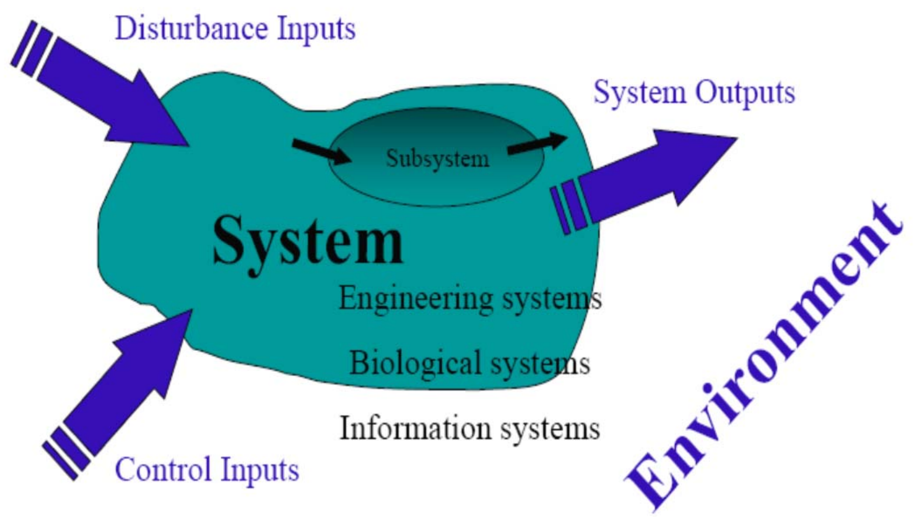
MM2 Essentials for Feedback Control

Readings:

- Section 2.1 (models of mechanical systems, p.20-33);
- Section 3.1 (Laplace transform, p.86-110)
- Section 3.2 (Block diagram, p.111-118)

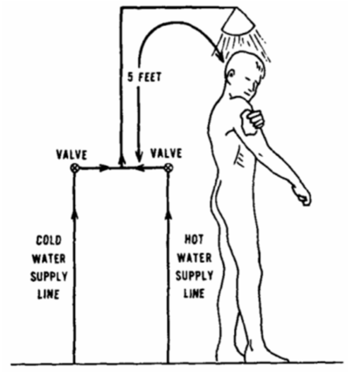
MM1: Basic Concept (I): System and its Variables

- A **system** is a collection of components which are coordinated together to perform a function
- Systems interact with their **environment**. The interaction is defined in terms of **variables**
  - System inputs
  - System outputs
  - Environmental disturbances
- **Dynamic system** is a system ehose performance could change according to time

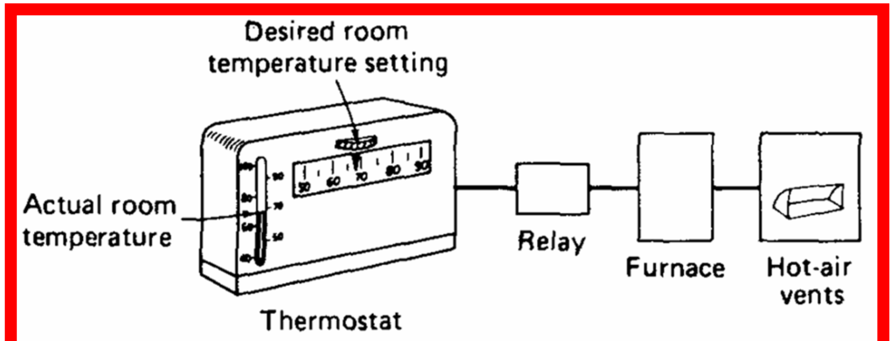


MM1: Basic Concept (II): Control

- **Control** is a process of causing a system (output) variable to conform to some desired status/value
- **Manual Control** is a process where the **control** is handled by human being(s)
- **Automatic Control** is a control process which involves machines only



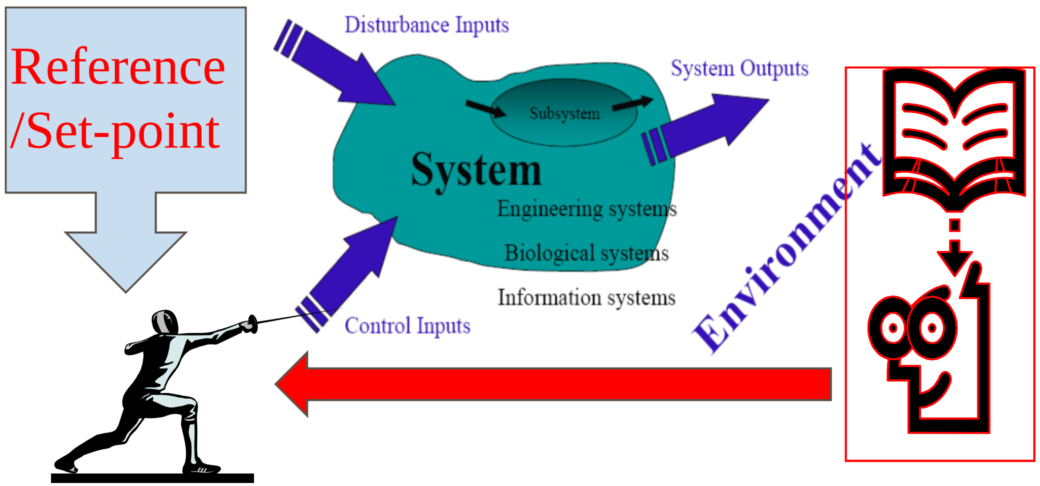
Flow diagram for shower example.



