System modeling and Simulation

Chapter 6. Electrical and Electromechanical Systems

6.5 Matlab Applications

邹渊
Yuan Zou
Tel: 68944115
Email: zouyuan@bit.edu.cn

group1	group2	group3	group4	group5

Matlab Applications

This section illustrates how to use the *lsim* and *step* functions with motor models in transfer function form and state-variable form.

Showing how to use the ode solvers to obtain the response of a nonlinear system.

- Step Response From Transfer Functions

$$\frac{I_a(s)}{V(s)} = \frac{Is + c}{L_a I s^2 + (R_a I + cL_a)s + cR_a + K_b K_T}$$

$$\frac{\Omega(s)}{V(s)} = \frac{K_T}{L_a I s^2 + (R_a I + cL_a)s + cR_a + K_b K_T}$$

the input is the armature voltage *v(t)*.

Im and cm represent the inertia and damping ofthe motor

The following program contains the motor parameters

```
% Program motor_par.m (Motor parameters in SI units)
global KT Kb La Ra Im cm
KT = 0.05;Kb = KT;
La = 2e-3;Ra = 0.5;
Im = 9e-5;cm = 1e-4;
```

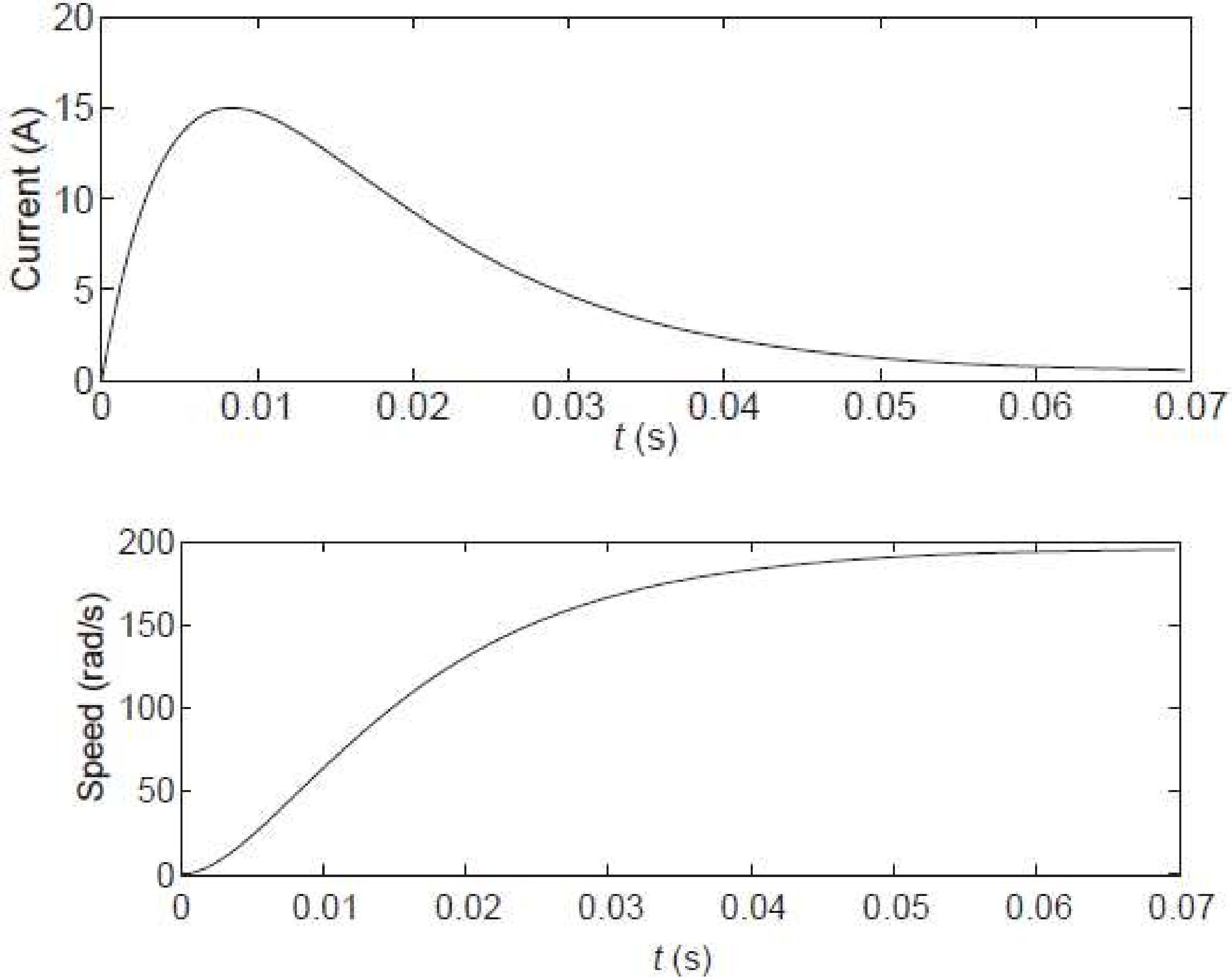
The following program creates the LTI models based on the motor transfer functions.

```
% Program motor_tf.m (Transfer functions for voltage input)
I = Im;
c = cm;
% current:
current = tf([I,c],[La*I,Ra*I+c*La,c*Ra+Kb*KT]);
% speed:
speed = tf(KT,[La*I,Ra*I+c*La,c*Ra+Kb*KT]);
```

