p_3 = -0.5$$
$$k_P = 6; k_D = 4.5; k_I = 2$$

    - Recovers PD controller

Hierarchical Control – Pole placement

- Double integrator system + integrator + inner-loop dynamics



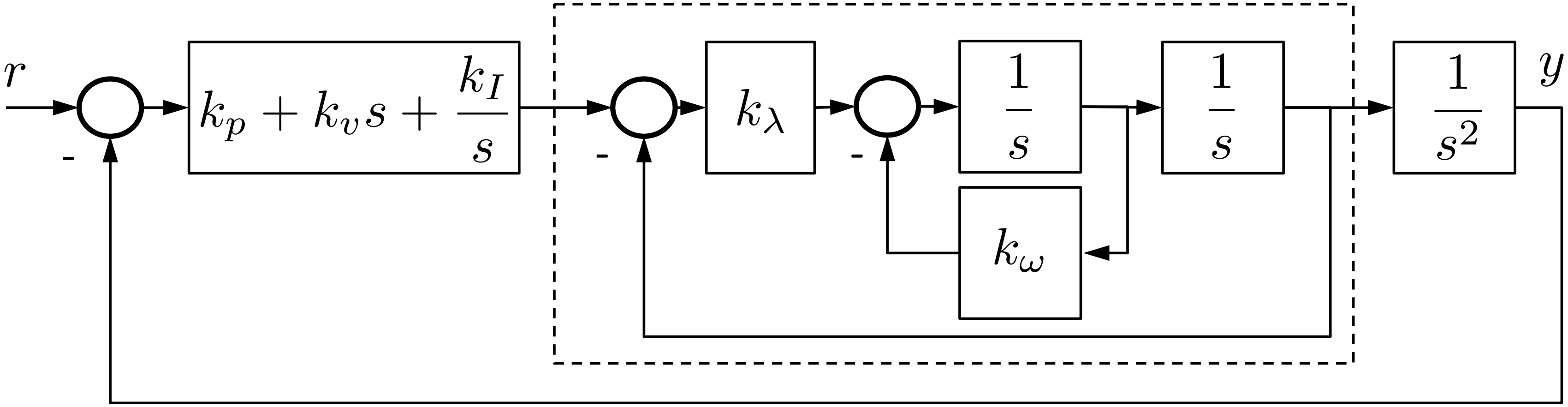
- Assume inner-loop system is fixed (k_λ, k_ω)
- State-space representation

$$\dot{x}_{in} = \begin{bmatrix} 0 & 1 \\ -k_\lambda & -k_\omega \end{bmatrix} x_{in} + \begin{bmatrix} 0 \\ k_\lambda \end{bmatrix} \theta_r \qquad \dot{x} = Ax - B\theta, \quad \theta = \begin{bmatrix} 1 & 0 \end{bmatrix} x_{in}$$

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{in} \end{bmatrix} = \begin{bmatrix} A & B\begin{bmatrix} 1 & 0 \end{bmatrix} \\ 0 & \begin{bmatrix} 0 & 1 \\ -k_\lambda & -k_\omega \end{bmatrix} \end{bmatrix} x_{in} + \begin{bmatrix} 0 \\ \begin{bmatrix} 0 \\ k_\lambda \end{bmatrix} \end{bmatrix} \theta_r$$

Hierarchical Control – Pole placement

- Double integrator system + integrator + inner-loop dynamics



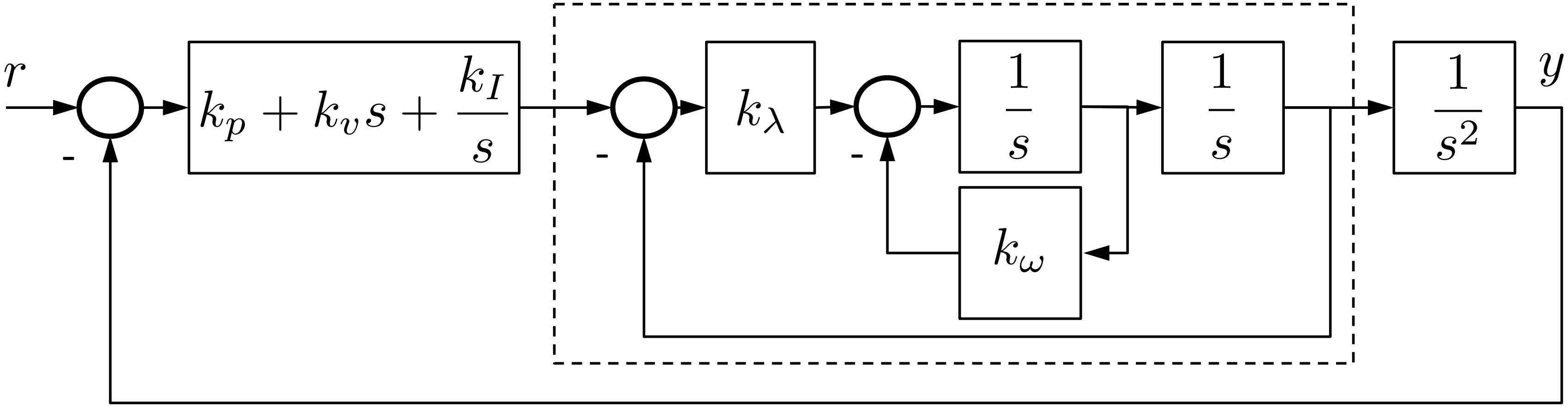
- State-space representation

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{in} \end{bmatrix} = \begin{bmatrix} A & B\begin{bmatrix} 1 & 0 \end{bmatrix} \\ 0 & \begin{bmatrix} 0 & 1 \\ -k_\lambda & -k_\omega \end{bmatrix} \end{bmatrix} x_{in} + \begin{bmatrix} 0 \\ \begin{bmatrix} 0 \\ k_\lambda \end{bmatrix} \end{bmatrix} \theta_r$$

- Keep inner-loop poles (roots(1,kw,kl))
- Place additional poles at -0.5+0.2i, -0.5-0.2i, -0.1

Pole placement example

- Double integrator system + integrator + inner-loop dynamics



- K=acker(A2,B2, [-0.5+0.2*i;-0.5-0.2*i;-0.1;roots(1,kw,kl)])

$$u = \theta_r = \begin{bmatrix} 0.0290 & 0.4190 & 1.4973 & 1.1975 & 0.2750 \end{bmatrix} \begin{bmatrix} x_I \\ y - r \\ \dot{y} - \dot{r} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

- No longer hierarchical!