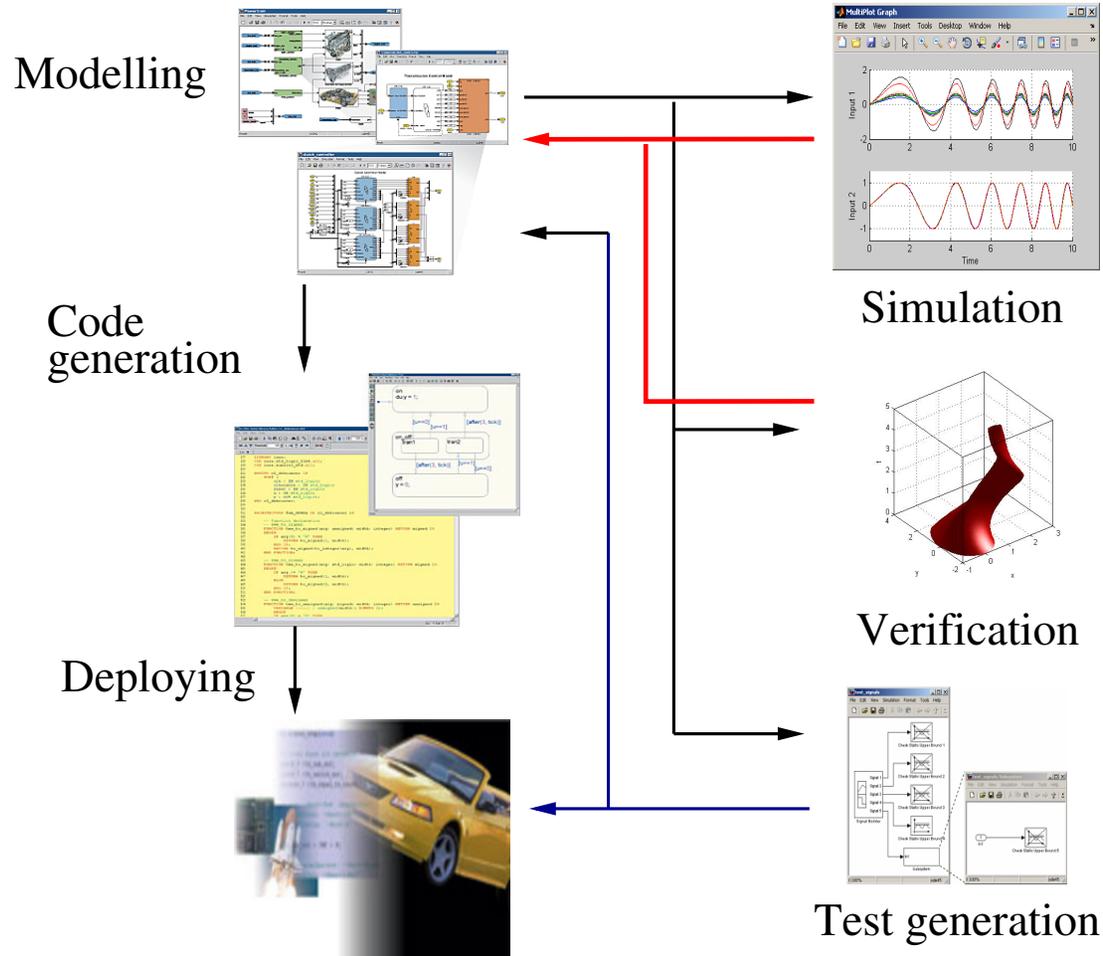




# Développement par modèles de systèmes

---



SCADE, Matlab/Simulink, Prover Technology, PolySpace, Wind River, TTTECH, etc.

**Détection and correction d'erreurs plus facile**

---

# Objectif du cours

---

- Implantation sûre de systèmes de commandes embarqués
- Outils pour l'implantation sûre

# Développement par modèles (I)

---

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

Les étapes du développement :

- **1. Modélisation** : Créer un modèle mathématique décrivant l'évolution des trajectoires du robot
  - **2. 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)
  - **3. 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) :

- **4. 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 (en utilisant l'outil **sim2lus**)
  2. Générer ensuite le code **NXC** (Not Exactly C) à partir du programme en Lustre (en adaptant le compilateur Lustre existant).
- **5. Expérimentation/Test** : charger le programme dans le robot et le tester

**Étapes 4 et 5 sont le sujet principal de ce 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
-

# Plan du cours

---

- Semaines 1, 2, 3 Commande de robot

Modélisation et commande de robot suiveur de ligne, planification de trajectoires

**Objectif** : fin de la Semaine 3, modèle SIMULINK de contrôleur conçu et testé

- Semaines 4, 5 – De Simulink vers Lustre

Implantation sûre, introduction au langage LUSTRE, traduction d'un modèle Simulink vers Lustre

**Objectif** : Apprentissage au travers des exemples.

- Semaine 6 – De Lustre vers NXC

Génération du code NXC pour LEGO à partir d'un programme Lustre, Environnement de programmation Bricxcc

**Objectif** : Apprentissage au travers des exemples.

- Semaines 7, 8 – Application au robot

**Objectif final** : Obtention du code à faire exécuter par le robot

- Semaines 9, 10, 11 – Expérimentation

Réglage du contrôleur, débogage en remontant la chaîne d'outils

**Objectif final** : Le robot marche comme spécifié

---

# EVALUATION

---

Le projet développé dans la partie pratique sera présenté lors d'un oral d'environ 15mn.

L'assiduité au cours est prise en compte.

---

# Page d'internet du cours

---

<http://www-verimag.imag.fr/~tdang/CoursISC-2010.html>