Lecture 3
Control and sampling systems

Harry J. Whitlow
Department of Physics
University of Jyväskylä

Topics

- Control and sampling system implementation
  - Transfer function
  - Block diagrams,
  - Single input - single output systems
    - Laplace transform approach,
  - Multi-input multi-output systems,
    - space-state equation approach.
  - MATLAB simulations of transient response

Solution of differential equations using Laplace transforms

A simple example differential equation with initial conditions

$$\ddot{x} + 3\dot{x} + 2x = 0; \quad x(0) = a; \quad \dot{x}(0) = b$$

Laplace transform

$$\mathcal{L}\left[x(t)\right] = X(s)$$

Then the real differential theorem gives:

$$\mathcal{L}\left[\dot{x}\right] = sX(s) - x(0); \quad \mathcal{L}\left[\ddot{x}\right] = s^2X(s) - sx(0) - \dot{x}(0)$$

Substitute into the differential eqn. then include initial conditions.

$$\mathcal{L}\left[s^2X(s) - sx(0) - \dot{x}(0)\right] + 3\left[sX(s) - x(0)\right] + 2X(s) = 0$$

$$\Rightarrow \left[s^2X(s) - sa - b\right] + +3\left[sX(s) - a\right] + 2X(s) = 0$$

$$\therefore \left(s^2 + 3s + 2\right)X(s) = as + b + 3a$$

Solve algebraically for $X(s)$.

$$\therefore X(s) = \frac{as + b + 3a}{\left(s^2 + 3s + 2\right)} = \frac{as + b + 3a}{(s+2)(s+1)} = -\frac{(a+b)}{(s+2)} + \frac{(2a+b)}{(s+1)}$$

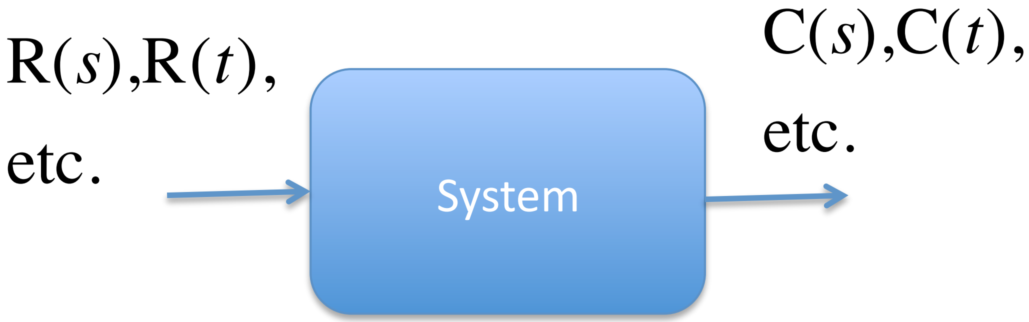
Transform back to get x(t).

$$\therefore x(t) = -(a+b)e^{-2t} + (2a+b)e^{-t}; \quad t > 0$$

$$\because \mathcal{L}\left[e^{-\alpha t}\right] = \frac{1}{s + \alpha}$$

(After Ogata)

The transfer function of a system



•System is said to be linear if the principle of superposition applies.

•Output for two inputs is not a superposition of the individual outputs for two inputs.

$$C(s) = G(s)R(s)$$

Transfer functions

- The transfer function for a *linear* and *time invariant* differential equation system is defined as:

$$G(s) = \left. \frac{\mathbf{L}\left[\text{output} = \text{response function}\right]}{\mathbf{L}\left[\text{input} = \text{driving function}\right]} \right|_{\text{Zero initial conditions}}$$

Transfer functions (cont)

- Assume a linear time invariant differential equation describes the system.

$$a_0 \overset{(m)}{y} + a_1 \overset{(m-1)}{y} + \cdots + a_{m-2}\ddot{y} + a_{m-1}\dot{y} + a_m y =$$
$$b_0 \overset{(n)}{y} + b_1 \overset{(n-1)}{y} + \cdots + b_{n-2}\ddot{y} + b_{n-1}\dot{y} + b_n y; \quad n \geq m$$

- Using the Laplace transform concept we can represent the differential equation description of the dynamics by an algebraic description in *s*.

$$G(s) = \frac{\mathbf{L}[\text{output}]}{\mathbf{L}[\text{input}]}$$
$$\Rightarrow G(s) = \frac{Y(s)}{X(s)}$$
$$= \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-2}s^2 + b_{n-1}s + b_n}{a_0 s^m + a_1 s^{m-1} + \cdots + a_{m-2}s^2 + a_{m-1}s + a_m}$$

- If the highest power of *s* is *n* we say the system is of *n*th-order

Properties of the transfer function

1. $G(s)$ is a mathematical model of the system that relates the output variable to the input variable

2. $G(s)$ is a property of the system and is not influenced by the driving or output function.

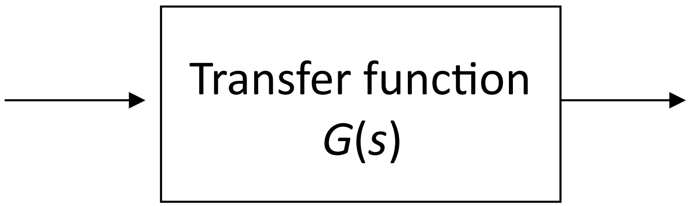
3. $G(s)$ does not say anything (directly!) about the physical structure of the system

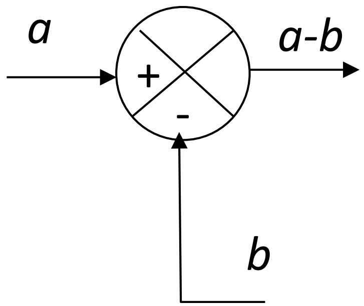
4. If $G(s)$ is known then the behavior for different kinds of driving functions can be determined.

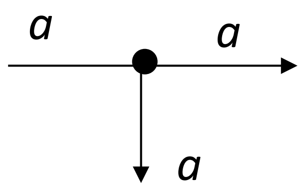
5. $G(s)$ may be established experimentally. Once known it provides a complete description of the dynamics of the system

BLOCK DIAGRAMS

Block diagram elements







Block diagram element

- Does not load other elements
- Signal flow denoted by arrows
- Input and output need not have same dimensions and units

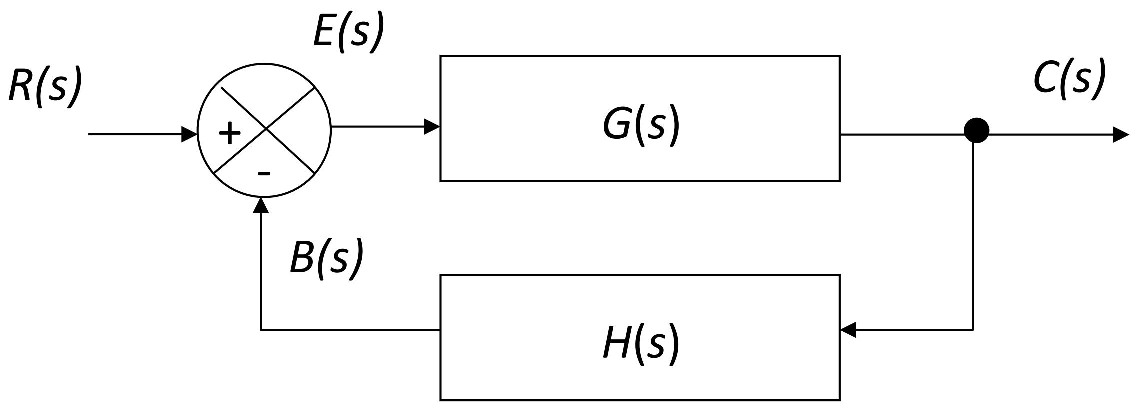
Summing point

- +/- signs denote adition subtraction
- Quantites added must have same units and same dimensions

Node point

- Branch point where signal is sent ot other blocks/summing points
- Quanttites on arms of node have same units and   dimensions

Closed loop block diagram



- *The open-loop transfer function*: describes ratio of $B(s)$ to error actuating signal $E(s)$:

$$\frac{B(s)}{E(s)} = G(s)H(s)$$

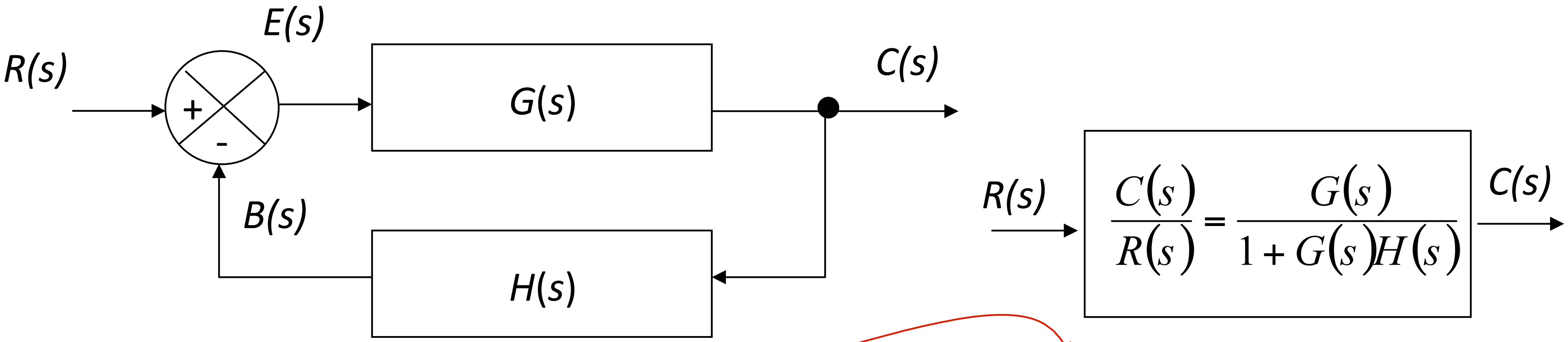
- *The feedforward transfer function* is the ratio of output $C(s)$ to the error actuating signal $E(s)$:

$$\frac{C(s)}{E(s)} = G(s)$$

- The *closed loop transfer function* is ratio of output to input signal:

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

Closed-loop transfer function



$$C(s) = G(s)E(s); \qquad E(s) = C(s)/G(s)$$

$$E(s) = R(s) - B(s) = R(s) - H(s)C(s)$$

$$\frac{E(s)}{E(s)} = 1 = \frac{G(s)}{C(s)}\left[R(s) - H(s)C(s)\right]$$

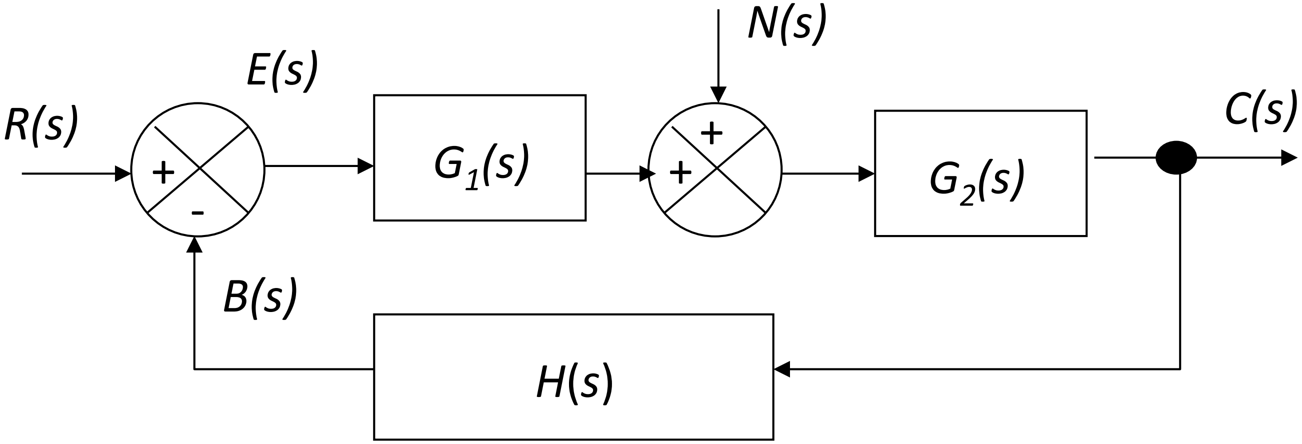
$$1 = \frac{G(s)}{C(s)}R(s) - \frac{G(s)}{C(s)}H(s)C(s)$$

$$\frac{G(s)R(s)}{C(s)} - 1 = \frac{G(s)}{C(s)}H(s)C(s)$$

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

$$\boxed{\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)}}$$

Closed-loops and disturbance



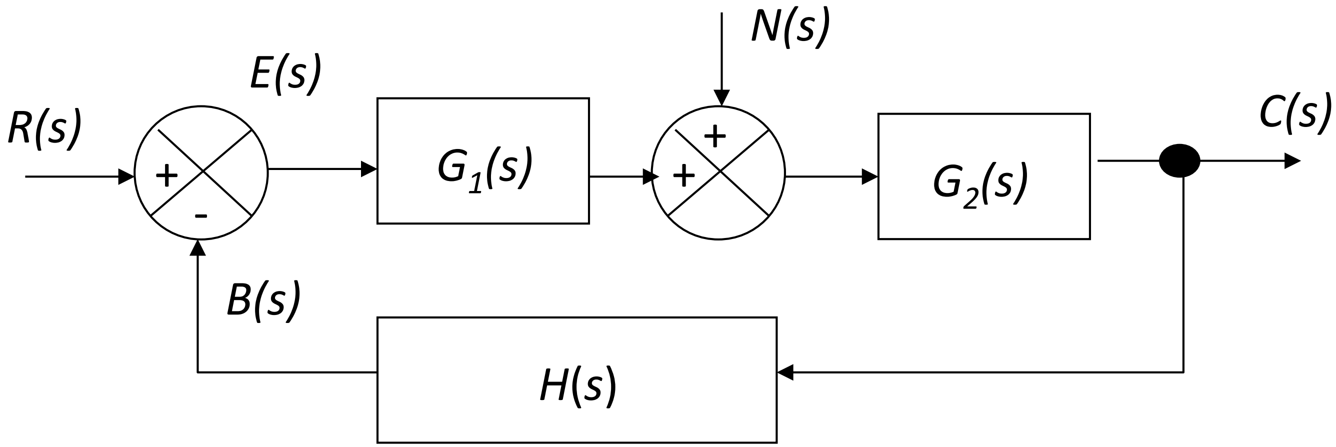
$$\frac{C_R(s)}{R(s)} = \frac{G_1(s)G_2(s)}{1 + G_1(s)G_2(s)H(s)}$$

Transfer function for signal

$$\frac{C_N(s)}{N(s)} = \frac{G_2(s)}{1 + G_1(s)G_2(s)H(s)}$$

Transfer function for disturbance

$$C(s) = C_N(s) + C_R(s) = \frac{C_N(s)}{R(s)} = \frac{G_2(s)}{1 + G_1(s)G_2(s)H(s)}\left[G_1(s)R(s) + N(s)\right]$$



Case 1: large open loop gain

$$|G_1(s)H(s)| >> 1 \quad \& \quad |G_1(s)G_2(s)H(s)| >> 1$$

$$\Rightarrow C_N(s)/N(s) \to 0$$

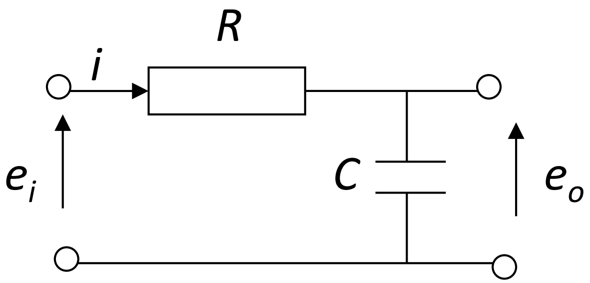
Case 2: Insensitivity to $G_1(s)$, $G_2(s)$

$$C_R(s)/R(s) \to \frac{1}{H(s)}$$

Case 3: Output follows input

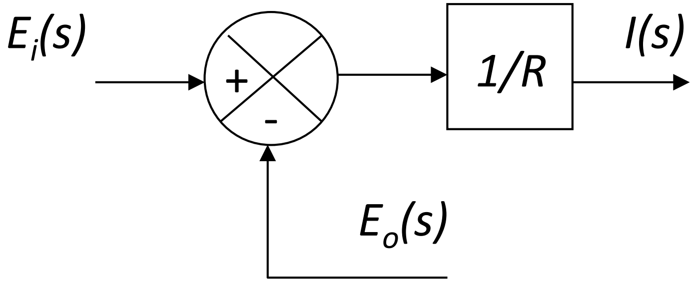
$$H(s) = 1 \Rightarrow C_R(s) \to R(s)$$

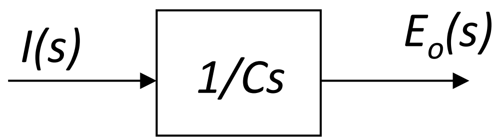
Drawing a block diagram for a real system

