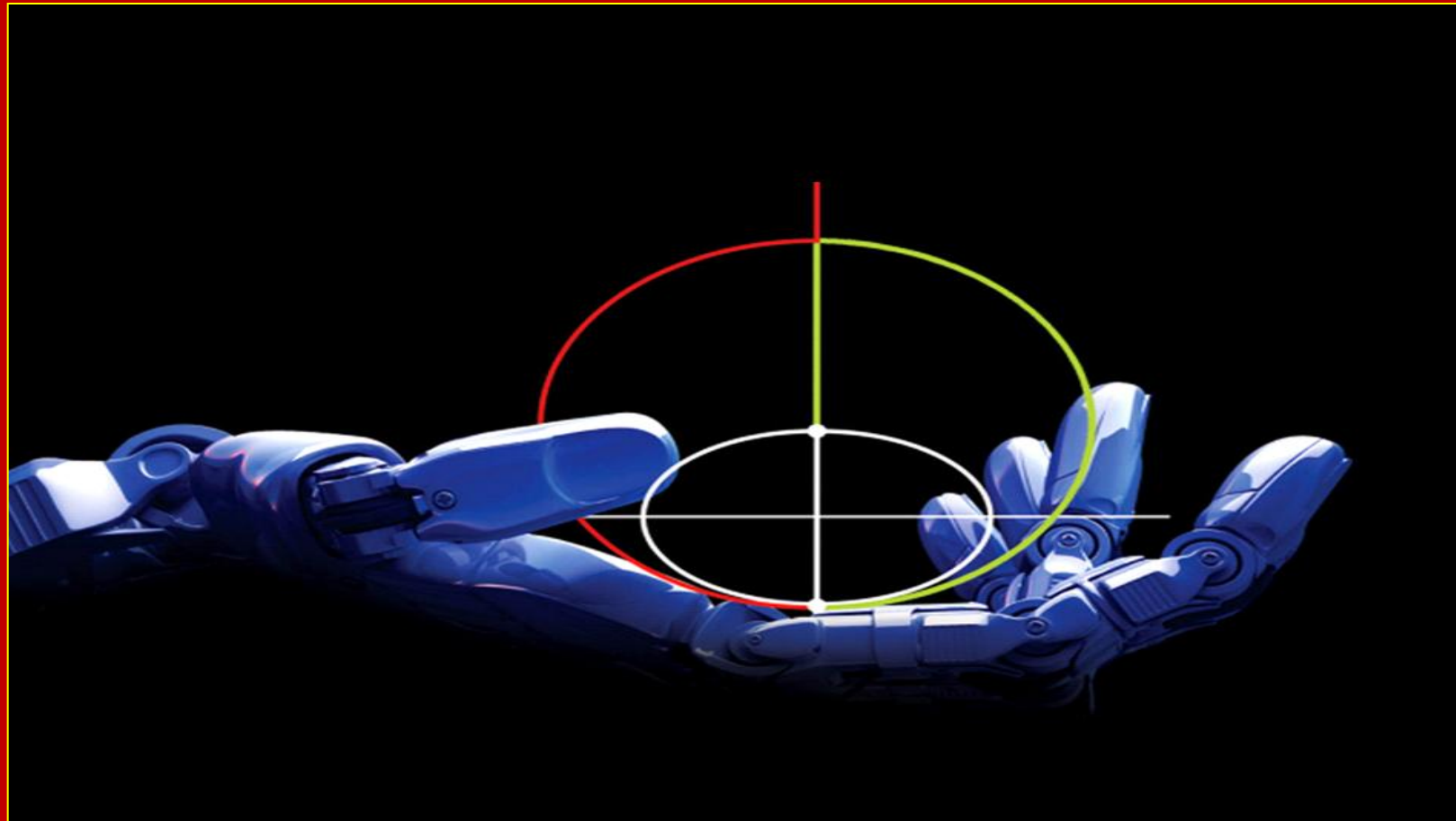




جَامِعَة
الْمَنَارَة

HAMARA UNIVERSITY

Introduction to Modeling and Simulation of Electromechanical Systems (DC MOTOR)



