

Robot ShapIO

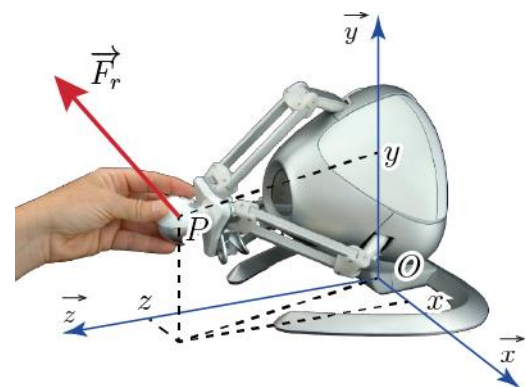
Livret de présentation pédagogique Bac général spé-SI & STI-2D

Système réel et Technologie connectée

Le robot ShapIO est une Interface Homme Machine pour l'immersion en réalité virtuelle ou pour la téléopération. Il est le lien entre **le réel et le numérique**.
Son architecture mécanique est basée sur une **structure « Delta »** à trois mobilités, équipée de trois codeurs et trois moteurs pilotés en courant.
Equipé d'une carte Raspberry Pi et de Python, il devient un Système Haptique pour l'Internet des Objets (ShapIO) - intelligent et connecté.

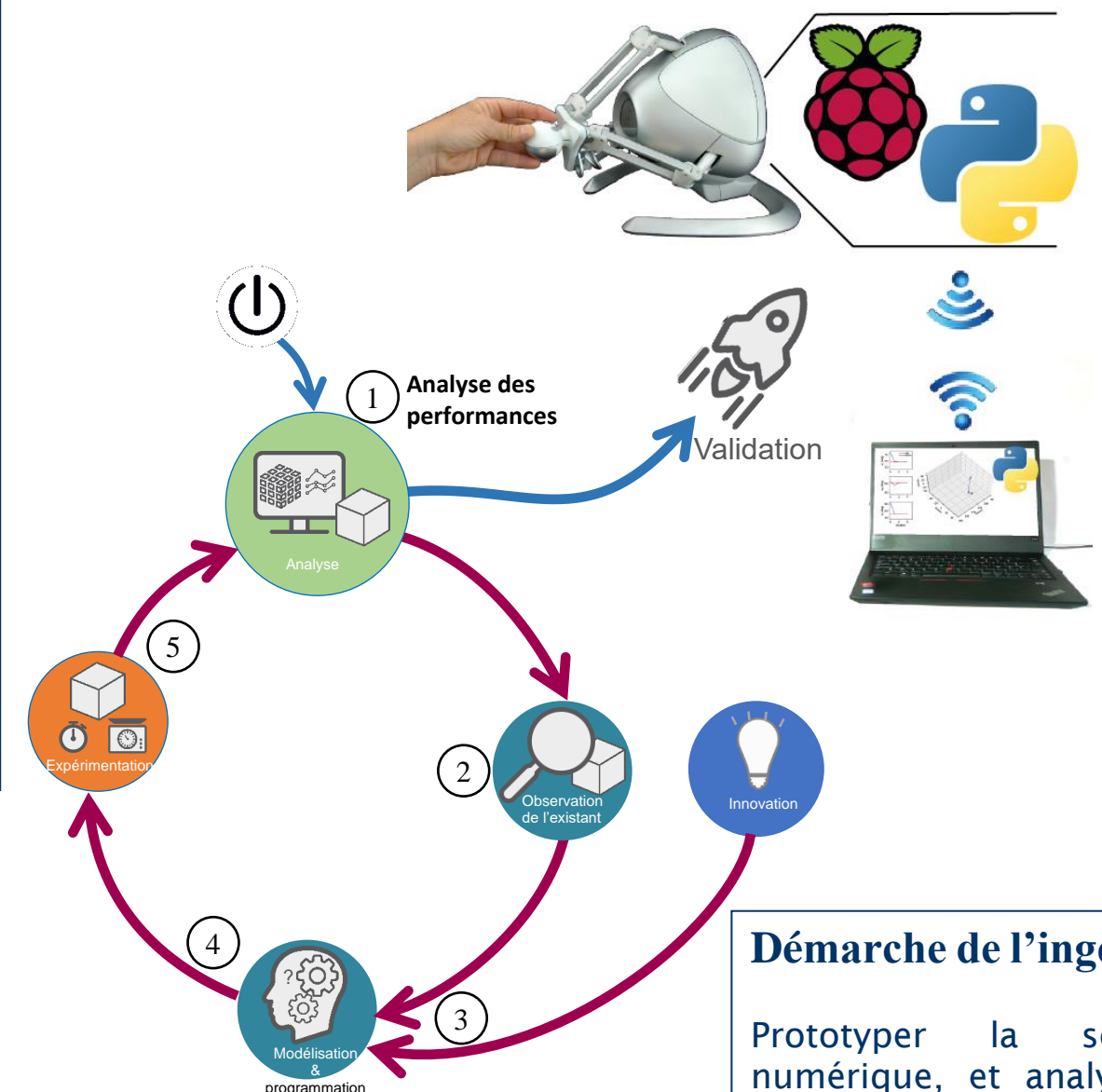
Il est un support d'**activités pratiques** pour l'acquisition des **fondamentaux** de l'informatique au service des Sciences de l'ingénieur : algorithmie & programmation, codage de l'information, conception informationnelle des produits (instrumentation et pilotage) et transmission de l'information (réseaux IP & Bluetooth).

