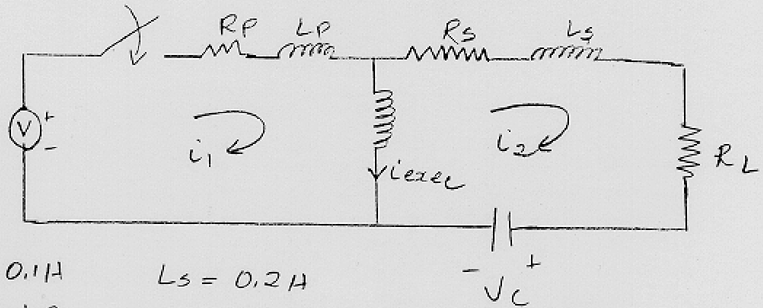
EE 743 (HW#1 Solution)

Luis Siqueira M.   Homework no1   January 20th, 1999

1-



$Lp = 0.1H$ $Ls = 0.2H$

$Rp = 1\Omega$ $Rs = 2\Omega$

$R_L = 1\Omega$ $C = 1\mu F$

$Mi = 0.1H$ $V = 10V$ (Step Input)

1- Differential equations

$$V = Rp i_1 + Lp \frac{di_1}{dt} + Mi\left(\frac{di_1}{dt} - \frac{di_2}{dt}\right)$$

$$0 = Mi\left(\frac{di_2}{dt} - \frac{di_1}{dt}\right) + Rs i_2 + Ls\frac{di_2}{dt} + R_L i_2 + V_C$$

$$0 = -\frac{dV_C}{dt}C + i_2$$

2- State space model

$$(Lp + Mi)\frac{di_1}{dt} - Mi\frac{di_2}{dt} = -Rp i_1 + V$$

$$(Ls + Mi)\frac{di_2}{dt} - \frac{Mi\, di_1}{dt} = -(Rs + R_L)i_2 - V_C$$

$$C\frac{dV_C}{dt} = i_2$$

$$\begin{bmatrix} (L_p+M_i) & -M_i & 0 \\ -M_i & (L_s+M_i) & 0 \\ 0 & 0 & C \end{bmatrix} \begin{bmatrix} \frac{di_1}{dt} \\ \frac{di_2}{dt} \\ \frac{dv_c}{dt} \end{bmatrix} = \begin{bmatrix} -R_p & 0 & 0 \\ 0 & -(R_s+R_L) & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ V_c \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} V$$

$$\begin{bmatrix} \frac{di_1}{dt} \\ \frac{di_2}{dt} \\ \frac{dV_c}{dt} \end{bmatrix} = \begin{bmatrix} -R_p & 0 & 0 \\ 0 & -(R_s+R_L) & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} (L_p+M_i) & -M_i & 0 \\ -M_i & (L_s+M_i) & 0 \\ 0 & 0 & C \end{bmatrix}^{-1} \begin{bmatrix} i_1 \\ i_2 \\ V_c \end{bmatrix} + \begin{bmatrix} (L_p+M_i) & -M_i & 0 \\ -M_i & (L_s+M_i) & 0 \\ 0 & 0 & C \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$A = \begin{bmatrix} -R_p & 0 & 0 \\ 0 & -(R_s+R_L) & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} (L_p+M_i) & -M_i & 0 \\ -M_i & (L_s+M_i) & 0 \\ 0 & 0 & C \end{bmatrix}^{-1}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} (L_p+M_i) & -M_i & 0 \\ -M_i & (L_s+M_i) & 0 \\ 0 & 0 & C \end{bmatrix}^{-1}$$

$$\dot{X} = \begin{bmatrix} \dot{i}_1 \\ \dot{i}_2 \\ \dot{V}_c \end{bmatrix}$$

$$\dot{X} = AX + BU$$

3- Trapezoidal Integration.

a) This section has been done using Matlab (See the attached program). The results has been plotted (attached pages).

b) The simulink section is also included, as well as the output.

5- Mi is defined as:

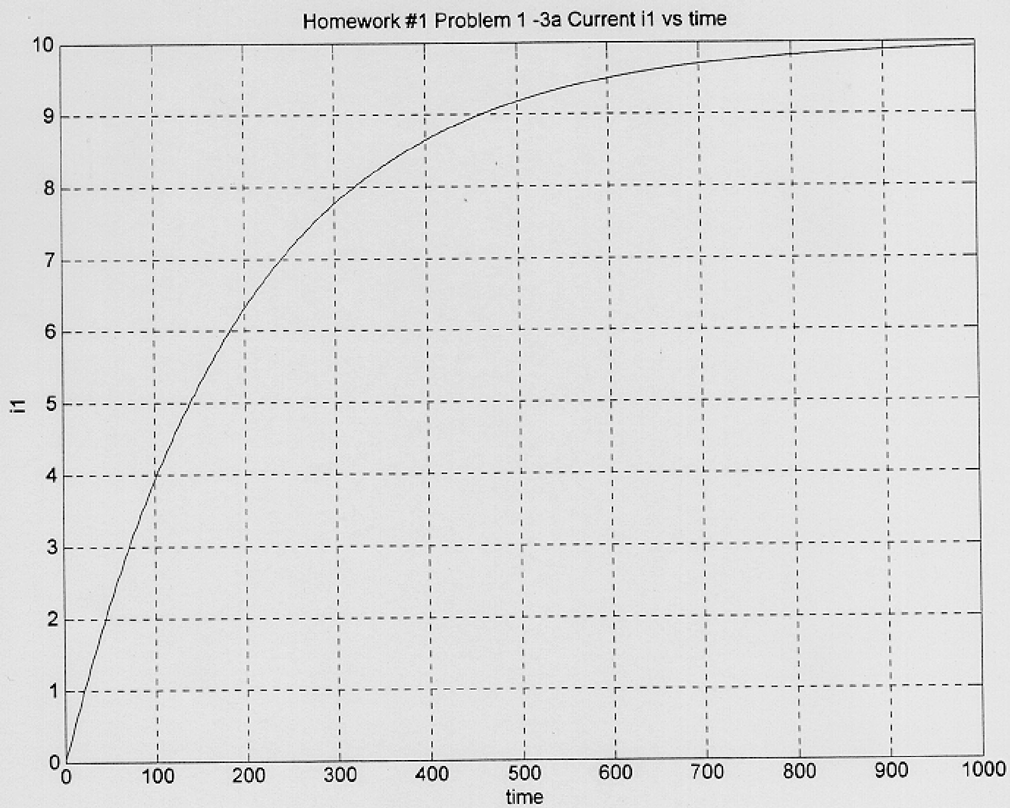
$$Mi = 0.1\,H \quad \text{if } I_{exc} < 4 \text{ amps}$$

$$Mi = \frac{0.1}{(1 + I_{exc}^2)} + 0.001\,H \quad \text{if } I_{exc} > 4 \text{ amps}$$

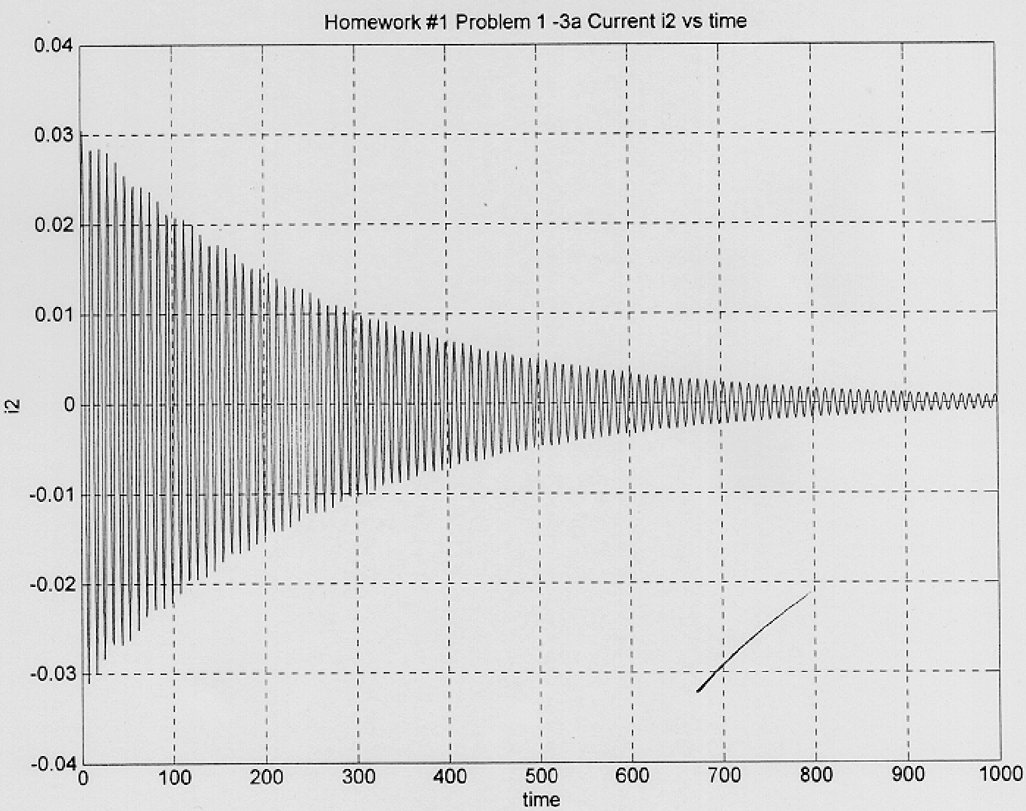
a) Matlab program and outputs enclosed.

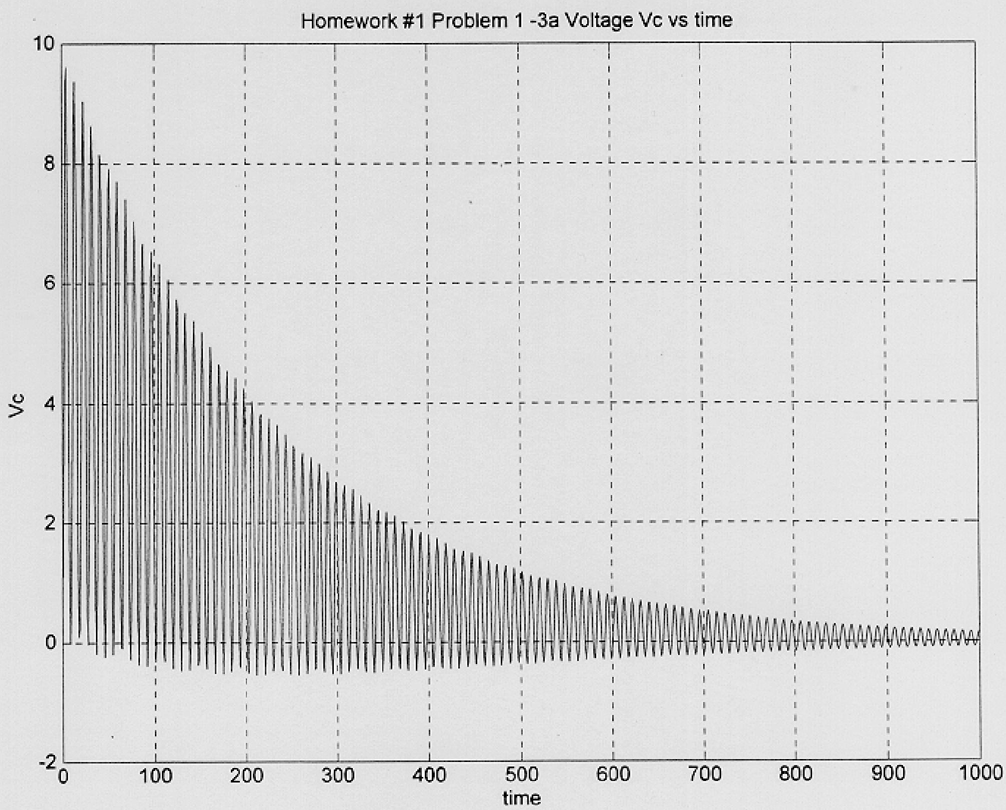
b) Simulink section. Program and output included.

```
% ASSIGNMENT #1 PROBLEM 3a
% TRAPEZOIDAL INTEGRATION TO DETERMINE DYNAMIC RESPONSE OF A SYSTEM USING MATLAB


clear
Lp=0.1
Ls=0.2
Mi=0.1
Rp=1
Rs=2
Rl=1
C=1e-5
V=10
alpha=0.1


   R = [-Rp 0 0
         0 -(Rs+Rl) -1
         0  1 0]

   D = [1
        0
        0]

   L = [(Lp+Mi) -Mi       0
        -Mi     (Ls+Mi)  0
         0       0        C];

   Linv = inv(L);
   A=Linv*R;
   B=Linv*D;

   X = [0; 0; 0]

   U = V


   T = 0.0001;

   for n=1:10000

      % TRAPEZOIDAL INTEGRATION

      Xest = X + T*(A*X + B*U);
      Xdotest = A*Xest + B*U;
      alpha1 = 1 + alpha;
      alpha2 = 1- alpha;
      term1 = alpha1*Xdotest;
      termint = A*X + B*U;
      term2 = alpha2* termint;
      X = X + (T/2)*[term1 + term2] ;

      i1(n) = X(1);
      i2(n) = X(2);
      Vc(n) = X(3);

   end
```



