Control Theory and Congestion

Glenn Vinnicombe and Fernando Paganini

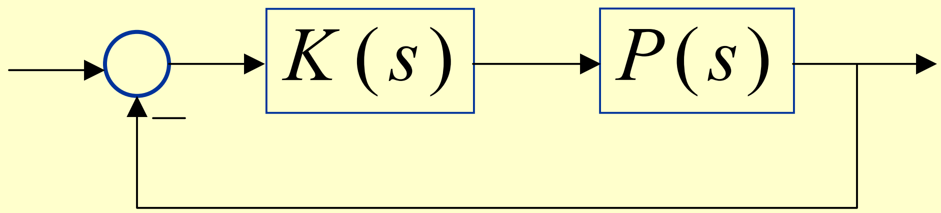
Cambridge/Caltech and UCLA

IPAM Tutorial – March 2002.

Outline of second part:

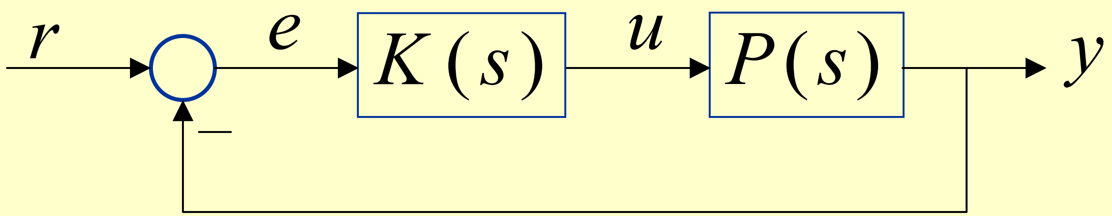
1. Performance in feedback loops: tracking, disturbance rejection, transient response. Integral control.
2. Fundamental design tradeoffs. The role of delay. Bode Integral formula
3. Extensions to multivariable control.

Performance of feedback loops



- Stability and its robustness are essential properties; however, they are only half of the story.
- The closed loop system must also satisfy some notion of performance:
  - Steady-state considerations (e.g. tracking errors).
  - Disturbance rejection.
  - Speed of response (transients, bandwidth of tracking).
- Performance and stability/robustness are often at odds.
- For single input-output systems, frequency domain tools (Nyquist, Bode) are well suited for handling this tradeoff.

Performance specs 1: Steady-state tracking



$$e(t) = r(t) - y(t)$$

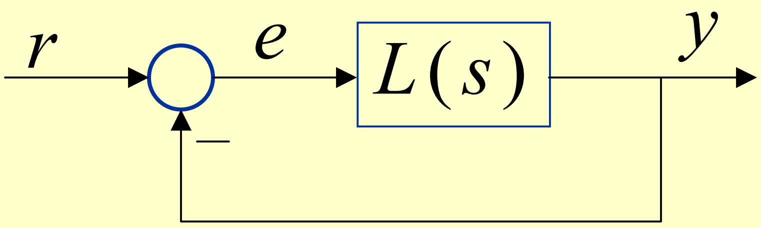
Error between reference signal r and output y.

Tracking means this error is kept small.

Suppose that $r(t) = r_0$, constant, and that the system is stable. Then as $t \to \infty$, $e(t) \to e(\infty)$, steady-state error.

Ideally, we would like the steady-state error to be zero.

Tracking, sensitivity and loop gain



The mapping from $r(t)$ to $e(t)$ has transfer function $S(s) = \dfrac{1}{1+L(s)}$.

That is, $R(s) = S(s)E(s)$ in Laplace.

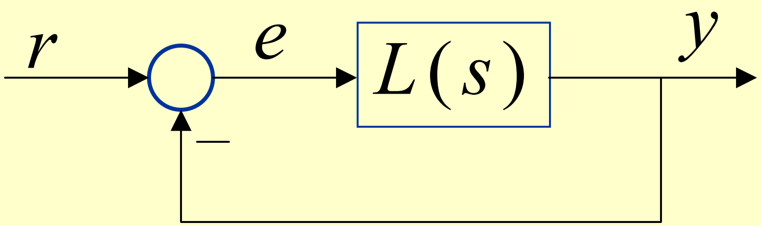
$S(s)$ is called the sensitivity function of the system.

Under stability, $S(s)$ has no poles in $\text{Re}[s] \geq 0$.

Then for $r(t) \equiv r_0$, we have $e(\infty) = S(0)r_0 = \dfrac{1}{1+L(0)} r_0$

Good steady-state tracking $\leftrightarrow$ $S(0)$ small $\leftrightarrow$ $L(0)$ large.

Integral control



Suppose $L(s)$ has a pole at $s = 0$.

Then $S(0) = \dfrac{1}{1 + L(0)} = 0.$

Zero steady-state error!

Example: $L(s) = \dfrac{K}{s}.$ $\qquad \dot{y}(t) = K\left(r - y\right).$

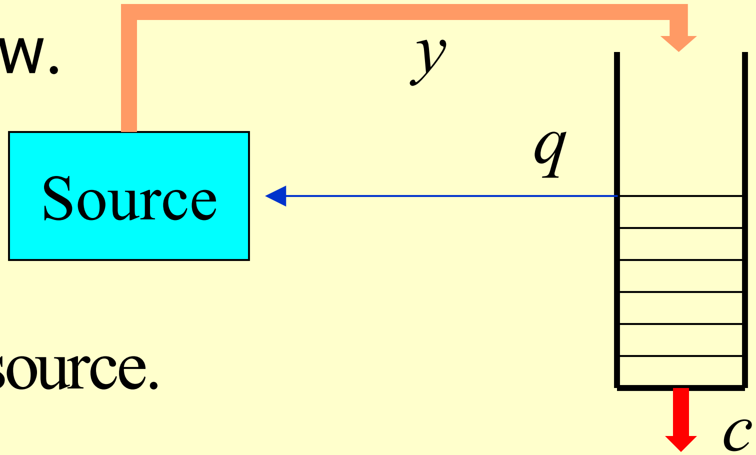
Loop is stable for $K > 0,$ and has a pole at $s = 0.$

Therefore, it has zero steady-state tracking error.

In the time domain: for $r(t) \equiv r_0$, $\boxed{y(t) = r_0\left(1 - e^{-Kt}\right) \to r_0}$

Simple congestion control example

Single link/source, no delays for now.



y: Transmission rate (pkts/sec)

c : Capacity of the link (pkts/sec)

q : Queue size; assume it is fed back to source.

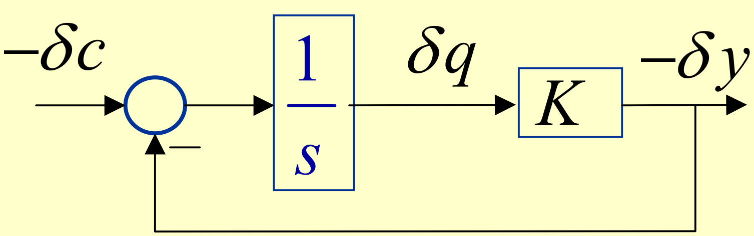
Suppose source control is $y = f(q)$, where f is a decreasing function.

Model: $\boxed{\dot{q} = y - c = f(q) - c}$ Equilibrium for $c = c_0$: $f(q_0) = y_0 = c_0$.

Linearize around it: $c = c_0 + \delta c,\ y = y_0 + \delta y,\ q = q_0 + \delta q.$

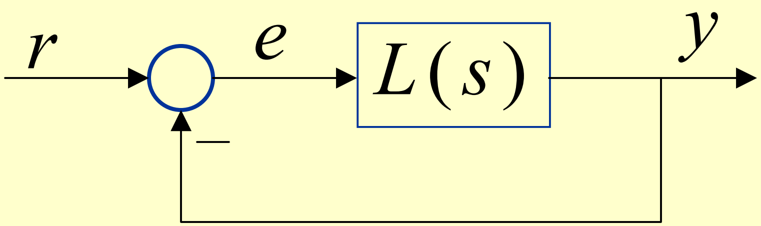
$$\delta y = -K\delta q,\ \ K = -\frac{\partial f}{\partial q}(q_0) > 0.$$

$$\delta \dot{q} = \delta y - \delta c.$$



Integral Control $\Rightarrow$ Perfect steady-state tracking for constant δc.

Performance specs 2: tracking of low-frequency reference signals.



Transfer function from $r(t)$ to $e(t)$ is the sensitivity $S(s) = \dfrac{1}{1+L(s)}$.