The next program computes and plots the step response for an input of 10 V

```
% Program motor_step.m (Motor step response)
motor_par
motor_tf
[current, tc] = step(current);
[speed, ts] = step(speed);
subplot(2,1,1),plot(tc,10*current),...
   xlabel('t (s)'),ylabel('Current (A)')
subplot(2,1,2),plot(ts,10*speed),...
   xlabel('t (s)'),ylabel('Speed (rad/s)')
```

The result is shown in the figure



- Step Response From State-Variable Model

Now use the state variable model of the motor to plot the step response.

$$\frac{di_a}{dt} = \frac{v - i_a R_a - K_b \omega}{L_a}$$

$$\frac{d\omega}{dt} = \frac{K_T i_a - c\omega - T_d}{I}$$

The state variables are the armature current *ia* and the speed *ω;* The inputs are the applied voltage *v* and the load torque *TL (Td = TL/N)*

where the state vector and input vector are

$$\mathbf{x} = \begin{bmatrix} i_a \\ \omega \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} v \\ T_L \end{bmatrix}$$

The appropriate state and input matrices are

$$\mathbf{A} = \begin{bmatrix} -R_a/L_a & -K_b/L_a \\ K_T/I & -c/I \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1/L_a & 0 \\ 0 & -1/(NI) \end{bmatrix}$$

The output matrices are

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

The following program contains the parameter values for the load.

```
% load_par.m Load parameters.
IL = 0;
cL = 0;
N = 1;
TL = 0;
```

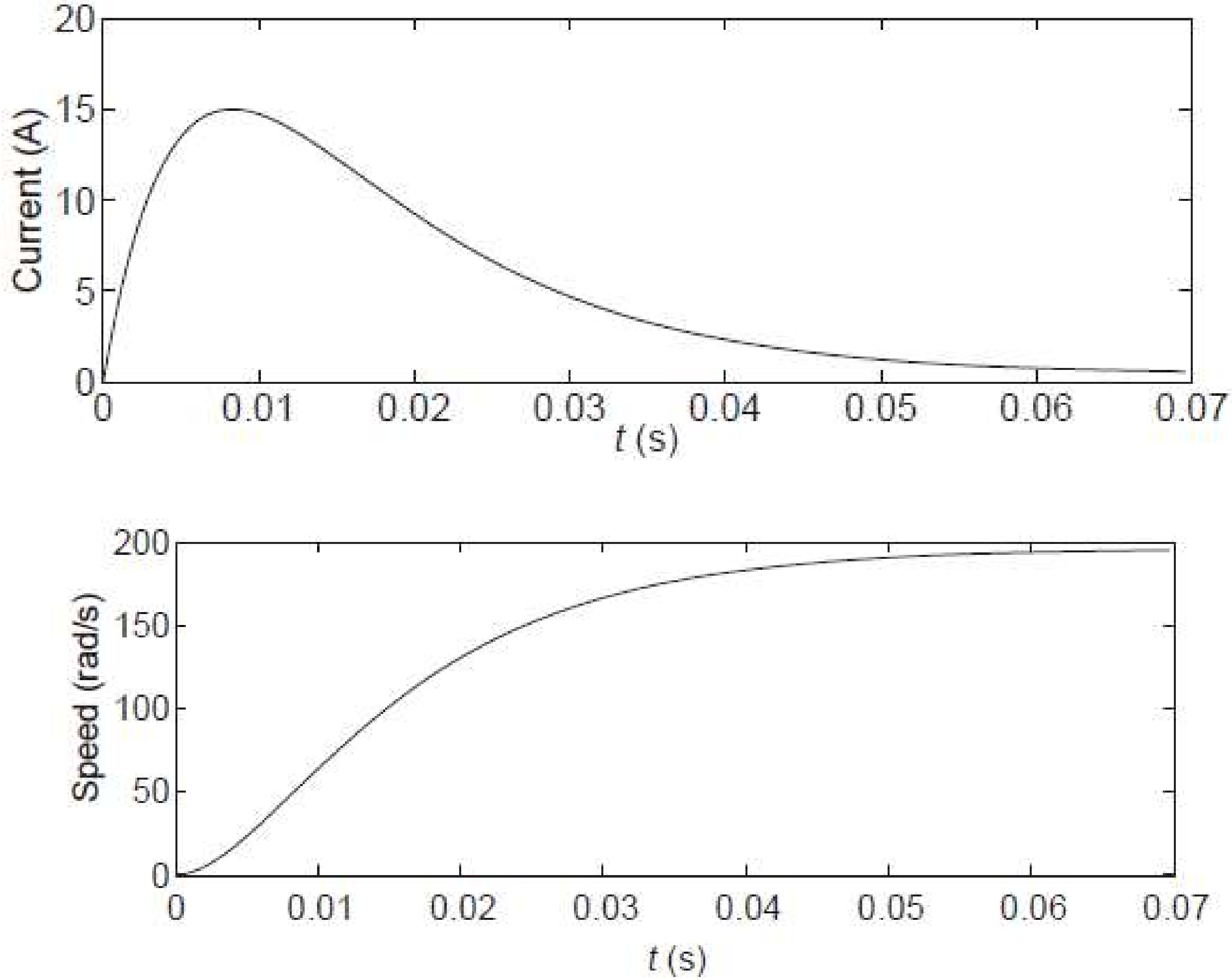
The following program computes reflected inertia and damping, the matrices for the motor model, and the state space model *sysmotor*.

```
% motor_mat.m Motor state matrices.
I = Im +IL/N^2;
c = cm + cL/N^2;
A = [-Ra/La,-Kb/La;KT/I,-c/I;];
B = [1/La,0;0, -1/(N*I)];
C = [1,0;0,1];
D = [0,0;0,0];
sysmotor = ss(A,B,C,D);
```

The following program computes and plots the response due to a step voltage of magnitude 10 v, with the load torque *TL* equal to zero.

