

Développement par modèles de systèmes de commandes embarqués (I)

Apprentissage au travers d'une étude de cas : robot LEGO Mindstorm NXT

Les étapes du développement :

- **Modélisation** : Créer un modèle mathématique décrivant l'évolution des trajectoires du robot
- **Conception de contrôleurs** : construire un contrôleur qui
 - satisfait des spécifications désirées, telles qu'atteindre un point de destination, suivre une ligne, éviter la collision avec des obstacles, réaliser des tâches plus complexes
 - prend en compte de contraintes physiques des capteurs et des actionneurs du robot (et éventuellement des critères d'optimalité tels que la minimisation d'énergie)
- **Validation par simulation** :
 - Écrire un modèle Simulink (comprenant le système et le contrôleur construit)
 - Simuler ce modèle Simulink (pour savoir si le contrôleur en effet permet de garantir les spécifications désirées).

Développement par modèles de systèmes de commandes embarqués (II)

Les étapes du développement (suite) :

- **Implantation du contrôleur** : Il faut générer un programme dans le langage accepté par l'environnement de programmation des robots LEGO NXT, tel que NXC (Not Exactly C). Pour générer un tel programme :
 1. Générer un programme en Lustre à partir du modèle en Simulink du contrôleur (il existe un outil pour faire ceci)
 2. Générer ensuite le code NXC (Not Exactly C) à partir du programme en Lustre (en adaptant le compilateur Lustre existant).
- **Expérimentation** : charger le programme dans le robot et le tester

Réalisation du projet

Connaissances nécessaires pour réussir le projet :

- Automatique : modélisation par des équations différentielles, commande de systèmes linéaires, contrôleurs PID, simulation en utilisant Matlab/Simulink
- Informatique : langage Lustre, langage NXT, génération de code, aspects temps-réel (ces sujets seront présentés au travers des petits cours)

Robot Mindstorms NXT



1. **NXT** : brick de contrôle intelligent, le cerveau du robot Mindstorms
 2. **'Touch Sensor'** : permet au robot de sentir l'environnement
 3. **'Sound Sensor'** : permet au robot d'entendre
 4. **'Light Sensor'** : permet au robot de détecter la lumière
 5. **'Ultrasonic Sensor'** : permet au robot de voir, de mesurer la distance à un obstacle
 6. **'Servo Motor'** : garantit des mouvements précis
-