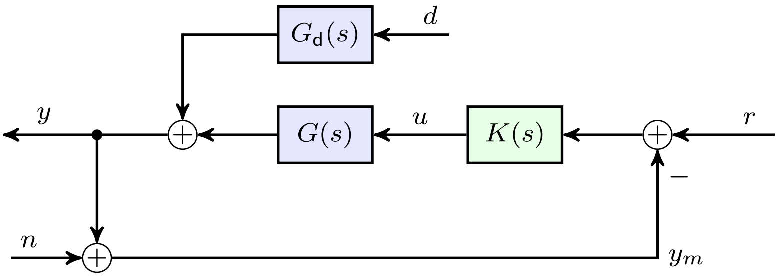
Control Systems 2

Lecture 2: Loopshaping design

Roy Smith

Loopshaping



$$
\begin{aligned}
e &= r - y \\
&= \underbrace{\frac{1}{1+L(s)}}_{S(s)} r - \underbrace{\frac{1}{1+L(s)}}_{S(s)} G_d(s) d + \underbrace{\frac{L(s)}{1+L(s)}}_{T(s)} n
\end{aligned}
$$

Loopshaping

$$e = \underbrace{\frac{1}{1+L(s)}}_{S(s)} r - \underbrace{\frac{1}{1+L(s)}}_{S(s)} G_{\mathsf{d}}(s)d + \underbrace{\frac{L(s)}{1+L(s)}}_{T(s)} n$$

Performance requirements can be approximated by requirements for $L(j\omega)$.

$|L(j\omega)| \gg 1 \quad \Longrightarrow \quad |S(j\omega)| \ll 1$ (good tracking performance)

Loopshaping

$$e = \underbrace{\frac{1}{1+L(s)}}_{S(s)} r - \underbrace{\frac{1}{1+L(s)}}_{S(s)} G_{\mathsf{d}}(s)d + \underbrace{\frac{L(s)}{1+L(s)}}_{T(s)} n$$

Performance requirements can be approximated by requirements for $L(j\omega)$.

$|L(j\omega)| \gg 1 \quad \Longrightarrow \quad |S(j\omega)| \ll 1$ (good tracking performance)

$|L(j\omega_c)| = 1$ gives bandwidth $\approx \omega_c$

Loopshaping

$$e = \underbrace{\frac{1}{1+L(s)}}_{S(s)} r - \underbrace{\frac{1}{1+L(s)}}_{S(s)} G_{\mathsf{d}}(s) d + \underbrace{\frac{L(s)}{1+L(s)}}_{T(s)} n$$

Performance requirements can be approximated by requirements for $L(j\omega)$.

$|L(j\omega)| \gg 1 \implies |S(j\omega)| \ll 1$ (good tracking performance)

$|L(j\omega_c)| = 1$ gives bandwidth $\approx \omega_c$

$|L(j\omega)| \ll 1 \implies |T(j\omega)| \ll 1$ (good noise rejection)

Bode gain-phase relationship

For minimum phase stable systems (with $L(0) > 0$),

Bode gain-phase relationship

For minimum phase stable systems (with $L(0) > 0$),

The phase of $L(j\omega)$ is determined by the *slope* of $|L(j\omega)|$.

If the slope is constant then:

$$\frac{d|L(j\omega)|}{d\omega} = -20 \text{ dB/decade} \quad \Longleftrightarrow \quad \angle L(j\omega) = -90^\circ$$
$$= -20n \text{ dB/decade} \quad \Longleftrightarrow \quad \angle L(j\omega) = -90n^\circ, \quad n = 1, 2, \ldots$$

Bode gain-phase relationship

For minimum phase stable systems (with $L(0) > 0$),

The phase of $L(j\omega)$ is determined by the *slope* of $|L(j\omega)|$.

