forward kinematics

input: joint variables, output: Cartesian point (n, o, a, p) and (x, y, z, φ, θ, ψ).

```
theta_input=[50,50,50,50,50,50]
```

output: Cartesian point (n, o, a, p) and (x, y, z, φ, θ, ψ).

```
from math import sin,cos,pi,radians,atan2,atan,sqrt,asin,acos
import numpy as np

#setting
d=[0,0,0.149,0.433,0,0]
a=[0,0.432,-0.02,0,0,0]
alpha=[-0.5*pi, 0, 0.5*pi, -0.5*pi, 0.5*pi, 0]
theta=np.radians(theta_input)

#transform matrix(frame i+1 relate to i)
def A(i):
  matrix=np.array([[cos(theta[i])  ,  -sin(theta[i])*cos(alpha[i])   ,  sin(theta[i])*sin(alpha[i])   ,  a[i]*cos(theta
                   [sin(theta[i])  ,  cos(theta[i])*cos(alpha[i])    ,  -cos(theta[i])*sin(alpha[i])  ,  a[i]*sin(theta
                   [0              ,  sin(alpha[i])                  ,  cos(alpha[i])                 ,  d[i]
                   [0              ,  0                              ,  0                             ,  1
  return matrix

#iterate to fin the transform matrix(frame 6 relate to 1)
Cartesian_point=np.identity(4)
for i in range(0,6):
  Cartesian_point=np.dot(Cartesian_point,A(i))

#Computation of the Orientation Angles and Position
x=Cartesian_point[0][3]
y=Cartesian_point[1][3]
```

```
z=Cartesian_point[2][3]
phi = atan(Cartesian_point[1][2]/Cartesian_point[0][2])+pi
thetaa = atan((cos(phi)*Cartesian_point[0][2] + sin(phi)*Cartesian_point[1][2])/Cartesian_point[2][2])+pi
psi = atan((-sin(phi)*Cartesian_point[0][0] + cos(phi)*Cartesian_point[1][0])/(-sin(phi)*Cartesian_point[0][1] + cos(p
output=np.array([x,y,z, phi*180/pi, thetaa*180/pi, psi*180/pi])

#print Ans
np.set_printoptions(precision=4)
print("Cartesian_point(n,o,a,p)=")
print(Cartesian_point)
print("\n (x, y, z, φ, θ, ψ)=",output)
```

```
Cartesian_point(n,o,a,p)=
[[-0.8955  0.4342 -0.0976  0.3407]
 [ 0.1912  0.5734  0.7966  0.6378]
 [ 0.4019  0.6947 -0.5965 -0.3864]
 [ 0.      0.      0.      1.    ]]

 (x, y, z, φ, θ, ψ)= [  0.3407   0.6378  -0.3864  96.9846 126.6226 120.0473]
```

inverse kinematics

input: Cartesian point (n, o, a, p)*italicized text*

```
Cartesian_point=np.array([[-0.895511    ,0.43420623  ,-0.09759607  ,0.34068237],
                          [ 0.19121987  ,0.57343036  ,0.79662575   ,0.63781229],
                          [ 0.40186441  ,0.69472482  ,-0.59654205  ,-0.38642471],
                          [ 0.          ,0.          ,0.           ,1.        ]] )
```

output: the corresponding joint variables.

```
from math import sin,cos,pi,radians,atan2,sqrt,asin,acos
import numpy as np

