Intro à la robotique

Architecture minimale d'un robot autonome : perception $\rightarrow$ décision $\rightarrow$ action

---

DIROBOTS — Université de Montréal

1) Robotique : c'est quoi, concrètement ?

Robotique : discipline qui construit des **systèmes physiques** capables d'interagir avec le monde réel.

Un robot n'est pas « juste du code » ni « juste du matériel » :

- **Capteurs** : observer (vision, distances, orientation, etc.).
- **Calcul / logiciel** : interpréter, décider, planifier.
- **Actionneurs** : bouger, manipuler, appliquer une force.

Idée clé : la robotique est un problème d'**intégration** (hardware + software + contraintes physiques). Une grande partie des difficultés vient du fait que tout doit fonctionner **ensemble**, en conditions réelles.

2) Robots et usages : pourquoi c'est un domaine « partout »

Exemples de robots :

- drones et robots sous-marins autonomes ;
- plateformes mobiles (entrepôt, livraison), robotaxis ;
- quadrupèdes, humanoïdes, exosquelettes ;
- manipulateurs industriels, robots de service, robots médicaux.

Domaines :

- industrie, santé, agriculture ;
- logistique, transport, exploration ;
- maison, éducation, sécurité publique, divertissement.

Point commun : données imparfaites, incertitude, temps réel, contraintes d'énergie et de sécurité.

3) Un robot = un système complet (pas une pièce)

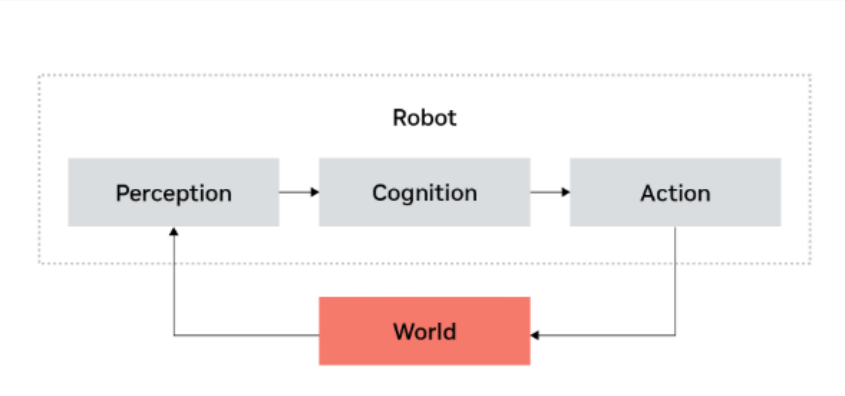
Vue système : un robot fonctionne comme une boucle fermée avec le monde.

- Le monde produit des signaux (images, contacts, obstacles, mouvements).
- Les capteurs transforment ces signaux en données.
- Le logiciel produit une décision (où aller, quoi faire, comment).
- Les actionneurs exécutent, et le monde change en retour.

Pourquoi c'est difficile : latence, bruit, calibrations, limites physiques, synchronisation, communication.

4) Architecture de base : Embodied AI (perception → cognition → action)

Schéma de référence : boucle perception–décision–action.



Lecture simple :

- **Perception :** « Qu'est-ce qui se passe autour de moi ? »
- **Cognition / décision :** « Que dois-je faire maintenant ? »
- **Action :** « Je bouge / j'agis », ce qui modifie le monde.

Analogie : yeux $\rightarrow$ cerveau $\rightarrow$ muscles.

5) Matériel : capteurs, actionneurs, contrôleurs

Capteurs (perception) : transforment le monde en mesures.

- **Caméras :** information visuelle (objets, repères).
- **LiDAR :** distances / géométrie (obstacles, carte 3D).
- **IMU :** orientation, accélérations (stabilité, navigation).
- **GPS :** position globale (surtout en extérieur).

Actionneurs (action) : appliquent un mouvement ou une force.

- moteurs (roues, articulations), servos (position précise),
- pneumatique / hydraulique (force, selon applications).

Contrôleurs (calcul embarqué) : exécutent les boucles de contrôle et gèrent la communication.

- microcontrôleurs, ordinateurs embarqués (ex. SBC), plateformes dédiées.

6) Logiciel : les briques qu'on retrouve partout

Même si chaque robot est différent, on retrouve souvent les mêmes modules.

- **Perception :** extraire une information utile à partir des capteurs (obstacles, objets, scène).
- **Localisation / état :** estimer la pose et la dynamique du robot (position, orientation, vitesse).
- **Planification :** choisir un objectif et produire une trajectoire réalisable (route + évitement local).
- **Contrôle :** suivre la trajectoire malgré le bruit, la latence et les perturbations.
- **Supervision / sûreté :** limites, arrêt d'urgence, modes dégradés, enregistrement (logs).

Idée clé : l'autonomie émerge de la coopération fiable de ces briques, pas d'un module isolé.

7) Autonomie : une chaîne typique (du but à l'exécution)

Une architecture d'autonomie se lit naturellement en couches.

- **Perception + localisation :** comprendre l'environnement et l'état du robot.
- **Planification globale :** définir une mission (objectif, route, waypoints).
- **Planification de comportement :** décider des actions (priorités, interactions, sous-tâches).
- **Planification locale :** produire des trajectoires courtes et réactives.
- **Contrôle / action :** exécuter et corriger en continu (boucle fermée).

Boucle de feedback : les capteurs alimentent en continu l'estimation d'état et la re-planification.

8) Navigation et manipulation : deux capacités fondamentales

Navigation autonome :

- se déplacer sans intervention humaine ;
- éviter obstacles, gérer incertitude capteurs, rester stable et sûr.

Manipulation d'objets :

- **grasping** : saisir des formes variées de manière robuste ;
- manipulation plus complexe : déposer précisément, assembler, effectuer des tâches fines ;
- gestion des contacts et des forces.

Point clé : ces tâches deviennent difficiles dès que l'environnement n'est pas parfaitement contrôlé.

9) Intégration : faire coopérer hardware et software

Dans un robot réel, les problèmes viennent souvent de l'**intégration** (interfaces, timing, cohérence).

Ce qu'il faut faire tenir ensemble :

- données capteurs (fréquences, latences, calibrations, référentiels) ;
- commande actionneurs (saturation, précision, sécurité) ;
- communication entre modules (messages, synchronisation, logs).

Pourquoi un middleware (ex. ROS) est utile :

- structurer le système en modules remplaçables ;
- standardiser la communication et le diagnostic ;
- faciliter le débogage et la reproductibilité via enregistrements.

10) Simulation et tests : avant le monde réel

Pourquoi simuler ?

- tester rapidement et à grande échelle ;
- réduire le risque matériel et humain ;
- valider la chaîne perception–décision–action avant déploiement.

Outils de simulation (exemples non exhaustifs) :

- Isaac Sim / Isaac Lab (écosystème NVIDIA) ;
- Gazebo ;
- MuJoCo (très utilisé en contrôle et robot learning).

Stratégies de tests :

- **SIL** (Software-in-the-loop) : système en simulation ;
- **HIL** (Hardware-in-the-loop) : intégrer du vrai matériel dans la boucle ;
- tests unitaires (modules) et tests système (pipeline complet).

Références

- Kevin M. Lynch, Frank C. Park. *Modern Robotics : Mechanics, Planning, and Control*. Cambridge University Press, 2017.
(ISBN 9781107156302)

- NVIDIA Deep Learning Institute. *A Beginner's Guide to Autonomous Robots*. Cours en ligne (NVIDIA Training).
`https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-OV-35+V1`