Review:
Feedback & PID Control

Dr. Keith Moffat

Control theory

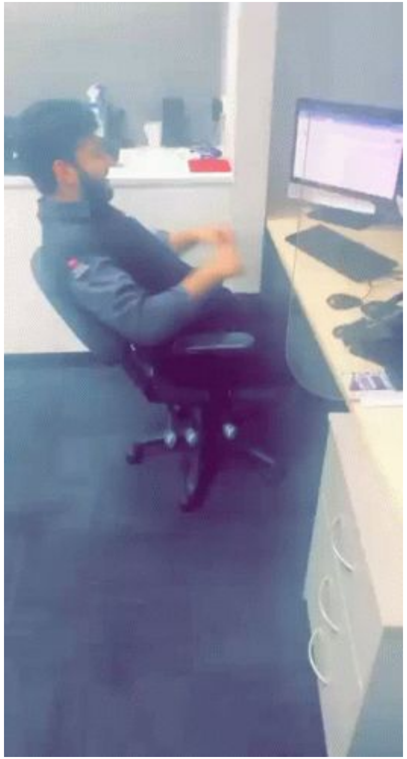
1. Model a system using differential equations

Control theory

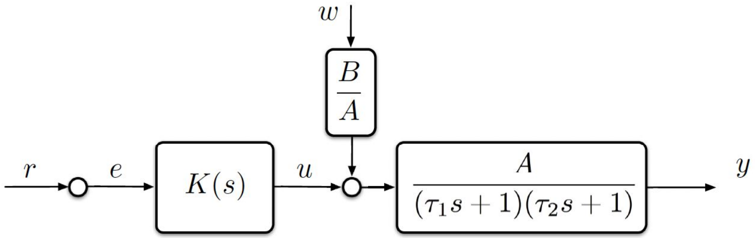
1. Model a system using differential equations
2. Modify the system to do what we want

Control theory

1. Model a system using differential equations
2. Modify the system to do what we want
3. Chill

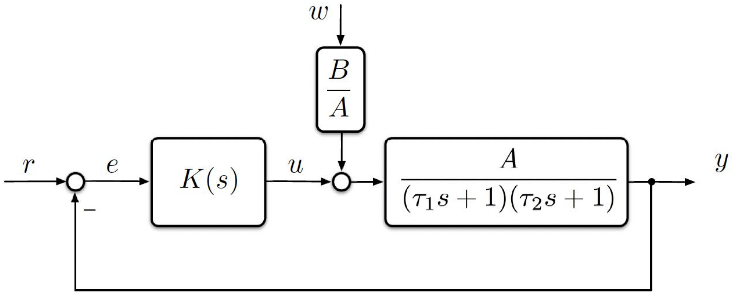


Open loop control



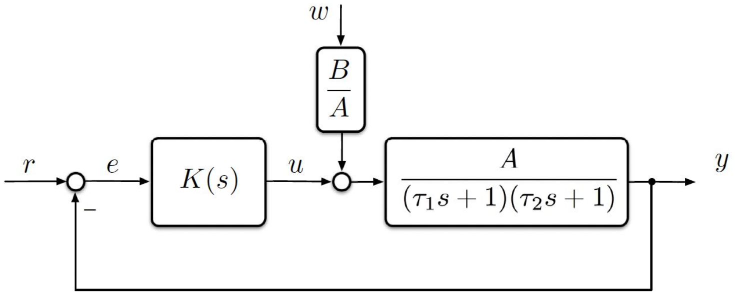
feedforward speed-control system

Closed loop control



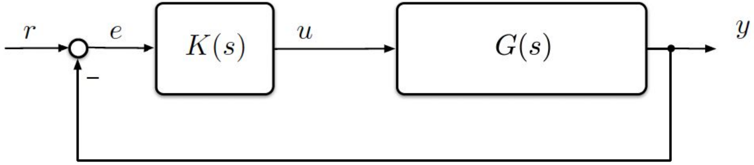
feedback speed-control system

Closed loop control

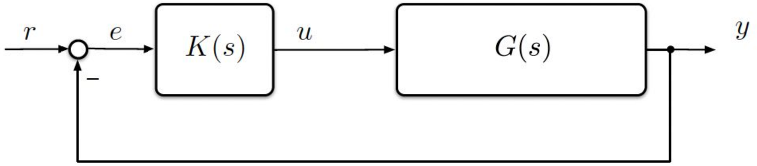


feedback speed-control system

Closed loop/feedback control

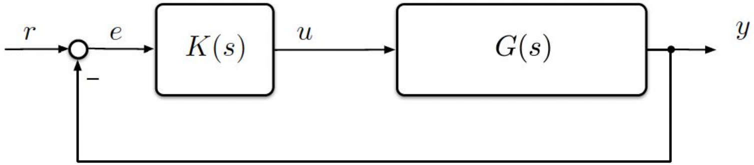


Closed loop/feedback control



Why feedback control?

