



ANTOINE - JOCELYN - LUCAS

SAÉ

PROJET ROBOT

OBJECTIFS

- ASSERVISSEMENT EN VITESSE (LINÉAIRE & ANGULAIRE)
- ASSERVISSEMENT EN POSITION

ODOMÉTRIE

Mesurer précisément le déplacement du robot

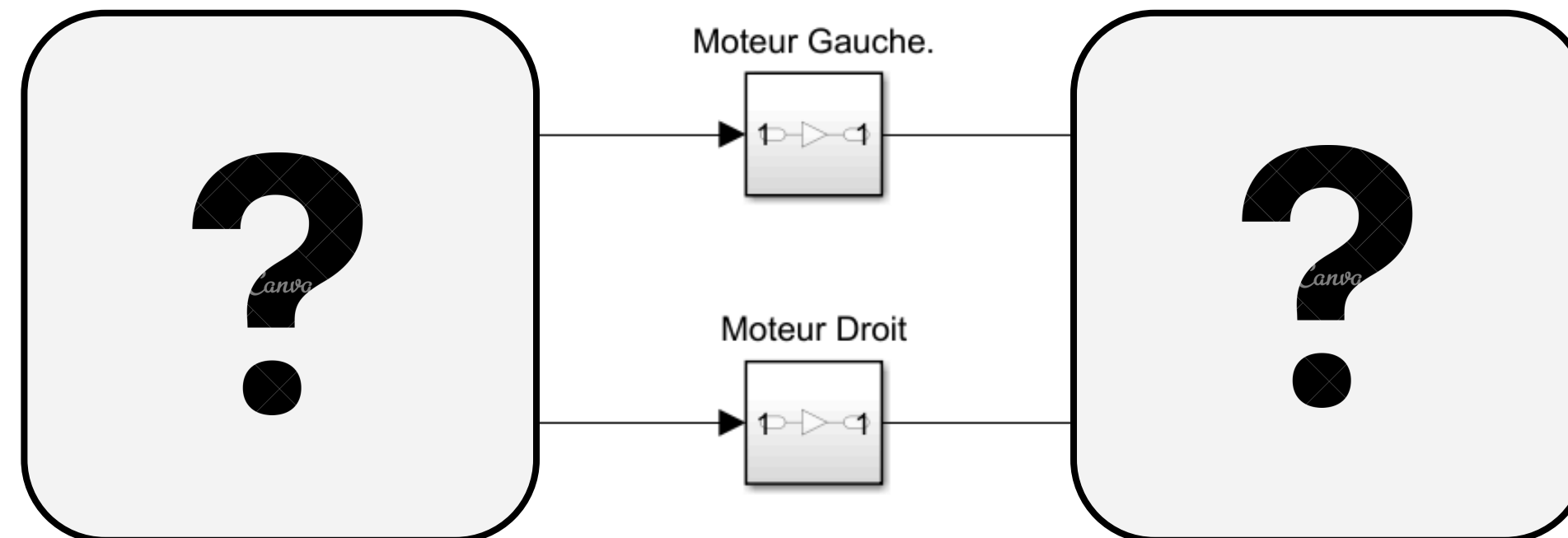
- Modules QEI & encodeurs
- Permettent de compter le mouvement des crans d'une roue
- Unité de mesures en mètres

