In [33]:

```
import math
import pandas as pd
```

In [34]:

```
#constants
c = math.pi/180 #conversion factor
g = 9.81 #gravity

#initial conditions

#joint acceleration of ankle throughout entire movement
x_1 = 0
y_1 = 0

#angular velocity segment 1 during 1st interval
w_1 = 0
w_2 = 0
w_3 = 0
w_lb1 = 0
w_a = 0

#segment dimensions
L_1 = 0.48
L_2 = 0.44
L_3 = 0.55
L_lb1 = 0.12
L_a = 0.69

#masses
M = 180 #total
M_b = 300 #bar

M_1 = M*0.043
M_2 = M*0.11
M_3 = M*0.53
M_3 += M*0.07 #adding head to the torso "simplification"
M_b1 = 0
M_a = M*0.049

#proximal distances for each segment
P_1 = 0.4459
P_2 = 0.4095
P_3 = 0.4486
P_lb1 = 0.5
P_a = 0.5280

#moment of inertia for each segment
I_1 = 1/12*M_1*L_1**2
I_2 = 1/12*M_2*L_2**2
I_3 = 1/12*M_3*L_3**2
I_lb1 = 1/12*M_b1*L_lb1**2
I_a = 1/12*M_a*L_a**2
```

In [35]:

```
#to find angular accelerations and velocities for each segment
class angular_acceleration_velocity:

    #angular acceleration
    def a_i(O__i, O_i, t_i, w_i):
        return(2*(O__i-O_i)/(t_i**2)-w_i*t_i)

    #angular velocity
    def w_i(O__i, O_i, a_i):
        return(2*a_i*(O__i-O_i))
```

In [36]:

```
#to find linear accelerations of joints and segments
class accelerations:

    #general (joint)
    def x__i(x_i, L_i, a_i, w_i, theta): #x__i = x_(i+1)
        return(x_i-L_i*((a_i)*math.sin(theta)+w_i**2*math.cos(theta)))

    def y__i(y_i, L_i, a_i, w_i, theta):
        return(y_i+L_i*((a_i)*math.cos(theta)-w_i**2*math.sin(theta)))

    #general (center of mass)
    def x_c(x_i, P_i, L_i, a_i, w_i, theta):
        return(x_i-(1-P_i)*L_i*((a_i)*math.sin(theta)+w_i**2*math.cos(theta)))

    def y_c(y_i, P_i, L_i, a_i, w_i, theta):
        return(y_i+(1-P_i)*L_i*((a_i)*math.cos(theta)-w_i**2*math.sin(theta)))
```

In [37]:

```
#to find forces experienced by each segment
class forces:

    def Fx_12(Fx__i,M_i,x_c):
        return(Fx__i - M_i*x_c)

    def Fy_12(Fy__i,M_i,y_c):
        return(Fy__i + M_i*(y_c+g))

    def Fx_3(Fx_lb1,M_3,x_c): #torso+head
        return(Fx_lb1 - M_3*x_c)

    def Fy_3(Fy_lb1,M_3,y_c):
        return(Fy_lb1 + M_3*(y_c+g))

    def Fx_lb1(Fx_a):
        return(Fx_a)

    def Fy_lb1(Fy_a):
        return(Fy_a)

    def Fx_a(Fx_b, M_5, x_c):
        return(Fx_b - M_5*x_c)

    def Fy_a(Fy_b, M_5, y_c):
        return(Fy_b + M_5*(y_c+g))

    def Fx_b(M_b, x_b):
        return(-M_b*x_b)

    def Fy_b(M_b, y_b):
        return(M_b*(y_b+g))
```

In [38]:

```
#to find moments experienced by each segment
class moments:
    def m__i(m_i, a_i, Fx__i, Fy__i, Fx_i, Fy_i, theta):
        return(m_i-I_a*a_i-Fx__i*P_a*L_a*math.sin(theta)-Fy__i*P_a*L_a*math.cos(theta)

    def m_lb1(m_a, I_lb1, a_lb1, Fy_4, P_lb1, L_lb1, theta):
        return(-I_lb1*a_lb1+m_a-Fy_4*(1-P_lb1)*L_lb1*math.cos(theta))
```

