Automatic Control

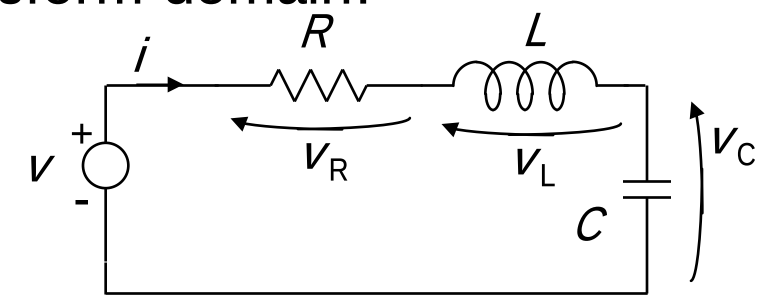
Transfer function of LTI dynamical systems

Representations of LTI dynamical systems

Transfer function of LTI SISO continuous time systems

Transfer function of the RLC circuit

In the presence of <u>zero initial conditions</u>, the solution of an LTI dynamical system can be obtained more directly by computing the relationship between the system input and output in the Laplace transform domain:



$$u(t) = v(t),\ y(t) = v_C(t)$$
$$i(0) = 0,\ v_C(0) = 0$$

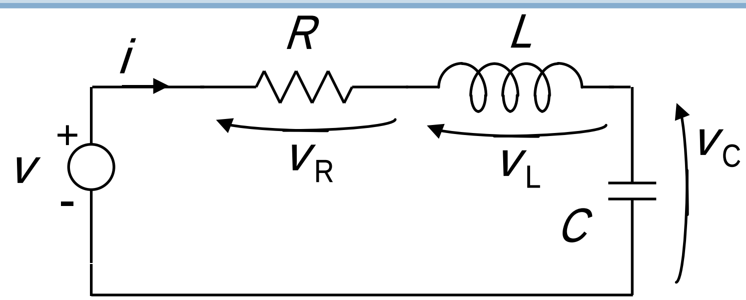
$$v(t) = v_R(t) + v_L(t) + v_C(t) =$$
$$= R\,i(t) + L\,di(t)/dt + v_C(t)$$
$$i(t) \quad = C\ dv_C(t)/dt$$

$$\rightarrow \quad V(s) = R\,I(s) + sL\,I(s) + V_C(s)$$
$$I(s) \quad = sC\,V_C(s)$$

$$V(s) = R\,I(s) + sL\,I(s) + V_C(s) \underset{\substack{\uparrow \\ I(s) = sC\,V_c(s)}}{=} sRC\,V_C(s) + s^2LC\,V_C(s) + V_C(s)$$

$$\rightarrow V(s) = \left(s^2LC + sRC + 1\right)V_C(s)$$

Transfer function of the RLC circuit



$$u(t) = v(t),\ y(t) = v_C(t)$$

$$i(0) = 0,\ v_C(0) = 0$$

$$V(s) = \left(s^2LC + sRC + 1\right)V_C(s)$$

$$\rightarrow V_C(s) = \underbrace{\frac{1}{s^2LC + sRC + 1}}_{H(s)}V(s)$$

$$\rightarrow H(s) = \frac{V_C(s)}{V(s)} = \frac{Y(s)}{U(s)} = \frac{1}{s^2LC + sRC + 1}$$

$H(s) \rightarrow$ **transfer function**

The **transfer function** *H(s)* of a single input single output (SISO) LTI dynamical system is the ratio between the system output and input Laplace transforms

$$\rightarrow H(s) = \frac{Y(s)}{U(s)}$$

The system transfer function

The **transfer function** *H(s)* is referred to as the

input-output representation

of a single input single output (SISO) LTI dynamical system

$$H(s) = \left.\frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}\right\} \rightarrow \text{input - output representation}$$

Note that, without loss of generality, the leading coefficient of the denominator of *H(s)* is assumed to be unitary

Transfer fuction as impulse response of LTI systems

Consider the zero state output response of an LTI dynamical system in the presence of a Dirac's delta impulse input (i.e. u(t) = δ(t) → U(s) = 1)

