

Flexible Link with Rotation and Translation

```
clear all; close all;

% Parameters Definition
L = 1.1; % 1.1 m
EI = 2.1*1.167;
zho = 0.4436;
rA = 0.05; % radius of the base
MA = 50;
IH = 0.5*50*(rA^2); % inertia of the base
a=1;
c = sqrt(a)*(L^3/3+IH/zho)^(-1/2);
```

```
% desired angle phi
phi_des = pi/4;
```

```
% get eigenfunctions
omegas = [];
ks = [];
inits = [];
total_N_limit = 8;
for init=0:0.5:10
    syms k real;
    eqn = IH/zho*k^3*(1+cos(k*L)*cosh(k*L))+(sin(k*L)*cosh(k*L)-cos(k*L)*sinh(k*L)) == 0;
    k = vpasolve(eqn,init*pi);
    omega = sqrt(EI/zho*k^4);

    if k<0
        continue
    elseif any(ks==double(round(k,8)))
        continue
    end
    ks = [ks double(round(k,8))];
    omegas = [omegas double(round(omega,8))];
    inits = [inits init];
    if length(omegas)>=total_N_limit
        break
    end
end
qD = length(ks);

syms l cn real;
sci_0 = c*l;
Ann=[0];
Bnn=[0];
Cnn=[c];
Dnn=[0];
for i=2:length(ks)
    k=ks(i);
    dn=-cn;
    an=(cos(k*L)*cosh(k*L)+sin(k*L)*sinh(k*L)+1)/(-sin(k*L)*cosh(k*L)+cos(k*L)*sinh(k*L));
```

```

bn=2*zho/(IH*k^3)*cn-an;
sci = cn*cos(k*l) + dn*cosh(k*l) + an*sin(k*l) + bn*sinh(k*l);
sci_p_0 = an*k + bn*k;

fcfn=int((sci)^2,l,0,L)+IH/zho*sci_p_0*sci_p_0 == a;
cnn = vpasolve(fcn,cn,1);
cnn = double(cnn(2));
dnn = -cnn;
ann = (cos(k*L)*cosh(k*L)+sin(k*L)*sinh(k*L)+1)/(-sin(k*L)*cosh(k*L)+cos(k*L)*sinh(k*L));
bnn = 2*zho/(IH*k^3)*cnn-ann;

Cnn = [Cnn cnn];
Dnn = [Dnn dnn];
Ann = [Ann ann];
Bnn = [Bnn bnn];

end

```

```

% system
CL=[];
for i=1:qD
    if i==1
        CL = [CL 0.5*Cnn(i)*L^2];
        continue
    end
    CL = [CL sci_int_fcn(Ann(i),Bnn(i),Cnn(i),Dnn(i),ks(i),L)];
end
beta = [];
for i=1:qD
    if i==1
        beta = [beta;Cnn(i)/zho];
        continue
    end
    beta = [beta;sci_p_fcn(Ann(i),Bnn(i),Cnn(i),Dnn(i),ks(i),0)/zho];
end
M = [eye(qD) CL.';CL (MA+zho*L)/zho];
Omega = diag(omegas);
A_ = -inv(M)*[Omega zeros(qD,1);zeros(1,qD) 0];
B_ = inv(M)*[beta zeros(qD,1);0 1/zho];

A = [zeros(qD,qD) eye(qD) zeros(qD,1) zeros(qD,1);...
    A_(1:qD,1:qD) zeros(qD,qD) A_(1:qD,end) zeros(qD,1);...
    zeros(1,qD) zeros(1,qD) 0 1;...
    A_(end,1:qD) zeros(1,qD) A_(end,end) 0];
B = [zeros(qD,1) zeros(qD,1);...
    B_(1:qD,1) B_(1:qD,end);...
    0 0;...
    B_(end,1) B_(end,end)];
C = zeros(1,2*qD+2);
C(1) = 1;
flex_sys = ss(A,B,C,0);