Closed loop/feedback control



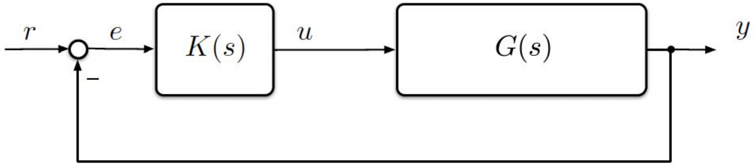
Why feedback control?

1. Reject unknown disturbances

2. Reduce the effect of model uncertainty

3. Alter the system dynamics
   - e.g. stabilize unstable systems

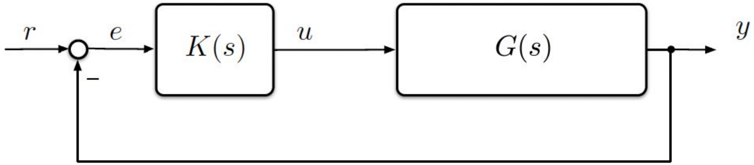
Closed loop/feedback control



A negative feedback block diagram is deceptively simple

$$\frac{y(s)}{r(s)} = \frac{GK}{1+GK}$$

Closed loop/feedback control

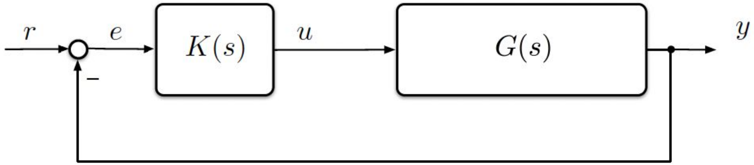


A negative feedback block diagram is deceptively simple

→ Developing intuition takes time and effort

$$\frac{y(s)}{r(s)} = \frac{GK}{1 + GK}$$

Closed loop/feedback control

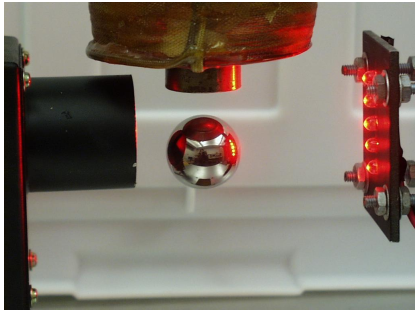


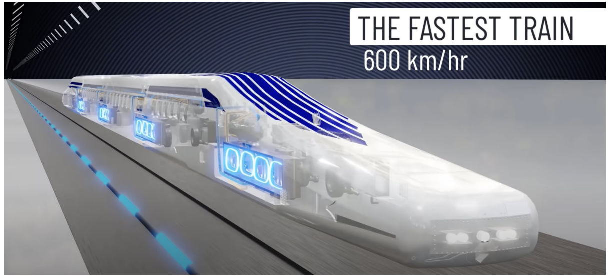
A negative feedback block diagram is deceptively simple

$$\frac{y(s)}{r(s)} = \frac{GK}{1+GK}$$

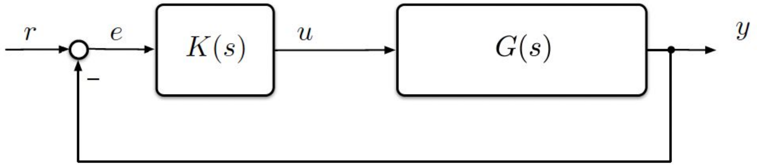
→ Developing intuition takes time and effort

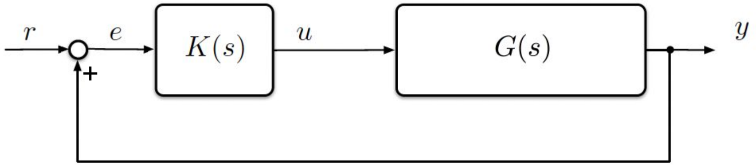
→ Allows us to design controllers that do very cool things