We have:

$$Y(s) = H(s)U(s) \underset{\substack{\uparrow \\ U(s)=1}}{=} H(s)\cdot 1 = H(s)$$

Therefore, the transfer function can be viewed also as the Laplace transform of the zero state output impulse response h(t)

$$\downarrow$$

$$H(s) = \mathcal{L}(h(t))$$

Transfer function

The transfer function of an LTI (SISO) dynamical system is a real rational function (i.e. the ratio of two polynomials) of the complex variable s :

$$H(s) = \frac{N_H(s)}{D_H(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}, \qquad m \le n$$

- $m < n$ → **strictly proper**
- $m = n$ → **proper**
- Roots of $N_H(s)$ → **system zeros**
- Roots of $D_H(s)$ → **system poles**

Examples: $H(s) = \dfrac{s+5}{s^2+3s+2}, H(s) = \dfrac{s+1}{s+2}$

- Definition of transfer functions with MatLab

$$H(s) = \frac{1}{s^2 + 3s + 2}$$

- Define the Laplace variabile **`s`** using **`tf`** statement

```
>> s=tf('s')
Transfer function:
s
```

- Define

```
>> H=1/(s^2+3*s+2)
 Transfer function:
      1
-------------
s^2 + 3 s + 2
```

Poles and zeros

- Computation of zeros and poles of a transfer function using MatLab

$$H(s) = \frac{s+5}{s^2+3s+2}$$

- Use the statements **zero** and **pole**

```
>> s=tf('s');
>> H=(s+5)/(s^2+3*s+2);
>> zeros_H = zero(H)
zeros_H =
    -5
>> poles_H = pole(H)
poles_H =
    -2
    -1
```

The system transfer function: zero-pole-gain form

$$H(s) = K_\infty \frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

- $z_1, \ldots, z_m$ → Zeros of $H(s)$
- $p_1, \ldots, p_n$ → Poles of $H(s)$
- K_∞ → gain

$$K_\infty = \lim_{s\to\infty} s^{n-m} H(s)$$

Example: $H(s) = \dfrac{s+5}{s^2+3s+2} = 1 \cdot \dfrac{s+5}{(s+1)(s+2)}$

Zero-pole-gain form

- Computation of the zero-poles-gain form of a transfer function using MatLab

$$H(s) = \frac{4(2s+6)}{s^2+3s+2}$$

- Use the statement **zpk**

```
>> s=tf('s');

>> H=4*(2*s+6)/(s^2+3*s+2);

>> zpk(H)

Zero/pole/gain:

 8 (s+3)

-----------

(s+2) (s+1)
```

The system transfer function: dc-gain form

$$H(s) = K\,\frac{(1 - s/z_1)(1 - s/z_2)\cdots(1 - s/z_m)}{s^r\,(1 - s/p_1)(1 - s/p_2)\cdots(1 - s/p_{n-r})}$$

- $z_1, \ldots, z_m$ → zeros of *H(s)*
- r → poles of *H(s)* at the origin
- $p_1, \ldots, p_{n-r}$ → poles of *H(s)*
- K → generalized static gain (dc-gain) $\rightarrow K = \lim_{s\to 0} s^r H(s)$

Example: $H(s) = \dfrac{s+5}{s^2+3s+2} = \dfrac{5(1+s/5)}{1\cdot(1+s)\cdot 2\cdot(1+s/2)} = \dfrac{5}{2}\,\dfrac{1+s/5}{(1+s)(1+s/2)}$

No specific MatLab statement

Computation example 3

Given the following transfer function of an LTI dynamical system:

$$H(s) = \frac{2s+1}{(s+4)^2}$$

compute the output response $y(t)$ when $u(t) = 2\ t\,\varepsilon(t)$ (linear ramp) The solution in Laplace domain can be computed as

$$Y(s) = H(s)U(s)$$

with

$$U(s) = \frac{2}{s^2}$$

Computation example 3

$$Y(s) = H(s)U(s) = \underbrace{\frac{2s+1}{(s+4)^2}}_{H(s)} \underbrace{\frac{2}{s^2}}_{U(s)} = \frac{2(2s+1)}{s^2(s+4)^2} =$$

$$= \frac{R_{1,1}}{s} + \frac{R_{1,2}}{s^2} + \frac{R_{2,1}}{s+4} + \frac{R_{2,2}}{(s+4)^2} =$$

$$= \frac{0.1875}{s} + \frac{0.125}{s^2} - \frac{0.1875}{s+4} - \frac{0.875}{(s+4)^2}$$