```

```

% feedback PD system

```

```

q_des = phi_des/c;
Kp = .1; Kd = .1;
kp1 = 100; kv1=80; kp2=.1; kv2=.1;
K = [Kp*beta.' Kd*beta.' 0 0;...
     -kp1*CL -kv1*CL kp2 kv2];
q0 = zeros(2*qD+2,1);
q0(1) = -q_des;
[ts,qout] = ode45(@(ts,q) odefun(ts,q,A,B,K), 0:0.02:10, q0);

u = K*qout';
qout(:,1) = qout(:,1)+q_des;
l_sample=0:0.01:L;
x_beam=[];y_beam=[];
x_nominal=[];y_nominal=[];
e = [];
target_x = L*cos(phi_des);target_y = L*sin(phi_des);
for i=1:length(ts)
    [w,phi] = q_wphi(qout(i,1:qD),ks,l_sample,Ann,Bnn,Cnn,Dnn);

    zT = qout(i,2*qD+1);
    x_a = -zT*sin(phi);
    y_a = zT*cos(phi);

    [x,y,xb,yb]=wphi_xy(w,phi,l_sample);
    x = x+x_a; y=y+y_a;
    xb = xb+x_a; yb=yb+y_a;
    x_beam=[x_beam;x]; y_beam=[y_beam;y];
    x_nominal=[x_nominal;xb]; y_nominal=[y_nominal;yb];

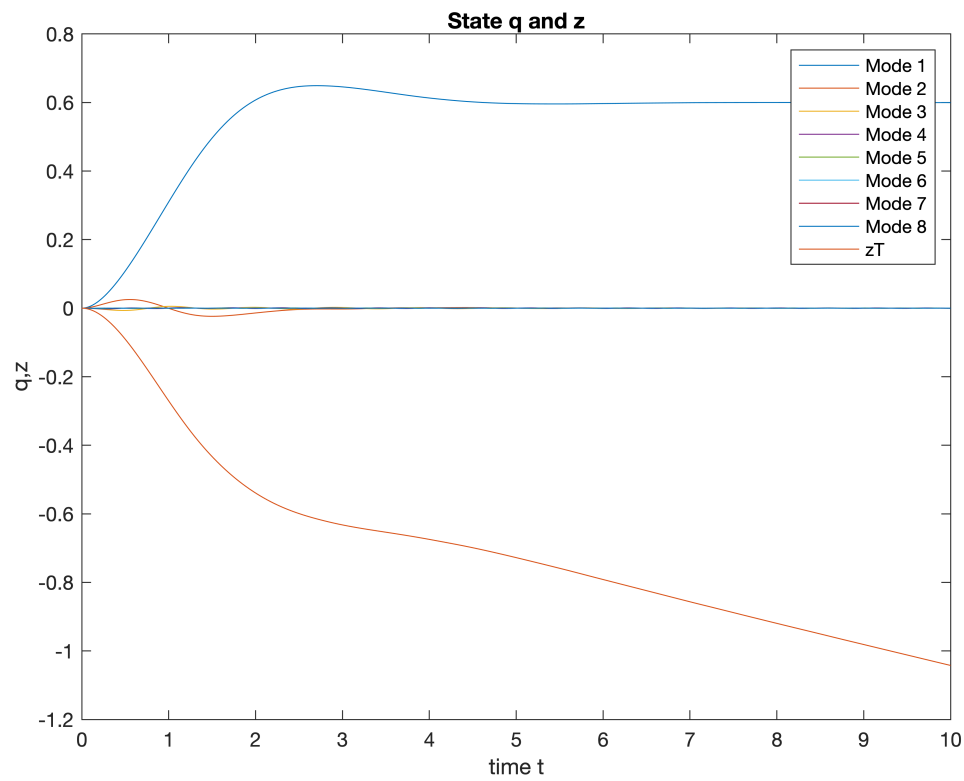
    e = [e norm([x(end)-x_a-target_x y(end)-y_a-target_y])];
end

