

Devoir sur le dessin du workspace du robot, la correction du code pour les limites du robot et un tableau montrant les tests

Pour le tableau si dessous psi sera égale à 0 pour chaque test

X	Y	Z	Position valide	Image
100	100	200	Valide	Fig 1 (rouge)
100	-100	200	Valide	Fig 1 (bleu)
-100	100	200	Valide	Fig 2 (rouge)
-100	-100	200	Valide	Fig 2 (bleu)
300	100	200	Valide	Fig 3 (rouge)
100	300	200	Valide	Fig 3 (bleu)
-400	0	200	<b>Valide</b>	Fig 4 (rouge)
-400	100	200	Non Valide	Fig 4 (bleu)
0	100	200	Non Valide	Fig 5 (rouge)
100	50	200	Non Valide	Fig 5 (bleu)

On observe que le point p1(-400, 0, 200, 0) est considéré comme valide alors qu'il est en dehors du workspace du robot. J'ai aussi testé avec un x = 400 pour vérifier si c'était pas mon dessin qui est inversé et les 2 cas pour y (400 et -400). C'est la 1ère boucle de vérification qui cause ce problème

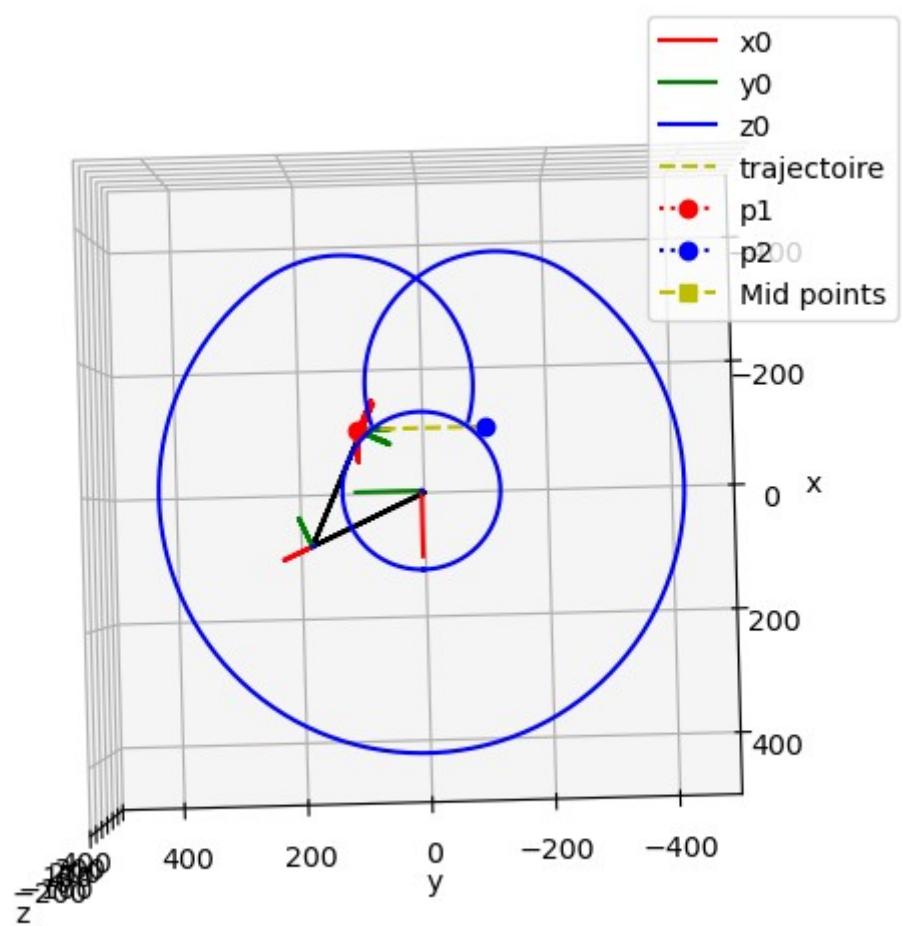
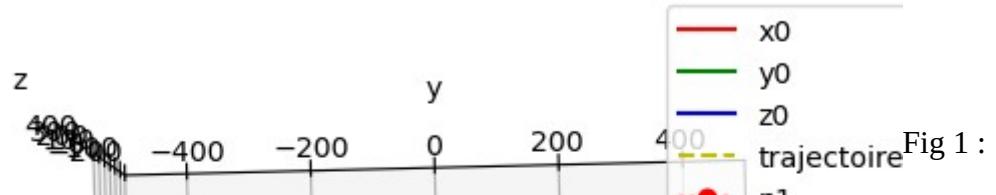
```

226     if (d>120.3 and d <=400):
227         theta = np.rad2deg(np.atan2(y,x))
228         if (theta<=142 or theta>=-142):
229             if(z<=340 and z>=140):
230                 print("A/ Le point est valide")
231                 answer=1
232             else:
233                 print("Le point n'est pas valide")
234                 answer=0
235         else:
236             print("Le point n'est pas valide")
237             answer=0
238