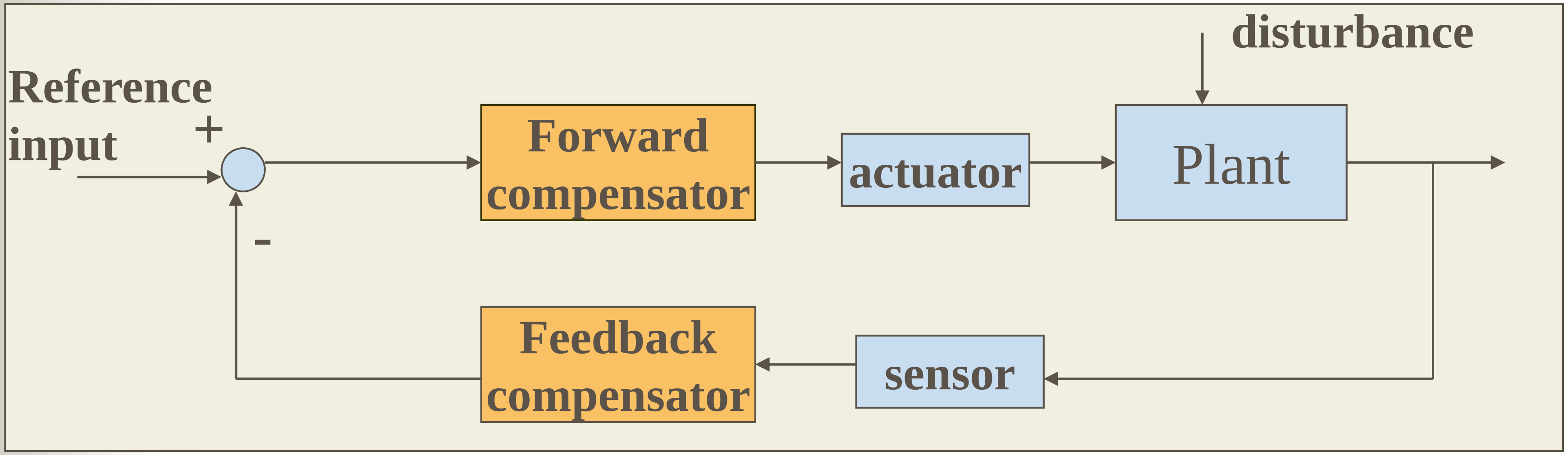
Home heating control system.

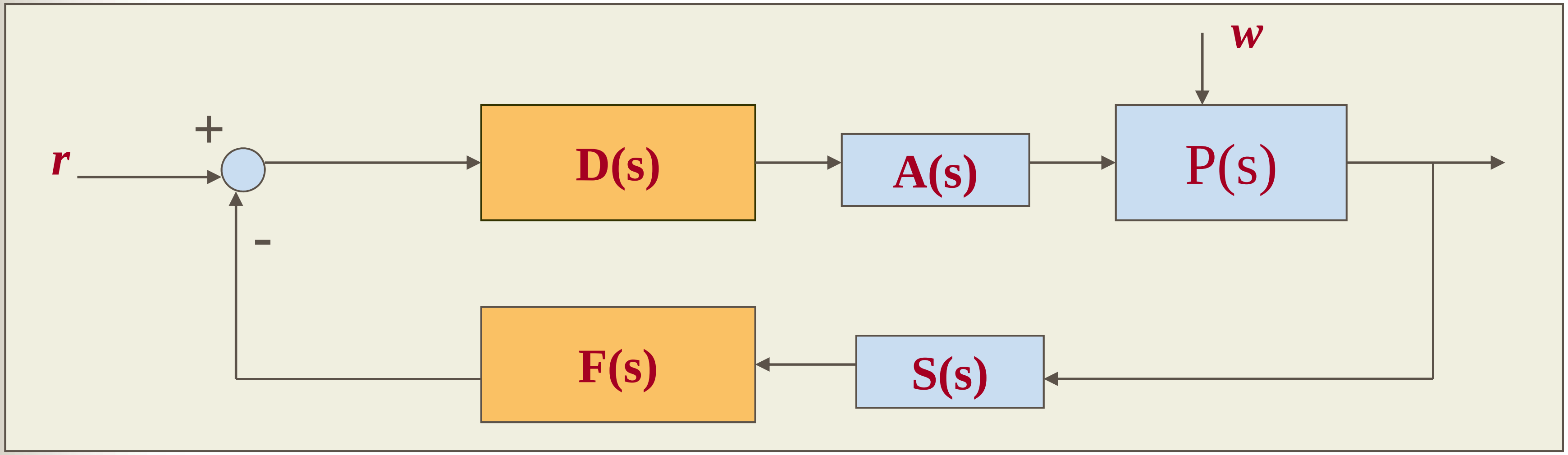
MM1: Control Classification

- **Open-loop Control:** A control process which does not utilize the feedback mechanism, i.e., the output(s) has no effect upon the control input(s)
- **Closed-loop Control:** A control process which utilizes the feedback mechanism, i.e., the output(s) does have effect upon the control input(s)