ASSERVISSEMENT EN VITESSE

2 CONFIGURATIONS POSSIBLES:

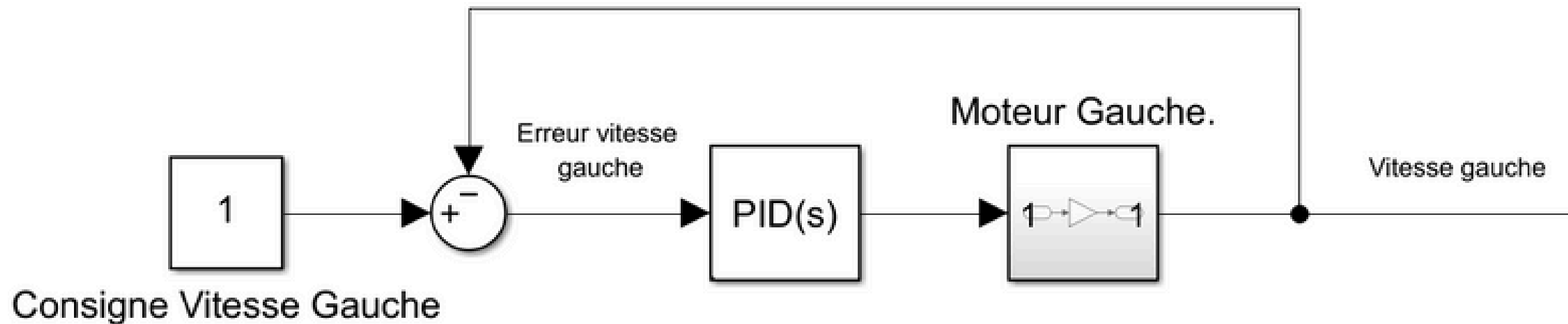
Asservissement indépendant

Asservissement polaire



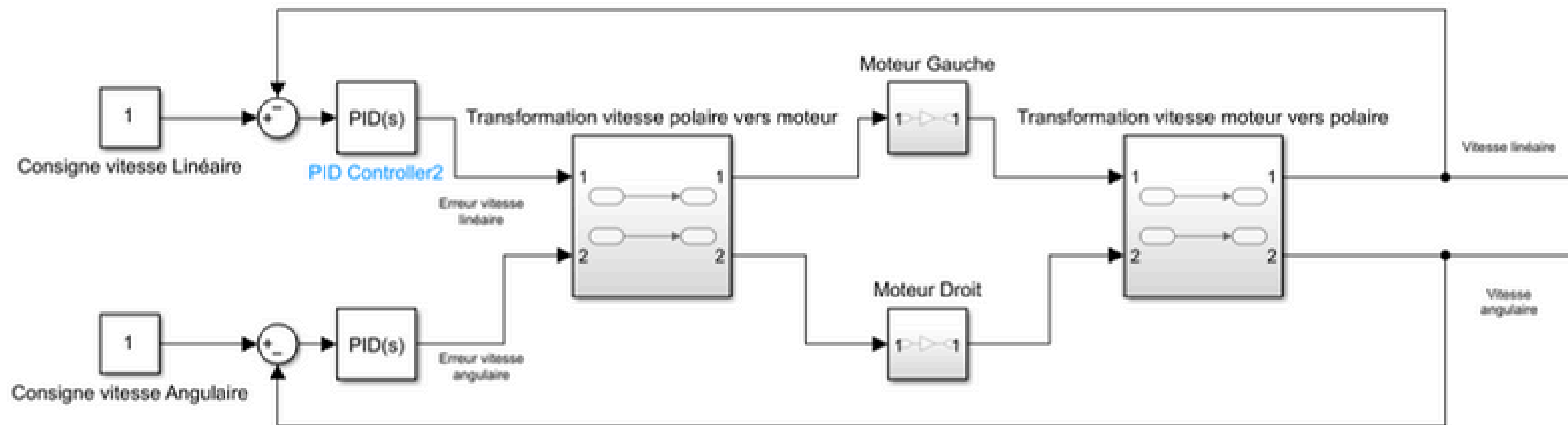
ASSERVISSEMENT INDÉPENDANT

Contrôle des deux moteurs séparément



ASSERVISSEMENT POLAIRE

Contrôle combiné des deux moteurs



ASSERVISSEMENT POLAIRE

► Saturation & anti-windup

Éviter dommager les moteurs en cas de blocage

► Transformations des vitesses moteurs indépendants ↔ polaires

Permettre la commande des moteurs

$$\begin{pmatrix} V_{Lineaire} \\ V_{Angulaire} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2R_{Robot}} & -\frac{1}{2R_{Robot}} \end{pmatrix} \begin{pmatrix} V_{Droit} \\ V_{Gauche} \end{pmatrix}$$

$$\begin{pmatrix} V_{Droit} \\ V_{Gauche} \end{pmatrix} = \begin{pmatrix} 1 & R_{Robot} \\ 1 & -R_{Robot} \end{pmatrix} \begin{pmatrix} V_{Lineaire} \\ V_{Angulaire} \end{pmatrix}$$

KEYLOGGER

- **Piloter le robot depuis un clavier**
- **Forcer le déplacement**
- **Vérifier la trajectoire**
- **Régler le coefficient**

COEF_SPEED_TO_PERCENT



Mode

Automatique

Asserv

☒ Mode Manuel

Valeurs OLI

Position X
0,410 m

Timer
18,8 s

Vitesse Linéaire
1,014 m/s

Position Y
3,957 m

Angle
-1,73 rad

Vitesse Angulaire
-0,73 rad/s

Consignes

Vitesse Linéaire :

0,00

Vitesse Angulaire :

0,00

Envoi Consignes

Envoi PD

Envoi PD Pos

	X	θ	M1	M2
Consigne	0,00	0,00	0,00	0,00
Mesure	1,01	-0,73	0,00	0,00
Erreur	0,00	0,00	0,00	0,00
Commande	0,00	0,00	0,00	0,00
KP	0,00	0,00	0,00	0,00
Correction P	0,00	0,00	0	0
Correction P Max	0,00	0,00	0,00	0,00
KI	0,00	0,00	0,00	0,00
Correction I	0,00	0,00	0	0
Correction I Max	0,00	0,00	0,00	0,00
KD	0,00	0,00	0,00	0,00
Correction D	0,00	0,00	0	0
Correction D Max	0,00	0,00	0,00	0,00