Negative vs. Positive feedback





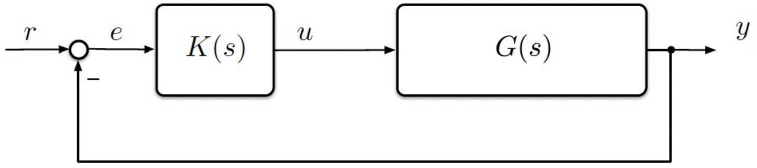
Negative feedback

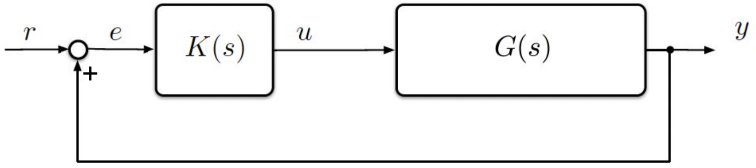
- Usually holds a system to an equilibrium state

Positive feedback

- Usually moves a system from an equilibrium state

Negative vs. Positive feedback





Negative feedback

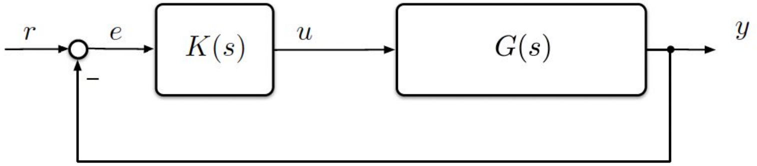
- Usually holds a system to an equilibrium state

Positive feedback

- Usually moves a system from an equilibrium state

Control theory (almost always) uses negative feedback.

Closed loop/feedback control



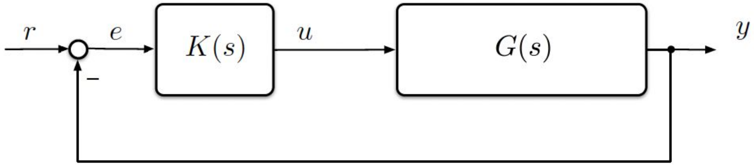
A negative feedback block diagram is deceptively simple

→ Developing intuition takes time and effort

y is less than r and K(s) and G(s) are both “positive” ⇒ Will y get bigger or smaller?

y is greater than r and K(s) and G(s) are both “positive” ⇒ Will y get bigger or smaller?

Closed loop/feedback control



A negative feedback block diagram is deceptively simple

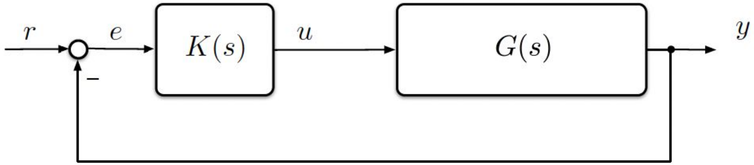
→ Developing intuition takes time and effort

$$\frac{y(s)}{r(s)} = \frac{GK}{1+GK}$$

Feedback changes the differential equations that define the system

→ Transfer functions make the design process easier

Closed loop/feedback control



A negative feedback block diagram is deceptively simple

→ Developing intuition takes time and effort

$$\frac{y(s)}{r(s)} = \frac{GK}{1 + GK}$$

Feedback changes the differential equations that define the system

→ Transfer functions make the design process easier

As a matter of idle curiosity, I once counted to find out what the order of the set of equations in an amplifier I had just designed would have been, if I had worked with the differential equations directly. It turned out to be 55.

- Henrik Bode, 1960

Transfer function poles

$$G(s)K(s) = k \cdot \frac{z(s)}{p(s)} = k \cdot \frac{\Pi_{i=1}^{n_z}(s - z_i)}{\Pi_{i=1}^{n_p}(s - p_i)}$$

What do transfer function poles represent?

Transfer function poles

$$G(s)K(s) = k \cdot \frac{z(s)}{p(s)} = k \cdot \frac{\Pi_{i=1}^{nz}(s-z_i)}{\Pi_{i=1}^{np}(s-p_i)}$$

Poles describe the “modes” of the system—They tell us if the system is stable or not.