MM1: Feedback Control – Block Diagrams





MM1: System Model: Transfer Function

- **Num-den form   sys=tf(num,den)**

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_1 s^m + b_2 s^{m-1} + \cdots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \cdots + b_{n+1}}, \quad e.g., \quad G(s) = \frac{K}{s^2 + 2\zeta\omega s + \omega^2}$$

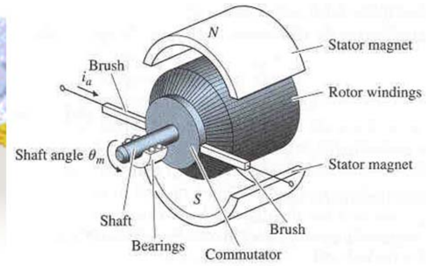
- **Zero-pole form   sys=zpk(Z,P,K)**

$$G(s) = \frac{Y(s)}{U(s)} = K \frac{\prod_{i=1}^{m} (s - z_i)}{\prod_{i=1}^{n} (s - p_i)} \quad e.g., \quad G(s) = \frac{K}{(s - p_1)(s - p_2)}$$

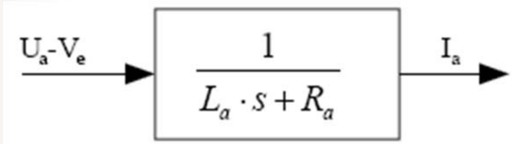
- Overview of system features

ltiview(sys)

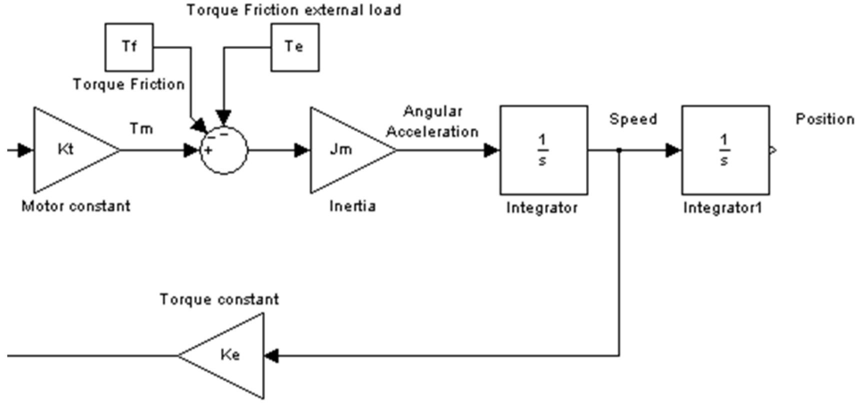
MM1: Modeling: DC-motor

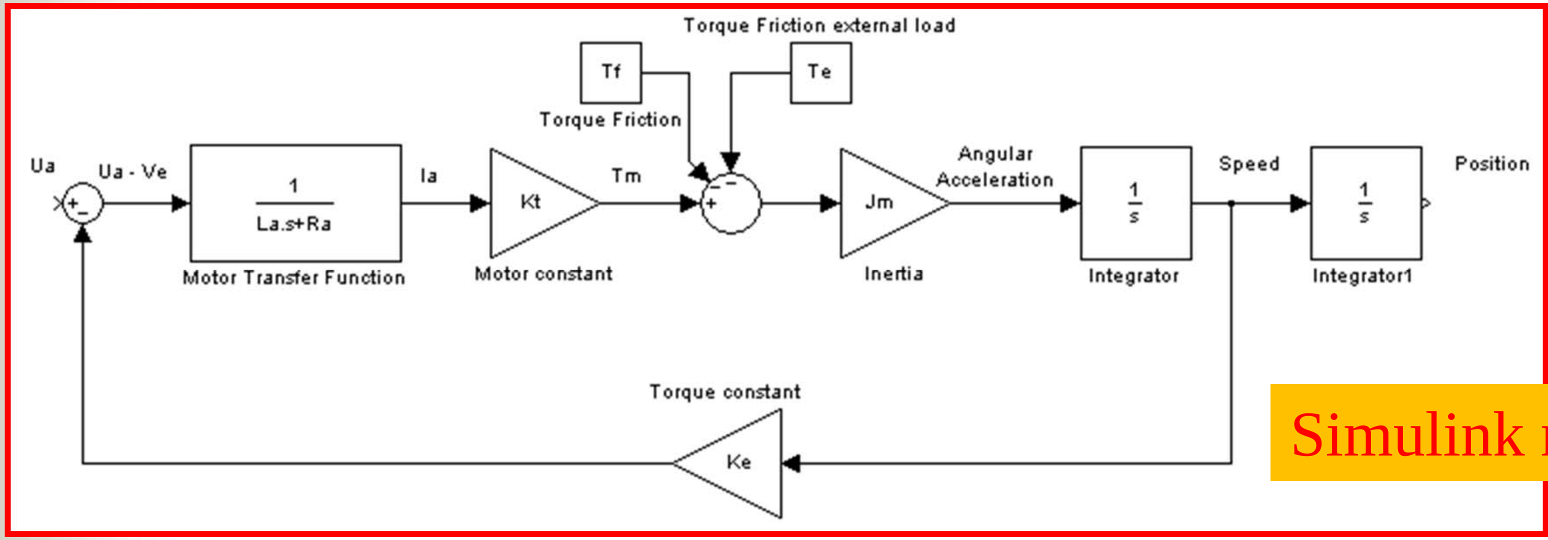


- Complete model:



: Block diagram of the electrical part of the DC-motor.





