

# Utiliser un circuit gyro Gy521

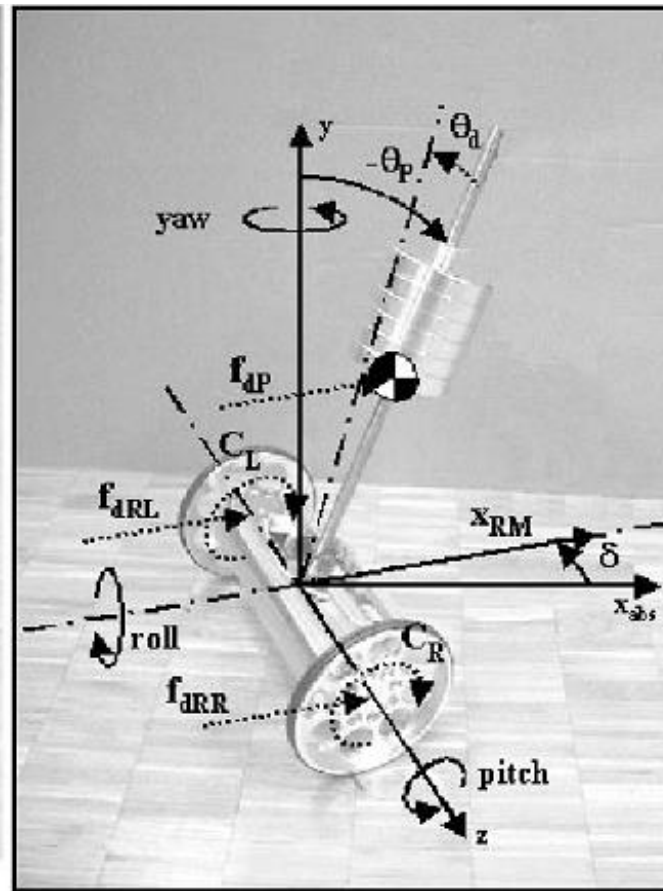
## Base du réglage PID

Microclub, 17 mai 2019

J.D. Nicoud



Figure 1. JOE



Thèse EPFL 1999

Si on connaît exactement les masses et les caractéristiques des moteurs, la théorie permet de tout calculer.

Pour nous autres pauvres bricoleurs, les problèmes de réglage sont abstraits, et des calculs mathématiques précis n'ont pas de sens si les données sont peu précises..