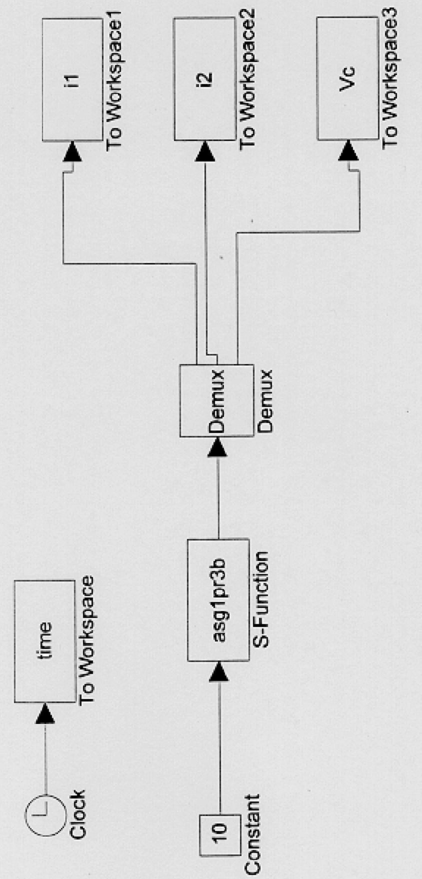
Homework #1 Problem 1 -3a Current i1 vs time
i1
0
1
2
3
4
5
6
7
8
9
10
0
100
200
300
400
500
600
700
800
900
1000
time


Homework #1 Problem 1 -3a Current i2 vs time
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
i2
0
100
200
300
400
500
600
700
800
900
1000
time

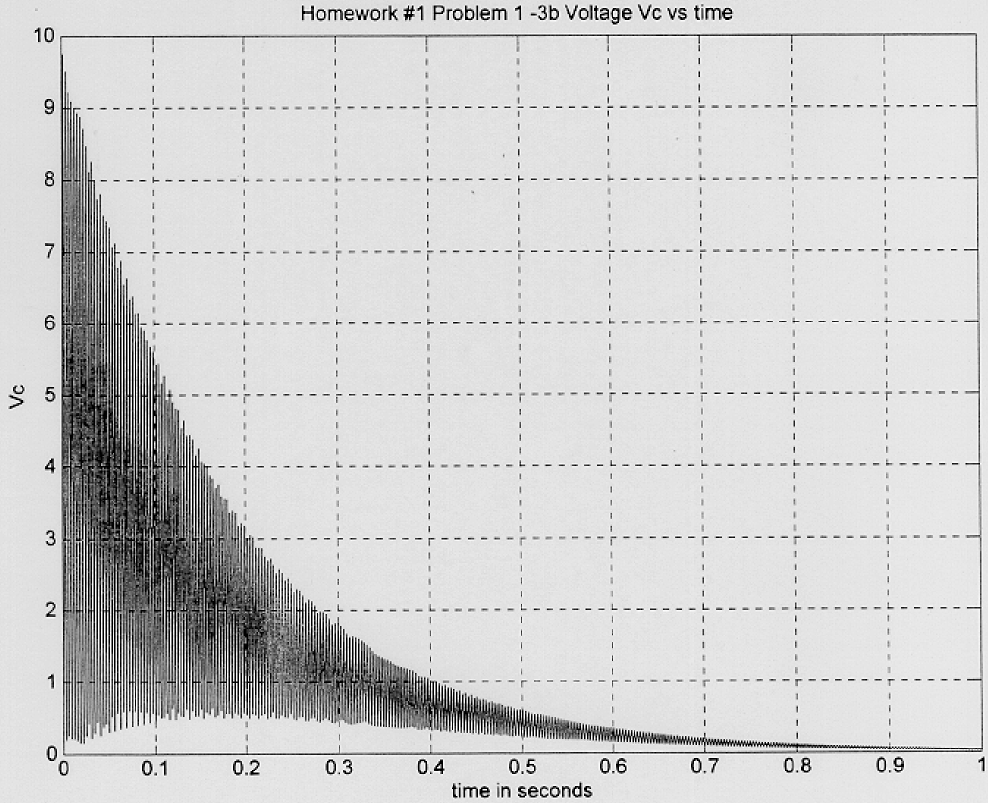

Homework #1 Problem 1 -3a Voltage Vc vs time
10
8
6
4
2
0
-2
Vc
0
100
200
300
400
500
600
700
800
900
1000
time

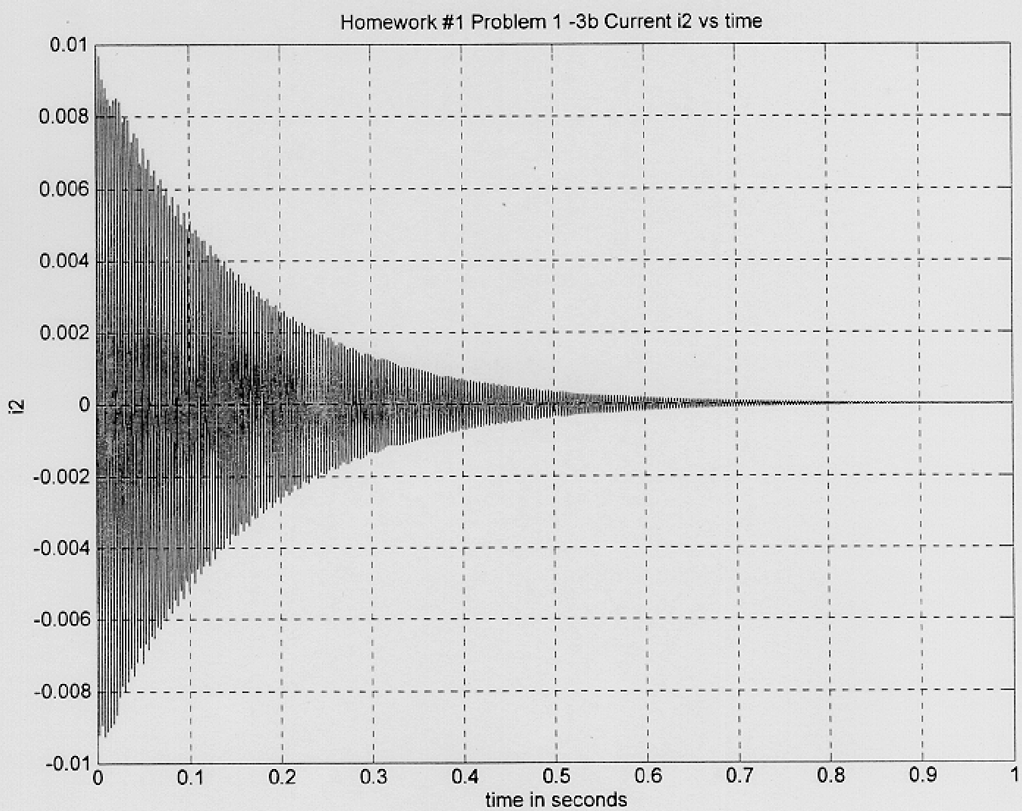
```
% ASSIGNMENT#1 PROBLEM 1-3b
% S-function FOR DETERMINATION OF DYNAMIC RESPONSE OF A SYSTEM USING SIMULINK

function [sys, x0]=probl(t,x,u,flag)

% System parameters:

Lp = 0.1;
Ls = 0.2;
Rp = 1;
Rs = 2;
Rl = 1;
Mi = 0.1;
C = 1e-6;

 R = [-Rp 0 0;
    0  -(Rs+Rl) -1;
    0 1 0];

D = [1 0 0]';

L = [(Lp+Mi) -Mi 0;
      -Mi (Ls+Mi) 0;
      0 0 C];

 Linv = inv(L);

 A = Linv * R;
 B = Linv * D;

% State Variables:
% ---------------
% x(1)=i1;
% x(2)=i2;
% x(3)=Vc;


% Inputs:
% ------

% V = 10;

% Outputs:
% ------
% sys(1)=i1;
% sys(2)=i2;
% sys(3)=Vc;

if abs(flag)==1

%  if flag=1, return state variables, x

   sys(1:3)=A*x(1:3)+B*u;

elseif abs(flag)==3

   sys(1:3) = x(1:3);


elseif flag==0   % return initial condition data, sizes and x0
     sys(1)=3;   % number of continuous states
     sys(2)=0;   % number of discrete state
     sys(3)=3;   % number of outputs
     sys(4)=1;   % number of inputs
     sys(5)=0;   % number of discontinous states
     sys(6)=0;
```

```
        x0=[0;0;0];
else
        % If flag is anything else, no need to return anything
        % since this is a continous system
        sys = [];
end;
```

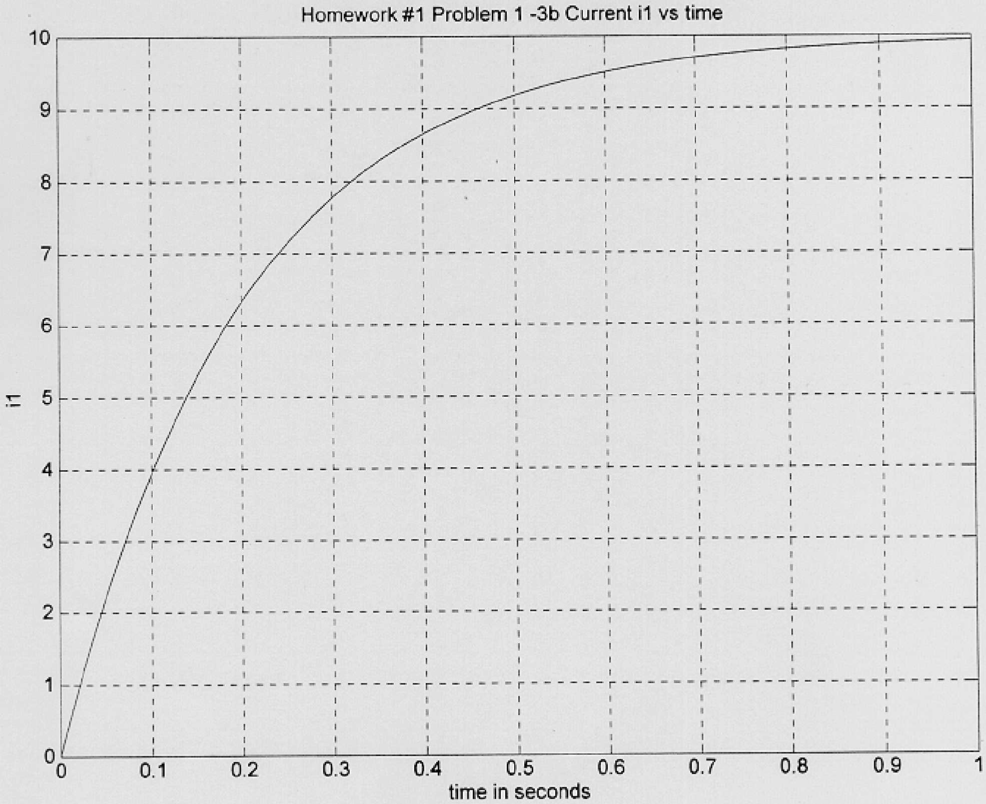

time
To Workspace
Clock
10
Constant
asg1pr3b
S-Function
Demux
Demux
i1
To Workspace1
i2
To Workspace2
Vc
To Workspace3

