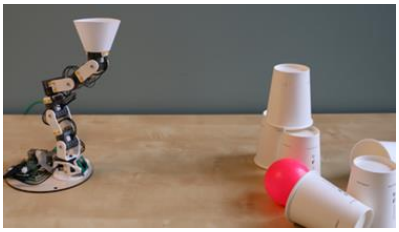
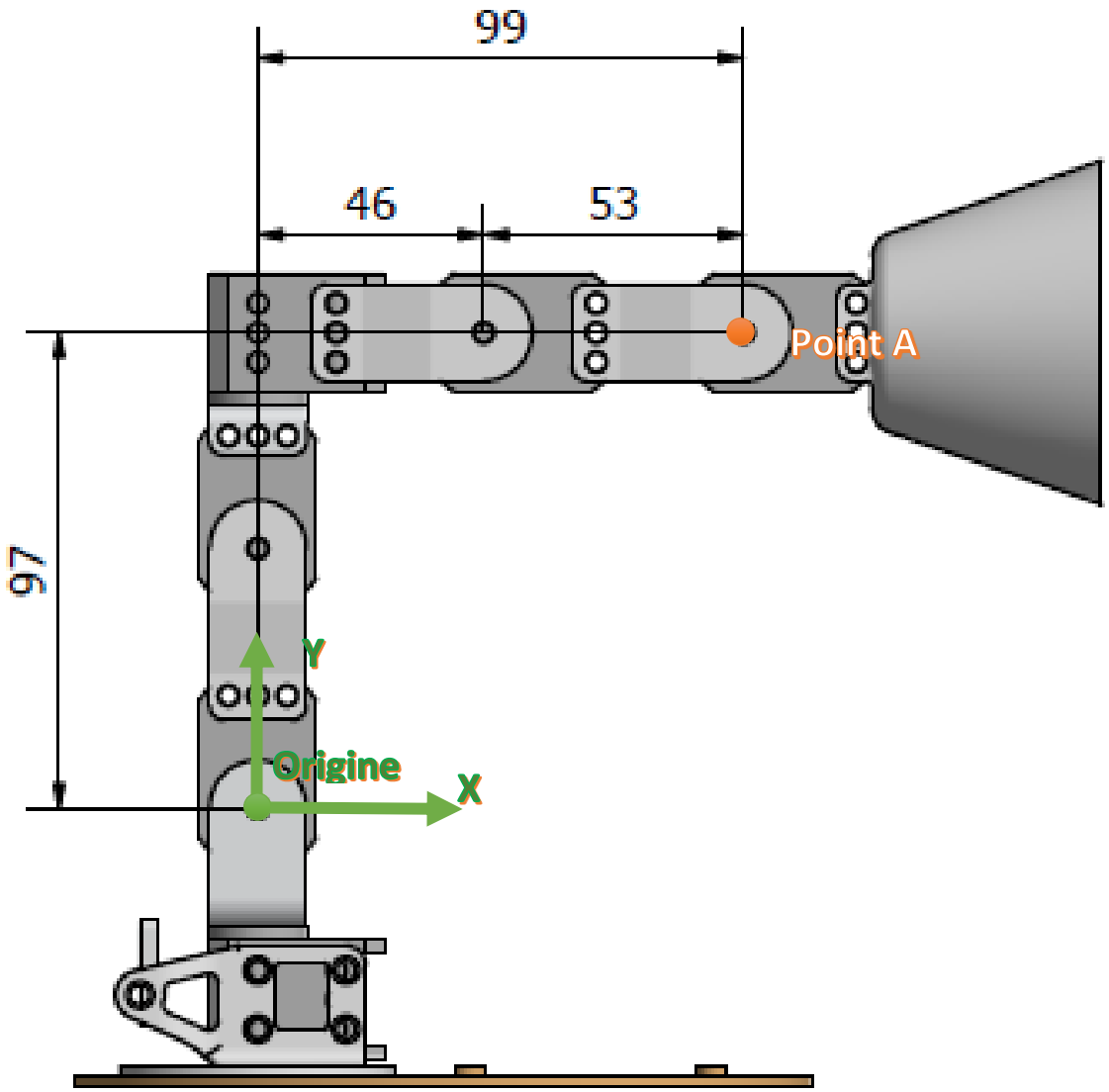


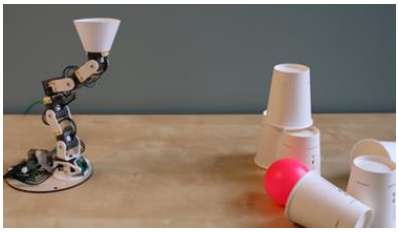
Poppy-ergo-jr

Je programme !

