ME 318M – Programming and Engineering Computational Methods

Homework # 10

Assigned: November 25th, 2020

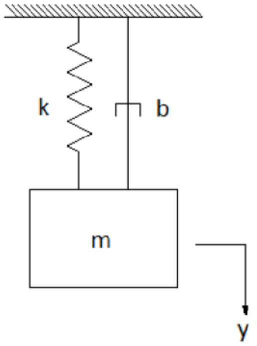
Due: Tuesday, December 1st, 2020 at 5pm (late homeworks accepted until 11:59pm)

Name: Jingsi Zhou Section Unique Number: 17460

UID: jz24729

Problem 1:

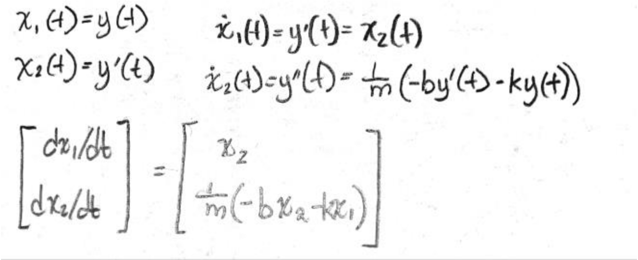
Consider a standard spring-mass-damper system shown in the figure below.



The equation for the system is given below. Consider ***y(t)*** to be the vertical displacement of the mass ***m*** of 100 kg, damping coefficient ***b*** to be 100 Ns/m, and spring constant ***k*** to be 10000 N/m. At $t = 0, y = 0.1$ and __!"# = 0.

$$m\frac{d^2y}{dt^2} + b\frac{dy}{dt} + ky = 0$$

a) Formulate the given second order equation as two first order equations.

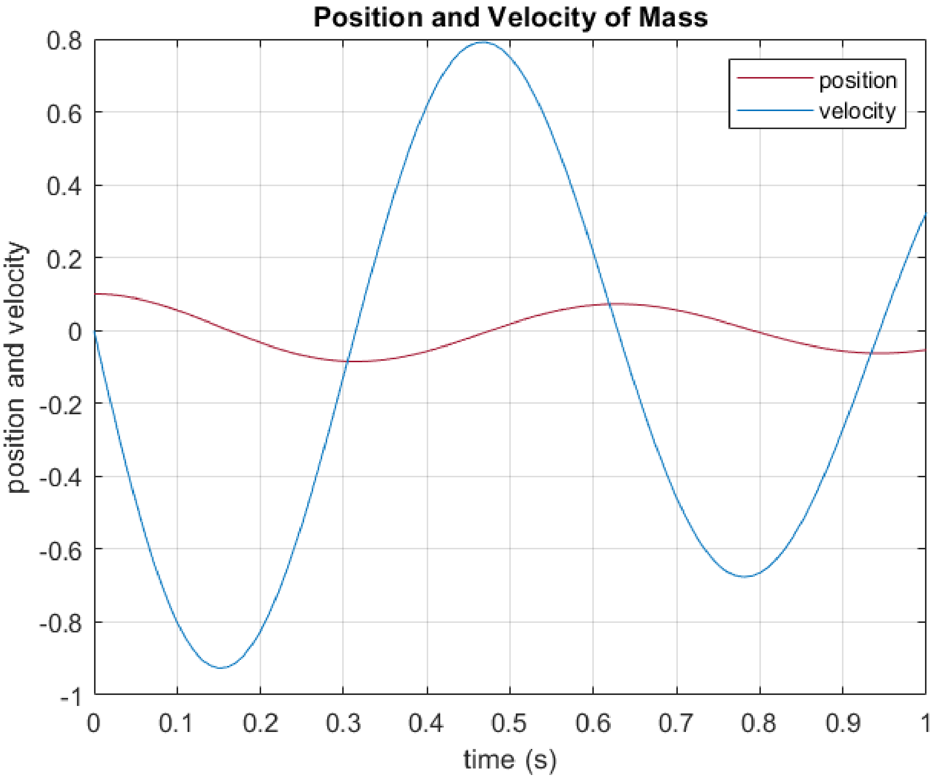


$x_1(t) = y(t)$ $\quad \dot{x}_1(t) = y'(t) = x_2(t)$

$x_2(t) = y'(t)$ $\quad \dot{x}_2(t) = y''(t) = \frac{1}{m}(-by'(t) - ky(t))$

$$\begin{bmatrix} dx_1/dt \\ dx_2/dt \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{m}(-bx_2 - kx_1) \end{bmatrix}$$

b) Solve the system above for displacement and velocity using the *4th order* ***Runge-Kutta method*** over the interval t=[0 , 1] for step size of 0.01. Plot your solutions on a single plot (use plot command and "hold on" trick; plot the position and velocity with different colors and enclose a legend).