If the slope is constant then:

$$\frac{d|L(j\omega)|}{d\omega} = -20 \text{ dB/decade} \quad \Longleftrightarrow \quad \angle L(j\omega) = -90^\circ$$
$$= -20n \text{ dB/decade} \quad \Longleftrightarrow \quad \angle L(j\omega) = -90n^\circ, \quad n = 1, 2, \ldots$$

Slope constraint at crossover:

$$\frac{d|L(j\omega)|}{d\omega} = -20 \text{ dB/decade} \quad \Longrightarrow \quad PM \approx 90^\circ$$

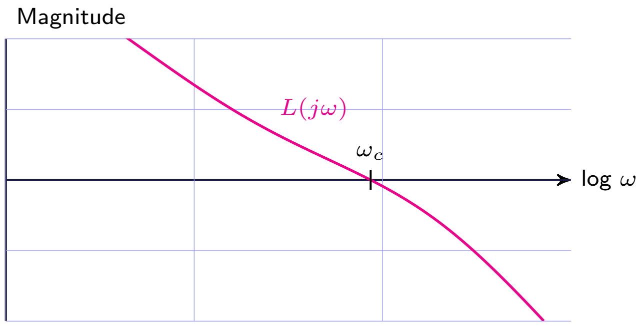
(this requires a constant slope for a wide region of frequency)

Loopshape specifications

1. $|L(j\omega)| \gg 1$ for frequencies requiring high performance.
2. Gain crossover frequency, ω_c, gives closed-loop bandwidth.
3. The slope of $L(j\omega)$ at crossover should be -20dB/decade.
4. The system type is the number of pure integrators in $L(s)$.

Loopshape specifications

1. $|L(j\omega)| \gg 1$ for frequencies requiring high performance.
2. Gain crossover frequency, ω_c, gives closed-loop bandwidth.
3. The slope of $L(j\omega)$ at crossover should be -20dB/decade.
4. The system type is the number of pure integrators in $L(s)$.



Inverse-based controller design

Stable, minimum-phase plant

Can choose:

$$L(s) = \frac{\omega_c}{s}$$

This will give a phase margin of 90°.

$$K(s) = \frac{\omega_c}{s} G^{-1}(s)$$

Inverse-based controller design

Stable, minimum-phase plant

Can choose:

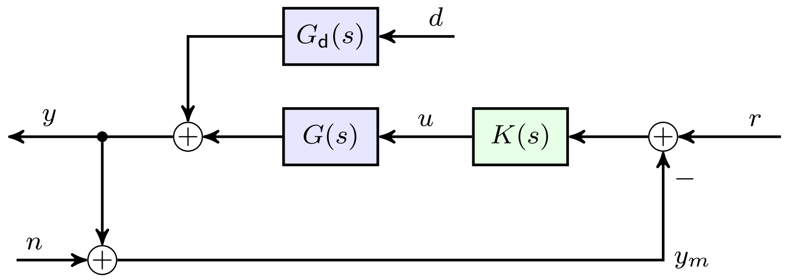
$$L(s) = \frac{\omega_c}{s}$$

This will give a phase margin of 90°.

$$K(s) = \frac{\omega_c}{s} G^{-1}(s)$$

Inverting the plant can often be done only approximately.

Example: disturbance process



$$G(s) = \frac{200}{(10s+1)} \frac{1}{(0.05s+1)^2}, \qquad G_d(s) = \frac{100}{10s+1}$$

Example: disturbance process

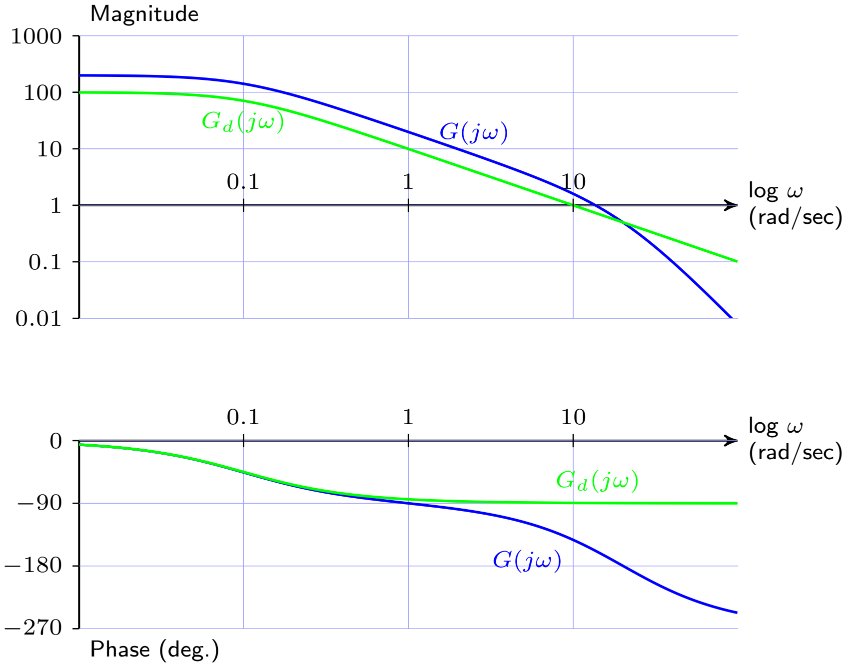
$$G(s) = \frac{200}{(10s+1)} \frac{1}{(0.05s+1)^2}, \qquad G_d(s) = \frac{100}{10s+1}$$