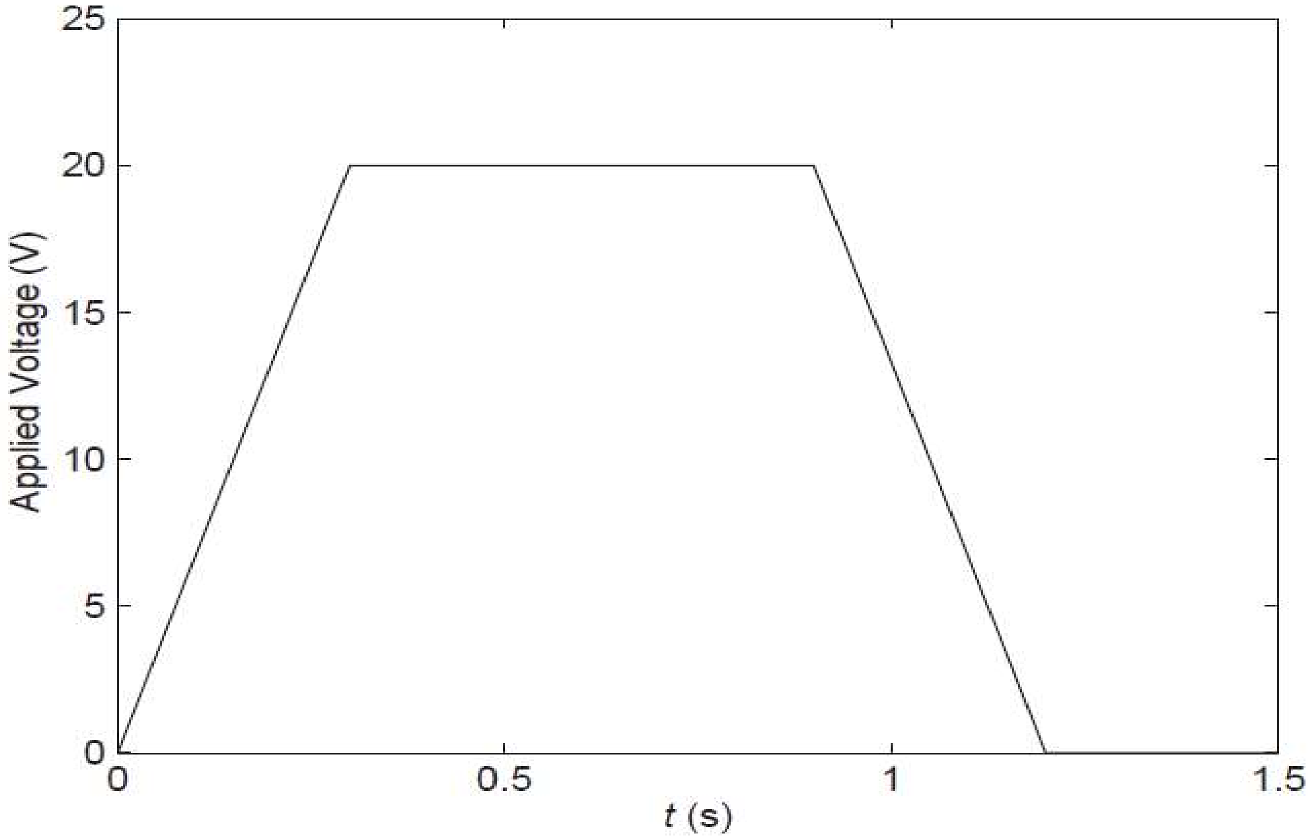
```
% state_step.m (Motor step response with state model)
motor_par
load_par
motor_mat
[y, t] = step(sysmotor);
subplot(2,1,1),plot(t,10*y(:,1)),...
   xlabel('t (s)'),ylabel('Current (A)')
subplot(2,1,2),plot(t,10*y(:,2)),...
   xlabel('t (s)'),ylabel('Speed (rad/s)')
```

The resulting plot looks like the following figure



- Trapezoidal Response

Suppose the input voltage is the following trapezoidal function where v_{max} = 20 V, t_1 = 0.3 s, t_2 = 0.9 s, t_f = 1.2 s, and t_3 = 1.5 s.



$$v(t) = \begin{cases} \dfrac{v_{\max}}{t_1} t & 0 \le t \le t_1 \\ v_{\max} & t_1 < t < t_2 \\ \dfrac{v_{\max}}{t_1}(t_f - t) & t_2 \le t \le t_f \\ 0 & t_f < t \le t_3 \end{cases}$$

The following program creates the voltage array v.

```
% Program trapezoid.m (Trapezoidal voltage profile)
t1 = 0.3; t2 = 0.9; tfinal = 1.2; t3 = 1.5;
v_max = 20;
dt = t3/1000;
t = (0:dt:t3);
for k = 1:1001
  if t(k) <= t1
     v(k) = (v_max/t1)*t(k);
  elseif t(k) <= t2
     v(k) = v_max;
  elseif t(k) <= tfinal
     v(k) = (v_max/t1)*(tfinal-t(k));
  else
     v(k) = 0;
  end
end
```