IMPLEMENTATION DES COEFFICIENTS

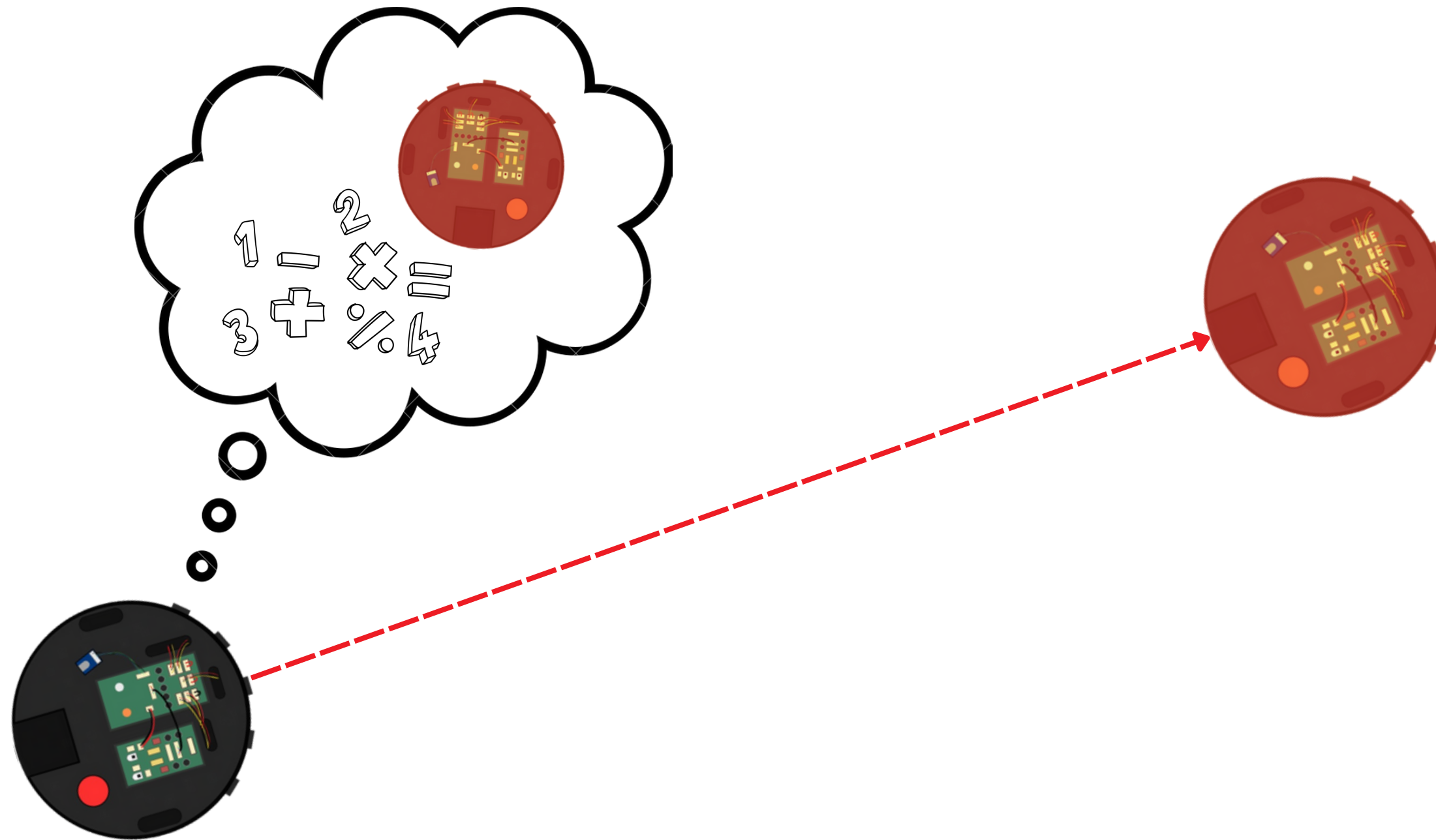
RÉGLAGES

- Consigne à 0
- PID vitesse linéaire
- PID vitesse angulaire

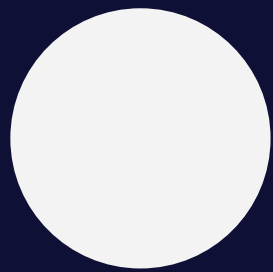
RÉSULTAT : ROBOT ASSERVI !!



ASSERVISSEMENT EN POSITION



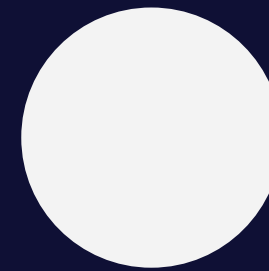
GÉNÉRATEUR DE TRAJECTOIRE : V1



Structure en switch case

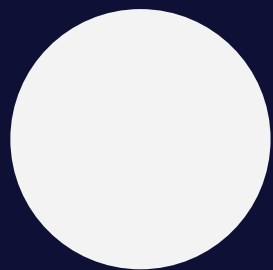
Attend un point puis :

- Tourne
- Avance
- Attend



Problèmes

- Erreur d'angle sur le ghost
- Plusieurs fonctions



Calcul des paramètres

- Calcul des distances d'accélération et de décélération
- Vitesse maximale adaptée en fonction de la distance
- Calcul de la position courante

GÉNÉRATEUR DE TRAJECTOIRE : V2

Points positifs

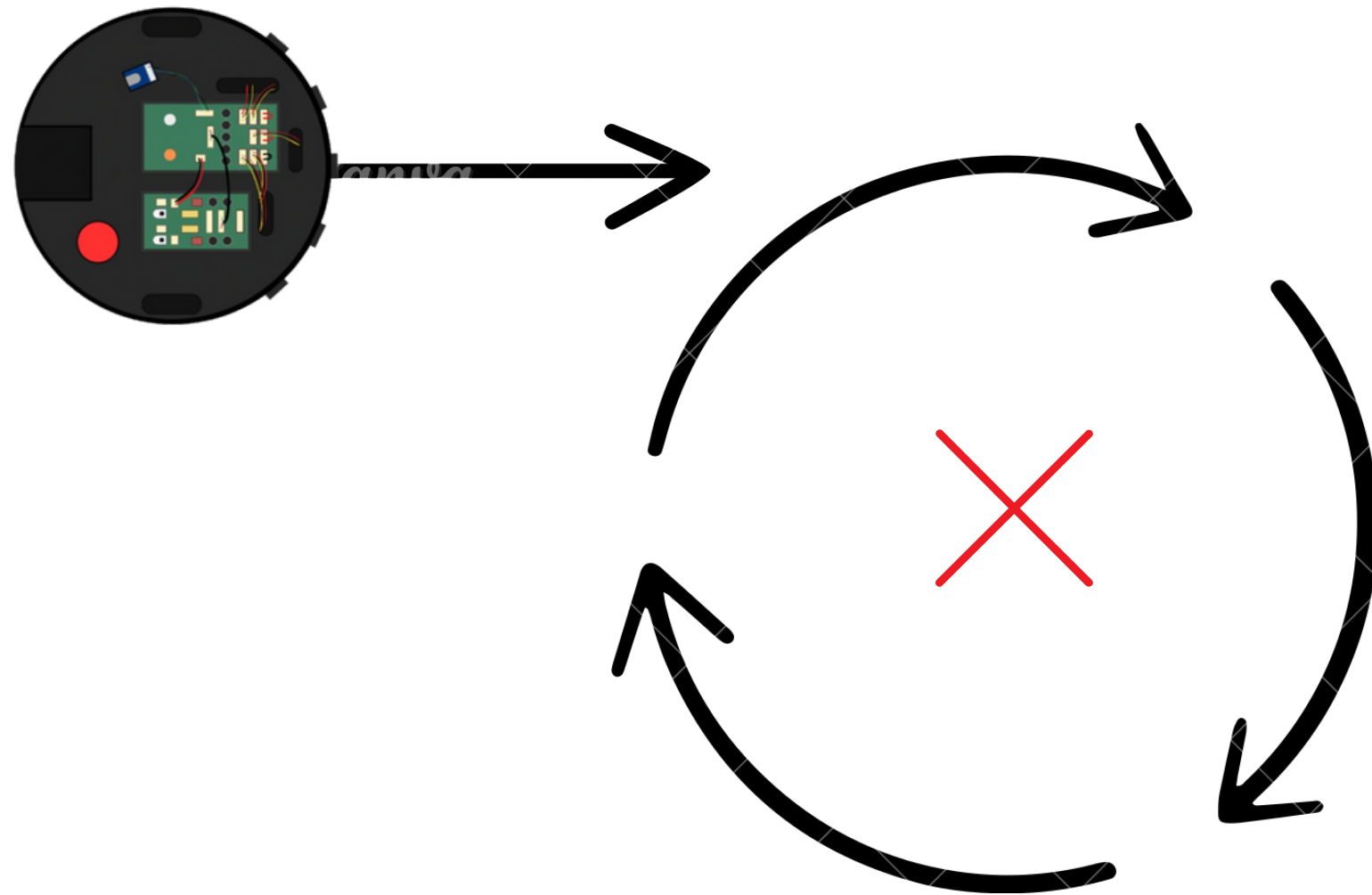
- Se rend parfaitement à la direction
- Vitesse correcte
- Erreur minime sur le ghost

