ROB-104 — Leçon 9

Intelligence décisionnelle et navigation collaborative

Maîtrise en robotique, intelligence artificielle et systèmes autonomes — Manuel universitaire illustré, formation à distance.

Objectif central : concevoir une architecture où plusieurs robots autonomes partagent l'espace, les informations et les ressources tout en respectant les priorités et les contraintes de sécurité.

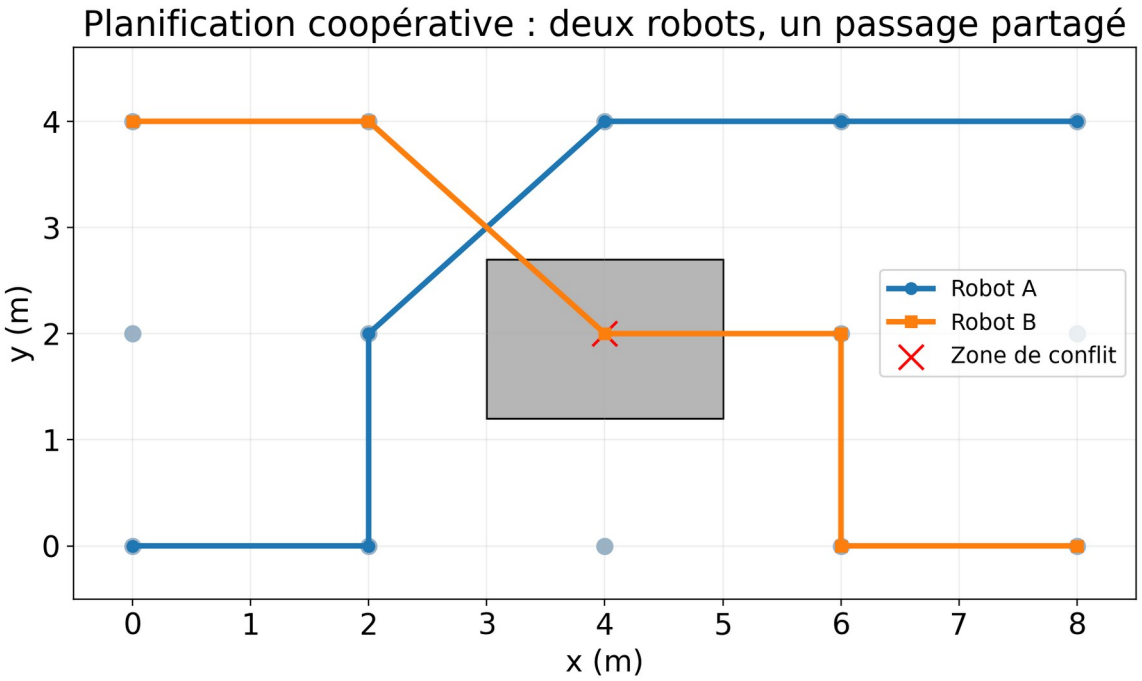


Figure 9.1 — Exemple pédagogique de deux itinéraires dans un espace commun. Les trajectoires sont fictives et servent à l'analyse.

1. Compétences visées

- Modéliser une décision robotique sous contraintes et incertitude.
- Comparer règles, arbres de comportement, planification et apprentissage.

- Définir une coordination centralisée ou distribuée pour plusieurs robots.
- Éviter les conflits de trajectoire et les interblocages.
- Évaluer les performances, les défaillances et les comportements de repli.

2. Architecture de décision autonome

L’autonomie opérationnelle comprend perception, estimation d’état, interprétation de mission, décision, planification, contrôle et surveillance. La couche de sécurité peut imposer une limite ou un arrêt même si le planificateur préfère poursuivre.

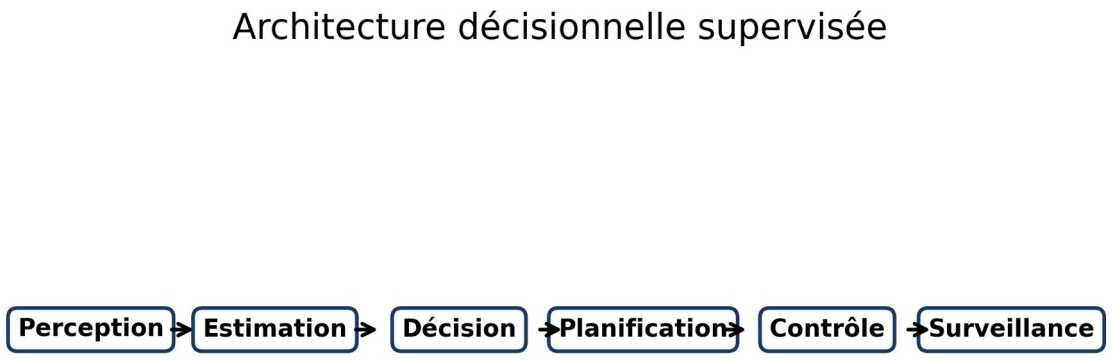


Figure 9.2 — Chaîne décisionnelle simplifiée; la surveillance doit agir transversalement.