The next program uses the *trapz* function to compute an integral with the trapezoidal rule.

```
% Program performance.m
% Computes motor performance measures.
ia = y(:,1);
speed = y(:,2);
E = trapz(t,Ra*ia.^2)+trapz(t,c*speed.^2)
i_max = max(ia)
i_rms = sqrt(trapz(t,ia.^2)/t3)
T_max = KT*i_max
T_rms = KT*i_rms
speed_max = max(speed);
v_max = Ra*i_max+Kb*speed_max
```

The next program uses performance and trapezoid to compute the response.

```
% Program trapresp.m (Motor trapezoidal response)
motor_par
load_par
 motor_mat
 trapezoid
 u = [v', TL*ones(size(v'))];
 y = lsim(sysmotor,u,t);
 subplot(2,1,1),plot(t,y(:,1)),...
    xlabel('t (s)'),ylabel('Current (A)')
 subplot(2,1,2),plot(t,y(:,2)),...
    xlabel('t (s)'),ylabel('Speed (rad/s)')
 performance
```

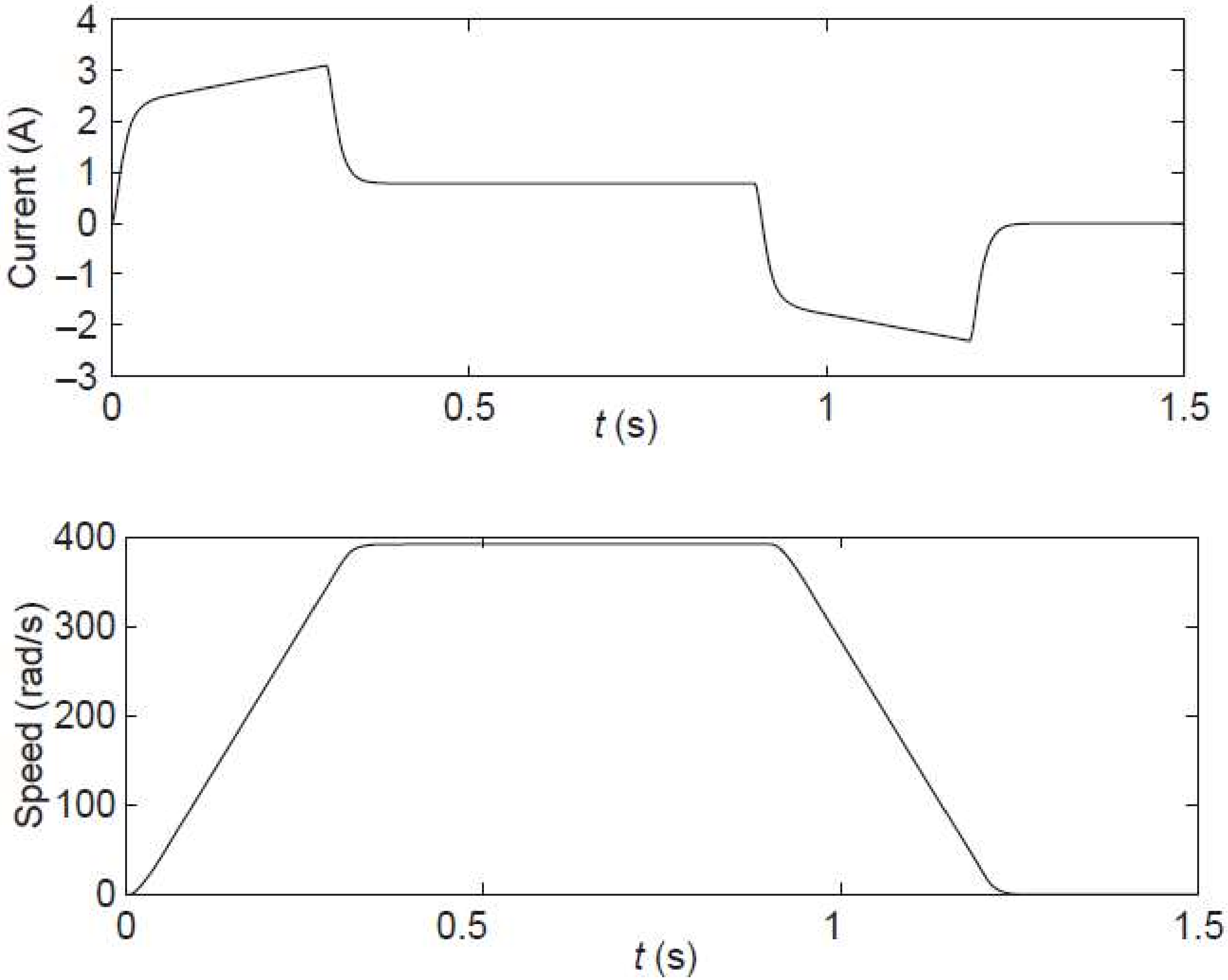
The computed performance measures are

$$E = 14.12 \text{ J/cycle} \qquad i_{\max} = 3.09 \text{ A} \qquad i_{\text{rms}} = 1.56 \text{ A}$$

$$T_{\max} = 0.15 \text{ N} \cdot \text{m} \qquad T_{\text{rms}} = 0.08 \text{ N} \cdot \text{m} \qquad v_{\max} = 21.15 \text{ V}$$

$$\omega_{\max} = 392 \text{ rad/s}$$

The resulting plot is shown in the following figure.