Homework #1 Problem 1 -3b Voltage Vc vs time



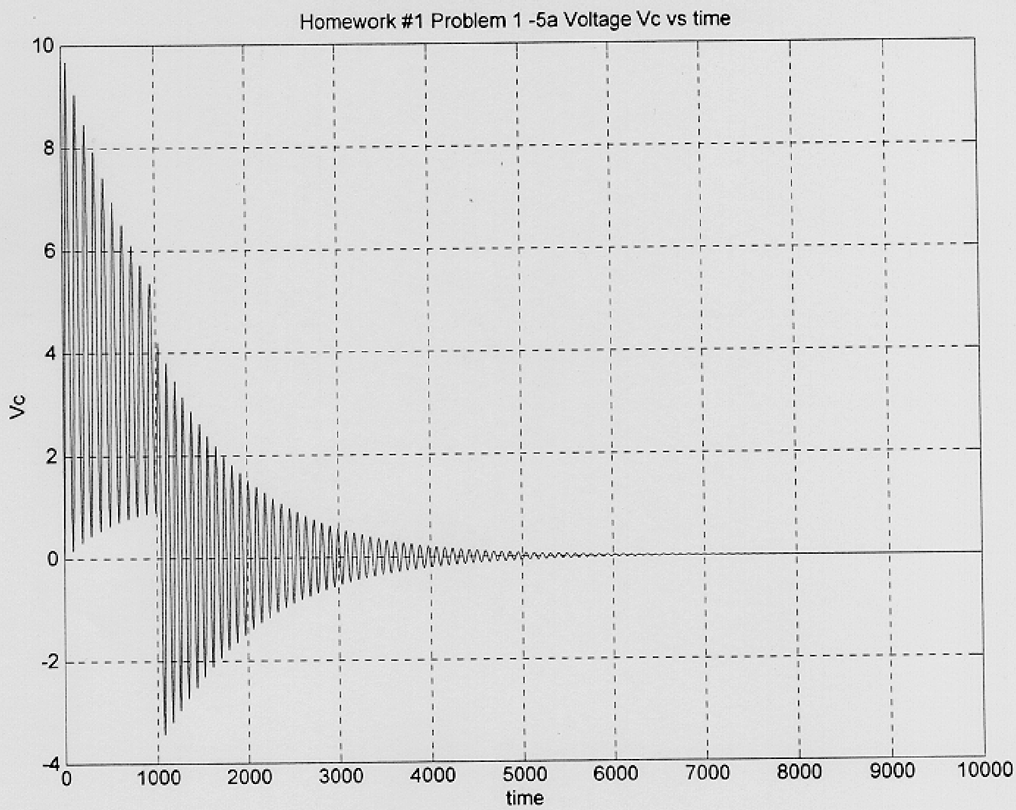

Homework #1 Problem 1 -3b Current i2 vs time
0.01
0.008
0.006
0.004
0.002
0
-0.002
-0.004
-0.006
-0.008
-0.01
i2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time in seconds

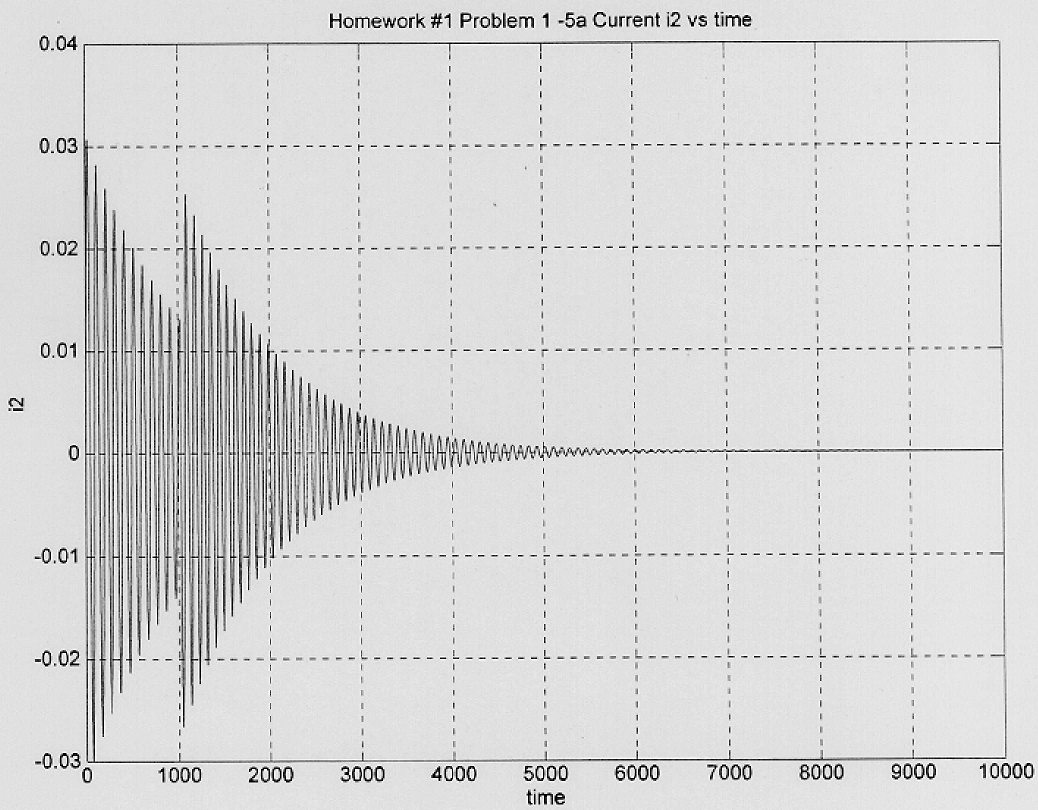
Homework #1 Problem 1 -3b Current i1 vs time



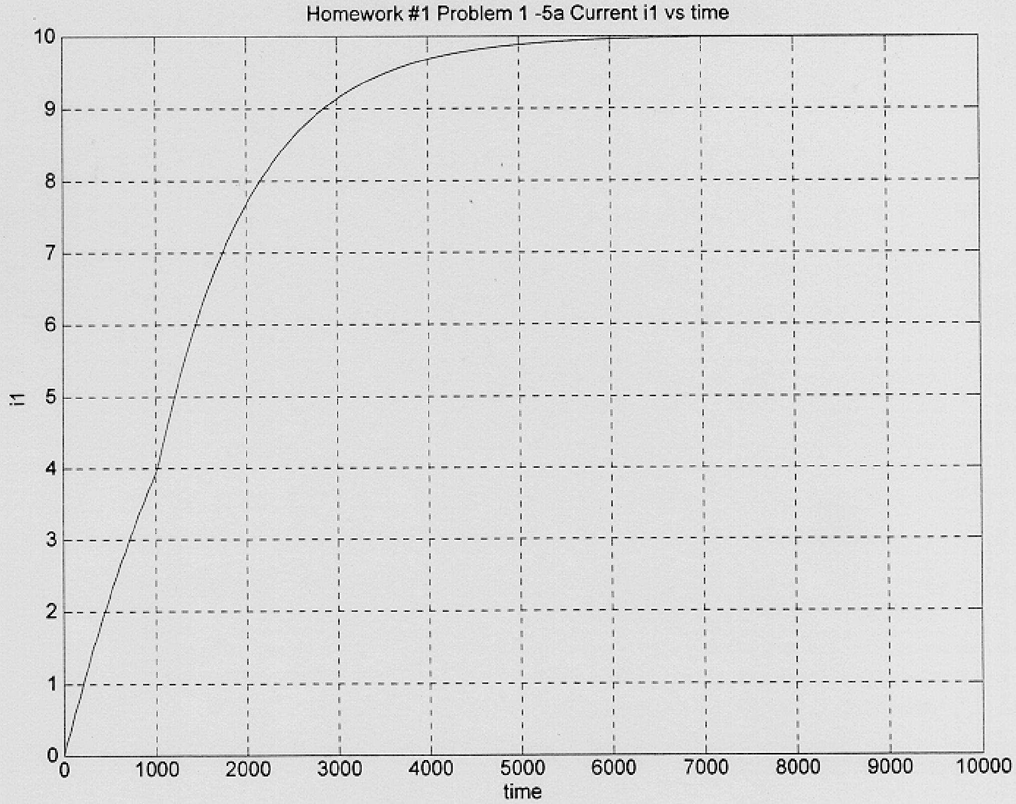
```
% ASSIGNMENT #1 PROBLEM 5a
% TRAPEZOIDAL INTEGRATION TO DETERMINE DYNAMIC RESPONSE OF A SYSTEM USING MATLAB
% MUTUAL INDUCTANCE IS A FUMCTION OF THE EXCITATION CURRENT


clear
Lp=0.1
Ls=0.2
Mi=0.1
Rp=1
Rs=2
Rl=1
C=1e-5
V=10
alpha=0.1


   R = [-Rp 0 0
         0 -(Rs+Rl) -1
         0  1 0]

   D = [1
        0
        0]

   X = [0; 0; 0]

   U = V



   T = 0.0001;

   for n=1:10000

      % COMPUTATION OF Mi

      iexec= (X(1) - X(2));

    if iexec <= 4
      Mi=0.1;
    else
      var = 1 + (iexec*iexec);
      Mi= 0.1/var;
     Mi= Mi + 0.0001;
   end

     L = [(Lp+Mi) -Mi       0
         -Mi     (Ls+Mi)  0
          0       0       C];

      Linv = inv(L);
      A=Linv*R;
      B=Linv*D;

      % TRAPEZOIDAL INTEGRATION

      Xest = X + T*(A*X + B*U);
      Xdotest = A*Xest + B*U;
      alpha1 = 1 + alpha;
      alpha2 = 1- alpha;
      term1 = alpha1*Xdotest;
      termint = A*X + B*U;
      term2 = alpha2* termint;
      X = X + (T/2)*[term1 + term2] ;

      i1(n) = X(1);
```

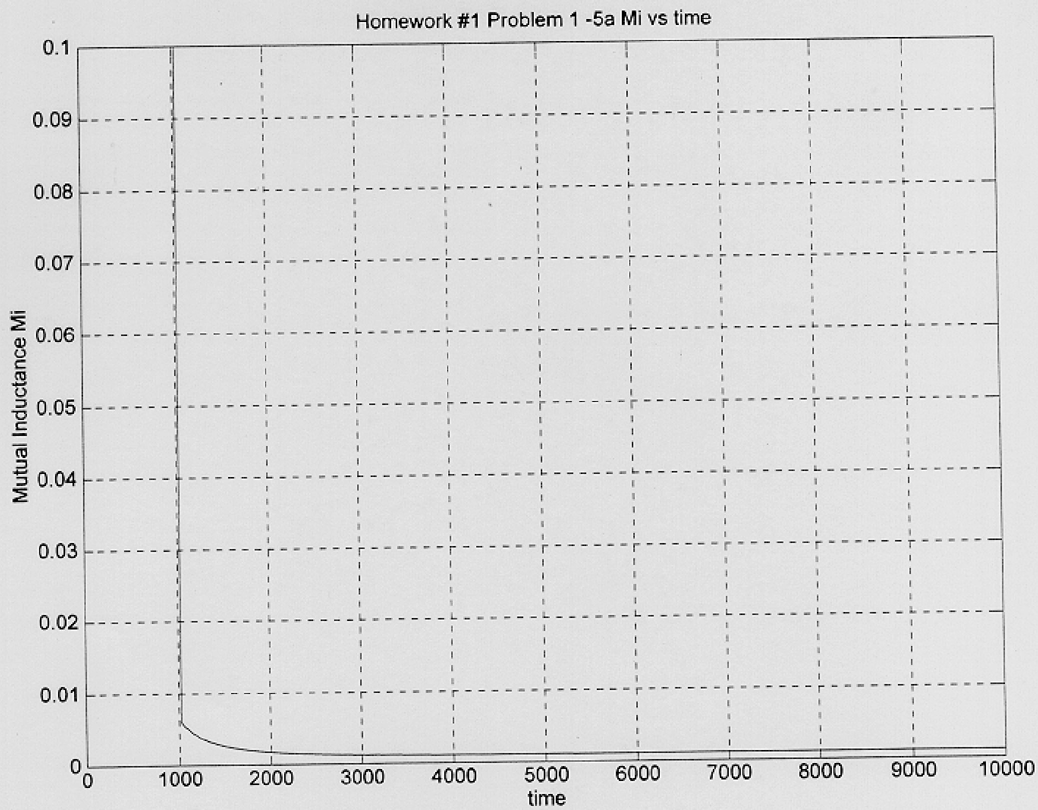
```
        i2(n) = X(2);
        Vc(n) = X(3);
        Mut_ind(n) = Mi;
    end
```


Homework #1 Problem 1 -5a Voltage Vc vs time
Vc
time
10
8
6
4
2
0
-2
-4
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000


Homework #1 Problem 1 -5a Current i2 vs time
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
i2
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
time

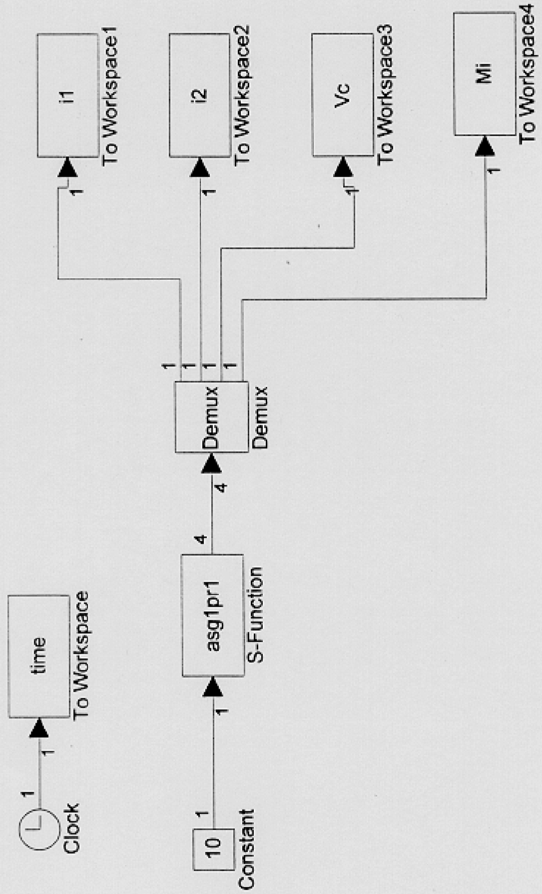
Homework #1 Problem 1 -5a Current i1 vs time