Computation example 3

$$Y(s) = \frac{0.1875}{s} + \frac{0.125}{s^2} - \frac{0.1875}{s+4} - \frac{0.875}{(s+4)^2}$$

$$Re^{at}\varepsilon(t) = \mathcal{L}^{-1}\left\{\frac{R}{s-a}\right\}, \quad Rte^{at}\varepsilon(t) = \mathcal{L}^{-1}\left\{\frac{R}{(s-a)^2}\right\}$$

$$y(t) = \left(0.1875 + 0.125t - 0.1875e^{-4t} - 0.875te^{-4t}\right)\varepsilon(t)$$

Computation example 3: MatLab procedure

- Define the Laplace variabile **`s`** using **`tf`** statement

```
>> s=tf('s')
Transfer function:
s
```

- Define the system input

```
>> U=2/s^2
Transfer function:
 2
---
s^2
```

Computation example 3: MatLab procedure

- Introduce the system transfer fuction

```
>> H=(2*s+1)/(s+4)^2


Transfer function:

   2 s + 1

--------------

s^2 + 8 s + 16
```

Computation example 3: MatLab procedure

- Compute $Y(s) = H(s)U(s)$

  use statements **`minreal`** and **`zpk`**, in order to simplify and highlights denominator roots respectively

```
>> Y=zpk(minreal(H*U))


Zero/pole/gain:

 4 (s+0.5)

-----------

s^2 (s+4)^2
```

Computation example 3: MatLab procedure

- For Y(s) , compute the PFE using the statements **`tfdata`** and **`residue`**

```
>> [num_Y,den_Y]=tfdata(Y,'v')
num_Y =
         0         0         0    4.0000    2.0000
den_Y =
    1.0000    8.0000   16.0000         0         0
>> [r,p]=residue(num_Y, den_Y)
r =-0.1875
   -0.8750
    0.1875
    0.1250
p =-4.0000
   -4.0000
         0
         0
```

$$\rightarrow Y(s) = \frac{0.1875}{s} + \frac{0.125}{s^2} - \frac{0.1875}{s+4} - \frac{0.875}{(s+4)^2}$$

$$\rightarrow y(t) = \left(0.1875 + 0.125t - 0.1875e^{-4t} - 0.875te^{-4t}\right)\varepsilon(t)$$

LTI systems representation

Representation of LTI dynamical systems

A SISO LTI system can be described through

- **State space representation** (→ ss)

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t) \\ y(t) = C\,x(t) + D\,u(t) \end{cases}$$

- **Transfer function** (input-output representation → tf)

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_0}$$

In the following, the relationships between these two representations will be discussed, i.e. ss → tf and tf → ss

Consider the Laplace transform of output response:

$$Y(s) = C(sI - A)^{-1}x(0) + \left[C(sI - A)^{-1}B + D\right]U(s)$$

of the LTI system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

In the presence of zero initial conditions x(0) = 0 (i.e. zero state response), we have

$$Y(s) = \left[C(sI - A)^{-1}B + D\right]U(s)$$

$$Y(s) = \underbrace{\left[C(sI - A)^{-1}B + D\right]}_{H(s)}U(s) = H(s)U(s)$$

Thus, given matrices A, B, C and D of the state space representation, the system transfer function can be computed as:

$$H(s) = C(sI - A)^{-1}B + D = C\,\frac{Adj(sI - A)}{\det(sI - A)}B + D$$

State space representation → Transfer function

The solution is unique

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t) \\ y(t) = C\,x(t) + D\,u(t) \end{cases}$$