Current (A)
4
3
2
1
0
−1
−2
−3
0
0.5
1
1.5
t (s)
Speed (rad/s)
400
300
200
100
0
0
0.5
1
1.5
t (s)

SystemDynamics

Chapter 6. Electrical and Electromechanical Systems

6.6 Simulink Applications

邹渊
Yuan Zou
Tel: 68944115
Email: zouyuan@bit.edu.cn

Simulink Applications

- Advantages of simulink resistance

1.Simulink is especially useful for obtaining the response of systems to input functions that are more complicated than step, impulse, ramp, or sine functions.

2. Simulink is also helpful for computing the response of systems that contain nonlinear elements whose behavior is difficult to analyze by hand and tedious to program in MATLAB.

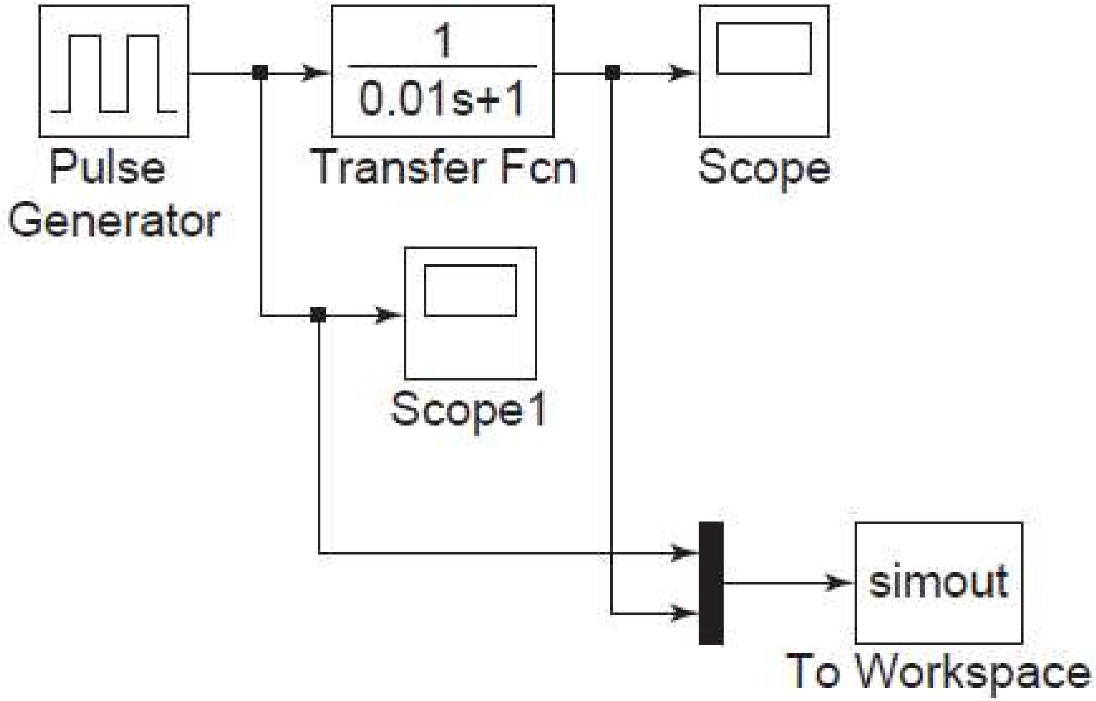
Simulation With A Pulse Input

Use the Pulse Generator block to find the response

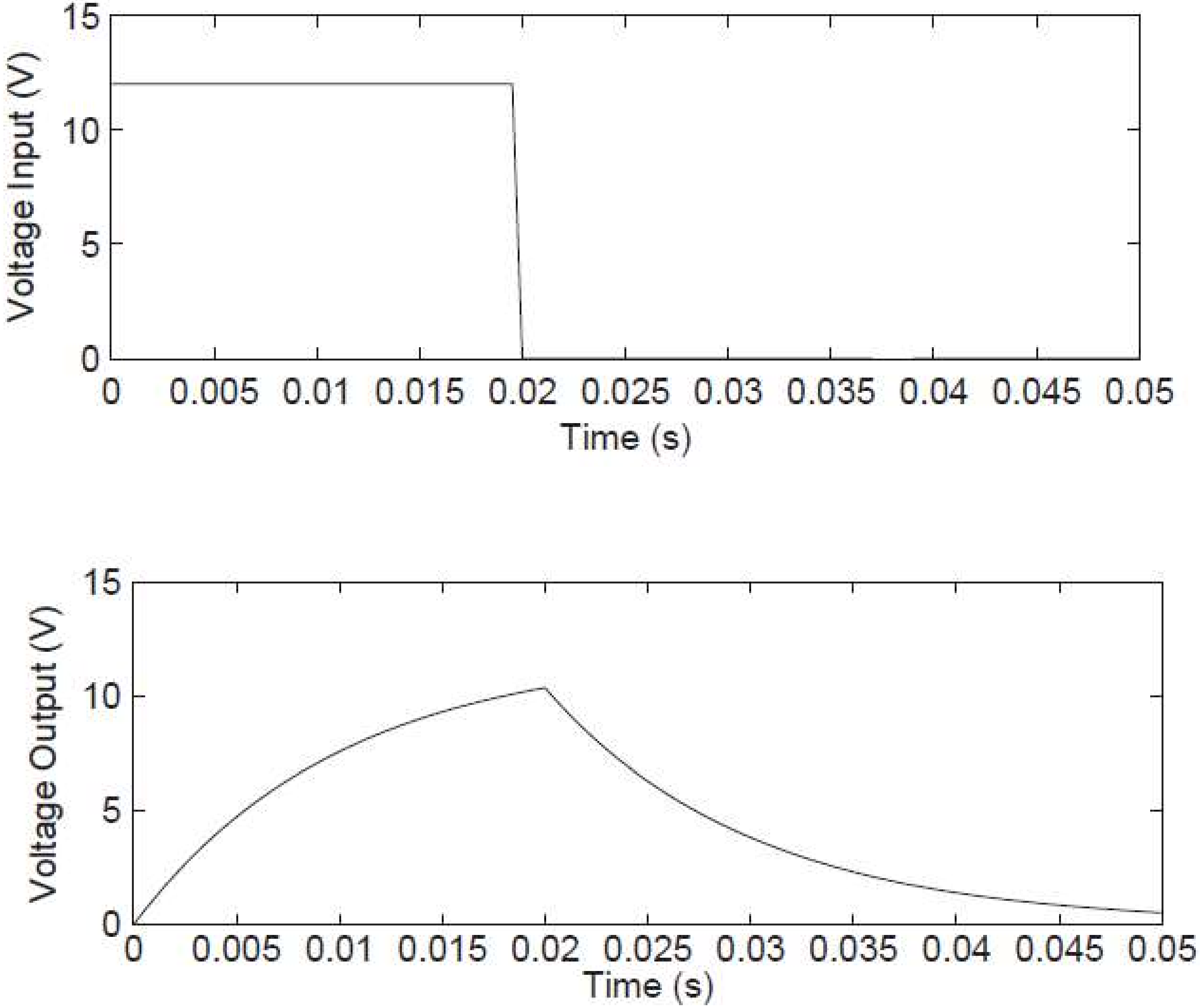
The transfer function is $$\frac{V_o(s)}{V_i(s)} = \frac{1}{RCs+1}$$

Use the values *R* 104 and *C* = 10−6 F; use a single pulse of amplitude 12 V and duration 0.02 s as an input.

Placing the blocks as shown in the following figure.

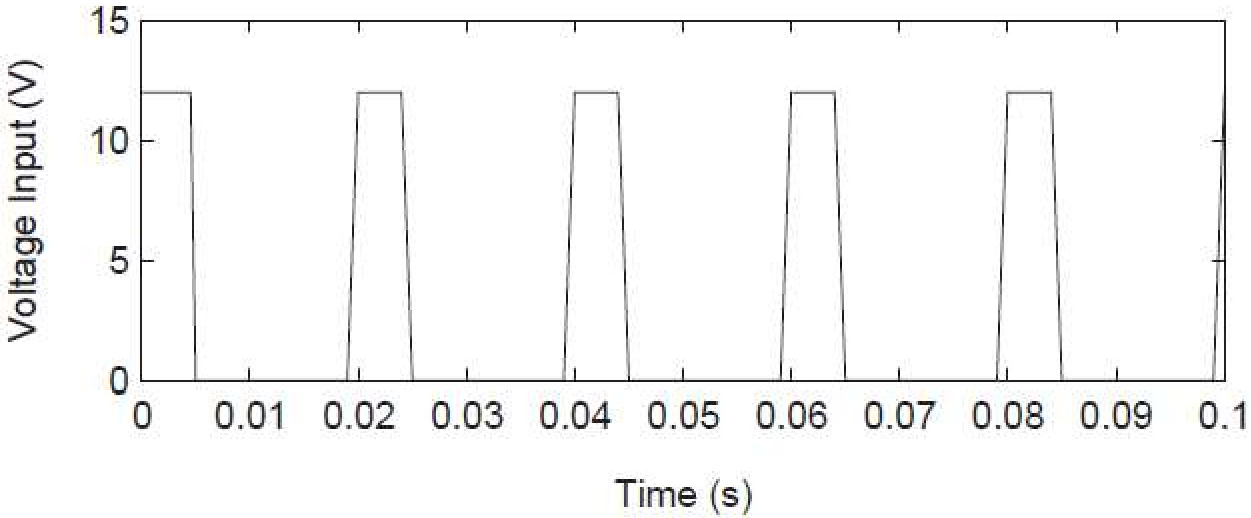


The resulting plot is shown in the following figure.