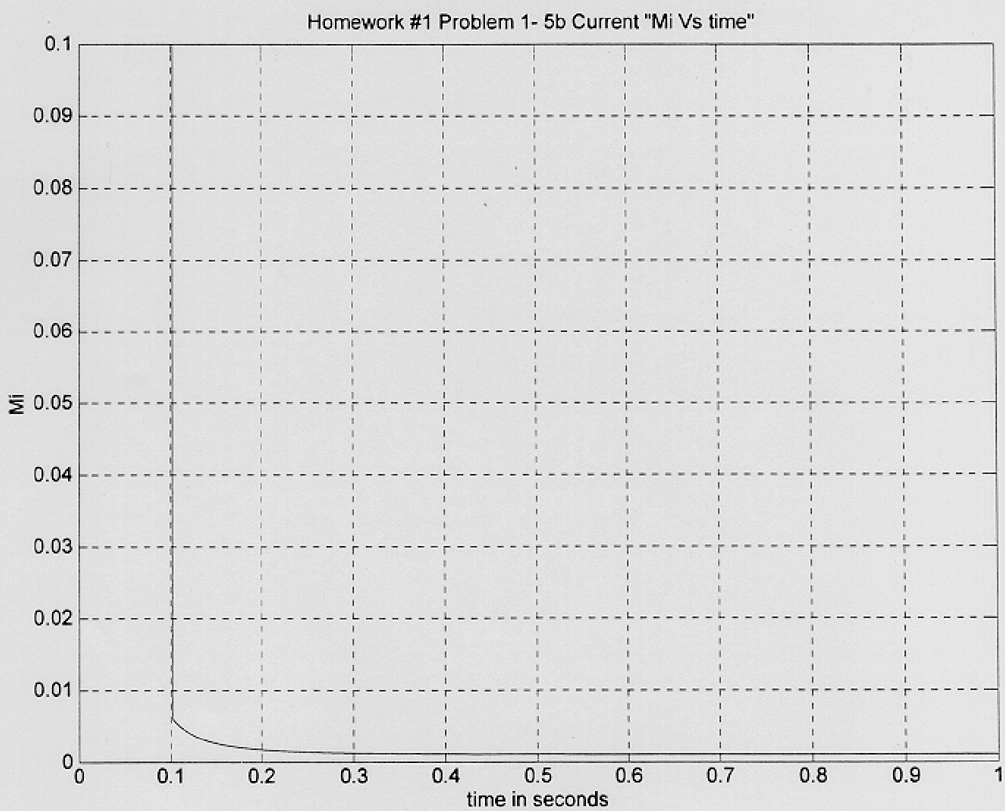



Homework #1 Problem 1 -5a Mi vs time
Mutual Inductance Mi
time
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

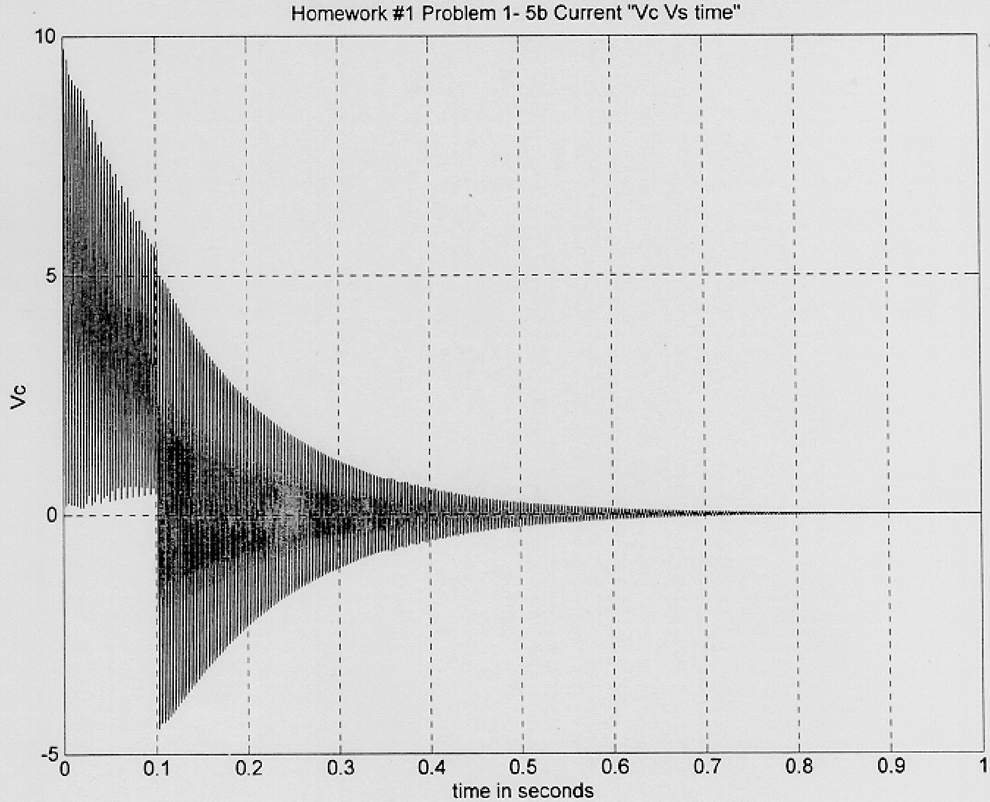
```
% ASSIGNMENT#1 PROBLEM 1-5b
% S-function FOR DETERMINATION OF DYNAMIC RESPONSE OF A SYSTEM USING SIMULINK
% MUTUAL INDUCATNCE IS A FUNCTION OF THE EXCITATION CURRENT

function [sys, x0]=prob1(t,x,u,flag)

% System parameters:

Lp = 0.1;
Ls = 0.2;
Rp = 1;
Rs = 2;
Rl = 1;
Mo = 0.1;
C = 1e-6;

 R = [-Rp 0 0;
    0  -(Rs+Rl) -1;
    0 1 0];

D = [1 0 0]';


% State Variables:
% ---------------
% % x(1)=i1;
% x(2)=i2;
% x(3)=Vc;

% Inputs:
% ------
% V = 10;

% Outputs:
% ------
% sys(1)=i1;
% sys(2)=i2;
% sys(3)=Vc;
% SYS(4)=Mi;

if abs(flag)==1

  % COMPUTATION OF Mi

  iexec= (x(1) - x(2));

    if iexec <= 4
       Mi=0.1;
    else
       var = 1 + (iexec*iexec);
   Mi= 0.1/var;
   Mi= Mi + 0.0001;
   end

      L = [(Lp+Mi) -Mi 0;
         -Mi (Ls+Mi) 0;
         0 0 C];

      Linv = inv(L);

      A = Linv * R;
      B = Linv * D;
      sys(1:3)=A*x(1:3)+B*u;
      iexec= (x(1) - x(2));
```

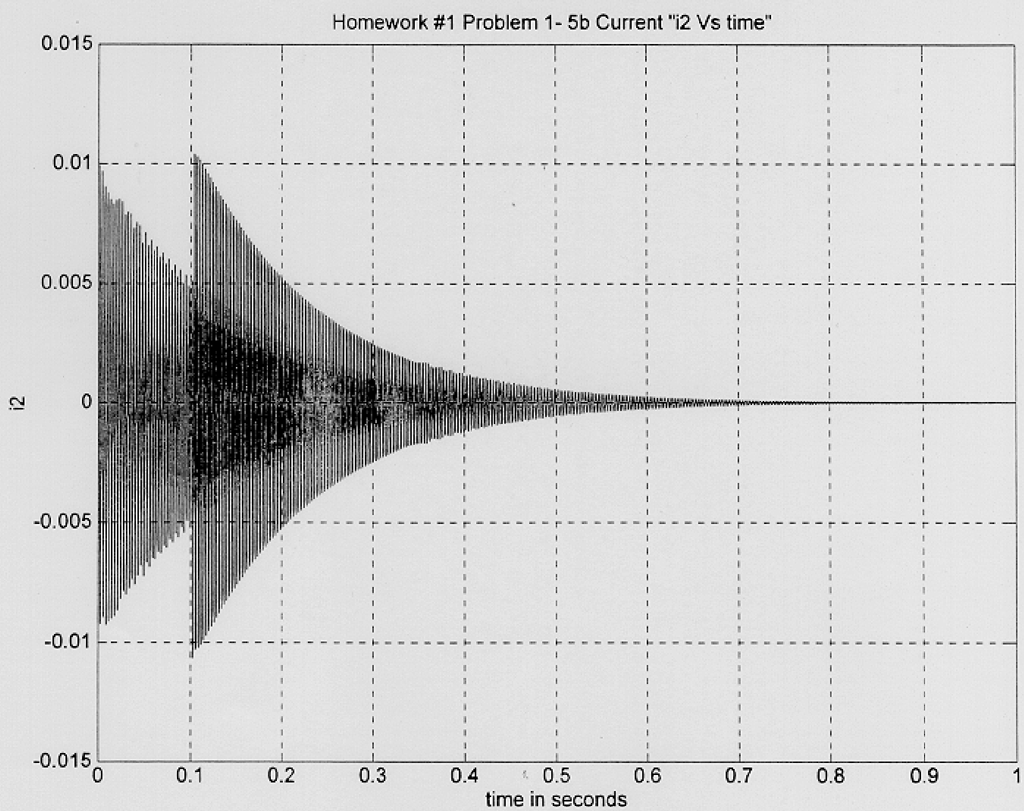
```
elseif abs(flag)==3
    iexec= (x(1) - x(2));

    if iexec <= 4
       Mi=0.1;
    else
       var = 1 + (iexec*iexec);
   Mi= 0.1/var;
   Mi= Mi + 0.0001;
   end

   sys(1:3) = x(1:3);
   sys(4)=Mi;


elseif flag==0   % return initial condition data, sizes and x0
     sys(1)=3;   % number of continuous states
     sys(2)=0;   % number of discrete state
     sys(3)=4;   % number of outputs
     sys(4)=1;   % number of inputs
     sys(5)=0;   % number of discontinous states
     sys(6)=0;

     x0=[0;0;0];
else
     % If flag is anything else, no need to return anything
     % since this is a continous system
     sys = [];
end;
```


time
To Workspace
Clock
10
Constant
asg1pr1
S-Function
Demux
Demux
i1
To Workspace1
i2
To Workspace2
Vc
To Workspace3
Mi
To Workspace4

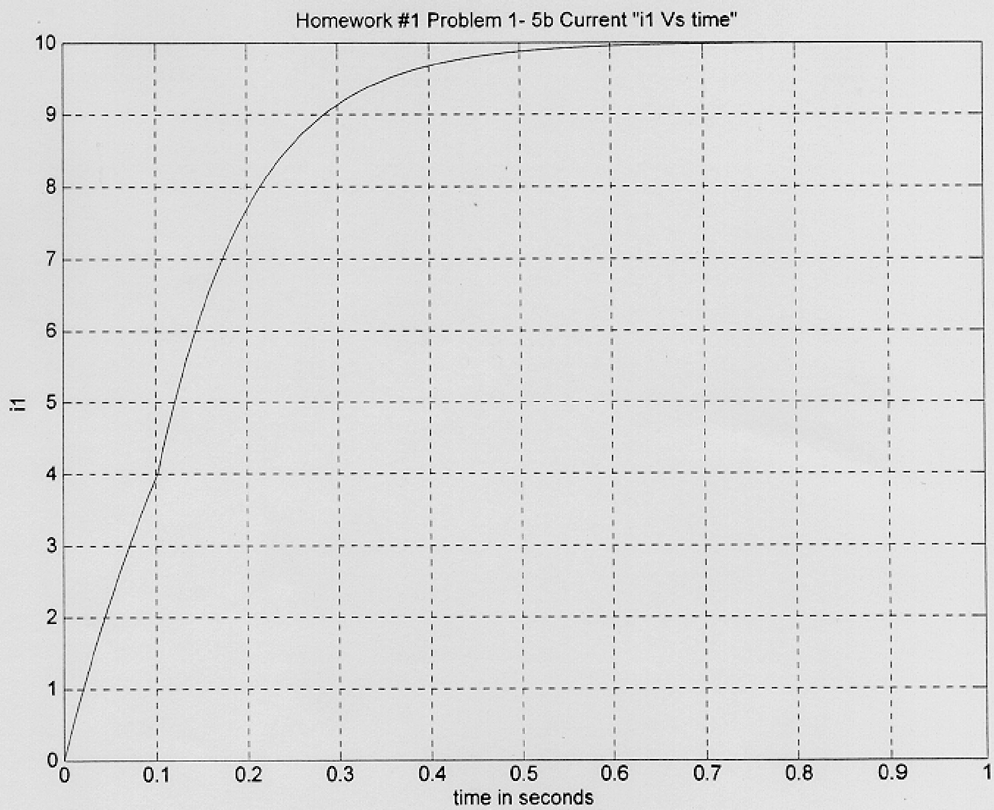

Homework #1 Problem 1- 5b Current "Mi Vs time"
Mi
time in seconds
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Homework #1 Problem 1- 5b Current "Vc Vs time"