Problèmes

- Vitesse linéaire n'atteignant jamais zéro
- Se met à tourner sur lui même autour du point

```
//Si on a un angle
{
    //Soit l'angle à parcourir est positif
    {
        //Soit l'angle à parcourir est supérieur à la distance
        d'arrêt
        {
            //On accélère en saturant à Vmax
        }
        //Soit l'angle à parcourir est inférieur à la distance
        d'arrêt
        {
            // On freine
        }
    }
}
```

CAS PROBLÉMATIQUE



2 SOLUTIONS :



Limiter à une séquence

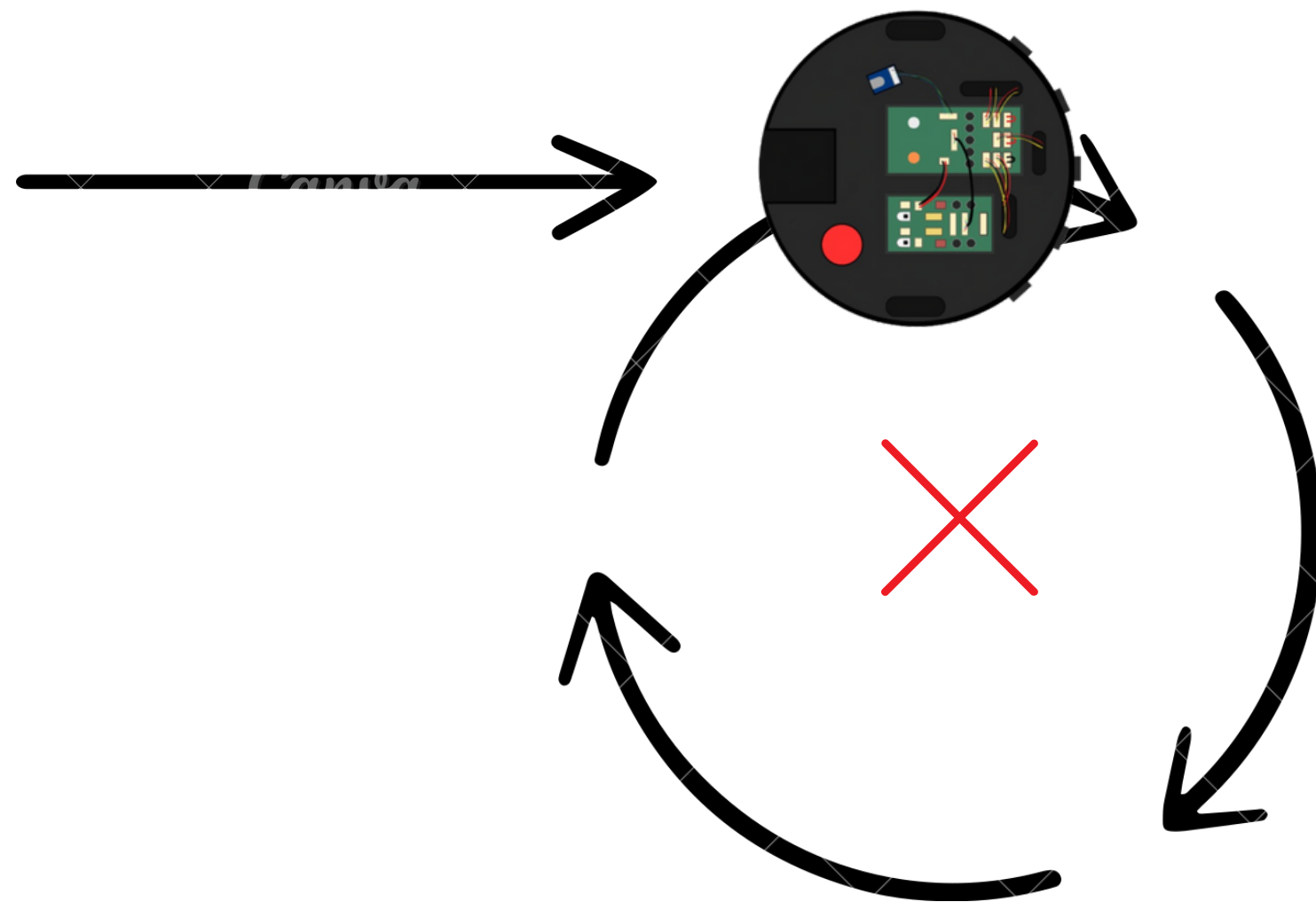
- Effectuer la séquence uniquement si le robot reçoit une nouvelle position



