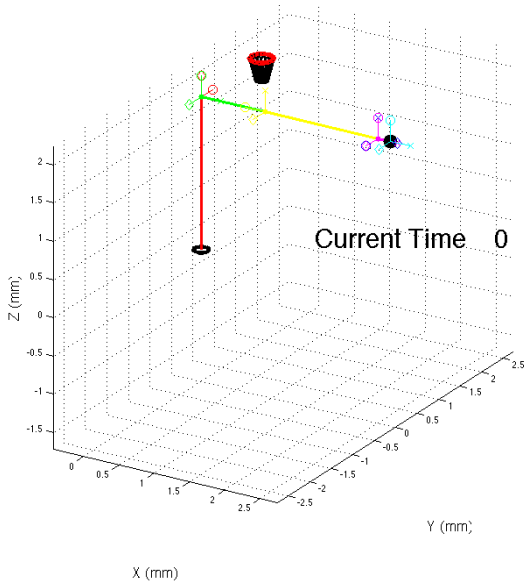
Table of Contents


```
function z401ManipulatorDesignc2()

clear all
close all
clc

robot = basketInit();

% robot = robotInit array

% Cross-sectional areas of each arm member
d1 = 1.5*robot.parameters.d_1;
d2 = 1.5*robot.parameters.d_2;
d3 = 1.5*robot.parameters.d_3;
d4 = 1.5*robot.parameters.d_4;

A1 = pi/4*d1^2;
A2 = pi/4*d2^2;
A3 = pi/4*d3^2;
A4 = pi/4*d4^2;

% Other parameters
m1 = robot.parameters.m_1;
m2 = robot.parameters.m_2;
m3 = robot.parameters.m_3;
m4 = robot.parameters.m_4;

%Constants for deflection,SF
g = robot.const.g;
rho = robot.const.rho;
E = robot.const.E;
Sy = robot.const.Sy;


l1 = robot.parameters.l_1;
l2 = robot.parameters.l_2;
l3 = robot.parameters.l_3;
l4 = robot.parameters.l_4;

m1 = pi/4*d1^2*l1*rho;
m2 = pi/4*d2^2*l2*rho;
m3 = pi/4*d3^2*l3*rho;
m4 = pi/4*d4^2*l4*rho;
```

```
m5 = 2;



% Test joint torques at the zero position
joint_angles = [0;-pi/2;pi/2;0;0];
joint_vel = [0;0;0;0;0];

[T, ~] = basketFK(joint_angles, robot);
x=T*[0;0;0;1]+[0;0;l1;0];

%Draw robot in the given load case
drawBasket(joint_angles,x,robot)

%Compare forces generated by code to FBD
[forces, moments]=z401ForceFinder(joint_angles,[0;0;0;0;0],[0;0;0;0;0])

[~, ~, G] = basketDynamics(joint_angles, joint_vel, robot);
tau = G;
```

```
ans =

  Columns 1 through 7

  245.0016  255.0016  265.0016  275.0016  285.0016  295.0016  296.0016

  Columns 8 through 9

  532.0013  533.0013

Velocity propagation for link 1
Velocity propagation for link 2
Velocity propagation for link 3
Velocity propagation for link 4
Velocity propagation for link 5
Velocity propagation for link 6
Calculating torques for joint 6
Calculating torques for joint 5
Calculating torques for joint 4
Calculating torques for joint 3
Calculating torques for joint 2
Calculating torques for joint 1

forces =

         0         0   33.3262    3.3701         0         0
   -0.0000   50.1765    0.0000    0.0000    3.3701         0
   50.1765    0.0000    0.0000   -0.0000    0.0000         0


moments =

   -0.0000   -0.0000   -0.0000   -0.0000         0         0
```

```
   -67.2364   -0.0000    0.0000    0.3033   -0.0000         0
     0.0000   67.2364   29.6603    0.0000    0.3033         0
```