Let $S(j\omega) = |S(j\omega)|e^{j\phi_S(\omega)}$ be the polar decomposition.

Assume the system is stable: then the steady-state response to a sinuoidal reference $r(t) = r_0 \cos(\omega_0 t)$ is

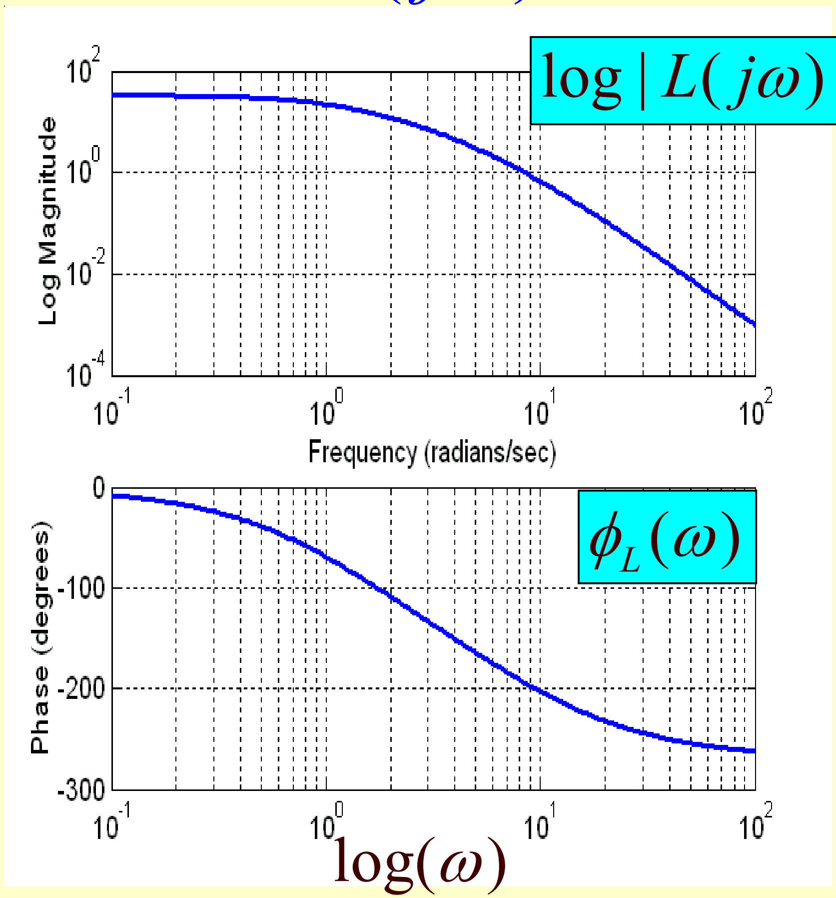
$e(t) = r_0 \,|S(j\omega_0)| \cos(\omega_0 t + \phi_S(\omega_0))$.

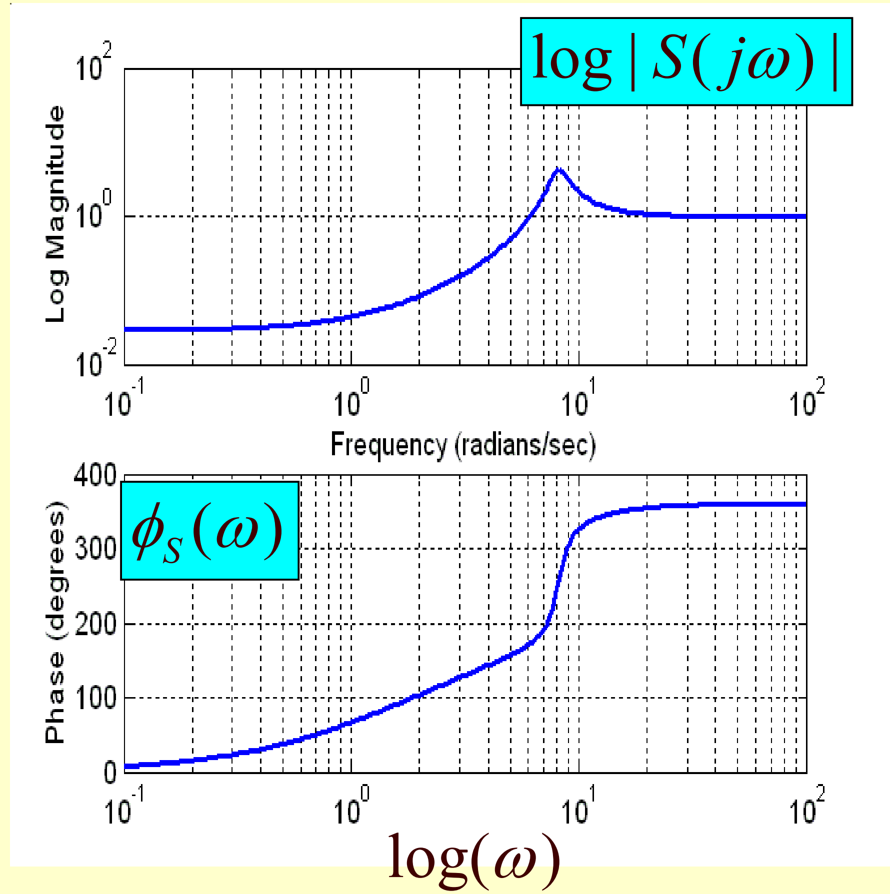
Good steady-state tracking $\leftrightarrow$ $|S(j\omega_0)|$ small $\overset{\approx}{\longleftrightarrow}$ $|L(j\omega_0)|$ large.

Representation of frequency functions

Bode plot

$L(j\omega)$ $S(j\omega)$

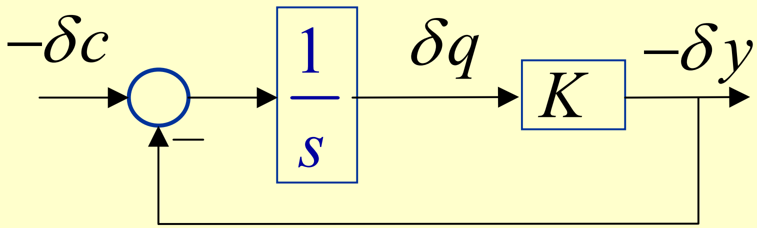


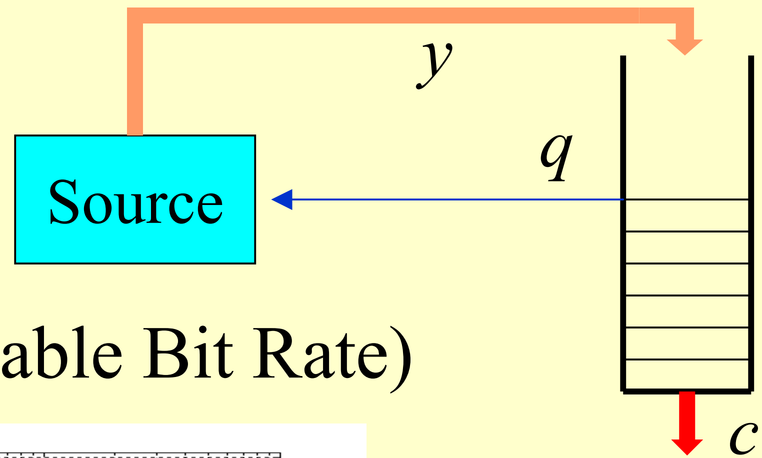


Tracking $\leftrightarrow$ Large $|L(j\omega)| \leftrightarrow$ Small $|S(j\omega)|$

in frequency range of interest.