Homework #1 Problem 1- 5b Current "i2 Vs time"
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
i2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
time in seconds


Homework #1 Problem 1- 5b Current "i1 Vs time"
i1
time in seconds
0
1
2
3
4
5
6
7
8
9
10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

set #1

100

Excellent

EE 743

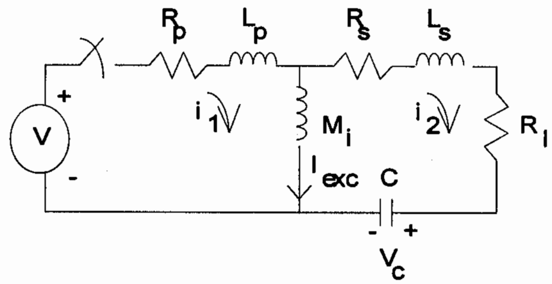
Professor Ali Keyhani

Problem Set #1
Computer Problem #1

Dynamic Response Simulation using Trapezoidal Integration and Simulink

EE743 set #1

I. Problem Statement and Objective: For the system given below:



where: $L_p = 0.1$ H, $L_s = 0.2$ H, $R_p = 1\ \Omega$, $R_s = 2\ \Omega$, $R_l = 1\ \Omega$, $M_i = 0.1$ H, $C = 1\ \mu$F, and $V = 10$ V (Step Input),

1. Write the differential equations for the system.
2. Derive the state-space model for the system.
3. Use trapezoidal integration and compute the dynamic response of i_1, i_2, and V_c and plot the results on the same page.
4. Use Simulink and repeat step 3.
5. Consider the case when M_i is nonlinear and is given by:

$$M_i = 0.1 \text{ H if } I_{exc} < 4 \text{ amps}$$

$$M_i = 0.1/(1+I_{exc}^2) + 0.0001 \text{ H if } I_{exc} > 4 \text{ amps.}$$

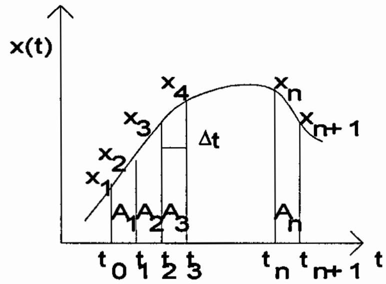
Use Simulink and repeat step 3. Plot i_1, i_2, V_c and M_i.

The objective of this computer problem is to introduce numerical and computer simulation methods used for obtaining the dynamic response from the state-space model of a nonlinear system.

II. Theory: A state-space model of a nonlinear system can be obtained from the differential equations derived from the circuit model representing the system. To obtain the dynamic response of the nonlinear system, some type of numerical method is used.

Trapezoidal integration is one type of numerical technique used to solve for the dynamic response of a nonlinear system. It is an adaptation of Euler's method, in which the area under a curve can be approximated by areas made up of trapezoids.

The following graph displays an arbitrary function with the areas used for this method.



where t is the time, x is the value of the function, and A is the area of the respective trapezoid. The area of the first trapezoid is:

$$A_1 = \Delta t\left(\frac{f(x(t_0),t_0) + f(x_1(t_1),t_1)}{2}\right).$$

Using this equation for each respective area, the value of the function at each increment of time can be found by:

$$x(t_1) = x(t_0) + A_1 = x(t_0) + \frac{\Delta t}{2}\left[f(x(t_0),t_0) + f(x(t_1),t_1)\right]$$

$$x(t_2) = x(t_1) + A_2 = x(t_1) + \frac{\Delta t}{2}\left[f(x(t_1),t_1) + f(x(t_2),t_2)\right]$$

$$\vdots$$

$$x(t_n) = x(t_{n-1}) + A_n = x(t_{n-1}) + \frac{\Delta t}{2}\left[f(x(t_{n-1}),t_{n-1}) + f(x(t_n),t_n)\right]$$

$$\vdots$$

$$x_{n+1} = x_n + A_n = x_n + \frac{\Delta t}{2}\left[f_{n+1} + f_n\right].$$

If a damping term is included with the function, the equation for trapezoidal integration becomes:

$$x_{n+1} = x_n + \frac{\Delta t}{2}\left[(1+\alpha)f_{n+1} + (1-\alpha)f_n\right]$$

where $0 < \alpha < 1$ is the damping factor.

The equations for the state-space model of a general system are given as:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$$
$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du}$$

where x is the state vector, y is the output vector, u is the input vector, and A, B, C, and D represent system matrices. The circuit's differential equations are used to derive the state-space model for the system. From the state-space model, the values for x_{n+1}, x_n, f_{n+1}, and f_n are determined. The value for the time step, Δt, is obtained from the eigenvalues of the system. The eigenvalues of the system are the roots of the characteristic equation for matrix A in the state-space model. The time step is selected as a fraction of the smallest settling time for the system. The smallest settling time corresponds to the system pole having the largest magnitude. The respective equations for the time step and smallest settling time are:

$$\Delta t = \frac{sst}{k},$$

$$sst = \frac{1}{\max\left(\sqrt{\sigma^2 + \omega_d^2}\right)}$$

where Δt is the time step, sst is the smallest settling time, k is a constant usually selected such that $10 < k < 100$, and $\sqrt{\sigma^2 + \omega_d^2}$ is the magnitude of the system's poles.

Simulink is a software tool that operates in conjunction with Matlab. It uses subject oriented blocks to simulate a system, or group of systems. The state-space model of the system, in this problem, is used by simulink to obtain the dynamic response for the system. The simulation is also performed on the system when it has a nonlinear mutual inductance.

III. Sample Computation: Part 1 and 2 of the problem statement are performed as a sample computation for this problem. This involves deriving the differential equations and the state-space model for the given circuit.

Performing KVL around loop 1, the loop consisting of R_p, L_p, M_i, and V:

$$V = R_p i_1 + L_p \frac{di_1}{dt} + M_i \frac{di_1}{dt} - M_i \frac{di_2}{dt}.$$

Likewise, KVL around the loop consisting of i_2 gives:

$$0 = R_s i_2 + L_s \frac{di_2}{dt} + R_l i_2 + V_c + M_i \frac{di_2}{dt} - M_i \frac{di_1}{dt}.$$

Finally, the equation for the current through a capacitor is given as:

$$i_c = i_2 = C \frac{dV_c}{dt}.$$

These three equations are the differential equations for the circuit displayed in the problem statement. Substituting component values into these equations gives:

$$10 = i_1 + 0.1 \frac{di_1}{dt} + 0.1 \frac{di_1}{dt} - 0.1 \frac{di_2}{dt}$$

$$0 = 2i_2 + 0.2 \frac{di_2}{dt} + i_2 + V_c + 0.1 \frac{di_2}{dt} - 0.1 \frac{di_1}{dt}$$

$$i_c = i_2 = 1 \times 10^{-6} \frac{dV_c}{dt}.$$

To derive the system's state-space model, the following states and outputs are first defined:

$$\begin{aligned} x_1 &= y_1 = i_1 \\ x_2 &= y_2 = i_2 \\ x_3 &= y_3 = V_c \end{aligned}$$

where x is the state and y is the output. The equations for the state-space model of a general system are given as:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \end{aligned}$$

where x is the state vector, y is the output vector, u is the input vector, and A, B, C, and D represent system matrices. Substituting the previously defined states into the differential equations for the system results in:

$$
\begin{aligned}
V &= R_p x_1 + L_p \dot{x}_1 + M_i \dot{x}_1 - M_i \dot{x}_2 \\
0 &= R_s x_2 + L_s \dot{x}_2 + R_l x_2 + x_3 + M_i \dot{x}_2 - M_i \dot{x}_1 \\
x_2 &= C \dot{x}_3 .
\end{aligned}
$$

Rearranging terms and representing them in matrix form:

$$
\begin{bmatrix} L_p + M_i & -M_i & 0 \\ -M_i & L_s + M_i & 0 \\ 0 & 0 & C \end{bmatrix}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}
=
\begin{bmatrix} -R_p & 0 & 0 \\ 0 & -\left(R_s + R_l\right) & -1 \\ 0 & 1 & 0 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
+
\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} V
$$

$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}
=
\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} .
$$

The state equation given above is in the general form of:

$$\mathbf{L\dot{x}} = \mathbf{Rx} + \mathbf{Du} .$$

Comparing this form with the state-space equation given for the general system:

$$
\begin{aligned}
&\mathbf{A} = \mathbf{L}^{-1}\mathbf{R} \text{, and} \\
&\mathbf{B} = \mathbf{L}^{-1}\mathbf{D} .
\end{aligned}
$$

Substituting the component values into the state-space equation above, the one with matrix **L**, **R**, and **D**, gives:

$$
\begin{bmatrix} 0.2 & -0.1 & 0 \\ -0.1 & 0.3 & 0 \\ 0 & 0 & 1\times10^{-6} \end{bmatrix}
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}
=
\begin{bmatrix} -1 & 0 & 0 \\ 0 & -3 & -1 \\ 0 & 1 & 0 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
+
\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} V .
$$

The corresponding state-space equation, using matrix A and B, is:

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}
=
\begin{bmatrix} -6 & -6 & 2 \\ -2 & -12 & -4 \\ 0 & 1\times10^{6} & 0 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
+
\begin{bmatrix} 6 \\ 2 \\ 0 \end{bmatrix} V .
$$

IV. Simulation Procedure: Steps 3, 4, and 5 of the problem statement are discussed in this section. To derive the dynamic response for i_1, i_2, and V_c, the state-space model for the system is derived from the states assigned as shown in the sample computation of section III. Initially, the circuit components have constant values. The state-space model representing these constant values are used in the simulation required from step 3 and 4 in the problem statement.

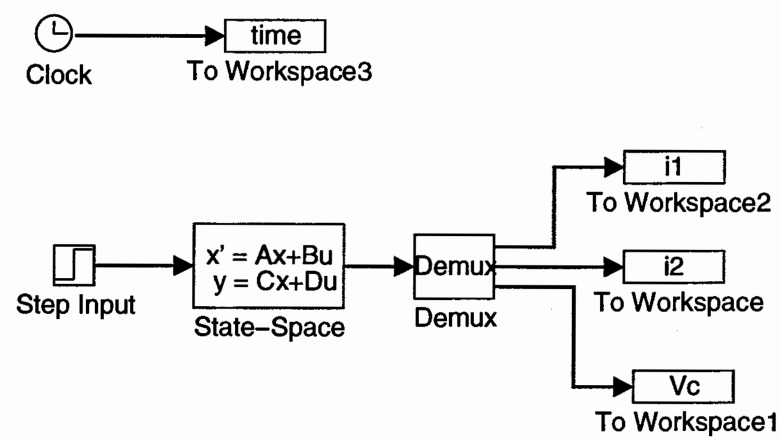
Trapezoidal integration is performed for step 3 of the problem statement. The Matlab program used to obtain the dynamic response of the system using this method is shown in appendix A. The results obtained from this program are discussed in the results section (refer to section V).

The program used for trapezoidal integration is subdivided into various sections. The first few sections establish the circuit components, initial conditions, and various constants. The next two sections establish the **L**, **R**, and **D** matrices (refer to section III, Sample Computation) and solves for the matrices representing the state-space model for the system. The time step is next calculated (refer to section II, Theory). An iterative loop then performs the trapezoid integration and the final section establishes the resultant plots.

Simulink is used to obtain the dynamic response for step 4 of the problem statement. The block diagram used to simulate the dynamic response for the system from its state-space model is displayed on the following page. An initialization program containing the circuit component values, state-space matrices, and time step calculation is shown in appendix A. Constant values needed by Simulink are entered into Matlab using the initialization program. The resultant plots obtained from this simulation are further discussed in the results section (refer to section V).

Both trapezoidal integration and Simulink are used in obtaining the dynamic response for the system in step 5 of the problem statement. The difference between this step and step 3, or 4, is that the mutual inductance, M_i, is made nonlinear. The mutual inductance depends on the excitation current, I_{exc}, which is the difference between i_1 and i_2 (refer to section I, Problem Statement and Objective). Matrix **L**, **A**, and **B** become nonlinear due to the mutual inductance term. Thus, the simulation

STEP 4: SIMULINK BLOCK DIAGRAM.