Forcer la vitesse linéaire à 0

- Vitesse à $t+1$
- Forcer à 0 sous une certaine valeur

CAS PROBLÉMATIQUE



2 SOLUTIONS :



Limiter à une séquence

- Effectuer la séquence uniquement si le robot reçoit une nouvelle position



Forcer la vitesse linéaire à 0

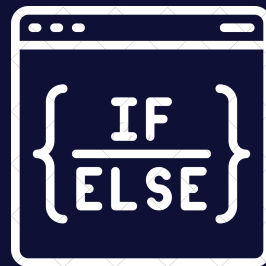
- Vitesse à $t+1$
- Forcer à 0 sous une certaine valeur



Ne fonctionne pas dans la "vraie vie"

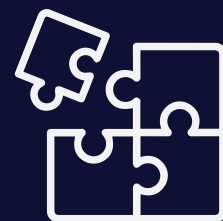
- Ne supporte pas un changement de waypoint en cours de déplacement

V3



Suppression du "else"

- Avec deux "if" le robot peut tourner et avancer en même temps
- Permet de constamment être en direction de la cible



Paramètre interdépendants

- Les mouvements combinés demandent des conditions plus complexes
- Le robot tourne si la distance respecte certaines conditions et inversement



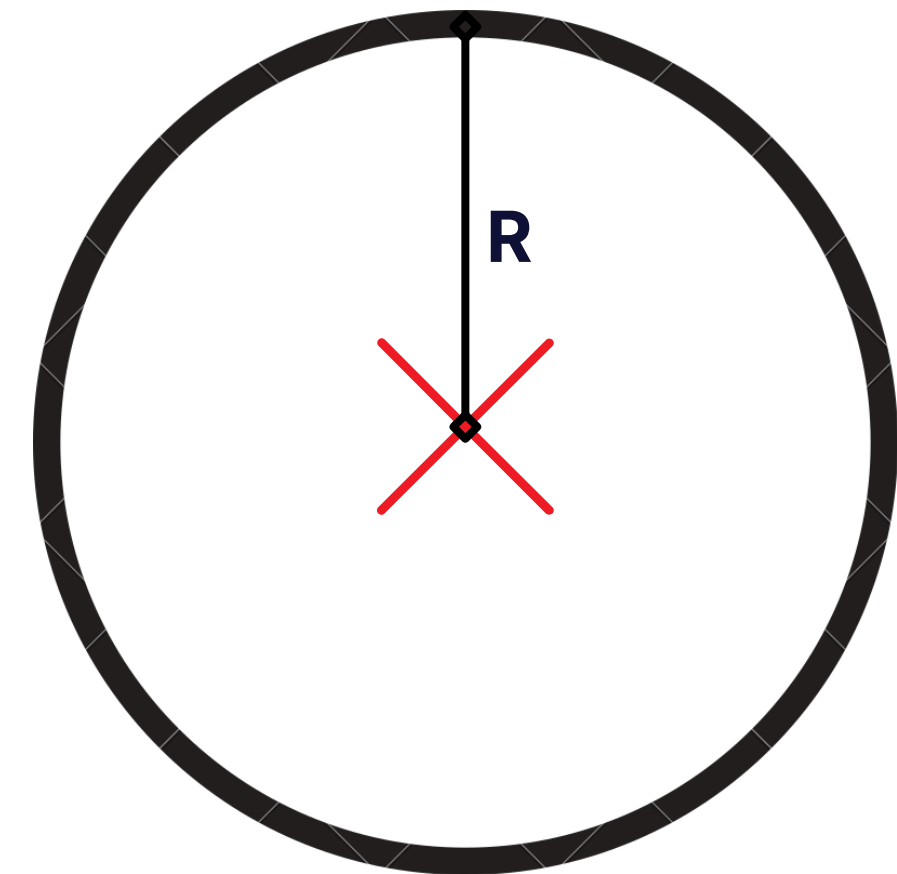
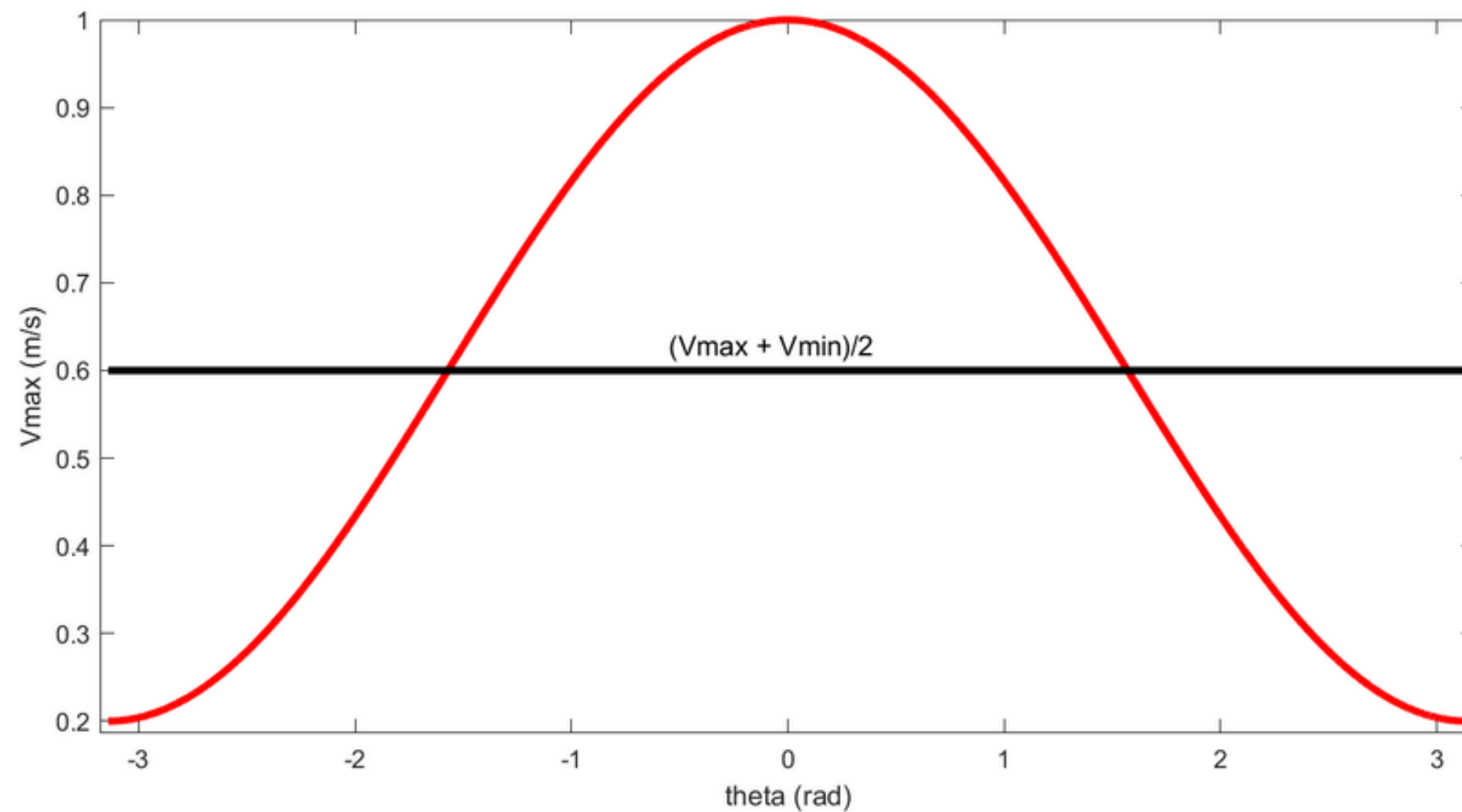
Conditions dynamiques