$$\downarrow$$

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

Example 1

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$
$$y(t) = C\,x(t) + D\,u(t)$$

$$A = \begin{bmatrix} -3 & 2 \\ -2 & -3 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}, D = 0$$

$$H(s) = C\,\frac{Adj(sI - A)}{\det(sI - A)}\,B + D$$

$$sI - A = \begin{bmatrix} s+3 & -2 \\ 2 & s+3 \end{bmatrix} \rightarrow \det(sI - A) = (s+3)^2 + 4 = s^2 + 6s + 13$$

$$Adj(sI - A) = \begin{bmatrix} s+3 & 2 \\ -2 & s+3 \end{bmatrix}$$

$$H(s) = C\,\frac{Adj(sI - A)}{\det(sI - A)}\,B + D \underset{\substack{\uparrow \\ D=0}}{=} \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} \frac{s+3}{s^2+6s+13} & \frac{2}{s^2+6s+13} \\ \frac{-2}{s^2+6s+13} & \frac{s+3}{s^2+6s+13} \end{bmatrix}}_{\frac{Adj(sI-A)}{\det(sI-A)}} \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{B} = -\frac{-2}{s^2+6s+13}$$

- The MatLab statement **`tf`** allows the computation of the transfer function of a dynamical system starting from its ss representation
- Example

$$\dot{x}(t) = \begin{bmatrix} -3 & 2 \\ -2 & -3 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t)$$

- Introduce the system matrices A, B, C (and D)

```
>> A=[-3 2;-2 -3]; B=[1;0]; C=[0 1]; D=0;
```

- Issue the **`ss`** statement

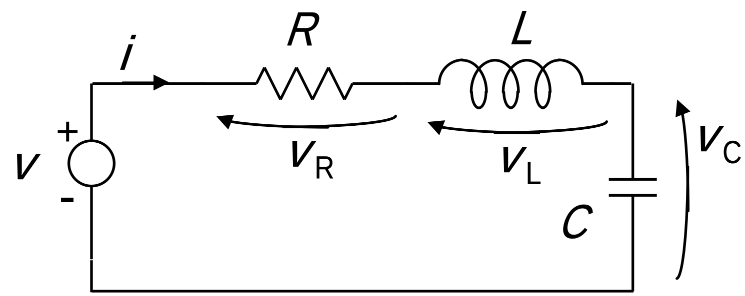
```
>> sys=ss(A,B,C,D)
```

- Use **`tf`** to obtain the transfer function H(s)

```
>> H=tf(sys)


Transfer function:
      -2
--------------
s^2 + 6 s + 13
```

Example 2



$$\begin{cases} \dot{x}_1(t) = \dfrac{1}{L}\left[-R\,x_1(t) - x_2(t) + u(t)\right] \\ \dot{x}_2(t) = \dfrac{1}{C}\,x_1(t) \end{cases}$$

$$y(t) = x_2(t)$$

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$
$$y(t) = C\,x(t) + D\,u(t)$$

$$A = \begin{bmatrix} -\dfrac{R}{L} & -\dfrac{1}{L} \\ \dfrac{1}{C} & 0 \end{bmatrix}, B = \begin{bmatrix} \dfrac{1}{L} \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}, D = 0$$

Example 2

$$H(s) = C\frac{Adj(sI - A)}{\det(sI - A)}B + D$$

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \end{bmatrix}, D = 0$$

$$sI - A = \begin{bmatrix} s + R/L & 1/L \\ -1/C & s \end{bmatrix} \rightarrow \det(sI - A) = (s + R/L)s + 1/(LC)$$