Le robot peut se représenter ainsi en position initiale :

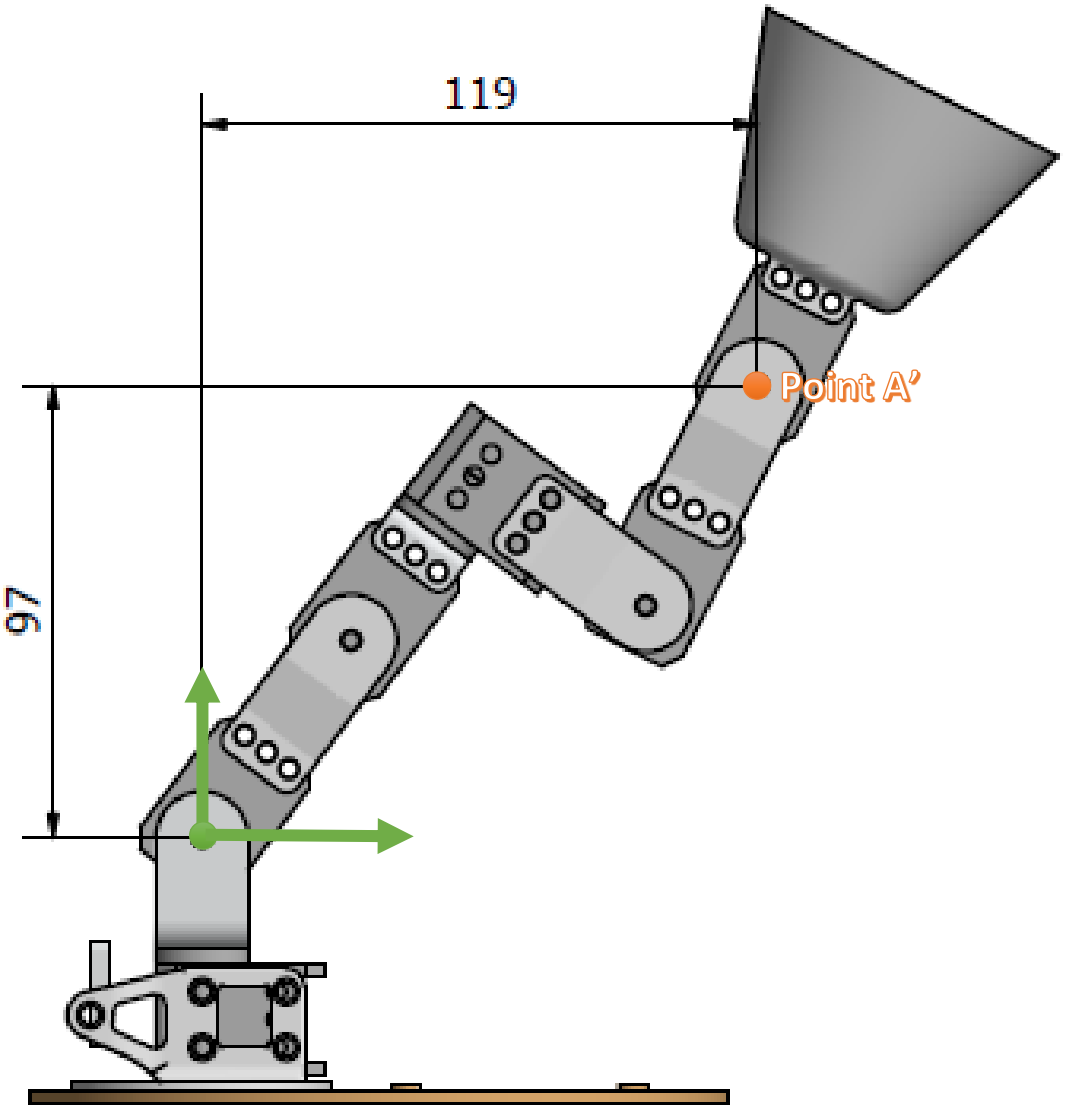


Le point A, centre de la liaison entre le moteur 6 et le bras Poppy est distant de 97 mm de l'origine suivant l'axe Y et 99mm suivant l'axe X.