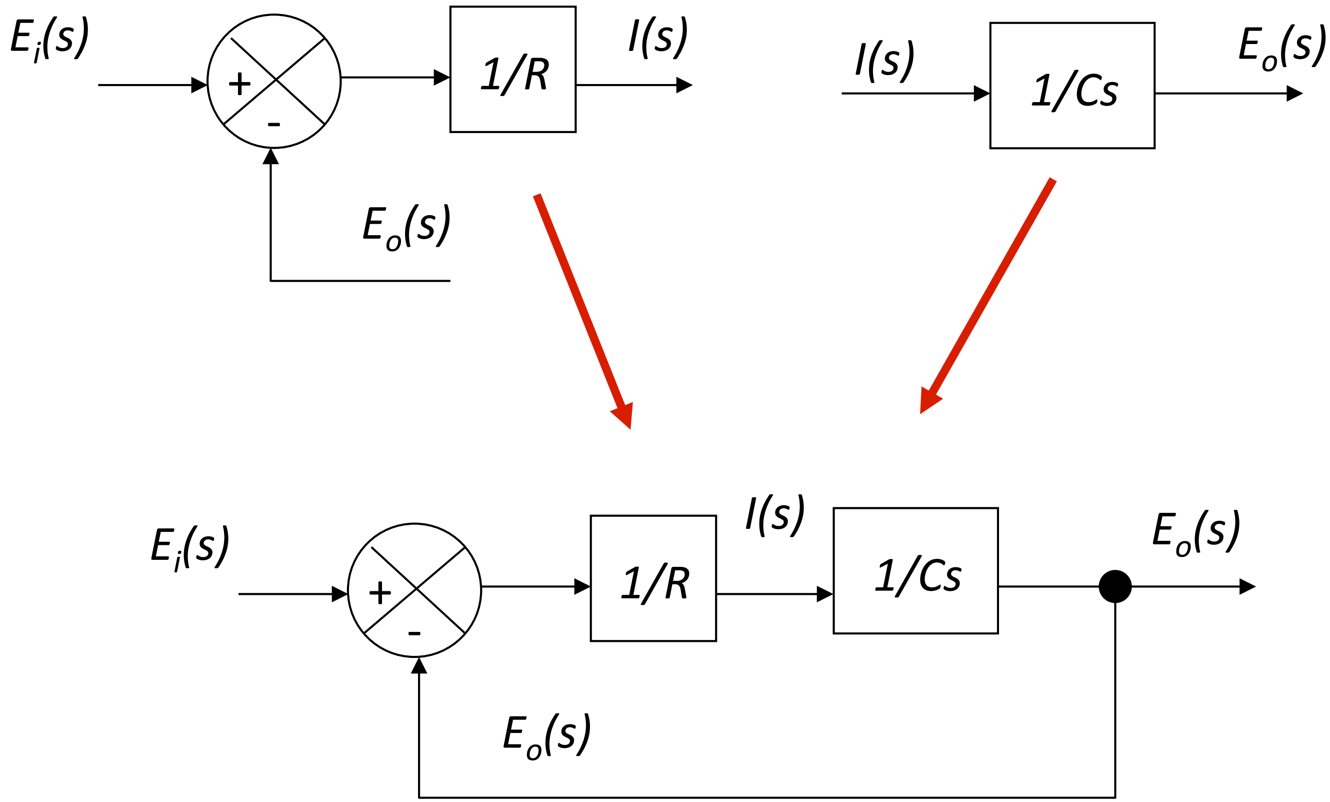


$$i = \frac{e_i - e_0}{R}; \qquad e_o = \frac{\int i \cdot \mathrm{d}t}{C}$$

$$\mathbf{L}(i) = I(s) = \frac{E_i(s) - E_o(s)}{R}; \qquad \mathbf{L}(e_o) = \frac{1}{C} \cdot \frac{I(s)}{s}$$







$E_i(s)$
+
-
$1/R$
$I(s)$
$E_o(s)$
$I(s)$
$1/Cs$
$E_o(s)$
$E_i(s)$
+
-
$1/R$
$I(s)$
$1/Cs$
$E_o(s)$
$E_o(s)$

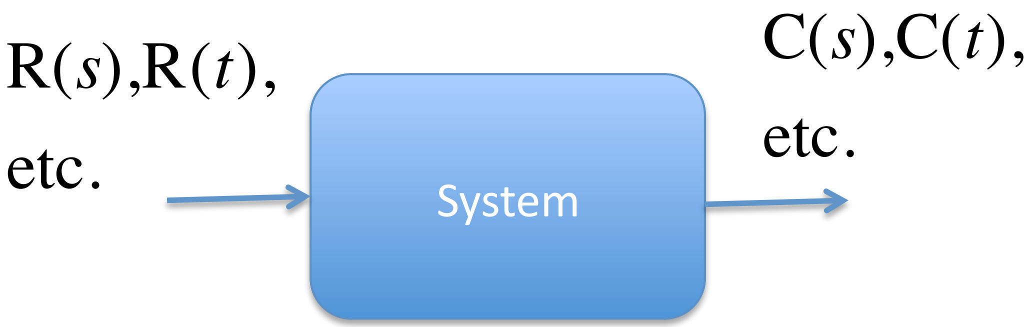
Rules for simplifying block diagrams

- Blocks can be connected in series only if the output of a block is not affected by connecting a following block (no-loading)

- The product of transfer functions in the feedforward direction must be the same

- The product of transfer functions around the loop must be the same

System response



$$C(s) = G(s)R(s)$$

The stimulus r(*t*) can be:

$$r(t) = \begin{cases} a: & \text{constant} \\ l(t): & \text{step function} \\ \delta(t): & \text{impulse function} \\ \sin \omega t,\ \cos\omega t: & \text{sinusoid} \\ \text{Combinations of above} & \end{cases}$$

The Laplace transform gives us a general method for calculating the response of a linear time independent single input – single output system to a stimulus function. (SISO system)

MIMO systems and the space-state approach

- MIMO = multiple input, multiple output systems
  - Car engine
  - Aircraft autopilot
- Can be linear or non-linear

- For a dynamic system the **state** is defined by the smallest set of variables that if know at $t = t_0$ define the state at $t>t_0$
- **State variables** are the set of variables defining the state.
- **State vector** is the vector with components that are th state variables.
- A hyperspace with dimensions corresponding the number of state variables. Any state can be reperesented by a point in this **state space**.

State variable form of differential equations

- Newton's laws and free body systems generally can be described by differential equations with d^2x/dt^2 terms

- Differential equations can be expressed as sets of first order differential equations.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$$

- Here the column vector **x** is the state of the system and $\underline{u}$ is the inputs the the output, y is:

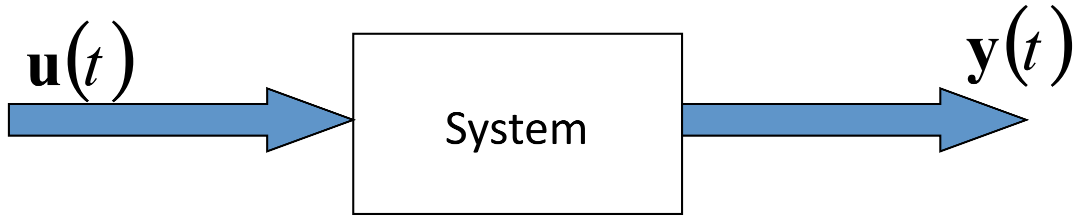
$$y = h(\mathbf{x}, u)$$

- The vector function **f** relates the state to its time derivative.
- For a linear system we have:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u; \quad y = \mathbf{H}x + Ju$$

- **F** is a $n \times n$ system matrix, **G** is *n×1 input matrix,* **H** *a 1×n* row output matrix and J is a scalar.

Space state representation forms



$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x},\mathbf{u},t)$$
$$\mathbf{y}(t) = \mathbf{H}(\mathbf{x},\mathbf{u},t)$$

General representation

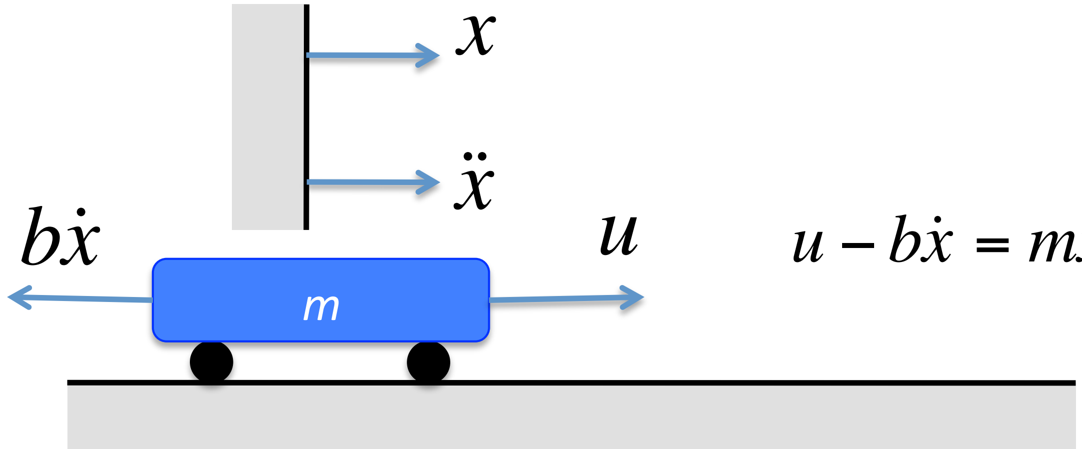
$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x},\mathbf{u})$$
$$\mathbf{y}(t) = \mathbf{H}(\mathbf{x},\mathbf{u})$$

Time invariant

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{u}(t)$$
$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + J\mathbf{u}(t)$$

Linear time invariant system

Automobile cruise control



$$u - b\dot{x} = m\ddot{x}$$

•Define the position and velocity of the car to be the state variables x_1 and x_2

$$\dot{x}_1 = x_2; \quad \dot{x}_2 = -\frac{b}{m}x_2 + \frac{1}{m}u$$

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

•If the output is the position of the car $y = x_1 = x$

$$y = \mathbf{H}\mathbf{x} + Ju; \quad J = 0$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 0 \cdot u$$