PARAMETERS:

START TIME = 0.0 SEC
STOP TIME = 0.5 SEC
MIN STEP SIZE = dt
MAX STEP SIZE = dt
TOLERANCE = 1e-6
METHOD: RUNGE-KUTTA 5

methods used earlier, for step 3 and 4, must be modified to account for the nonlinear effect.

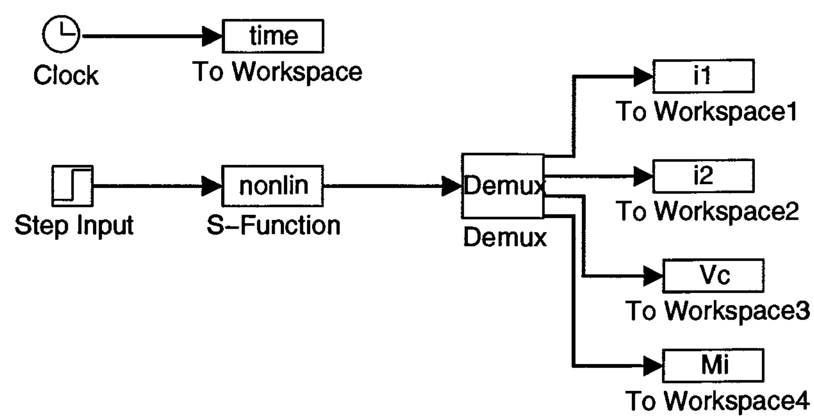
For trapezoidal integration, the nonlinear components (M_i, **L**, **A**, and **B**) must be calculated prior to predicting and correcting each state. Calculating the time step from the smallest settling time implies it changes as well, since the eigenvalues of **A** vary. To prevent this from occurring, the time step is set at a constant 5×10^{-5} seconds. This value approximates the time step achieved in steps 3 and 4 of the problem statement. The simulation program for the trapezoidal integration of the nonlinear system is in appendix A.

The Simulink block diagram used to simulate the nonlinear system is displayed on the following page. The main difference between this block diagram and that previously used in step 4 of the problem statement is with the system block. The previous system used a state-space model block since the system's matrices were constant value. Since the nonlinear system has time varying values, Simulink cannot simulate the system with the state-space model block. Instead, an S-function block is used. The S-function block allows a Matlab function to be used to define the system. The Matlab function, as well as an initialization program and an output file program are in appendix A.

The initialization program is similar to that previously discussed. This program has been modified though to exclude any parameter that is nonlinear. The output file program simply allows formatting of the output plots. This program was used to format the previous Simulink simulation plots. The S-function block performs a function call in Matlab. Constant value parameters are passed to the block as is a flag. The mutual inductance and system model is updated within the function program and the outputs returned upon receiving the applicable flag value.

V. Results: The following pages display the results obtained from this problem in the order they were obtained. The first two plots contain the dynamic response of i_1, i_2, and V_c obtained from trapezoidal integration on the system with a constant mutual inductance. One plot is obtained with a damping coefficient equal to 0.1 whereas the

STEP 5: SIMULINK BLOCK DIAGRAM.



PARAMETERS:

START TIME = 0.0 SEC
STOP TIME = 0.5 SEC
MIN STEP SIZE = 5e-5
MAX STEP SIZE = 5e-5
TOLERANCE = 1e-6
METHOD: RUNGE-KUTTA 5

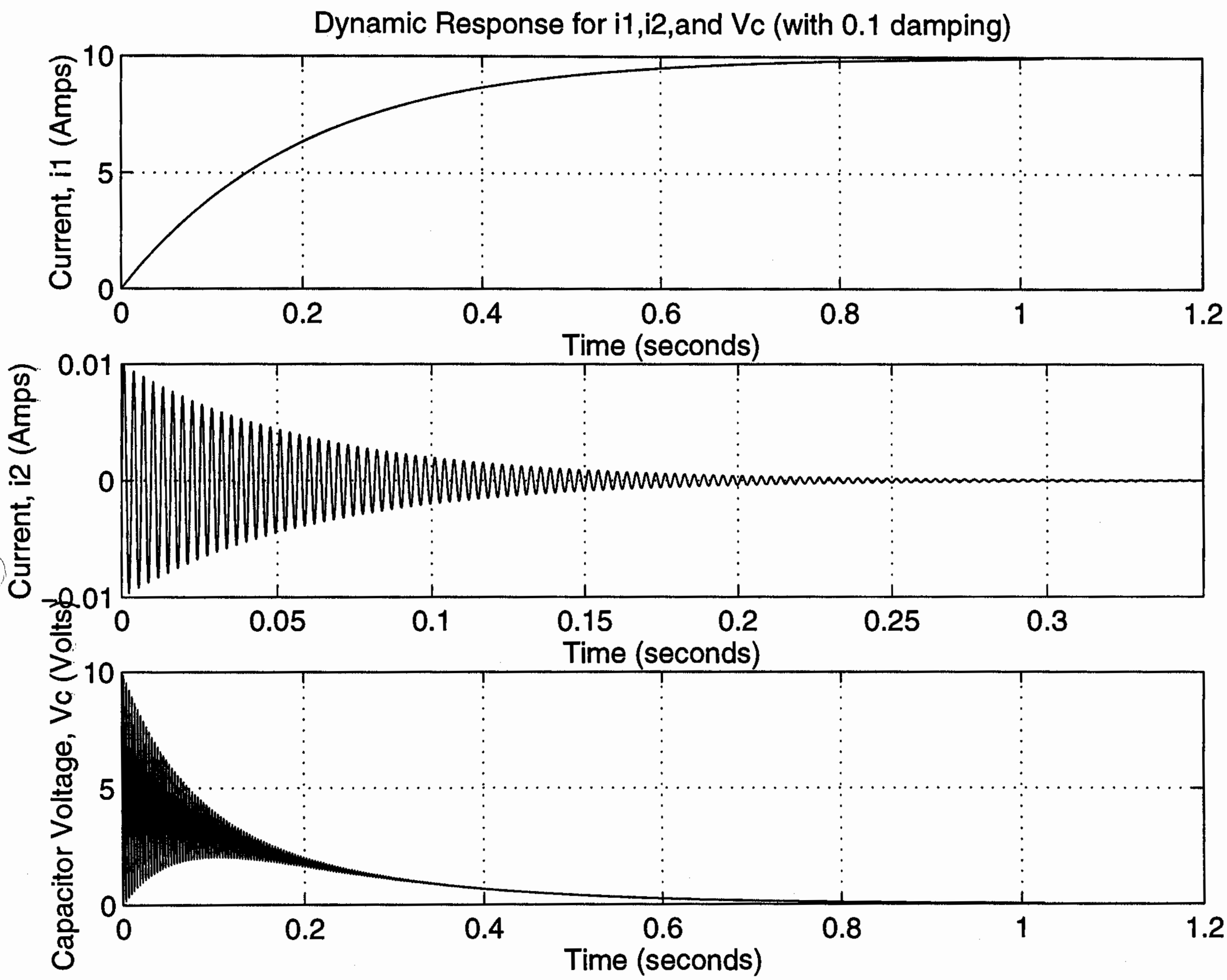
other is plotted for a damping coefficient equal to 0.9. This is done to show the effects of damping. Comparing the two plots, oscillations in the dynamic response die out quicker if the system has a larger value of damping.

The steady state value of i_1 approaches ten amperes as expected. The step input of voltage produces a DC current in the circuit after the switching transient is complete. For a DC current, the inductors in the circuit act as 'shorts' therefore, with R_p equal to one ohm, the current from the voltage source is equal to one ampere.

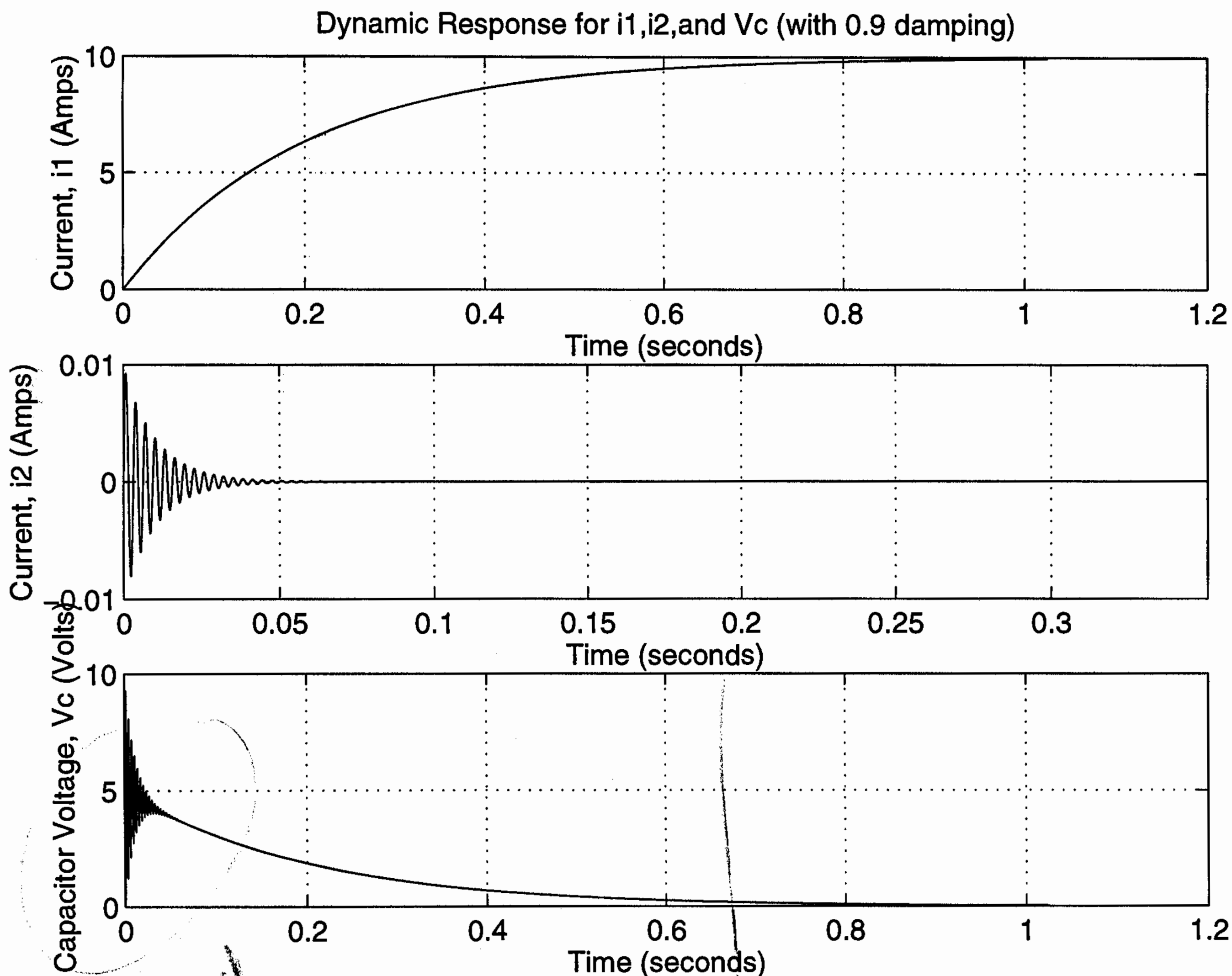
The oscillations occurring between i_2 and V_c occur as energy is transferred between the capacitor and inductors in the loop in which i_2 flows. This part of the circuit acts as a damped resonant tank.

The plot labeled step 4 displays the dynamic response obtained from the Simulink simulation on the system with a constant mutual inductance. Comparing this plot with the two plots obtained from trapezoidal integration, it appears as if little, or no, damping is used in the Simulink simulation. Other than that, these three plots resemble one another.

A drastic change is noted in the plots obtained from trapezoidal integration on the system with nonlinear mutual inductance. Again, one plot is obtained for a 0.1 damping and the other is obtained for a 0.9 damping. Recall, the value of the mutual inductance depended on the excitation current through it which, in turn, depended on the difference between i_1 and i_2. A step change in mutual inductance, of approximately one tenth of a Henry, occurs at one tenth of a second where the excitation current drops below four amperes. This change in mutual inductance causes a distinct change in the dynamic response of i_1, i_2, and V_c. Likewise, the results from the Simulink simulation display the same nonlinear effects.