Partial fraction decomposition of G(s)K(s): $\frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \ldots + \frac{r_n}{s-p_n}$

Thus, the response to an impulse is: $y(t) = r_1 e^{p_1 t} + r_2 e^{p_2 t} + \ldots r_n e^{p_n t}$

Transfer function zeros

$$G(s)K(s) = k \cdot \frac{z(s)}{p(s)} = k \cdot \frac{\Pi_{i=1}^{n_z}(s - z_i)}{\Pi_{i=1}^{n_p}(s - p_i)}$$

What do transfer function zeros represent?

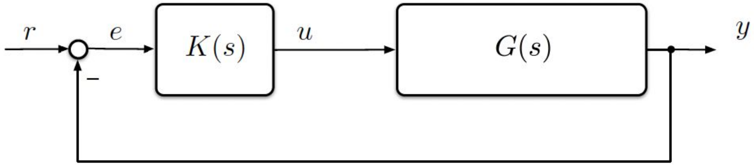
Transfer function zeros

$$G(s)K(s) = k \cdot \frac{z(s)}{p(s)} = k \cdot \frac{\Pi_{i=1}^{nz}(s-z_i)}{\Pi_{i=1}^{np}(s-p_i)}$$

Zeros are harder to interpret. 3 interpretations are:

1. They factor into the residue (r terms) of $\frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \ldots + \frac{r_n}{s-p_n}$
   - Thus, they factor into the weights of the impulse response $y(t) = r_1 e^{p_1 t} + r_2 e^{p_2 t} + \ldots r_n e^{p_n t}$
2. They are the “blind spots” of the transfer function
   - Inputs that match the zero do not affect the output
3. In the time domain, they describe derivative action on the input

Open and Closed Loop

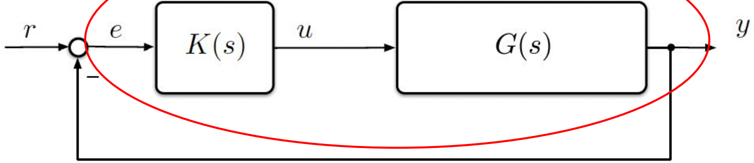


$$\frac{y(s)}{r(s)} = G(s)\,K(s)$$

$$\frac{y(s)}{r(s)} = \frac{GK}{1 + G(s)\,K(s)}$$

Open and Closed Loop

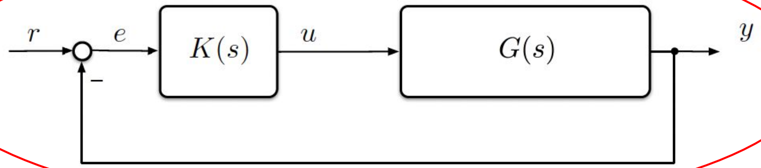
Open Loop (OL)



$$\frac{y(s)}{r(s)} = G(s)\,K(s)$$

$$\frac{y(s)}{r(s)} = \frac{G(s)\,K(s)}{1 + G(s)\,K(s)}$$

Open and Closed Loop

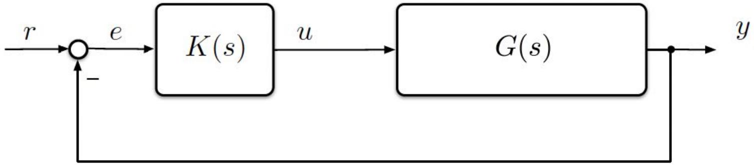


Closed Loop (CL)

$$\frac{y(s)}{r(s)} = G(s)\,K(s)$$

$$\frac{y(s)}{r(s)} = \frac{G(s)\,K(s)}{1 + G(s)\,K(s)}$$

Closed loop poles

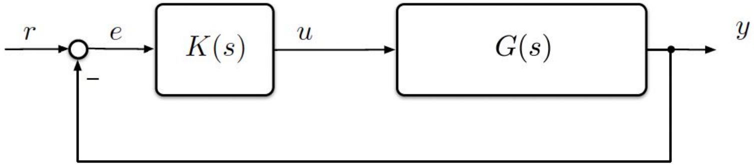


We care about the poles of the closed loop transfer function

(not so much about the zeros)

$$\frac{y(s)}{r(s)} = \frac{G(s)\,K(s)}{1 + G(s)\,K(s)}$$

Closed loop poles



We care about the poles of the closed loop transfer function

(not so much about the zeros)