Contents

DC Motor Operating Principles

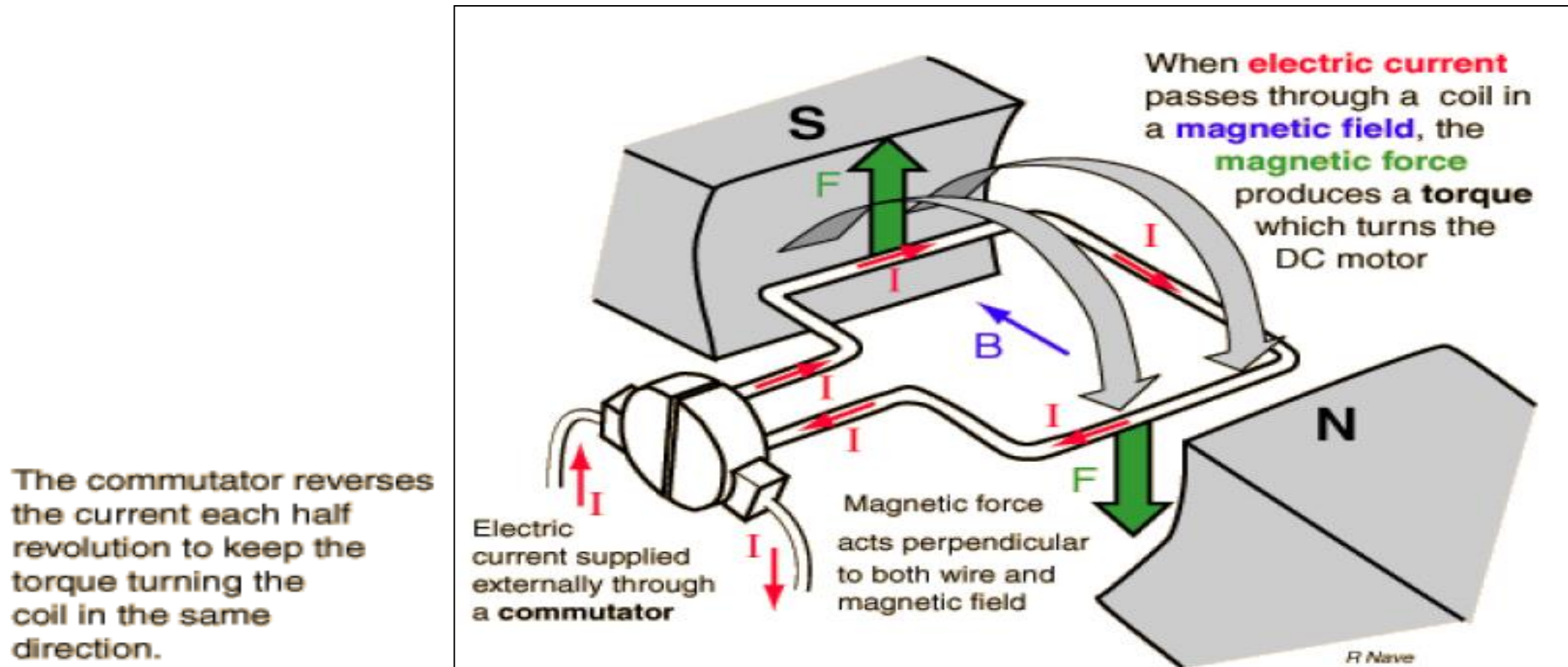
Building the Model using Simulink

Open-Loop Response

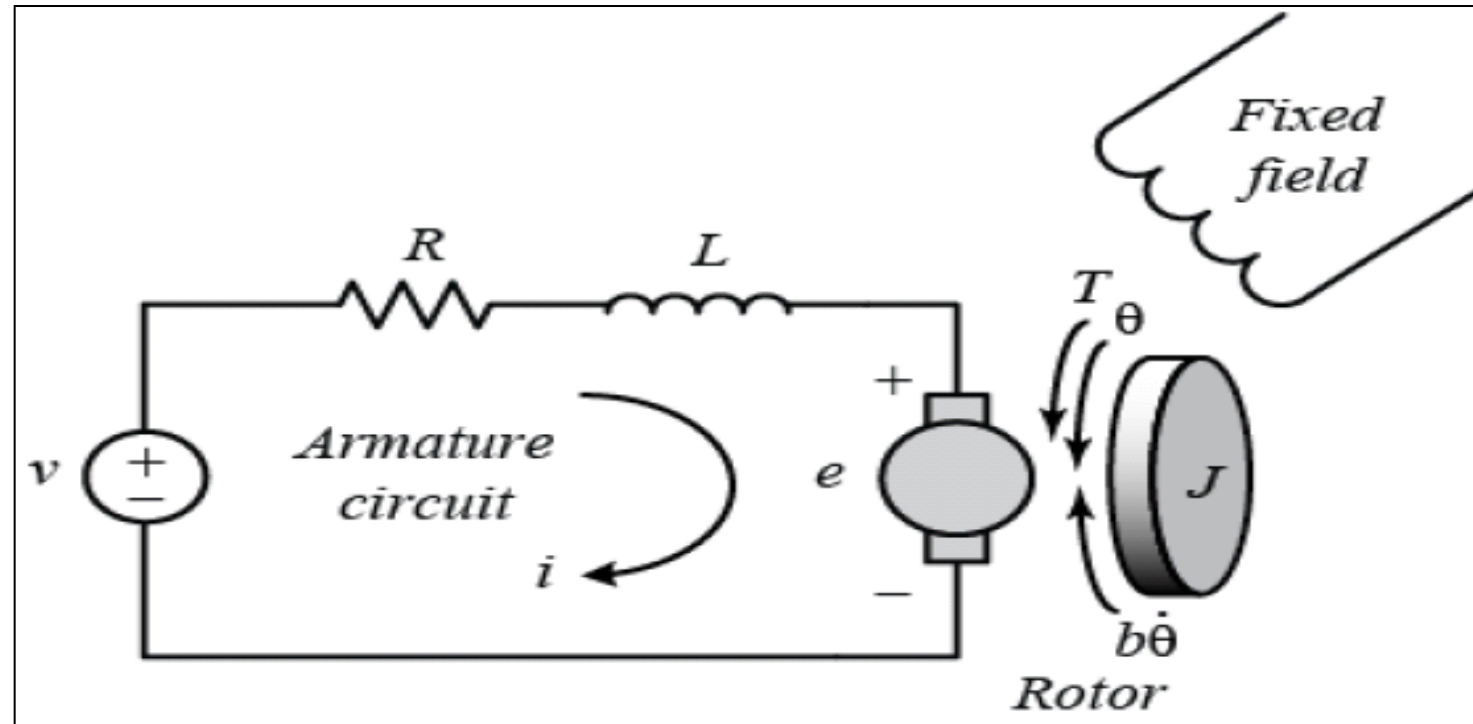
System Modeling using Transfer Function

DC Motor Operating Principles

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion.



The electric circuit of the armature and the free-body diagram of the rotor are shown in the following figure:



The input to the system is the voltage applied to the motor's armature (V), while the output is the angular position of the shaft (θ).