In [39]:

```
#to find shear and compression forces for the spine

class shear_compression:

    def shear(Fx_i, Fy_i, M_i, x_i, y_i, theta):
        shear = abs(Fx_i*math.sin(theta)+Fy_i*math.cos(theta)-M_i*g*math.cos(theta)-(M
        return shear

    def compression(Fx_i, Fy_i, M_i, x_i, y_i, theta):
        compression = abs(Fx_i*math.cos(theta)-Fy_i*math.sin(theta)+M_i*g*math.sin(the
        return compression
```

In [40]:

```
#interval 12
#inputs
#angle
O_1 = 80*c
O_2 = 170*c
O_3 = 37*c
O_4 = 180*c

#angle'
O__1=88*c
O__2=141*c
O__3=40*c
O__4=180*c

#time
t_i = 0.5

#accelerations (angular and linear)
#segment 1 & joint 2
a_1 = angular_acceleration_velocity.a_i(O__1, O_1, t_i, w_1)
w_1 = angular_acceleration_velocity.w_i(O__1, O_1, a_1)
x_1c = accelerations.x_c(x_1, P_1, L_1, a_1, w_1, O__1)
y_1c = accelerations.y_c(y_1, P_1, L_1, a_1, w_1, O__1)
x_2 = accelerations.x__i(x_1, L_1, a_1, w_1, O__1)
y_2 = accelerations.y__i(y_1, L_1, a_1, w_1, O__1)

#segment 2 & joint 3
a_2 = angular_acceleration_velocity.a_i(O__2, O_2, t_i, w_2)
w_2 = angular_acceleration_velocity.w_i(O__2, O_2, a_2)
x_2c = accelerations.x_c(x_2, P_2, L_2, a_2, w_2, O__2)
y_2c = accelerations.y_c(y_2, P_2, L_2, a_2, w_2, O__2)
x_3 = accelerations.x__i(x_2, L_2, a_2, w_2, O__2)
y_3 = accelerations.y__i(y_2, L_2, a_2, w_2, O__2)

#segment 3 & joint 4
a_3 = angular_acceleration_velocity.a_i(O__3, O_3, t_i, w_3)
w_3 = angular_acceleration_velocity.w_i(O__3, O_3, a_3)
x_3c = accelerations.x_c(x_3, P_3, L_3, a_3, w_3, O__3)
y_3c = accelerations.y_c(y_3, P_3, L_3, a_3, w_3, O__3)
x_lb1 = accelerations.x__i(x_3, L_3, a_3, w_3, O__3)
y_lb1 = accelerations.y__i(y_3, L_3, a_3, w_3, O__3)
```

```
#segment 4 & joint 5
a_lb1 = 0
w_lb1 = 0

#segment 5 & joint 6
a_4 = angular_acceleration_velocity.a_i(O__4, O_4, t_i, w_a)
w_a = angular_acceleration_velocity.w_i(O__4, O_4, a_4)
x_lb1c = accelerations.x_c(x_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
y_lb1c = accelerations.y_c(y_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
x_a = accelerations.x__i(x_lb1, L_lb1, a_4, w_a, O__4)
y_a = accelerations.y__i(y_lb1, L_lb1, a_4, w_a, O__4)

#the bar
x_b = x_a
y_b = y_a


#forces
Fy_b = forces.Fy_b(M_b, y_b)
Fx_b = forces.Fx_b(M_b, x_b)

Fy_a = forces.Fy_a(Fy_b,M_a,y_a)
Fx_a = forces.Fx_a(Fx_b,M_a,x_a)

Fx_lb1 = Fx_a
Fy_lb1 = Fy_a

Fy_3 = forces.Fy_3(Fy_lb1,M_3,y_3c)
Fx_3 = forces.Fx_3(Fx_lb1,M_3,x_3c)

Fy_2 = forces.Fy_12(Fy_3,M_2,y_2c)
Fx_2 = forces.Fx_12(Fx_3,M_2,x_2c)

Fy_1 = forces.Fy_12(Fy_2,M_1,y_1c)
Fx_1 = forces.Fx_12(Fx_2,M_1,x_1c)


#moments
m_a = moments.m__i(0, a_4, Fx_b, Fy_b, Fx_a, Fy_a, O__4)

m_lb1 = moments.m_lb1(m_a, I_lb1, a_lb1, Fy_a, P_lb1, L_lb1, 180)

m_3 = moments.m__i(m_lb1, a_3, Fx_3, Fy_3, Fx_lb1, Fy_lb1, O__3)

m_2 = moments.m__i(m_3, a_2, Fx_3, Fy_3, Fx_2, Fy_2, O__2)

m_1 = moments.m__i(m_2, a_1, Fx_2, Fy_2, Fx_1, Fy_1, O__1)

#output
d = {'x_acc':[x_1, x_2, x_3, x_lb1, x_a, x_b], 'y_acc':[y_1, y_2, y_3, y_lb1, y_a, y_b

i = ['shank', 'thigh', 'torso', 'shoulder', 'arm', 'bar']

int_12 = pd.DataFrame(d, i)

#compression & shear output for interval 12
spine_c12 = shear_compression.compression(Fx_3, Fy_3, M_3, x_3, y_3, O__3)

spine_s12 = shear_compression.shear(Fx_3, Fy_3, M_3, x_3, y_3, O__3)
```