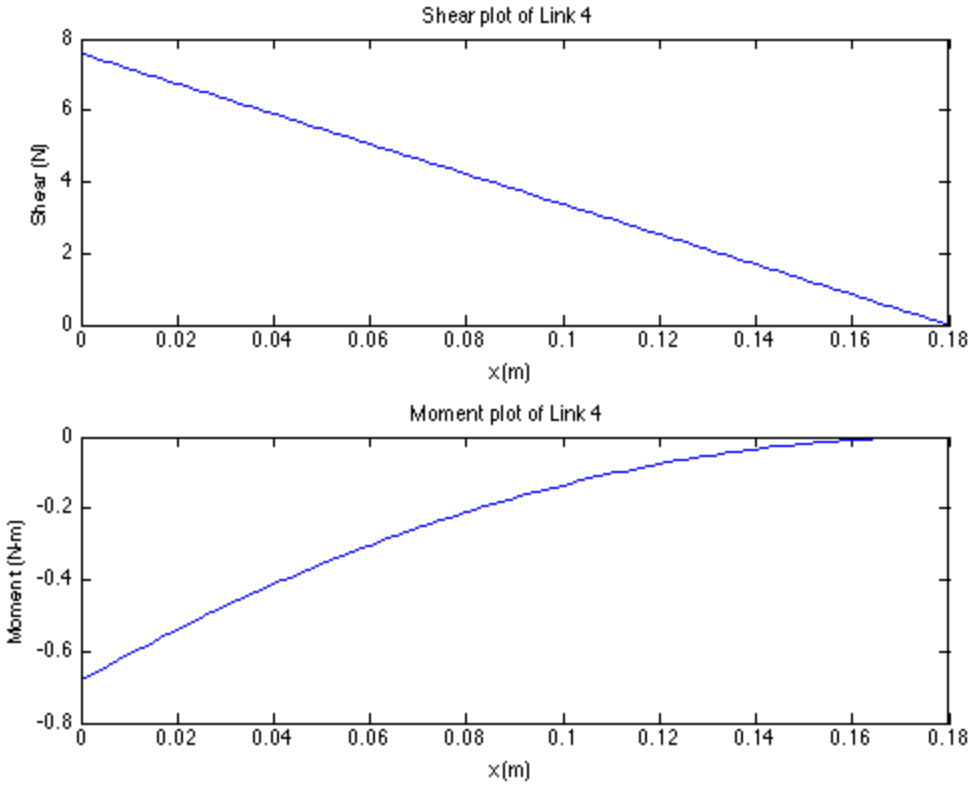


Member 4

```
%George's inferior code

F5 = m4*g + m5*g;
% x4 = [0, l4/2, l4];
% shear4 = [F5, F5-m4*g, 0];
% moment4 = [tau(5), tau(5)+F5*l4/2, 0];
%
% stairs(x4,shear4);
% title('Member 4 Shear Diagram');
% xlabel('x (m)');
% ylabel('Shear (N)');
%
% figure
% plot(x4,moment4);
% title('Member 4 Moment Diagram');
% xlabel('x (m)');
% ylabel('Moment (N-m)');

%Treating weight as a distributed load

x=linspace(0,l4,100);
w=m4*g/l4;
V4=w*(l4-x);
M4=-w/2*(l4-x).^2;

%Deflection of the first member
```

```
%Ymax1=(-w*l4^4)/(8*E*I4)

%bending Stress
Mmax=max(m4);
c=d4/2;
I4=(pi*d4^4)/64;
Sy
Sig_max=Mmax*c/I4
n=Sy/Sig_max

T5=w*l4^2/2;
F1=w*l4;

figure
subplot(2,1,1)
plot(x,V4)
xlabel('x (m)');
ylabel('Shear (N)');
title('Shear plot of Link 4')

subplot(2,1,2)
plot(x,M4)
xlabel('x (m)');
ylabel('Moment (N-m)');
title('Moment plot of Link 4')
```

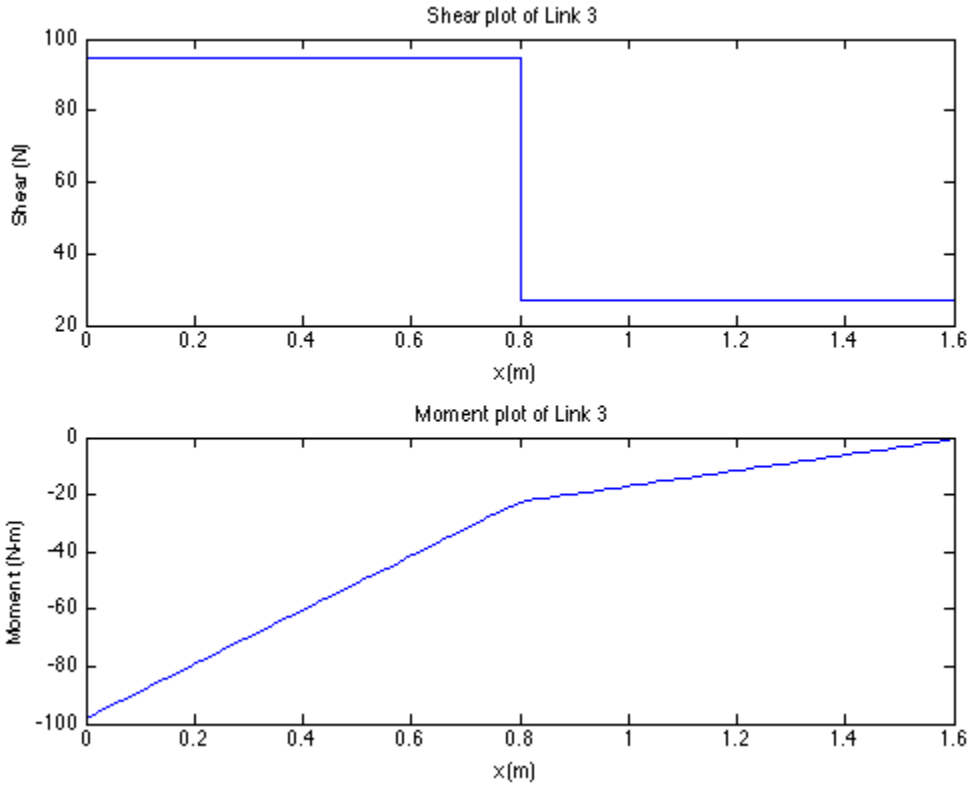
```
Sy =

   2.7600e+11


Sig_max =

   8.6400e+04


n =

   3.1944e+06
```



