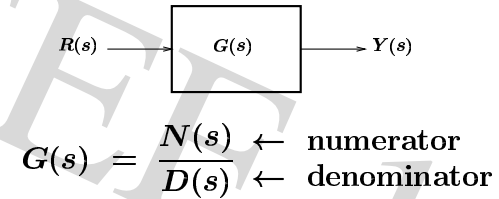


Today's EEE 101 Lecture

- Some more about transfer functions.
- General control system, definitions and objectives.
- LTI response to sinusoidal functions.
- Familiarization with Matlab.

Transfer Functions

- $Y(s) = G(s)R(s)$



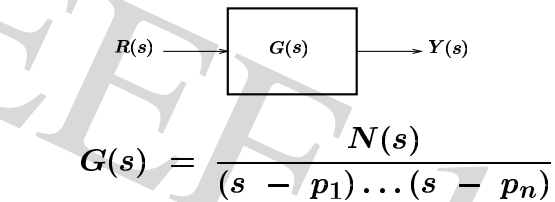
- $N(s)$ polynomial in s of order m .
 $D(s)$ polynomial in s of order n .

Transfer Functions

- Zeros of the TF : roots of $N(s)$, i.e.,
 $s = \{z_1, z_2, \dots, z_m\}$ such that $N(s) = 0$.
- Poles of the TF : roots of $D(s)$, i.e.,
 $s = \{p_1, p_2, \dots, p_n\}$ such that $D(s) = 0$.
- Zeros and poles affect the open-loop stability as well as the stability of a system.

Transfer Functions

- Open-loop system.



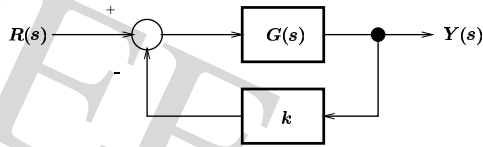
response.

$$y(t) = k_1 e^{p_1 t} + \dots + k_n e^{p_n t}$$

⇒ "poles affect the system stability."

Transfer Functions

- Closed-loop system.

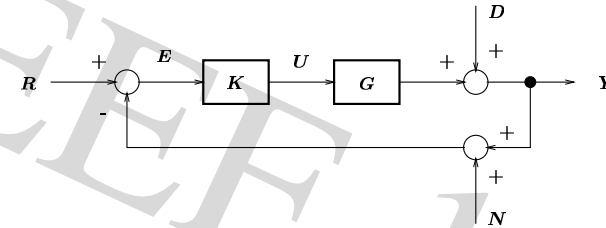


$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + kG(s)} = \frac{N(s)}{D(s) + kN(s)}$$

⇒ "poles and affect the system"

General Control System

- Control system.



R : reference

D : disturbance.

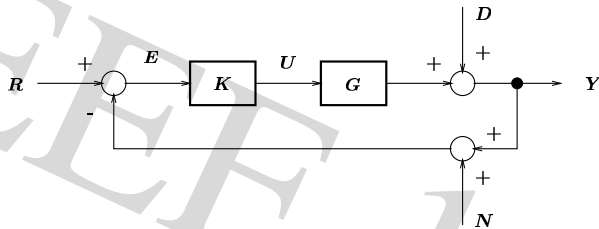
– known / unknown

– random / deterministic

N : sensor or noise.

General Control System

- Y : output.



using $Y = KGE + D$ and $E = R - (Y + N)$,

$$Y = \frac{KG}{1 + KG}R + \frac{1}{1 + KG}D - \frac{KG}{1 + KG}N$$

General Control System

- E : tracking

using $E = R - (Y + N)$ and $Y = KGE + D$,

$$E = \frac{1}{1 + KG}R - \frac{1}{1 + KG}D - \frac{1}{1 + KG}N$$

- U : actuator input.

since $U = KE$,

$$U = \frac{K}{1 + KG}(R - D - N)$$

Control System Objectives

- Small error E .
- Small input U .
- Small effect of disturbance D on output Y and error E .
- Small effect of noise N on Y and E .