$$= s^2 + R/L\, s + 1/(LC)$$

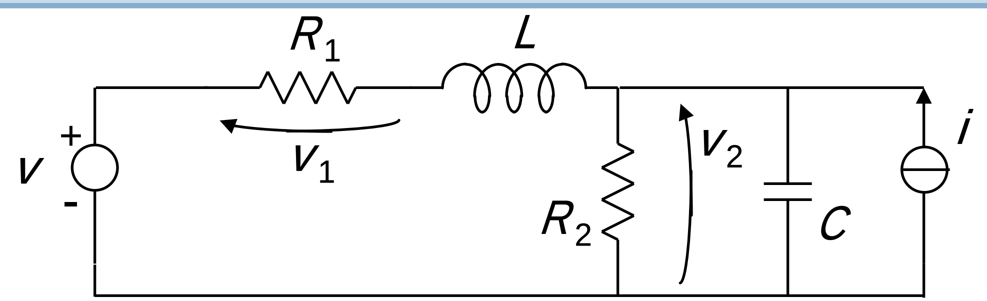
$$Adj(sI - A) = \begin{bmatrix} s & -1/L \\ 1/C & s + R/L \end{bmatrix}$$

Example 2

$$H(s) = C\,\frac{Adj(sI - A)}{\det(sI - A)}B + D =$$

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{s}{s^2 + R/L\,s + 1/(LC)} & \dfrac{-1/L}{s^2 + R/L\,s + 1/(LC)} \\[2ex] \dfrac{1/C}{s^2 + R/L\,s + 1/(LC)} & \dfrac{s + R/L}{s^2 + R/L\,s + 1/(LC)} \end{bmatrix} \begin{bmatrix} 1/L \\ 0 \end{bmatrix} =$$

$$= \frac{1/LC}{s^2 + R/L\,s + 1/(LC)} = \frac{1}{LCs^2 + RCs + 1}$$



State space representation:

$$\begin{cases} \dot{x}_1 = -\dfrac{R_1}{L} x_1 - \dfrac{1}{L} x_2 + \dfrac{1}{L} u_1 \\ \dot{x}_2 = \dfrac{1}{C} x_1 - \dfrac{1}{R_2 C} x_2 + \dfrac{1}{C} u_2 \end{cases}$$

$$\begin{cases} y_1 = R_1 x_1 \\ y_2 = x_2 \end{cases}$$

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$
$$y(t) = C\,x(t) + D\,u(t)$$

$$A = \begin{bmatrix} -R_1/L & -1/L \\ 1/C & -1/R_2 C \end{bmatrix}, B = \begin{bmatrix} 1/L & 0 \\ 0 & 1/C \end{bmatrix}, C = \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Example 3

$$H(s) = C\frac{Adj(sI-A)}{\det(sI-A)}B + D$$

$$A = \begin{bmatrix} -R_1/L & -1/L \\ 1/C & -1/R_2C \end{bmatrix},\ B = \begin{bmatrix} 1/L & 0 \\ 0 & 1/C \end{bmatrix},\ C = \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix},\ D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$sI - A = \begin{bmatrix} s + R_1/L & 1/L \\ -1/C & s + 1/R_2C \end{bmatrix} \rightarrow \det(sI-A) = (s + R_1/L)(s + 1/R_2C) + 1/(LC)$$

$$= s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)$$

$$Adj(sI-A) = \begin{bmatrix} s + 1/R_2C & -1/L \\ 1/C & s + R_1/L \end{bmatrix}$$

Example 3

$$H(s) = C\,\frac{Adj(sI - A)}{\det(sI - A)}\,B + D =$$

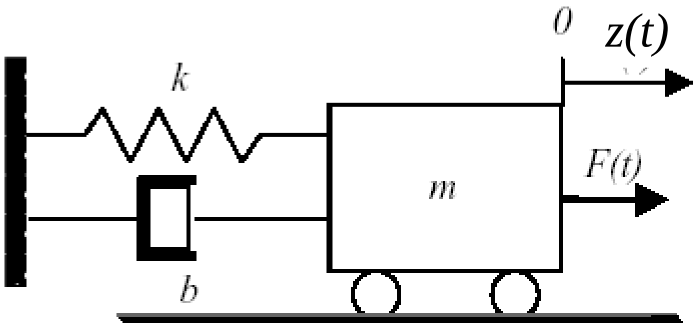
$$= \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{s + 1/R_2C}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} & \dfrac{-1/L}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} \\ \dfrac{-1/C}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} & \dfrac{s + R_1/L_1}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} \end{bmatrix} \begin{bmatrix} 1/L & 0 \\ 0 & 1/C \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{R_1(s + 1/R_2C)/L}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} & \dfrac{-R_1/CL}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} \\ \dfrac{-1/CL}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} & \dfrac{(s + R_1/L_1)C}{s^2 + (R_1/L + 1/R_2C)s + R_1/(LR_2C) + 1/(LC)} \end{bmatrix}$$

→The transfer function *H(s)* of a system with p inputs and q outputs is a $q \times p$ matrix of real rational functions.