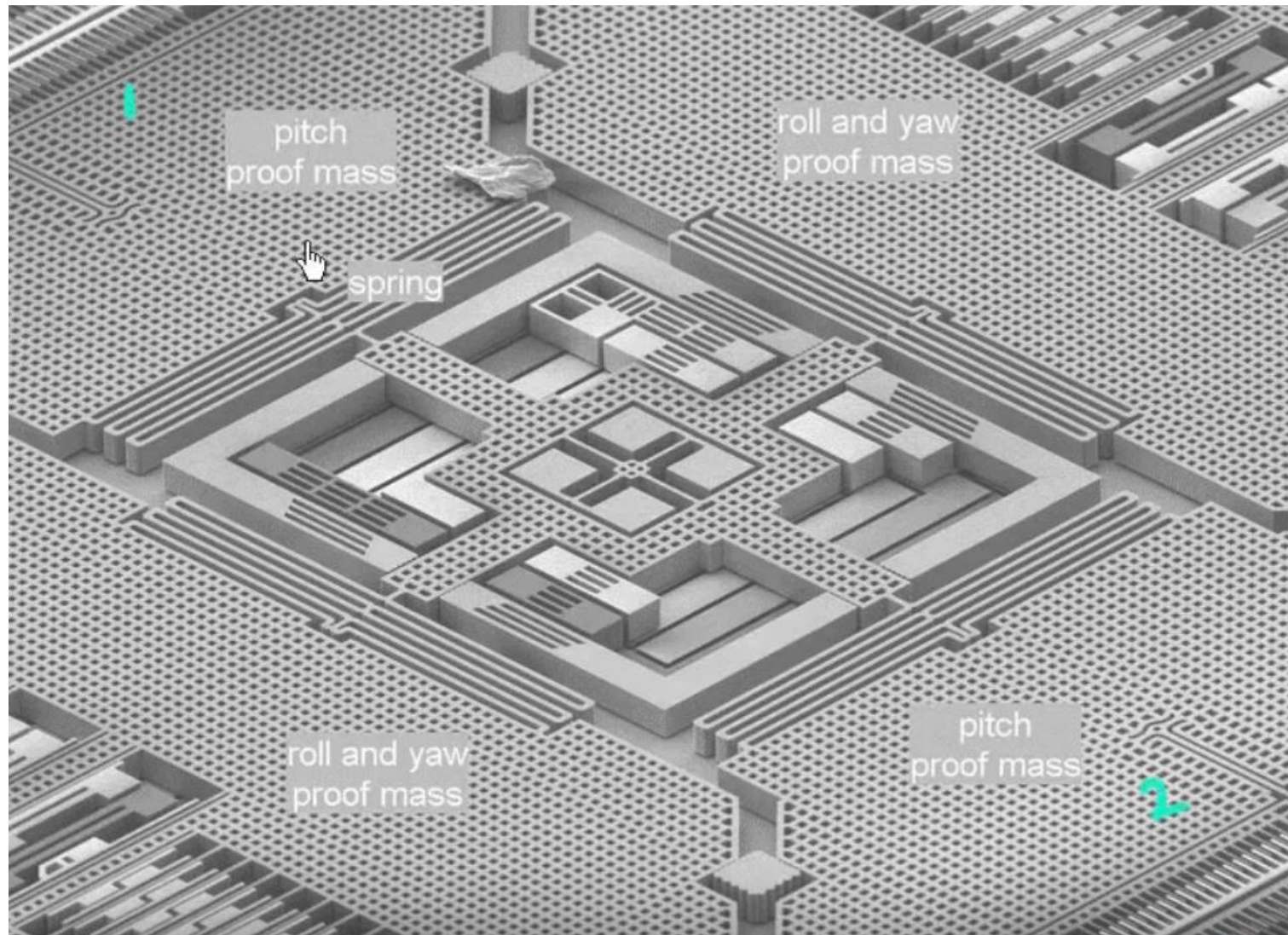
### **Capteurs**

Problèmes de précision, de bruit, de temps de réponse-

### **Actionneurs**

Manques de puissance, inertie inconnue, couple irrégulier.





**Technologie MEMS (dimension < 2mm)**

## Configuration

Les registres internes sont accessibles via une interface I2C:

R104/0x68=0 power management register 1

- doit être mis à zéro

R25/0x19 Sample Rate Divider

- par défaut 1kHz, peut être diminué

R26/0x1A Configuration +/- 1G par défaut

- option 2G et 4G

Ignorer les 120 autres registres

- calibrés à la fabrication

La mise à jour des registres de donnée toutes les 4ms

Transfert I2C à 100kb/s, durée 1,2ms pour les 6 valeurs 16 bits

## Lecture I2C des 3 valeurs gyroscope

```
Lib Wire //2412b lib standard Arduino
```

```
Wire.beginTransmission(MPU);
```

```
Wire.write(0x43); //starts with MPU register 43
```

```
Wire.endTransmission(false);
```

```
Wire.requestFrom(MPU,6,true); //requests 6
```

```
registers
```

```
GyX=Wire.read()<<8|Wire.read();
```

```
GyY=Wire.read()<<8|Wire.read();
```

```
GyZ=Wire.read()<<8|Wire.read();
```

```
Lib I2cTwi //760b lib Dide1
```

```
GyX=ReadWordAt(0x43);
```

```
GyY=ReadWordAt(0x45);
```

```
GyZ=ReadWordAt(0x47);
```

# Résultats de lecture

Horizontal z=1G 1g is given as 16384

AcX= +01508	AcY= -00468	AcZ= +17400
AcX= +01448	AcY= -00396	AcZ= +17404
AcX= +01592	AcY= -00532	AcZ= +17376

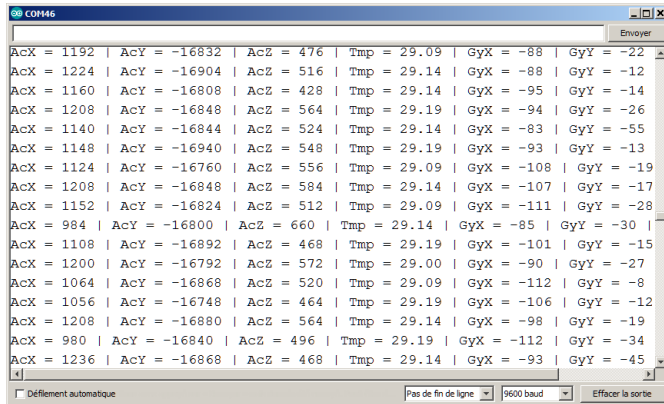
Vertical y =1g

AcX= +01520	AcY= +16132	AcZ= +01096
AcX= +01596	AcY= +16096	AcZ= +00876
AcX= +01592	AcY= +16080	AcZ= +00888

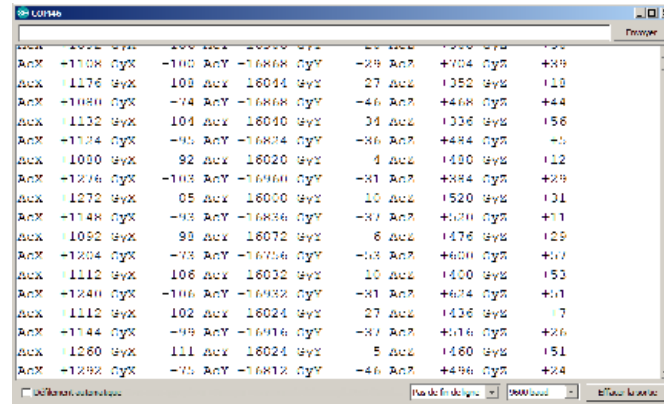
Basculé 1 x=1g

AcX= +17696	AcY= -00540	AcZ= +00508
AcX= +17676	AcY= -00532	AcZ= +00684
AcX= +17728	AcY= -00600	AcZ= +00652