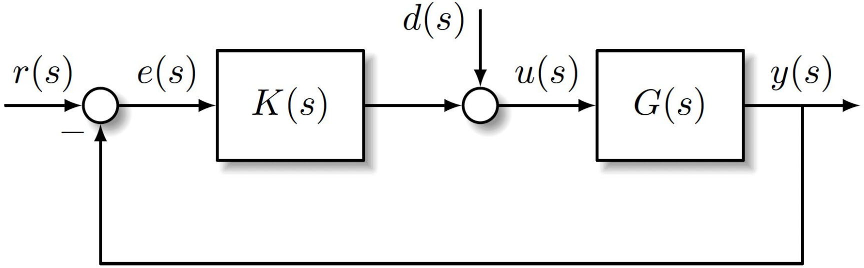
$$\frac{y(s)}{r(s)} = \frac{G(s)\,K(s)}{1+G(s)\,K(s)}$$

But..

Both the poles and the zeros of the open loop TF G(s)K(s) affect the closed loop TF poles

→ We care about both the poles and the zeros of the open loop transfer function

Feedback stability

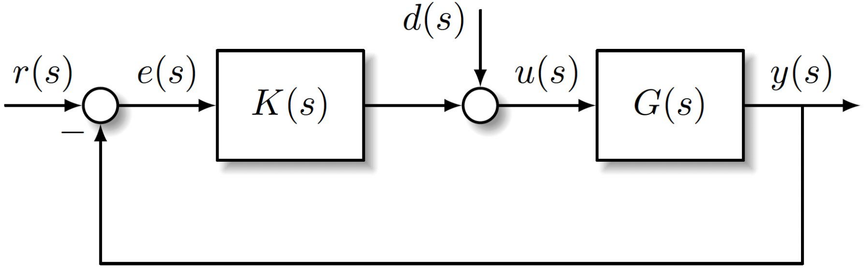


All 6 input/output pairs must be stable:

$$\frac{e(s)}{r(s)} = \frac{1}{1+GK} \qquad \frac{u(s)}{r(s)} = \frac{K}{1+GK} \qquad \frac{y(s)}{r(s)} = \frac{GK}{1+GK}$$

$$\frac{e(s)}{d(s)} = \frac{-G}{1+GK} \qquad \frac{u(s)}{d(s)} = \frac{1}{1+GK} \qquad \frac{y(s)}{d(s)} = \frac{G}{1+GK}$$

Feedback stability



All 6 input/output pairs must be stable:

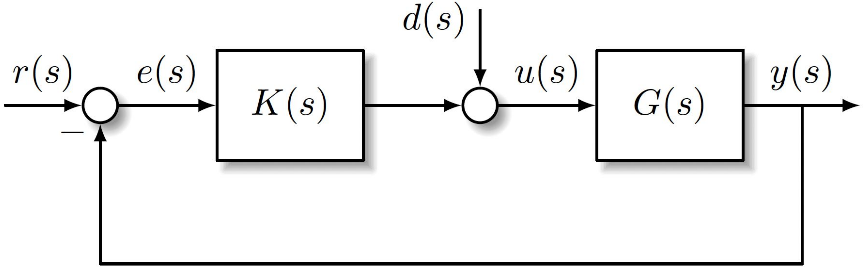
$$\frac{e(s)}{r(s)} = \frac{1}{1+GK} \qquad \frac{u(s)}{r(s)} = \frac{K}{1+GK} \qquad \frac{y(s)}{r(s)} = \frac{GK}{1+GK}$$

$$\frac{e(s)}{d(s)} = \frac{-G}{1+GK} \qquad \frac{u(s)}{d(s)} = \frac{1}{1+GK} \qquad \frac{y(s)}{d(s)} = \frac{G}{1+GK}$$

Note: All six TFs do not have the same denominator because the numerator could be a fraction

→ How do you determine the stability of all six TFs?

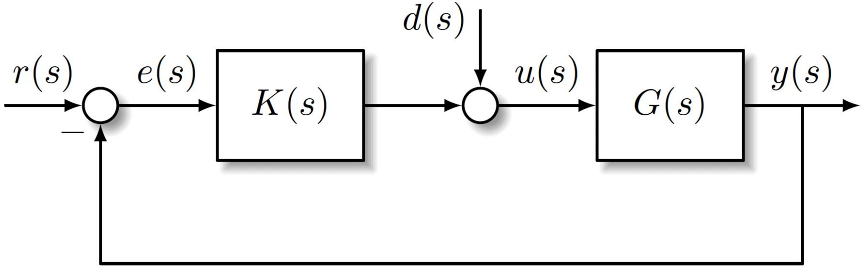
Feedback stability



feedback stable $\Leftrightarrow$ *all* transfer functions are BIBO stable

$\Leftrightarrow$ roots of $0 = 1 + G(s)\,K(s)$ in $\mathbb{C}_-$ & no unstable pole/zero cancellation