```

```
#restrict of the corresponding_joint_variables
def check_output(var_array):
  var_range=[160,125,135,140,100,260]
  for i in range(var_array.size):
    if var_array[i]>var_range[i] or var_array[i]<-var_range[i]:
      print('!!!  θ',i+1,'  is out od range  !!!')









f11=Cartesian_point[0][0]
f12=Cartesian_point[0][1]
f13=Cartesian_point[0][2]
px=Cartesian_point[0][3]

f21=Cartesian_point[1][0]
f22=Cartesian_point[1][1]
f23=Cartesian_point[1][2]
py=Cartesian_point[1][3]

f31=Cartesian_point[2][0]
f32=Cartesian_point[2][1]
f33=Cartesian_point[2][2]
pz=Cartesian_point[2][3]

# 8 solutions totally
for c1 in range(2):#2 solutions of theta1
  for c3 in range(2):#2 solutions of theta3
    for c5 in range(2):#2 solutions of theta5
      d3=d[3-1]
      if c1==0:
        cta1=atan2(py,px)-atan2(d3, sqrt(px**2+py**2-d3**2))
      else:
        cta1=atan2(py,px)-atan2(d3, -sqrt(px**2+py**2-d3**2))
```

```
      cta1=atan2(py,px)-atan2(d3,-sqrt(px**2+py**2-d3**2))


    a2=a[2-1]
    a3=a[3-1]
    d3=d[3-1]
    d4=d[4-1]
    M=(px**2+py**2+pz**2-a2**2-a3**2-d3**2-d4**2)/(2*a2)
    if c3==0:
      cta3=atan2(M,sqrt(a3**2+d4**2-M**2))-atan2(a3,d4)
    else:
      cta3=atan2(M,-sqrt(a3**2+d4**2-M**2))-atan2(a3,d4)



    TMP=np.array([[cos(cta1)*px+sin(cta1)*py,-pz],[pz,cos(cta1)*px+sin(cta1)*py]])
    TMP_inv=np.linalg.inv(TMP)
    matrix23=np.dot(TMP_inv, np.array([[a3+a2*cos(cta3)],[d4+a2*sin(cta3)]]))
    cta23=atan2(matrix23[1],matrix23[0])
    cta2=cta23-cta3



    if c5==0:
      cta5=acos( cos(cta1)*sin(cta23)*f13+sin(cta1)*sin(cta23)*f23+cos(cta23)*f33 )
    else:
      cta5=-acos(cos(cta1)*sin(cta23)*f13+sin(cta1)*sin(cta23)*f23+cos(cta23)*f33)



    c4s5 = cos(cta1)*cos(cta23)*f13 + sin(cta1)*cos(cta23)*f23 - sin(cta23)*f33;
    s4s5 = -sin(cta1)*f13 + cos(cta1)*f23;
    s5c6 = -1*(cos(cta1)*sin(cta23)*f11+sin(cta1)*sin(cta23)*f21+cos(cta23)*f31);
    s5s6 = (cos(cta1)*sin(cta23)*f12+sin(cta1)*sin(cta23)*f22+cos(cta23)*f32);
    cta4 = atan2(s4s5, c4s5);
    cta6 = atan2(s5s6, s5c6);
    if cta5<0:
      if cta4>=0:cta4=cta4-pi
```

```
        else :cta4=cta4+pi
        if cta6>=0:cta6=cta6-pi
        else :cta6=cta6+pi
```

```
    corresponding_joint_variables=np.array([cta1,cta2,cta3,cta4,cta5,cta6])*180/pi
    np.set_printoptions(precision=4)
    print("corresponding_joint_variables=",corresponding_joint_variables)
    check_output(corresponding_joint_variables)
    print('-----------------------------------------------------------------------------------------------------------')
```

```
corresponding_joint_variables= [50. 50. 50. 50. 50. 50.]
-----------------------------------------------------------------------------------------------------------
corresponding_joint_variables= [  50.   50.   50. -130.  -50. -130.]
-----------------------------------------------------------------------------------------------------------
corresponding_joint_variables= [ 50.       7.2799 135.2892  97.0136  36.2456 -11.2196]
!!!  θ 3   is out od range  !!!
-----------------------------------------------------------------------------------------------------------
corresponding_joint_variables= [ 50.       7.2799 135.2892 -82.9864 -36.2456 168.7804]
!!!  θ 3   is out od range  !!!
-----------------------------------------------------------------------------------------------------------
corresponding_joint_variables= [-106.2171 -187.2799   50.      -66.5364   20.1631   -9.1114]
!!!  θ 2   is out od range  !!!
-----------------------------------------------------------------------------------------------------------
corresponding_joint_variables= [-106.2171 -187.2799   50.      113.4636  -20.1631  170.8886]
!!!  θ 2   is out od range  !!!
-----------------------------------------------------------------------------------------------------------
corresponding_joint_variables= [-106.2171 -230.      135.2892 -149.3667   38.3558   80.7952]
!!!  θ 2   is out od range  !!!
!!!  θ 3   is out od range  !!!
!!!  θ 4   is out od range  !!!
-----------------------------------------------------------------------------------------------------------
corresponding_joint_variables= [-106.2171 -230.      135.2892   30.6333  -38.3558  -99.2048]
!!!  θ 2   is out od range  !!!
!!!  θ 3   is out od range  !!!
-----------------------------------------------------------------------------------------------------------
```

```
1
```