In [41]:

```
int_12
```

Out[41]:

| | x_acc | y_acc | x_force | y_force | moment |
|---|---|---|---|---|---|
| **shank** | 0.000000 | 0.000000 | -2650.469259 | 3011.358261 | 5615.325797 |
| **thigh** | -0.537468 | -0.027963 | -2652.774319 | 2935.548788 | 3858.719276 |
| **torso** | 6.328837 | -3.295656 | -2583.136036 | 2780.069995 | 1191.240817 |
| **shoulder** | 6.179939 | -3.119852 | -1908.488701 | 2066.051494 | 1478.265022 |
| **arm** | 6.179939 | -3.119852 | -1908.488701 | 2066.051494 | 1404.078063 |
| **bar** | 6.179939 | -3.119852 | -1853.981640 | 2007.044389 | 0.000000 |

In [42]:

```
#interval 23
#inputs
#angle
O_1 = 88*c
O_2 = 141*c
O_3 = 40*c
O_4 = 180*c

#angle'
O__1=88*c
O__2=134*c
O__3=44*c
O__4=180*c

#time
t_i = 0.4


#accelerations (angular and linear)
#segment 1 & joint 2
a_1 = angular_acceleration_velocity.a_i(O__1, O_1, t_i, w_1)
w_1 = angular_acceleration_velocity.w_i(O__1, O_1, a_1)
x_1c = accelerations.x_c(x_1, P_1, L_1, a_1, w_1, O__1)
y_1c = accelerations.y_c(y_1, P_1, L_1, a_1, w_1, O__1)
x_2 = accelerations.x__i(x_1, L_1, a_1, w_1, O__1)
y_2 = accelerations.y__i(y_1, L_1, a_1, w_1, O__1)

#segment 2 & joint 3
a_2 = angular_acceleration_velocity.a_i(O__2, O_2, t_i, w_2)
w_2 = angular_acceleration_velocity.w_i(O__2, O_2, a_2)
x_2c = accelerations.x_c(x_2, P_2, L_2, a_2, w_2, O__2)
y_2c = accelerations.y_c(y_2, P_2, L_2, a_2, w_2, O__2)
x_3 = accelerations.x__i(x_2, L_2, a_2, w_2, O__2)
y_3 = accelerations.y__i(y_2, L_2, a_2, w_2, O__2)

#segment 3 & joint 4
a_3 = angular_acceleration_velocity.a_i(O__3, O_3, t_i, w_3)
w_3 = angular_acceleration_velocity.w_i(O__3, O_3, a_3)
x_3c = accelerations.x_c(x_3, P_3, L_3, a_3, w_3, O__3)
y_3c = accelerations.y_c(y_3, P_3, L_3, a_3, w_3, O__3)
x_lb1 = accelerations.x__i(x_3, L_3, a_3, w_3, O__3)
y_lb1 = accelerations.y__i(y_3, L_3, a_3, w_3, O__3)
```

```
#segment 4 & joint 5
a_lb1 = 0
w_lb1 = 0

#segment 5 & joint 6
a_4 = angular_acceleration_velocity.a_i(O__4, O_4, t_i, w_a)