Le terminal Arduino enlève les zéros non significatifs et l'alignement se perd. Ce n'est pas le cas de la librairie git didel/Terser utilisée ci-dessus



```
COM146
AcX = 1192 | AcY = -16832 | AcZ = 476 | Tmp = 29.09 | GyX = -88 | GyY = -22
AcX = 1224 | AcY = -16904 | AcZ = 516 | Tmp = 29.14 | GyX = -88 | GyY = -12
AcX = 1160 | AcY = -16808 | AcZ = 428 | Tmp = 29.14 | GyX = -95 | GyY = -14
AcX = 1208 | AcY = -16848 | AcZ = 564 | Tmp = 29.19 | GyX = -94 | GyY = -26
AcX = 1140 | AcY = -16844 | AcZ = 524 | Tmp = 29.14 | GyX = -83 | GyY = -55
AcX = 1148 | AcY = -16940 | AcZ = 548 | Tmp = 29.19 | GyX = -93 | GyY = -13
AcX = 1124 | AcY = -16760 | AcZ = 556 | Tmp = 29.09 | GyX = -108 | GyY = -19
AcX = 1208 | AcY = -16848 | AcZ = 584 | Tmp = 29.14 | GyX = -107 | GyY = -17
AcX = 1152 | AcY = -16824 | AcZ = 512 | Tmp = 29.09 | GyX = -111 | GyY = -28
AcX = 984 | AcY = -16800 | AcZ = 660 | Tmp = 29.14 | GyX = -85 | GyY = -30
AcX = 1108 | AcY = -16892 | AcZ = 468 | Tmp = 29.19 | GyX = -101 | GyY = -15
AcX = 1200 | AcY = -16792 | AcZ = 572 | Tmp = 29.00 | GyX = -90 | GyY = -27
AcX = 1064 | AcY = -16868 | AcZ = 520 | Tmp = 29.09 | GyX = -112 | GyY = -8
AcX = 1056 | AcY = -16748 | AcZ = 464 | Tmp = 29.19 | GyX = -106 | GyY = -12
AcX = 1208 | AcY = -16880 | AcZ = 564 | Tmp = 29.14 | GyX = -98 | GyY = -19
AcX = 980 | AcY = -16840 | AcZ = 496 | Tmp = 29.19 | GyX = -112 | GyY = -34
AcX = 1236 | AcY = -16868 | AcZ = 468 | Tmp = 29.14 | GyX = -93 | GyY = -45
```



```
COM146
AcX = 1192 | AcY = -16832 | AcZ = 476 | Tmp = 29.09 | GyX = -88 | GyY = -22
AcX = 1224 | AcY = -16904 | AcZ = 516 | Tmp = 29.14 | GyX = -88 | GyY = -12
AcX = 1160 | AcY = -16808 | AcZ = 428 | Tmp = 29.14 | GyX = -95 | GyY = -14
AcX = 1208 | AcY = -16848 | AcZ = 564 | Tmp = 29.19 | GyX = -94 | GyY = -26
AcX = 1140 | AcY = -16844 | AcZ = 524 | Tmp = 29.14 | GyX = -83 | GyY = -55
AcX = 1148 | AcY = -16940 | AcZ = 548 | Tmp = 29.19 | GyX = -93 | GyY = -13
AcX = 1124 | AcY = -16760 | AcZ = 556 | Tmp = 29.09 | GyX = -108 | GyY = -19
AcX = 1208 | AcY = -16848 | AcZ = 584 | Tmp = 29.14 | GyX = -107 | GyY = -17
AcX = 1152 | AcY = -16824 | AcZ = 512 | Tmp = 29.09 | GyX = -111 | GyY = -28
AcX = 984 | AcY = -16800 | AcZ = 660 | Tmp = 29.14 | GyX = -85 | GyY = -30
AcX = 1108 | AcY = -16892 | AcZ = 468 | Tmp = 29.19 | GyX = -101 | GyY = -15
AcX = 1200 | AcY = -16792 | AcZ = 572 | Tmp = 29.00 | GyX = -90 | GyY = -27
AcX = 1064 | AcY = -16868 | AcZ = 520 | Tmp = 29.09 | GyX = -112 | GyY = -8
AcX = 1056 | AcY = -16748 | AcZ = 464 | Tmp = 29.19 | GyX = -106 | GyY = -12
AcX = 1208 | AcY = -16880 | AcZ = 564 | Tmp = 29.14 | GyX = -98 | GyY = -19
AcX = 980 | AcY = -16840 | AcZ = 496 | Tmp = 29.19 | GyX = -112 | GyY = -34
AcX = 1236 | AcY = -16868 | AcZ = 468 | Tmp = 29.14 | GyX = -93 | GyY = -45
```



On voit que les résultats sont bruités.  
Faut-il passe par un moyennage?