```

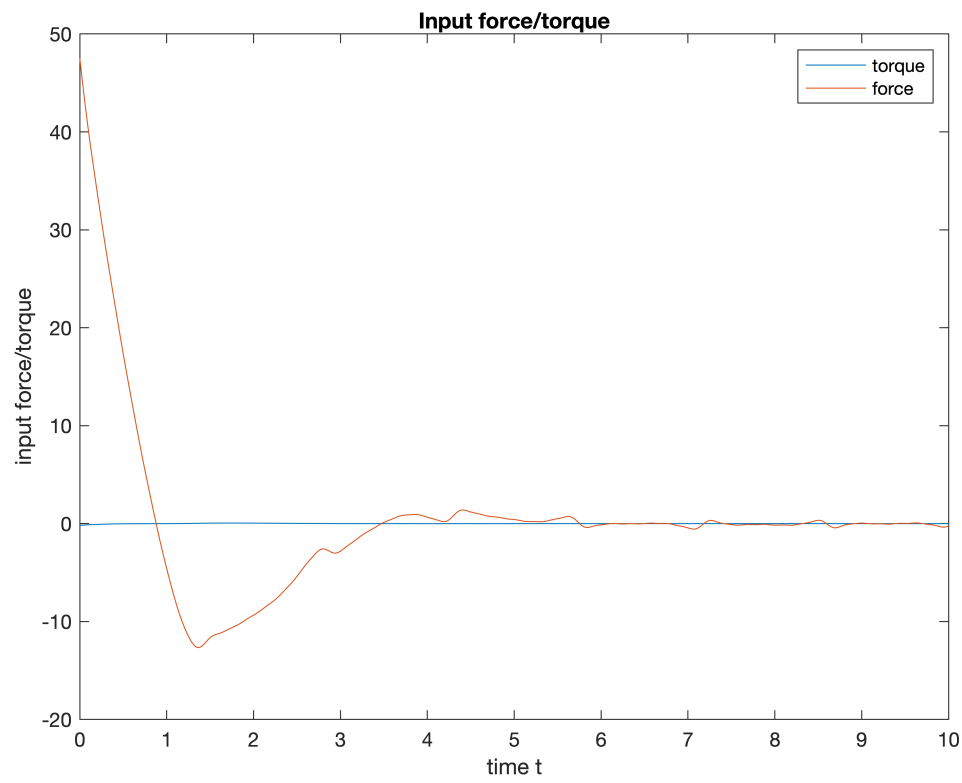
```

figure(2);
plot(ts,qout(:,1:qD)); hold on;
plot(ts,qout(:,2*qD+1));
xlabel('time t');
ylabel('q,z')
legend('Mode 1','Mode 2','Mode 3','Mode 4','Mode 5','Mode 6','Mode 7','Mode 8','zT')
title('State q and z');

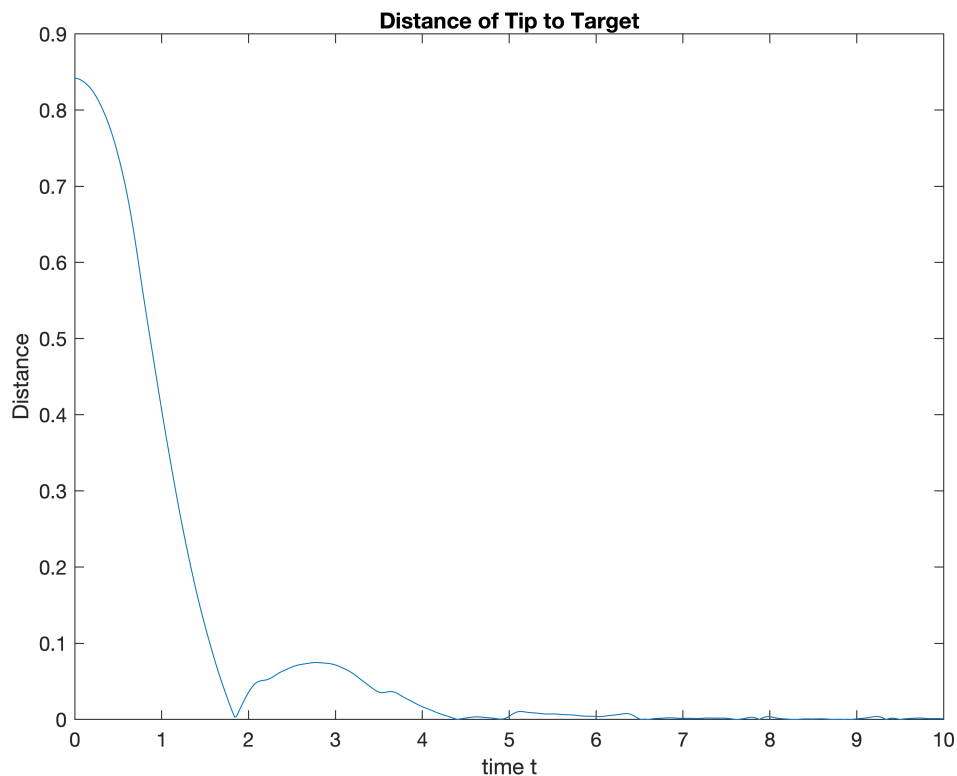
```



```
figure(4)
plot(ts,u);
xlabel('time t');
ylabel('input force/torque')
legend('torque','force');
title('Input force/torque');
```



```
figure(5)
plot(ts,e);
xlabel('time t');
ylabel('Distance')
title('Distance of Tip to Target');
```



```
% Animation
% skip=1;
% ts_sample = ts(1:skip:end);
% x_beam = x_beam(1:skip:end,:); y_beam = y_beam(1:skip:end,:);
% x_nominal = x_nominal(1:skip:end,:); y_nominal = y_nominal(1:skip:end,:);
% aLX = max(x_beam(:,end))-min(x_beam(:,1));
% aLY = max(y_beam(:,end))-min(y_beam(:,1));
% aL = max([aLX aLY]);
% margin=0.2;
% axisL = [min(x_beam(:,1))-margin min(x_beam(:,1))+aL+margin...
%         min(y_beam(:,1))-margin min(y_beam(:,1))+aL+margin];
% figure(3);
% pause(1);
% for i=1:length(ts_sample)-1
%     x=x_beam(i,:); y=y_beam(i,:);
%     xb=x_nominal(i,:);yb=y_nominal(i,:);
%     plot(xb,yb,'g--'); hold on;
%     plot(x,y,'b-');
%     s=scatter(xb(1),yb(1),'r. ');s.SizeData = 300;hold off;
%     axis(axisL);