Member 3

```
F3=m3*g+F5;

T3=m3*g*l3/2+T5+F5*l3;

x1=linspace(0,l3/2,50);
V31=F3+0*x1;
M31=F3*x1-T3;

x2=linspace(l3/2,l3,50);
V32=F5+0*x2;
M32=-F5*(l3-x2)-T5;

figure

subplot(2,1,1)
plot([x1 x2],[V31 V32],'-')
xlabel('x (m)');
ylabel('Shear (N)');
title('Shear plot of Link 3')

subplot(2,1,2)
plot([x1 x2],[M31 M32])
xlabel('x (m)');
```

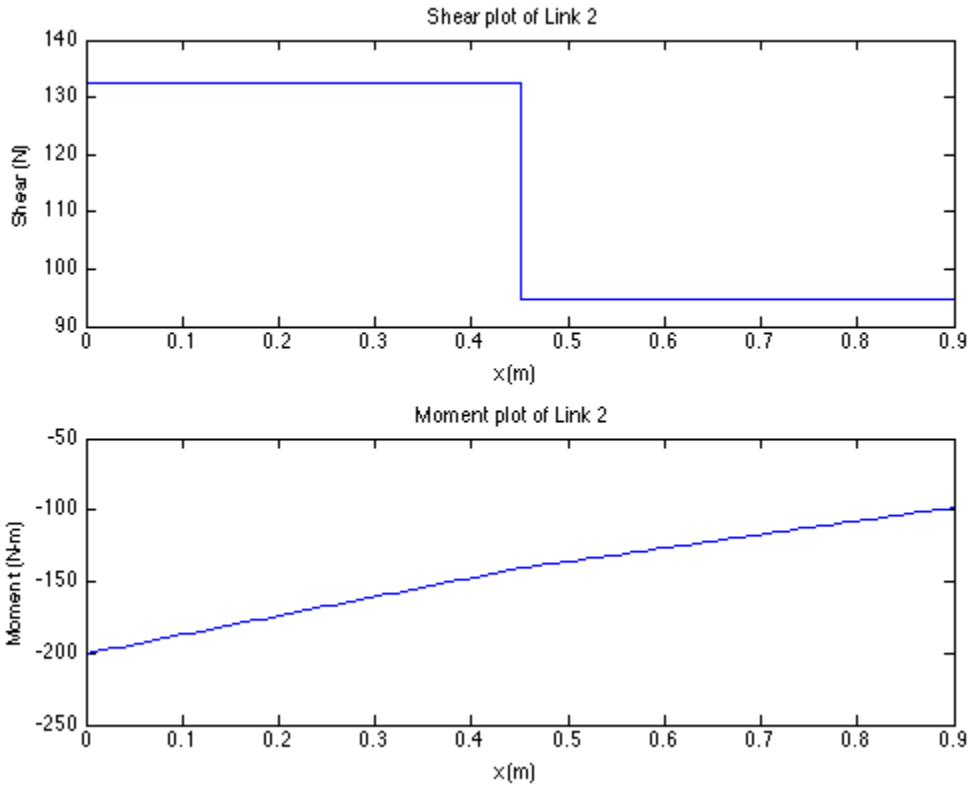
```
ylabel('Moment (N-m)');
title('Moment plot of Link 3')
```



Member 2

```
F2=F3+m2*g;
T2=m2*g*l2/2+T3+F3*l2;

x1=linspace(0,l2/2,50);
V21=F2+0*x1;
M21=F2*x1-T2;

x2=linspace(l2/2,l2,50);
V22=F3+0*x2;
M22=-F3*(l2-x2)-T3;

figure

subplot(2,1,1)
plot([x1 x2],[V21 V22],'-')
xlabel('x (m)');
ylabel('Shear (N)');
title('Shear plot of Link 2')

subplot(2,1,2)
plot([x1 x2],[M21 M22])
```

```
xlabel('x (m)');
ylabel('Moment (N-m)');
title('Moment plot of Link 2')
```