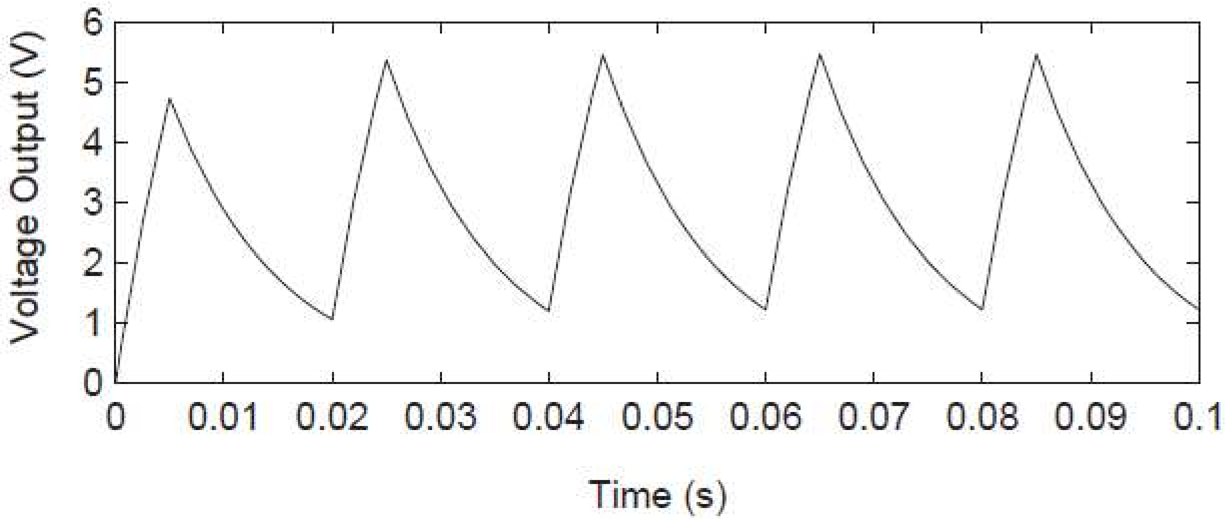


To obtain the response for a series of pulses of amplitude 12 V, of duration 0.005 s, and period 0.02 s, double-click on the Pulse Generator block and change the Period to 0.02, and the Pulse width to 25%.

The resulting plot is shown in the following figure.



15
10
5
0
Voltage Input (V)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time (s)



6
5
4
3
2
1
0
Voltage Output (V)
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time (s)

- Torque Limitation In Motors

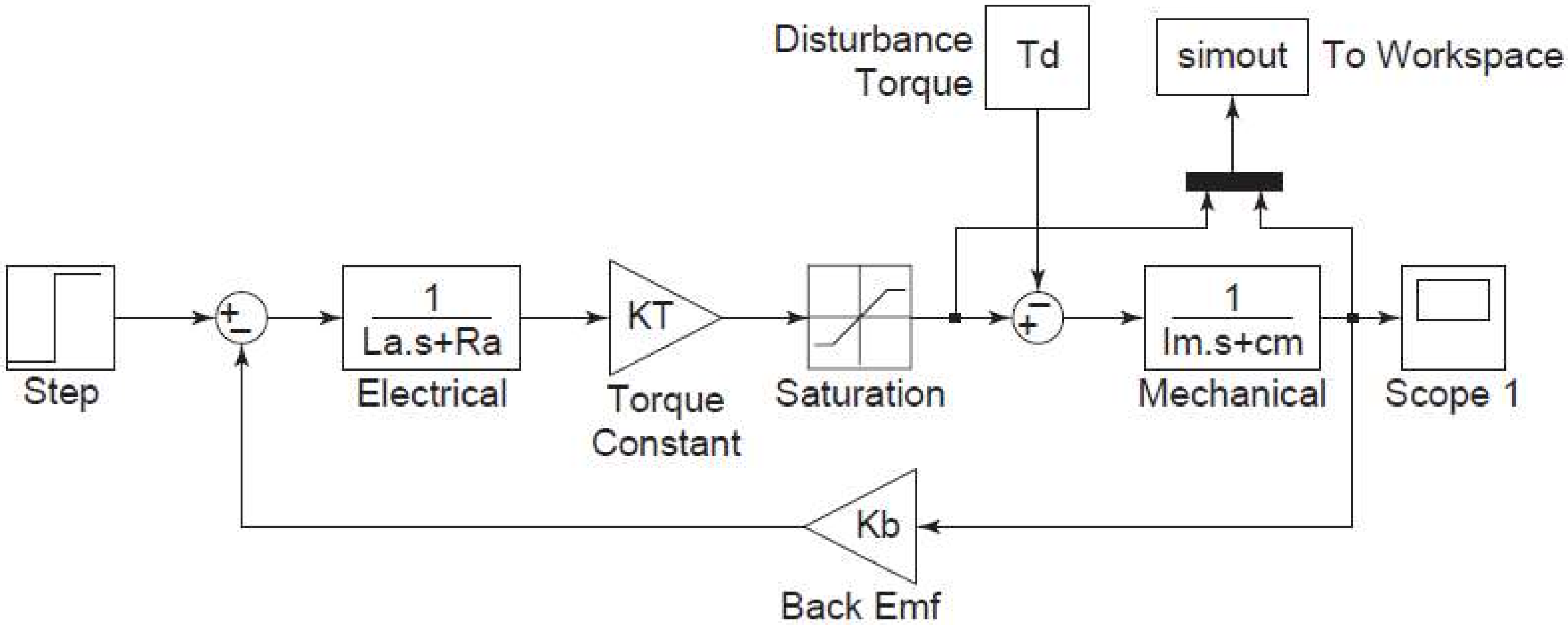
Use Simulink to examine the effects of torque limits on the step response.

$$\Omega(s) = \frac{1}{I_m s + c_m}[T(s) - T_d(s)]$$

$$T(s) = K_T I_a(s) = K_T \frac{1}{L_a s + R_a} V_a(s)$$

$$V_a(s) = V(s) - K_b \Omega(s)$$

Placing the blocks as shown in the following figure.



Notice the feature: the Constant block, labeled Disturbance Torque, and the use of variables as coefficients in a block.