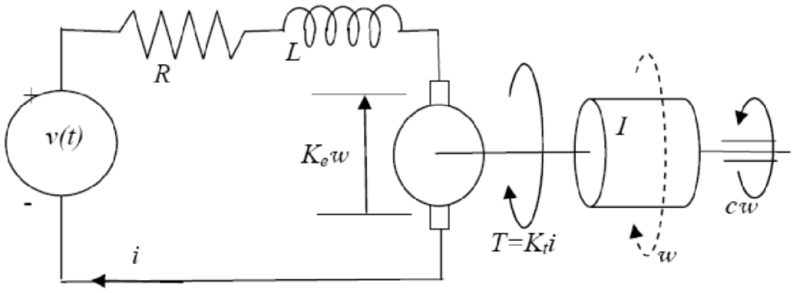


MATLAB code:

```
function dx = MSD(x,t)
m = 100;
b = 100;
k = 10000;
M(1, :) = [0, 1];
M(2, :) = 1/m*[-k, -b];
dx = M*x;


x(1:2,1) = [0.1; 0];
t0 = 0;
h = 0.01;
t_end = 1;
t = t0:h:t_end;
for i = 1:numel(t)-1
    k1n = MSD(x(:,i), t(i));
    k2n = MSD(x(:,i) + h/2*k1n, t(i) + h/2);
    k3n = MSD(x(:,i) + h/2*k2n, t(i) + h/2);
    k4n = MSD(x(:,i) + h*k3n, t(i) + h);
    x(:, i+1) = x(:, i) + h/6*(k1n + 2*k2n + 2*k3n + k4n);
end

a1 = plot(t, x(1,:));
grid on
hold on
a2 = plot(t, x(2,:));
xlabel('time (s)')
ylabel('position and velocity')
legend([a1; a2], ['position'; 'velocity']);
title('Position and Velocity of Mass')
```

Problem 2:

Old take home exam problem (relatively tough – was worth 45 points that year)



Armature Controlled DC Motor

R = 1 Ω
L = 0.012 H
K_t = 0.050 Nm/A
K_e = 0.061 V·s/rad
c = 0.005 N·s/m
I = 10.2 · 10^{-5} N·m·s^2

Consider this armature controlled DC motor. You can calculate the angular velocity *ω* of the inertial element with moment of inertia *I* that is being rotated by the DC motor using the following model.

$$I\frac{d\omega}{dt} = K_{\#}i - c\omega$$

$$L\frac{di}{dt} = -Ri - K_{\%}\omega + v(t)$$

The parameters *L* and *R* are respectively the inductance and resistance of the motor armature (windings on the rotor), while *i* denotes the current passing through the armature windings. Constants $K_{\%}$ and $K_{\#}$ are the so-called "Back emf" (back electro-magnetic force) and torque constants, respectively. A bearing supports the shaft opposite of the motor and creates a torque due to viscous friction. The resistance torque created by the bearing is proportional to the angular velocity *ω* of the shaft with the damping coefficient *c* being the constant of proportionality. Numerical values of all parameters are enclosed next to the figure above.

(a) Let $z_{\&} = \omega$ and $z_{'} = i$. Rewrite the above equations in matrix form

$$\text{A}\dot{z}z^{\&'}\text{C}=\mathbf{A}\text{D}z^{Z\&'}\text{E}+\mathbf{b}v(t)$$

where *v*(*t*) is the input voltage into the motor armature. Please note that **A** is a 2 by 2 matrix, while **b** is a 2 by 1 vector. You need to determine the 2 by 2 matrix **A** and 2 by 1 vector **b**.