%     pause(ts_sample(i+1)-ts_sample(i));
% end
```

```
function dqdt = odefun(t,q,A,B,K)
    dqdt = (A-B*K)*q;
end
```

```

function [x,y,xb,yb]=wphi_xy(w,phi,l)
    x=[];
    y=[];
    xb=[];
    yb=[];
    for i=1:length(l)
        x = [x l(i)*cos(phi)-w(i)*sin(phi)];
        y = [y l(i)*sin(phi)+w(i)*cos(phi)];
        xb = [xb l(i)*cos(phi)];
        yb = [yb l(i)*sin(phi)];
    end
end
function [w,phi] = q_wphi(q,ks,l,Ann,Bnn,Cnn,Dnn)
    % evaluate phi
    v_prime = 0;
    for i=1:length(q)
        if i==1
            v_prime = v_prime+q(i)*Cnn(i);
            continue
        end
        v_prime = v_prime+q(i)*sci_p_fcn(Ann(i),Bnn(i),Cnn(i),Dnn(i),ks(i),0);
    end
    phi = v_prime;

    % evaluate w
    v = zeros(size(l));
    for i=1:length(q)
        if i==1
            v = v+q(i)*Cnn(i)*l;
            continue
        end
        v = v+q(i)*sci_fcn(Ann(i),Bnn(i),Cnn(i),Dnn(i),ks(i),l);
    end
    w = v-l*phi;
end
function y=sci_int_fcn(an,bn,cn,dn,k,L)
    y=(cn*sin(L*k) + dn*sinh(L*k) + 2*an*sin((L*k)/2)^2 + 2*bn*sinh((L*k)/2)^2)/k;
end
function y = sci_fcn(an,bn,cn,dn,k,l)
    y = an*sin(k*l) + bn*sinh(k*l) + cn*cos(k*l) + dn*cosh(k*l);
end
function y = sci_p_fcn(an,bn,cn,dn,k,l)
    y = an*k*cos(k*l) + bn*k*cosh(k*l) - cn*k*sin(k*l) + dn*k*sinh(k*l);
end

```