Example 2: tracking of variable references

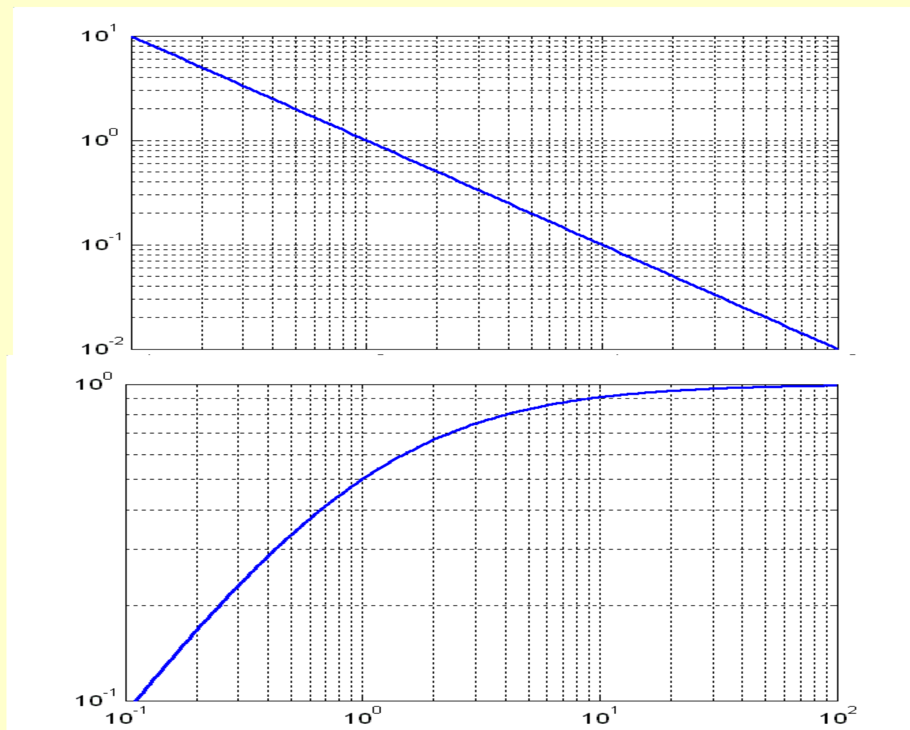




Variations in capacity (e.g. Available Bit Rate)

$\log|L(j\omega)|$

$\log|S(j\omega)|$

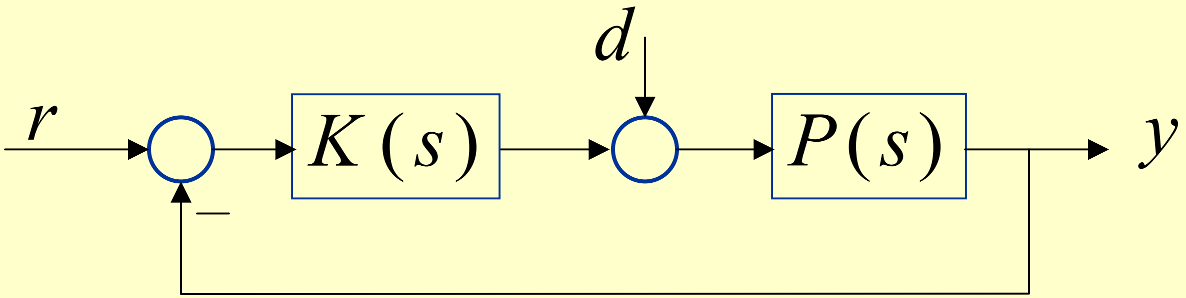


$\log \omega$

$$L(j\omega) = \frac{K}{j\omega}$$

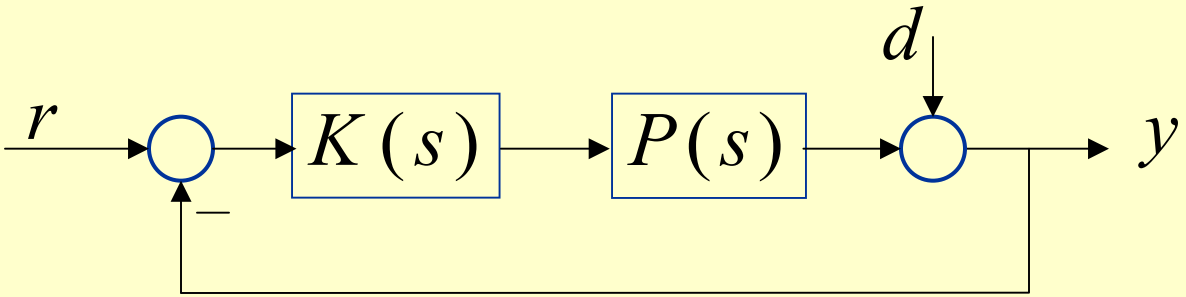
As K grows, track higher bandwidth

Performance specs 3: disturbance rejection.



Input disturbance:

$$T_{d\to y}(s) = \frac{P(s)}{1+L(s)}$$

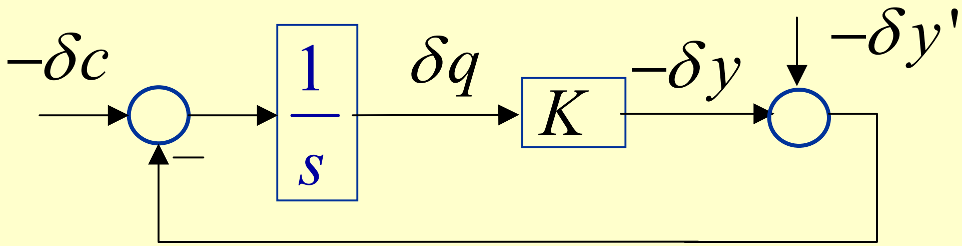


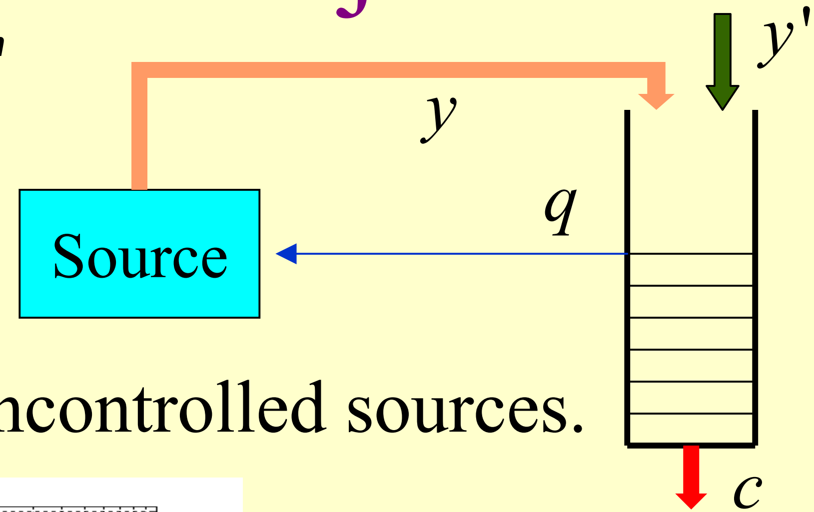
Output disturbance:

$$T_{d\to y}(s) = \frac{1}{1+L(s)}$$

To reject disturbances, we need attenuation in the frequency range of interest $\approx$ Large $|L(j\omega)|$.

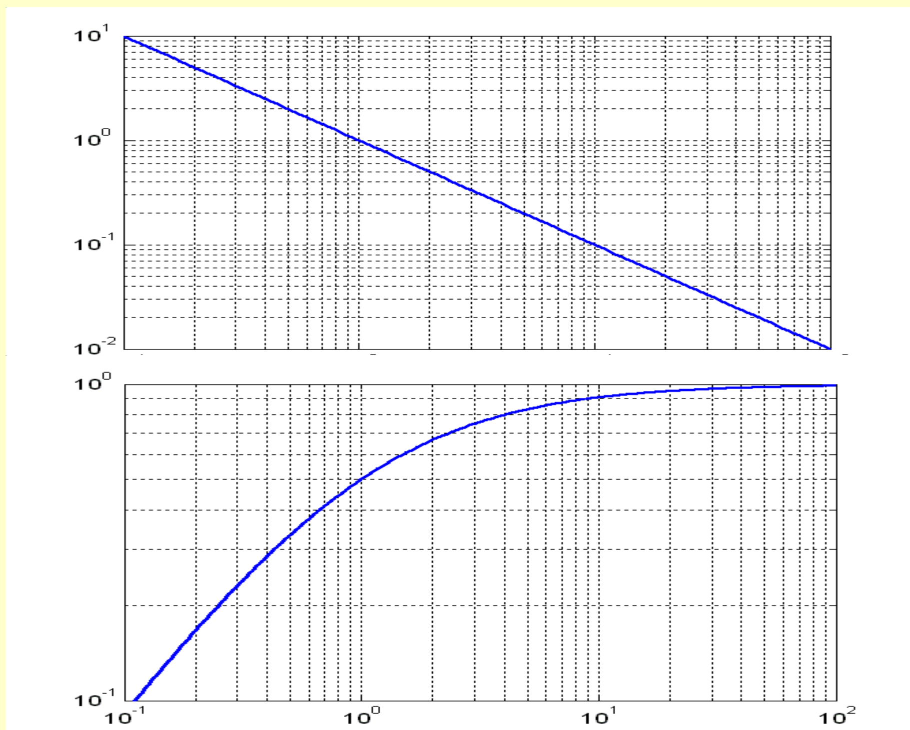
Example 3: disturbance rejection





Noise traffic generated by other uncontrolled sources.