If you cannot do that, please continue the rest of the problem using

$$\mathbf{A} = \text{D}^{1}\ {}^{2}\text{E},\ \mathbf{b} = \text{D}^{5}\text{E}$$

3 4 6

These are not the correct matrices and if you use them, you will obtain only up to 60% of the points available in this problem.

$$z_1 = \omega \qquad I\frac{d\omega}{dt} = K_t i - c\omega$$

$$z_2 = i \qquad L\frac{di}{dt} = -Ri - K_e\omega + v(t)$$

$$v(t) = L\frac{di}{dt} + Ri + K_e\omega$$

$$\dot{z}_1 = \frac{d\omega}{dt} = \frac{1}{I}(K_t i - c\omega) = \frac{1}{I}(K_t z_2 - cz_1) + 0\,v(t)$$

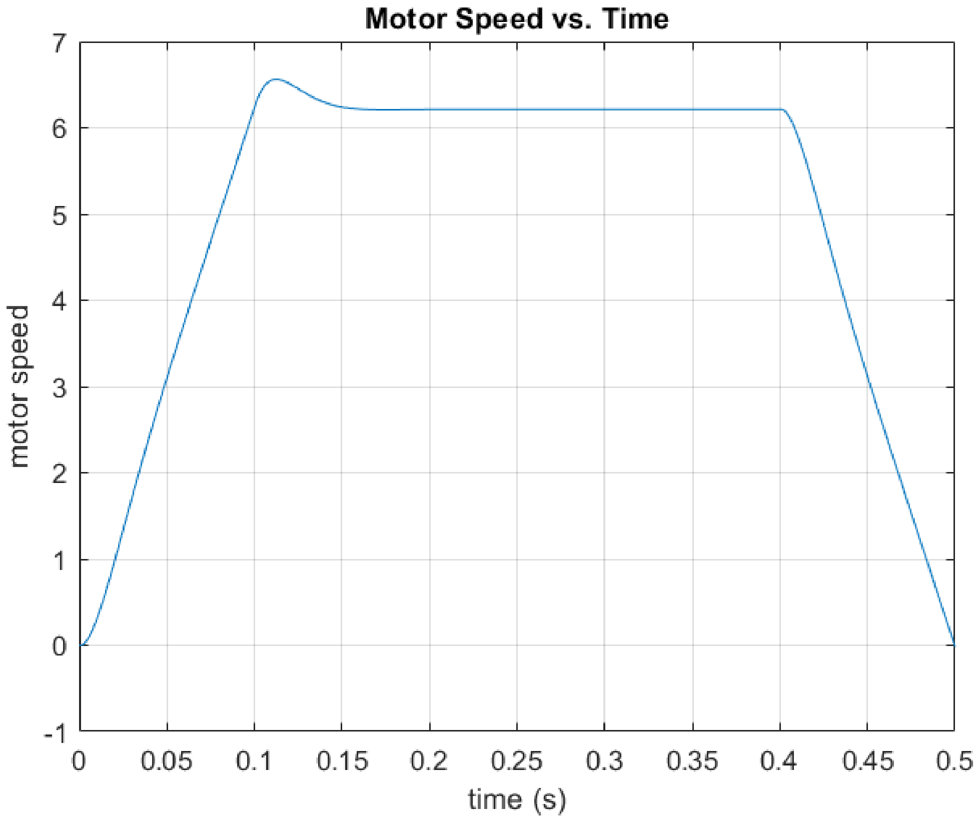
$$\dot{z}_2 = \frac{di}{dt} = \frac{1}{L}(-Ri - K_e\omega + v(t)) = \frac{1}{L}(-Rz_2 - K_e z_1) + \frac{1}{L}v(t)$$