Mais il reste aussi totalement ouvert, connecté et interfaçable pour un usage dans le cadre de **projets** (pilotage d'un drone, téléopération par exemple).



Haptique ?

Pour réaliser une tâche virtuelle ou distante, l'opérateur indique son intention en déplaçant la poignée.
Il ressent aussi un effort restitué : ce retour sensoriel est qualifié d'**haptique**.



Démarche de l'ingénieur

Prototyper la solution numérique, et analyser le gain de performance



Compétences et connaissances visées

- Analyser le traitement de l'information
- Analyser les principaux protocoles pour un réseau de communication et les supports matériels
- Traduire un algorithme en un programme exécutable
- Caractériser les échanges d'informations
- Relever les grandeurs caractéristiques d'un protocole de communication

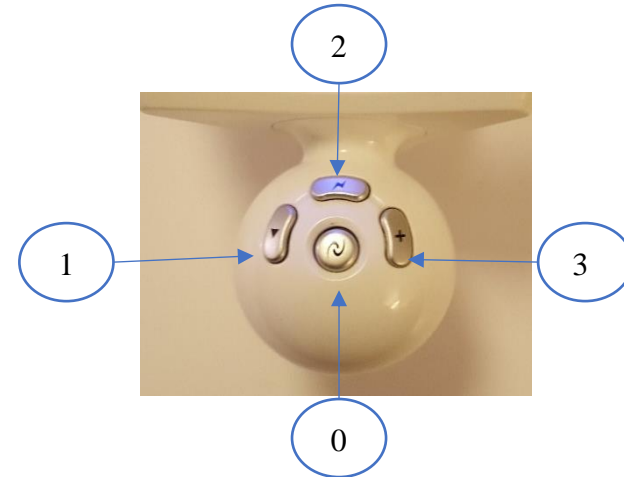
Exemples d'activités pédagogiques proposées

ACQUERIR ET TRANSMETTRE L'ETAT DES BOUTONS

Première partie :

Acquérir l'information

Codage de l'information : binaire - décimal



Bouton	0	1	2	3
Poids	1	2	4	8
Etat (0 ou 1)				
N	}			

```

Open ()

N=GetButtonMask ()
print(format (N, '04b'))

# ou
for i in range (4):
    print ("bouton.", i, ":", N%2)
    N=N//2

Close ()
    
```

Seconde partie :