$\log |L(j\omega)|$



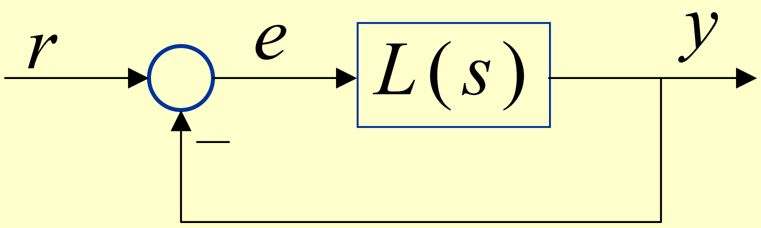
$\log |S(j\omega)|$

$\log \omega$

$$L(j\omega) = \frac{K}{j\omega}$$

As K grows, reject disturbances over a higher bandwidth

Performance specs 4: speed of response

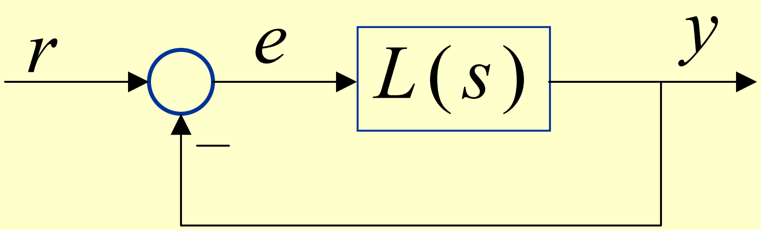


- Superimposed to the steady-state solutions discussed before, we have transient terms of the form $\sum_i C_i e^{-s_i t}$. Here the modes s_i are the roots of $1+L(s)=0$.
- For fast response, $\mathrm{Re}[s_i]$ must be as negative as possible.

Example: $L(s)=\dfrac{K}{s}$. $\quad 1+L(s)=0 \leftrightarrow s_1=-K.$

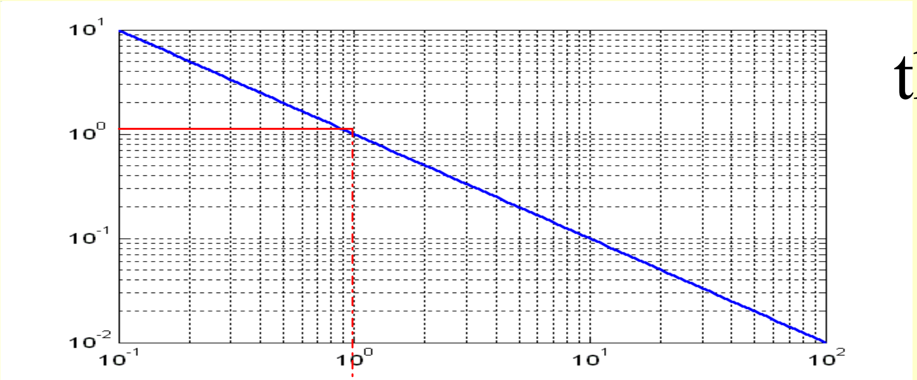
The higher K, the faster our transient response.

For instance for $r(t)\equiv r_0$, solution is $y(t)=r_0\left(1-e^{-Kt}\right)$



Heuristic look based on Fourier: frequencies where $|L(j\omega)| << 1$ cannot occur (filtered out). So the speed of response is roughly the bandwidth where $|L(j\omega)| \geq 1$.

$\log |L(j\omega)|$



ω_c

Transient decays in a time of the order of $\dfrac{1}{\omega_c}$

For $L(s) = \dfrac{K}{s}$ (e.g. our congestion control with queue feedback)

$\omega_c = K \rightarrow$ decays in the order of $\dfrac{1}{K}$ seconds.

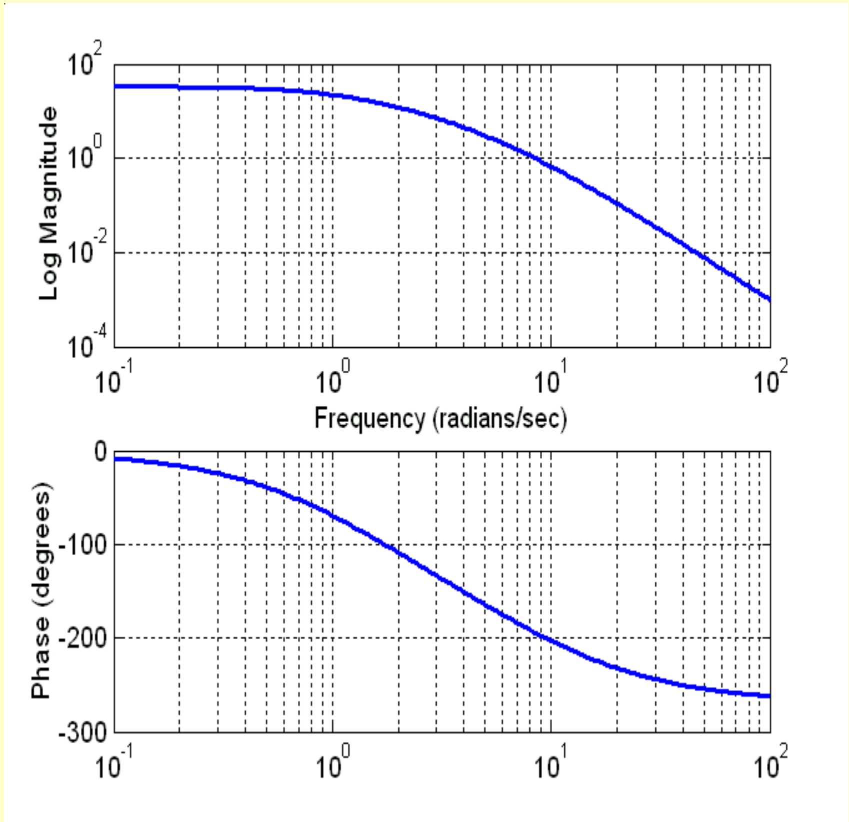
For faster response, increase the open loop bandwidth.

Performance specifications: recap

- Tracking of constant, or varying reference signals.
- Disturbance rejection.
- Transient response.

Rule of thumb for all: increase the gain or bandwidth of the loop transfer function $L(j\omega)$.

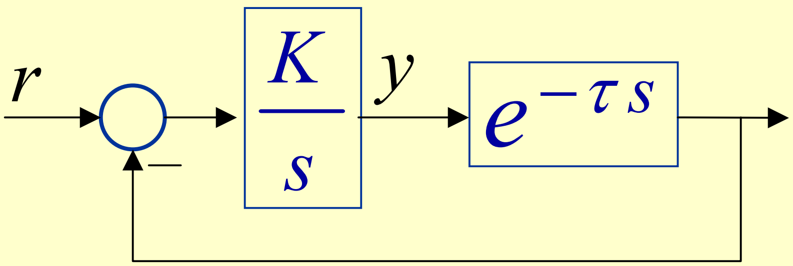
What stops us from arbitrarily good performance?
Answer: stability/robustness.