```

Fig 6 : morceau du code pour vérifier la validité d'un point-100

en remplaçant le <or> du if ( $\theta \leq 142$  or  $\theta \geq -142$ ) par un <and> ça a l'aire d'avoir résolu le problème.



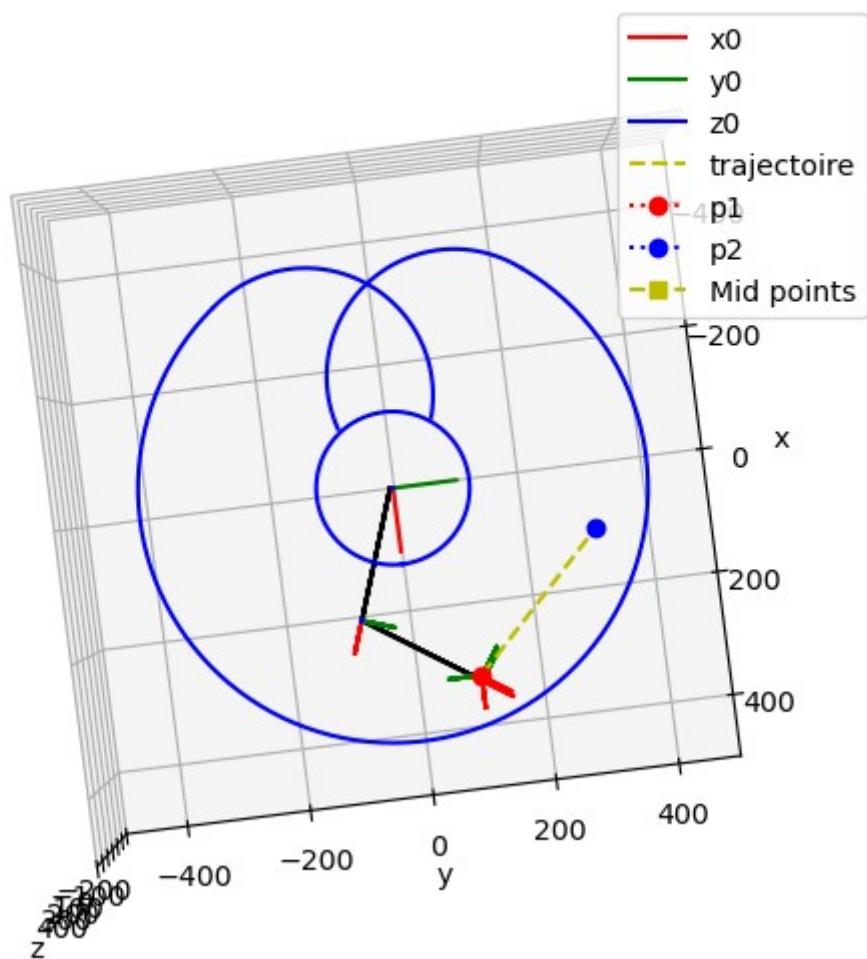


Fig 3 :