Computation of a physical system transfer function

In order to compute the transfer function of a physical system, it is more convenient to $\mathcal{L}$-transform directly the dynamic equations in the presence of zero initial conditions than deriving at first the ss representation and then applying the relation $C(sI-A)^{-1}B+D$



$F(t) = u(t)$ input

(applied force)

$z(t) = y(t)$ output

(mass position)

$$m\ddot{z}(t) = -kz(t) - b\dot{z}(t) + F(t)$$

$$m\ddot{y}(t) = -ky(t) - b\dot{y}(t) + u(t)$$

Computation of a physical system transfer function

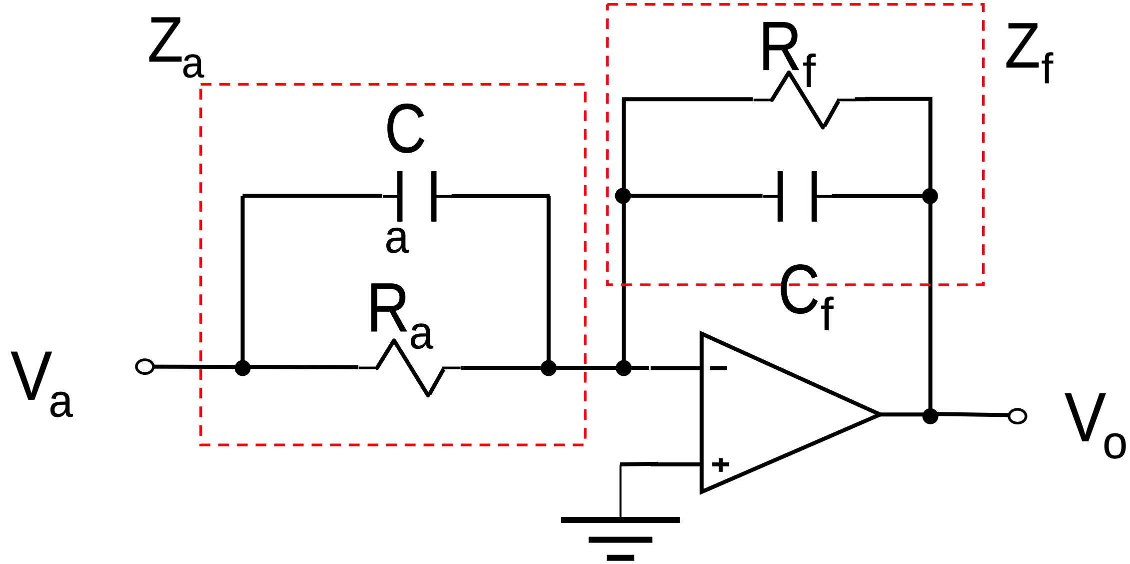
$$m\ddot{y}(t) = -ky(t) - b\dot{y}(t) + u(t)$$

$$\begin{array}{ccccc} m\ddot{y}(t) & = & -ky(t) & -b\dot{y}(t) & +u(t) \\ \downarrow \mathcal{L} & & \downarrow \mathcal{L} & \downarrow \mathcal{L} & \downarrow \mathcal{L} \\ ms^2Y(s) & = & -kY(s) & -bsY(s) & +U(s) \end{array}$$

$$(ms^2 + bs + k)Y(s) = U(s)$$

$$Y(s) = \underbrace{\frac{1}{ms^2 + bs + k}}_{H(s)} U(s)$$

Computation of a physical system transfer function



$$V_o(s) = -\frac{Z_f(s)}{Z_a(s)}V_a(s) = -\frac{\left(C_f s + \frac{1}{R_f}\right)^{-1}}{\left(C_a s + \frac{1}{R_a}\right)^{-1}}V_a(s) = \underbrace{-\frac{R_f}{R_a}\cdot\frac{1+R_aC_a s}{1+R_fC_f s}}_{H(s)}V_a(s)$$