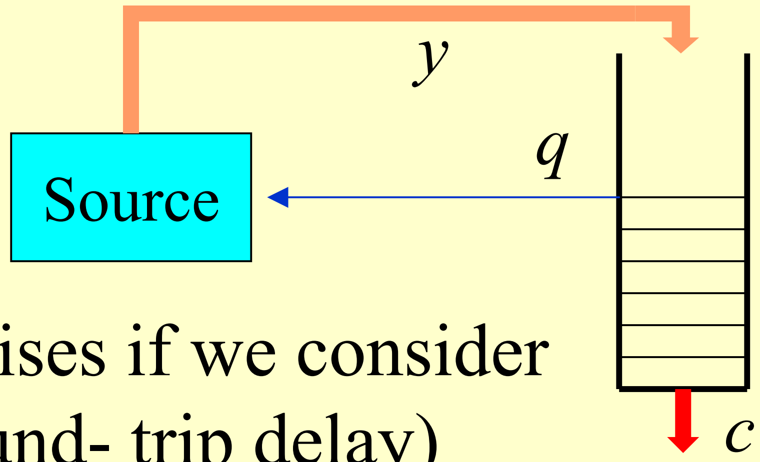


Example: loop with integrator and delay.

$$\dot{y}(t) = K\left(r(t) - y(t-\tau)\right)$$

Example:





$$L(s) = \frac{Ke^{-\tau s}}{s}$$

(arises if we consider round- trip delay)

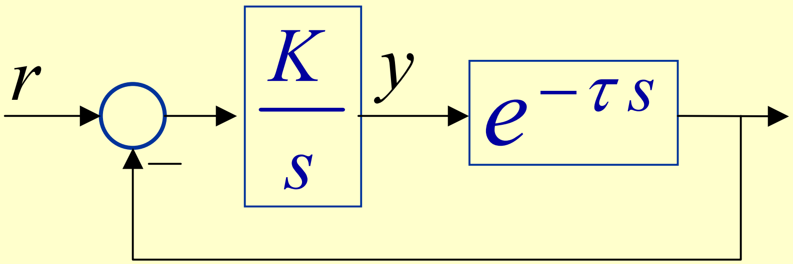
$$L(j\omega) = \frac{Ke^{-\tau j\omega}}{j\omega} \Rightarrow \left|L(j\omega)\right| = \frac{K}{|\omega|}$$ (independent of delay!)

Our earlier rule says: increase K for performance.

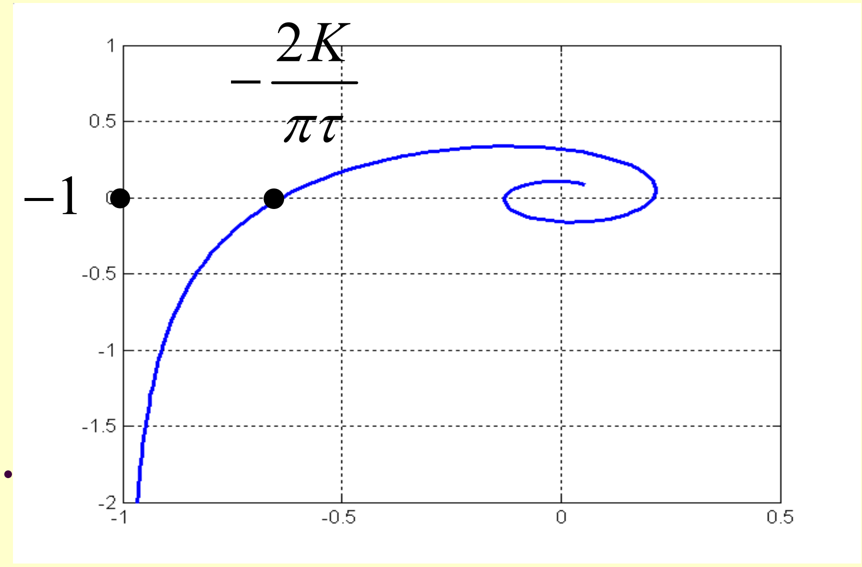
Stability? $1 + L(s) = 0 \leftrightarrow s + Ke^{-\tau s} = 0.$

Transcendental equation. However, use Nyquist.

Stability analysis via Nyquist:



Nyquist plot of $L(j\omega)$:



Loop function $L(s) = \dfrac{Ke^{-\tau s}}{s}$

$$|L(j\omega)| = \frac{K}{|\omega|}, \quad \phi(\omega) = -\frac{\pi}{2} - \omega\tau.$$

To avoid encirclements, impose

$$|L(j\omega_0)| < 1 \text{ at } \omega_0 \text{ where } \phi(\omega_0) = -\pi$$

Stable for $K < \dfrac{\pi}{2\tau}$

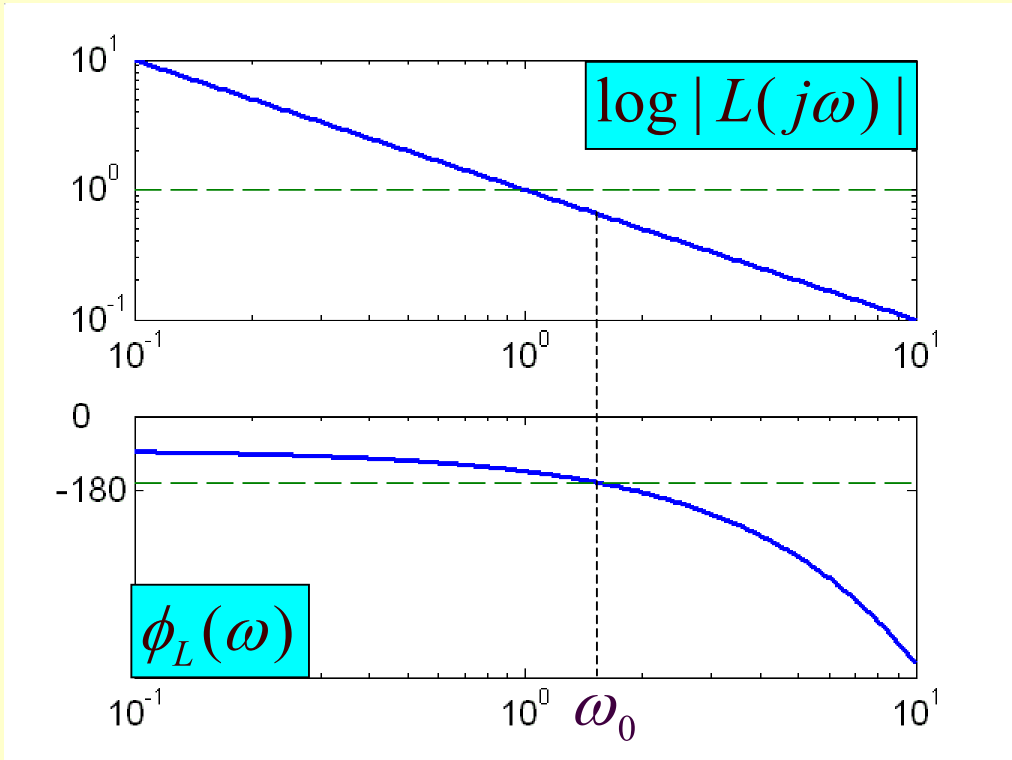
Not much harder than analysis without delay!
Much simpler than other alternatives
(transcendental equations, Lyapunov functionals,…)

Stability in the Bode plot

$$|L(j\omega)| = \frac{K}{|\omega|},\ \phi(\omega) = -\frac{\pi}{2} - \omega\tau.$$

Impose $|L(j\omega_0)| < 1$

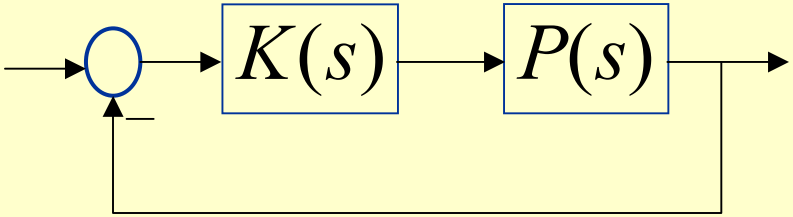
at ω_0: $\phi(\omega_0) = -\pi$



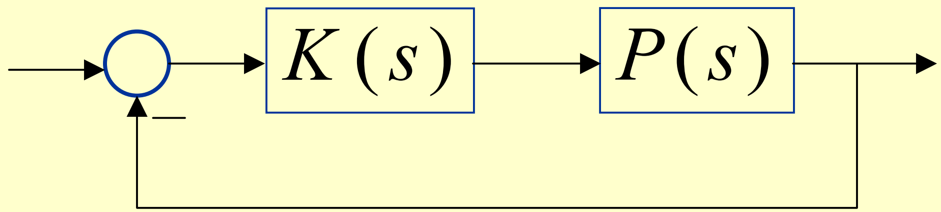
Increasing K moves the top plot upwards.

$\Rightarrow$ Constraint on K for stability.

Conclusion: delay limits the achievable performance. Also, other dynamics of the plant (known or uncertain) produce a similar effect.