Simulink model

The Goals of this lecture (MM2) …

- Essentials in using (ordinary) differential equation model
  - Why use ODE model
  - Linear vs. nonlinear ODE models
  - How to solve an ODE
  - Numerical methods (Matlab)
- Refresh of Laplace transform
  - Key features
  - Transformation from ODE to TF model
- Block diagram transformation
  - Composition /decomposition
  - Signal-flow graph

The Goals of this lecture…

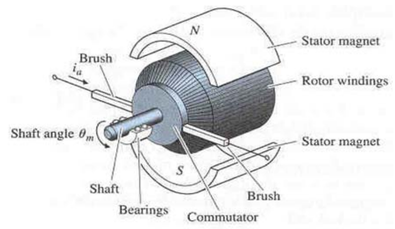
- Essentials in using (ordinary) differential equation model
  - Why use ODE model
  - Linear vs. nonlinear ODE models
  - How to solve an ODE
  - Numerical methods (Matlab)
- Laplace transform
  - Key features
  - Transformation from ODE to TF model
- Block diagram transformation
  - Composition /decomposition
  - Signal-flow graph

Essentials in ODE - **What's an ODE Model (I)?**

- An **ODE model** is a set of ODEs that describe the dynamic behavior of the considered system (in terms of system input and output variables – **IO Model**)

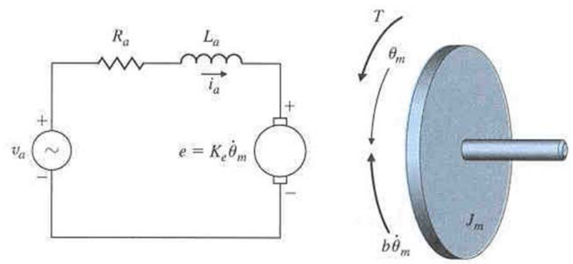
$$L_a \frac{di_a}{dt} + R_a i_a = v_a - K_e \dot{\theta}_m$$

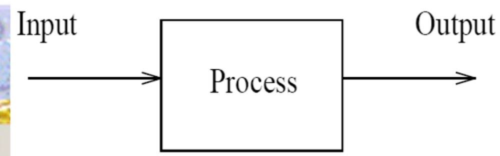
$$J_m \ddot{\theta}_m + b\dot{\theta}_m = K_t i_a - T_e - T_f$$



- Variables: input, output, internal

$$J_m \ddot{\theta}_m + (b + \frac{K_t K_e}{R_a})\dot{\theta}_m = \frac{K_t}{R_a} v_a - T_e - T_f$$





Essentials in ODE - **What's an ODE Model (II)?**

- A general ODE model:

$$a_n \frac{d^n y(t)}{dt^n} + \cdots + a_1 \frac{dy(t)}{dt} + a_0\, y(t) = b_m \frac{d^m u}{dt^m} + \cdots + b_1 \frac{du}{dt} + b_0\, u$$

- Assumption: $n > m$
- Single-input single-output (**SISO**) model
- SIMO, MISO, MIMO models
- Linear system
- Time-invance
- Linear Time-Invarnace (**LTI**) model

Essentials in ODE – **Why Use ODE Model?**

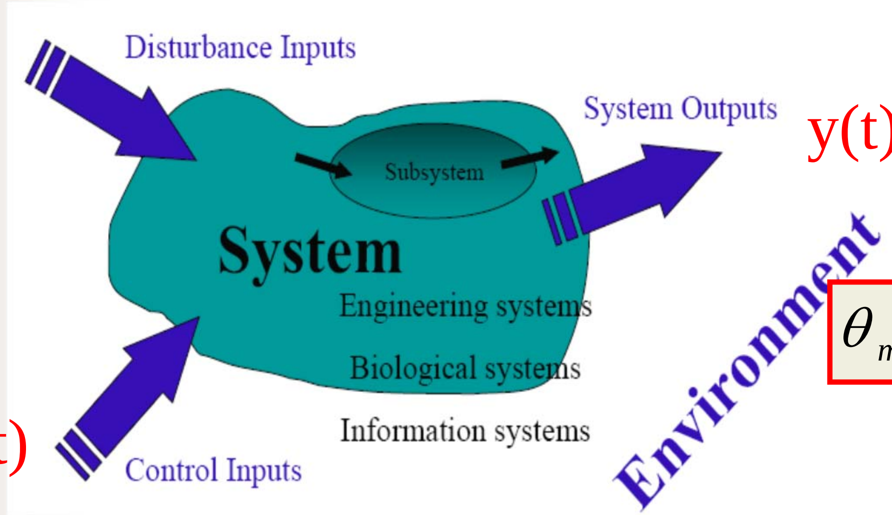
- The ODE model can be naturally obtained through modeling procedure by following some physical principles

    Get your feeling of that through Modeling lectures...

