

TP - Détection du mouvement

December 7, 2016

Le but de ce TP sera de détecter des objets en mouvement. Le Thymio est immobile, et on déplace devant lui un objet (balle, autre thymio). Il faudra déterminer le sens de ce déplacement.

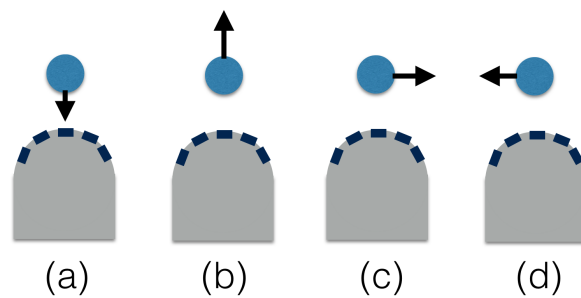


Figure 1: Le Thymio est immobile, l'objet face au Thymio peut (a) s'approcher, (b) s'éloigner, (c,d) longer le Thymio

1 Rappel

Pour rappel, si l'on veut afficher en continue le tableau de proximité, on peut utiliser le programme suivant:

```
def releveDistance():  
    while True:  
        x = yield "prox"  
        x = x[:5]  
        print("capteurs de proximité:",x)  
  
runThymioControl(releveDistance)
```

2 Détection d'un mouvement

1. Ecrire un programme relève en permanence la valeur du capteur de proximité. Votre programme devra enregistrer dans une variable **p1** le tableau de proximité courant, et dans une autre variable **p0** le tableau de proximité précédent.
2. Créez une fonction **diff** qui prenne deux listes de nombres **p0** et **p1** en paramètre, et qui renvoie la différence terme à terme entre **p1** et **p0**.
3. créez dans votre programme la variable **dP**, qui soit la différence entre **p1** et **p0**, et affichez **dP**
4. Reproduisez les 4 situations de la figure 1, et indiquez les valeurs de **dP** pour ces 4 situations.

Remarque:

Voici des valeurs qu'on pourrait obtenir pour **dP** dans les 4 situations de la figure 1:

- (a) **dP**=[13,18,22,15,12]
- (b) **dP**=[-12,-15,-20,-13,-10]
- (c) **dP**=[0,-20,0,20,0]
- (d) **dP**=[0,20,0,-20,0]

```
def diff(p0,p1):  
    # Autre facon d'ecrire cette fonction:  
    # import numpy as np  
    # return list(np.array(p1)-np.array(p0))  
    #  
    dp = []  
    for i in range(len(p0)):  
        dp.append(p1[i]-p0[i])  
    return dp
```

```
def exo2():  
  
    p0 = yield "prox"  
    p0 = p0[:5]  
  
    while True:  
        p1 = yield "prox"  
        p1 = p1[:5]  
  
        dp = diff(p0,p1)  
        print(dp)
```

```

        p0 = p1

runThymioControl(exo2)

```

3 Détection d'un objet qui s'éloigne ou se rapproche

On souhaite détecter les situations décrites sur la Figure 1 en (a) et (b).

1. Affichez la moyenne des termes `dp` (pour cela, utilisez la fonction `sum` de Python).
2. Déplacez des objets devant le robot (selon les 4 situations). Comment la moyenne des termes de `dp` évolue-t-elle ?
3. Votre programme doit afficher "l'objet s'éloigne" ou "l'objet se rapproche" selon qu'il détecte ces deux situations.

```

def exo3():

    p0 = yield "prox"
    p0 = p0[:5]

    while True:
        p1 = yield "prox"
        p1 = p1[:5]

        dp = diff(p0,p1)
        s = sum(dp)

        print(s)
        if sum(dp) > 100:
            print("l'objet s'approche")
        elif sum(dp) < -100:
            print("l'objet s'eloigne")

        p0 = p1

runThymioControl(exo2)

```

4 Détection d'un objet qui longe le Thymio

On souhaite détecter les situations décrites sur la Figure 1 en (c) et (d).

1. Calculez $dP[3]+dP[2]-dP[1]-dP[0]$
2. Utilisez cette mesure pour détecter les objets qui longent le Thymio. Votre programme doit afficher “l’objet longe le Thymio par la droite” ou “l’objet longe le Thymio par la gauche” selon qu’il détecte ces deux situations.

5 Prédiction d’impact

Dans le cas où un objet s’approche du Thymio (Figure 1 (a)), le Thymio doit en fonction de dP et de $p1$ estimer en seconde le temps avant que l’objet ne lui rentre dedans. Pour cela, partez du principe que l’intervalle de temps entre deux mesure est de $1/100$ de secondes.

6 Lissage des mesures

Les relevés de télémètre sont très bruités. Donc, calculer une différence dP basée sur seulement deux instants est très imprécis. Au lieu de ne conserver en mémoire que les deux derniers tableaux $p1$ et $p0$, on va construire une liste L comportant les 10 derniers tableaux relevés. Le dernier élément de cette liste $L[9]$ correspond donc à $p1$. L’avant-dernier $L[8]$ à $p0$. Au lieu de calculer dP comme étant $p1-p0$, on calculera maintenant dP comme $L[9]+L[8]+\dots+L[5]-L[4]-L[3]-\dots-L[0]$. Autrement dit, dP , est donc la différence entre la somme des 5 derniers relevés et la somme des 5 relevés précédents. Relancez vos programme avec ces nouvelles estimations.