In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. In this example we will assume that the magnetic field is constant and, therefore, that the motor torque is proportional to only the armature current i by a constant factor K_t as shown in the equation below.

This is referred to as an armature-controlled motor. $T = K_t i$ (1)

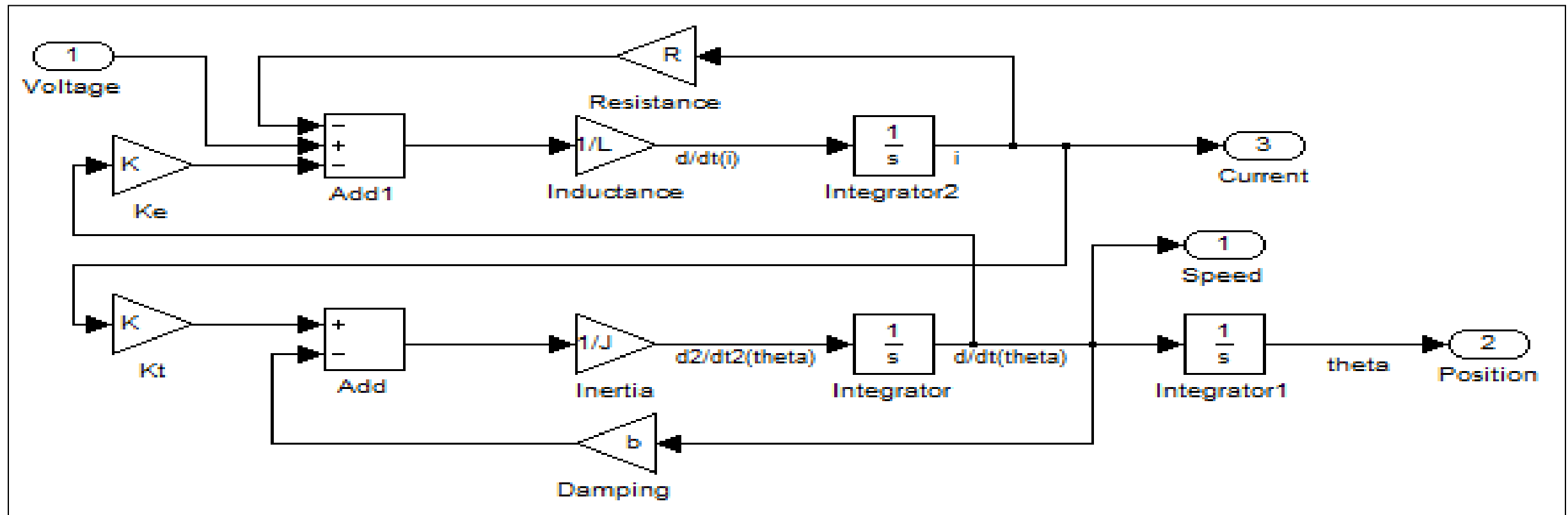
The back emf, e , is proportional to the angular velocity of the shaft by a constant factor K_e . $e = K_e \dot{\theta}$ (2)