Hiérarchie des décisions

Décision stratégique : choisir la mission et les objectifs. Décision tactique : choisir le passage, l’ordre des tâches et les priorités. Décision

opérationnelle : générer des commandes de vitesse et de direction à court horizon. Les fréquences et les horizons de calcul diffèrent.

3. Intelligence artificielle : méthodes et limites

Un système à règles utilise des conditions explicites (« si obstacle proche, ralentir »). Un arbre de comportement organise des tâches avec succès, échec et exécution en cours. Une recherche sur graphe choisit un chemin. Un modèle appris peut prédire le comportement d'un obstacle ou classer une scène, mais sa sortie doit être vérifiée avant de commander un mouvement.

L'apprentissage par renforcement peut optimiser une politique dans un environnement simulé. Il faut ensuite tester les décalages entre simulation et réalité, les situations rares et les contraintes physiques; la performance moyenne seule ne prouve pas la sécurité.

4. Formalisation mathématique de la décision

Un processus décisionnel de Markov peut être représenté par (S,A,T,R,γ), où S est l'ensemble des états, A celui des actions, T le modèle de transition, R la récompense et γ le facteur d'actualisation. En perception partielle, l'agent raisonne plutôt sur une croyance b(s), distribution de probabilité sur les états possibles.

Pour une politique π, l'objectif théorique est de maximiser E[Σ γ^t R(s_t,a_t)], sous contraintes de collision, vitesse, énergie et règles de priorité. Une fonction de récompense ne doit jamais permettre de compenser une collision par un gain de temps.

Exemple numérique : choix d'un itinéraire

Itinéraire A : 18 m, 35 s, risque normalisé 0,10. Itinéraire B : 14 m, 30 s, risque 0,35. Avec J=temps + 40×risque, on obtient J_A=35+4=39 et

J_B=30+14=44 : A est préféré selon ce modèle. Les coefficients sont illustratifs et les deux options doivent d’abord satisfaire les contraintes de sécurité.

5. Coordination de flotte : centralisée ou distribuée

En coordination centralisée, un répartiteur reçoit les missions et attribue les tâches ou passages. En coordination distribuée, les robots échangent directement des intentions et négocient les ressources. Une solution hybride utilise un planificateur de flotte avec autonomie locale et arrêt sûr indépendant.

Les informations utiles incluent identifiant, pose, vitesse, covariance, état de mission, priorité, réservation de zone et horodatage. Une information de position ancienne ne doit pas être considérée comme une réservation actuelle.

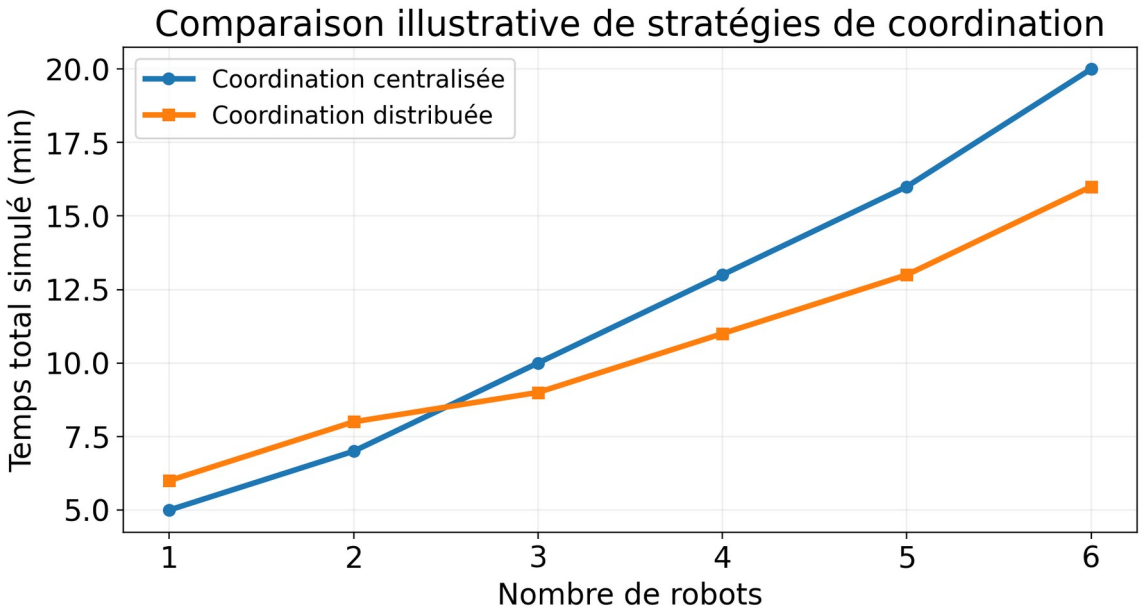


Figure 9.3 — Comparaison illustrative, non issue de mesures réelles : chaque stratégie possède ses coûts de communication et de calcul.