Transfer function → State space representation

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_0}$$

$$\downarrow$$

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t) \\ y(t) = C\,x(t) + D\,u(t) \end{cases}$$

$$A = ??, B = ??, C = ??, D = ??$$

The solution is not unique

In order to compute a ss representation of a given tf $H(s)$, some preliminary manipulations of $H(s)$ are needed when $H(s)$ is not strictly proper (i.e. $m = n$):

$$H(s) = \frac{b'_n s^n + b'_{n-1} s^{n-1} + \ldots + b'_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_0} \underset{\substack{\uparrow \\ \text{divide numerator by denominator}}}{=}$$

$$= \frac{b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} + b_n$$

When $H(s)$ is strictly proper (i.e. $m < n$) preliminary manipulations are not needed:

$$H(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_0}$$

Consider

$$H(s) = \frac{b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} + b_n$$

$$H(s) = \frac{Y(s)}{U(s)} \rightarrow Y(s) = H(s)U(s) = \left(\frac{b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} + b_n \right) U(s)$$

Define the intermediate variable $X_1(s)$ as:

$$X_1(s) = \frac{1}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} U(s)$$

$$Y(s) = X_1(s)\left(b_{n-1}s^{n-1} + \ldots + b_1 s + b_0\right) + b_n U(s)$$

$$X_1(s) = \frac{1}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} U(s) \rightarrow X_1(s)(s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0) = U(s)$$

$$\rightarrow \frac{d^n x_1(t)}{dt^n} + a_{n-1}\frac{d^{n-1}x_1(t)}{dt^{n-1}} + \ldots + a_1 \frac{dx_1(t)}{dt} + a_0 x_1(t) = u(t)$$

Define the states $x_2, \ldots, x_n$ as

$$\begin{cases} x_2(t) = \dfrac{dx_1(t)}{dt} = \dot{x}_1(t) \\ x_3(t) = \dfrac{d^2 x_1(t)}{dt^2} = \dfrac{dx_2(t)}{dt} = \dot{x}_2(t) \\ \ldots \\ x_{n-1}(t) = \dfrac{d^{n-2}x_1(t)}{dt^{n-2}} = \dfrac{dx_{n-2}(t)}{dt} = \dot{x}_{n-2}(t) \\ x_n(t) = \dfrac{d^{n-1}x_1(t)}{dt^{n-1}} = \dfrac{dx_{n-1}(t)}{dt} = \dot{x}_{n-1}(t) \\ \rightarrow \dfrac{dx_n(t)}{dt} = \dot{x}_n(t) = \dfrac{d^n x_1(t)}{dt^n} = -a_{n-1}x_{n-1}(t) - \ldots - a_1 x_2(t) - a_0 x_1(t) + u(t) \end{cases}$$

The state equation is:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = x_3(t) \\ \vdots \\ \dot{x}_n(t) = -a_0 x_1(t) - a_1 x_2(t) - \ldots - a_{n-1} x_n(t) + u(t) \end{cases} \rightarrow$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Note that matrix A is expressed in the lower companion form

→ the characteristic polynomial is $p_A(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_1\lambda + a_0$

As to the output equation, we have:

$$Y(s) = X_1(s)\left(b_{n-1}s^{n-1} + \ldots + b_1 s + b_0\right) + b_n U(s)$$

$$y(t) = b_{n-1}\underbrace{\frac{d^{n-1}x_1(t)}{dt^{n-1}}}_{x_n(t)} + \ldots + b_1 \underbrace{\frac{dx_1(t)}{dt}}_{x_2(t)} + b_0 x_1(t) + b_n u(t) =$$

$$= b_{n-1}x_n(t) + \ldots + b_1 x_2(t) + b_0 x_1(t) + b_n u(t)$$

$$C = \begin{bmatrix} b_0 & b_1 & \cdots & b_{n-1} \end{bmatrix} \qquad D = \begin{bmatrix} b_n \end{bmatrix}$$

