

Planification multitâche Robots LEGO Mindstorm

Thao Dang

Implantation multi-tâche

Pour illustrer l'implantation multi-tâche, on va créer une application de commande de robots LEGO

Tâche 1 - Planification de trajectoires (définir des consignes en temps-réel)

Tâche 2 - Régulation de trajectoires (corriger la trajectoire en fonction de consignes données par Tâche 1)

Tâche 1 - Planification

2 sous-tâches

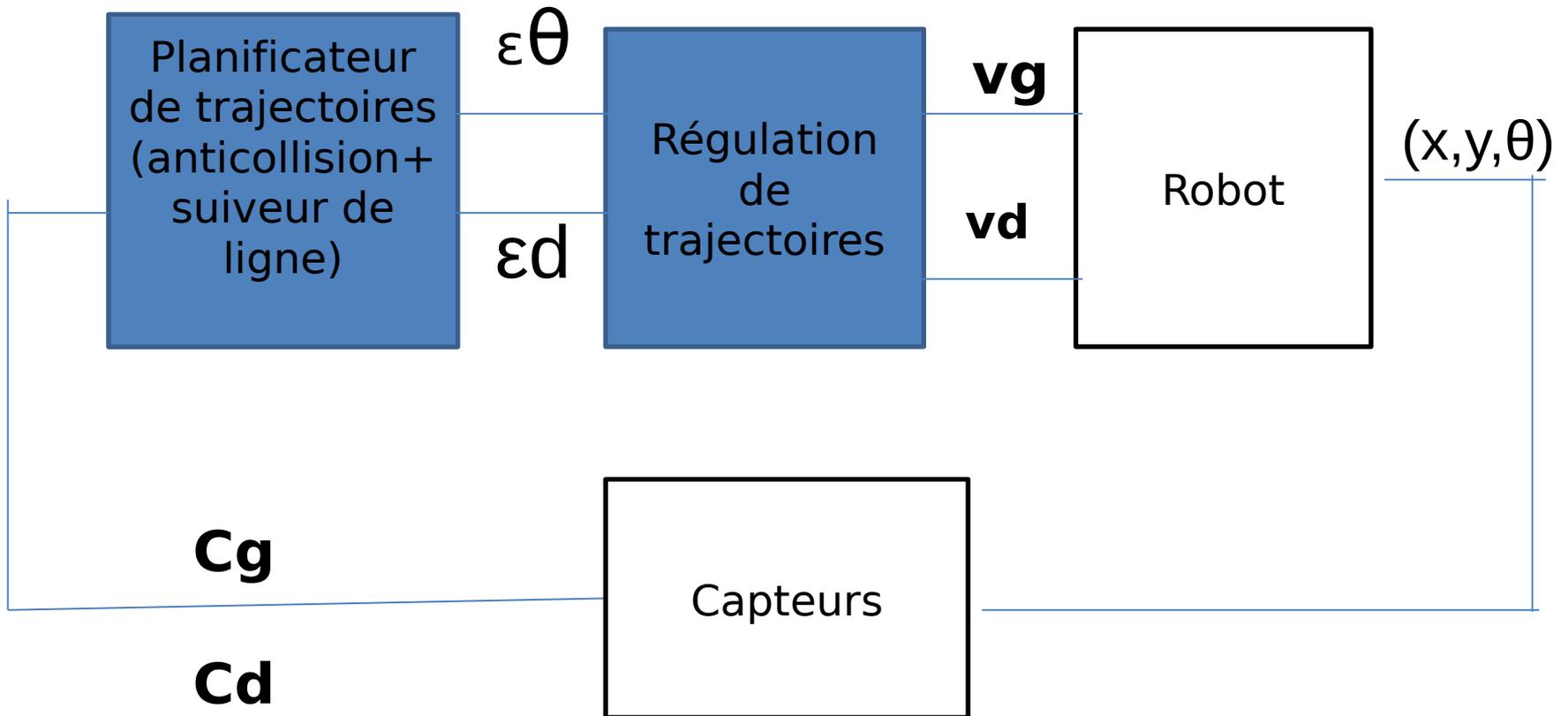
Anti-collision - Quand le robot détecte un obstacle, cette sous-tâche génère une consigne pour changer la direction et/ou la vitesse (eg., arrêt, marche arrière, avancement en vitesse réduite).

Suiveur de ligne - A partir des données de capteurs cette sous-tâche détermine la déviation par rapport a la ligne, cad les valeurs des consignes ε_d et ε_θ , pour corriger la trajectoire du robot

Tâche 2 - Régulation de trajectoires

- Cette tâche assure que le robot suit les consignes définies par le planificateur
- Elle envoie périodiquement des valeurs de vitesse de deux roues aux moteurs.
- Ces valeurs sont calculées par deux contrôleurs **PID** en fonction des informations sur la position (x,y) et l'orientation θ

Schéma de Commande



Projet Robot LEGO Mindstorm

1. Modélisation réaliste

- Analyse de la **robustesse** de contrôleurs sous différentes sources de **perturbations** (discrétisation, erreurs numériques, délais et imprécisions des capteurs) (**à faire**)
- On utilise les **méthodes polynomiales** (stabilité de polynômes avec paramètres incertains)

2. Implantation Multi-tâche

- En **Simulink**, concevoir 2 modèles correspondant a 2 tâches:

TP

Créer un modèle SIMULINK de **planificateur**

- Le planificateur prend comme entrées
 - Données fournies par 2 capteurs de lumières (pour déduire la déviation par rapport a la ligne noire).
 - Données fournies par 1 capteur de distance (pour détecter un obstacle a proximité)
- Le planificateur retourne comme sorties
 - Les valeurs de consignes $\varepsilon\theta$ et εd

Anti-collision

Si un obstacle est détecté, *une* stratégie possible est:

- D'abord, le robot va s'arrêter (le robot prend un certain temps pour s'arrêter complètement)
- Ensuite, le robot va faire un demi-tour sur place.