Moyenne simple

Moyenne glissante

Moyenne pondérée

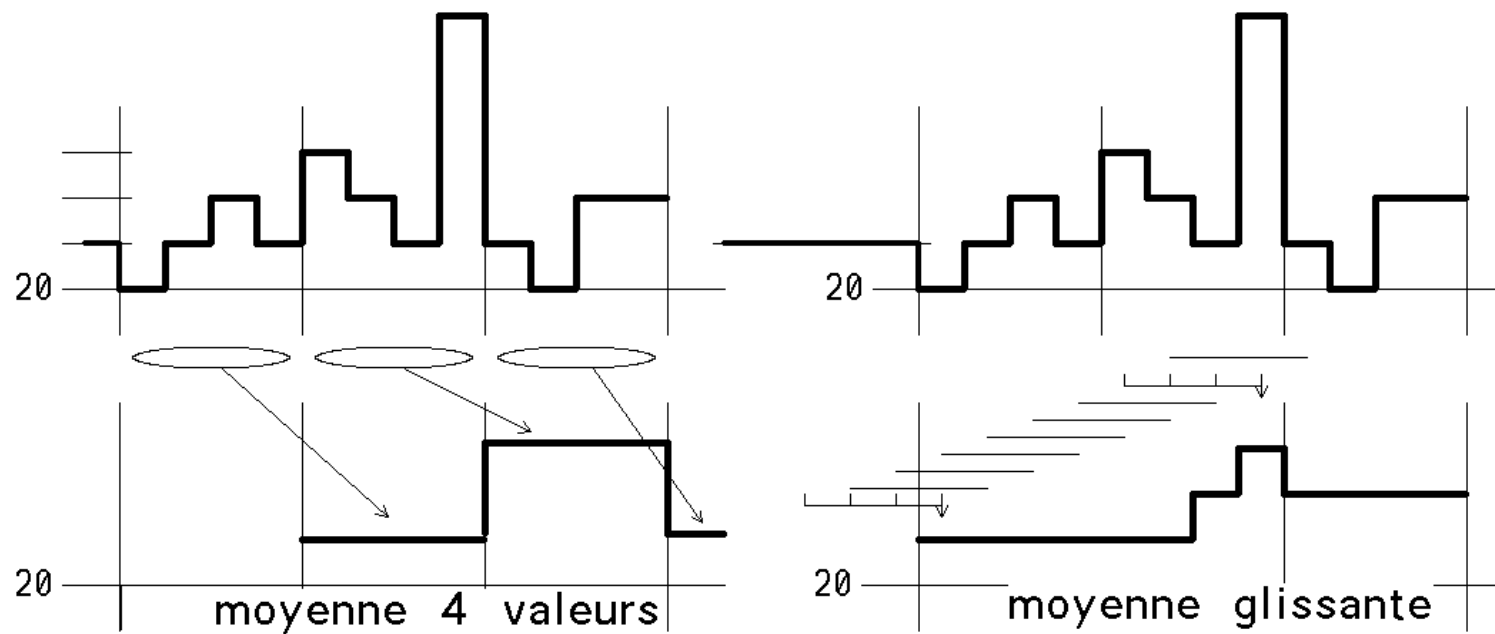
Non illustré, parfois complexe à mettre en oeuvre

Moyenne quadratique, harmonique

Kalman filter

Complementary filter

Particle filter



La moyenne glissante se calcule facilement et n'introduit pas pas de délais.



