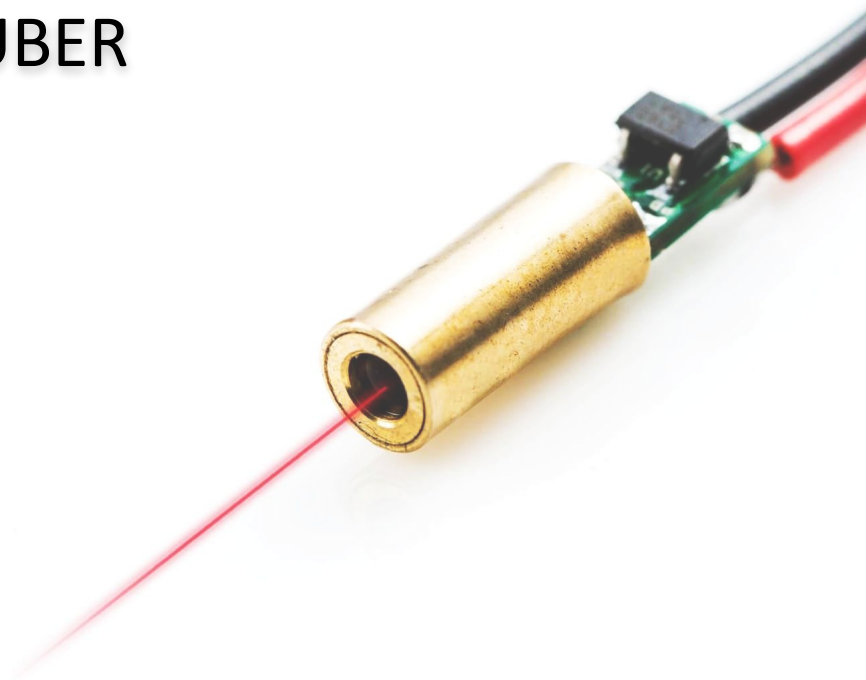
# ProTIS - Asservissement d’un laser en position

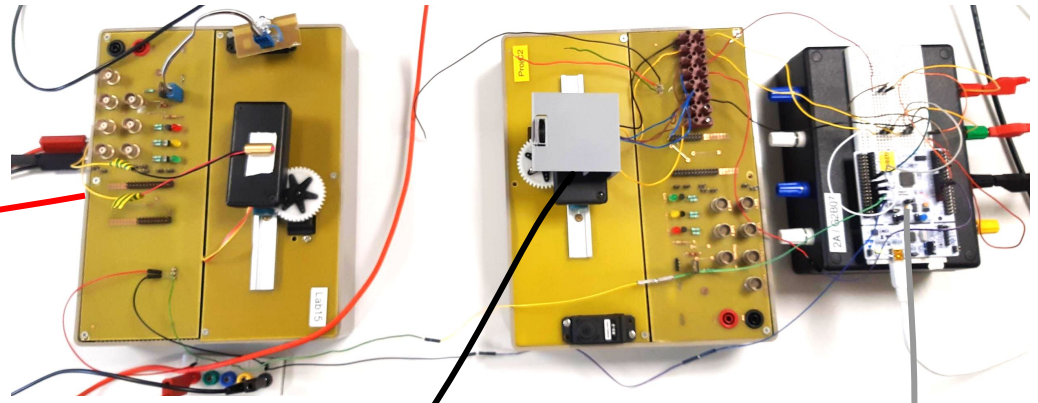
Anne DAUMAS, Olivier MARTINACHE, Nafissa HACHIMI SOBA,
Thomas BESSENAY, Guillaume HUBER



# Sommaire

- Fonctionnement du système et cahier des charges;
- Lecture de la position;
- Commande des moteurs;
- Asservissement PID.

# Présentation du système



Perturbation

*Servomoteur*

Source laser

*Servomoteur*

Barrette CCD

*Nucléo*

- Lire valeur tension des pixels
- Trouver position maximum tache laser
- Calculer l'écart entre position du max et milieu laser
- Commander le servomoteur / Appliquer la consigne