w_a = angular_acceleration_velocity.w_i(O__4, O_4, a_4)
x_lb1c = accelerations.x_c(x_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
y_lb1c = accelerations.y_c(y_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
x_a = accelerations.x__i(x_lb1, L_lb1, a_4, w_a, O__4)
y_a = accelerations.y__i(y_lb1, L_lb1, a_4, w_a, O__4)

#the bar
x_b = x_a
y_b = y_a


#forces
Fy_b = forces.Fy_b(M_b, y_b)
Fx_b = forces.Fx_b(M_b, x_b)

Fy_a = forces.Fy_a(Fy_b,M_a,y_a)
Fx_a = forces.Fx_a(Fx_b,M_a,x_a)

Fx_lb1 = Fx_a
Fy_lb1 = Fy_a

Fy_3 = forces.Fy_3(Fy_lb1,M_3,y_3c)
Fx_3 = forces.Fx_3(Fx_lb1,M_3,x_3c)

Fy_2 = forces.Fy_12(Fy_3,M_2,y_2c)
Fx_2 = forces.Fx_12(Fx_3,M_2,x_2c)

Fy_1 = forces.Fy_12(Fy_2,M_1,y_1c)
Fx_1 = forces.Fx_12(Fx_2,M_1,x_1c)


#moments
m_a = moments.m__i(0, a_4, Fx_b, Fy_b, Fx_a, Fy_a, O__4)

m_lb1 = moments.m_lb1(m_a, I_lb1, a_lb1, Fy_a, P_lb1, L_lb1, 180)

m_3 = moments.m__i(m_lb1, a_3, Fx_3, Fy_3, Fx_lb1, Fy_lb1, O__3)

m_2 = moments.m__i(m_3, a_2, Fx_3, Fy_3, Fx_2, Fy_2, O__2)

m_1 = moments.m__i(m_2, a_1, Fx_2, Fy_2, Fx_1, Fy_1, O__1)

#output
d = {'x_acc':[x_1, x_2, x_3, x_lb1, x_a, x_b], 'y_acc':[y_1, y_2, y_3, y_lb1, y_a, y_b

i = ['shank', 'thigh', 'torso', 'shoulder', 'arm', 'bar']

int_23 = pd.DataFrame(d, i)

#compression & shear output for interval 23
spine_c23 = shear_compression.compression(Fx_3, Fy_3, M_3, x_3, y_3, O__3)

spine_s23 = shear_compression.shear(Fx_3, Fy_3, M_3, x_3, y_3, O__3)
```

In [43]:
```
int_23
```

Out[43]:

| | x_acc | y_acc | x_force | y_force | moment |
|---|---|---|---|---|---|
| **shank** | 0.000000 | 0.000000 | -411.879774 | 4814.426215 | 3188.291191 |
| **thigh** | 0.059854 | -0.002090 | -411.623077 | 4738.505779 | 3019.286302 |
| **torso** | 1.245159 | 0.776316 | -396.579504 | 4535.208112 | 598.030476 |
| **shoulder** | 0.912810 | 1.109186 | -281.894098 | 3372.063133 | 2412.719622 |
| **arm** | 0.912810 | 1.109186 | -281.894098 | 3372.063133 | 2291.636914 |
| **bar** | 0.912810 | 1.109186 | -273.843111 | 3275.755910 | 0.000000 |