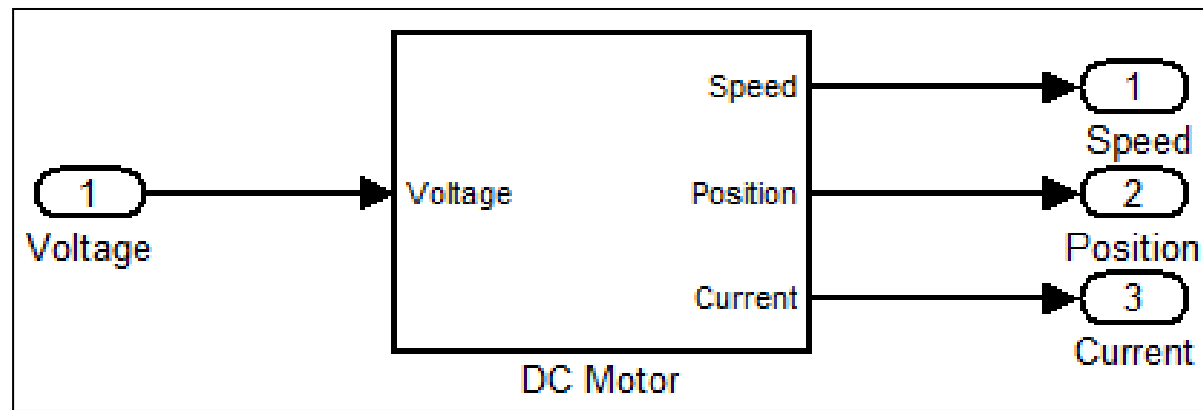
In SI units, the motor torque and back emf constants are equal, that is, $K_t = K_e$; therefore, we will use K to represent both the motor torque constant and the back emf constant.

Building the Model using Simulink

$$J \frac{d^2\theta}{dt^2} = T - b \frac{d\theta}{dt} \implies \frac{d^2\theta}{dt^2} = \frac{1}{J} \left(K_t i - b \frac{d\theta}{dt} \right)$$

$$L \frac{di}{dt} = -Ri + V - e \implies \frac{di}{dt} = \frac{1}{L} \left(-Ri + V - K_e \frac{d\theta}{dt} \right)$$