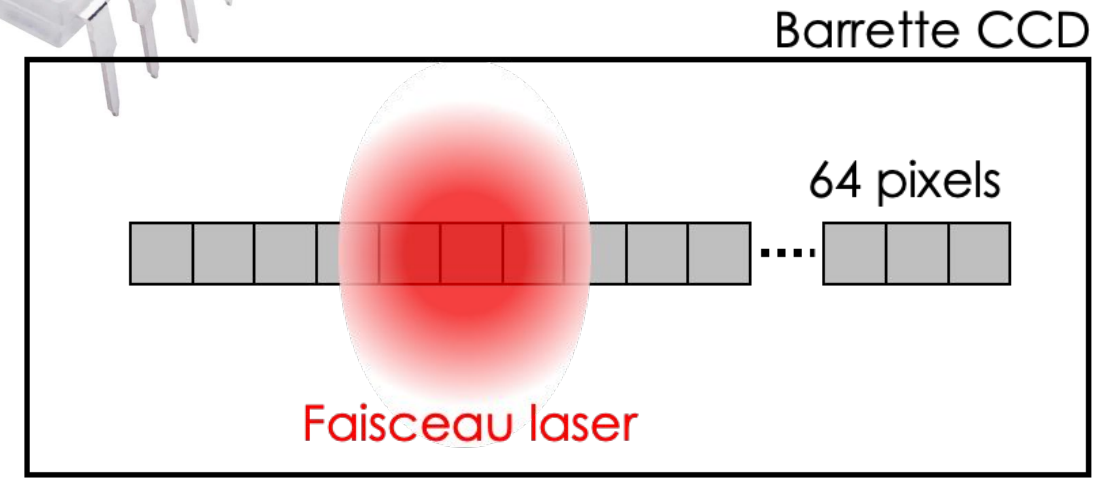
*Nucléo*

Asservissement PID

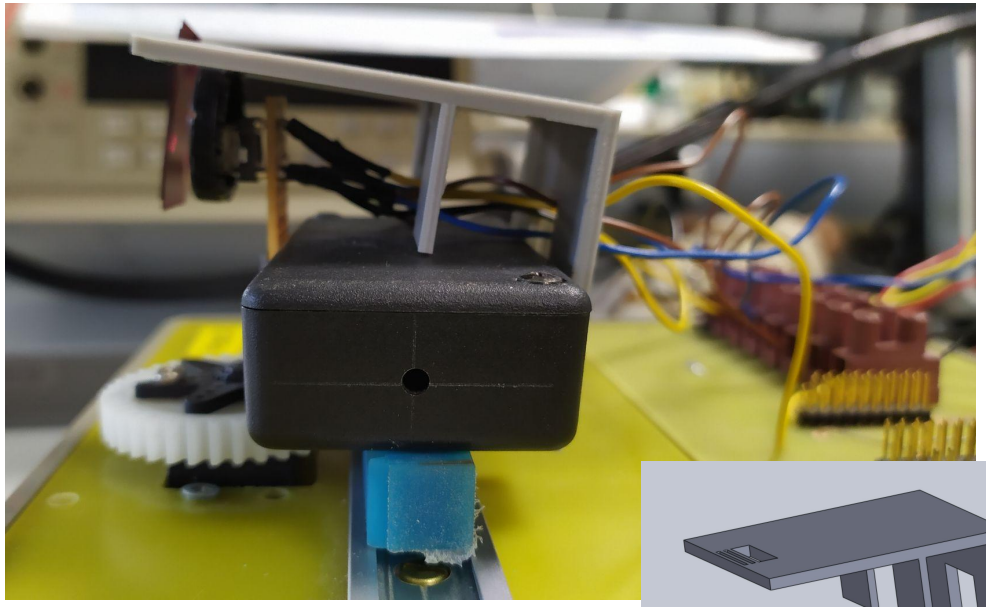
*Matlab*

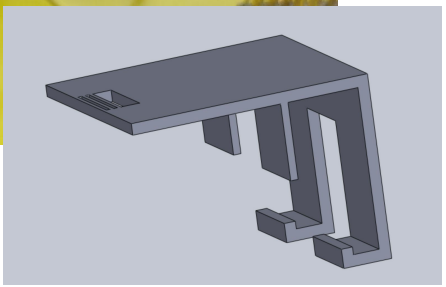
Interface Homme Machine

# Lecture de la position




- pixel = photodiode : éclairement converti en tension
- Saturation des niveaux de tension, due à l'intensité du faisceau + lumière ambiante
- Densité + filtre absorbant laser devant barrette, support par impression 3D
- Taille de la tache laser ~⅓ barrette