In [44]:
```
#interval 34
#inputs
#angle
O_1 = 88*c
O_2 = 134*c
O_3 = 44*c
O_4 = 180*c

#angle'
O__1=88*c
O__2=124*c
O__3=55*c
O__4=180*c

#time
t_i = 1.4


#accelerations (angular and linear)
#segment 1 & joint 2
a_1 = angular_acceleration_velocity.a_i(O__1, O_1, t_i, w_1)
w_1 = angular_acceleration_velocity.w_i(O__1, O_1, a_1)
x_1c = accelerations.x_c(x_1, P_1, L_1, a_1, w_1, O__1)
y_1c = accelerations.y_c(y_1, P_1, L_1, a_1, w_1, O__1)
x_2 = accelerations.x__i(x_1, L_1, a_1, w_1, O__1)
y_2 = accelerations.y__i(y_1, L_1, a_1, w_1, O__1)

#segment 2 & joint 3
a_2 = angular_acceleration_velocity.a_i(O__2, O_2, t_i, w_2)
w_2 = angular_acceleration_velocity.w_i(O__2, O_2, a_2)
x_2c = accelerations.x_c(x_2, P_2, L_2, a_2, w_2, O__2)
y_2c = accelerations.y_c(y_2, P_2, L_2, a_2, w_2, O__2)
x_3 = accelerations.x__i(x_2, L_2, a_2, w_2, O__2)
y_3 = accelerations.y__i(y_2, L_2, a_2, w_2, O__2)

#segment 3 & joint 4
a_3 = angular_acceleration_velocity.a_i(O__3, O_3, t_i, w_3)
w_3 = angular_acceleration_velocity.w_i(O__3, O_3, a_3)
x_3c = accelerations.x_c(x_3, P_3, L_3, a_3, w_3, O__3)
y_3c = accelerations.y_c(y_3, P_3, L_3, a_3, w_3, O__3)
x_lb1 = accelerations.x__i(x_3, L_3, a_3, w_3, O__3)
y_lb1 = accelerations.y__i(y_3, L_3, a_3, w_3, O__3)

#segment 4 & joint 5
```

```
a_lb1 = 0
w_lb1 = 0

#segment 5 & joint 6
a_4 = angular_acceleration_velocity.a_i(O__4, O_4, t_i, w_a)
w_a = angular_acceleration_velocity.w_i(O__4, O_4, a_4)
x_lb1c = accelerations.x_c(x_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
y_lb1c = accelerations.y_c(y_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
x_a = accelerations.x__i(x_lb1, L_lb1, a_4, w_a, O__4)
y_a = accelerations.y__i(y_lb1, L_lb1, a_4, w_a, O__4)

#the bar
x_b = x_a
y_b = y_a


#forces
Fy_b = forces.Fy_b(M_b, y_b)
Fx_b = forces.Fx_b(M_b, x_b)

Fy_a = forces.Fy_a(Fy_b,M_a,y_a)
Fx_a = forces.Fx_a(Fx_b,M_a,x_a)

Fx_lb1 = Fx_a
Fy_lb1 = Fy_a

Fy_3 = forces.Fy_3(Fy_lb1,M_3,y_3c)
Fx_3 = forces.Fx_3(Fx_lb1,M_3,x_3c)

Fy_2 = forces.Fy_12(Fy_3,M_2,y_2c)
Fx_2 = forces.Fx_12(Fx_3,M_2,x_2c)

Fy_1 = forces.Fy_12(Fy_2,M_1,y_1c)
Fx_1 = forces.Fx_12(Fx_2,M_1,x_1c)


#moments
m_a = moments.m__i(0, a_4, Fx_b, Fy_b, Fx_a, Fy_a, O__4)

m_lb1 = moments.m_lb1(m_a, I_lb1, a_lb1, Fy_a, P_lb1, L_lb1, 180)

m_3 = moments.m__i(m_lb1, a_3, Fx_3, Fy_3, Fx_lb1, Fy_lb1, O__3)

m_2 = moments.m__i(m_3, a_2, Fx_3, Fy_3, Fx_2, Fy_2, O__2)

m_1 = moments.m__i(m_2, a_1, Fx_2, Fy_2, Fx_1, Fy_1, O__1)

#output
d = {'x_acc':[x_1, x_2, x_3, x_lb1, x_a, x_b], 'y_acc':[y_1, y_2, y_3, y_lb1, y_a, y_b

i = ['shank', 'thigh', 'torso', 'shoulder', 'arm', 'bar']

int_34 = pd.DataFrame(d, i)

#compression & shear output for interval 34
spine_c34 = shear_compression.compression(Fx_3, Fy_3, M_3, x_3, y_3, O__3)

spine_s34 = shear_compression.shear(Fx_3, Fy_3, M_3, x_3, y_3, O__3)
```

