Lecture 6 - SISO Loop Analysis

SISO = Single Input Single Output

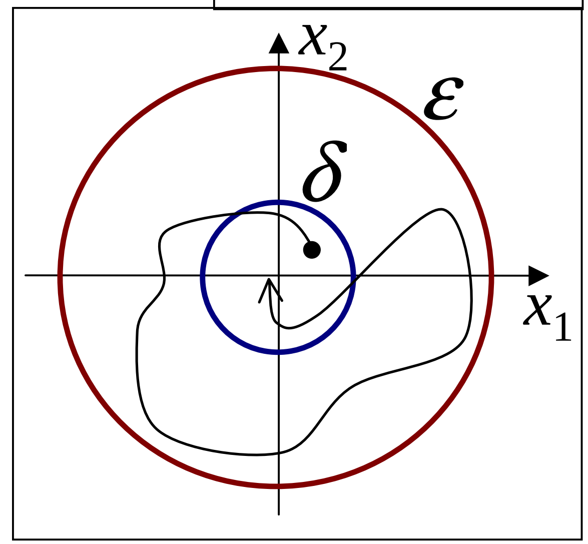
Analysis:

- Stability
- Performance
- Robustness

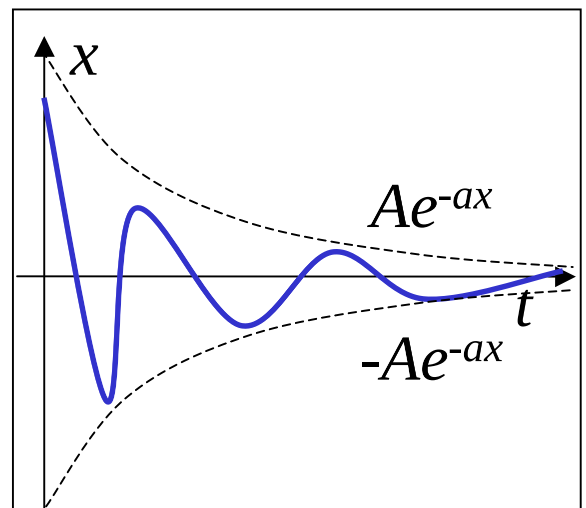
ODE stability

$$\dot{x} = f(x,t)$$

- Lyapunov's stability theory - nonlinear systems
  - stability definition
  - first (direct) method
    - exponential convergence
  - second method: Lyapunov function
    - generalization of energy dissipation



- Lyapunov's exponent
  - dominant exponent of the convergence
  - for a nonlinear system
  - for a linear system defined by the poles

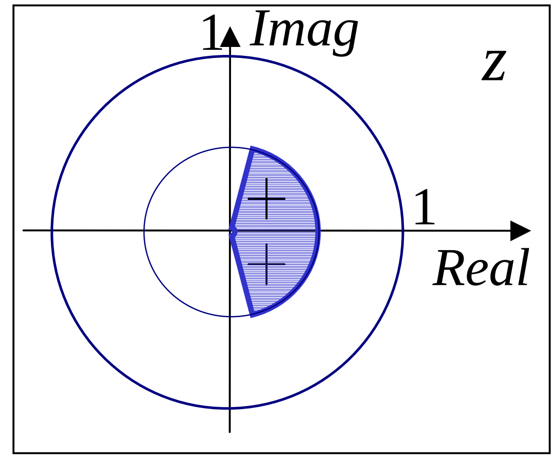


Stability: poles

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

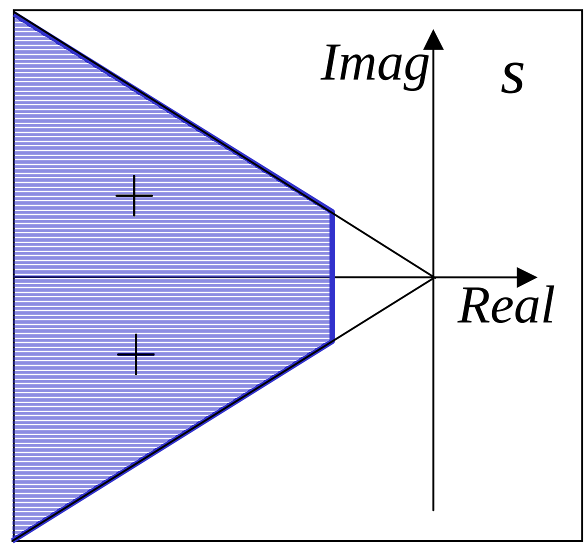
$$y = H(s) \cdot u$$

$$H(s) = C(Is - A)^{-1} B + D$$

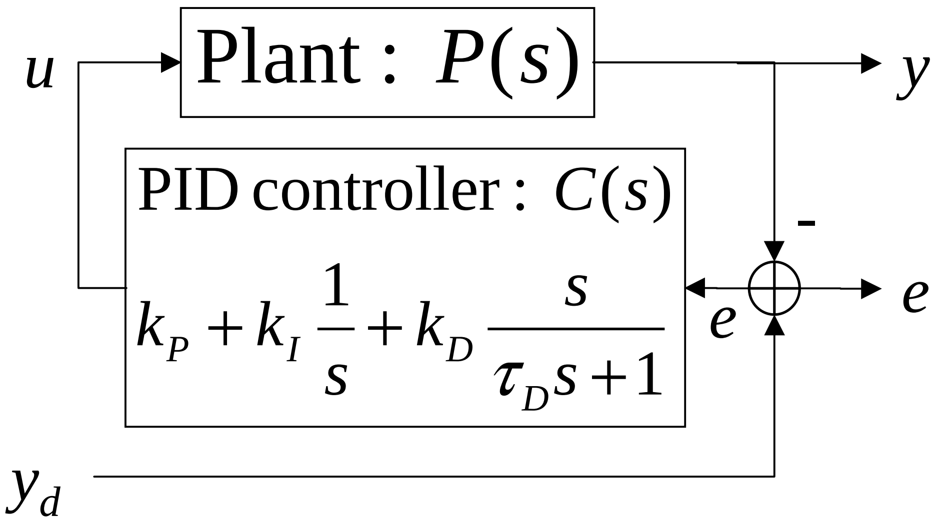


- Characteristic values = transfer function poles
  - l.h.p. for continuous time
  - unit circle for sampled time
- I/O model vs. internal dynamics

$$H(s) = \frac{N(s)}{D(s)} = \frac{g_1}{s - p_1} + \ldots + \frac{g_n}{s - p_n} + g_0$$