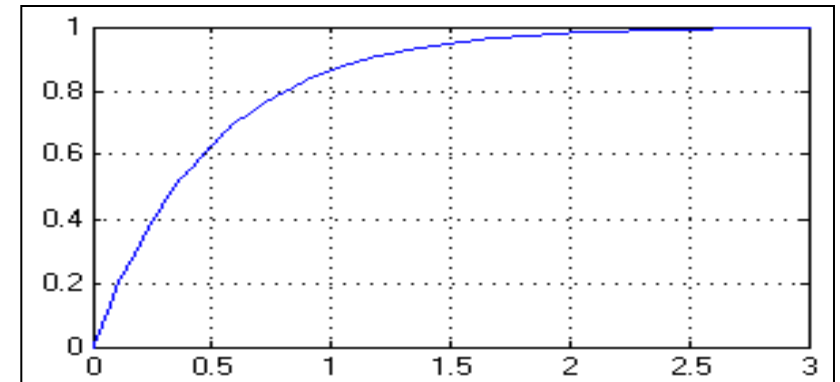
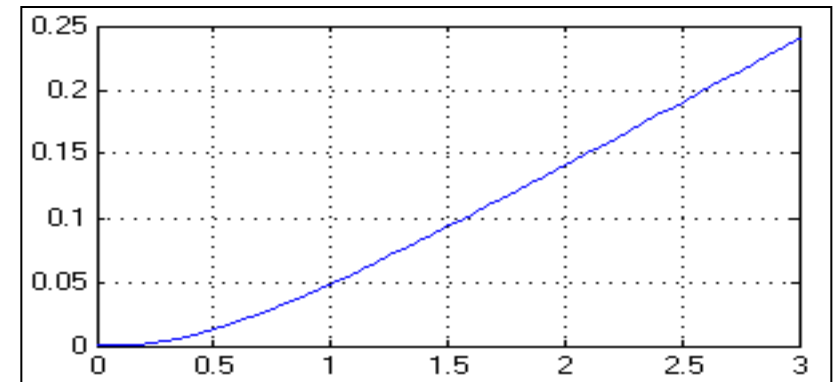
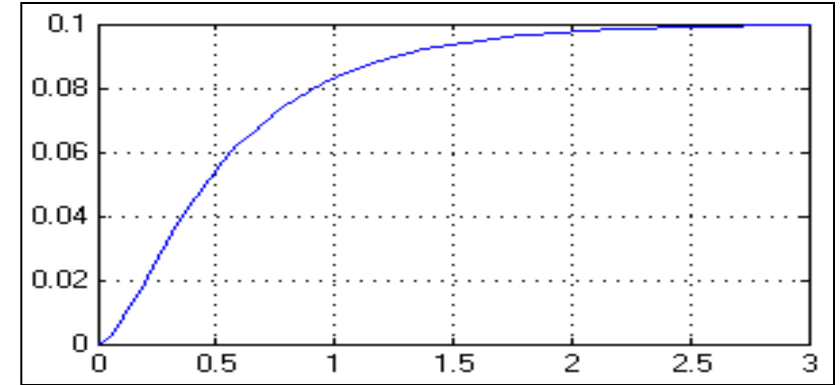
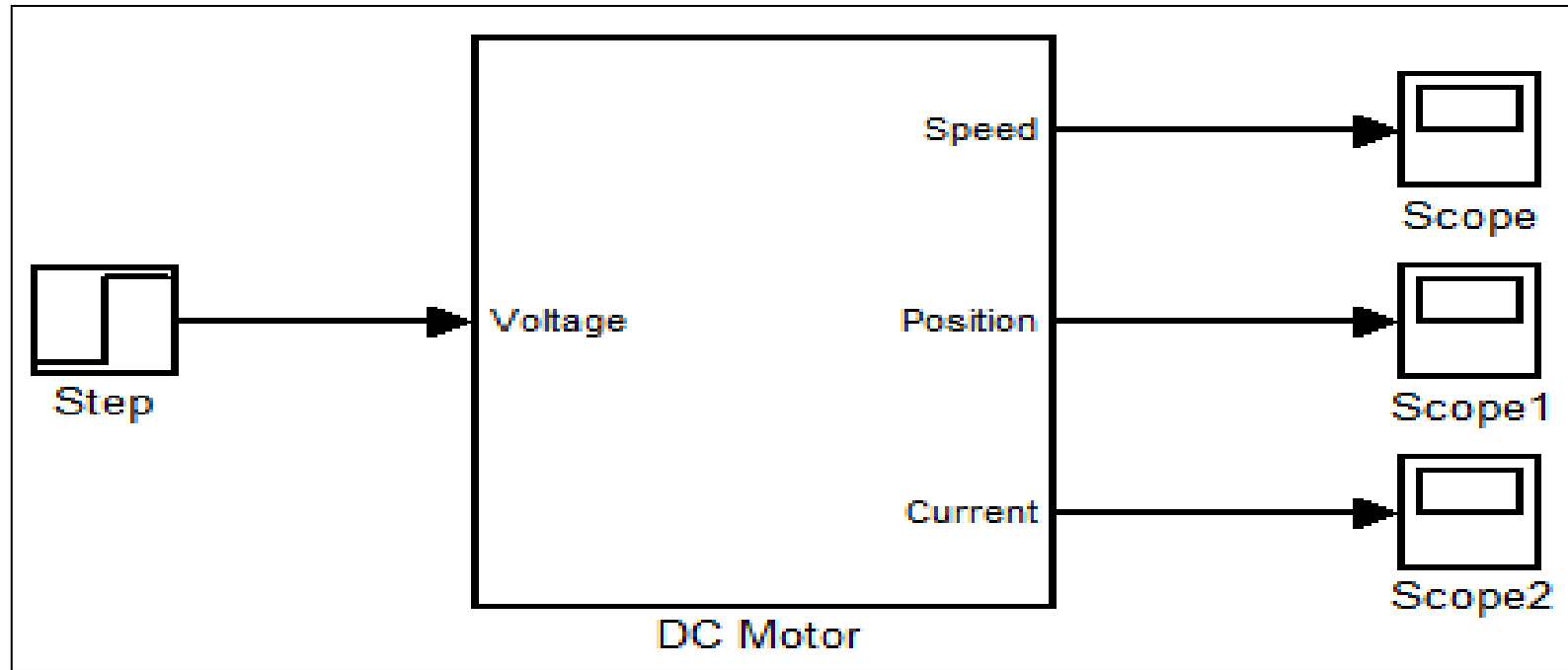




We now need to define the parameters used in our simulation. Type the following commands at the prompt of the MATLAB command window.

moment of inertia of the rotor	$J = 0.01 \text{ kg} \cdot \text{m}^2$
damping (friction) of the mechanical system	$b = 0.1 \text{ Nms}$
(back-)electromotive force constant	$K = 0.01 \text{ Nm/A}$
electric resistance	$R = 1 \Omega$
electric inductance	$L = 0.5 \text{ H}$

Open-Loop Response



System Modeling using Transfer Function

For this example, we will assume that the input of the system is the voltage source (V) applied to the motor's armature, while the output is the rotational speed of the shaft $d(\theta)/dt$. The rotor and shaft are assumed to be rigid. We further assume a viscous friction model, that is, the friction torque is proportional to shaft angular velocity.