After Franklin, Powel, Emanai-Naeini

Automobile cruise control

•If the output is the velocity of the car $v = x_2$

$$y = \mathbf{H}\mathbf{x} + Ju; \quad J = 0$$

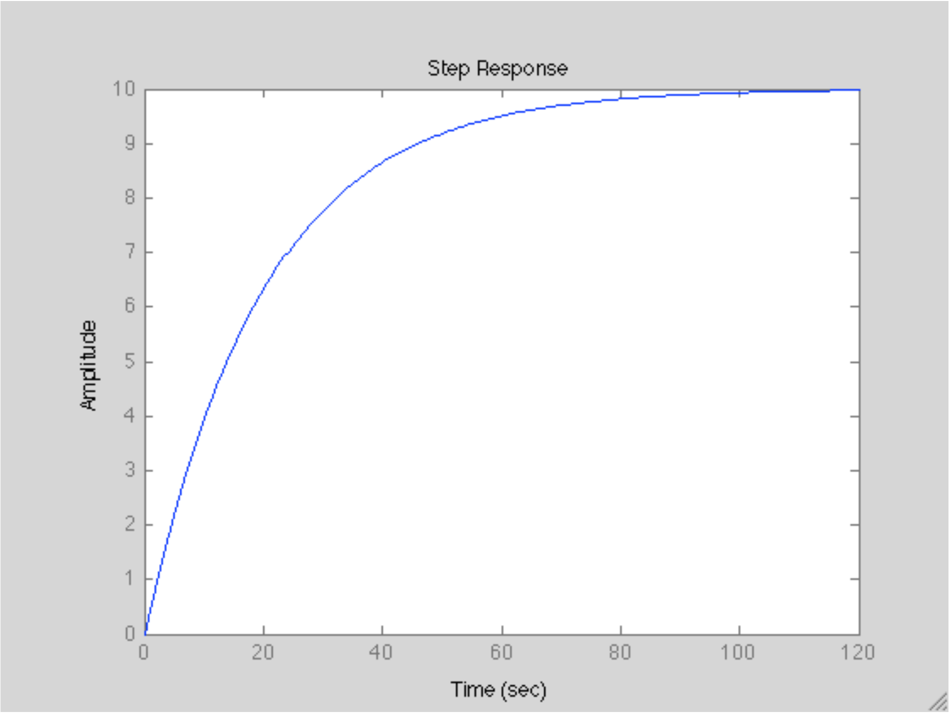
$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 0 \cdot u$$

•Taking a mass of 1000 kg and a constant drag of 500 N

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{m} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -0.05 \end{bmatrix}; \quad \mathbf{G} = \begin{bmatrix} 1 \\ 1/m \end{bmatrix} = \begin{bmatrix} 1 \\ 0.001 \end{bmatrix}; \quad \mathbf{H} = \begin{bmatrix} 0 & 1 \end{bmatrix}; \quad J = 0$$

Matlab commands for 500 N input step

```
F=[0,1;0,-0.05]
G=[0;0.001]
H=[0,1]
J = 0.
step(F,500*G,H,J)
```



After Franklin, Powel, Emanai-Naeini

Obtaining the transfer function from state variables

In Matlab a linear system may be defined in state space (ss) form in terms of **F**,**G**,**H**,*J*

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u; \quad y = \mathbf{H}x + Ju$$

Or polynomial ratio (tf)

$$H(s) = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-2}s^2 + b_{n-1}s + b_n}{a_0 s^m + a_1 s^{m-1} + \cdots + a_{m-2}s^2 + a_{m-1}s + a_m}$$

Or factored zero-pole form (zp)

$$H(s) = K\frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)}$$

Conversion in Matlab

State space to polynomial ratio

```
>> [num,den]=ss2tf(F,G,H,J)
num =
       0    0.0010         0
den =
    1.0000    0.0500         0
```

$$\Rightarrow H(s) = \frac{0.001}{s^2 + 0.05s + 0}$$

Polynomial ratio to factored z-p

```
[z,p,k] = tf2zp(num,den)
z =
    0
p =
       0
   -0.0500
k =
    0.0010
```

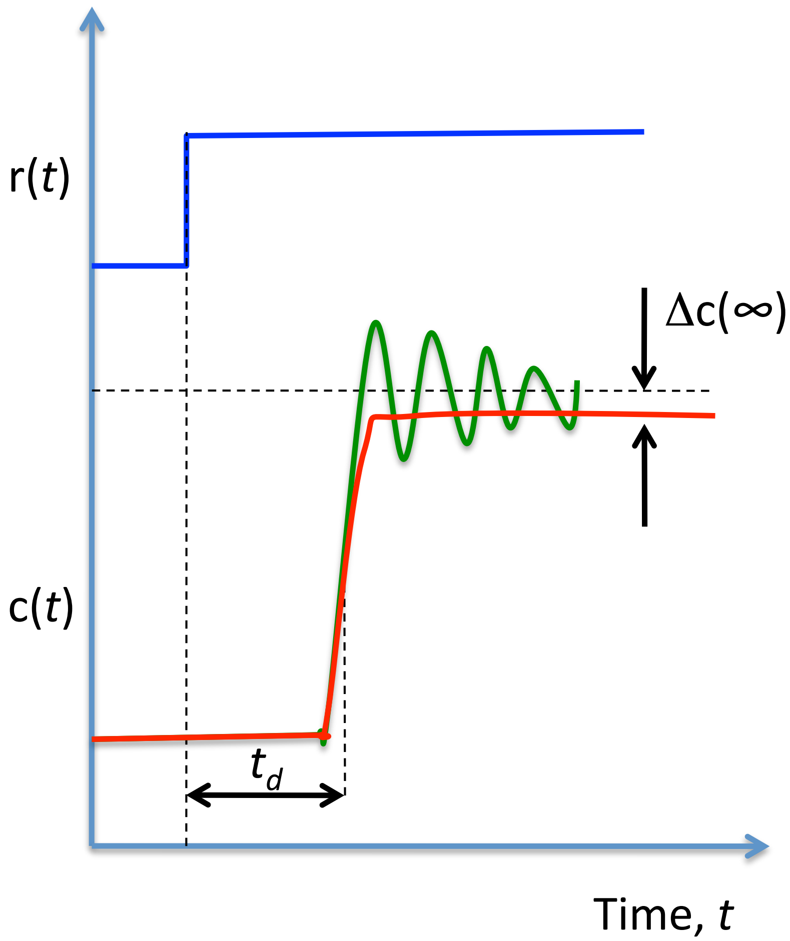
$$\Rightarrow H(s) = 0.001\frac{1}{(s + 0.05)s}$$

TRANSIENT RESPONSE ANALYSIS

Transients

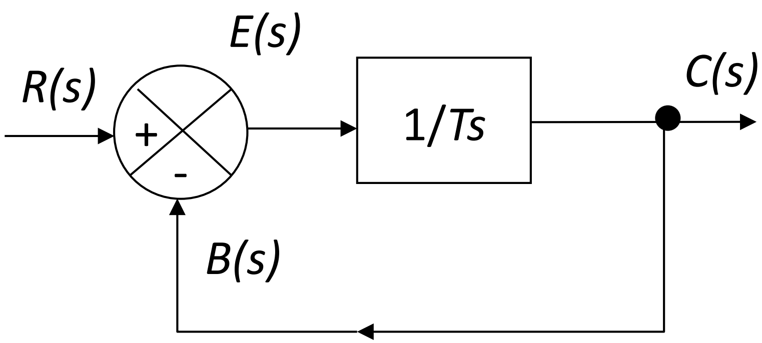
- Transients in systems originate from many kinds of real signals.
  - Driving a car over a step e.g. kerbstone
  - Abrupt change of oven temperature setting
  - “Typical” electric test signals
- Transient response is in two parts
  - The transient *per sec*
    - Transition from initial to final state
    - Delay
  - The steady state response
    - Stable or unstable?
    - Oscillations?
    - Steady-state error

Response to a step transient



Ist order system (CR circuit, thermal system etc)

For a first order system



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

$$\Rightarrow C(s) = \left(\frac{1}{Ts+1}\right)R(s)$$

The Laplace transform of a step function l(t) is:

$$l(t) = \begin{cases} 0; & t < 0 \\ 1; & t \geq 0 \end{cases}$$

$$\mathcal{L}\left[l(t)\right] = \frac{1}{s}$$

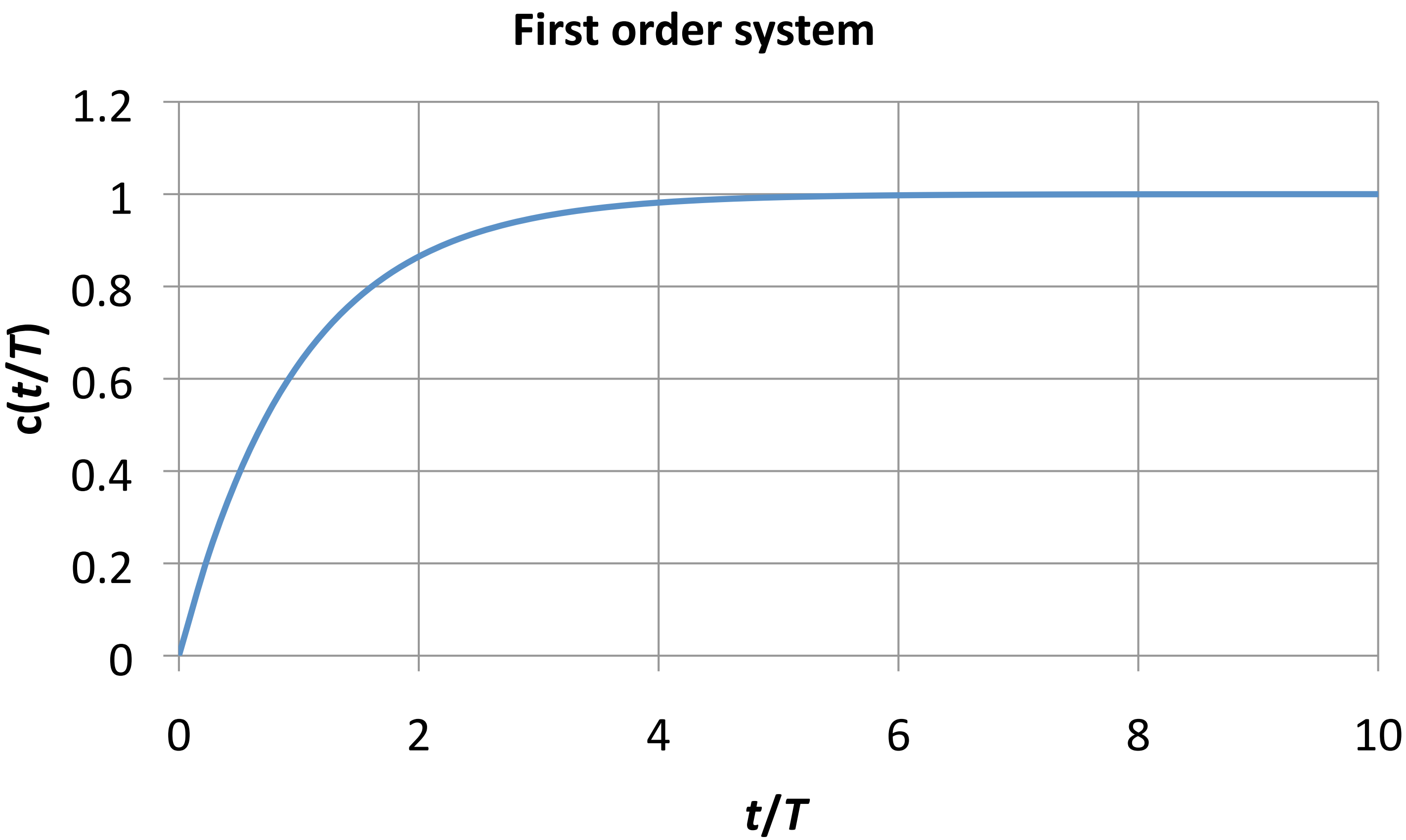
Then using algebra

$$\Rightarrow C(s) = \left(\frac{1}{Ts+1}\right)\left(\frac{1}{s}\right) = \left(\frac{1}{s+(1/T)}\right)$$

Finally take the inverse transform to get the response function wrt time.

$$c(t) = \mathcal{L}^{-1}\left[\frac{1}{s+(1/T)}\right] = 1 - e^{-t/T}$$

Step-response of first order system



Step response in Matlab

- Take our first order output response C(*s*)

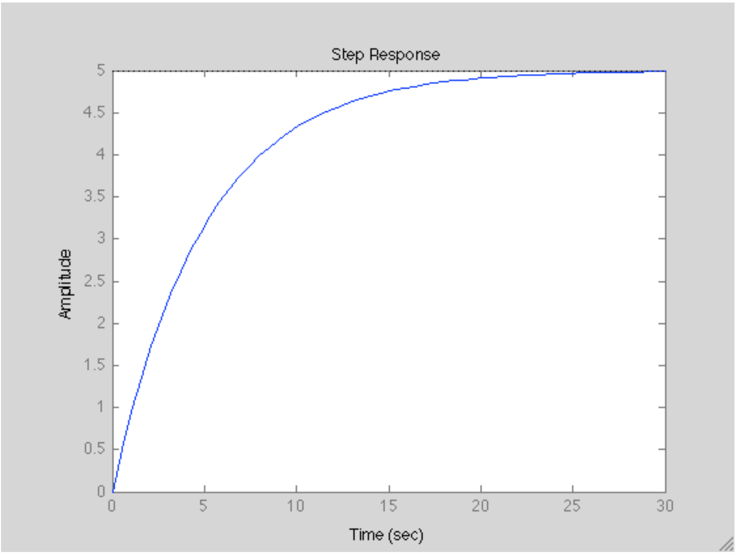
$$C(s) = \left(\frac{1}{s + (1/T)}\right)$$

- The coefficients in the numerator is 1 and 1 and 1/T in the denominator. Take $T = 5$

- Plot the step response

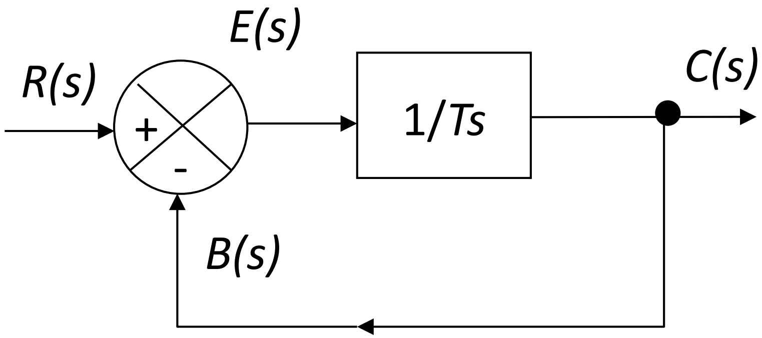
```
>> num =[1]
num =
    1
>> den=[1, 0.2]
den =
   1.0000    0.2000
```

```
step(num,den)
```



Unit-impulse response

For a first order system



$$G(s) = \frac{C(s)}{R(s)} = \frac{1}{Ts-1}$$

$$\Rightarrow C(s) = \left(\frac{1}{Ts-1}\right)R(s)$$

The Laplace transform of a unit impulse function $\delta(t)$ is:

$$\mathcal{L}\left[\delta(t)\right] = 1$$

Then using algebra

$$\Rightarrow C(s) = \left(\frac{1}{Ts-1}\right)(1) = \left(\frac{1}{Ts-1}\right)$$

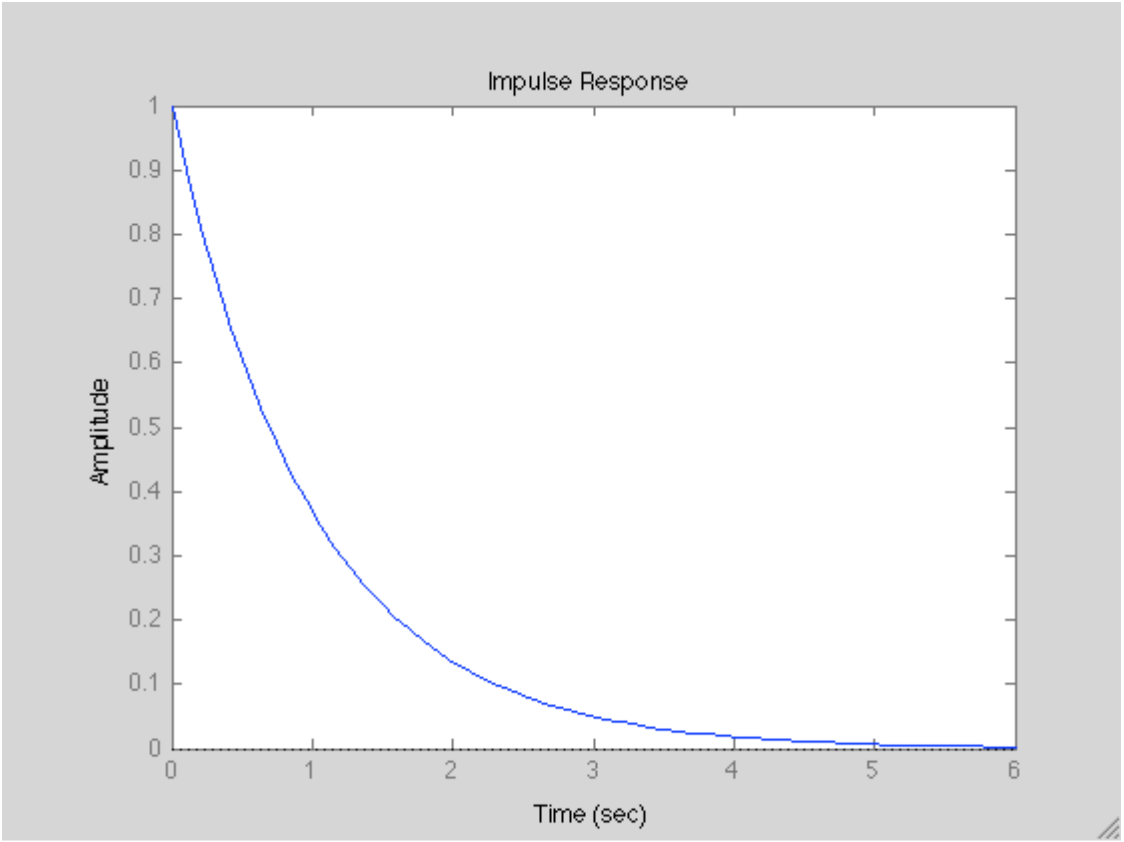
Finally take inverse transform to get the response function wrt time.