Transmettre l'information : paramétrage d'une communication IP

Exploration de trames IP : encapsulation de l'information

```

27
28 for i in range(10):
29     N=GetButtonMask ()
30     msg_envoi = str(N).encode ()
31     print (N)
32     connexion_avec_client.send (msg_envoi)
33     time.sleep(1)
    
```

```

0000 7C A1 77 28 F2 62 B8 27 EB F0 3C 9C 08 00 45 00 |.w(.b.'...<...E.
0010 00 36 C4 A5 40 00 40 06 FB 3A AC 11 11 07 AC 11 |.6..@.@:.....
0020 11 B8 32 00 A4 F8 5D 01 9A DD E3 DE 05 CE 80 18 |..2...].....
0030 01 FE C5 03 00 00 01 01 08 0A FB 63 78 EE 00 DA |.....cx...
0040 D5 EC 31 32 |..12
    
```

7C A1 77 28 F2 62 : très explicite, adresse MAC de la destination

B8 27 EB F0 3C 9C : très explicite, adresse MAC de la source

AC 11 11 07 : adresse IP de la source codée en hexadécimal

AC 11 11 B8 : adresse IP de la destination codée en hexadécimal

32 00 : port 12800 codée en hexadécimal

A4 F8 : port 42232 codée en hexadécimal

31 32 → 12 : la donnée est en ASCII, elle se reconnaît facilement. La table ASCII indique le caractère 1 pour le code hexadécimal 31, et 2 pour le code hexadécimal 32



COMPETENCES ET CONNAISSANCES VISEES

- Analyser le traitement de l'information
- Modéliser les mouvements
- Rendre compte de résultats
- Quantifier les écarts de performances entre les valeurs attendues et les valeurs mesurées

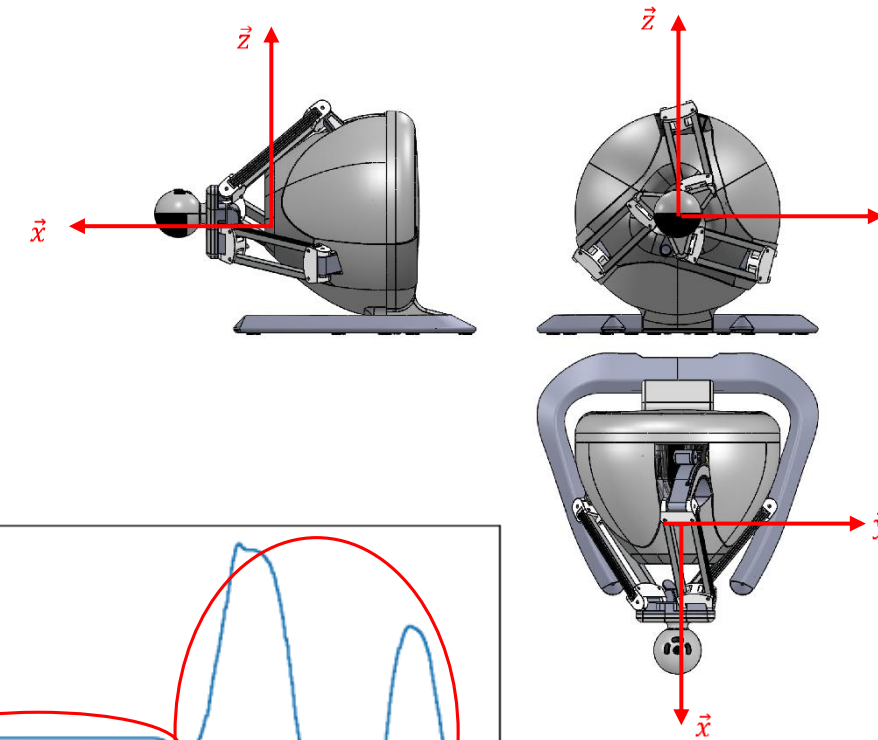
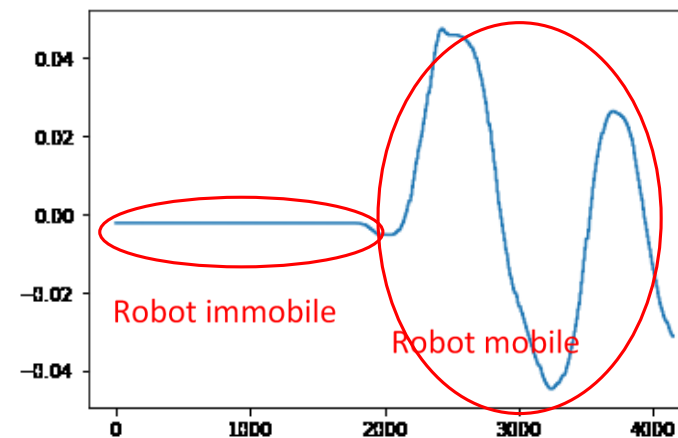
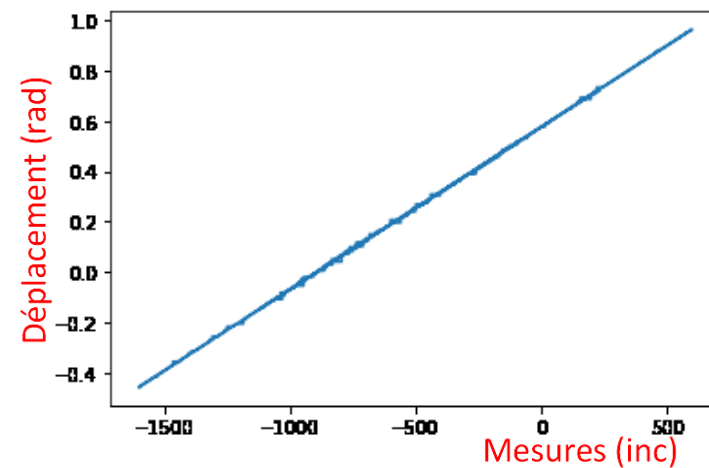
Exemples d'activités pédagogiques proposées