Objectives:

1. Rise time < 0.3 seconds.
2. Overshoot $< 5\%$
3. Disturbance response, $y_d(t)$, satisfies $|y(t)| \leq 1$.
4. Disturbance response, $y_d(t)$, satisfies $|y(t)| < 0.1$ within 3 seconds.
5. $|u(t)| \leq 1$ at all times.

$|G_d(j\omega)| > 1$ up to $\omega_d \approx 10$ rad/sec $\quad\Longrightarrow\quad \omega_c \geq 10$ rad/sec.

Example: disturbance rejection problem

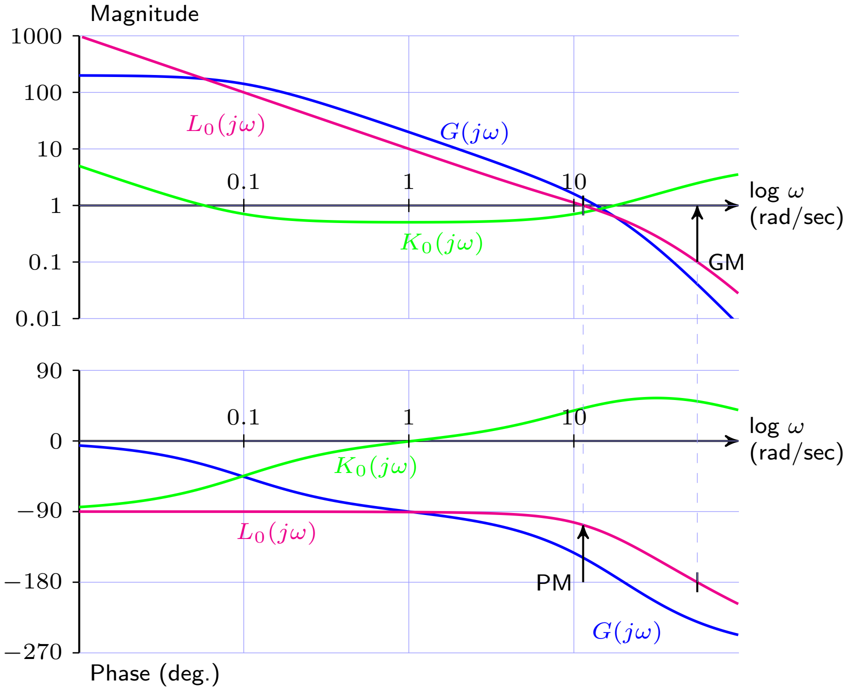


Inverse-based control design

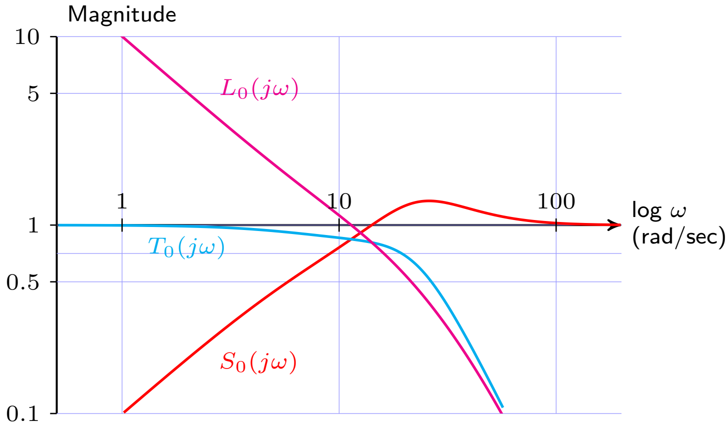
The plant is stable and minimum-phase:

$$
\begin{aligned}
K_0(s) &= \frac{\omega_c}{s} G^{-1}(s) \\
&= \frac{\omega_c}{s} \frac{(10s+1)}{200} (0.05s+1)^2 \\
&\approx \frac{\omega_c}{s} \frac{(10s+1)}{200} \frac{(0.1s+1)}{(0.01s+1)}
\end{aligned}
$$