$$c(t) = \mathcal{L}^{-1}\left[\frac{1}{Ts-1}\right] = \frac{1}{T}e^{-t/T}$$

Unit impulse response in Matlab

$$C(s) = \left(\frac{1}{Ts-1}\right)$$

```
>> clear
>> num=[1]
num =
    1
>> den=[1,1]
den =
    1    1
>> impulse(num,den);
>>
```



Unit-ramp response

The response to a steadily rising signal is an important class of transient

$$r(t) = t$$

$$\mathcal{L}[r(t)] = \frac{1}{s^2}$$

Then for our first order system

$$C(s) = \left(\frac{1}{Ts-1}\right)R(s) = \left(\frac{1}{Ts-1}\right)\left(\frac{1}{s^2}\right)$$

$$\text{P.F.} \Rightarrow C(s) = \frac{1}{s^2} - \frac{T}{s} + \frac{T^2}{Ts-1}$$

Then taking inverse transforms

$$C(s) = \left(\frac{1}{Ts-1}\right)R(s) = \left(\frac{1}{Ts-1}\right)\left(\frac{1}{s^2}\right)$$

$$c(t) = \mathcal{L}^{-1}\left[\frac{1}{s^2} - \frac{T}{s} + \frac{T^2}{Ts-1}\right]$$

$$\Rightarrow c(t) = t - T + Te^{-t/T}$$

Unit ramp response in Matlab

Matlab does not have a unit ramp response

However, we can play a trick and transform the function so a step or impulse plotting function can be used.

$$C(s) = \left(\frac{1}{Ts-1}\right)R(s) = \left(\frac{1}{Ts-1}\right)\left(\frac{1}{s^2}\right)$$

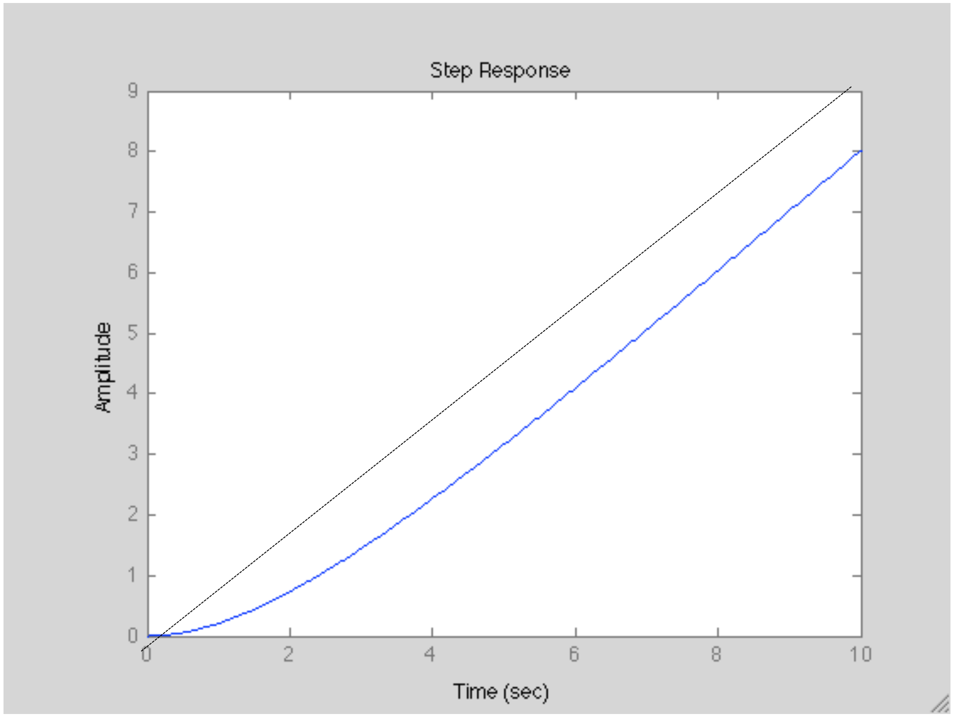
$$\mapsto C(s) = \left(\frac{1}{Ts^2-s}\right)\left(\frac{1}{s}\right)$$

Here:

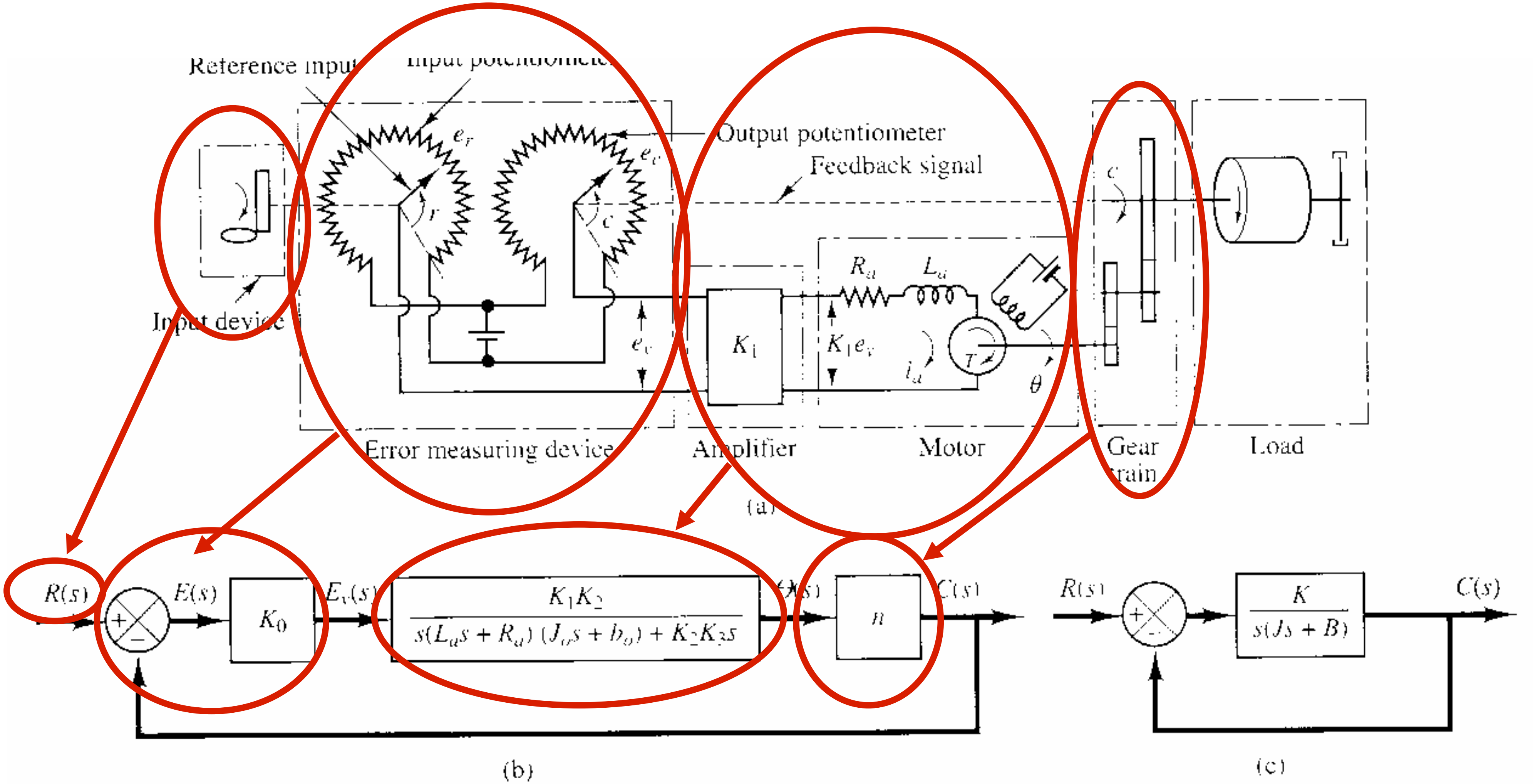
$$l(t) = \begin{cases} 0; & t < 0 \\ 1; & t \geq 0 \end{cases}$$

$$\mathcal{L}\left[l(t)\right] = \frac{1}{s}$$

```
>> num=[0 0 1]
num =
     0     0     1
>> den=[2 1 0]
den =
     2     1     0
>> step(num,den,10)
```