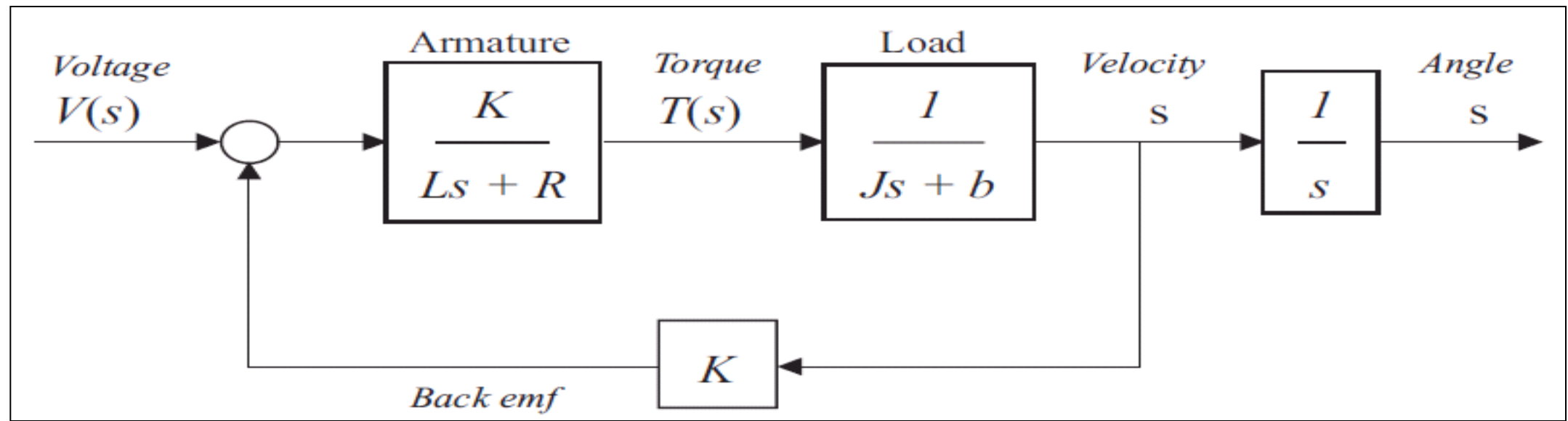
$$J \frac{d^2\theta}{dt^2} = T - b \frac{d\theta}{dt} \implies \frac{d^2\theta}{dt^2} = \frac{1}{J} (K_t i - b \frac{d\theta}{dt})$$

$$L \frac{di}{dt} = -Ri + V - e \implies \frac{di}{dt} = \frac{1}{L} (-Ri + V - K_e \frac{d\theta}{dt})$$

Applying the Laplace transform

$$I(s) = \frac{V(s) - K_s \theta(s)}{R + Ls}$$

$$Js^2\theta(s) + bs\theta(s) = K \frac{V(s) - K_s \theta(s)}{R + Ls}$$



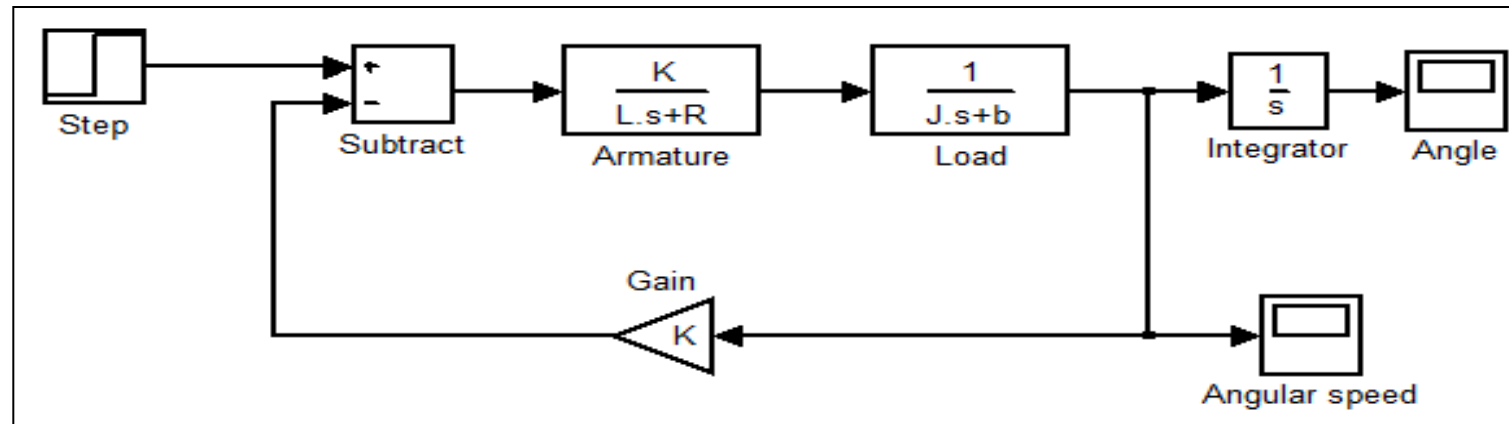
$$G_a(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s[(R + Ls)(Js + b) + K^2]}$$