Control System Objectives

- System variables.
 - output : $Y = SD + T(R - N)$
 - error : $E = S(R - D - N)$
 - input : $U = KS(R - D - N)$
- Disturbance : reducing the effect of D .
→ needs small S .

Definitions

- Return ratio J .
 $J = 1 + KG$
- Sensitivity S .
 $S = \frac{1}{1 + KG} = \frac{1}{J}$
- Complementary sensitivity T .
 $T = 1 - S = \frac{KG}{1 + KG}$

Control System Objectives

- Good tracking : small E .
→ needs small S .
- Noise immunity : reducing the effect of N .
→ needs small T , large S .
- Small actuator signals : small U .
→ needs small $KS \approx \frac{1}{G}$ for large K .

Typical Control System Design

- Large loop gain at low frequencies.
→ for tracking and rejection.
- Low loop gain at middle frequencies.
→ for .
- Low loop gain at high frequencies.
→ for noise .

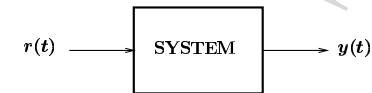
System Response

- System type and error constants.
Can we classify a system in terms of its response to standard reference inputs (e.g. step or ramp inputs)?

- Decompose system response.

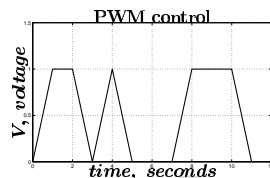
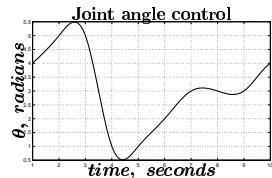
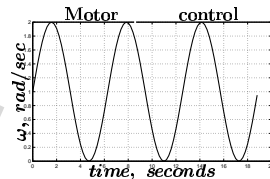
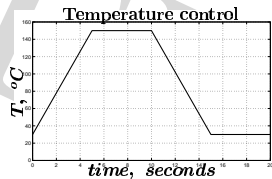
$$y(t) = y_{transient}(t) + y_{steady-state}(t)$$

where $\lim_{t \rightarrow \infty} y_{transient}(t) = 0$.



System Response

- Examples of reference functions.



Standard Reference Inputs

- Why use standard reference inputs?
– take advantage of linearity and superposition.
– simple Laplace transforms.

- Step input (e.g. setpoints).

$$u(t) \Leftrightarrow \frac{1}{s}$$

- Ramp input (e.g. gradual temp or position changes).

$$tu(t) \Leftrightarrow \frac{1}{s^2}$$

Standard Reference Inputs

- Parabolic input (e.g. splines and robot trajectories).

$$\frac{t^2}{2}tu(t) \Leftrightarrow \frac{1}{s^3}$$

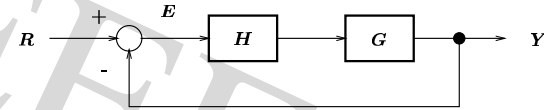
- Sinusoidal input (rotating machines).

$$\cos\omega t \Leftrightarrow \frac{s}{s^2 + \omega^2}$$

$$\sin\omega t \Leftrightarrow \frac{\omega}{s^2 + \omega^2}$$

Error Response

- Consider the gain feedback system.



$$E = \frac{1}{1 + HG}R$$

- Steady-state

$$e_{ss} = \lim_{t \rightarrow \infty} e(t)$$

Error Response

- Using the value theorem.

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} s \frac{R(s)}{1 + HG(s)}$$

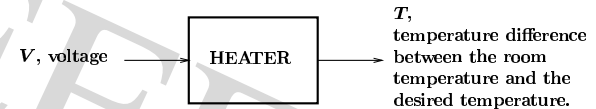
- System type.

$$HG(s) = k \frac{(s - z_1) \dots (s - z_m)}{s^j (s - p_1) \dots (s - p_n)}$$

definition : system $\triangleq j$.

System Type

- Why consider system type?



simple model (first-order, type 0).