- The ODE model is an efficient but impelicit description of the dynamic system's behavior

d(t)

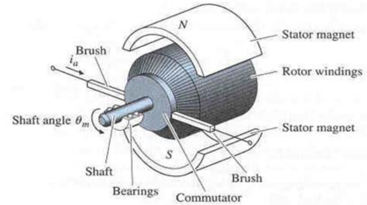
u(t)



y(t)

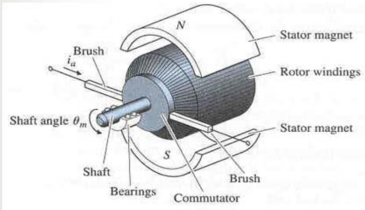
$$y(t)=\mathbf{f(}d(t), u(t), y(0)\mathbf{)}$$

$$\theta_m = f(\theta_m(0), \dot{\theta}_m(0), v_a, T_e, T_f)$$



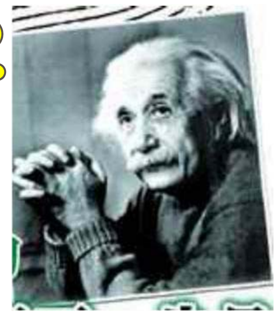
Essentials in ODE –**How to interperate ODE Model?**

- The ODE model is an impelicit description of the dynamic system's behavior



$$J_m \ddot{\theta}_m + \left(b + \frac{K_t K_e}{R_a}\right)\dot{\theta}_m = \frac{K_t}{R_a} v_a - T_e - T_f$$

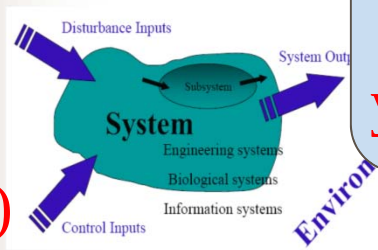
Solution of ODE!



$$\theta_m = f(\theta_m(0), \dot{\theta}_m(0) v_a, T_e, T_f)$$

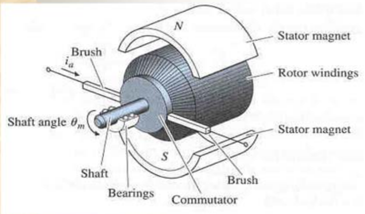
y(t)=**f(**d(t), u(t), y(0)**)**

d(t)



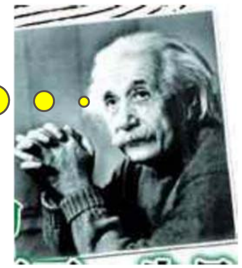
u(t)

Essentials in ODE – **Uniqueness of ODE Solution**



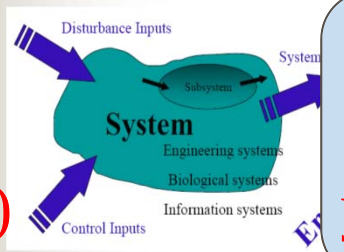
$$J_m \ddot{\theta}_m + (b + \frac{K_t K_e}{R_a}) \dot{\theta}_m = \frac{K_t}{R_a} v_a - T_e - T_f$$

Solution of ODE
Is an explicit model



d(t)

u(t)



$$\theta_m = f(\theta_m(0), \dot{\theta}_m(0), v_a, T_e, T_f)$$

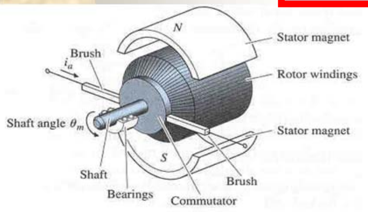
y(t)=**f(**d(t), u(t), y(0)**)**

Conditions for unique solution of an ODE:

- Initial conditions
- Lipschiz condition

Essentials in ODE – **How to Solve an ODE?**

$$J_m \ddot{\theta}_m + \left(b + \frac{K_t K_e}{R_a}\right)\dot{\theta}_m = \frac{K_t}{R_a} v_a - T_e - T_f$$



$$\theta_m(0) = \alpha, \quad \dot{\theta}_m(0) = \beta$$

$$\theta_m = f(\theta_m(0), \dot{\theta}_m(0), v_a, T_e, T_f)$$

Solving an ODE:

- Time-domain method
- Complex-domain method (**Laplace transform**)
- Numerical solution – CAD methods