Stability: closed loop



$$y = P(s)u$$

$$u = -C(s)e; \qquad e = y - y_d$$

$$e = \underbrace{\left[1 + P(s)C(s)\right]^{-1}}_{S(s)} y_d$$

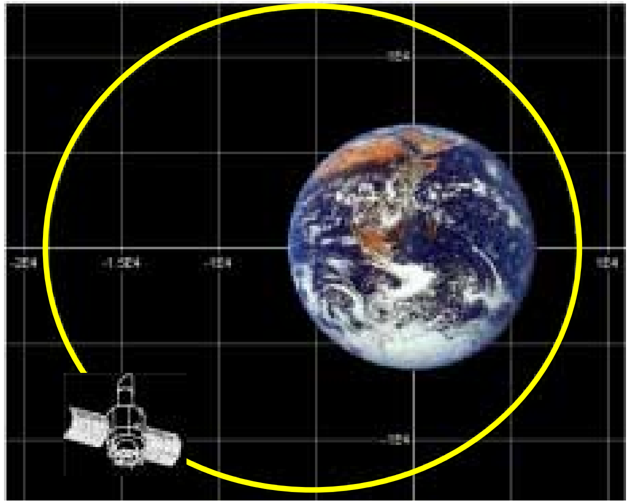
- The transfer function poles are the zeros of

$$1 + P(s)C(s)$$

- Watch for pole-zero cancellations!
- Poles define the closed-loop dynamics (including stability)
- Algebraic problem, easier than state space sim

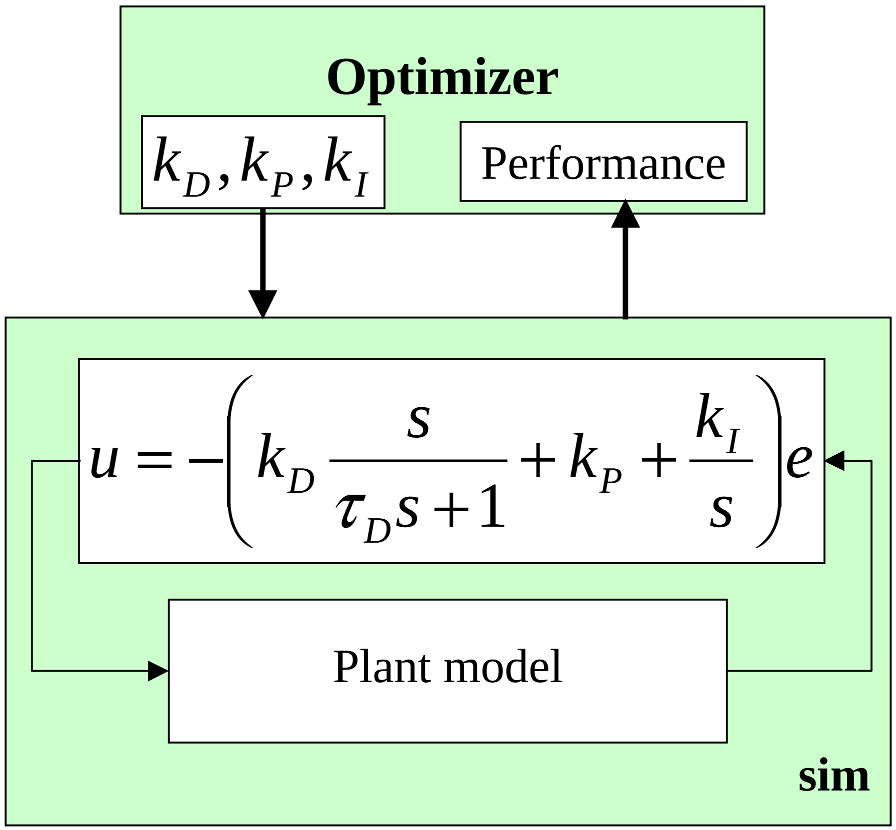
Stability

- **For linear system poles describe stability**
- … almost, except the critical stability
- For nonlinear systems
  - linearize around the equilibrium
  - might have to look at the stability theory - Lyapunov
- Orbital stability:
  - trajectory converges to the desired
  - the state does not - the timing is off
    - spacecraft
    - FMS, aircraft arrival



Performance

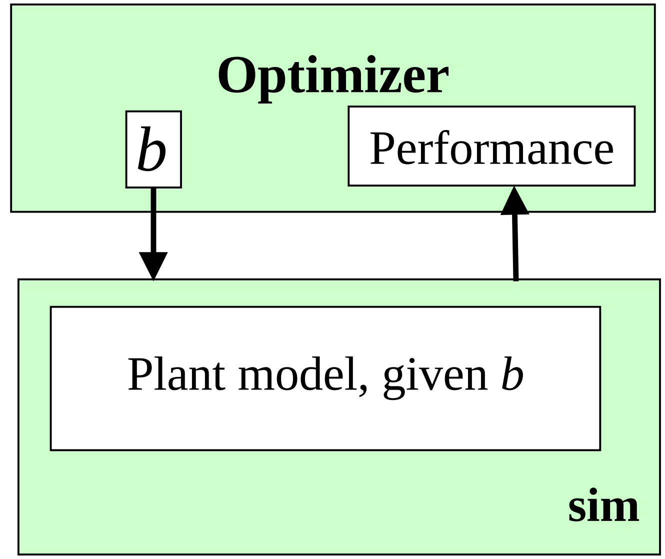
- Need to describe and analyze performance so that we can design systems and tune controllers