ACQUERIR LA POSITION DE LA POIGNEE

Première partie :

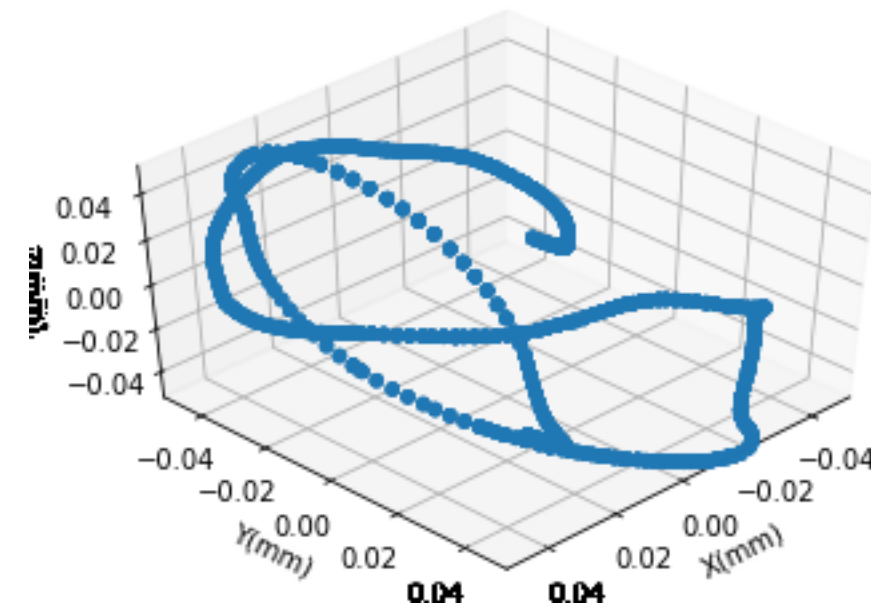
Repérer et paramétrer le mécanisme
Analyse les moyens d'acquisition
Lier la mesure brute et la position
Représenter les déplacements :