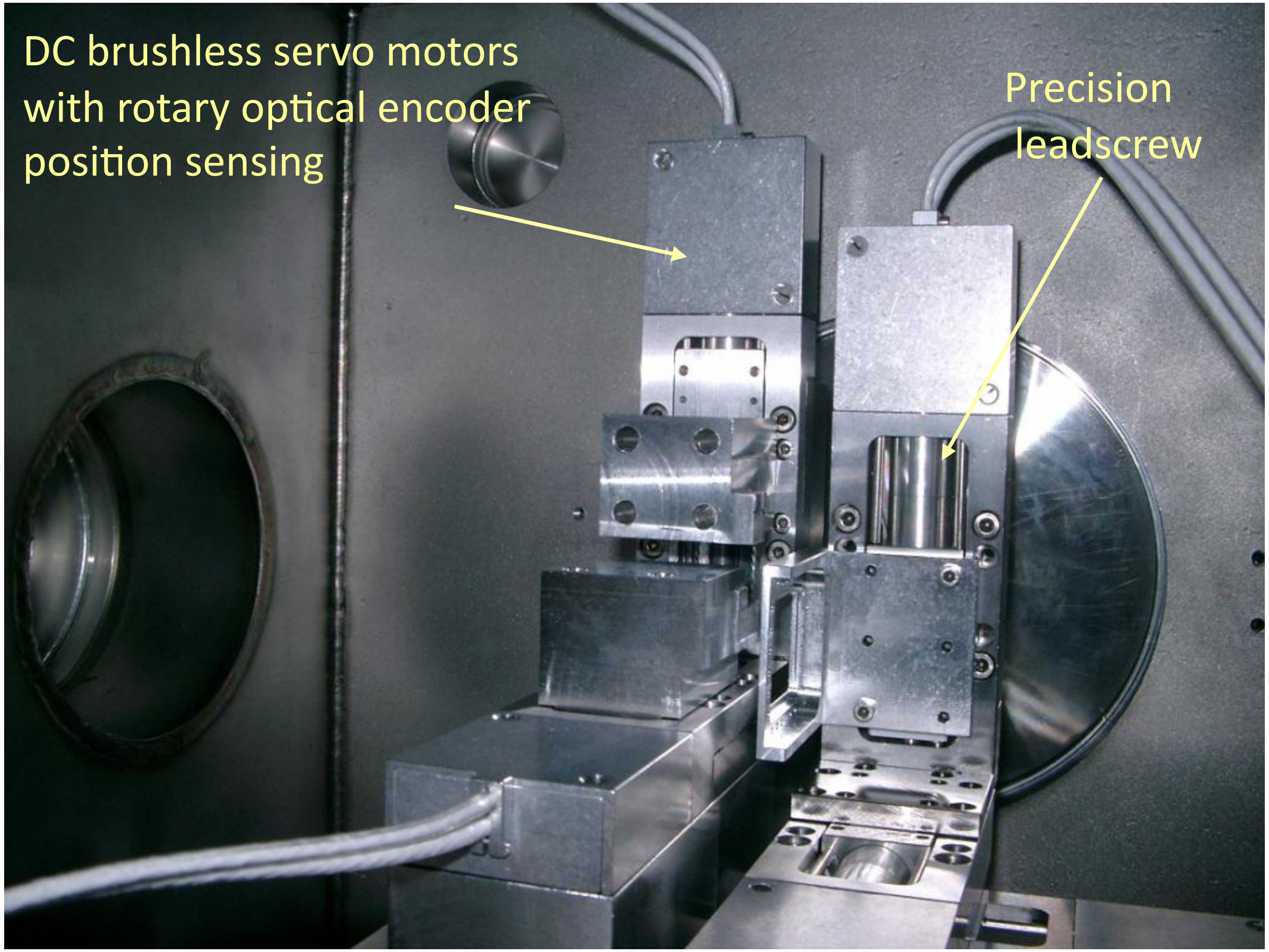


Second order systems: servo system

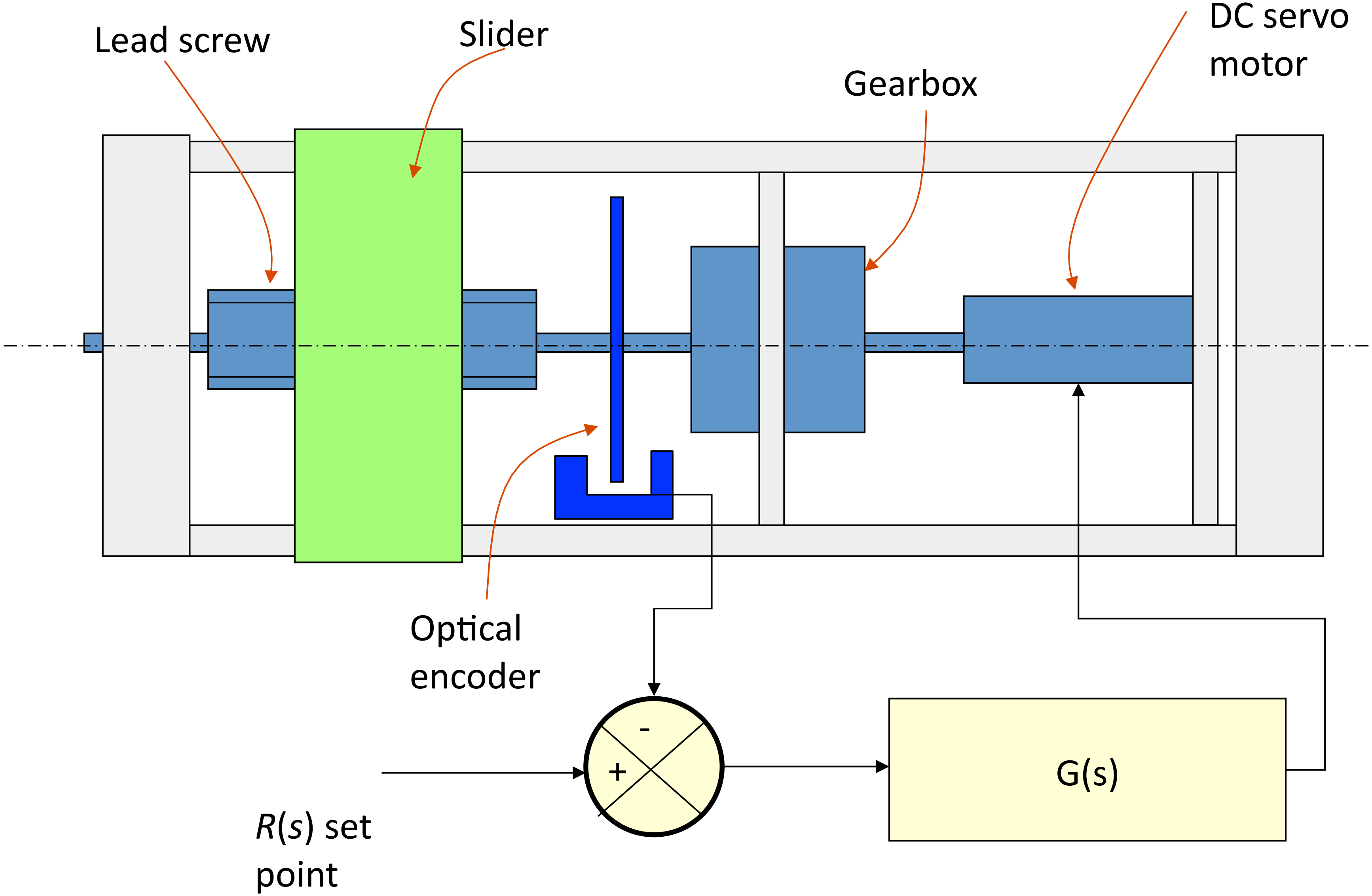


After: K. Ogata *Modern control engineering* 3rd. Ed (Prentice Hall, London, 1997)

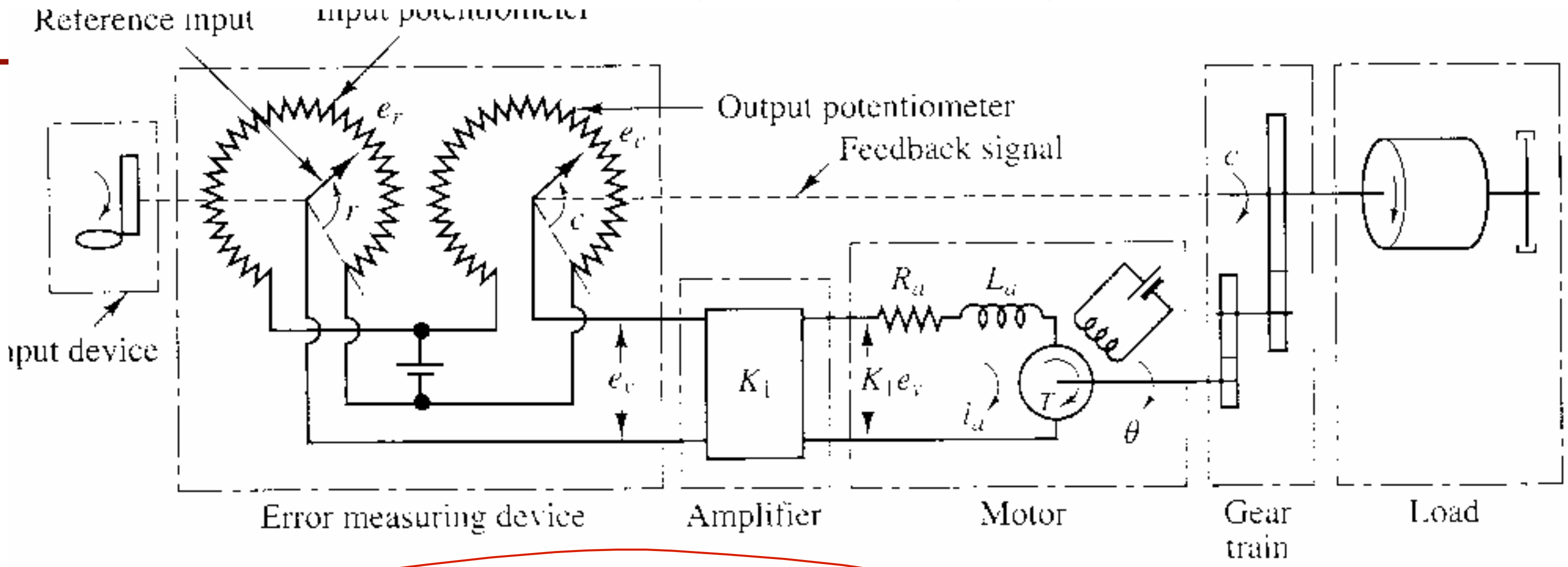
Servomotor controlled positioners



DC servo-motor positioners



From: K. Ogata *Modern control engineering* 3rd. Ed (Prentice Hall, London, 1997)



$$T = K_2 i_a \qquad e_b = K_3 \frac{\mathrm{d}\theta}{\mathrm{d}t}$$

Speed of motor amature governed by armature voltage

Armature inductance and resistance

$$e_a = K_1 e_v = L\dot{i}_a + R_a i_a + e_b$$

$$K_1 e_v = L\dot{i}_a + R_a i_a + K_3\dot{\theta}$$

Torque equilibrium condition $\quad J_0\ddot{\theta} + b\dot{\theta}_0 = T = K_2 i_a$