Example: inverse-based controller design



Example: inverse-based controller design

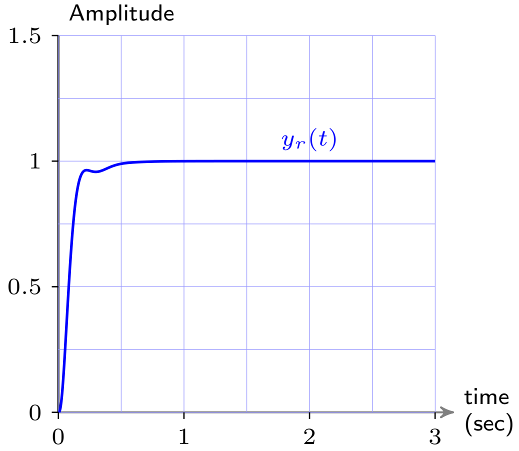


The closed-loop bandwidth is very close to $\omega_c \approx 10$ rad/sec.

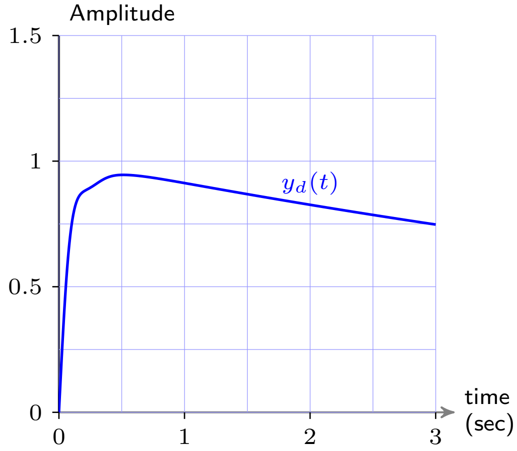
Example: inverse-based controller design

$$K_0 = \frac{\omega_c}{s} \frac{(10s+1)}{200} \frac{(0.1s+1)}{(0.01s+1)}$$

Reference tracking



Disturbance response



Loopshaping for disturbance rejection

Disturbance response: $y_d = SG_\mathsf{d}\, d + \ldots$

To achieve $|y_d(t)| \leq 1$ for $|d(t)| \leq 1$,

we want $|SG_\mathsf{d}(j\omega)| < 1$ for all ω.

So, we want:

$|1 + L(j\omega)| > |G_\mathsf{d}(j\omega)|$ for all ω.

or, approximately, $|L(j\omega)| > |G_\mathsf{d}(\omega)|$ for all ω.

Initial guess:

$|L_\mathsf{min}| \approx |G_\mathsf{d}|$ or $|K_\mathsf{min}| \approx |G^{-1}G_\mathsf{d}|$

Loopshaping for disturbance rejection

Step 1:

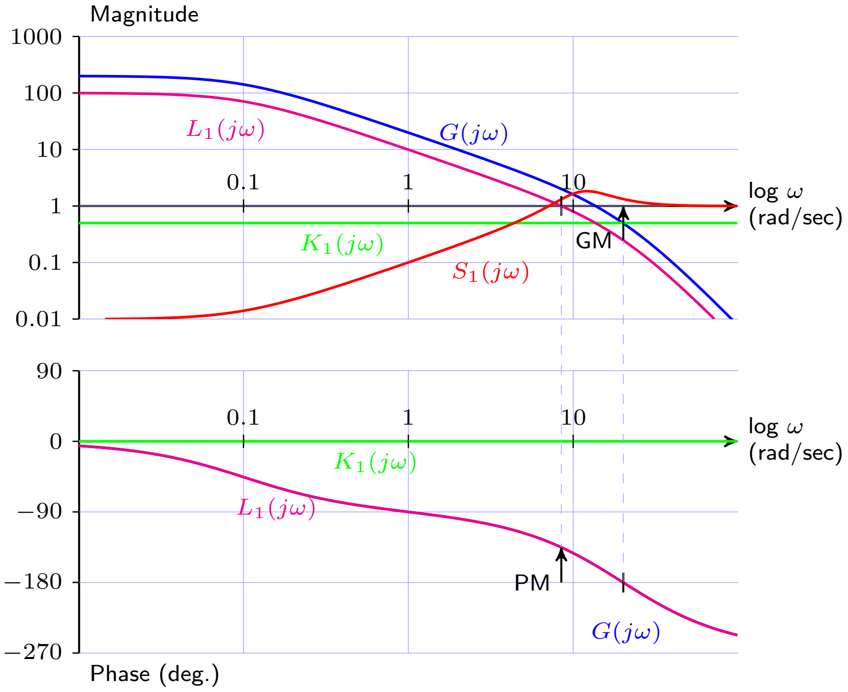
Initial guess:

$$|K_{\mathsf{min}}| \approx |G^{-1}G_{\mathsf{d}}|$$

Choose:

$$\begin{aligned} K_1(s) &\approx G^{-1}(s)G_{\mathsf{d}}(s) \\ &= 0.5(0.05s+1)^2 \\ &\approx 0.5 \end{aligned}$$

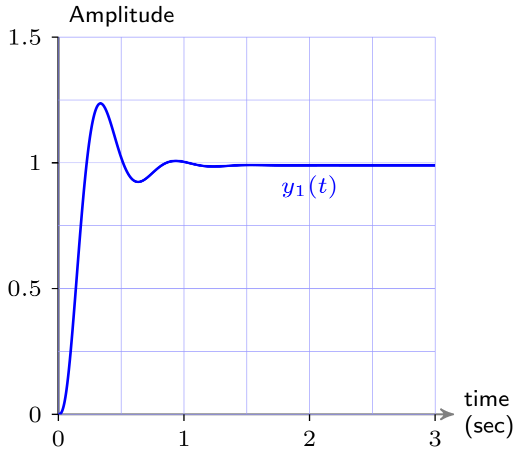
Example: disturbance rejection design



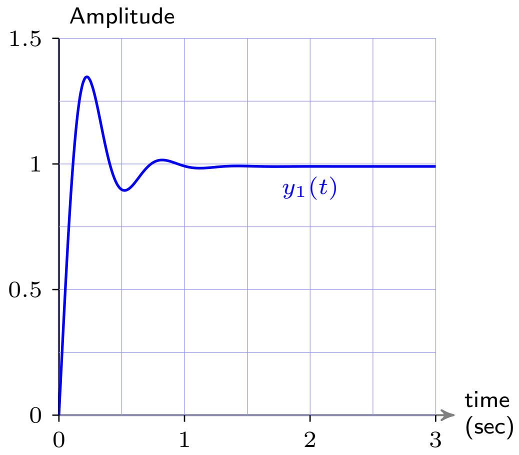
Example: disturbance rejection design

$$K_1 = 0.5 \qquad \Longrightarrow \qquad L_1(s) \approx G_{\mathsf{d}}(s)$$

Reference tracking



Disturbance response



Loopshaping for disturbance rejection

Step 2:

Increase the gain at low frequency.

To get integral action multiply the controller by:

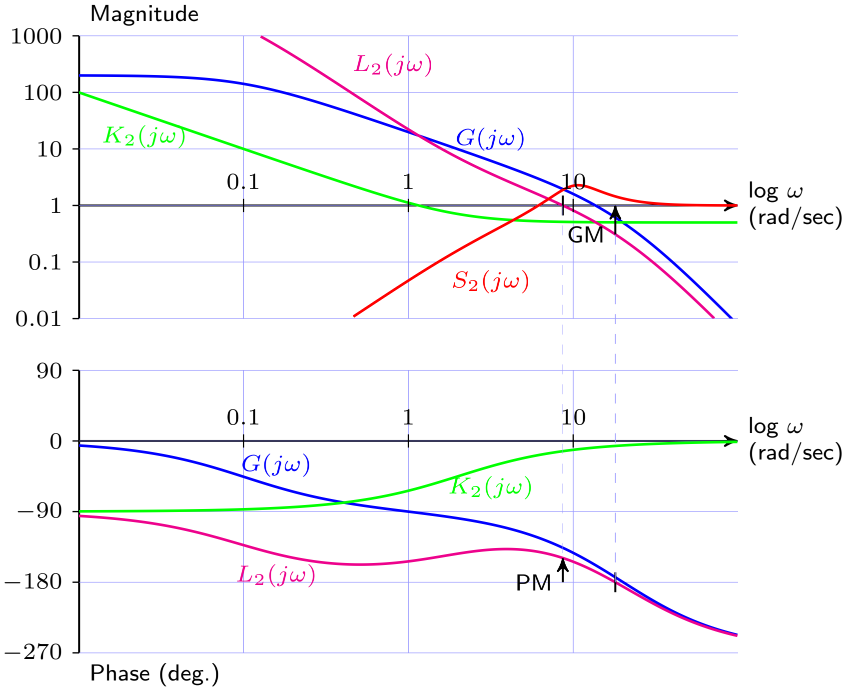
$$K_2(s) = \frac{s + \omega_I}{s} K_1(s).$$

If $\omega_I = 0.2\omega_c$ we get 11° more phase at ω_c than with K_1 alone.