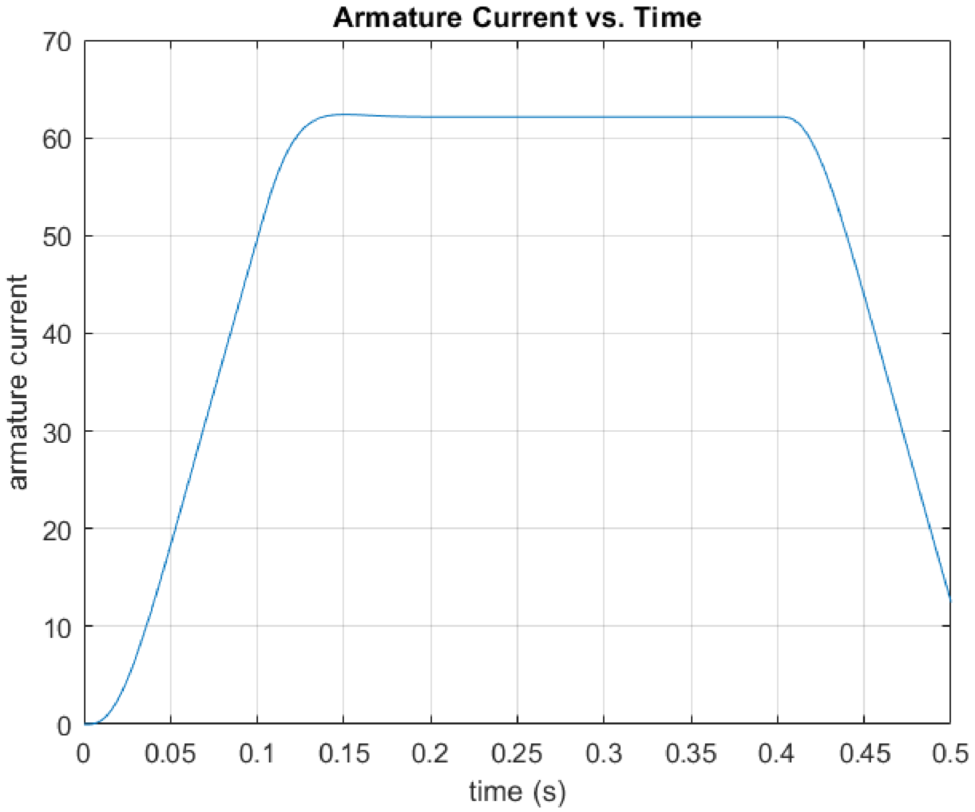
$$A = \begin{bmatrix} \frac{-0.005}{10.2\cdot10^{-5}} & \frac{0.050}{10.2\cdot10^{-5}} \\ \frac{-0.061}{0.012} & \frac{-1}{0.012} \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ \frac{1}{0.012} \end{bmatrix}$$

(b) Assume the motor is initially in full rest and the armature current of the motor is zero Amperes, when the voltage starts linearly rising over 0.1 seconds to 10V, where it stays for 0.3 seconds, after which it linearly falls back to zero Volts in the next 0.1 seconds. In other words, the voltage follows the following function[1]

Calculate the motor speed ω and current i, with a constant time step of 0.001 seconds, using Euler's method. Turn in separate plots of motor's speed and armature current versus time. Execute the Euler's method for long enough to notice that the motor speed eventually becomes constant.

[1] It's a10V pulse that lasts for 0.5 seconds, but in the real world, voltage cannot be instantly switched on, and even if it was possible, doing it may be undesirable since large electrical inertial load would be placed on the electrical components).

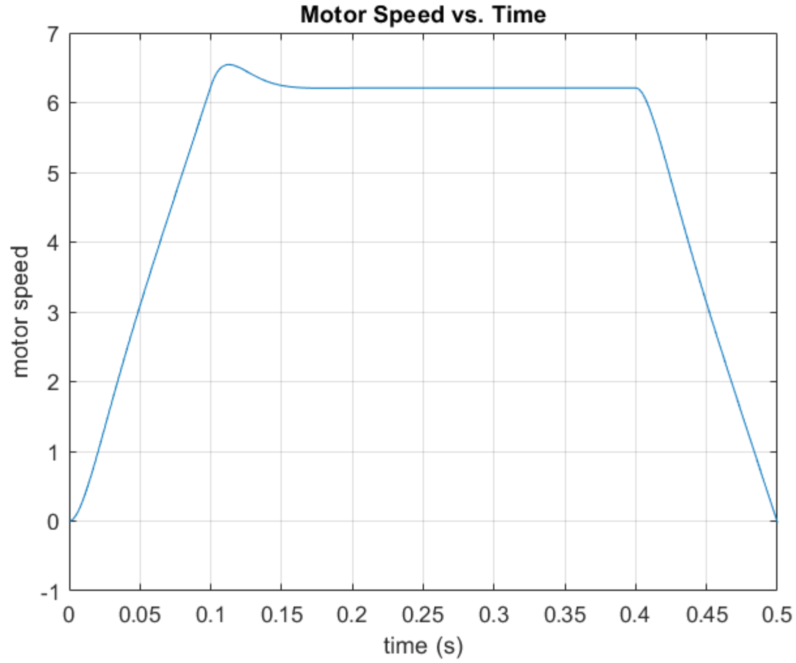

Motor Speed vs. Time
motor speed
time (s)
7
6
5
4
3
2
1
0
-1
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