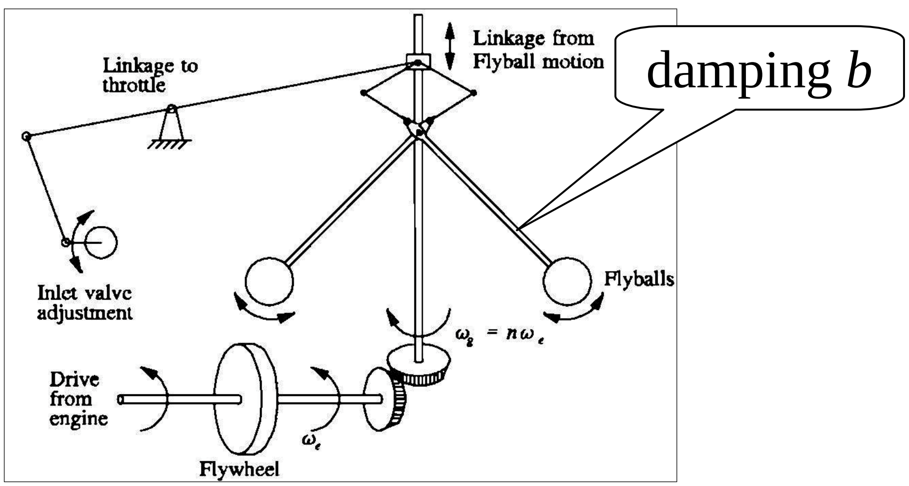


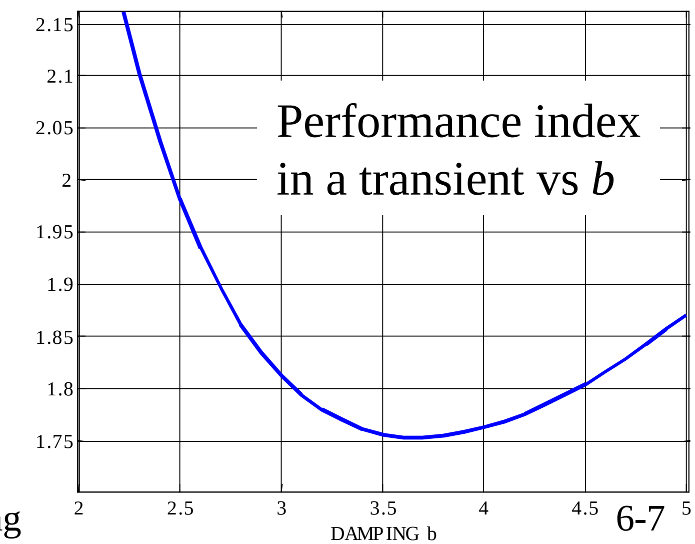
- There are usually many conflicting requirements
- Engineers look for a reasonable trade-off

Performance: Example

- Selecting optimal *b* in the Watt's governor - HW Assignment 1



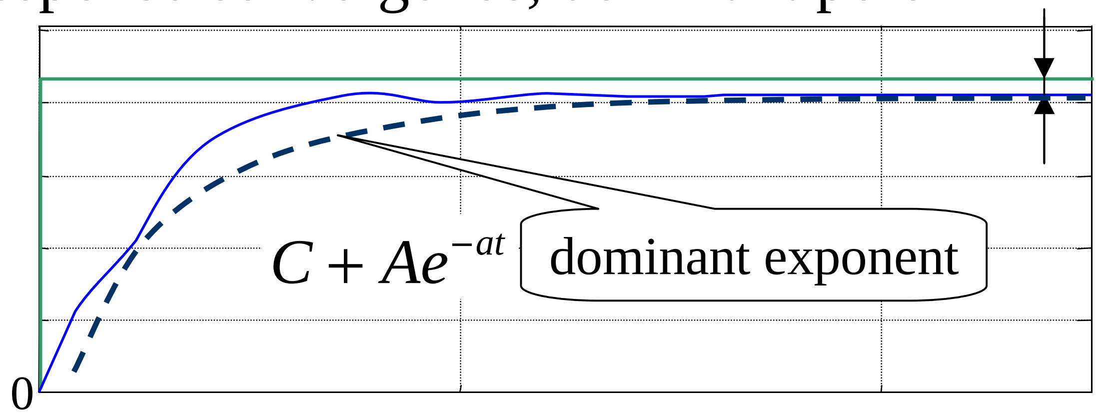




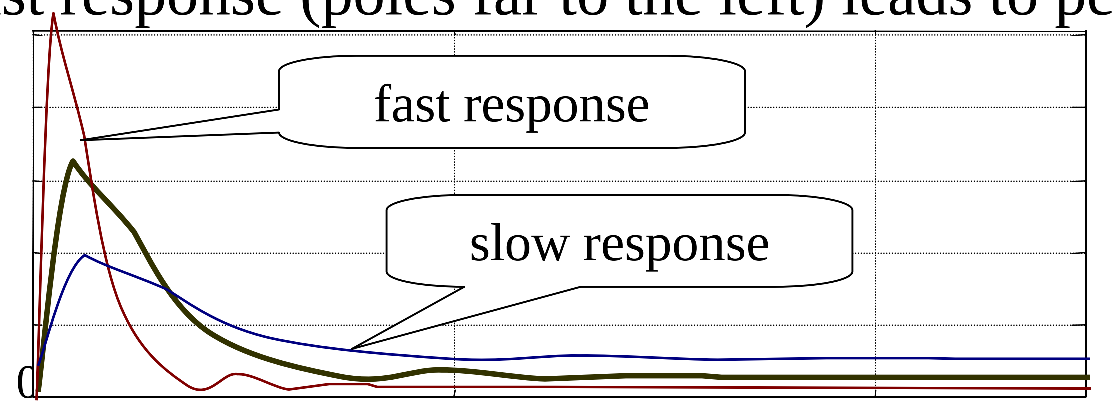
Performance - poles

- Steady state error: study transfer functions at s=0.
- Step/pulse response convergence, dominant pole