Reordering we obtain the **controller canonical form**:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = x_3(t) \\ \vdots \\ \dot{x}_n(t) = -a_0 x_1(t) - a_1 x_2(t) - \ldots - a_{n-1} x_n(t) + u(t) \end{cases}$$

$$y(t) = b_0 x_1(t) + b_1 x_2(t) + \ldots + b_{n-1} x_n(t) + b_n u(t)$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} b_0 & b_1 & \cdots & b_{n-1} \end{bmatrix} \quad D = \begin{bmatrix} b_n \end{bmatrix}$$

tf → ss (controller canonical form)

$$H(s) = \frac{b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} + b_n$$

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} b_0 & b_1 & \cdots & b_{n-1} \end{bmatrix} \qquad D = \begin{bmatrix} b_n \end{bmatrix}$$

Controller canonical form: example

$$H(s) = \frac{s^2+3s+1}{s^3+s^2+s+1} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} + b_3$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} C = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} D = \begin{bmatrix} b_3 \end{bmatrix}$$

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 3 & 1 \end{bmatrix} x(t)$$

tf → ss observer canonical form

$$H(s)=\frac{b_{n-1}s^{n-1}+\ldots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\ldots+a_1 s+a_0}+b_n$$

$$\begin{cases}\dot{x}(t)=Ax(t)+Bu(t)\\ y(t)=Cx(t)+Du(t)\end{cases}$$

$$A=\begin{bmatrix}0 & \cdots & 0 & -a_0\\ 1 & \ddots & \ddots & -a_1\\ 0 & \ddots & 0 & \vdots\\ 0 & \cdots & 1 & -a_{n-1}\end{bmatrix}\qquad B=\begin{bmatrix}b_0\\ b_1\\ \vdots\\ b_{n-1}\end{bmatrix}$$

$$C=\begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}\qquad D=\begin{bmatrix}b_n\end{bmatrix}$$

Observer canonical form: example

$$H(s)=\frac{s^2+3s+1}{s^3+s^2+s+1}=\frac{b_2s^2+b_1s+b_0}{s^3+a_2s^2+a_1s+a_0}+b_3$$

$$A=\begin{bmatrix}0 & \cdots & 0 & -a_0\\ 1 & \ddots & \ddots & -a_1\\ 0 & \ddots & 0 & \vdots\\ 0 & \cdots & 1 & -a_{n-1}\end{bmatrix}\quad B=\begin{bmatrix}b_0\\ b_1\\ \vdots\\ b_{n-1}\end{bmatrix}$$

$$C=\begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}\qquad D=\begin{bmatrix}b_n\end{bmatrix}$$

$$\dot{x}(t)=\begin{bmatrix}0 & 0 & -1\\ 1 & 0 & -1\\ 0 & 1 & -1\end{bmatrix}x(t)+\begin{bmatrix}1\\ 3\\ 1\end{bmatrix}u(t)$$

$$y(t)=\begin{bmatrix}0 & 0 & 1\end{bmatrix}x(t)$$

- Computation of a ss representation of a tf using MatLab

$$H(s) = \frac{s^2 + 3s + 1}{s^3 + s^2 + s + 1}$$

- Define the transfer fuction as usual

```
>> s=tf('s')

 Transfer function:

 s

 >> H=(s^2+3*s+1)/(s^3+s^2+s+1)


 Transfer function:

   s^2 + 3 s + 1

 -----------------

 s^3 + s^2 + s + 1
```

- Use **ss** to obtain the A, B, C, D matrices

```
>> sys=ss(H)
a =       x1    x2    x3
   x1     -1  -0.5  -0.5
   x2      2     0     0
   x3      0     1     0
b =     u1
   x1   2
   x2   0
   x3   0
c =     x1    x2    x3
   y1   0.5  0.75  0.25

d =     u1
    y1   0
```