- En fonction du temps
- Dans l'espace



```
X=[]
Y=[]
Z=[]

while GetButtonMask()==0:
    position=vecteur(GetPosition()[0:3])
    X.append(position[0])
    Y.append(position[1])
    Z.append(position[2])
```



COMPETENCES ET CONNAISSANCES VISEES

- Traduire le comportement attendu ou observé d'un objet
- Traduire un algorithme en un programme exécutable
- Analyser des résultats d'expérimentation et de simulation
- Quantifier les écarts de performances entre les valeurs attendues, les valeurs mesurées et les valeurs obtenues par simulation

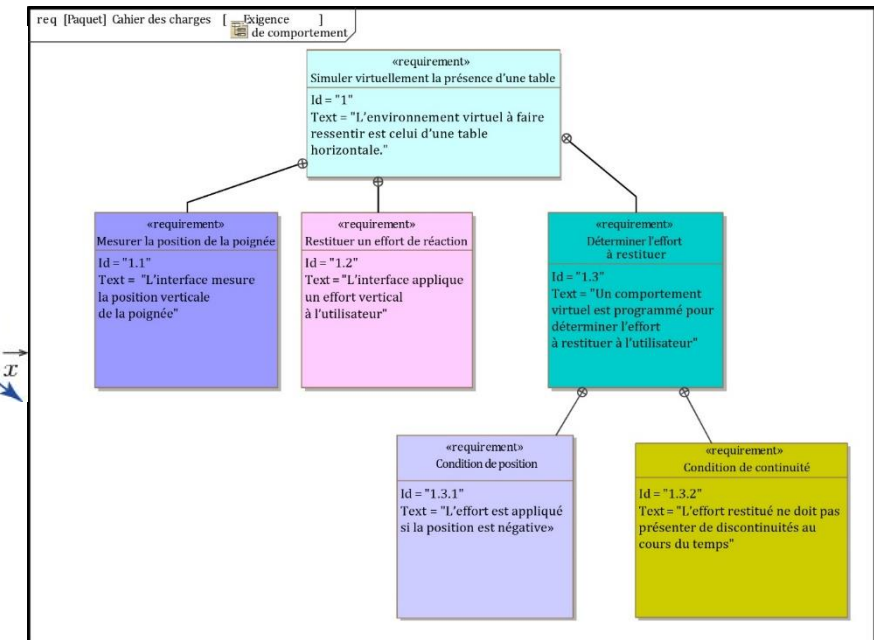
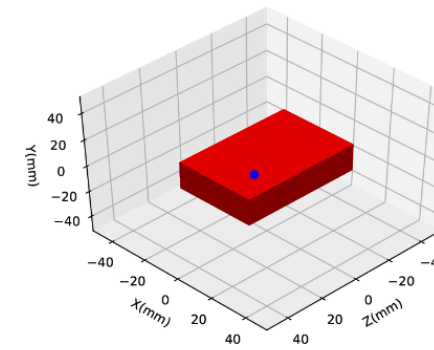
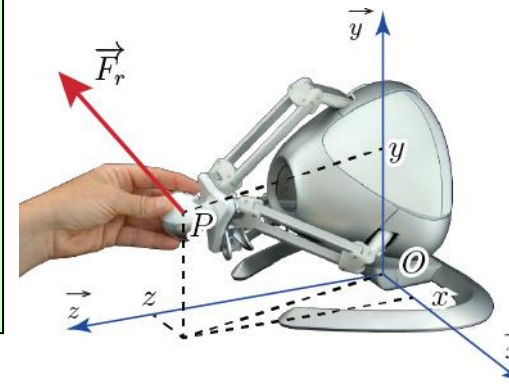
Exemples d'activités pédagogiques proposées