- Vmax dynamique pour empêcher le robot de drifter
- Détermination d'un rayon d'arrêt

VITESSE MAXIMALE & RAYON D'ARRÊT

$$V_{lin_{max}} = \frac{V_{max_{max}} + V_{max_{min}}}{2} + \frac{V_{max_{max}} - V_{max_{min}}}{2} * \cos \theta$$

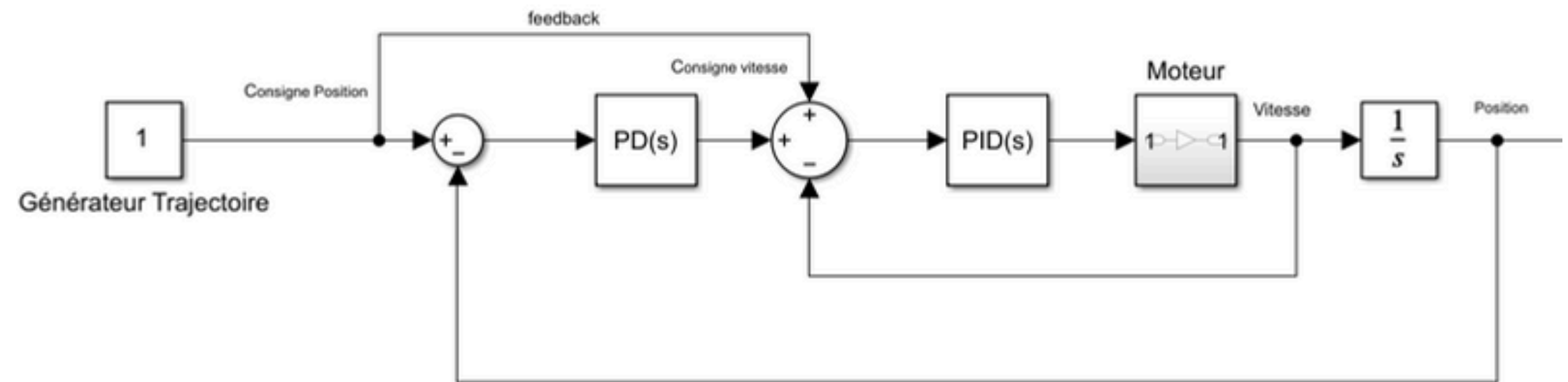
$$R = \frac{V_{lin_{max}} \left(\frac{\pi}{2} \right)}{\Omega_{max}}$$