Feedback stability

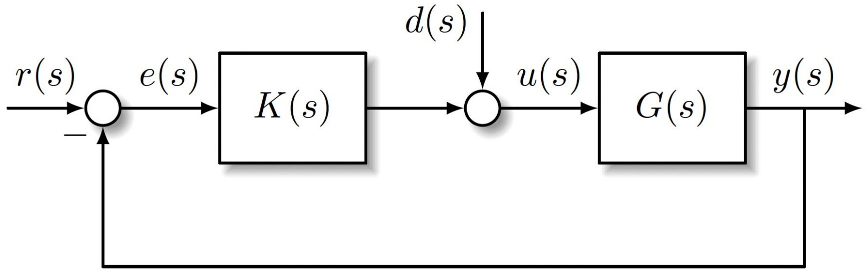


feedback stable $\Leftrightarrow$ *all* transfer functions are BIBO stable

$\Leftrightarrow$ roots of $0 = 1 + G(s)\,K(s)$ in $\mathbb{C}_-$ & no unstable pole/zero cancellation

What if G(s)K(s) *does* have unstable pole/zero pairs?

Feedback stability



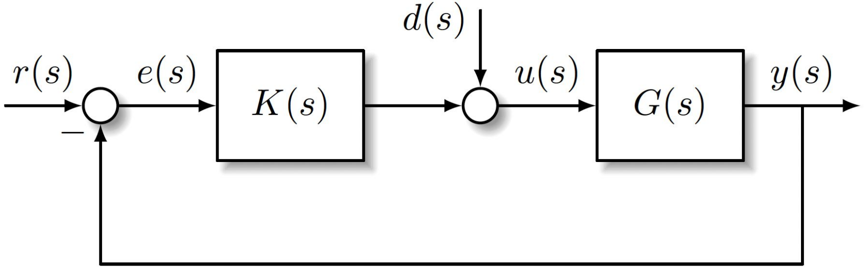
feedback stable $\Leftrightarrow$ *all* transfer functions are BIBO stable

$\Leftrightarrow$ roots of $0 = 1 + G(s)\,K(s)$ in $\mathbb{C}_-$ & no unstable pole/zero cancellation

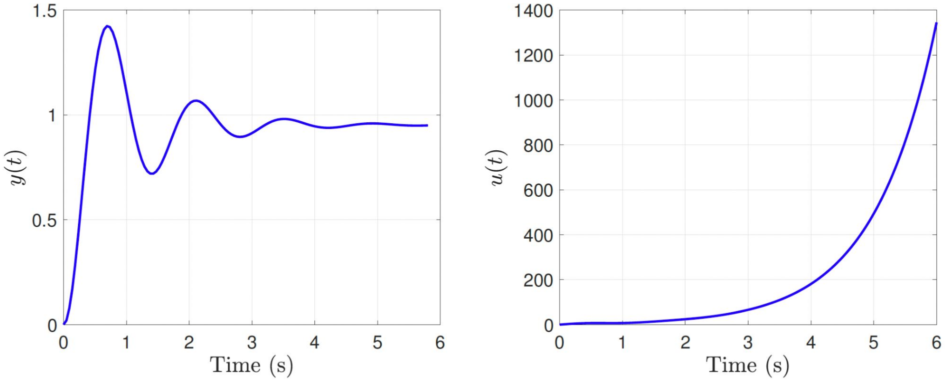
What if G(s)K(s) *does* have unstable pole/zero pairs?