In [45]:

```
int_34
```

Out[45]:

| | x_acc | y_acc | x_force | y_force | moment |
|---|---|---|---|---|---|
| **shank** | 0.000000 | 0.000000 | -212.823638 | 4465.180078 | 2682.791906 |
| **thigh** | 0.000000 | 0.000000 | -212.823638 | 4389.250678 | 2642.592092 |
| **torso** | 0.507825 | 0.239638 | -206.886201 | 4192.210851 | 868.781105 |
| **shoulder** | 0.494834 | 0.248653 | -152.814722 | 3106.313099 | 2222.574807 |
| **arm** | 0.494834 | 0.248653 | -152.814722 | 3106.313099 | 2111.034546 |
| **bar** | 0.494834 | 0.248653 | -148.450284 | 3017.595783 | 0.000000 |

In [46]:

```
#interval 45
#inputs
#angle
O_1 = 88*c
O_2 = 124*c
O_3 = 55*c
O_4 = 180*c

#angle'
O__1=88*c
O__2=101*c
O__3=77*c
O__4=180*c

#time
t_i = 0.3


#accelerations (angular and linear)
#segment 1 & joint 2
a_1 = angular_acceleration_velocity.a_i(O__1, O_1, t_i, w_1)
w_1 = angular_acceleration_velocity.w_i(O__1, O_1, a_1)
x_1c = accelerations.x_c(x_1, P_1, L_1, a_1, w_1, O__1)
y_1c = accelerations.y_c(y_1, P_1, L_1, a_1, w_1, O__1)
x_2 = accelerations.x__i(x_1, L_1, a_1, w_1, O__1)
y_2 = accelerations.y__i(y_1, L_1, a_1, w_1, O__1)

#segment 2 & joint 3
a_2 = angular_acceleration_velocity.a_i(O__2, O_2, t_i, w_2)
w_2 = angular_acceleration_velocity.w_i(O__2, O_2, a_2)
x_2c = accelerations.x_c(x_2, P_2, L_2, a_2, w_2, O__2)
y_2c = accelerations.y_c(y_2, P_2, L_2, a_2, w_2, O__2)
x_3 = accelerations.x__i(x_2, L_2, a_2, w_2, O__2)
y_3 = accelerations.y__i(y_2, L_2, a_2, w_2, O__2)

#segment 3 & joint 4
a_3 = angular_acceleration_velocity.a_i(O__3, O_3, t_i, w_3)
w_3 = angular_acceleration_velocity.w_i(O__3, O_3, a_3)
x_3c = accelerations.x_c(x_3, P_3, L_3, a_3, w_3, O__3)
y_3c = accelerations.y_c(y_3, P_3, L_3, a_3, w_3, O__3)
x_lb1 = accelerations.x__i(x_3, L_3, a_3, w_3, O__3)
y_lb1 = accelerations.y__i(y_3, L_3, a_3, w_3, O__3)

#segment 4 & joint 5
```

```
a_lb1 = 0
w_lb1 = 0

#segment 5 & joint 6
a_4 = angular_acceleration_velocity.a_i(O__4, O_4, t_i, w_a)
w_a = angular_acceleration_velocity.w_i(O__4, O_4, a_4)
x_lb1c = accelerations.x_c(x_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
y_lb1c = accelerations.y_c(y_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
x_a = accelerations.x__i(x_lb1, L_lb1, a_4, w_a, O__4)
y_a = accelerations.y__i(y_lb1, L_lb1, a_4, w_a, O__4)

#the bar
x_b = x_a
y_b = y_a


#forces
Fy_b = forces.Fy_b(M_b, y_b)
Fx_b = forces.Fx_b(M_b, x_b)

Fy_a = forces.Fy_a(Fy_b,M_a,y_a)
Fx_a = forces.Fx_a(Fx_b,M_a,x_a)

Fx_lb1 = Fx_a
Fy_lb1 = Fy_a

Fy_3 = forces.Fy_3(Fy_lb1,M_3,y_3c)
Fx_3 = forces.Fx_3(Fx_lb1,M_3,x_3c)

Fy_2 = forces.Fy_12(Fy_3,M_2,y_2c)
Fx_2 = forces.Fx_12(Fx_3,M_2,x_2c)

Fy_1 = forces.Fy_12(Fy_2,M_1,y_1c)
Fx_1 = forces.Fx_12(Fx_2,M_1,x_1c)


#moments
m_a = moments.m__i(0, a_4, Fx_b, Fy_b, Fx_a, Fy_a, O__4)

m_lb1 = moments.m_lb1(m_a, I_lb1, a_lb1, Fy_a, P_lb1, L_lb1, 180)

m_3 = moments.m__i(m_lb1, a_3, Fx_3, Fy_3, Fx_lb1, Fy_lb1, O__3)

m_2 = moments.m__i(m_3, a_2, Fx_3, Fy_3, Fx_2, Fy_2, O__2)

m_1 = moments.m__i(m_2, a_1, Fx_2, Fy_2, Fx_1, Fy_1, O__1)

#output
d = {'x_acc':[x_1, x_2, x_3, x_lb1, x_a, x_b], 'y_acc':[y_1, y_2, y_3, y_lb1, y_a, y_b

i = ['shank', 'thigh', 'torso', 'shoulder', 'arm', 'bar']

int_45 = pd.DataFrame(d, i)

#compression & shear output for interval 45
spine_c45 = shear_compression.compression(Fx_3, Fy_3, M_3, x_3, y_3, O__3)

spine_s45 = shear_compression.shear(Fx_3, Fy_3, M_3, x_3, y_3, O__3)
```