AJOUT FUTUR : CORRECTEUR PD

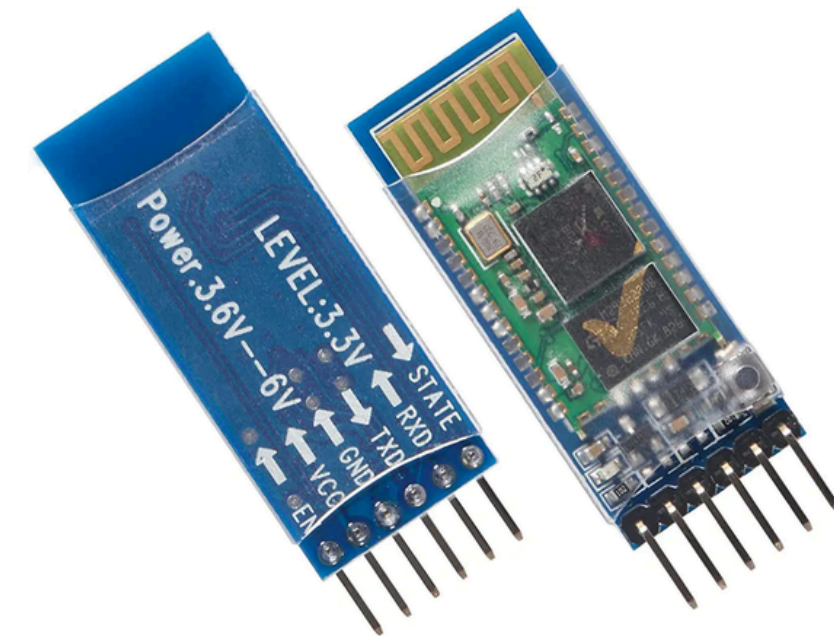
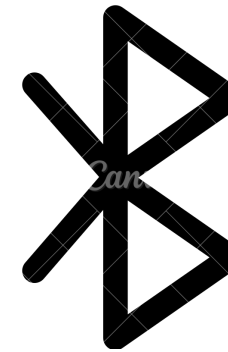
Asservissement en position

- Permet de corriger en direct les erreurs
- Evite les gros écarts d'angles remarqués en pratique



MODULE BLUETOOTH HC-05

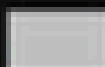
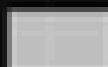
PILOTER & SUPERVISER À DISTANCE



