EML6281

Homework 2

Elias Reyes

In [20]:
```
# import libraries
import numpy as np
from numpy import square as sqr
import math
from math import sin
from math import cos
from IPython.display import Image
```

In [6]:
```
def determine_transformation_about_vector(m, theta):
    # normalize the vector which is being rotated about
    m_hat = m / np.linalg.norm(m)

    mx = m_hat[0]
    my = m_hat[1]
    mz = m_hat[2]
    # convert theta to radians
    theta = theta*math.pi/180
    # term for simplification
    v = 1 - cos(theta)
    # transformation to A from B
    A_R_B = np.array([[sqr(mx)*v + cos(theta), mx*my*v-mz*sin(theta), mx*mz*v+my*sin(theta)],
                      [mx*my*v+mz*sin(theta), sqr(my)*v + cos(theta), my*mz*v-mx*sin(theta)],
                      [mx*mz*v-my*sin(theta), my*mz*v + mx*sin(theta), sqr(mz)*v+cos(theta)]])

    return A_R_B
```

In [10]:
```
def get_axis_and_angle(R):

    # extract matrix componants
    r11 = R[0,0]
    r12 = R[0,1]
    r13 = R[0,2]
    r21 = R[1,0]
    r22 = R[1,1]
    r23 = R[1,2]
    r31 = R[2,0]
    r32 = R[2,1]
    r33 = R[2,2]

    # calculate theta
    cosTheta = (r11+r22+r33-1)/2
    theta    = math.acos(cosTheta)

    if r32 - r23 > 0:
        mx = math.sqrt((r11-cosTheta)/(1-cosTheta))
    else:
        mx = -math.sqrt((r11-cosTheta)/(1-cosTheta))

    if r13 - r31 > 0:
        my = math.sqrt((r22-cosTheta)/(1-cosTheta))
```

```
    else:
        my = -math.sqrt((r22-cosTheta)/(1-cosTheta))

    if r21 - r12 > 0:
        mz = math.sqrt((r33-cosTheta)/(1-cosTheta))
    else:
        mz = -math.sqrt((r33-cosTheta)/(1-cosTheta))

    if math.fabs(mx) > math.fabs(my):
        my = (r12+r21)/(2*mx*(1-cosTheta))
    if math.fabs(mx) > math.fabs(mz):
        my = (r13+r31)/(2*mx*(1-cosTheta))

    return [mx,my,mz], theta*180/math.pi
```

Problem 2.9

In [17]:

```
# first rotation
A_R_C = determine_transformation_about_vector([1,0,0],35)
#print(A_R_C)

# second rotation
C_R_B = determine_transformation_about_vector([0,1,0],120)

# rotation from B to A
A_R_B = A_R_C@C_R_B

# rotation from A to B
B_R_A = A_R_B.transpose()

m, theta =  get_axis_and_angle(B_R_A)

print('Axis of rotation:',m)
print('Angle of rotation:',theta,"\N{DEGREE SIGN}")
```

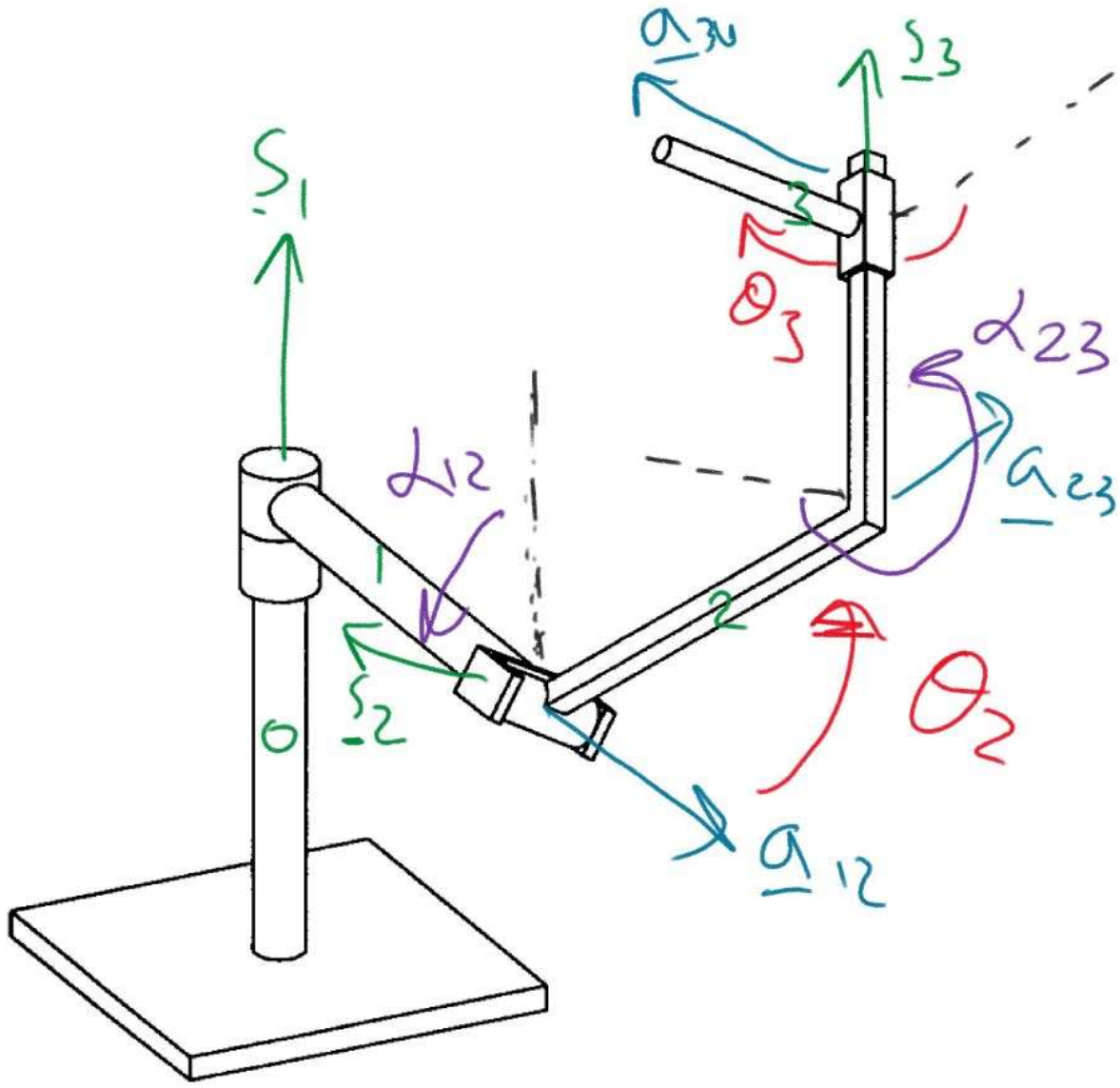
```
Axis of rotation: [-0.1710538177903302, -0.9396607715965427, -0.296273903241481]
Angle of rotation: 123.03915140249427 °
```

Problem 3.4

In [21]:

```
Image('3-4.JPG', width=700)
```

Out[21]:



In [29]:
```
print('The vairable parameters for this manipulator are \u03B8_2, \u03B8_3, & S_3.')
```

```
The vairable parameters for this manipulator are θ_2, θ_3, & S_3.
```