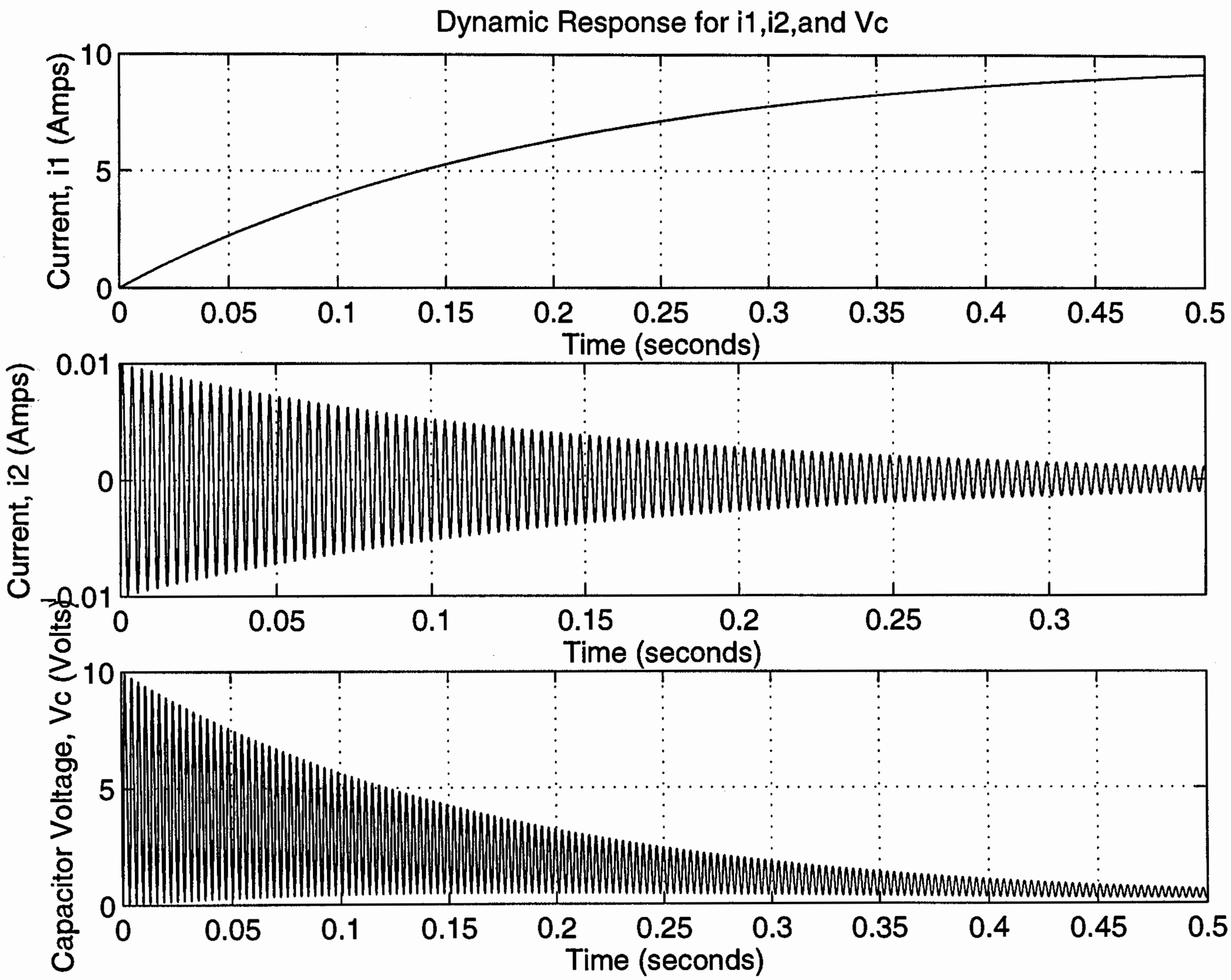


STEP 3: TRAPEZOIDAL INTEGRATION.