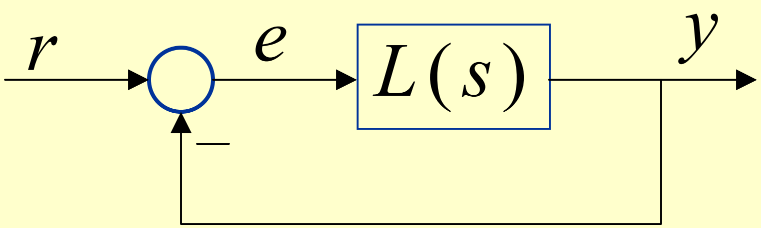


The performance/robustness tradeoff



- As we have seen, we can improve performance by increasing the gain and bandwidth of the loop transfer function $L(jw)$.
- $L(s)$ can be designed through $K(s)$. By canceling off $P(s)$, one could think $L(s)$ would be arbitrarily chosen. However:
  - Unstable dynamics cannot be canceled.
  - Delay cannot be canceled (othewise $K(s)$ would not be causal).
  - Cancellation is not robust to variations in $P(s)$.
- Therefore, the given plant poses essential limits to the performance that can be achieved through feedback.
- Good designs address this basic tradeoff. For single I-O systems,"loopshaping" the Bode plot is an effective method.

The Bode Integral formula



Recall: the mapping from $r(t)$ to $e(t)$ has transfer function $S(s) = \dfrac{1}{1+L(s)}$.

For tracking, we want the sensitivity $|S(j\omega)|$ to be small, for as large a frequency range as possible. How large can it be?

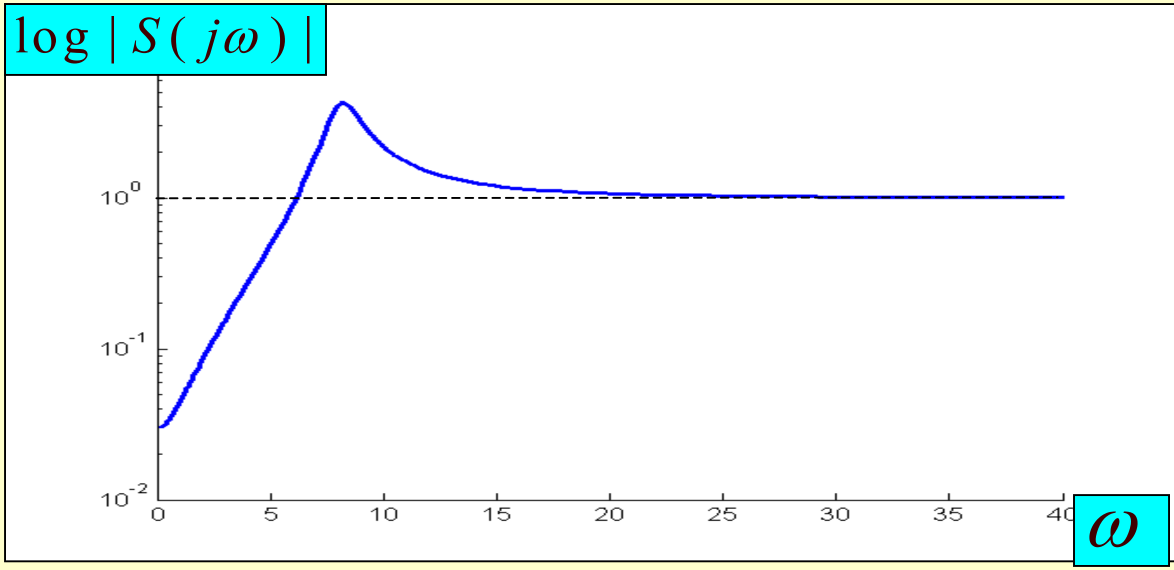
Theorem (Bode): Suppose $L(s) = \dfrac{n(s)}{d(s)}$, a rational function with $\deg(d(s)) - \deg(n(s)) \geq 2.$

Let $\{p_i\}$ be the set of poles of $L(s)$ in $\mathrm{Re}[s] > 0$. Then

$$\int_0^\infty \log\left|S(j\omega)\right| d\omega = \pi \log(e) \sum \mathrm{Re}[p_i]$$

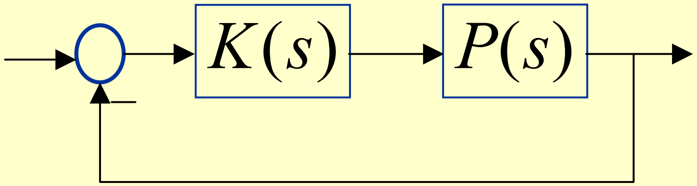
The Bode Integral formula.

$$\int_0^\infty \log|S(j\omega)|\,d\omega = \pi \log(e) \sum \mathrm{Re}[p_i]$$



The unstable poles p_i that come from the plant $P(s)$ cannot be eliminated by $K(s)$

$\Rightarrow$ Integral of sensitivity is a conserved quantity over all stabilizing feedbacks.

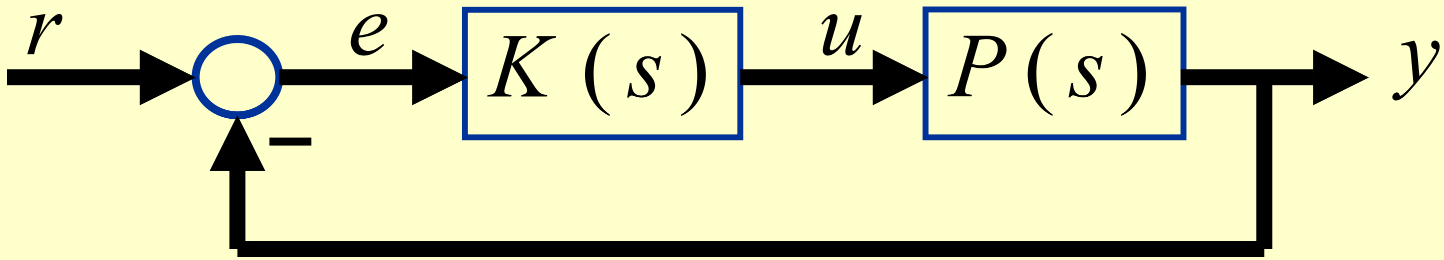


Small sensitvity at low frequencies must be "paid" by a larger than 1 sensitivity at some higher frequencies.

But all this is only linear!

- The above tradeoff is of course present in nonlinear systems, but harder to characterize, due to the lack of a frequency domain (partial extensions exist).
- So most successful designs are linear based, followed up by nonlinear analysis or simulation.
- Beware of claims of superiority of "truly nonlinear" designs. They rarely address this tradeoff, so may have poor performance or poor robustness (or both).
- A basic test: linearized around equilibrium, the nonlinear controller should not be worse than a linear design.

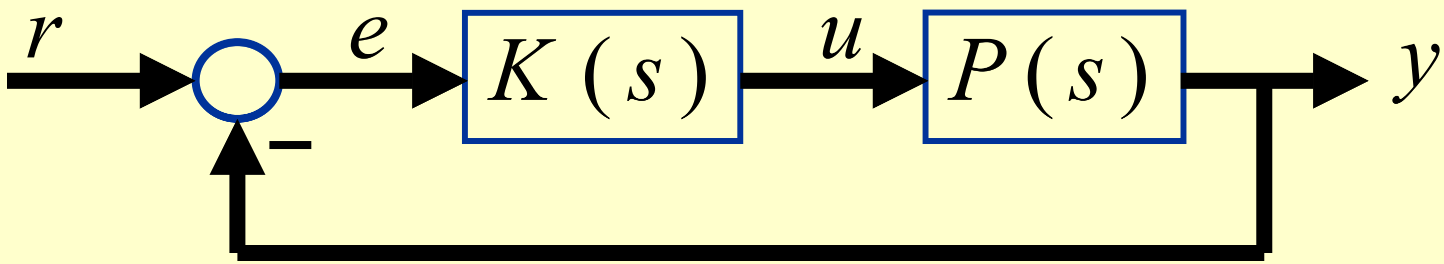
Multivariable control



Signals are now vector-valued (many inputs and outputs).
Transfer functions are matrix-valued.

$$y(s) = \begin{bmatrix} y_1(s) \\ \vdots \\ y_m(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & \cdots & P_{1n}(s) \\ \vdots & \ddots & \vdots \\ P_{m1}(s) & \cdots & P_{mn}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ \vdots \\ u_n(s) \end{bmatrix} = P(s)u(s)$$

$$\left.\begin{aligned} y(s) &= \overbrace{P(s)K(s)}^{L(s)} e(s) \\ e(s) &= r(s) - y(s) \end{aligned}\right\} \Rightarrow \left[I + L(s)\right] e(s) = r(s)$$