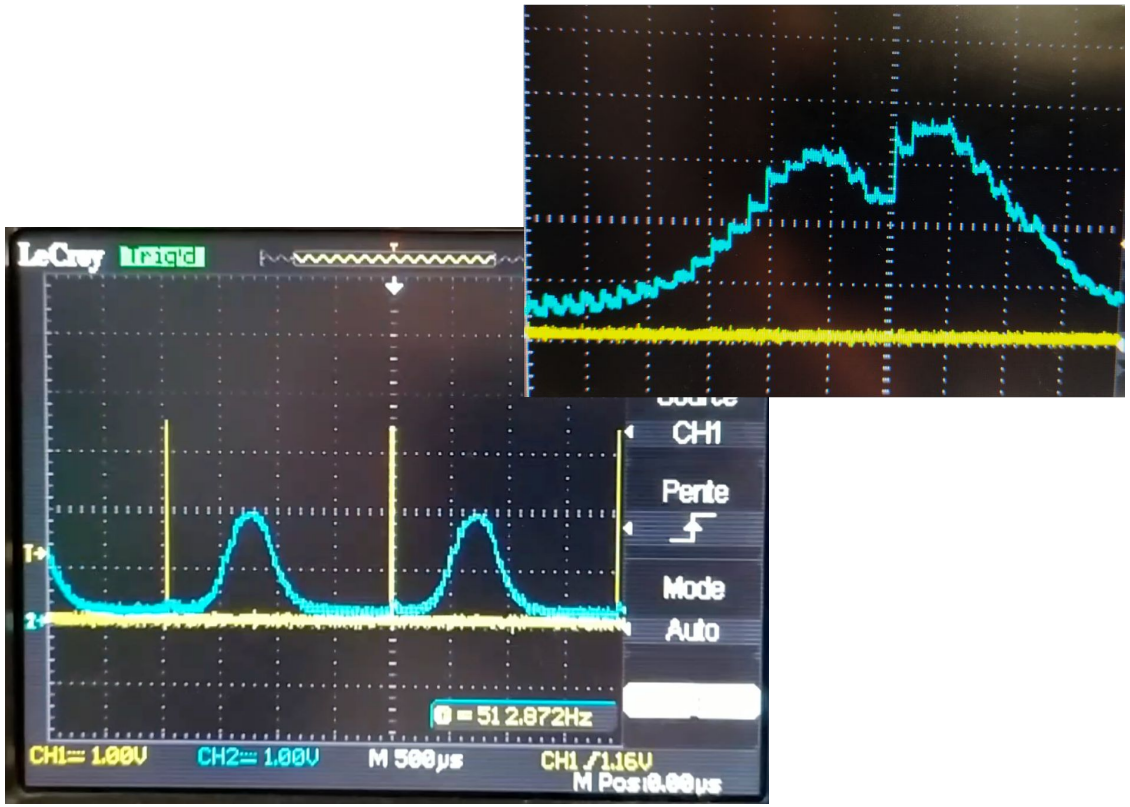




*SolidWorks*

# Lecture de la position

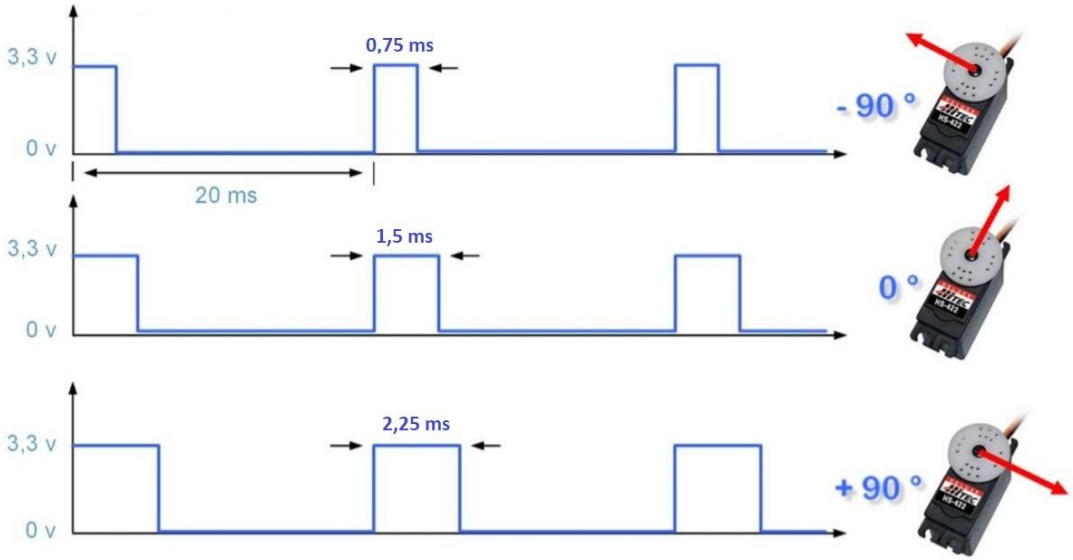
- Génération des signaux Clock et SI
- Récupération de tension_CCD
- Détermination de la position du max
  - Méthode du max direct
  - Détermination du barycentre
  - faire une moyenne glissante de tension_CCD




Commander les servomoteurs...