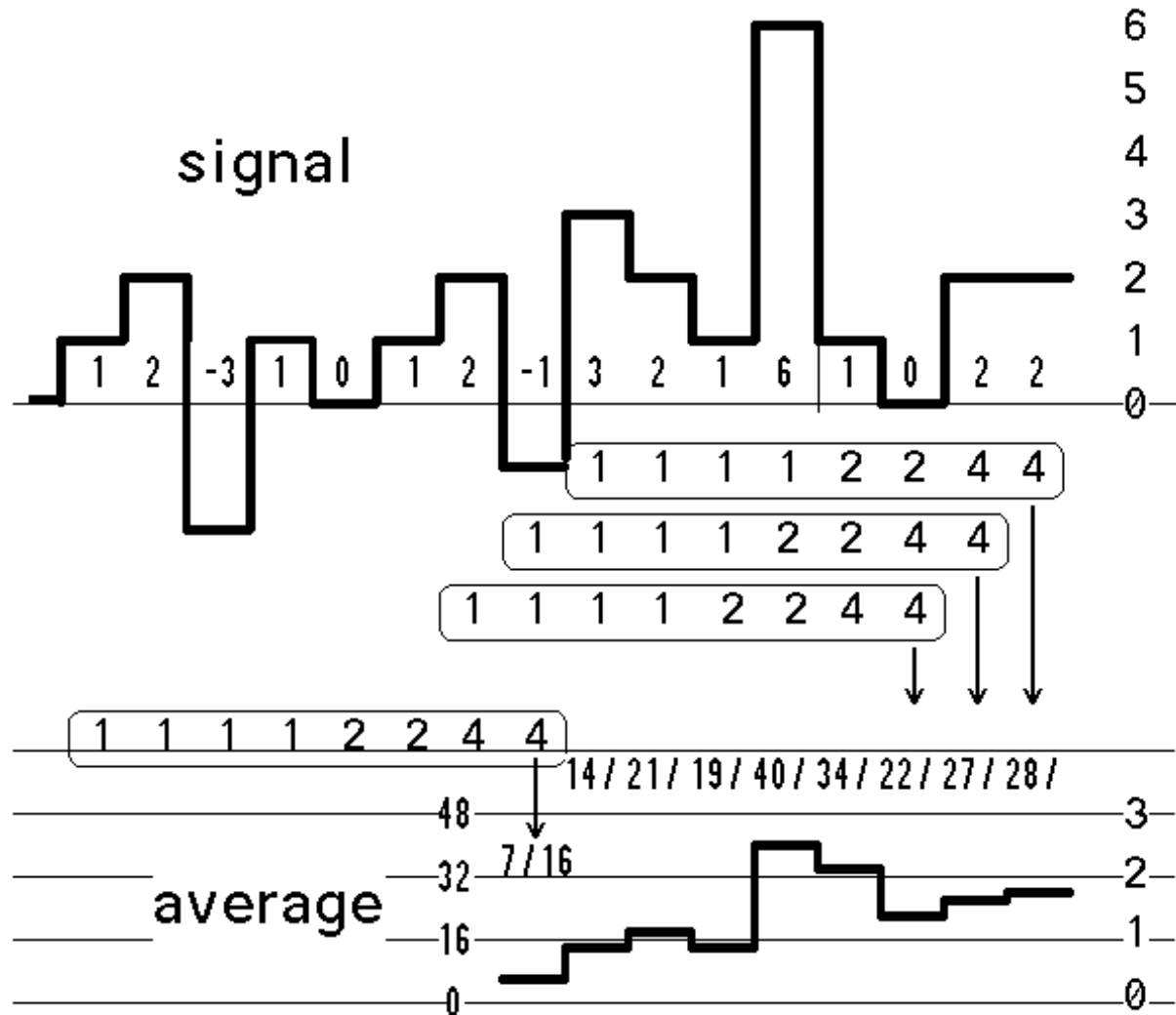
Gy521 initial values

iniAcZ	+1632		iniGyX	-387	
AcZ	+1576	-3 ‰	GyX	-367	+1 ‰
AcZ	+1592	-2 ‰	GyX	-389	0 ‰
AcZ	+1732	+6 ‰	GyX	-386	0 ‰
AcZ	+1676	+2 ‰	GyX	-389	0 ‰
AcZ	+1636	0 ‰	GyX	-385	0 ‰
AcZ	+1692	+3 ‰	GyX	-389	0 ‰
AcZ	+1624	0 ‰	GyX	-358	+1 ‰
AcZ	+1636	0 ‰	GyX	-378	0 ‰
AcZ	+1596	-2 ‰	GyX	-365	+1 ‰
AcZ	+1724	+5 ‰	GyX	-356	+1 ‰
AcZ	+1644	0 ‰	GyX	-388	0 ‰
AcZ	+1592	-2 ‰	GyX	-368	+1 ‰
AcZ	+1672	+2 ‰	GyX	-379	0 ‰
AcZ	+1640	0 ‰	GyX	-382	0 ‰
AcZ	+1564	-4 ‰	GyX	-372	0 ‰

Même avec un moyennage, les valeurs fluctuent. Ce qui compte, c'est la précision, calculée ci-dessus en pour-mille

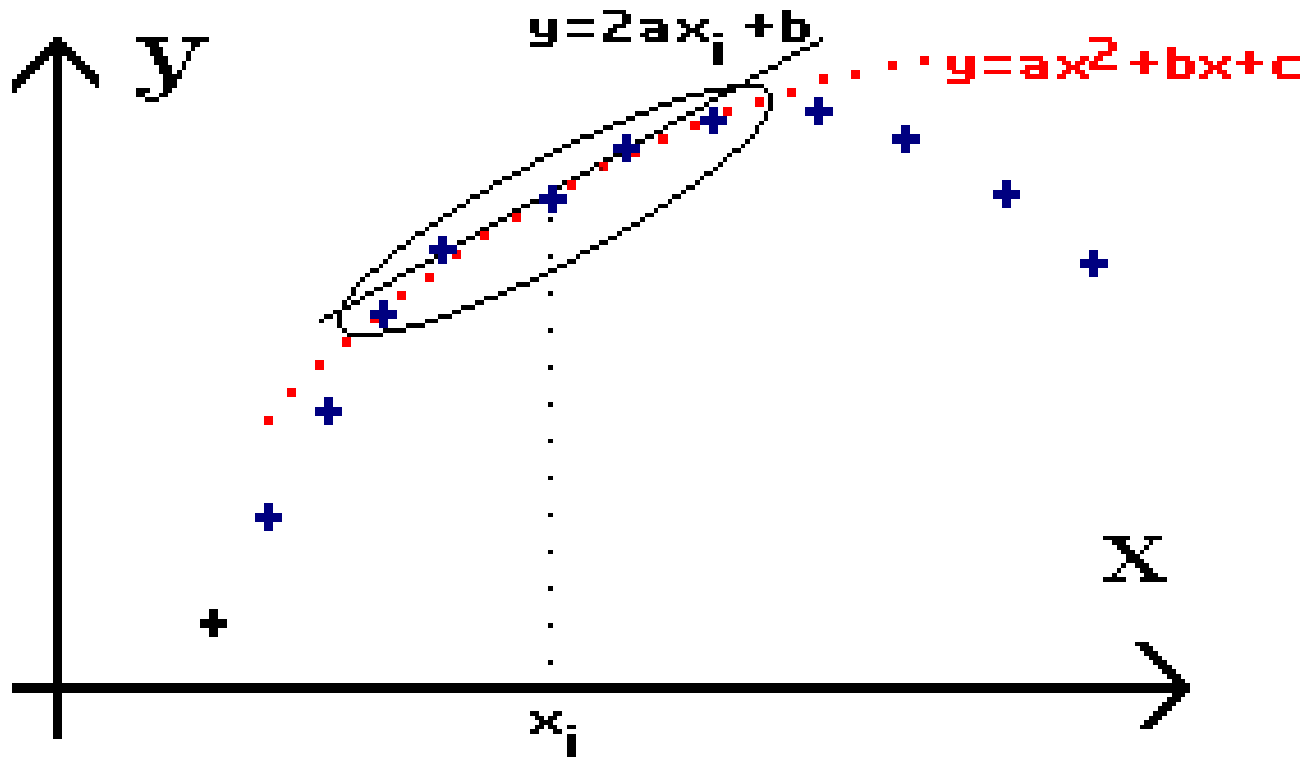
**Always think relative and not absolute**