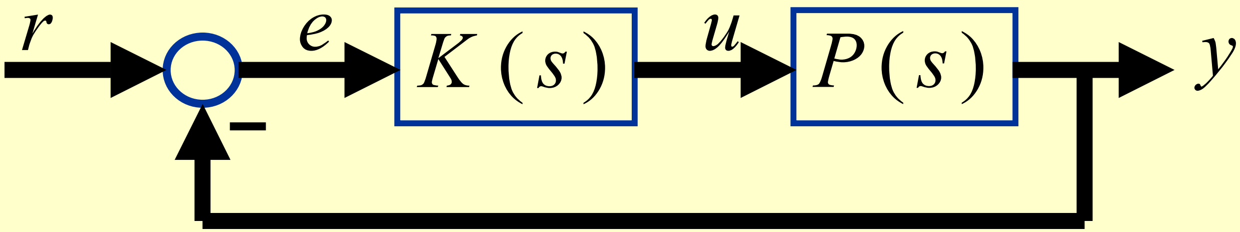
$$e(s) = \underbrace{\left[I + L(s)\right]^{-1}}_{S(s)} r(s)$$

$$y(s) = L(s)\left[I + L(s)\right]^{-1} r(s)$$

Stability: poles of $\left[I + L(s)\right]^{-1}$ (i.e., roots of $\det\left[I + L(s)\right] = 0$) must have negative real part.

Multivariable Nyquist criterion: study encirclements of the origin of $\det\left[I + L(j\omega)\right] = \prod\left(1 + \lambda_i(j\omega)\right)$, where $\lambda_i(j\omega)$ are the eigenvalues of $L(j\omega)$.

Performance of multivariable loops



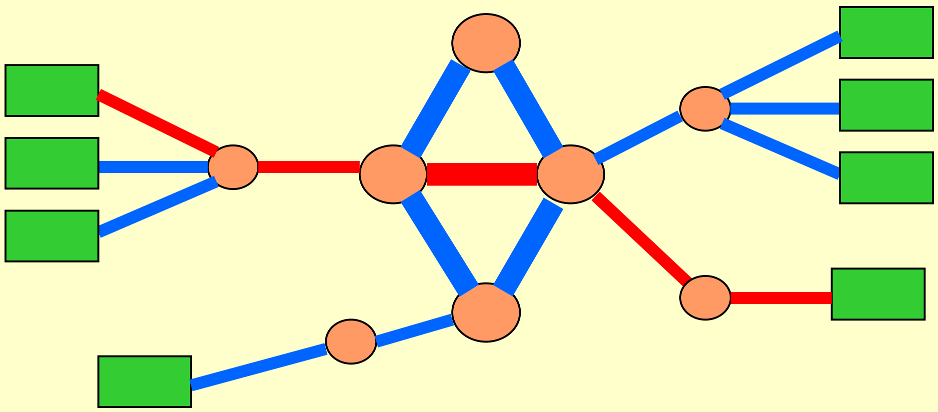
$$e(j\omega) = S(j\omega)\,r(j\omega) = \left[I + L(j\omega)\right]^{-1} r(j\omega)$$

The tracking error will depend on frequency, but also on the direction of the vector $r(j\omega)$. The worst-case direction is captured by the maximum singular value:

$$\overline{\sigma}(S(j\omega)) = \max\left\{|\,S(j\omega)v\,| : v \in \mathbb{C}^n, |v| = 1\right\}.$$

Network congestion control example

L communication links shared by S source-destination pairs.



Routing matrix:

$$R_{li} = \begin{cases} 1 & \text{if link } l \text{ serves source } i \\ 0 & \text{otherwise} \end{cases}$$

x_i : Rate of $i-$th source (pkts/sec)

$y_l = \sum_{\text{i uses l}} x_i$: Total rate of $l-$th link (pkts/sec)

c_l : Capacity of the $l-$th link (pkts/sec)

b_l : Backlog of the $l-$th link (pkts)

$q_i = \sum_{\text{i uses l}} b_l$: Total backlog for $i-$th source (pkts)

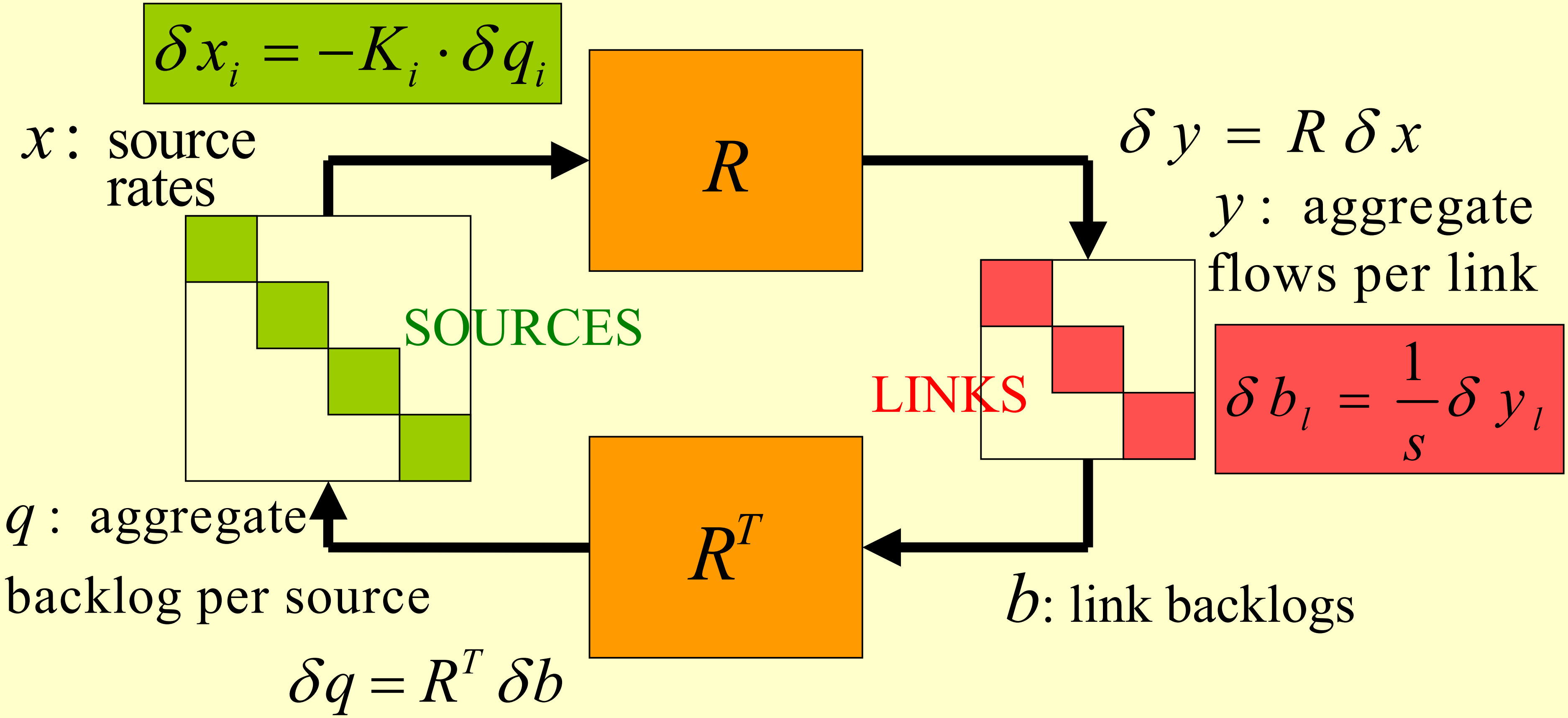
$$y = Rx$$

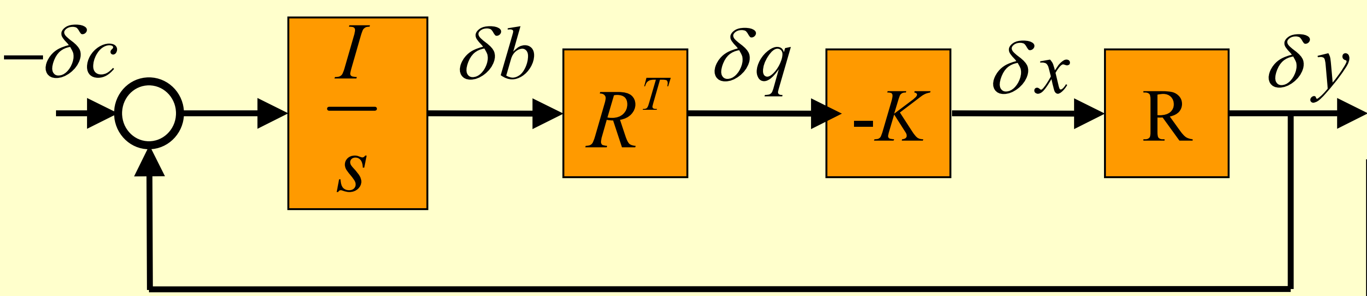
$$\frac{db_l}{dt} = (y_l - c_l)$$

$$q = R^T b$$

Suppose sources receive q_i by feedback, and set $x_i = f_i(q_i)$

Linearized multivariable model, around equilibrium.