$$a = \min\{\mathrm{Re}\, p_j\}_{j=1}^{n}$$

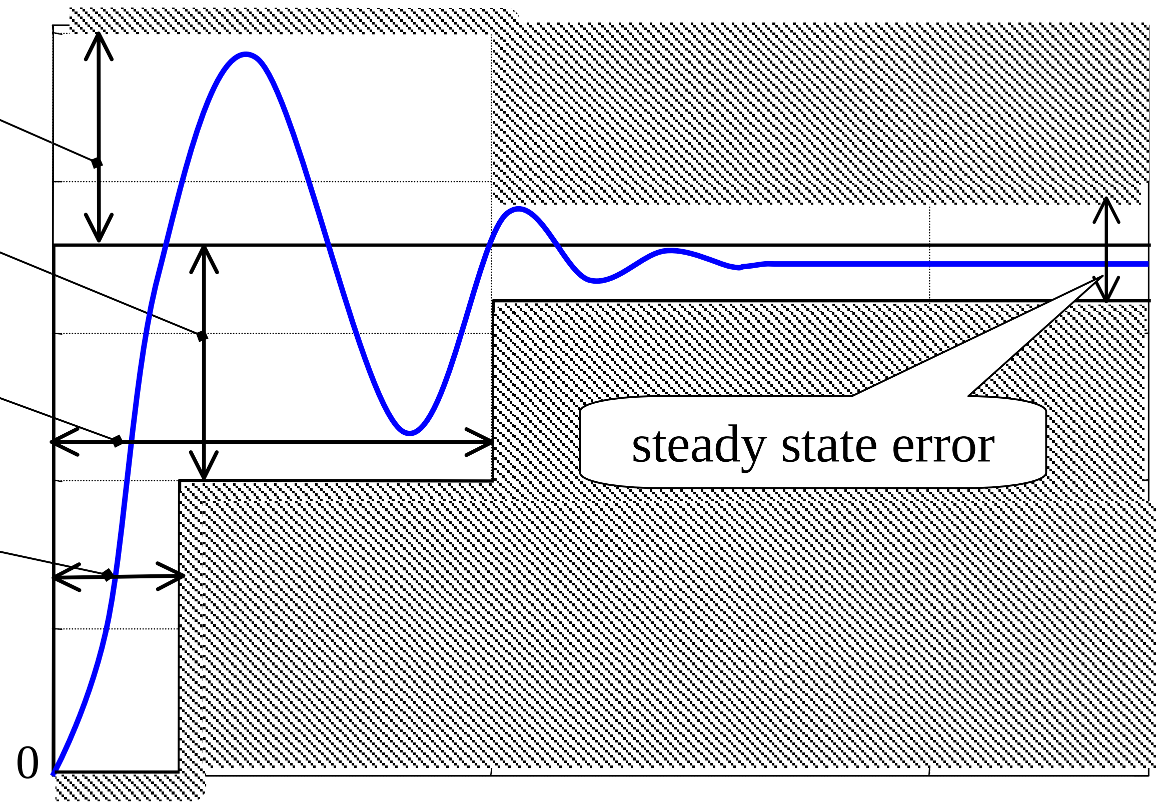


- Caution! Fast response (poles far to the left) leads to peaking



Performance - step response

- Step response shape characterization:
  - overshoot
  - undershoot
  - settling time
  - rise time

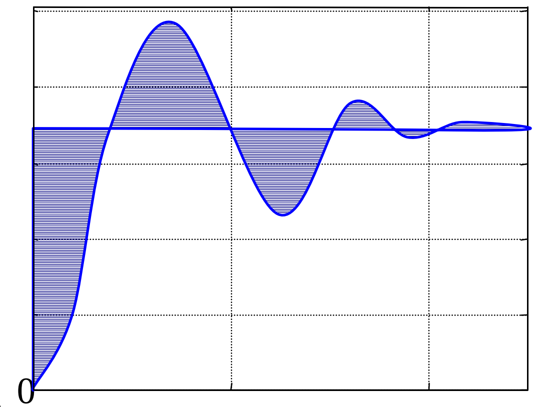


Performance - quadratic index

- Quadratic performance
  - response, in frequency domain

$$J = \int_{t=0}^{\infty} \left| y(t) - y_d(t) \right|^2 dt = \frac{1}{2\pi} \int_{\infty}^{\infty} \left| \tilde{e}(i\omega) \right|^2 d\omega =$$

$$\frac{1}{2\pi} \int \left| S(i\omega) \tilde{y}_d(i\omega) \right|^2 d\omega = \frac{1}{2\pi} \int \left| S(i\omega) \right|^2 \underbrace{\frac{1}{\omega^2}}_{\text{STEP}} d\omega$$

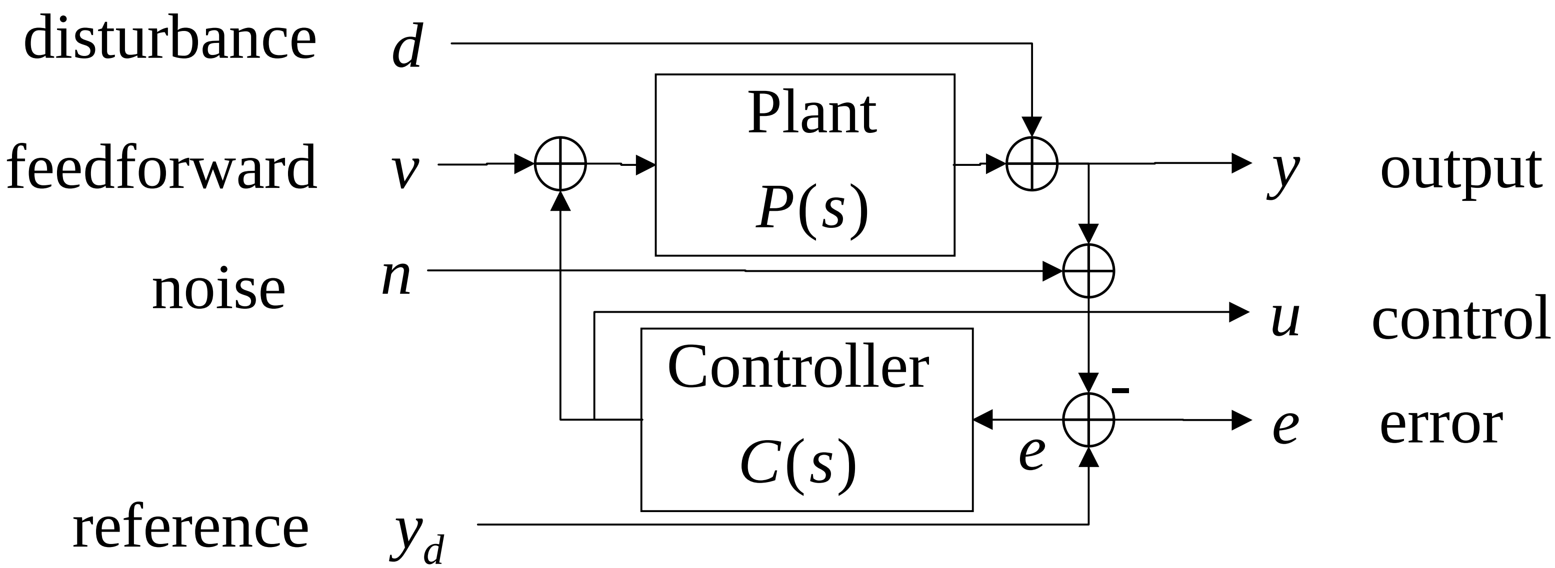


$$S(s) = \left[1 + P(s)C(s) \right]^{-1}$$

- If $y_d(t)$ is a zero mean random process with the spectral power $Q(i\omega)$

$$J = E\left(\int_{t=0}^{\infty} \left| y(t) - y_d(t) \right|^2 dt \right) = \frac{1}{2\pi} \int \left| S(i\omega) \right|^2 Q(i\omega) d\omega$$