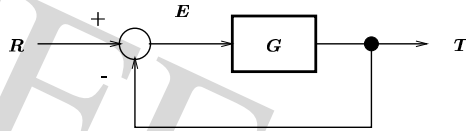
$$G(s) = \frac{1}{s + 1}$$

- Step response.

>> step(1, [1 1])

System Type

- Now add feedback.

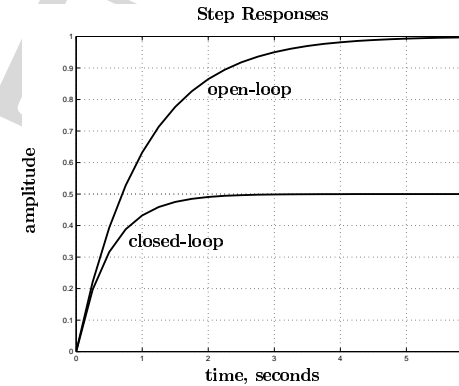


- Step response of the system.

```
>> [n,d] = cloop(1, [1 1], -1)
>> step(n, d)
```

System Type

- Simulation results for the open-loop and closed-loop systems.



System Type

- Steady-state error.

$$\frac{E}{R} = \frac{R - T}{R} = \frac{1}{1 + G} = \frac{s + 1}{s + 2}$$

- For a step input: $r(t) = u(t)$, then

$$e_{ss} = \lim_{s \rightarrow 0} s \frac{s + 1}{s + 2} = \frac{1}{2}$$

- For a type 0 system.
⇒ non-zero tracking error in to a step input.

Familiarization with Matlab

- Polynomial in s .

Example. Consider $a_1s^2 + a_2s + a_3$.

```
>> p = [a_1 a_2 a_3]
```

- Fractional polynomial in s .

Example.

$$\frac{a_1s^2 + a_2s + a_3}{b_1s^2 + b_2s + b_3} \Leftarrow \begin{matrix} \text{num} \\ \text{den} \end{matrix}$$

```
>> num = [a_1 a_2 a_3]
```

```
>> den = [b_1 b_2 b_3]
```

Familiarization with Matlab

- **Simulating response.**

Example. Determine the step response of

$$G(s) = \frac{10}{s^2 + 2s + 10}$$

```
>> num = 10; den = [1 2 10];
>> t = [0:0.1:10]';
>> step(num, den, t);
```

- **Determine the ramp response of $G(s)$.**

```
>> ramp = t;
>> lsim(num, den, ramp, t);
```

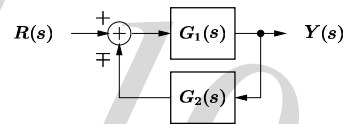
Familiarization with Matlab

- **Feedback**

```
>> [nf, df] = feedback(n1, d1, n2, d2, sign)
```

$$\text{sign} = \begin{cases} +1, & \text{positive feedback} \\ -1, & \text{negative feedback} \end{cases}$$

$$G_p(s) = \frac{G_1(s)}{1 \pm G_1(s)G_2(s)}$$

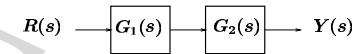


Familiarization with Matlab

- **Series**

```
>> [ns, ds] = series(n1, d1, n2, d2)
```

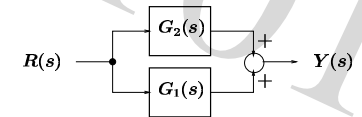
$$G_s(s) = G_1(s)G_2(s)$$



- **Parallel**

```
>> [np, dp] = parallel(n1, d1, n2, d2)
```

$$G_p(s) = G_1(s) + G_2(s)$$



Familiarization with Matlab

- **Exercise.** Compute the step response of $G(s)$ with unity gain positive feedback. Hint.

```
>> [nf, df] = feedback(num, den, 1, 1, +1)
```

- **Exercise.** Compute the step response of $G(s)$ with unity gain negative feedback.

- **Methods of determining response.**

- lsim (Matlab)
- lode (octave)

Summary

- Transfer functions, and poles and zeros.
- What do you want a control system to do?
- System type for closed-loop systems.
- Some more interesting facts and step response.