Moyenne pondérée, les valeurs récentes ont ici plus de poids (joue le rôle de filtre passe-haut)



# Dérivée (vitesse d'évolution)

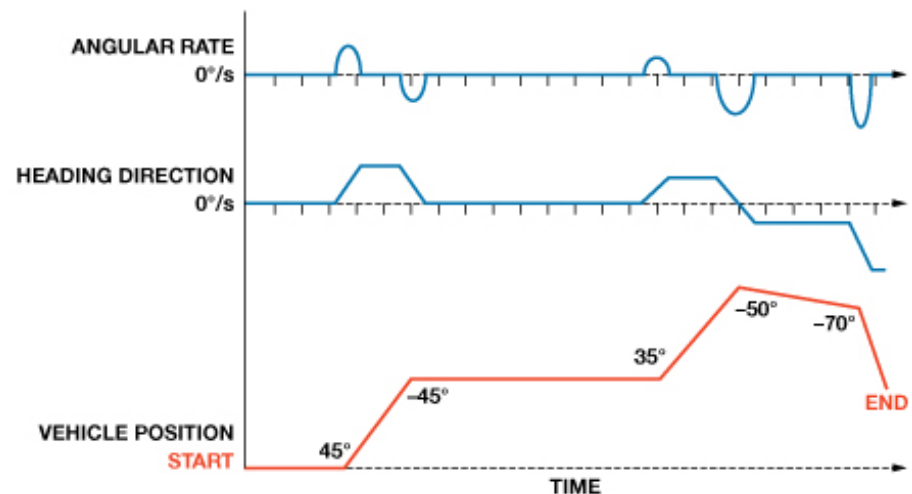
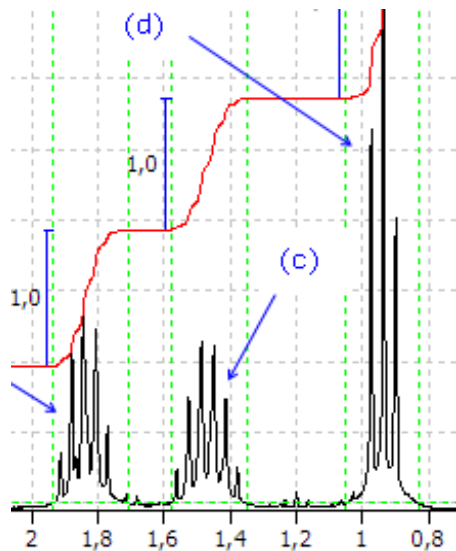
Le dérivée est très sensible au bruit des mesures.  
Il faut lisser sur plusieurs mesures consécutives.



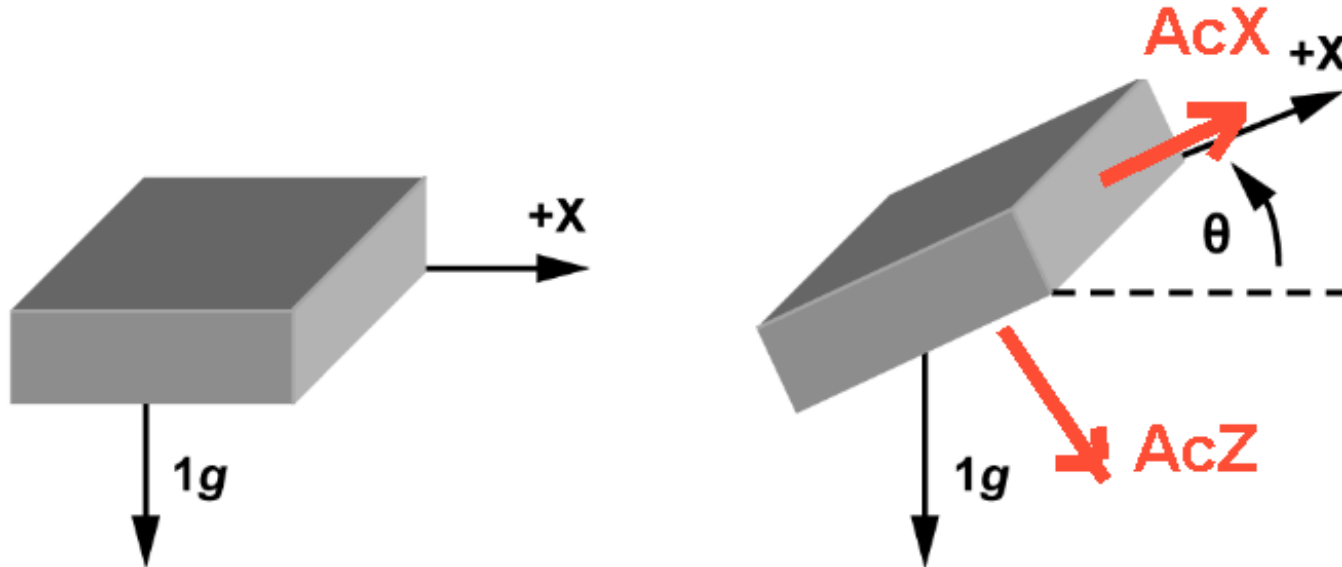
# Intégration

On somme les valeurs pour avoir par exemple la distance à partir de la vitesse (qui est la distance par unité de temps).  
L'intégrale du couple gyroscopique est proportionnelle à l'angle.