Transfer functions in control loop



$$e = S(s)d - S(s)y_d + T(s)n + S_y(s)v$$

$$y = S(s)d + T(s)y_d + T(s)n + S_y(s)v$$

$$u = -S_u(s)d + S_u(s)y_d + S_u(s)n + T(s)v$$

Transfer functions in control loop

$$e = y - y_d + n$$
$$y = P(s)(u+v) + d$$
$$u = -C(s)e$$

$\Rightarrow$

$$e = S(s)d - S(s)y_d + T(s)n + S_y(s)v$$
$$y = S(s)d + T(s)y_d + T(s)n + S_y(s)v$$
$$u = -S_u(s)d + S_u(s)y_d + S_u(s)n + T(s)v$$

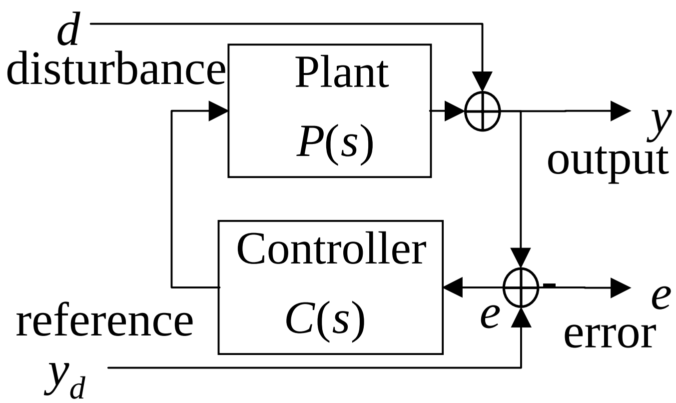
Sensitivity $S(s) = [1 + P(s)C(s)]^{-1}$

Complementary sensitivity $T(s) = [1 + P(s)C(s)]^{-1} P(s)C(s)$

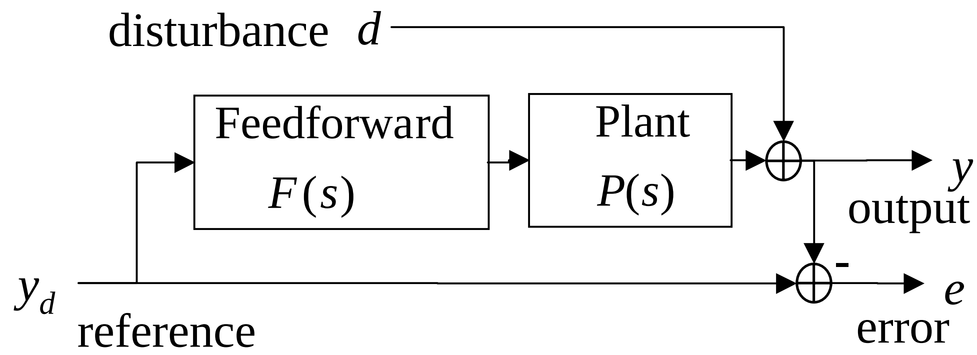
Noise sensitivity $S_u(s) = [1 + P(s)C(s)]^{-1} C(s)$

Load sensitivity $S_y(s) = [1 + P(s)C(s)]^{-1} P(s)$

Sensitivities



$$y = S(s)d + T(s)y_d$$



$$y = d + F(s)P(s)y_d$$

$$S(i\omega) = \frac{1}{1 + L(i\omega)}, \qquad L(s) = P(s)C(s)$$

- Feedback sensitivity
  - $|S(i\omega)| << 1$ for $|L(i\omega)| >> 1$
  - $|S(i\omega)| \approx 1$ for $|L(i\omega)| << 1$
  - can be bad for $|L(i\omega)| \approx 1$ - ringing, instability

$$S_{FF}(i\omega) = 1$$

- Feedforward sensitivity
  - good for any frequency
  - never unstable

Sensitivity requirements

$$e = S(s)d - S(s)y_d + T(s)n + S_y(s)v$$
$$y = S(s)d + T(s)y_d + T(s)n + S_y(s)v$$
$$u = -S_u(s)d + S_u(s)y_d + S_u(s)n + T(s)v$$

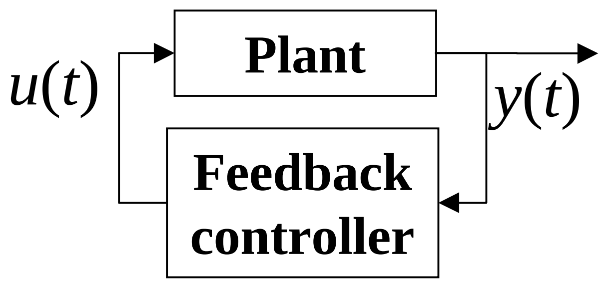
$$S(i\omega) = \frac{1}{1 + P(i\omega)C(i\omega)}$$
$$S_y(i\omega) = \frac{P(i\omega)}{1 + P(i\omega)C(i\omega)}$$
$$S_u(i\omega) = \frac{C(i\omega)}{1 + P(i\omega)C(i\omega)}$$

- Disturbance rejection and reference tracking
  - $|S(i\omega)| \ll 1$ for the disturbance d ; $|S_y(i\omega)| \ll 1$ for the input 'noise' v
- Limited control effort
  - $|S_u(i\omega)| \ll 1$ conflicts with disturbance rejection where $|P(i\omega)| < 1$
- Noise rejection
  - $|T(i\omega)| \ll 1$ for the noise n, conflicts with disturbance rejection

Robustness

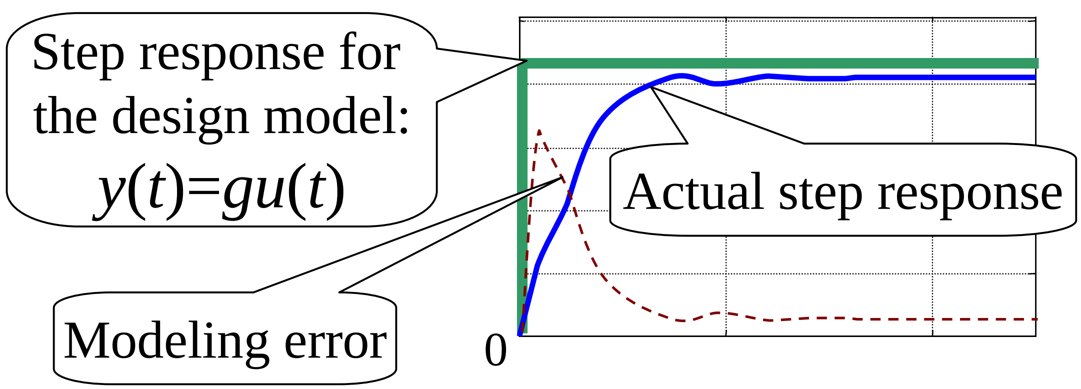
- Ok, we have a controller that works for a *nominal* model.
- Why would it ever would work for *real system*?
  - Will know for sure only when we try - V&V - similar to debugging process in software
- Can check that controller works for a *range* of different models and hope that the real system is covered by this range
  - This is called robustness analysis, robust design
  - Was an implicit part of the classical control design - Nyquist, Bode
  - Multivariable robust control - Honeywell: G.Stein, G.Hartmann, ‘81
  - Doyle, Zames, Glover - robust control theory