SIMULER UN ENVIRONNEMENT HAPTIQUE TABLE HORIZONTALE

Repérer et paramétrer le mécanisme
Analyse les moyens d'acquisition et de pilotage
Traduire le comportement exigé en algorithme puis en programme

```

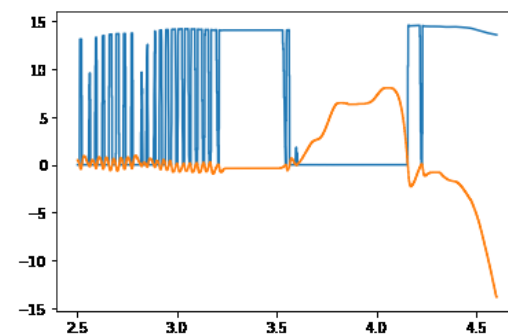
Open ()
while GetButton (0) == .0 :
  if GetPosition () [2] < 0 :
    SetForce (0, 0, 15)
  else :
    SetForce (0, 0, 0)
Close ()
  
```



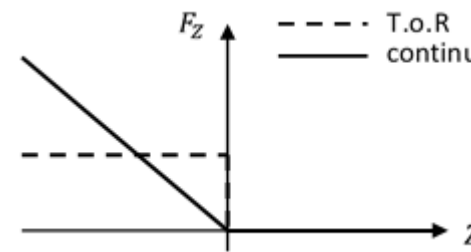
- Matérialiser une solution virtuelle (Prototypage de la commande)
- Analyser le traitement de l'information
- Traduire le comportement attendu ou observé d'un objet (Structures algorithmiques)
- Traduire un algorithme en un programme exécutable
- Modifier les paramètres influents et le programme de commande en vue d'optimiser les performances du produit

AMELIORER UN ENVIRONNEMENT HAPTIQUE

Analyser la performance actuelle
Identifier la cause des vibrations
Modifier un programme
Valider la nouvelle performance



Vibrations dues à un pilotage tout ou rien (ToR)

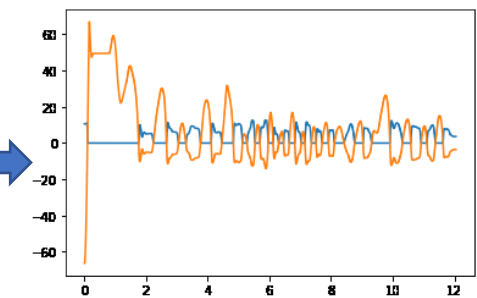


Définition d'un meilleur comportement

```

if z < 0 :
  SetForce (0, 0, -1000 * z)
else :
  SetForce (0, 0, 0)
  