Waypoint

Cible X

Cible Y



ENVOI WAYPOINT

AJOUTER AU PARCOURS

Centre

Zone Haut Gauche

Balise Haut Gauche

Zone Milieu Gauche

Balise Bas Gauche

Zone Bas Gauche

Balise Haut Droit

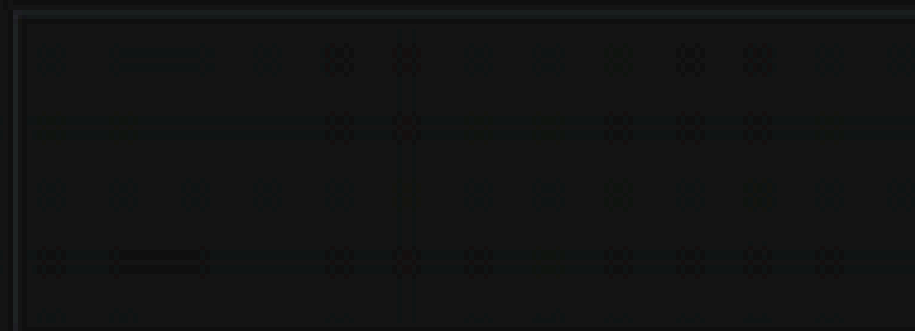
Zone Haut Droit

Balise Bas Droit

Zone Milieu Droit

Zone Bas Droit

Parcours

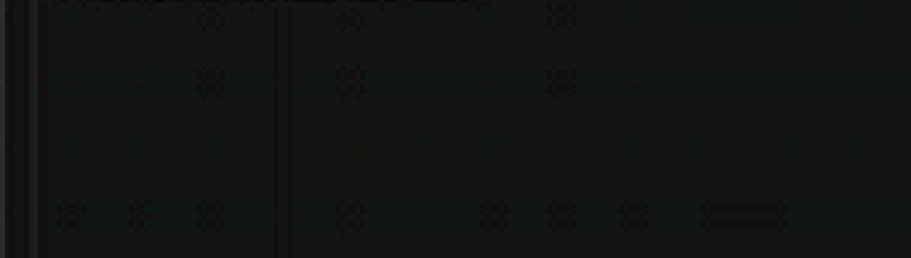


EXÉCUTER

VIDER

Feedback

→ ATTENTE DE LA PROCHAINE CIBLE



Distance à la cible : 0,00 m

Ghost X : 0,00

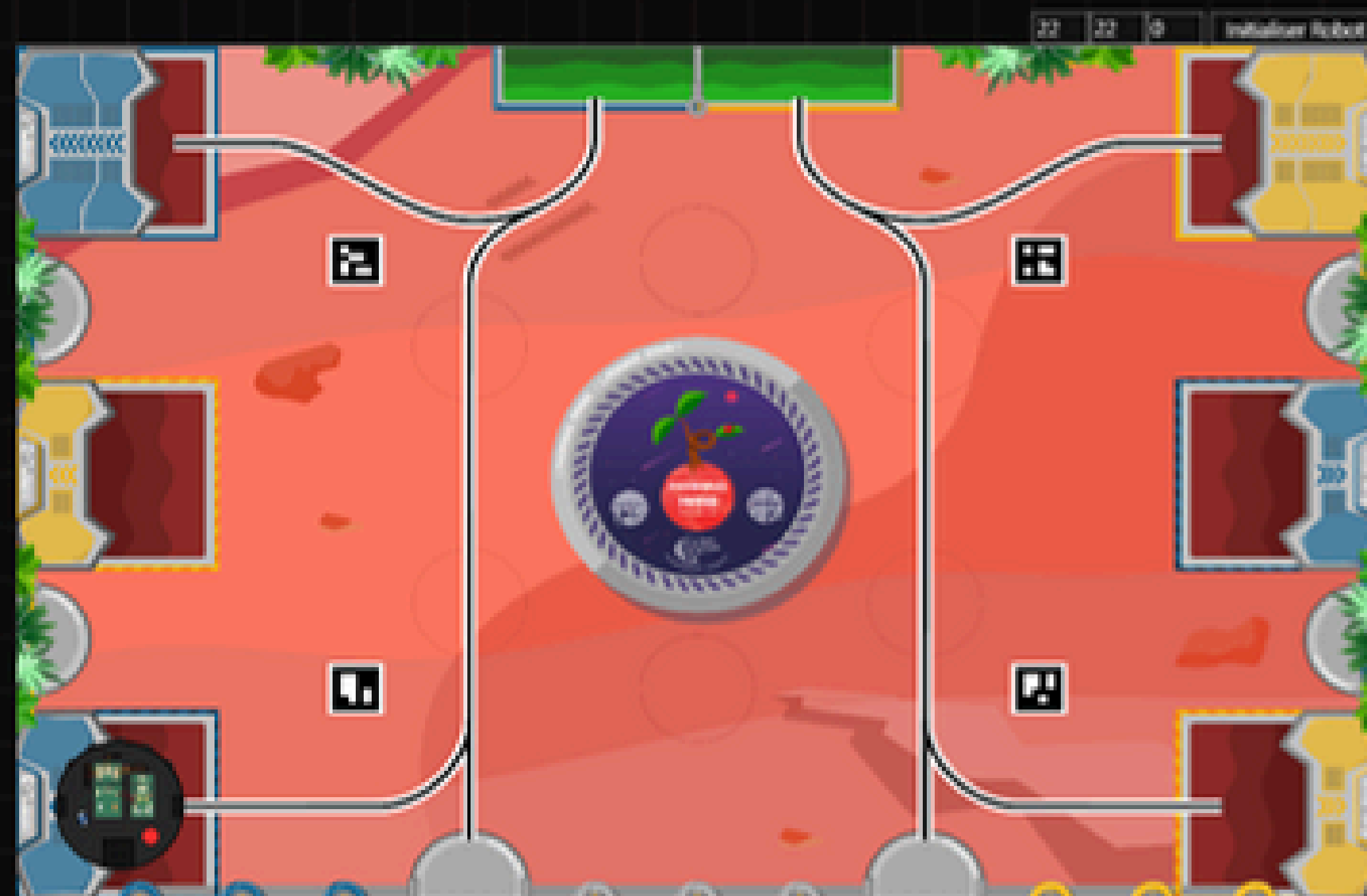
Cible X : 0,00

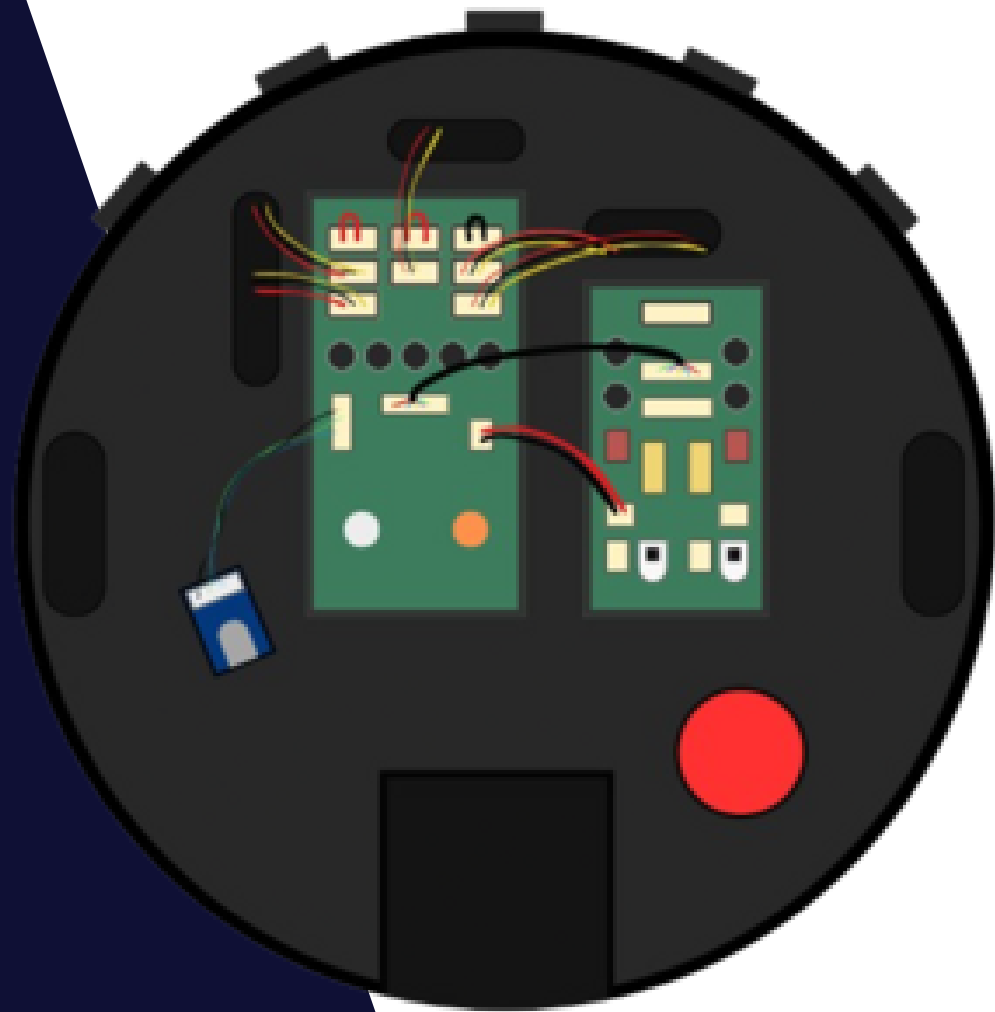
Angle à la cible : 0,00 rad

Ghost Y : 0,00

Cible Y : 0,00

Position





MERCI !