→ *Don't cancel them!*

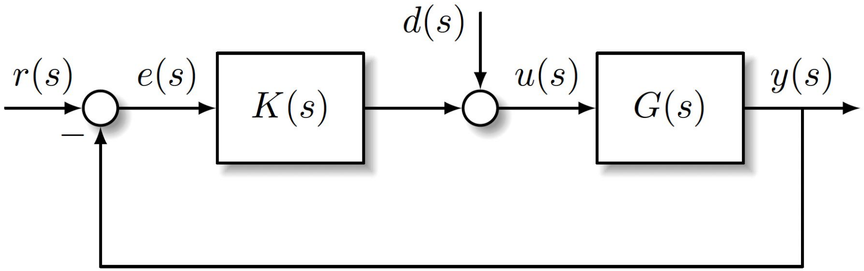
Feedback stability



Roots of 1 + GK in LHP only after unstable pole/zero cancelation ⇒ One of the six TFs will be unstable



Feedback stability

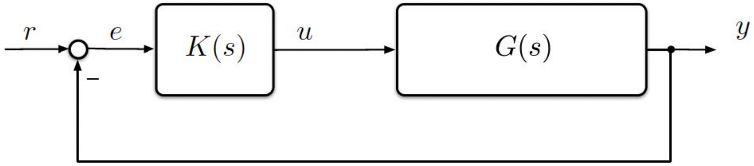


The form on the right (below) does not allow you to make pole-zero cancellations:

$$\frac{Y(s)}{R(s)} = \frac{G(s)K(s)}{1 + G(s)K(s)} = \frac{N_{\mathrm{g}}N_{\mathrm{k}}}{N_{\mathrm{g}}N_{\mathrm{k}} + D_{\mathrm{g}}D_{\mathrm{k}}}$$

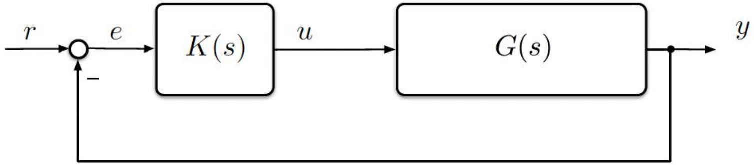
…so if you use this form you don’t have to worry about unstable pole/zero cancellations.

Controller design



Given G(s), how do you design K(s)?

Classical control

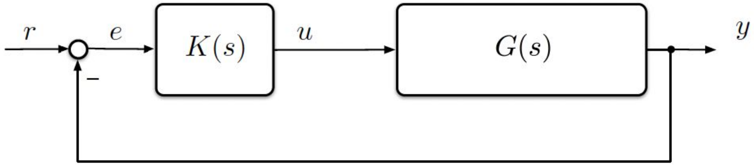


Known: G(s)

Desired: Good closed loop behavior

Approach: Choose K(s) with as few calculations as possible

Classical control



Known: G(s)

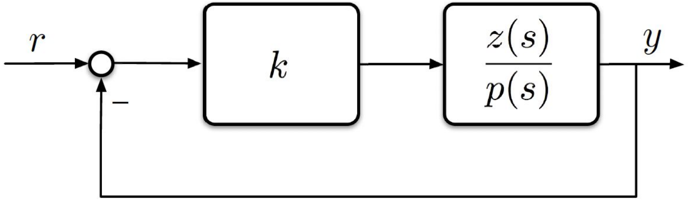
Desired: Good closed loop behavior

Approach: Choose K(s) with as few calculations as possible

Note:

- Many of these methods were developed before computers
- With computers, they are still useful because they provide intuition

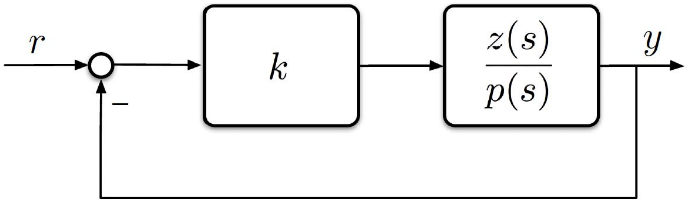
Root Locus



Gives the roots of the CL transfer function as a function of the gain k

$$\frac{y(s)}{r(s)} = \frac{G(s)\,K(s)}{1+G(s)\,K(s)}$$

Root Locus



Gives the roots of the CL transfer function as a function of the gain k

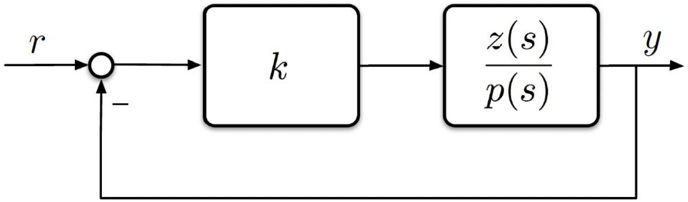
$$\frac{y(s)}{r(s)} = \frac{G(s)\,K(s)}{1 + G(s)\,K(s)}$$