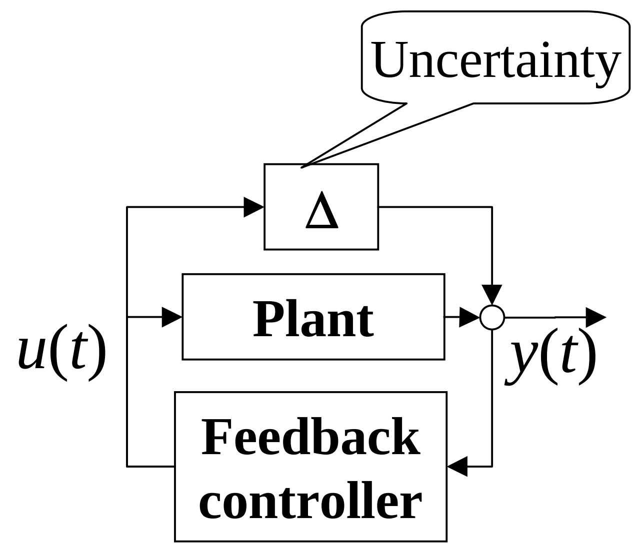
Control loop analysis



$$y = gu$$
$$\dot{u} = -k_I(y - y_d)$$
$$\Rightarrow \quad \dot{y} + gk_I(y - y_d) = 0$$

- Why control might work if the process differs from the model?
- Key factors
  - modeling error (uncertainty) characterization
  - time scale (bandwidth) of the control loop





Robustness - Small gain theorem

- Nonlinear uncertainty!

- Operator gain

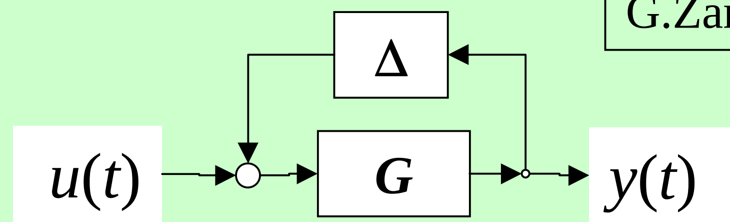
  $$\|Gu\| \le \|G\| \cdot \|u\|$$

  – G **can be a nonlinear operator**

- L_2 norm

$$\|u\|^2 = \int u^2(t)dt = \frac{1}{2\pi}\int |\tilde{u}(i\omega)|^2 d\omega$$

G.Zames



The loop is guaranteed stable if

$$\|G\| \cdot \|\Delta\| < 1$$

(Open-loop stability assumed)

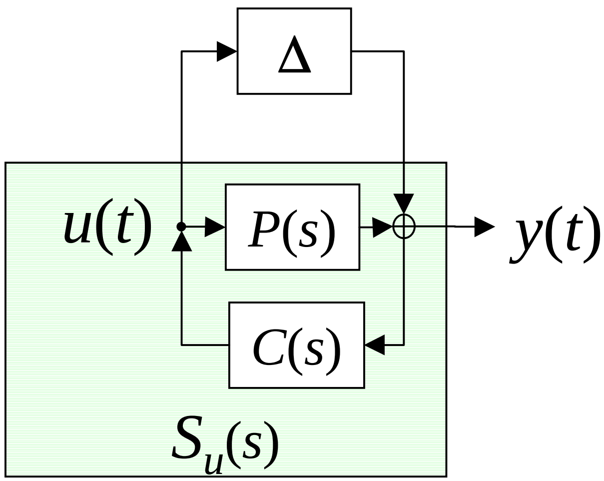
Desoer and Vidyasagar, *Feedback Systems: Input-Output Properties*, 1975

- L_2 gain of a linear operator

$$\|Gu\|^2 = \frac{1}{2\pi}\int |G(i\omega)\tilde{u}(i\omega)|^2 d\omega \le \underbrace{\sup\left(|G(i\omega)|^2\right)}_{\|G\|^2} \cdot \underbrace{\frac{1}{2\pi}\int |\tilde{u}(i\omega)|^2 d\omega}_{\|u\|^2}$$

Robustness

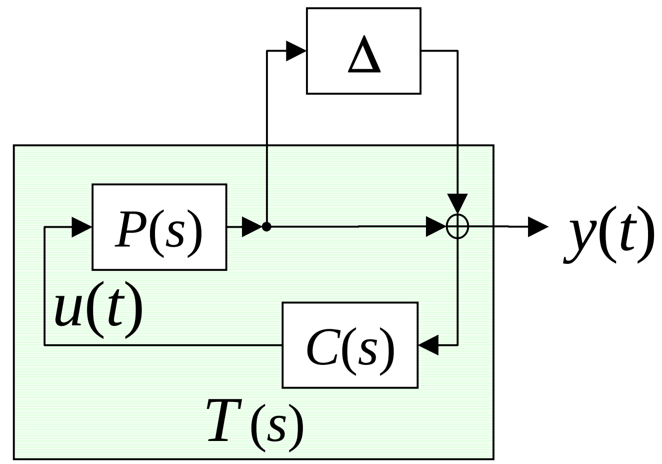
- Additive uncertainty



Condition of robust stability

$$\underbrace{\left|\frac{C(i\omega)}{1+P(i\omega)C(i\omega)}\right|}_{\|S_u\|}\cdot\underbrace{\left|\Delta(i\omega)\right|}_{\|\Delta\|}<1$$