Poppy-ergo-jr

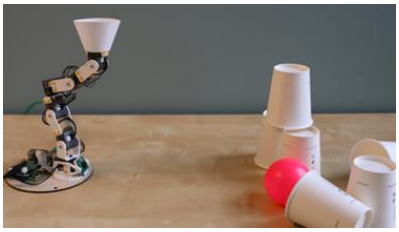
On souhaite faire un programme pour que le point A se déplace comme ceci :



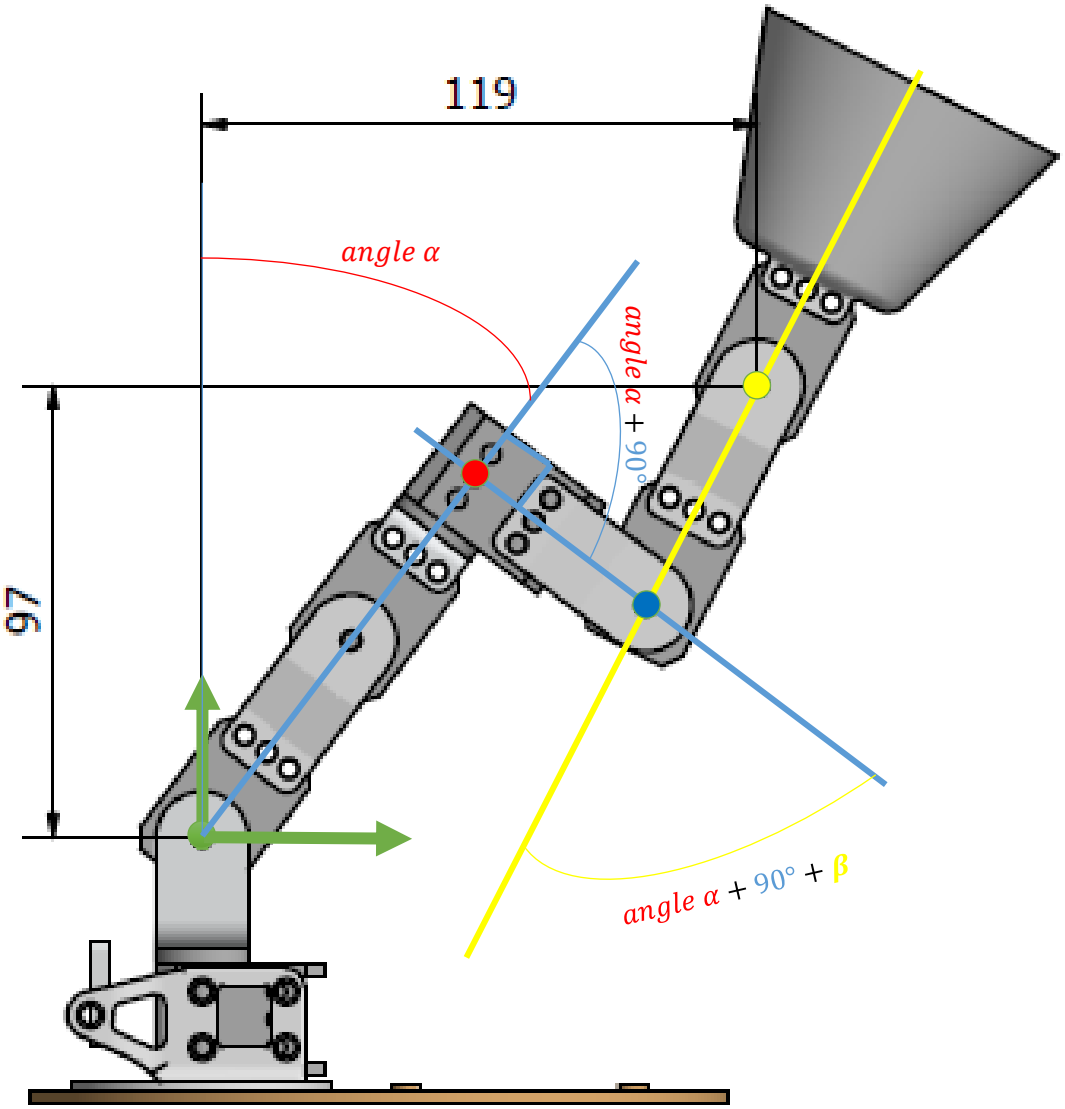
Coordonnées du point A' : X = 119mm et Y = 97mm.