Essentials in ODE – **ODE Solvers in Matlab**

Differential Equation Solver

MATLAB's functions for solving ordinary differential equations are

ode23 – 2nd/3rd order Runge-Kutta method

ode45 – 4th/5th order Runge-Kutta method

example,

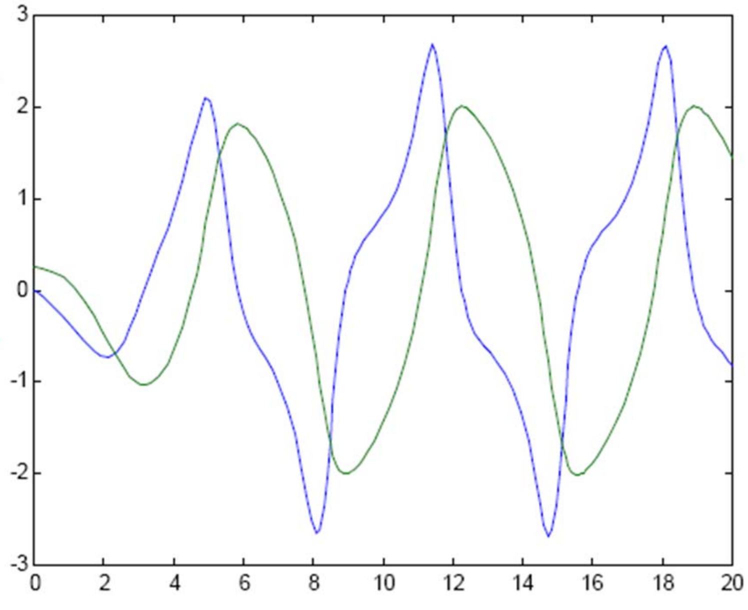
$$\dot{x}_1 = x_1(1 - x_2^2) - x_2$$

$$\dot{x}_2 = x_1$$

Create a file called "vdpol.m",

```
function xdot = vdpol(t,x)
xdot = zeros(2,1);
xdot(1) = x(1).*(1-x(2).^2)-x(2);
xdot(2) = x(1);
```

```
t0 = 0; tf = 20;
x0 = [0 0.25]';  % Initial c
[t,x] = ode23('vdpol', [t0
plot (t, x)
```



"help ode23" to find out all the other options.

The Goals of this lecture…

- Essentials in using (ordinary) differential equation model
  - Why use ODE model
  - Linear vs. nonlinear ODE models
  - How to solve an ODE
  - Numerical methods (Matlab)
- **Refresh of Laplace transform**
  - **Key features**
  - **Transformation from ODE to TF model**
- Block diagram transformation
  - Composition /decomposition
  - Signal-flow graph

Laplace Transform - **Definition**

Time function $\overset{\text{Transform}}{\leftrightarrow}$ Complex function

$(f(t) \leftrightarrow F(s)$, where $s = \sigma + j\omega$, $j = \sqrt{-1})$

Definitions:

- Laplace transform:

$$\mathcal{L}[f(t)] = \int_{0_-}^{\infty} f(t)e^{-st}\,dt = F(s)$$

- Inverse Laplace transform:

$$\mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st}\,ds = f(t)$$

Laplace Transform – **Basic Pairs**

Time function $f(t)$	Laplace transform $F(s)$
$\delta(t)$, unit impulse	1
1, unit step	$\frac{1}{s}$
t	$\frac{1}{s^2}$
e^{-at}	$\frac{1}{s+a}$
$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$

Laplace Transform - **Properties**

$$(a)\ \mathcal{L}[\alpha f(t) + \beta g(t)] = \alpha F(s) + \beta G(s)$$

$$(b)\ \mathcal{L}^{-1}[\alpha F(s) + \beta G(s)] = \alpha f(t) + \beta g(t)$$

$$(c)\ \mathcal{L}[\frac{df(t)}{dt}] = sF(s) - f(0_-)$$

$$(d)\ \mathcal{L}[\int_{-\infty}^{t} f(t)\,dt] = \frac{F(s)}{s} + \frac{1}{s}\int_{-\infty}^{0} f(t)\,dt$$

$$(e)\ \mathcal{L}[e^{-at} f(t)] = F(s+a)$$

$$(f)\ \lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$$

Laplace Transform – **ODE**

$$a_n \frac{d^n y(t)}{dt^n} + \cdots + a_1 \frac{dy(t)}{dt} + a_0\, y(t) = b_m \frac{d^m u}{dt^m} + \cdots + b_1 \frac{du}{dt} + b_0\, u$$

⇓

$$(a_n s^n + \cdots + a_1 s + a_0) Y(s) = (b_m s^m + \cdots + b_1 s + b_0) U(s)$$

⇓

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \cdots + b_1 s + b_0}{a_n s^n + \cdots + a_1 s + a_0}$$

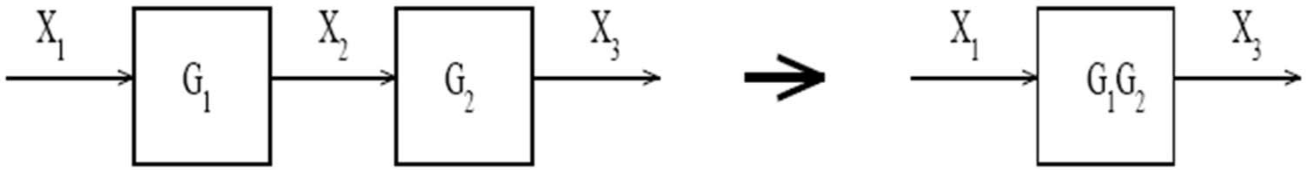
$$\mathcal{L}[\frac{d^k f(t)}{dt^k}] = s^k F(s) - s^{k-1} f(0_-) - \cdots - s f^{(k-2)}(0_-) - f^{(k-1)}(0_-)$$