$$L(s) = \frac{1}{s} R K R^T$$

Now: $L(s) = \frac{1}{s} RKR^T$ is easily diagonalized.

$$RKR^T = V^T \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_L \end{bmatrix} V, \quad \lambda_l \geq 0 \Rightarrow L(s) = V^T \begin{bmatrix} \frac{\lambda_1}{s} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{\lambda_L}{s} \end{bmatrix} V.$$

- Modes: roots of $\det(I + L(s)) = 0 \;\rightarrow\; s = -\lambda_l, \quad l = 1, \ldots, L.$

Therefore: stable if RKR^T is full rank. Transient response dominated by slowest mode, $\lambda_{\min}(RKR^T)$.

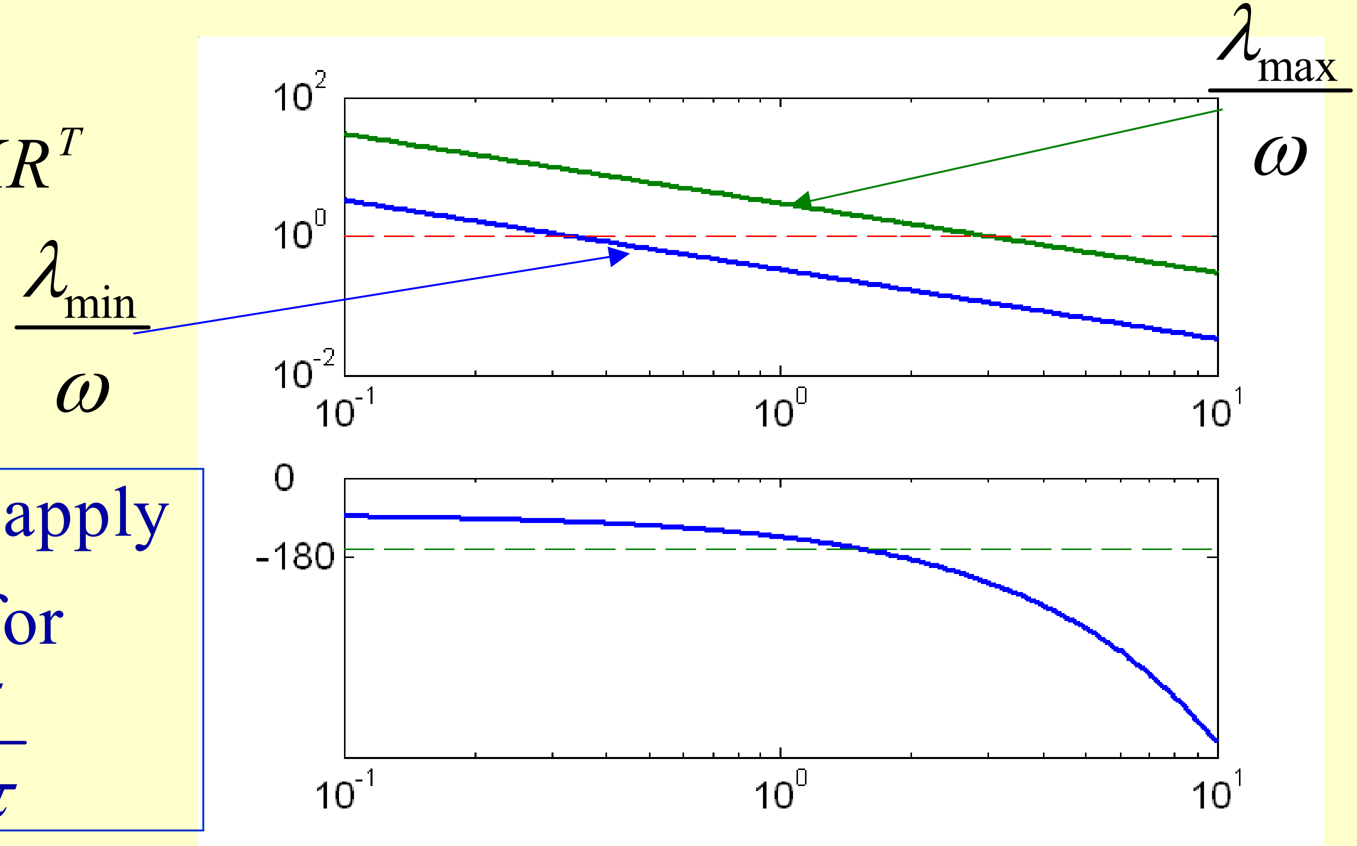
- Singular values of $S(j\omega) = (I + L(j\omega))^{-1}$ are

$$\left| 1 + \frac{\lambda_l}{j\omega} \right|^{-1} = \left| \frac{\omega}{\lambda_l + j\omega} \right| \Rightarrow \bar{\sigma}\left(S(j\omega)\right) = \left| \frac{\omega}{\lambda_{\min} + j\omega} \right|$$

Performance analysis reduces to the scalar case.

Now, consider stability in the presence of delay. For simplicity, use a common delay (RTT) for all loops.

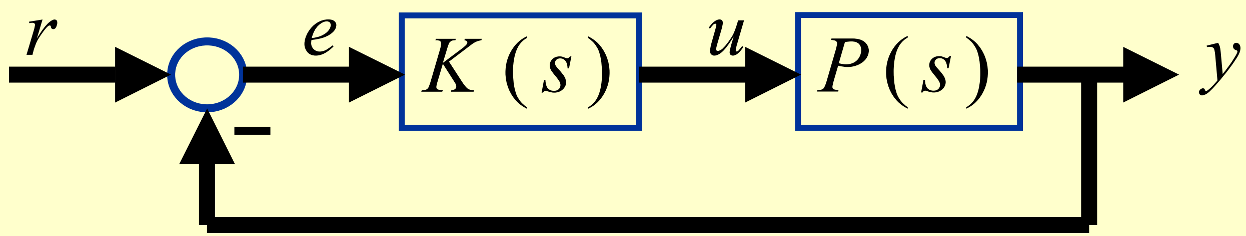
$$L(s) = \frac{e^{-\tau s}}{s} RKR^T$$



Diagonalize and apply Nyquist: Stable for $\lambda_{max}(RKR^T) < \frac{\pi}{2\tau}$

Summary: performance defined by $\lambda_{min}(RKR^T)$, delay robustness by $\lambda_{max}(RKR^T)$. Tradeoff is harder for ill-conditioned RKR^T!

More generally, eigenvalues don't tell the full story.



- Performance: for transfer functions which are not self-adjoint, $\bar{\sigma}(S(j\omega))$ can be much larger-than the maximum eigenvalue.

- Robust stability: consider a ball of plants $P(s) = P_0(s) + \Delta(s)$, $\bar{\sigma}(\Delta(j\omega)) \leq 1/\alpha(\omega)$. Nyquist not very useful to establish stability for all Δ, since $\det(I + KP)$ depends on it in a complicated way.

  However, it can be shown that the condition $\bar{\sigma}(S(j\omega)K(j\omega)\alpha(\omega)) < 1 \quad \forall \omega$ gives robust stability.

Singular values are more important than eigenvalues.

Summary

- A well designed feedback will respond as quickly as possible to regulate, track references or reject disturbances.
- The fundamental limit to the above features is the potential for instability, and its sensitivity to errors in the model. A good design must balance this tradeoff (robust performance).
- In SISO, linear case, tradeoff is well understood by frequency domain methods. This explains their prevalence in design.
- Nonlinear aspects usually handled a posteriori. Nonlinear control can potentially (but not necessarily) do better. A basic test: linearization around any operating point should match up with linear designs.
- In multivariable systems, frequency domain tools extend with some complications (ill conditioning, singular values versus eigenvalues,…)
- All of this is relevant to network flow control: performance vs delay/robustness, ill-conditioning,… Nonlinearity seems mild.