$$\text{angle} = \sum_{i=1}^n r_i \times \tau$$



L'accélération varie avec l'angle du capteur.  
Ici,  $AcX$  varie rapidement. On pourrait intégrer  $GyY$  à partir de la position initiale. et comparer.



$$AcZ = \cos(\theta) \quad \cos 5 \text{ degrés} = 0.99$$

$$AcX = \sin(\theta) \quad \sin 5 \text{ degrés} = 0.08$$

Pour des petits angles, une approximation linéaire est bien plus rapide à calculer que via la librairie trigonométrique.

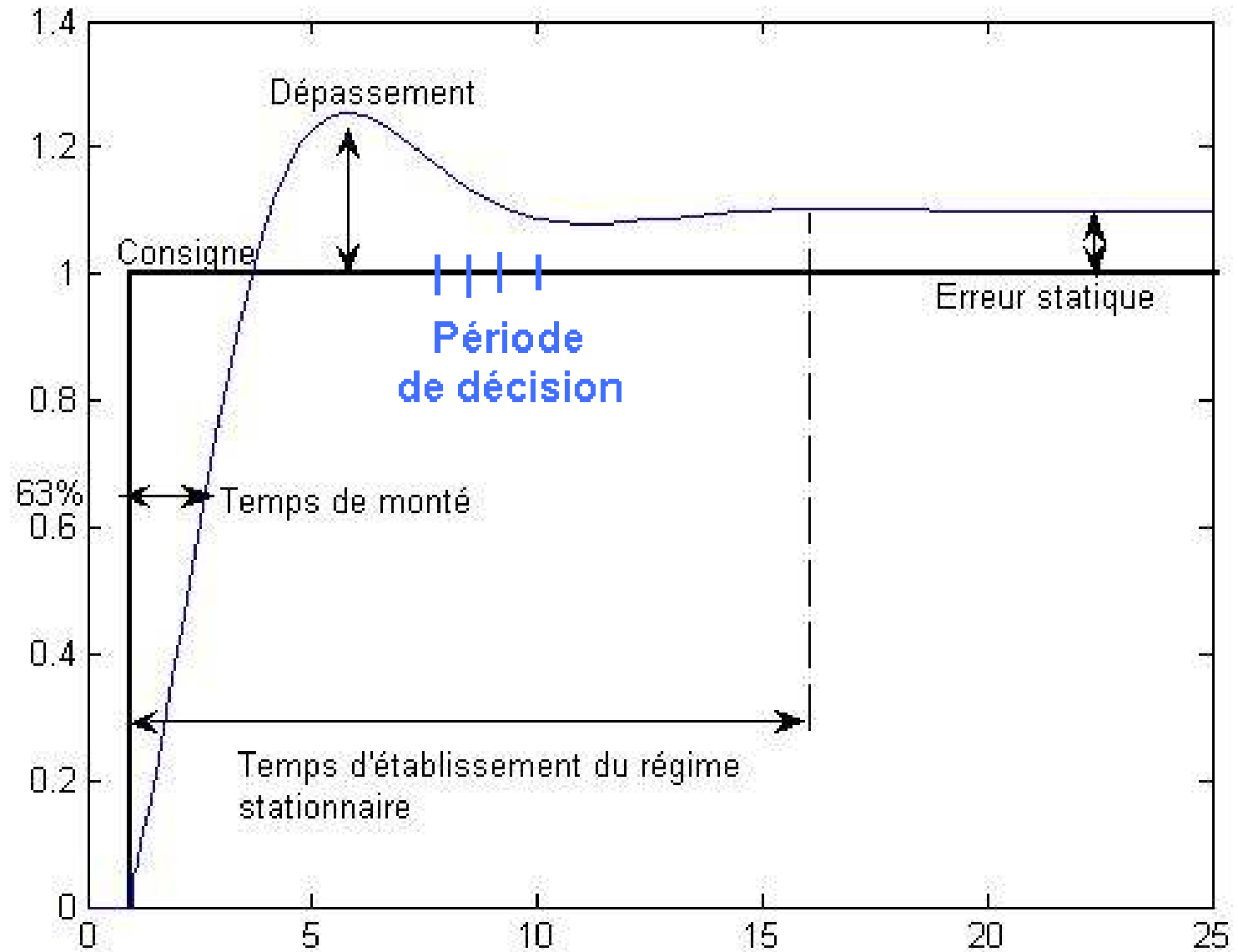
# **Moteurs, actuateurs**

**Beaucoup trop à dire!  
Autant de problèmes et  
solutions pour améliorer  
qu'avec les capteurs**

**Passons directement au  
problème du réglage:**

**On est dans une position initiale  
et le mécanisme doit aller vers  
valeur de consigne**





## **Pensons à un servo de télécommande:**

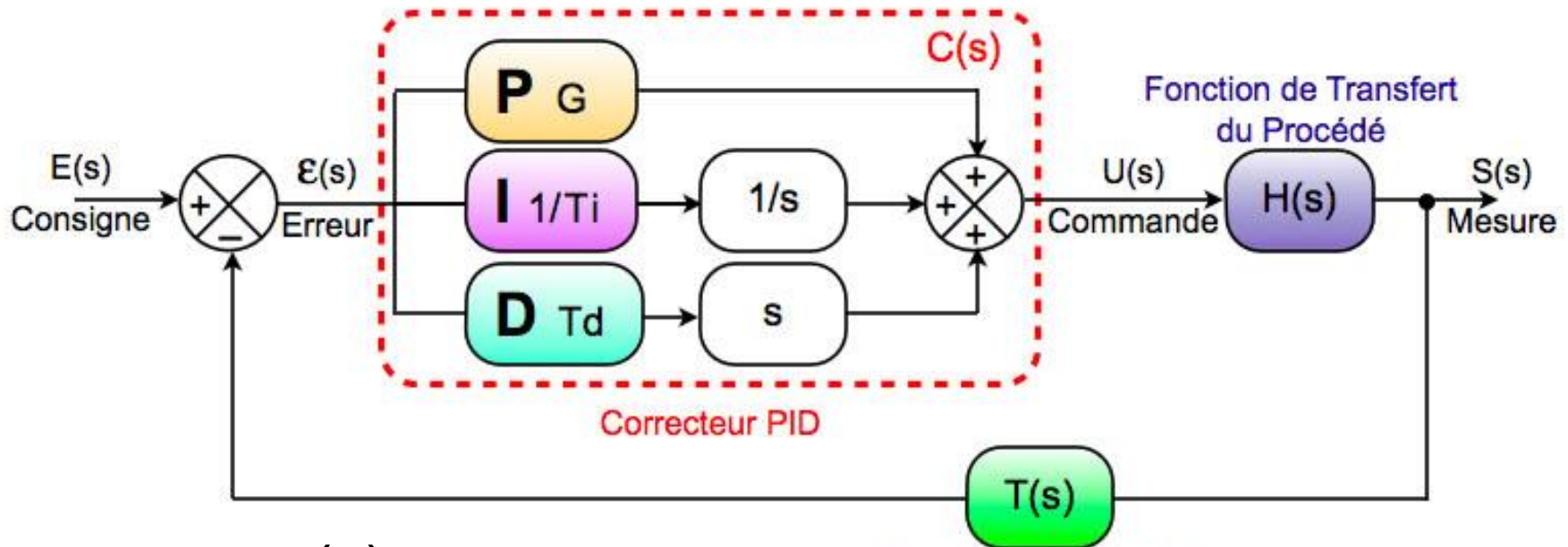
Si la commande moteur est proportionnelle à l'erreur (différence entre angle actuel et angle désiré) on a un réglage **P** pour proportionnel.

Le frottement peut empêcher d'atteindre la position voulue ; c'est l'erreur statique. Pour corriger on ajoute une fraction de l'intégrale de l'erreur. C'est la composante **I** pour intégration.

S'il y a des oscillations, il faut tenir compte de la variation de l'écart. C'est la composante **D**, pour dérivée.