→ Uses the poles and zeros of the open loop transfer function

$$G(s)K(s) = k \cdot \frac{z(s)}{p(s)} = k \cdot \frac{\Pi_{i=1}^{nz}(s - z_i)}{\Pi_{i=1}^{np}(s - p_i)}$$

$$p(s) + k \cdot z(s) = 0$$

Root Locus



Gives the roots of the CL transfer function as a function of the gain k

$$\frac{y(s)}{r(s)} = \frac{G(s)\,K(s)}{1 + G(s)\,K(s)}$$

→ Uses the poles and zeros of the open loop transfer function

$$G(s)K(s) = k \cdot \frac{z(s)}{p(s)} = k \cdot \frac{\Pi_{i=1}^{nz}(s - z_i)}{\Pi_{i=1}^{np}(s - p_i)}$$

$$p(s) + k \cdot z(s) = 0$$

What happens when k is large?

What happens when k is small?

PID control

$$K(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right) \quad \Longrightarrow \quad u(t) = K\left(e + \frac{1}{T_I}\int_{t_0}^{t} e(\tau)d\tau + T_D \dot{e}\right)$$

PID control

$$K(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right) \implies u(t) = K\left(e + \frac{1}{T_I}\int_{t_0}^{t} e(\tau)d\tau + T_D \dot{e}\right)$$

The vast majority of controllers “in the wild” are PID controllers.

This is because PID are

- Intuitive to understand
- Can be tuned with heuristics/intuition (sometimes)
- Work well with second order differential equations

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

PID tuning options

- Guess-and-check
- Decide Ti and Td then using Root Locus
- Heuristics: Zeigler-Nichols, Åström and Hagglund, others
- Pole placement (this is not really a PID method, but it could be used)

Pole placement

Specify the desired closed loop poles/roots of $\Pi(s)$

→ calculate on coefficients K(s) to achieve those roots

$$\Pi(s) = N_{\mathrm{g}}(s)N_{\mathrm{k}}(s) + D_{\mathrm{g}}(s)D_{\mathrm{k}}(s)$$

Pole placement

Specify the desired closed loop poles/roots of $\Pi(s)$

→ calculate on coefficients K(s) to achieve those roots

$$\Pi(s) = N_{\mathrm{g}}(s)N_{\mathrm{k}}(s) + D_{\mathrm{g}}(s)D_{\mathrm{k}}(s)$$

1. Determine the order of the controller:

$$K(s) = \frac{d_{n-1}s^{n-1} + \cdots + d_1 s + d_0}{c_{n-1}s^{n-1} + \cdots c_1 s + c_0}$$

2. Substitute into the yet-undetermined controller into $\Pi(s)$

3. Set $\Pi(s) = \Pi_{\mathrm{des}}(s)$ and solve the system of linear equations (one equation for each polynomial coefficient) to get the controller K(s)

Pole placement

Specify the desired closed loop poles/roots of $\Pi(s)$

→ calculate on coefficients K(s) to achieve those roots

$$\Pi(s) = N_{\mathrm{g}}(s)N_{\mathrm{k}}(s) + D_{\mathrm{g}}(s)D_{\mathrm{k}}(s)$$

1. Determine the order of the controller:

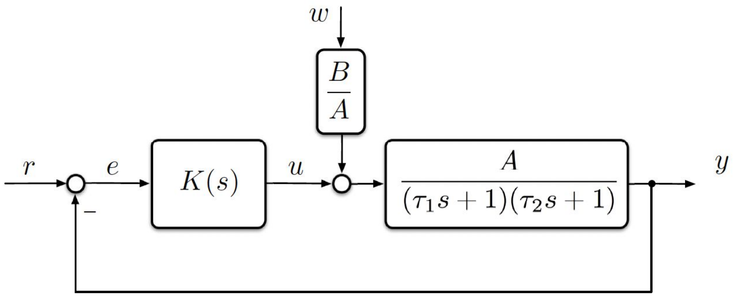
$$K(s) = \frac{d_{n-1}s^{n-1} + \cdots + d_1 s + d_0}{c_{n-1}s^{n-1} + \cdots c_1 s + c_0}$$

2. Substitute into the yet-undetermined controller into $\Pi(s)$

3. Set $\Pi(s) = \Pi_{\mathrm{des}}(s)$ and solve the system of linear equations (one equation for each polynomial coefficient) to get the controller K(s)

…easy, right?

Closed loop control



feedback speed-control system