Member 1

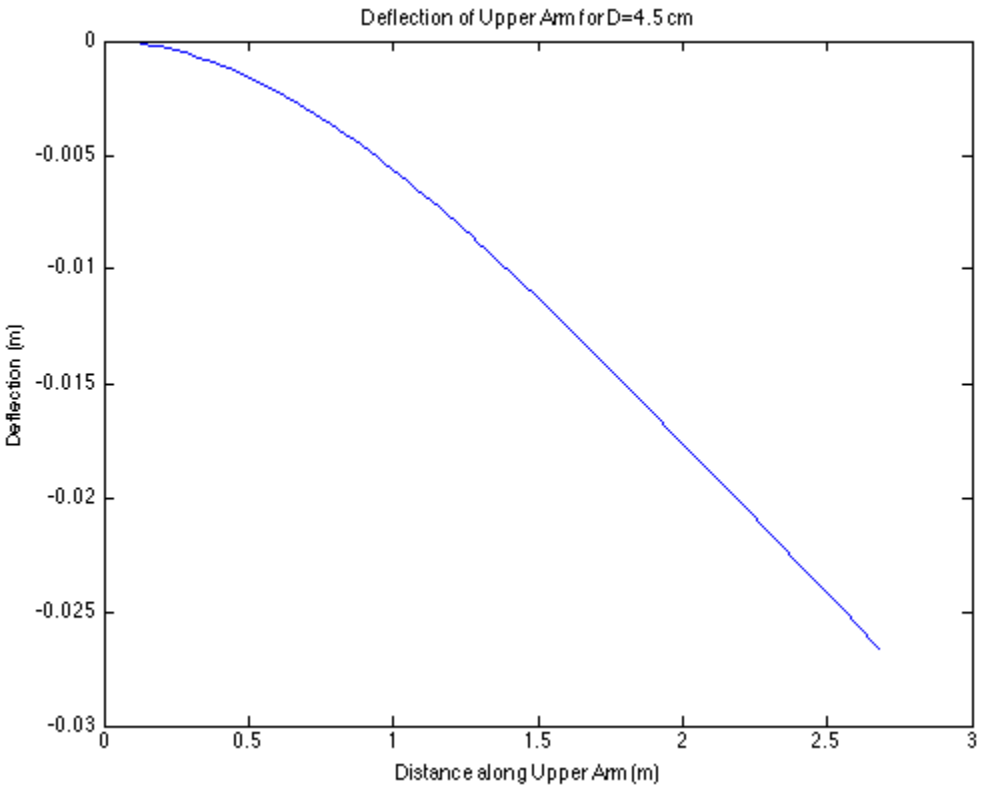
```
%Reaction forces at the ground
Rf=F2+m1*g;
Rm=T2;
```

Deflection of upper Arm

```
%Deflection Constants
I2=(pi*d2^4)/64;
I3=(pi*d3^4)/64;
I4=(pi*d4^4)/64;
L1=l2/2;
L2=l2+l3/2;
L3=L2+l4/2;
L4=l2+l3+l4;

clear x
x=linspace(0,l2+l3+l4,100);
y1=zeros(length(x),1);
y2=zeros(length(x),1);
```

```
y3=zeros(length(x),1);
y4=zeros(length(x),1);
for i=1:length(x)
    if x(i)<l2
        I=I2;
    elseif x(i)<l3
        I=I3;
    else
        I=I4;
    end

    %deflection due to load 1
    if x(i)<L1
        y1(i)=((m2*g*x(i)^2)/(6*E*I))*(x(i)-3*L1);
    else
        y1(i)=((m2*g*L1^2)/(6*E*I))*(L1-3*x(i));
    end

    %deflection due to load 2
    if x(i)<L2
        y2(i)=((m3*g*x(i)^2)/(6*E*I))*(x(i)-3*L2);
    else
        y2(i)=((m3*g*L2^2)/(6*E*I))*(L2-3*x(i));
    end

    %deflection due to load 3
    if x(i)<L3
        y3(i)=((m4*g*x(i)^2)/(6*E*I))*(x(i)-3*L3);
    else
        y3(i)=((m4*g*L3^2)/(6*E*I))*(L3-3*x(i));
    end

        %deflection due to load 3
    if x(i)<L4
        y4(i)=((m5*g*x(i)^2)/(6*E*I))*(x(i)-3*L4);
    else
        y4(i)=((m5*g*L4^2)/(6*E*I))*(L4-3*x(i));
    end
end


%Superposition of loads
y=y1+y2+y3+y4;
maxdef=min(y)
%plotting Deflection of links 3 and 4
figure
plot(x,y)
title('Deflection of Upper Arm for D=4.5 cm')
xlabel('Distance along Upper Arm (m)')
ylabel('Deflection (m)')


return
```

```
maxdef =

   -0.0266
```



Published with MATLAB® R2013a