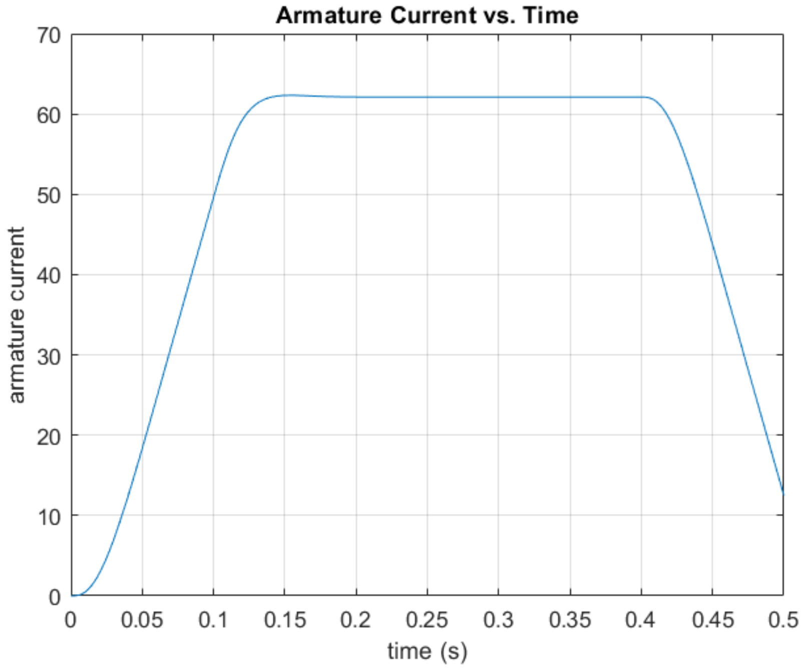

Armature Current vs. Time
armature current
time (s)
70
60
50
40
30
20
10
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

MATLAB code:

```
t0 = 0;
t_end = 0.5;
h = 0.001;
t = t0:h:t_end;
z(1:2, 1) = [0; 0];
for i = 1:numel(t)-1
    z(:, i+1) = z(:,i) + elec(z(:, i), t(i))*h;
end

figure(1);
a1 = plot(t, z(1,:));
xlabel('time (s)')
ylabel('armature current')
title('Armature Current vs. Time')
grid on

figure(2)
a2 = plot(t, z(2,:));
xlabel('time (s)')
ylabel('motor speed')
title('Motor Speed vs. Time')
grid on

function voltage = v(t)
if 0 <= t && t< 0.1
    voltage = 100*t;
elseif 0.1 <= t && t <0.4
    voltage = 10;
elseif 0.4 <= t && t< 0.5
    voltage = -100*(t-0.4) + 10;
else
    voltage = 0;
end
end

function dz = elec(z, t)
A = [-0.005/(10.2*10^-5), 0.050/(10.2*10^-5); -0.061/0.012, -1/0.012];
b = [0; 1/0.012];
dz = A*z + b*v(t)
end
```

(c) Repeat part (b) using 4th order Runge Kutta method, with a step of 0.001 seconds. Compare your answers from (b) with those obtained in this part of the problem. Please comment on which results would you trust more and why.