# Commande des moteurs

- Commande des servomoteurs à partir de la position du laser :




<u>Servomoteur : commande par MLI (Modulation de Largeur d'Impulsion)</u>

➔ Programme position laser : écart k en pixel par rapport au centre (-32,32)

➔ Commande MLI -> commande angulaire

<u>Pignon - crémaillière</u>

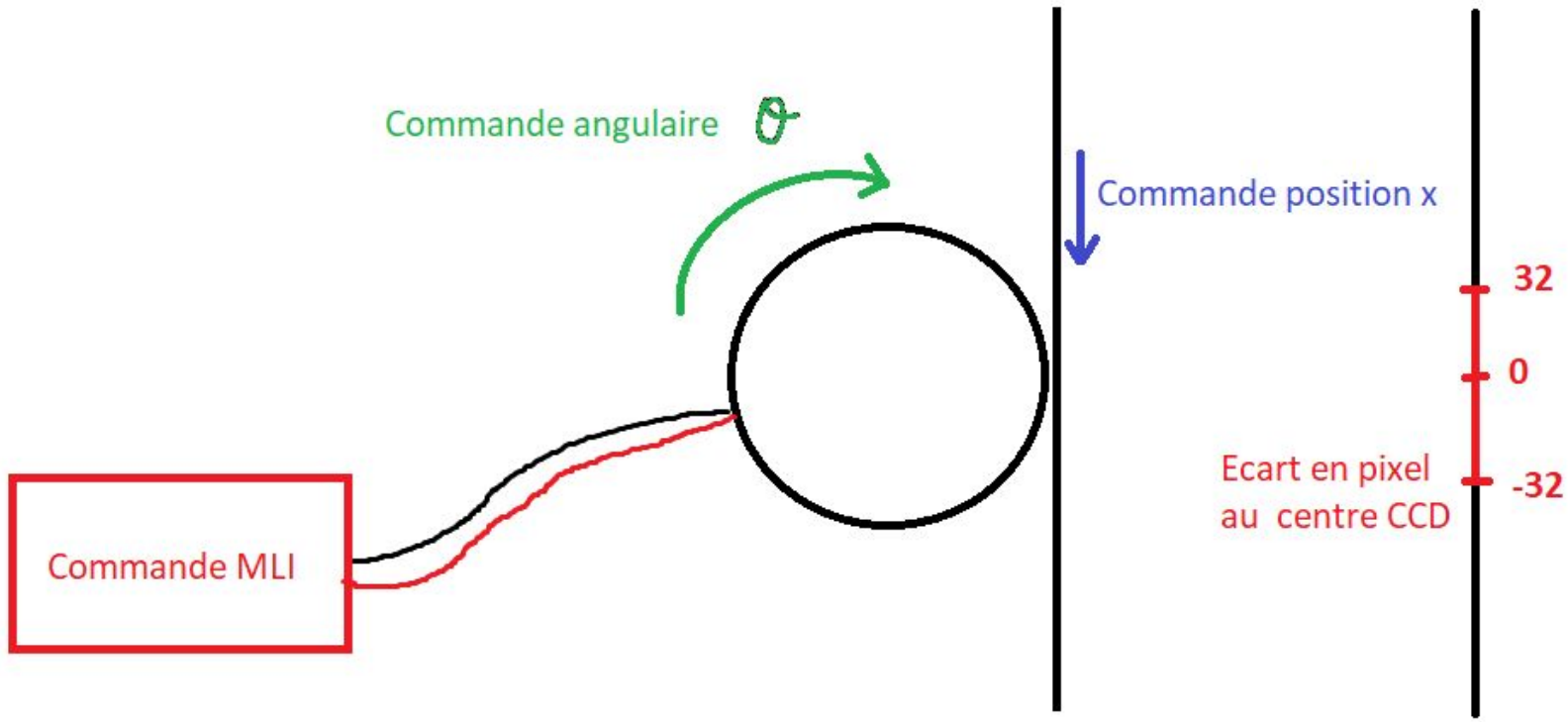
➔ Transformation de mouvement : rotation vers position angulaire