$$G_v(s) = \frac{\omega(s)}{V(s)} = \frac{K}{(R + Ls)(Js + b) + K^2}$$

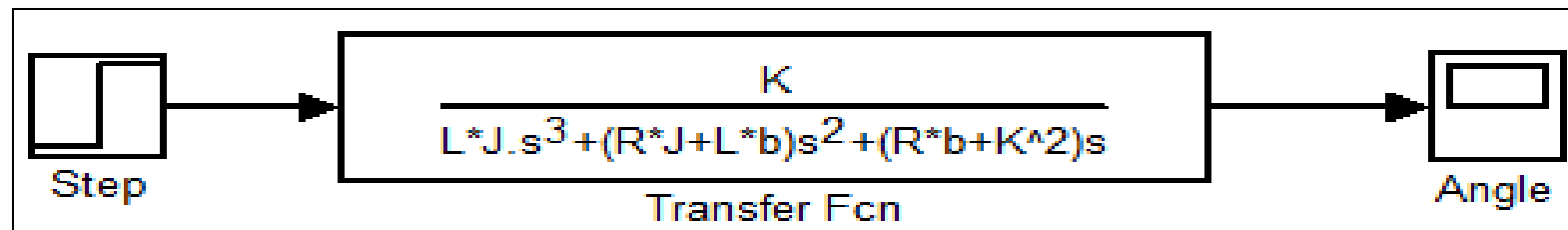
moment of inertia of the rotor	$J = 0.01 \text{ kg} \cdot \text{m}^2$
damping (friction) of the mechanical system	$b = 0.1 \text{ Nms}$
(back-)electromotive force constant	$K = 0.01 \text{ Nm/A}$
electric resistance	$R = 1 \Omega$
electric inductance	$L = 0.5 \text{ H}$

Simulink Models

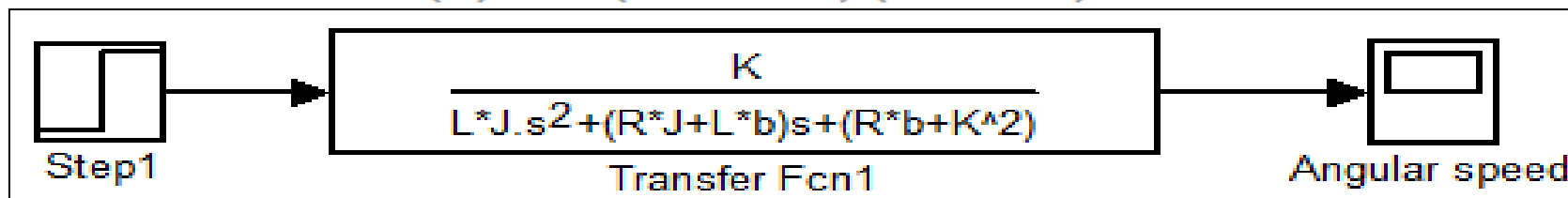
$$Js^2\theta(s) + bs\theta(s) = K \frac{V(s) - Ks\theta(s)}{R + Ls}$$



$$G_a(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s[(R + Ls)(Js + b) + K^2]}$$



$$G_v(s) = \frac{\omega(s)}{V(s)} = \frac{K}{(R + Ls)(Js + b) + K^2}$$



As $G_a(s)$ can be expressed as $G_v(s) \cdot \frac{1}{s}$, we can enter these two transfer functions separately and combine them in series:

```
J=0.01;      aux = tf(K,conv([L R],[J b]));
b=0.1;       Gv = feedback(aux,K)
K=0.01;      Ga = tf(1,[1 0])*Gv
R=1;         Gv =
L=0.5;          0.01
                -----
                0.005 s^2 + 0.06 s + 0.1001
                Continuous-time transfer function.

                Ga =
                0.01
                -----
                0.005 s^3 + 0.06 s^2 + 0.1001 s
                Continuous-time transfer function.
```

Here, we made use of the function `feedback` to create a feedback connection of two transfer functions and the multiplication operator

Instead of using convolution, the first of the above three commands can be replaced by the product of two transfer functions:

```
aux = tf(K,[L R])*tf(1,[J b]);
```

Another possibility (perhaps the most convenient one) is to define the transfer function in a symbolic way. First introduce a system representing the Laplace operator s (differentiator) and then enter the transfer function as an algebraic expression:

```
s = tf([1 0],1);  
Gv = K/((L*s + R)*(J*s + b) + K^2)  
Ga = Gv/s
```

```
Gv =  
      0.01  
-----  
0.005 s^2 + 0.06 s + 0.1001
```

Continuous-time transfer function.

```
Ga =  
      0.01  
-----  
0.005 s^3 + 0.06 s^2 + 0.1001 s
```

Continuous-time transfer function.