The Goals of this lecture…

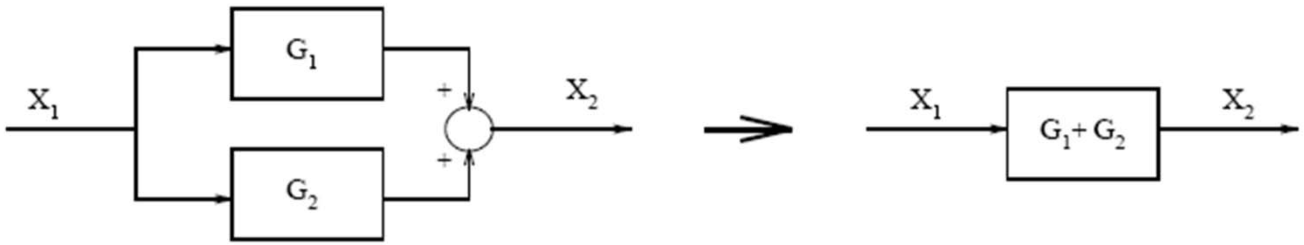
- Essentials in using (ordinary) differential equation model
  - Why use ODE model
  - Linear vs. nonlinear ODE models
  - How to solve an ODE
  - Numerical methods (Matlab)
- Refresh of Laplace transform
  - Key features
  - Transformation from ODE to TF model
- **Block diagram transformation**
  - **Composition /decomposition**
  - **Signal-flow graph**

Block diagram – **Rules (I)**

Combining blocks in cascade:

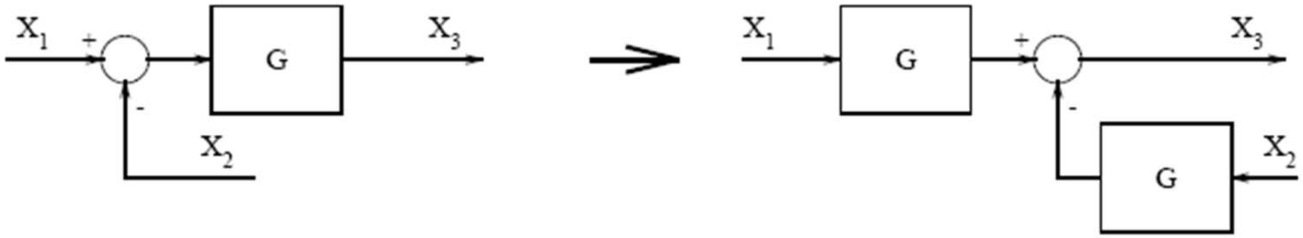


Combining blocks in parallel:

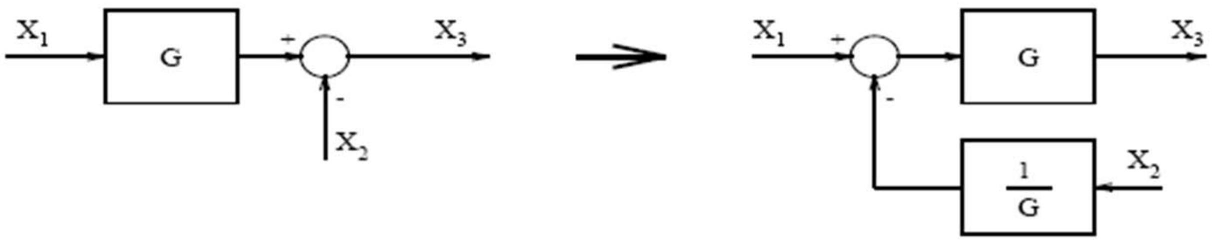


Block diagram – **Rules (II)**

Moving a summing point forward:

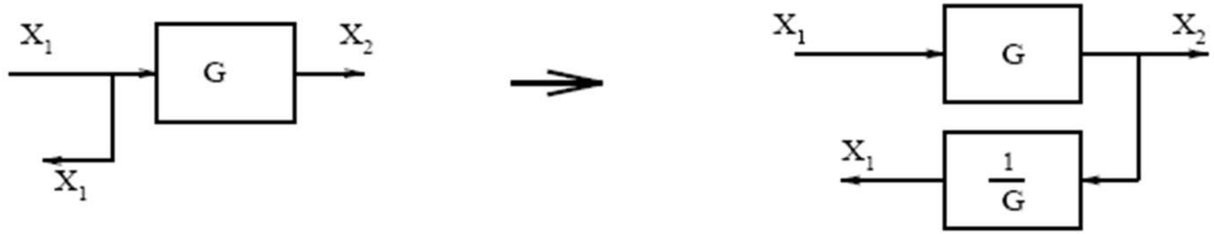


Moving a summing point backward:

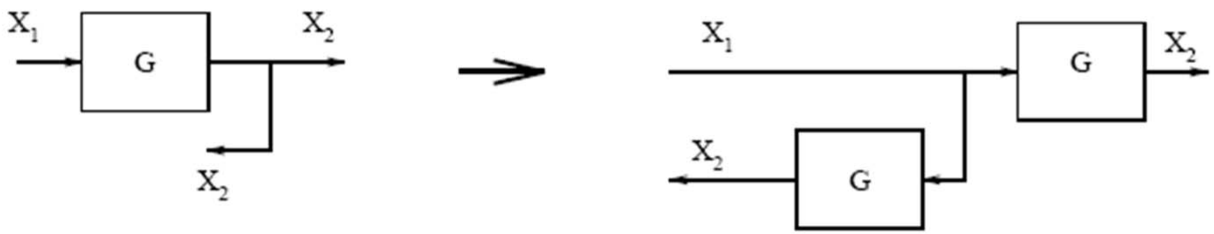


Block diagram – **Rules (III)**

Moving a pickoff point forward:

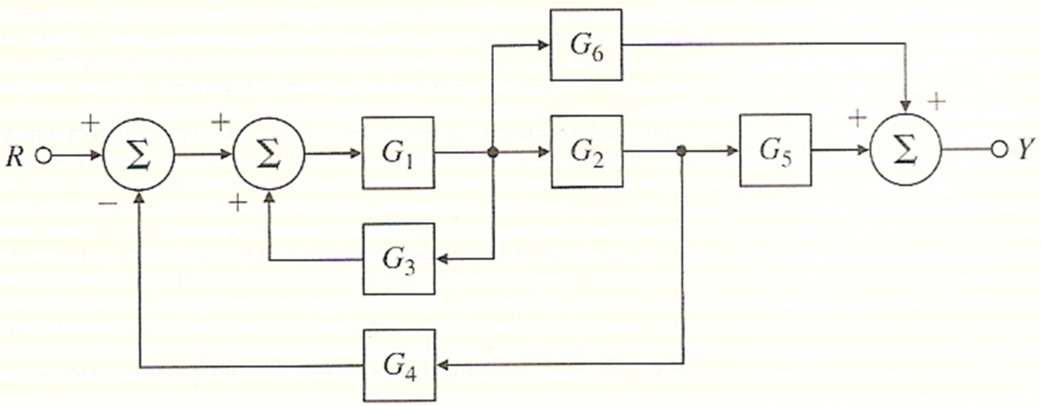


Moving a pickoff point backward:

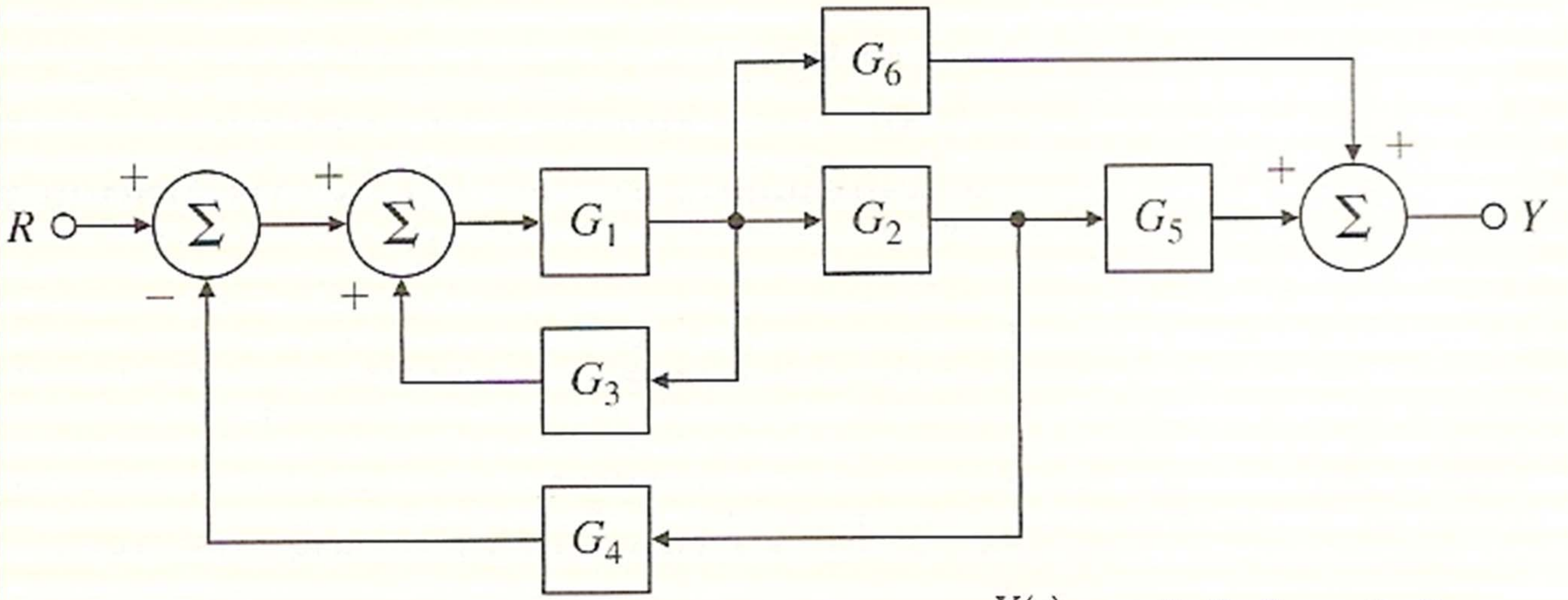


Block diagram – **Rules (IV)**

Eliminating a feedback loop:



Block diagram – **Example**



$$\frac{Y(s)}{R(s)} = \frac{G_1G_2G_5 + G_1G_6}{1 - G_1G_3 + G_1G_2G_4}$$

The Goals of this lecture…

- Essentials in using (ordinary) differential equation model
  - Why use ODE model
  - Linear vs. nonlinear ODE models
  - How to solve an ODE
  - Numerical methods (Matlab)
- Refresh of Laplace transform
  - Key features
  - Transformation from ODE to TF model
- Block diagram transformation
  - Composition /decomposition
  - Signal-flow graph

Execise Two

- See the distributed paper