Appendix

Matlab Code for the Computation of Controlled Trajectories and Steering Control of MSOL System

```
%-----------------------------------------------------------------------%
%   System Considered : x" + A^2 x = Bu
%   A : Constant Matrix of order n x n
%   B : Constant Vector of size n
%   m : parameter taken as mentined in Algorithm 3.5.1
%   n : order of A
%   x0 : Initial State - column vector of size n
%   y0 : Initial Valocity - column vector of size n
%   x1 : Final State - column vector of size n
%   T : Final time (system should reach x1 from x0 in interval [0,T])
%   W_0_T : Controllability Grammian
%   sinnew() and cosinenew() : Functions computing Matrix Sine and
%       Cosine using Pade Approximation (developed inhouse)
%   intt() : function to compute integration numerically
%-----------------------------------------------------------------------%
```

```
while(1)
    clear  clc
    disp('Linear System x'' + A^2 x = Bu where A and B are given
          as follows:')
    A = input('Enter the matrix A');
    B = input('Enter the column vector B');
    m = input('Enter the value of m');
    n = input('Enter the order of the matrix');
    d = det(A);
    pause
% Computation of Controllability Matrix:
    disp('The controllability Matrix of the system is:')
    C = B;
    Q = B;
    AS = A^2;
    for i=1:n-1
        C = (AS^i)*B
        Q = [Q C]
    end
    pause
% The Rank of the Controllability Matrix :
    disp('The rank of the controllability matrix is:')
    r = rank(Q)
 if(r == n)% Checking necessary condition
% Input intitial and final state
    disp('The initial state is:')
    x0 = input('Enter the initial state column vector');
```

```
    y0 = input('Enter the initial velocity column vector');
    pause
    disp('The final state is:')
    x1 = input('Enter the final state column vector');
    pause
    disp('We want to to reach the final state in time')
    T  = input('Enter the time to reach the final state');
    pause
% The Controllability Grammian :
    W_0_T =intt('inv(A)*sinenew(A*(t-s),3,10)*B*transpose(B)*
        transpose(inv(A)*sinenew(A*(t-s),3,10))',A,B,n,T,0,0);
    i=0;
% The Solution and the steering Control of MSOL :
    for t = 0:0.01:T
        i = i+1;
        ss =intt('inv(A)*sinenew(A*(t-s),3,10)*B*transpose(B)*
            transpose(inv(A)*sinenew(A*(2-  s),3,10))',A,B,n,t,0,0);
        u(:,i) = B'*(inv(A)*sinenew(A*(T-t),3,10))'*inv(W_0_T)*
            (x1 - cosinenew(A*T,3,10)*x0 -inv(A)*sinenew(A*T,3,10)*y0);
        x(:,i) = cosinenew(A*t,3,10)*x0 + inv(A)*sinenew(A*t,3,10)*y0
            + ss*inv(W_0_T)*(x1 - cosinenew(A*T,3,10)*x0 -inv(A)*
            sinenew(A*T,3,10)*y0);
    end
% Plotting the graph of the solution and the steering control :
    t = 0:0.01:T;
    subplot(2,1,1), plot(t,x(1,:),'r')
    hold on
```

```
    subplot(2,1,1), plot(t,x(2,:),'b')
    hold on
    subplot(2,1,1), plot(t,x(3,:),'g')
    hold off
    xlabel('TIME  t')
    ylabel('STATE  x(t)')
    title('CONTROLLED TRAJECTORIES OF THE "MSOL" SYSTEM')
    grid on
    subplot(2,1,2), plot(t,u)
    grid on
    xlabel('TIME  t')
    ylabel('STEERING CONTROL  u(t)')
    title('THE GRAPH OF THE STEEERING CONTROL')
  else
    ans= input ('The system is not controllable. Would you like
          to try other system (y/n)?')