In [47]:

```
int_45
```

Out[47]:

| | x_acc | y_acc | x_force | y_force | moment |
|---|---|---|---|---|---|
| **shank** | 0.000000 | 0.000000 | 63.352611 | -13172.699174 | -7449.510647 |
| **thigh** | 0.000000 | 0.000000 | 63.352611 | -13248.628574 | -7723.996580 |
| **torso** | 8.344797 | -22.055078 | 160.919140 | -13185.000809 | -5909.494314 |
| **shoulder** | -1.534361 | -43.992209 | 473.841470 | -10556.149892 | -7552.951701 |
| **arm** | -1.534361 | -43.992209 | 473.841470 | -10556.149892 | -7173.905650 |
| **bar** | -1.534361 | -43.992209 | 460.308403 | -10254.662805 | 0.000000 |

In [48]:

```
#interval 56
#inputs
#angle
O_1 = 88*c
O_2 = 101*c
O_3 = 77*c
O_4 = 180*c

#angle'
O__1=90*c
O__2=90*c
O__3=90*c
O__4=180*c

#time
t_i = 0.4


#accelerations (angular and linear)
#segment 1 & joint 2
a_1 = angular_acceleration_velocity.a_i(O__1, O_1, t_i, w_1)
w_1 = angular_acceleration_velocity.w_i(O__1, O_1, a_1)
x_1c = accelerations.x_c(x_1, P_1, L_1, a_1, w_1, O__1)
y_1c = accelerations.y_c(y_1, P_1, L_1, a_1, w_1, O__1)
x_2 = accelerations.x__i(x_1, L_1, a_1, w_1, O__1)
y_2 = accelerations.y__i(y_1, L_1, a_1, w_1, O__1)

#segment 2 & joint 3
a_2 = angular_acceleration_velocity.a_i(O__2, O_2, t_i, w_2)
w_2 = angular_acceleration_velocity.w_i(O__2, O_2, a_2)
x_2c = accelerations.x_c(x_2, P_2, L_2, a_2, w_2, O__2)
y_2c = accelerations.y_c(y_2, P_2, L_2, a_2, w_2, O__2)
x_3 = accelerations.x__i(x_2, L_2, a_2, w_2, O__2)
y_3 = accelerations.y__i(y_2, L_2, a_2, w_2, O__2)

#segment 3 & joint 4
a_3 = angular_acceleration_velocity.a_i(O__3, O_3, t_i, w_3)
w_3 = angular_acceleration_velocity.w_i(O__3, O_3, a_3)
x_3c = accelerations.x_c(x_3, P_3, L_3, a_3, w_3, O__3)
y_3c = accelerations.y_c(y_3, P_3, L_3, a_3, w_3, O__3)
x_lb1 = accelerations.x__i(x_3, L_3, a_3, w_3, O__3)
y_lb1 = accelerations.y__i(y_3, L_3, a_3, w_3, O__3)

#segment 4 & joint 5
```

```
a_lb1 = 0
w_lb1 = 0

#segment 5 & joint 6
a_a = angular_acceleration_velocity.a_i(O__4, O_4, t_i, w_a)
w_a = angular_acceleration_velocity.w_i(O__4, O_4, a_a)
x_lb1c = accelerations.x_c(x_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
y_lb1c = accelerations.y_c(y_lb1, P_lb1, L_lb1, a_4, w_a, O__4)
x_a = accelerations.x__i(x_lb1, L_lb1, a_4, w_a, O__4)