- ◆ $V = \omega.r$ ---> $\mathbf{x = \theta.r + Cte}$ avec r=2cm

➔ Taille de pixel connue

➔ Commande angulaire -> commande en position

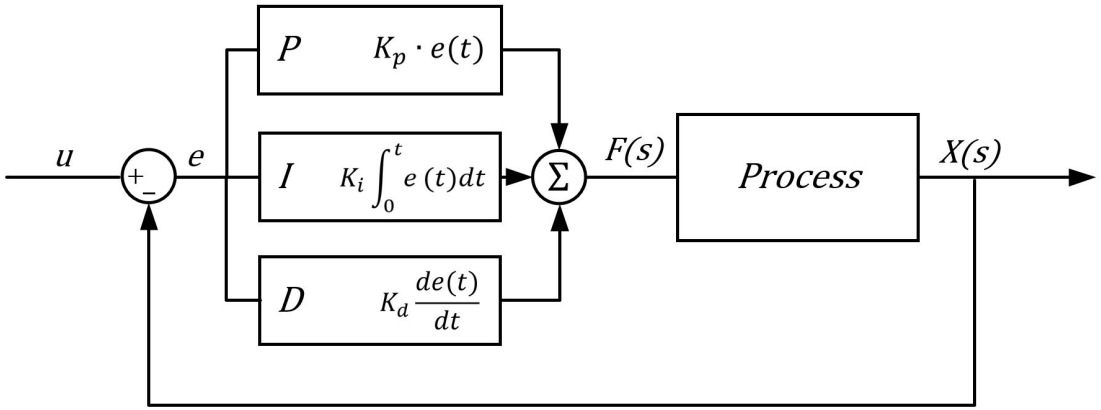
# Commande des moteurs



# Correcteur PID

But du correcteur : performances efficaces de l'asservissement
→ en précision
→ en rapidité
→ en stabilité

Correcteur PID  calcule la commande en fonction de :
→ l'erreur (correction proportionnelle) : **réaction immédiate** à l'erreur
→ l'intégrale de l'erreur (correction intégrale) : **réaction progressive** à l'erreur
→ la dérivée de l'erreur (correction dérivée) : **réaction au changement de l'erreur** pour l'amener à 0




*Programmation du PID*

Utilisation d'une bibliothèque PID déjà existante sur MBED

Prévention du windup pour la correction intégrale

Période de boucle supérieure ou égale à celle de réponse du moteur

# Commande moteur corrigée

```
//Init PID
correcteur.setInputLimits(-32,32); //car on prend comme entrée l'erreur pour la minimiser
correcteur.setOutputLimits(-94,94); //commande MLI du servomoteur pour compenser l'erreur
correcteur.setBias(0); // pas d'offset pour l'instant
correcteur.setSetPoint(0); //consigne : réduire l'écart à 0
```

```
//Init moteur
servo_mot.period_ms(20);          // Initialisation période
servo_mot.pulsewidth_us(positionMLI);    // Initialisation en position 0
wait(1);
servo_mot.pulsewidth_us(positionMLI+100);    // Initialisation en position 0
wait(1);
servo_mot.pulsewidth_us(positionMLI);    // Initialisation en position 0
wait(5);
```

```
//Commande moteur
ecartPixels=32-kmax;
correcteur.setProcessValue(ecartPixels);
ecartMLI=correcteur.compute();
positionMLI=positionMLI-(ecartMLI); //position précédente corrigée par le correcteur
if (positionMLI<=2250||positionMLI>=750){
    int pos = (int) positionMLI;
    servo_mot.pulsewidth_us(pos); }
if (positionMLI>=2250){positionMLI=2250;}
if (positionMLI<=750){positionMLI=750;}



wait(RATE);
```

<u>Initialisation PID</u>

- ➔ Indication des bornes de l'écart (entrée) et de la commande (sortie)
- ➔ Indication de la consigne

<u>Initialisation de la commande moteur</u>

- ➔ Initialisation MLI au centre
- ➔ Vérification que le moteur fonctionne par petit décalage
- ➔ Temps d'attente pour positionner le capteur

<u>Commande du moteur</u>

- ➔ Récupération de l'écart au pixel central
- ➔ Correction PID qui s'incrémente à la commande en position du moteur
- ➔ Limitations pour récupérer la commande moteur si le système s'emballe

# Synthèse

- Lecture de la position ;
- Commande des moteurs ;
- Asservissement PI(D?).