So,

$$K_2(s) = 0.5\frac{(s+2)}{s}$$

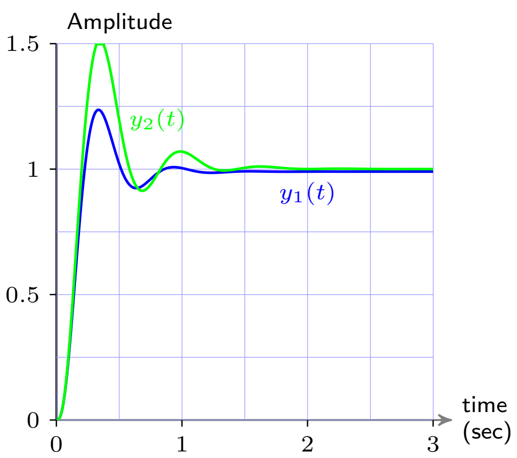
Example: disturbance rejection design



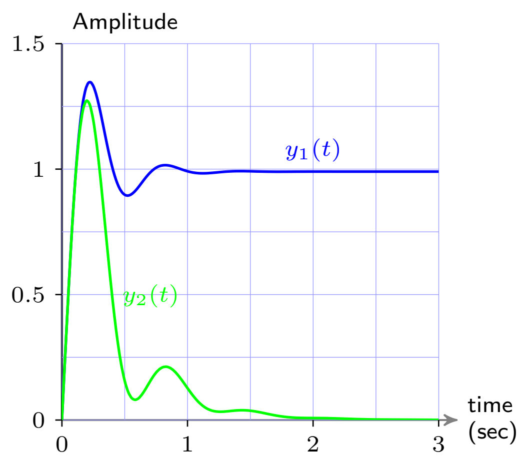
Example: disturbance rejection design

$$K_1 = 0.5 \qquad \text{and} \qquad K_2 = 0.5\frac{(s+2)}{s}$$

Reference tracking



Disturbance response



Loopshaping for disturbance rejection

Step 3:

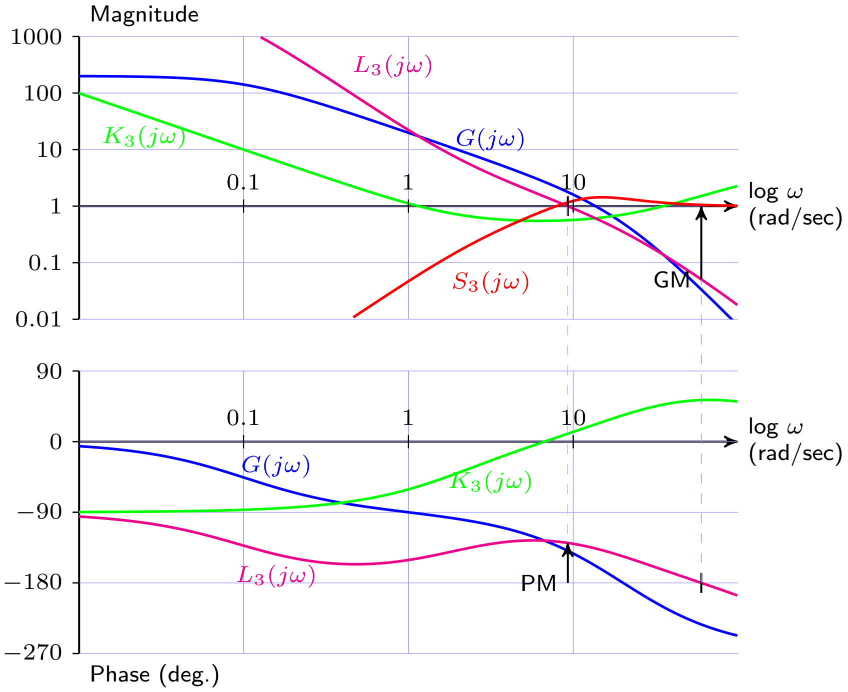
High frequency correction:

Augment with a lead-lag for "derivative action".