Comment faire ?

On souhaite que le programme calcule le chemin le plus court pour les moteurs M2 et M5 qui seront les seuls utilisés cette fois.



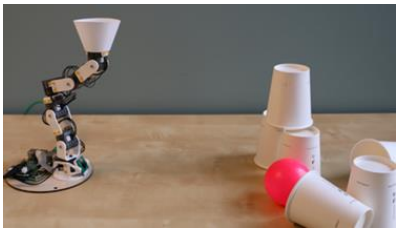
Poppy-ergo-jr



Le programme devra utiliser les calculs suivants pour déterminer la position la plus proche du point d'arrivée :

x = 9.7*sin(radians(a0)) + 4.6*sin(radians(a0+90))+ 5.3*sin(radians(a0+90+a2))

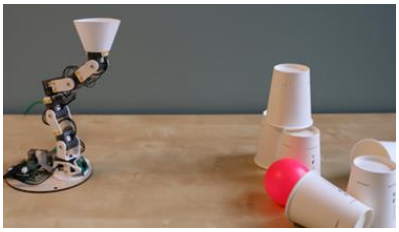
y = 9.7*cos(radians(a0)) + 4.6*cos(radians(a0+90))+ 5.3*cos(radians(a0+90+a2))



Poppy-ergo-jr

Voici le programme en question :

```
from math import cos,sin,radians,sqrt # appel de la bibliothèque de mathématiques et des fonctions cosinus, sinus, radians, et racine (sqrt)

import time #appel de la bibliothèque du temps

ergo.goto_position({'m1': 0,'m2':0,'m3': 0,'m4': 0,'m5': 0,'m6': 0},5,wait=True) #mise en position de poppy dans une position initiale


X = 14.7 #coordonnée X à atteindre

Y = 6.3  #coordonnée Y à atteindre


D = 10 #écart au départ

A0 = 45# angle α au départ

A2 = 45# angle β au départ

for a0 in range (-90,90): # on fait une boucle pour tester degré par degré de -90° à +90° l'angle α

    for a2 in range (120,-120,-1): # on fait la même boucle pour l'angle β mais de -120° à + 120°

        x = 9.7*sin(radians(a0)) + 4.6*sin(radians(a0+90))+ 5.3*sin(radians(a0+90+a2))  #on calcule x pour chaque angle

        y = 9.7*cos(radians(a0)) + 4.6*cos(radians(a0+90))+ 5.3*cos(radians(a0+90+a2))  #on calcule y pour chaque angle

        d = sqrt((X-x)**2 + (Y-y)**2) # à chaque fois, on calcule l'écart entre les coordonnées de départ et les coordonnées qui évolue de degré en degré

            if d < D: #si l'écart est moindre que l'écart au départ, on a donc une position plus performante qu'avant. Si l'écart est plus grand (donc moins performant) il ne se passe rien

                A0 = a0 # on remplace alors l'angle α de départ par l'angle plus performant

                A2 = a2 # on remplace alors l'angle β de départ par l'angle plus performant

                D = d # on remplace alors la valeur de D par ce nouvel écart

pos = {'m1': 0,'m2': A0,'m3': 0,'m4': 0,'m5': A2,'m6': 0} # lorsque tous les angles ont été testé, A0 et A2 contiennent les angles les plus performants

ergo.goto_position(pos, 3, wait=True) # on déplace Poppy
```



Poppy-ergo-jr

Je programme !

Après avoir testé ce programme pour différentes coordonnées, à vous de **rédiger** le programme pour faire écrire ICN par poppy !