```

Modification du programme



Validation du nouveau pilotage



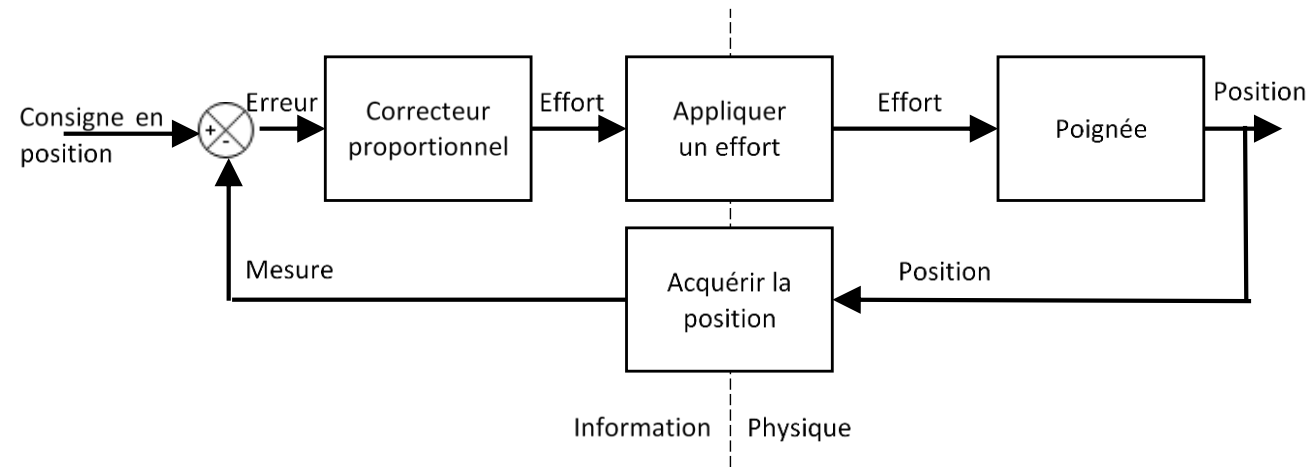
COMPETENCES ET CONNAISSANCES VISEES

- Analyser le traitement de l'information
- Traduire un algorithme en un programme exécutable
- Analyser le comportement d'un système asservi
- Associer un modèle à un système asservi
- Modifier les paramètres influents et le programme de commande en vue d'optimiser les performances du produit
- Rendre compte de résultats

Exemples d'activités pédagogiques proposées

ASSERVISSEMENT DE LA POSITION DE LA POIGNEE

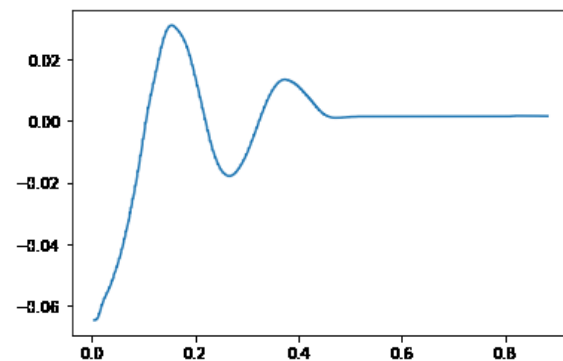
Structure d'asservissement
Traduction en algorithme puis en programme



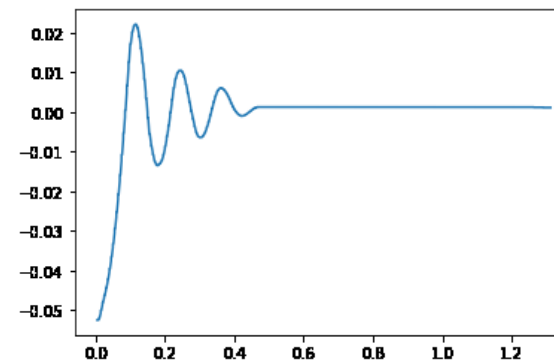
```
while GetButton(0) == 0:
    z=GetPosition()[2]
    ecart=consigne-z
    Fz=Kp*ecart
    SetForce(0,0,Fz)
```

Comparaison des performances en fonction des paramètres du correcteur

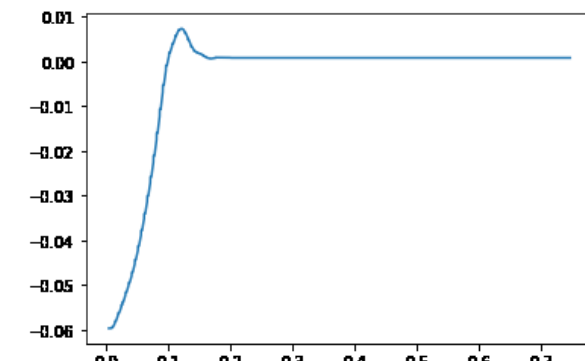
- Influence du K_p (facteur de correction proportionnelle)
- Influence du K_d (facteur de correction dérivée)



$K_p = 250 \text{ N/m}$



$K_p = 750 \text{ N/m}$



$K_p = 750 \text{ N/m}$
 $K_d = 10 \text{ N/(m/s)}$

www.setdidact.com