This will also improve the phase margin.

$$K_3(s) = K_2(s)\frac{(0.05s+1)}{(0.005s+1)}$$
$$= 0.5\,\frac{(s+2)}{s}\,\frac{(0.05s+1)}{(0.005s+1)}$$

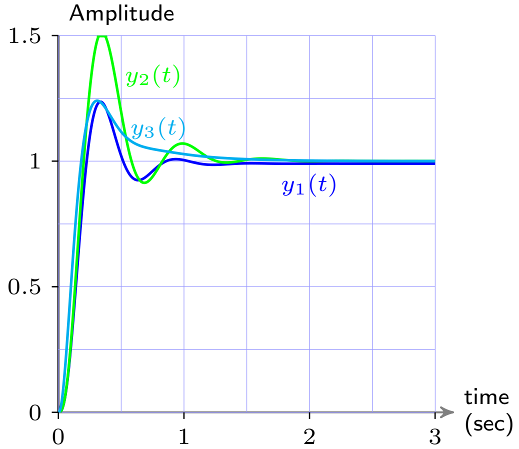
Example: disturbance rejection design



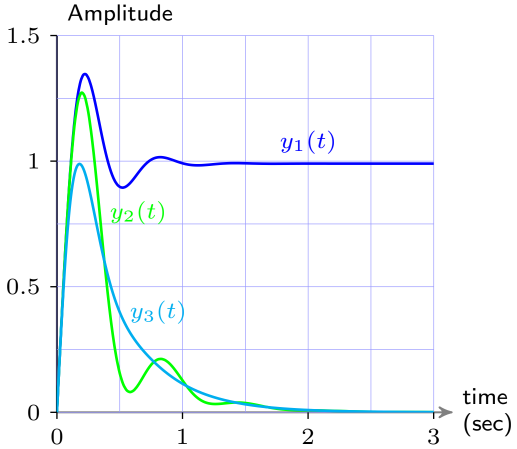
Example: disturbance rejection design

$$K_1 = 0.5, \qquad K_2 = 0.5\frac{(s+2)}{s}, \qquad K_3 = 0.5\frac{(s+2)}{s}\frac{(0.05s+1)}{(0.005s+1)}$$