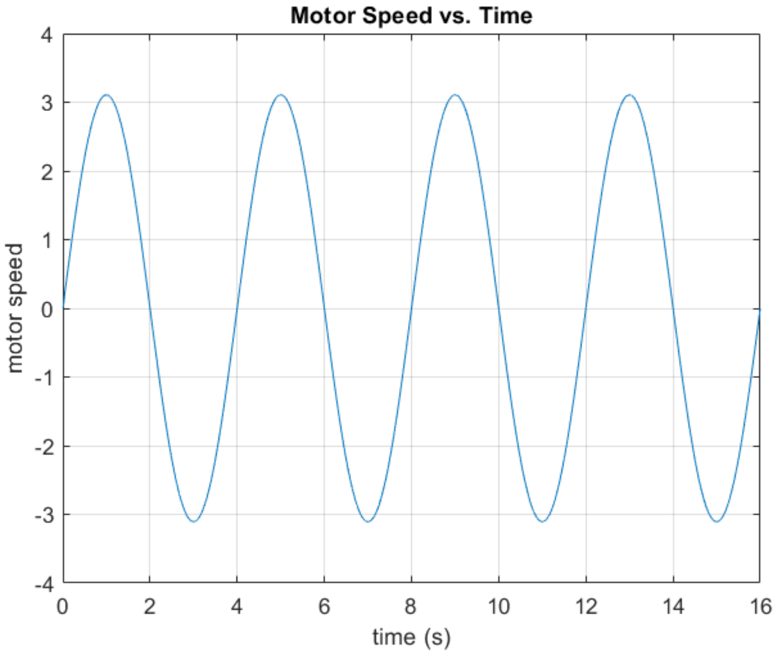


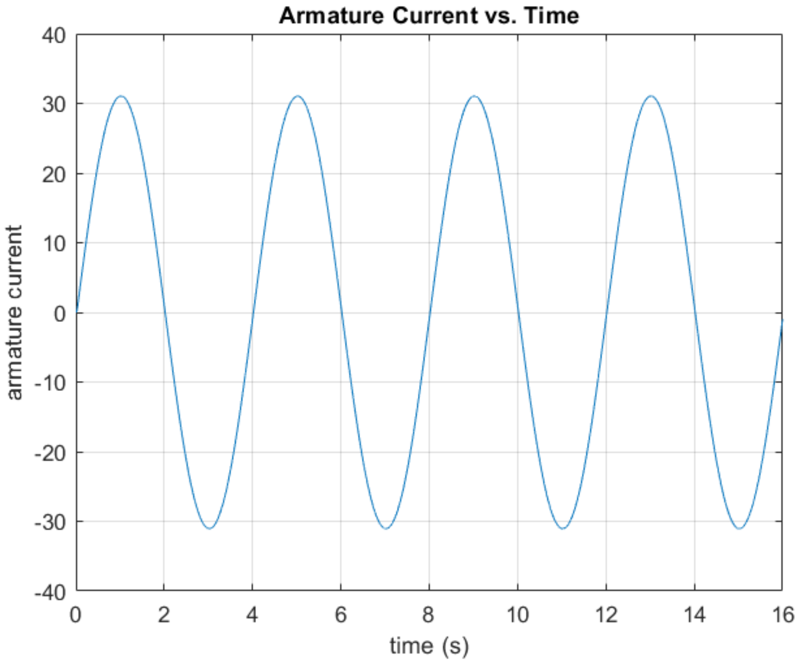
I would trust the Runge Kutta method more because it uses a 4th order Taylor series estimation and is therefore more accurate. The Euler method is only first order Taylor series approximation. When part (b) and (c) are compared, the shape of the functions are pretty much identical to the naked eye. However, the values for part (c) vary slightly from those of part (b).

(d) Again, assume that the motor is initially at full rest and that initial armature current is zero Amperes. Calculate and plot the angular velocity of this motor using 4th order Runge Kutta method for step of 0.001 seconds, over the period $0 < t < 16$ seconds, with input voltage $v(t)$ being

$$v(t) = 5 \cdot \sin(0.5\pi t)$$

Comment on what the motor speed seems to be doing as time passes (Do you see oscillations developing? What is their periodicity? Compare that period to the periodicity of the input voltage.)





The motor speed oscillates with a period of 4 seconds as time passes. The period of voltage input is also 4 seconds.

MATLAB code (sim. part (c)):

```
z(1:2,1) = [0; 0];
t0 = 0;
h = 0.001;
t_end = 16;
t = t0:h:t_end;
for i = 1:numel(t)-1
    k1n = elec(z(:,i), t(i));
    k2n = elec(z(:,i) + h/2*k1n, t(i) + h/2);
    k3n = elec(z(:,i) + h/2*k2n, t(i) + h/2);
    k4n = elec(z(:,i) + h*k3n, t(i) + h);
    z(:, i+1) = z(:, i) + h/6*(k1n + 2*k2n + 2*k3n + k4n);
end

figure(1);
a1 = plot(t, z(1,:));
xlabel('time (s)')
ylabel('armature current')
title('Armature Current vs. Time')
grid on

figure(2)
a2 = plot(t, z(2,:));
xlabel('time (s)')
ylabel('motor speed')
title('Motor Speed vs. Time')
grid on


function dz = elec(z, t)
A = [-0.005/(10.2*10^-5), 0.050/(10.2*10^-5); -0.061/0.012, -1/0.012];
b = [0; 1/0.012];
dz = A*z + b*v(t);
end

function voltage = v(t)
voltage = 5*sin(0.5*pi*t);
end

%{
function voltage = v(t)
if 0 <= t && t< 0.1
    voltage = 100*t;
elseif 0.1 <= t && t <0.4
    voltage = 10;
elseif 0.4 <= t && t< 0.5
    voltage = -100*(t-0.4) + 10;
else
    voltage = 0;
end
end
%}
```