y_a = accelerations.y__i(y_lb1, L_lb1, a_4, w_a, O__4)

#the bar
x_b = x_a
y_b = y_a


#forces
Fy_b = forces.Fy_b(M_b, y_b)
Fx_b = forces.Fx_b(M_b, x_b)

Fy_a = forces.Fy_a(Fy_b,M_a,y_a)
Fx_a = forces.Fx_a(Fx_b,M_a,x_a)

Fx_lb1 = Fx_a
Fy_lb1 = Fy_a

Fy_3 = forces.Fy_3(Fy_lb1,M_3,y_3c)
Fx_3 = forces.Fx_3(Fx_lb1,M_3,x_3c)

Fy_2 = forces.Fy_12(Fy_3,M_2,y_2c)
Fx_2 = forces.Fx_12(Fx_3,M_2,x_2c)

Fy_1 = forces.Fy_12(Fy_2,M_1,y_1c)
Fx_1 = forces.Fx_12(Fx_2,M_1,x_1c)


#moments
m_a = moments.m__i(0, a_4, Fx_b, Fy_b, Fx_a, Fy_a, O__4)

m_lb1 = moments.m_lb1(m_a, I_lb1, a_lb1, Fy_a, P_lb1, L_lb1, 180)

m_3 = moments.m__i(m_lb1, a_3, Fx_3, Fy_3, Fx_lb1, Fy_lb1, O__3)

m_2 = moments.m__i(m_3, a_2, Fx_3, Fy_3, Fx_2, Fy_2, O__2)

m_1 = moments.m__i(m_2, a_1, Fx_2, Fy_2, Fx_1, Fy_1, O__1)

#output
d = {'x_acc':[x_1, x_2, x_3, x_lb1, x_a, x_b], 'y_acc':[y_1, y_2, y_3, y_lb1, y_a, y_b

i = ['shank', 'thigh', 'torso', 'shoulder', 'arm', 'bar']

int_56= pd.DataFrame(d, i)

#compression & shear output for interval 56
spine_c56 = shear_compression.compression(Fx_3, Fy_3, M_3, x_3, y_3, O__3)

spine_s56 = shear_compression.shear(Fx_3, Fy_3, M_3, x_3, y_3, O__3)
```

In [49]:

```
int_56
```

Out[49]:

| | x_acc | y_acc | x_force | y_force | moment |
|---|---|---|---|---|---|
| **shank** | 0.000000 | 0.000000 | -864.481367 | 3574.111455 | 3458.454673 |
| **thigh** | -0.209440 | -0.000445 | -865.379598 | 3498.183966 | 2861.787974 |
| **torso** | 2.125643 | -1.827501 | -842.224950 | 3325.316540 | 2271.254654 |
| **shoulder** | 2.006780 | -1.832791 | -619.733922 | 2463.521688 | 1762.655941 |
| **arm** | 2.006780 | -1.832791 | -619.733922 | 2463.521688 | 1674.196780 |
| **bar** | 2.006780 | -1.832791 | -602.034119 | 2393.162704 | 0.000000 |

In [50]:

```
#spine shear and compression output
d = {'Interval 12':[spine_s12, spine_c12], 'Interval 23':[spine_s23, spine_c23], 'Inte

i = ['Shear', 'Compression']

shear_compression= pd.DataFrame(d, i)
```

In [52]:

```
shear_compression
```

Out[52]:

| | Interval 12 | Interval 23 | Interval 34 | Interval 45 | Interval 56 |
|---|---|---|---|---|---|
| **Shear** | 932.700289 | 2110.746529 | 1592.604847 | 3620.409076 | 842.224950 |
| **Compression** | 2589.416142 | 2589.634617 | 2635.170522 | 16397.066362 | 1963.087308 |