# PID

Toutes les 20 ms par exemple, on effectue 3 multiplications. La difficulté est de calculer avec d'avoir des bons paramètres.



$e(s)$  error value

$U(s) =$  motor control

$K_p * e(s) +$  prop coefficient

$K_i * \text{int}(e(s)) +$  integral coefficient

$K_d * \text{diff}(e(s))$  differential coefficient



# M5Stack

**Ce robot est programmé en Python.**

**On voit (en cherchant bien pour trouver la doc)  
qu'il n'y a apparemment pas de moyennage sur les  
mesures du gyro.**

**On peut facilement modifier les paramètre pour  
chercher à obtenir l'équilibre.**

**La construction est propre et les outils logiciels  
abondant, si on sait se débrouiller avec une  
documentation minimale.**

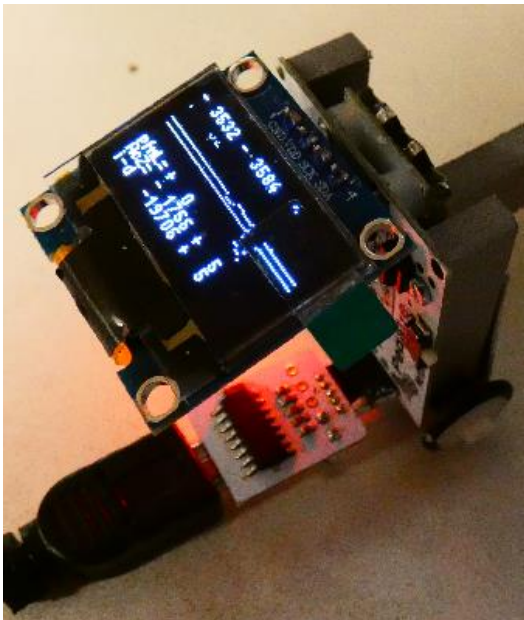
# Mon approche avec Pegase et Witty miniature

Continuer les programmes de test pour bien manipuler et représenter les mesures.

Utiliser un affichage Oled en le programmant pour avoir des temps de réponse acceptables (githug Didel/Oled compatible TerSer)

Développer des outils d'aide à la mise au point temps réel (TerGet pour modifier deux variables et Janus pour remplacer l'oscilloscope).

Tester séparément les réglages P I D, Comprendre comment agir sur chaque paramètres pour améliorer le réglage PID.



**Merci pour votre attention**