Reference tracking

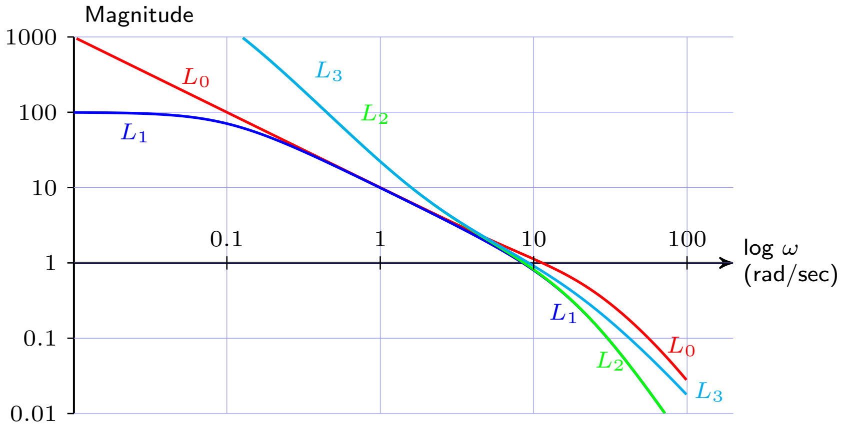


Disturbance response



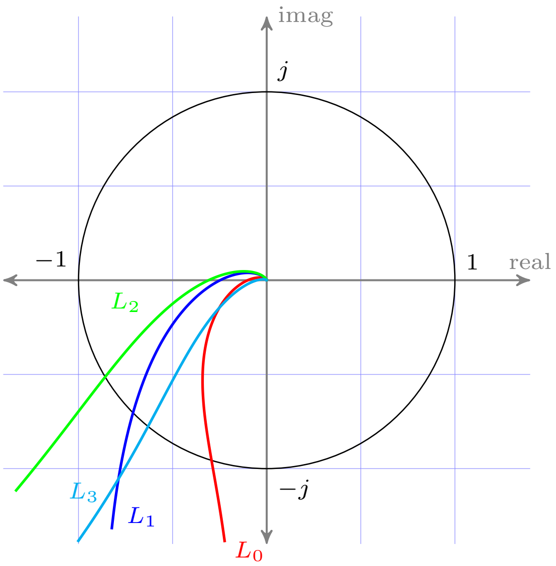
Example: disturbance rejection design

Loopshape comparisons

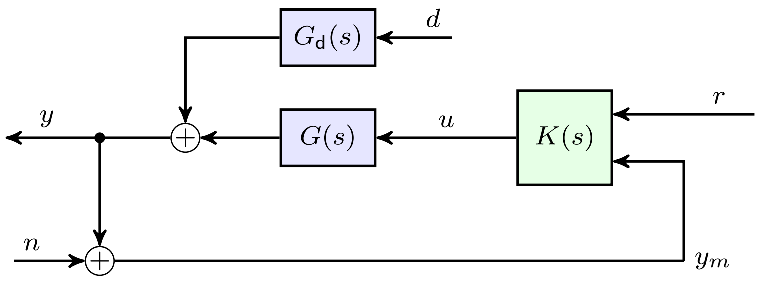


Example: disturbance rejection design

Loopshape comparisons



2 degrees-of-freedom designs



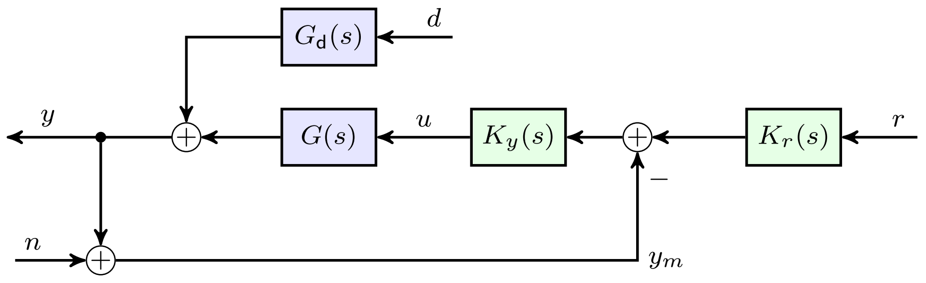
$$u = K(s) \begin{bmatrix} r \\ y_m \end{bmatrix}$$

Effectively choose different transfer functions for

$\frac{y}{d}$ and $\frac{y}{r}$.

2 degrees-of-freedom designs

Reference prefiltering:



Control structure: $u = K_y\,(K_r\,r - y_m)$

So $y = \dfrac{GK_y}{1+GK_y} K_r\,r \;+\; \dfrac{1}{1+GK_y} G_d\,d \quad = \quad TK_r\,r \;+\; SG_d\,d$

with $L = GK_y$.

2 degrees-of-freedom designs

Reference prefiltering:

The reference prefiltered step response is $y = TK_r\,r \;+\; SG_d\,d$

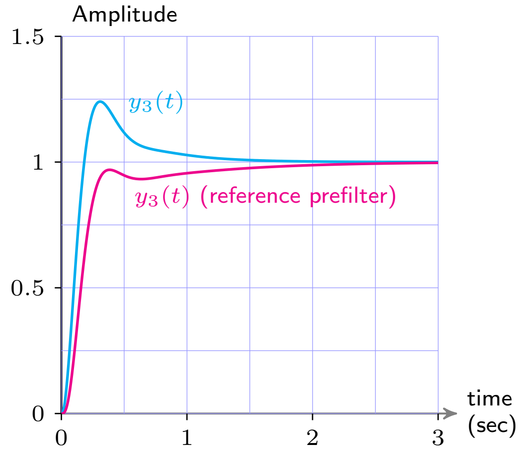
Choose $K_r = T^{-1}T_{ideal}$ (approximately)

$$T(s) \approx \frac{1.5}{0.1s+1} - \frac{0.5}{0.5s+1} = \frac{(0.7s+1)}{(0.1s+1)(0.5s+1)}$$

This implies that, $K_r(s) = \dfrac{0.5s+1}{0.7s+1}$, is a reasonable choice.

2 degrees-of-freedom designs

Reference tracking



$$K_{3r} = \frac{(0.5s+1)}{(0.7s+1)(0.03s+1)}$$

The additional pole was added to prevent $u(t)$ peaking above one.

Notes and references

Skogestad & Postlethwaite (2nd Ed.)

Loopshaping: Section 2.6