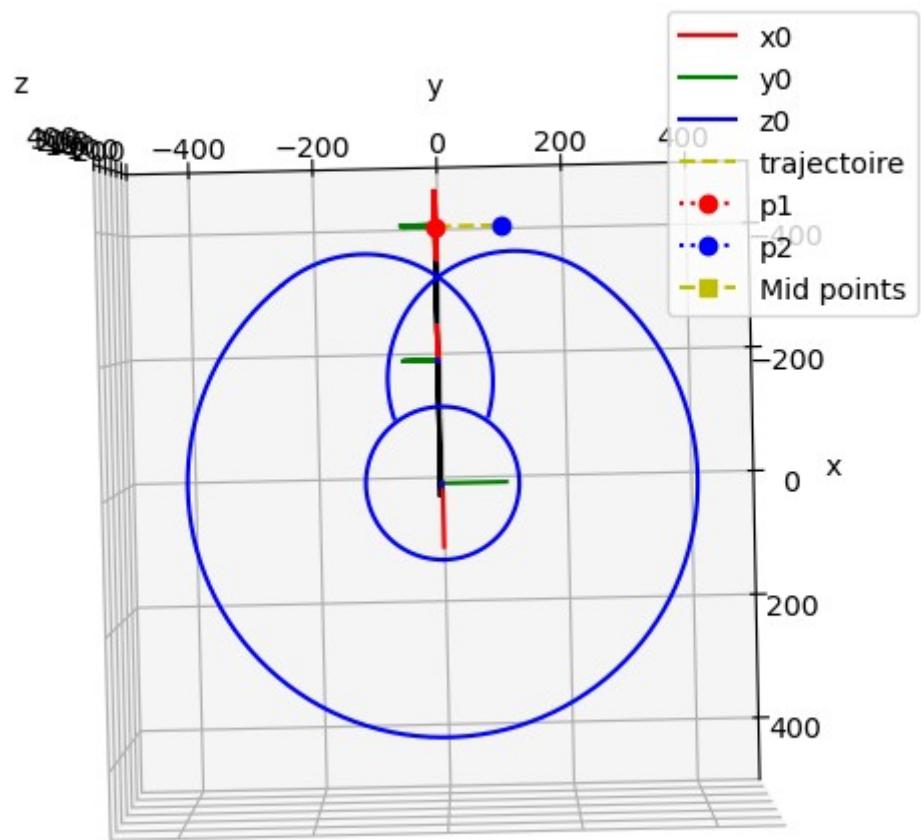


Fig 4 :

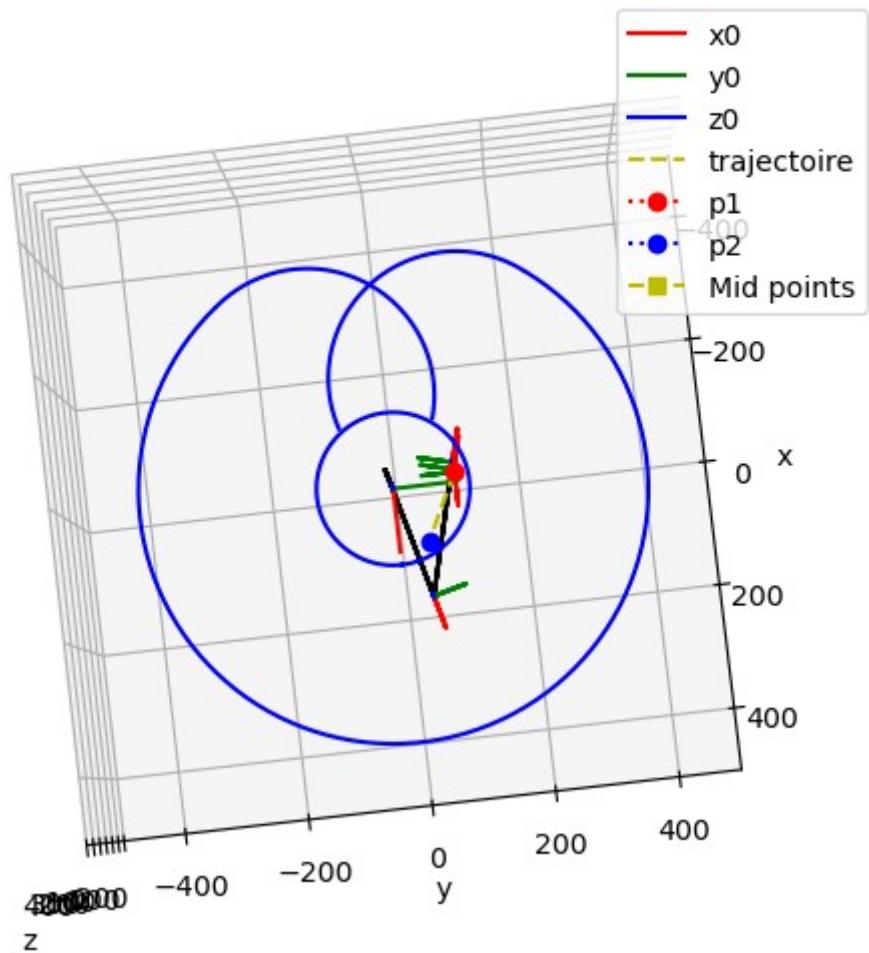


Fig 5 :