    if(ans == 'y')
         continue;
    else
        break;
    end
end
```

Matlab Code for the Computation of Controlled Trajectories and Steering Control of MSON System

```
%------------------------------------------------------------------%
%   System Considered : x" + A^2 x = Bu + f(t,x)
%   A : Constant Matrix of order n x n
%   B : Constant Vector of size n
%   f : Nonlinear function
%   m : parameter taken as mentined in Algorithm 3.5.1
%   n : order of A
%   x0 : Initial State - column vector of size n
%   y0 : Initial Valocity - column vector of size n
%   x1 : Final State - column vector of size n
%   T : Final time (system should reach x1 from x0 in interval [0,T])
%   W_O_T : Controllability Grammian
%   intt() : function to compute integration numerically
%------------------------------------------------------------------%
    while(1)
    clear
    clc
    disp('Nonlinear System x'' + A^2 x = Bu + f(t,x) where A and B
          are given as follows:')
    A = input('Enter the matrix A');
    B = input('Enter the column vector B');
    m = input('Enter the value of m');
```

```
    n = input('Enter the order of the matrix');
    d = det(A);
    pause
% Computation of Controllability Matrix:
    disp('The controllability Matrix of the system is:')
    C = B;
    Q = B;
    AS = A^2;
    for i=1:n-1
        C = (AS^i)*B
        Q = [Q C]
    end
    pause
% The Rank of the Controllability Matrix :
    disp('The rank of the controllability matrix is:')
    r = rank(Q)

    if(r == n) % Checking necessary condition
% Input intitial and final state
        disp('The initial state is:')
        x0 = input('Enter the initial state column vector');
        y0 = input('Enter the initial velocity column vector');
        pause
        disp('The final state is:')
        x1 = input('Enter the final state column vector');
        pause
        disp('We want to to reach the final state in time')
```

```
        T  = input('Enter the time to reach the final state');
        pause
% The Controllability Grammian :
        W_0_T = intt('inv(A)*cosnew(A*(t-s),n,m)*B*transpose(B)*
                transpose(inv(A)*funm(A*(t-s),@sin))',A,B,n,T,0,0);
        x = x0;
        for t = 0:0.01:T-0.01
            x=[x x0];
        end
        xold = zeros(size(x));
        k1=0;
        while norm (x - xold) >= 0.001
            xold = x;
% Nonlinear Function
            f =[sin(xold(1,:))/90;cos(xold(2,:))/89;xold(3,:)/88];
            k1 = k1+1;
            i=0;
% The Solution and the steering Control of MSON :
            for t =  0:0.01:T
                i = i+1;
                ss =
                intt('inv(A)*funm(A*(1-s),@sin)*x(:,k)',A,B,n,T,f,0);
                u(:,i) = B'*(inv(A)*funm(A*(T-t),@sin))'*inv(W_0_T)*
                 (x1 - funm(A*T,@cos)*x0 - inv(A)*funm(A*T,@sin)*y0-ss);
            end
            i=0;
            for t=0:0.01:T
```

```
            i=i+1;
            aa1 =
            intt('inv(A)*funm(A*(t-s),@sin)*B*u(:,k)',A,B,n,t,0,u);
            aa2 =intt('inv(A)*funm(A*(t-s),@sin)*x(:,k)',A,B,n,t,f,0);
            x(:,i) = funm(A*t,@cos)*x0 + inv(A)*funm(A*t,@sin)*y0 +
    aa1 + aa2;
        end
        x
    end
    k1
% Plotting the graph of the solution and the steering control
    t=  0:0.01:T
    for k=1:n
        plot(t,x(k,:),'k')
        hold on
    end
    hold off
    grid on
    xlabel('TIME t')
    ylabel('STATE x(t)')
    title('CONTROLLED TRAJECTORIES OF "MSON" SYSTEM')
    figure
    grid on
    plot(t,u)
    xlabel('TIME t')
    ylabel('STEERING CONTROL u(t)')
    title('THE GRAPH OF THE STEEERING CONTROL')
```

```
    else
    ans= input ('The system is not controllable. Would you like
     to try other system (y/n)?'  )
    if(ans == 'y')
        continue;
    else
        break;
    end
end
```

Matlab Code for the Computation of Controlled Trajectories and Steering Control of Linear System Using Spectral Mathod

```
%-------------------------------------------------------------------%