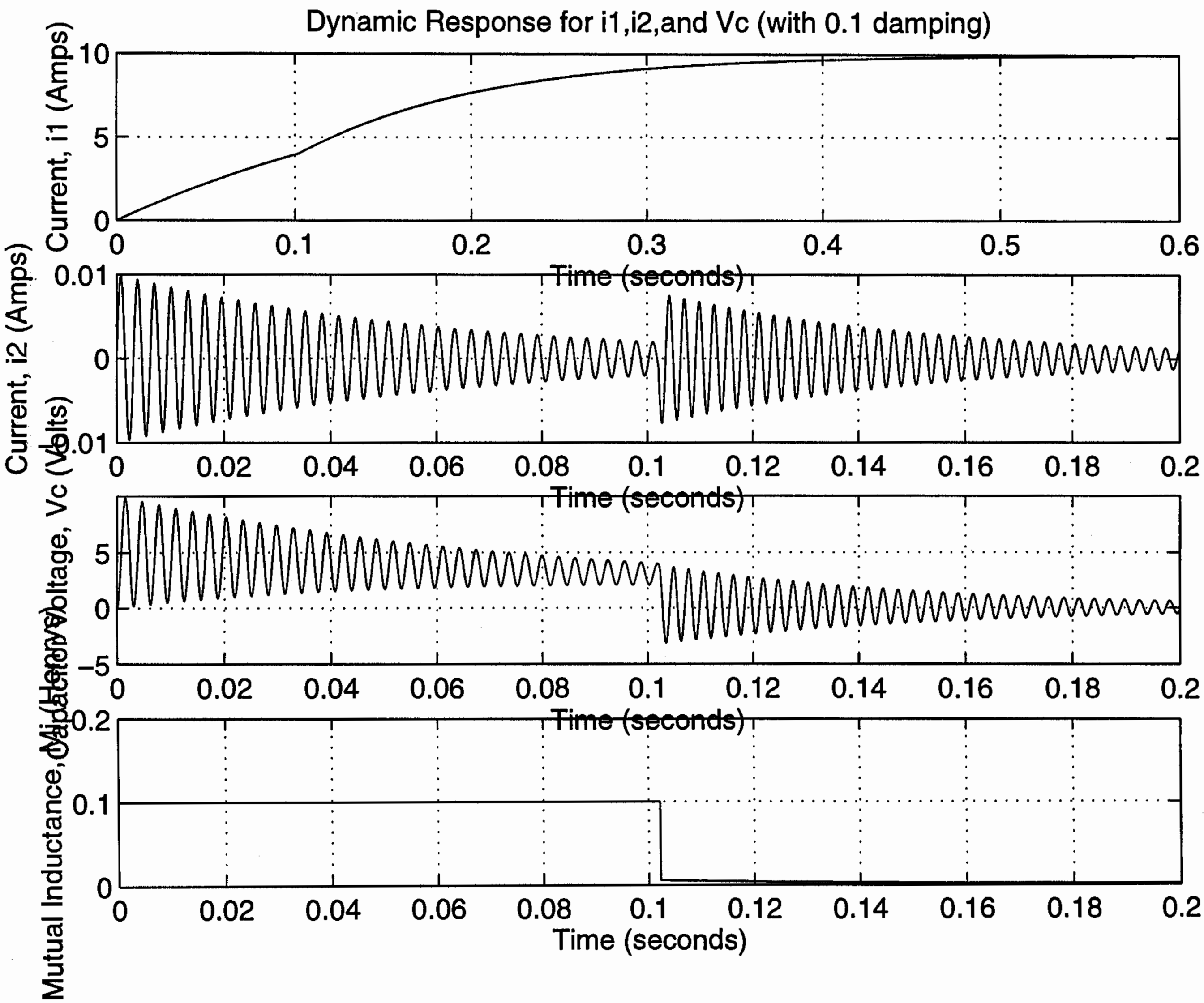


Expand scale here

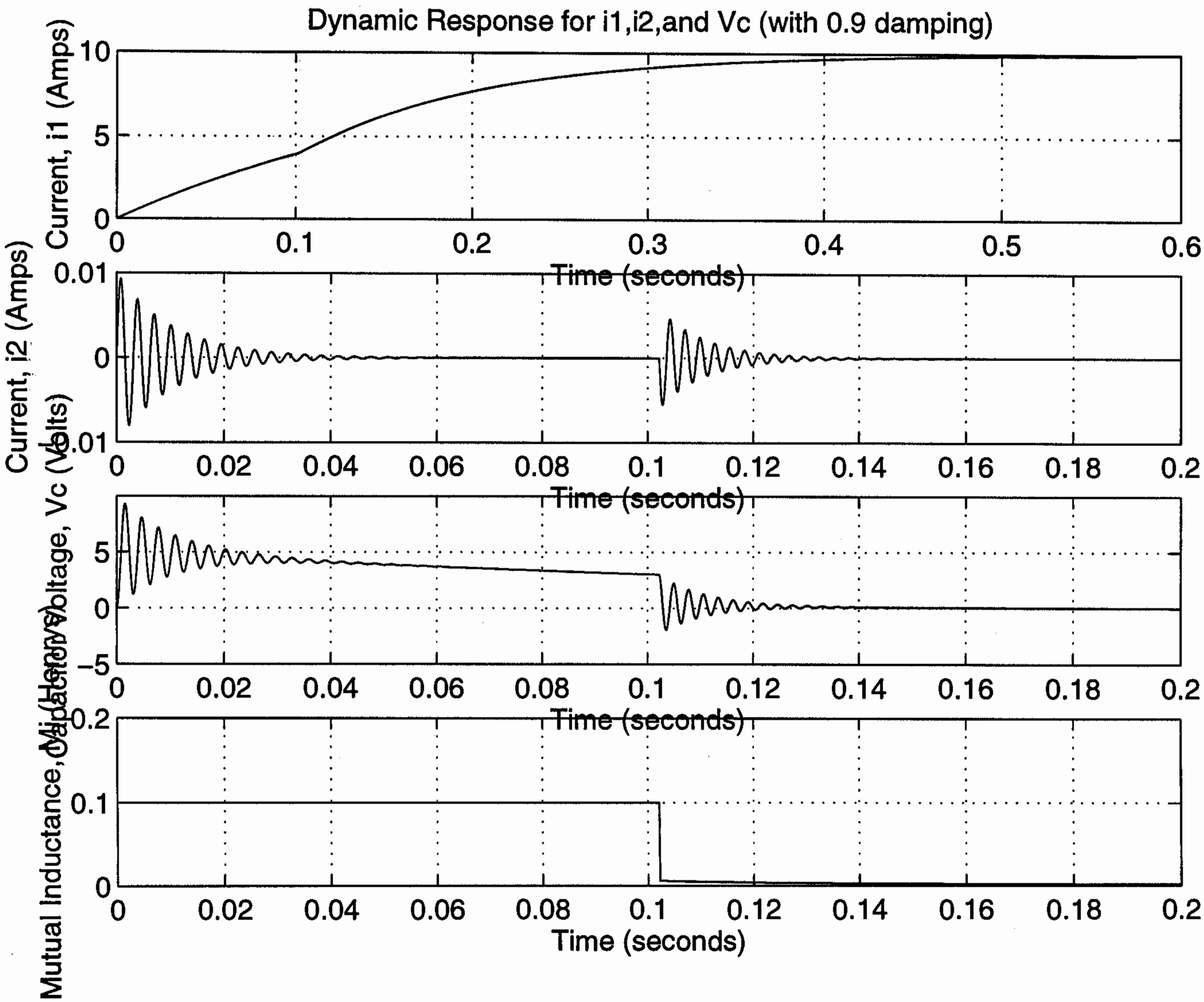
STEP 3: TRAPEZOIDAL INTEGRATION



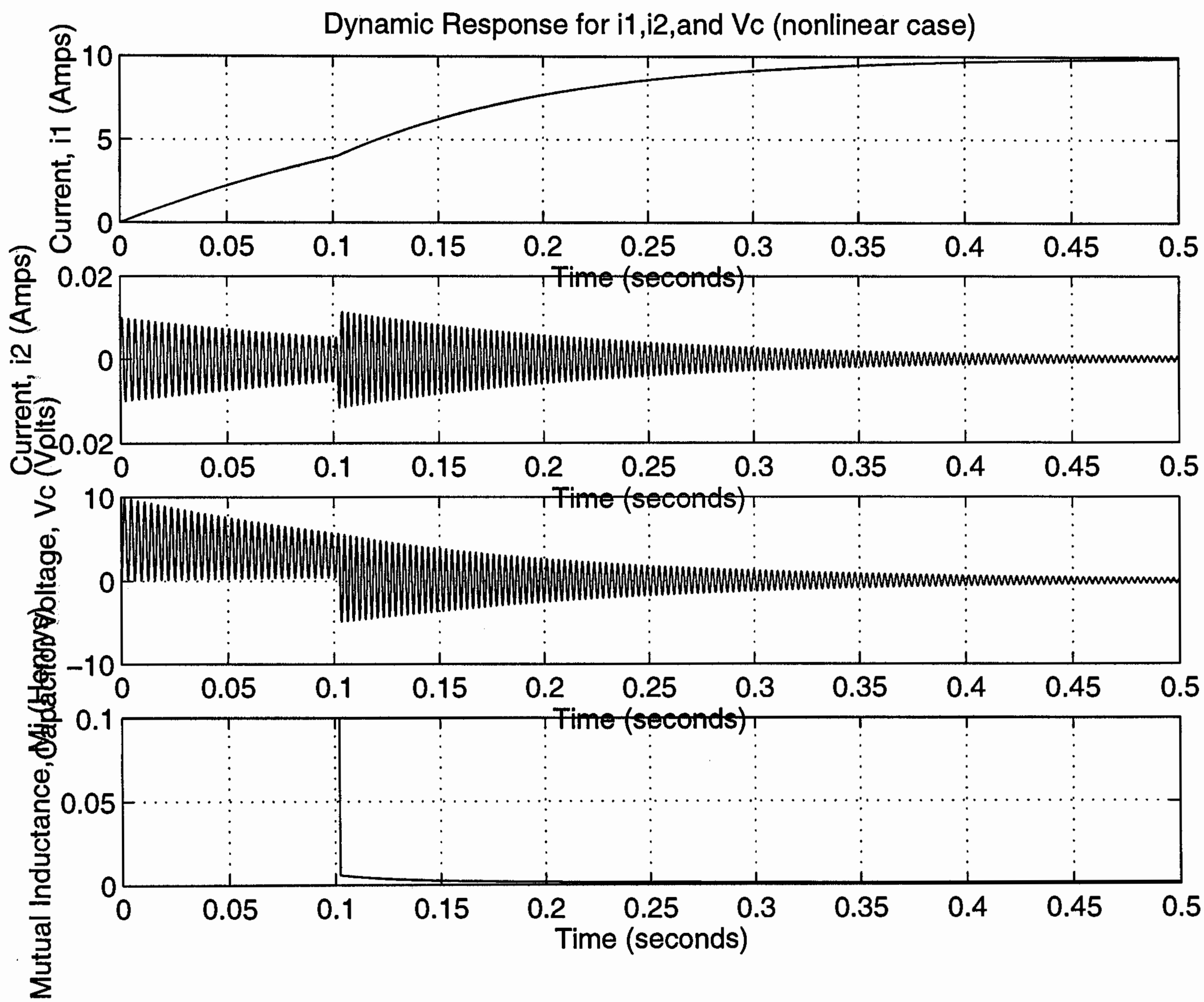
STEP 4: SIMULINK



STEP 5: TRAPEZOIDAL INTEGRATION



STEP 5: TRAPEZOIDAL INTEGRATION.



STEP 5: SIMULINK.

Appendix A

Computer Programs used for Simulating the Dynamic Response of the System.

```
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%   Coy B. Studer      %
%                      %
%   EE743              %
%   15 January 1997    %
                       %
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%   Hw #1              %
%                      %
%   Computer Problem #1 %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%

% This program obtains the dynamic response of a system from its state-space
%  model using trapezoidal integration.

   clear
   format long e

% The circuit components:

  Rp = 1.0;
  Lp = 0.1;
  Rs = 2.0;
  Ls = 0.2;
  Rl = 1.0;
  C = 1E-06;
  Mi = 0.1;

% Initial conditions and constants:

  V = 10;                   % input voltage.
  x = [0;0;0];              % initial value for the state vector.
  k = 10;                   % constant such that 10<k<100 (arbitrary).
  alpha = 0.9;              % damping constant such that 0<alpha<1 (arbitrary).

% Establishing the L, R, and D matrices:

  L = [(Lp+Mi) -Mi 0;-Mi (Ls+Mi) 0;0 0 C];
  R = [-Rp 0 0;0 (-Rs-Rl) -1;0 1 0];
  D = [1;0;0];

% Solving for the A and B matrices:

  A = inv(L)*R;
  B = inv(L)*D;

% Solving for the time step:

  sst = 1/max(abs((eig(A))));             % smallest settling time.
  dt = (sst/k);                           % time step.

% Performing trapezoidal integration (with damping):

for i = 1:20500,                   % iterations.

   xp = x+dt*(A*x+B*V);                                         % predictor.
   x = x+(dt/2)*((1+alpha)*(A*xp+B*V)+(1-alpha)*(A*x+B*V));     % corrector.
```

```
    i1(i) = x(1);
    i2(i) = x(2);
    Vc(i) = x(3);
    time(i) = dt*i;      % forms time matrix.
end;

% To obtain response plots of i1, i2, and Vc:

  subplot(3,1,1),plot(time,i1);

  title('Dynamic Response for i1,i2,and Vc (with 0.9 damping)');
  xlabel('Time (seconds)');
  ylabel('Current, i1 (Amps)');
  grid;

  subplot(3,1,2),plot(time,i2);

  xlabel('Time (seconds)');
  ylabel('Current, i2 (Amps)');
  axis([0 0.35 -0.01 0.01]);
  grid;

  subplot(3,1,3),plot(time,Vc);

  xlabel('Time (seconds)');
  ylabel('Capacitor Voltage, Vc (Volts)')
  grid;

end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%   Coy B. Studer      %
%                      %
%   EE743              %
%   15 January 1997    %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%   Hw #1              %
%                      %
%   Computer Problem #1 %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%

% This program contains the system matrices needed as an initialization file
%  for using Simulink.

   clear
   format long e

% The circuit components:

  Rp = 1.0;
  Lp = 0.1;
  Rs = 2.0;
  Ls = 0.2;
  Rl = 1.0;
  C = 1E-06;
  Mi = 0.1;

% Initial conditions and constants:

  V = 10;                % input voltage.
  x = [0;0;0];           % initial value for the state vector.
  k = 10;                % constant such that 10<k<100.
  alpha = 0.1;           % damping constant such that 0<alpha<1

% Establishing the L, R, and D1 matrices:

  L = [(Lp+Mi) -Mi 0;-Mi (Ls+Mi) 0;0 0 C];
  R = [-Rp 0 0;0 (-Rs-Rl) -1;0 1 0];
  D1 = [1;0;0];

% Solving for the A, B, C, and D matrices:

  A = inv(L)*R;
  B = inv(L)*D1;
  C = eye(size(A));
  D = zeros(size(D1,1),1);

% Solving for the time step (time step is used in Simulink):

  sst = 1/max(abs((eig(A))));           % smallest settling time.
  dt = (sst/k);                         % time step.

end;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Coy B. Studer       %
%                      %
%  EE743               %
%  15 January 1997     %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Hw #1               %
%                      %
%  Computer Problem #1 %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%

% This program obtains the dynamic response of a system from its state-space
%  model using trapezoidal integration (NOTE: Mi is nonlinear for this case).

   clear
   format long e

% The circuit components:

  Rp = 1.0;
  Lp = 0.1;
  Rs = 2.0;
  Ls = 0.2;
  Rl = 1.0;
  C = 1E-06;
  Mconst = 0.1;              % constant portion of Mi.
  dt = 5E-05;                % arbitrary value for the time step.

% Initial conditions and constants:

  V = 10;                    % input voltage.
  x = [0;0;0];               % initial value for the state vector.
  k = 10;                    % constant such that 10<k<100 (arbitrary).
  alpha = 0.1;               % damping constant such that 0<alpha<1 (arbitrary).

% Establishing the L, R, and D matrices:

  R = [-Rp 0 0;0 (-Rs-Rl) -1;0 1 0];
  D = [1;0;0];

% Performing trapezoidal integration (with damping):

  for i = 1:11500,                   % iterations.

% Determine Mi (Mi depends on lexc = i1-i2 = x(1)-x(2)):

    if (x(1)-x(2))<=4
      Mi = Mconst;
    else
      Mi = Mconst/(1+(x(1)-x(2))^2)+0.0001;
    end

% Determine the L, A, and B matrices from Mi:

    L = [(Lp+Mi) -Mi 0;-Mi (Ls+Mi) 0;0 0 C];
    A = inv(L)*R;
```

```
   B = inv(L)*D;

% Solve the predictor equation:

   xp = x+dt*(A*x+B*V);

% Need to determine Mi, L, A, and B from the predictor values:

   if (xp(1)-xp(2))<=4
     Mi = Mconst;
   else
     Mi = Mconst/(1+(xp(1)-xp(2))^2)+0.0001;
   end

   L = [(Lp+Mi) -Mi 0;-Mi (Ls+Mi) 0;0 0 C];
   A = inv(L)*R;
   B = inv(L)*D;

% Solve the predictor equation:

   x = x+(dt/2)*((1+alpha)*(A*xp+B*V)+(1-alpha)*(A*x+B*V));    % corrector.

   i1(i) = x(1);
   i2(i) = x(2);
   Vc(i) = x(3);
   Mi(i) = Mi;
   time(i) = dt*i;      % forms time matrix.
end;

% To obtain response plots of i1, i2, and Vc:

 subplot(4,1,1),plot(time,i1);

 title('Dynamic Response for i1,i2,and Vc (with 0.1 damping)');
 xlabel('Time (seconds)');
 ylabel('Current, i1 (Amps)');
 grid;

 subplot(4,1,2),plot(time,i2);

 xlabel('Time (seconds)');
 ylabel('Current, i2 (Amps)');
 axis([0 0.4 -0.01 0.01]);
 grid;

 subplot(4,1,3),plot(time,Vc);

 xlabel('Time (seconds)');
 ylabel('Capacitor Voltage, Vc (Volts)')
 axis([0 0.4 -5 10]);
 grid;

 subplot(4,1,4),plot(time,Mi);

 xlabel('Time (seconds)');
 ylabel('Mutual Inductance, Mi (Henrys)')
 axis([0 0.3e-03 0 1.5E-03]);
 grid;

end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Coy B. Studer       %
%                      %
%  EE743               %
%  15 January 1997     %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Hw #1               %
%                      %
%  Computer Problem #1 %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%

% This program contains the system matrices needed as an initialization file
%  for using Simulink (with the nonlinear mutual inductance term).

   clear
   format long e

% The circuit components, initial conditions, and constant matrices:

  Rp = 1.0;
  Lp = 0.1;
  Rs = 2.0;
  Ls = 0.2;
  Rl = 1.0;
  C = 1E-06;
  Mconst = 0.1;            % constant part of Mi.

  V = 10;                  % input voltage.
  x = [0;0;0];             % initial value for the state vector.
  alpha = 0.1;             % damping constant such that 0<alpha<1.

  R = [-Rp 0 0;0 (-Rs-Rl) -1;0 1 0];
  D1 = [1;0;0];

end;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Coy B. Studer       %
%                      %
%  EE743               %
%  15 January 1997     %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Hw #1               %
%                      %
%  Computer Problem #1 %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%

% This program contains the system function (M-file) used by Simulink for the
%   nonlinear S-function.

function[sys,x0]=nonlin(t,x,u,flag,Lp,Ls,C,R,D1,Mconst)

  format long e

  if nargin~=10
  if nargin==0
  flag=0;
  else
  end
  end

  if abs(flag)==1          % state derivatives returned.

  if (x(1)-x(2))<=4;       % determine nonlinear value of Mi.
    Mi = Mconst;
  else
    Mi = Mconst/(1+(x(1)-x(2))^2)+0.0001;
  end

% Modifying the Mi dependent matrices.

  L = [(Lp+Mi) -Mi 0;-Mi (Ls+Mi) 0;0 0 C];
  A = inv(L)*R;
  B = inv(L)*D1;

  sys = A*x+B*u;           % state-space model.

  elseif abs(flag)==3      % state outputs returned.

  if (x(1)-x(2))<=4;               % determine nonlinear value of Mi.
    Mi = Mconst;
  else
    Mi = Mconst/(1+(x(1)-x(2))^2)+0.0001;
  end

  sys(1) = x(1);
  sys(2) = x(2);
  sys(3) = x(3);
  sys(4) = Mi;

  elseif flag==0           % initial condition data and sizes returned.
```

```
  sys(1) = 3;            % number of continuous states.
  sys(2) = 0;            % number of discrete states.
  sys(3) = 4;            % number of outputs.
  sys(4) = 1;            % number of inputs.
  sys(5) = 0;            % number of discontinuous states.
  sys(6) = 0;

  x0 = zeros(3,1);       % the initial state vector.

  else
  sys = [];

end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Coy B. Studer       %
%                      %
%  EE743               %
%  15 January 1997     %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%
%                      %
%  Hw #1               %
%                      %
%  Computer Problem #1 %
%                      %
%%%%%%%%%%%%%%%%%%%%%%%%

% This program obtains the desired output plots from simulink.

% To obtain response plots of i1, i2, and Vc:

  subplot(3,1,1),plot(time,i1);

  title('Dynamic Response for i1,i2,and Vc (nonlinear case)');
  xlabel('Time (seconds)');
  ylabel('Current, i1 (Amps)');
  grid;

  subplot(3,1,2),plot(time,i2);

  xlabel('Time (seconds)');
  ylabel('Current, i2 (Amps)');
%  axis([0 0.35 -0.01 0.01]);
  grid;

  subplot(3,1,3),plot(time,Vc);

  xlabel('Time (seconds)');
  ylabel('Capacitor Voltage, Vc (Volts)')
  grid;

end
```