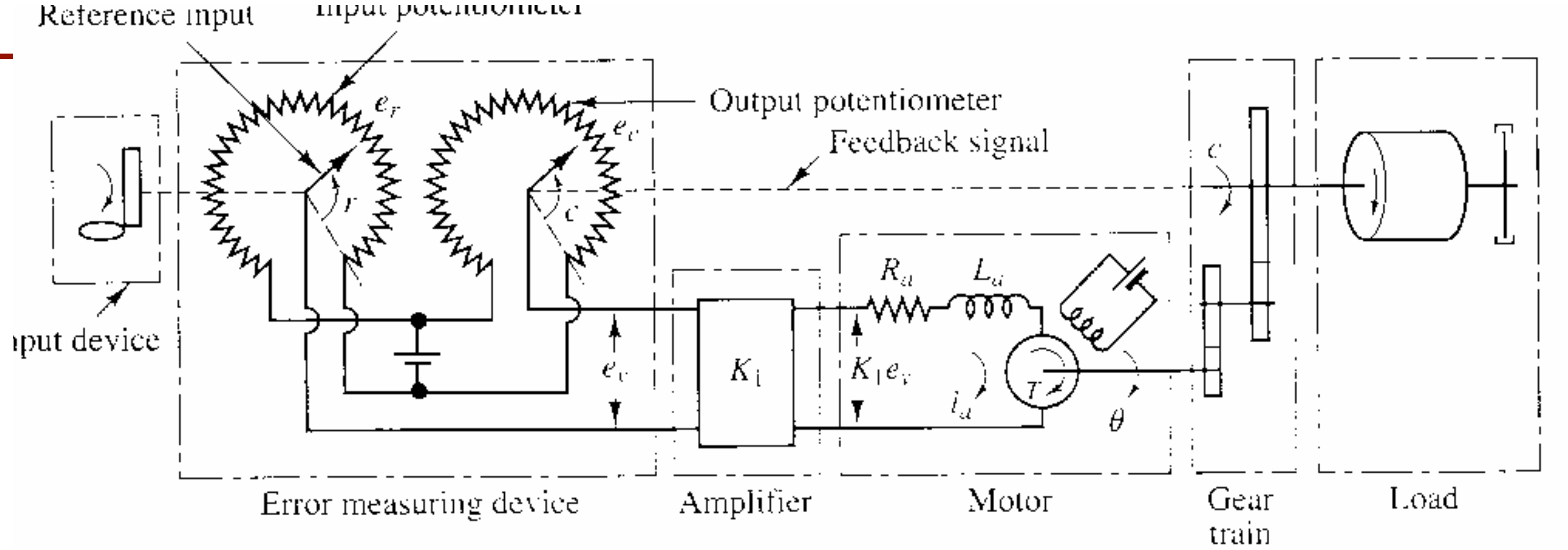
Inertia coeff

Viscous friction

$$\mathbf{L}[\theta(t)] = \Theta(s), \qquad E_v(s) = \mathbf{L}[e_v(t)]$$

$$\frac{\Theta(s)}{E_v(s)} = \frac{K_1 K_2}{s(L_a s + R_a)(J_0 s + b_0) + K_2 K_3 s}$$

From: K. Ogata *Modern control engineering* 3rd. Ed (Prentice Hall, London, 1997)



$$\frac{\Theta(s)}{E_v(s)} = \frac{K_1 K_2}{s(L_a s + R_a)(J_0 s + b_0) + K_2 K_3 s} \qquad C(s) = n\Theta(s)$$

$$E_v(s) = K_0[R(s) - C(s)] = K_0 E(s)$$

$$\Rightarrow G(s) = \frac{C(s)}{\Theta(s)} \cdot \frac{\Theta(s)}{E_v(s)} \cdot \frac{E_v(s)}{E(s)} = n \cdot \frac{K_1 K_2}{s(L_a s + R_a)(J_0 s + b_0) + K_2 K_3 s} \cdot K_0$$

$$L_a \to 0$$

$$\Rightarrow G(s) = \frac{nK_0 K_1 K_2}{s[R_a(J_0 s + b_0) + K_2 K_3]}$$

$$\Rightarrow G(s) = \frac{nK_0 K_1 K_2 / R_a}{J_0 s^2 + s\left(b_0 + \frac{K_2 K_3}{R_a}\right)}$$

Back emf increases effective viscous friction

$$\Rightarrow G(s) = \frac{nK_0K_1K_2/R_a}{J_0 s^2 + s\left(b_0 + \frac{K_2K_3}{R_a}\right)}$$

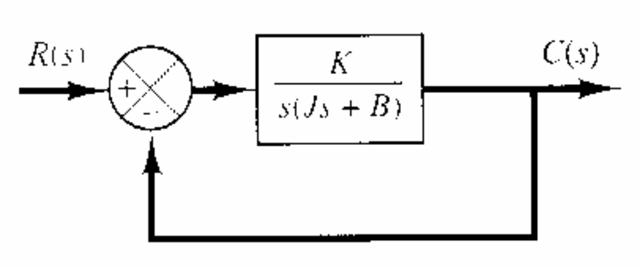
$$G(s) = \frac{K}{Js^2 + Bs}; \quad \mathrm{J} = \mathrm{J}_0/n^2; B = \left(b_0 + \frac{K_2K_3}{R_a}\right)\Big/n^2; \quad K = K_0K_1K_2/(nR_a)$$

Refered to output shaft

Or simplified further to:

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

$$K_m = \frac{K}{B}; \quad T_m = \frac{J}{B} = \frac{R_a J_0}{R_a b_0 + K_2K_3}$$



(c)

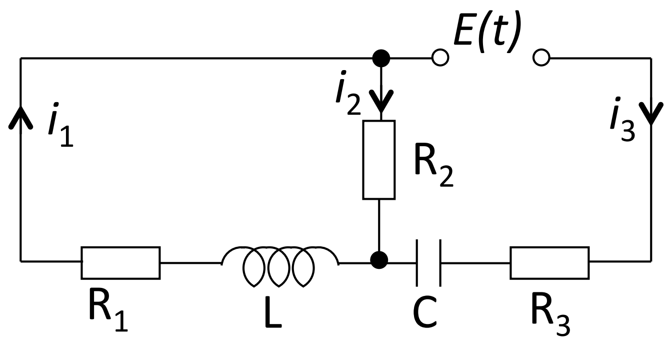
From: K. Ogata *Modern control engineering* 3rd. Ed (Prentice Hall, London, 1997)

(For today at least....)

THE END

DE solution by Laplace using Matlab-1

Use Kirchoff's and Ohm's Law to set up differential equations



$$\frac{dI_1}{dt} + \frac{R_2}{L}\frac{dQ}{dt} = \left(\frac{R_1 - R_2}{L}\right) I_1; \quad I_1(0) = I_0$$

$$\frac{dQ}{dt} = \left(\frac{1}{R_3 + R_2}\right)\left[E(t) - \frac{Q(t)}{C}\right] + \left(\frac{R_2}{R_3 + R_2}\right) I_1; \quad Q(0) = Q_0$$

```
syms R1 R2 R3 L C real
dI1 = sym('diff(I1(t),t)'); dQ = sym('diff(Q(t),t)');
I1 = sym('I1(t)'); Q = sym('Q(t)');
syms t s
E = sin(t);  % Voltage
eq1 = dI1 + R2*dQ/L - (R2 - R1)*I1/L;
eq2 = dQ - (E - Q/C)/(R2 + R3) - R2*I1/(R2 + R3);
```

Take Laplace transforms:

```
L1 = laplace(eq1,t,s)
L2 = laplace(eq2,t,s)
```

Giving:

```
L1 =
s*laplace(I1(t), t, s) - I1(0)
+ ((R1 - R2)*laplace(I1(t), t, s))/L
- (R2*(Q(0) - s*laplace(Q(t), t, s)))/L

L2 =
s*laplace(Q(t), t, s) - Q(0)
- (R2*laplace(I1(t), t, s))/(R2 + R3) - (C/(s^2
+ 1)
- laplace(Q(t), t, s))/(C*(R2 + R3))
```