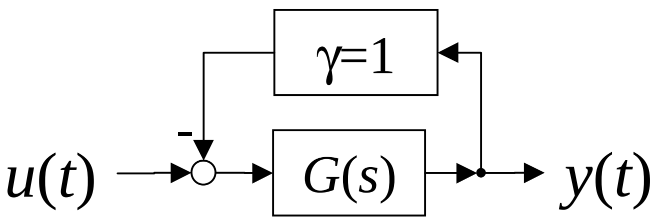
- Multiplicative uncertainty

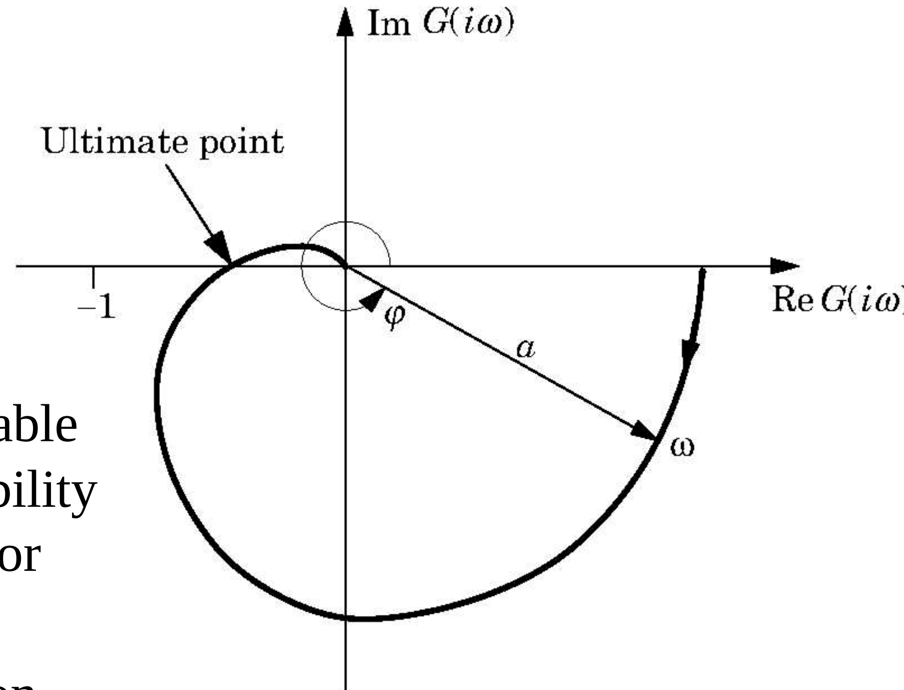


Condition of robust stability

$$\underbrace{\left|\frac{P(i\omega)C(i\omega)}{1+P(i\omega)C(i\omega)}\right|}_{\|T\|}\cdot\underbrace{\left|\Delta(i\omega)\right|}_{\|\Delta\|}<1$$

Nyquist stability criterion

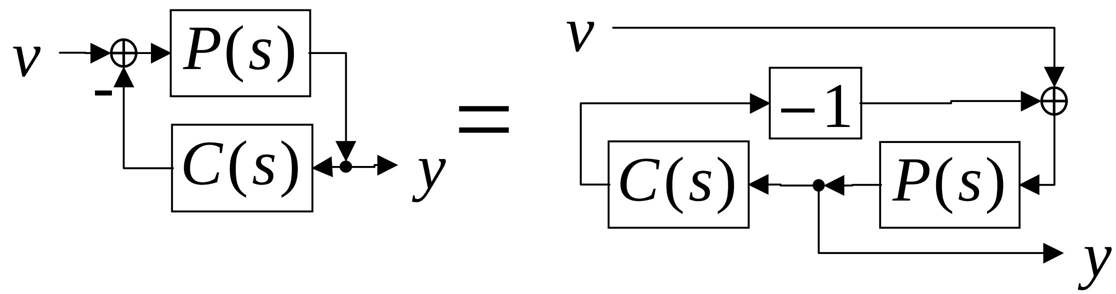




- Homotopy "Proof"
  - $G(s)$ is stable, hence the loop is stable for $\gamma=0$. Increase γ to 1. The instability cannot occur unless $\gamma G(iw)+1=0$ for some $0 \leq \gamma \leq 1$.
  - $|G(i\omega_{180})|<1$ is a *sufficient* condition
- Subtleties: r.h.p. poles and zeros
  - Formulation and real proof using the agrument principle, encirclements of -1
  - stable $\rightarrow$ unstable $\rightarrow$ stable as $0\rightarrow\gamma\rightarrow 1$

Compare against Small Gain Theorem:

Gain and phase margins

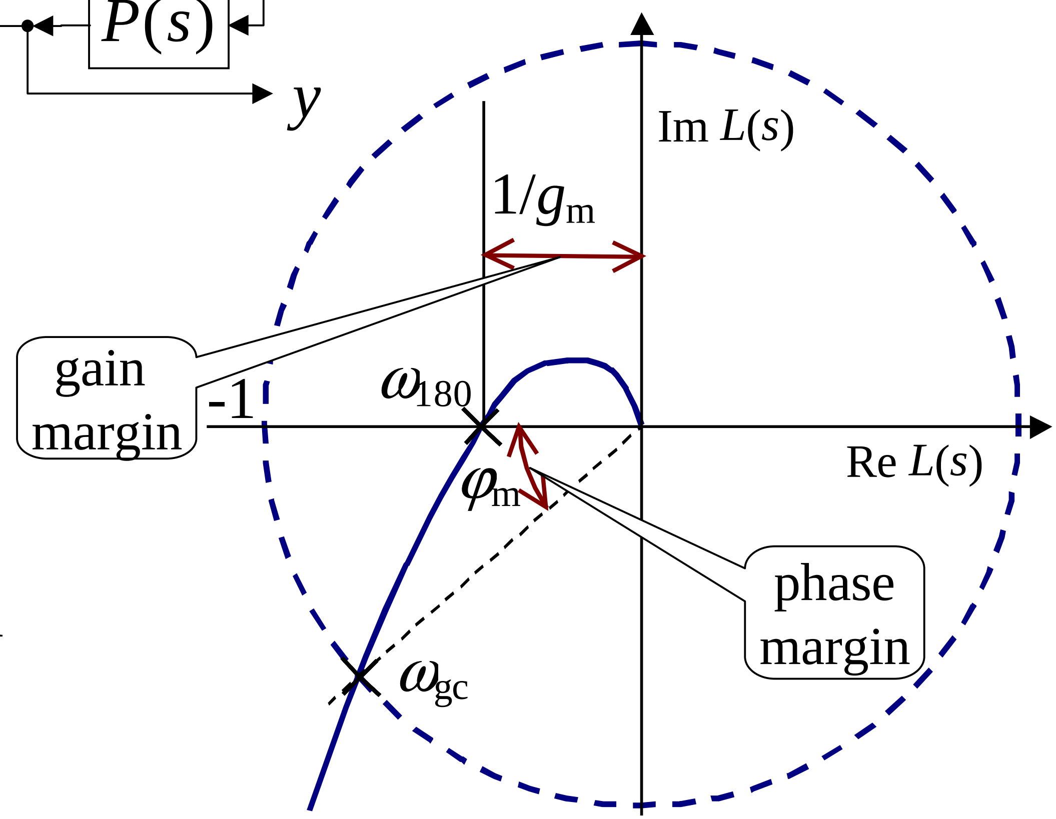


- Loop gain

  $$L(s) = P(s)C(s)$$

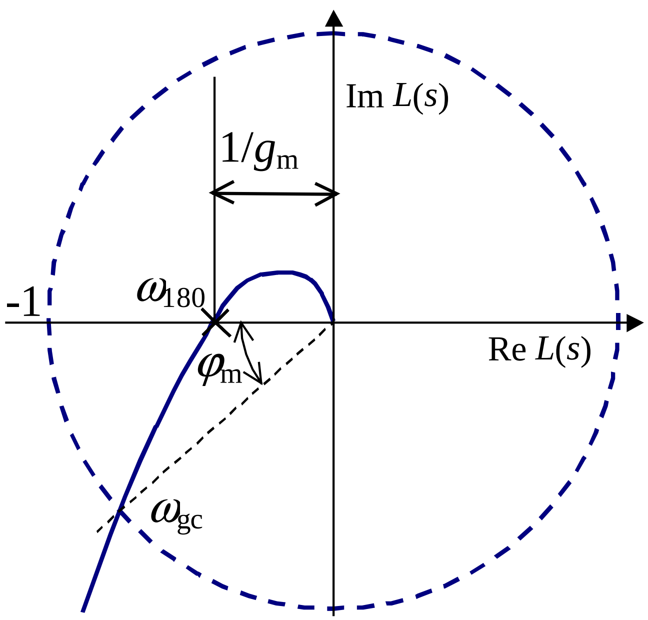
  $$S(s) = [1 + L(s)]^{-1}$$

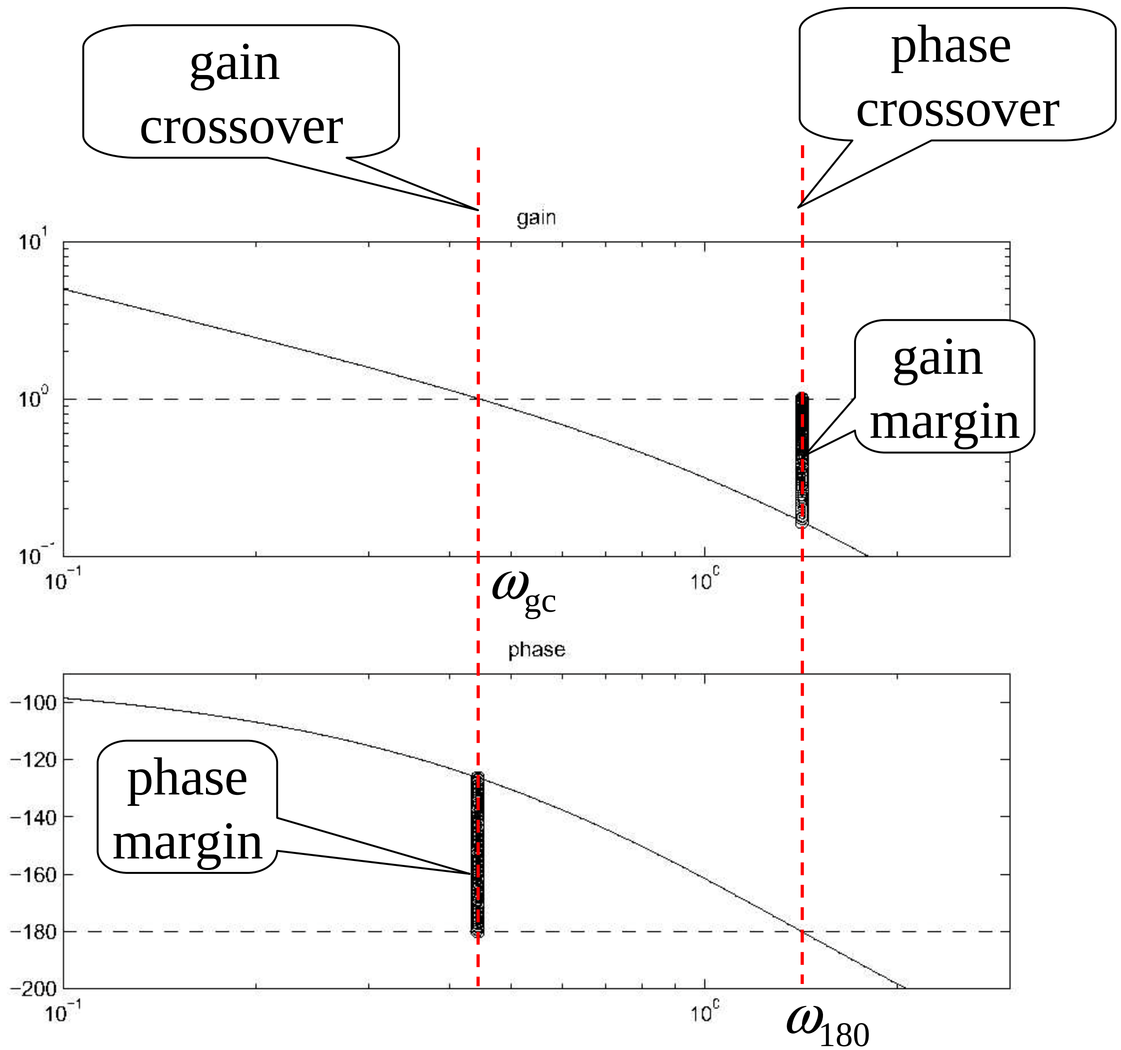
- Nyquist plot for L
  - at high frequency $|L(i\omega)| \le 1$



Gain and phase margins

- Bode plots





Advanced Control

- Observable and controllable system
  - Can put poles anywhere
  - Can drive state anywhere
- Why cannot we just do this?
  - Large control
  - Error peaking
  - Poor robustness, margins
    - Observability and controllability = matrix rank
    - Accuracy of solution is defined by condition number
- Analysis of this lecture is valid for *any* LTI control, including advanced