%   System Considered : x' + A x = Bu
%   A : Constant Matrix of order n x n
%   B : Constant Vector of size n
%   n : order of A
%   x0 : Initial State - column vector of size n
%   y0 : Initial Valocity - column vector of size n
%   x1 : Final State - column vector of size n
%   T : Final time (system should reach x1 from x0 in interval [0,T])
%   W: Controllability Grammian
%   intt() : function to compute integration numerically
%-------------------------------------------------------------------%
clear
clc
% Initialization
A = [1 2 1;3 1 0;1 1 0];
B = [1;1;0]
n = size(B);
% Computation of Controllability Matrix:
mat = [B A*B (A^2)*B]
% The Rank of the Controllability Matrix :
rank(mat)
```

```
% Intitial and final states:
x0 = [-1 1 0]'
x1 = [1 -1 2]'
T  = 4
% The Controllability Grammian :
W = intt('expm(A*(t-s))*B*transpose(B)*transpose(expm(A*(t-s)))',
    A,B,n(1),T,0,0)
% The Eigen values and Eigen Vectors of Controllability Grammian:
[v L] = eig(W)
c = inv(v)*(x1 - expm(A*T)*x0)
sum = zeros(size(B))
for i = 1:n(1)
      sum = sum + (c(i)*v(:,i))/L(i,i)
end
i=0
% The Solution and the steering Control of the system:
for t = 0:0.01:T
     i = i+1;
     j = 0;
     for s = 0:0.01:t
         j = j+1;
         u(:,j) = B'*expm(A*(T-s))'*sum;
     end
     aa = intt('expm(A*(t-s))*B*u(:,k)',A,B,n(1),t,0,u);
     x(:,i) = expm(A*t)*x0 + aa;
 end
% Plotting the graph of the solution:
```

```
t = 0:0.01:T

plot(t,x(1,:),'r')

hold on

plot(t,x(2,:),'b')

hold on

plot(t,x(3,:),'g')

xlabel('TIME  t')

ylabel('STATE  x(t)')

title('CONTROLLED TRAJECTORIES OF THE LINEAR SYSTEM')

grid on
```

Matlab Code for the Computation of Controlled Trajectories and Steering Control of Nonlinear System Using Spectral Method

```
%-----------------------------------------------------------------------%
%   System Considered : x' + A x = Bu + f(t, x)
%   A : Constant Matrix of order n x n
%   B : Constant Vector of size n
%   n : order of A
%   x0 : Initial State - column vector of size n
%   y0 : Initial Valocity - column vector of size n
%   x1 : Final State - column vector of size n
%   T : Final time (system should reach x1 from x0 in interval [0,T])
%   W: Controllability Grammian
%   intt() : function to compute integration numerically
%-----------------------------------------------------------------------%
clear
clc
% Initialization
A = [1 2 1;3 1 0;1 1 0];
B = [1;1;0];
n = size(B);
% Computation of Controllability Matrix:
mat = [B A*B (A^2)*B];
% The Rank of the Controllability Matrix :
rank(mat)
```

```
% Intitial and final states:
x0 = [-1 1 0]';
x1 = [0 -1 1]';
T  = 1;
% The Controllability Grammian :
W = intt('expm(A*(t-s))*B*transpose(B)*transpose(expm(A*(t-s)))',
    A,B,n(1),T,0,0);
% The Eigen values and Eigen Vectors of Controllability Grammian:
[v L] = eig(W);
x = x0;
for t=0:0.01:T-0.01
    x=[x x0];
end
xold = zeros(size(x));
k1=0;
while norm(x - xold) >= 0.001
    xold = x;
% The nonlinear function:
    f = [sin(xold(1,:))/90;cos(xold(2,:))/89;xold(3,:)/88];
    k1 = k1+1;
    i=0;
% The Solution and the steering Control of the system:
    for t = 0:0.01:T
        i = i+1;
        ss = intt('expm(A*s)*x(:,k)',A,B,n(1),T,f,0);
        c = inv(v)*(x1 - expm(A*T)*x0 - ss);
        sum = zeros(size(B));
```

```
        for i = 1:n(1)
            sum = sum + (c(i)*v(:,i))/L(i,i);
        end
        j = 0;
        for s = 0:0.01:t
           j = j+1;
           u(:,j) = B'*expm(A*(T-s))'*sum;
        end
     end
     i=0;
     for t=0:0.01:T
       i=i+1;
       aa1 = intt('expm(A*(t-s))*B*u(:,k)',A,B,n(1),t,0,u);
       aa2 = intt('expm(A*(t-s))*x(:,k)',A,B,n(1),t,f,0);
       x(:,i) = expm(A*t)*x0 + aa1 + aa2;
     end
    x
end
% Plotting the graph of the solution:
t = 0:0.01:T
plot(t,x(1,:),'r')
hold on
plot(t,x(2,:),'b')
hold on
plot(t,x(3,:),'g')
xlabel('TIME  t')
ylabel('STATE  x(t)')
```

```
title('CONTROLLED TRAJECTORIES OF THE NONLINEAR SYSTEM')
grid on
hold off
```