It is convenient to label the inputs and outputs by the names of the physical variables they represent:

```
Gv.InputName = 'Voltage';  
Gv.OutputName = 'Velocity'  
Ga.InputName = 'Voltage';  
Ga.OutputName = 'Angle'
```

Gv =

From input "Voltage" to output "Velocity":

0.01

0.005 s² + 0.06 s + 0.1001

Continuous-time transfer function.

Ga =

From input "Voltage" to output "Angle":

0.01

0.005 s³ + 0.06 s² + 0.1001 s

Continuous-time transfer function.

As we may want plot the responses for the velocity and angle in one figure, it convenient to group the two transfer functions into a single system with one input, the voltage, and two outputs, the velocity and the angle:

$$G = [G_v; G_a]$$

G =

From input "Voltage" to output...

0.01

Velocity: -----

$$0.005 s^2 + 0.06 s + 0.1001$$

0.01

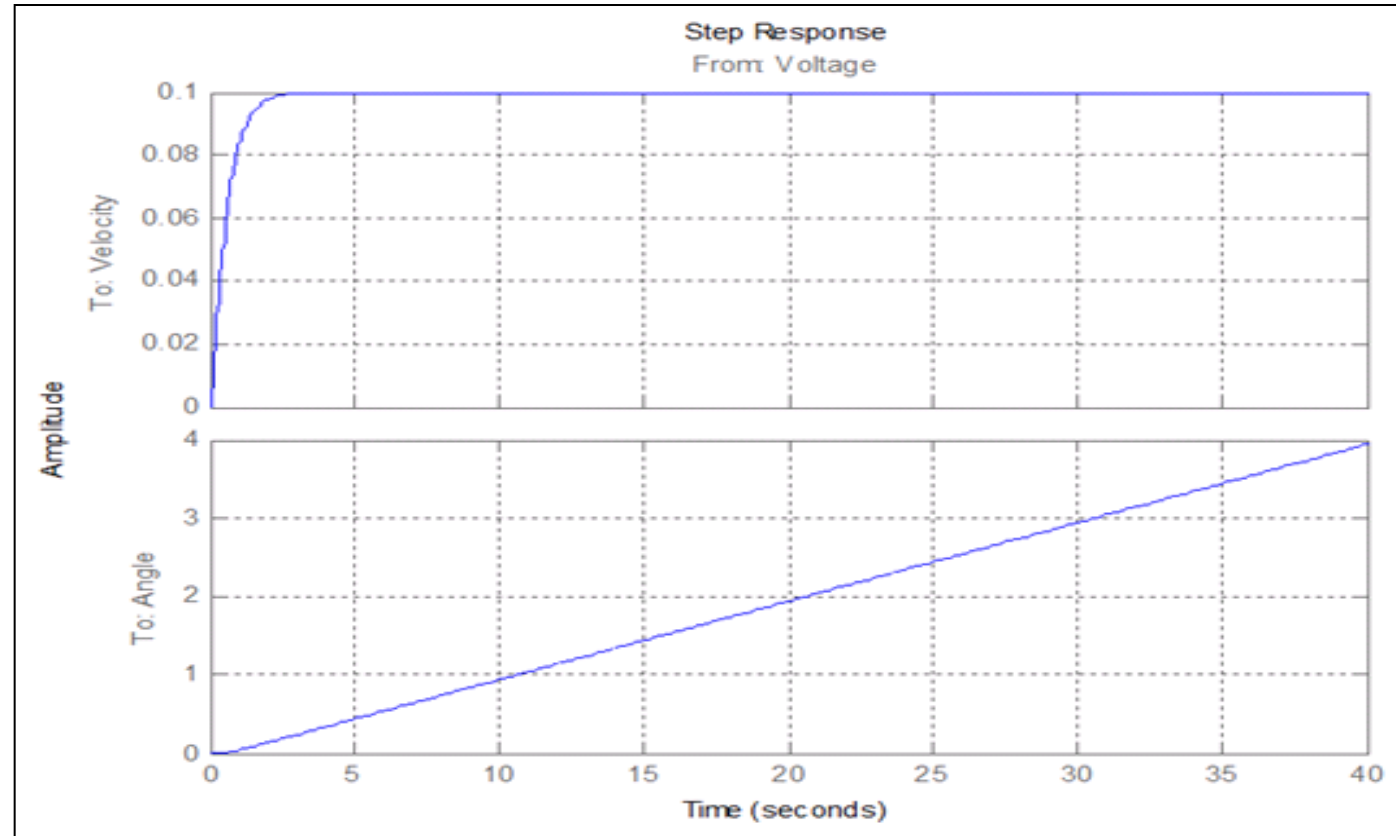
Angle: -----

$$0.005 s^3 + 0.06 s^2 + 0.1001 s$$

Continuous-time transfer function.

Now, we can plot the step of the motor model:

```
step(G);  
grid
```





جَامِعَة
الْمَنَارَة

HAMARA UNIVERSITY