Enter the denominator of the Electrical block as [La, Ra] and the denominator of the Mechanical block as [Im, cm]. Similarly, set the gains to KT and Kb, and set the constant in the DisturbanceTorque block toTd. Note that these parameters do not yet have values.

Set the lower and upper limits in the Saturation block to −0.4 and 0.4, respectively, to limit the motor torque to ± 0.4 N · m.

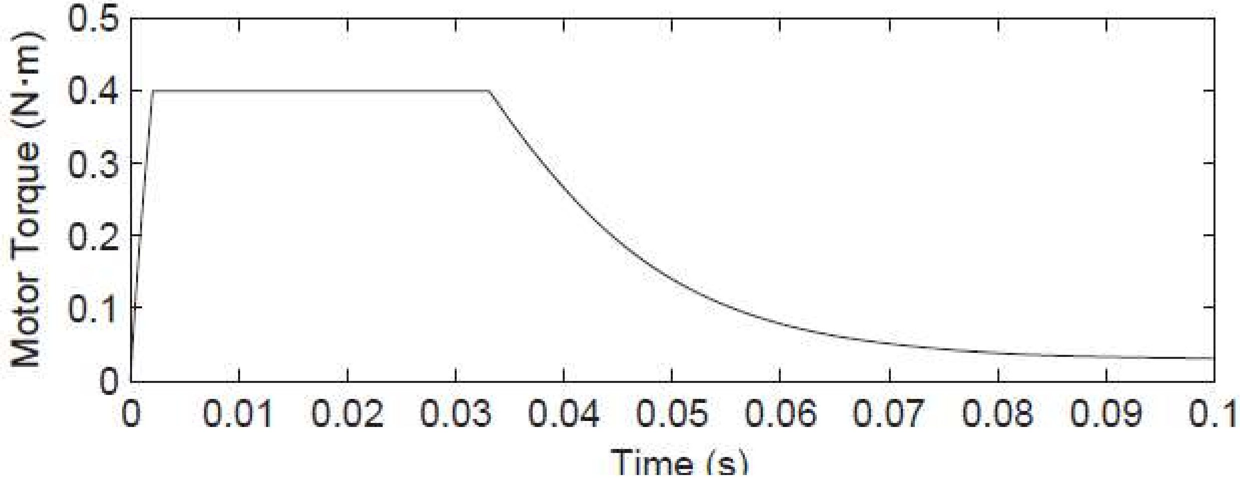
Set the Step Time of the Step block to 0 and the Final Value to 10, which corresponds to a 10 V input.

Then, in the MATLAB Command window, run the program *motor_par* described in Section 6.5

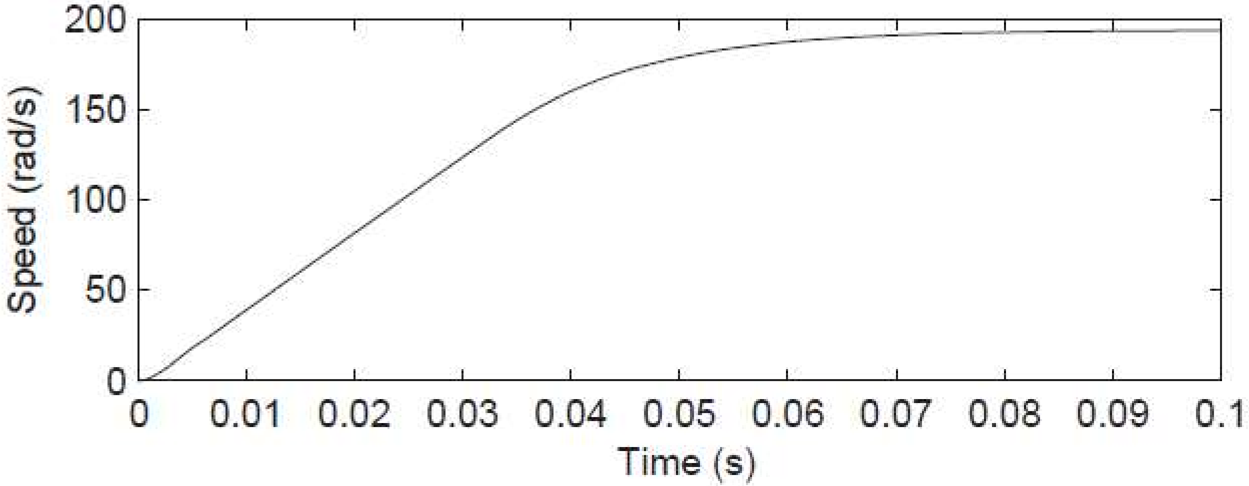
This will set the values of all the parameters except for the torque *Td* , which can be set to 0.01 N · m by typing Td = 0.01 in the Command window.

Then run the Simulink model

The results can be plotted as shown in following figure.



Motor Torque (N·m)
0.5
0.4
0.3
0.2
0.1
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time (s)



Speed (rad/s)
200
150
100
50
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time (s)

Note that the motor torque is limited, and this limits the slope of the speed curve. You can check the effects of this limitation by setting larger limits (say ±1 N・ m) in the Saturation block.

If the torque was not limited, it would reach a peak value of about 0.8 N・ m, and the speed would approach its steady-state value faster.

Note that it would be difficult to use the State Space block to simulate torque limitation. As we will see many more times, each model form—state variable and transfer function—have their own advantages.