6. Évitement des conflits et interblocages

Deux robots peuvent se bloquer mutuellement à l’entrée d’un corridor. Une règle de priorité déterministe, une réservation temporelle ou une négociation avec délai permet de résoudre ce conflit. Les priorités doivent être stables et connues des participants.

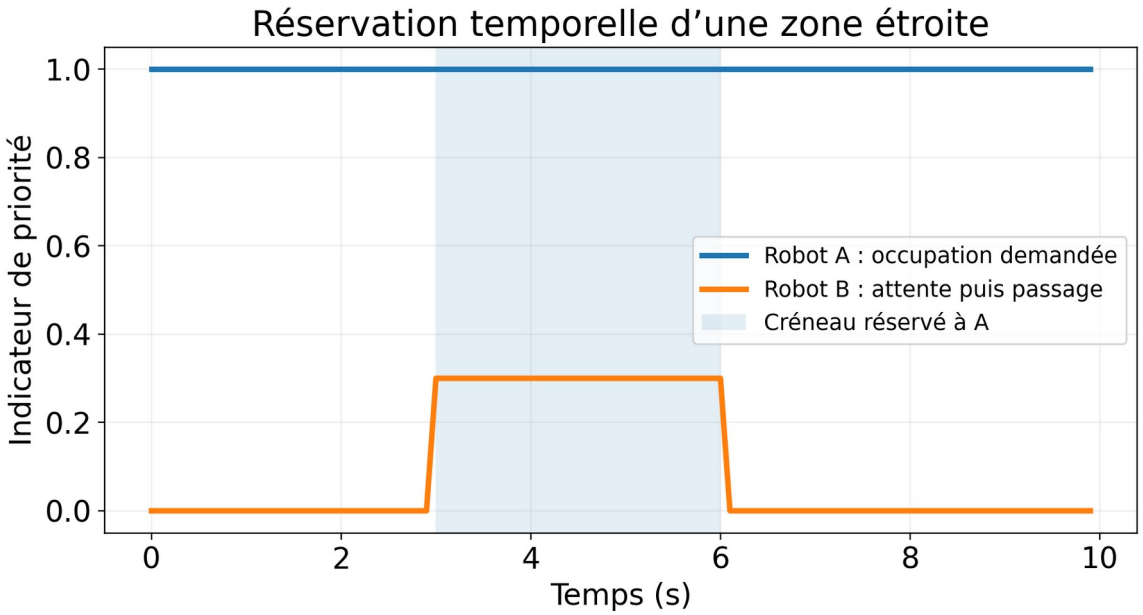


Figure 9.4 — Réservation d’un créneau de circulation pour un passage étroit.

Exemple de réservation

Robot A demande la zone Z de t=3 à t=6 s; robot B demande Z de t=4 à t=7 s. Les intervalles se chevauchent sur [4,6]. Si A obtient la réservation, B attend jusqu’à la libération et la confirmation que la zone est libre, puis planifie son passage. Une réservation ne remplace pas la détection locale d’obstacles.

7. ROS 2 et communications de flotte

ROS 2 permet l’échange de messages, services et actions. Les messages de coordination doivent inclure des timestamps, des identifiants de robot, des séquences et des états explicites. Les politiques QoS (fiabilité, durabilité, profondeur, délai) doivent correspondre aux besoins des données.

La perte de réseau peut rendre une coordination centralisée indisponible. Le robot doit alors continuer uniquement les actions localement sûres, ou s’arrêter si les réservations et obstacles ne peuvent plus être validés.

Exemples de commandes d’inspection sous Ubuntu

```
ros2 node list

ros2 topic list

ros2 topic hz /robot_1/odom

ros2 topic echo /robot_1/odom --once

ros2 action list
```

Les noms de topics sont des exemples et dépendent de la configuration de la simulation. Les robots doivent utiliser des espaces de noms et repères TF non ambigus.

8. Étude de cas : trois robots sur un campus

Robot A transporte des documents, B effectue une inspection et C assure le nettoyage. Ils partagent un corridor et deux portes. Le système attribue des créneaux de passage, surveille les positions et réévalue les itinéraires lorsque des piétons arrivent. Aucune mission ne justifie de dépasser une distance minimale de sécurité.

Indicateurs : durée de mission, temps d’attente, distance parcourue, énergie consommée, conflits détectés, interblocages résolus, distance minimale et nombre d’arrêts sûrs.

9. Laboratoire virtuel : cinq séances

Séance 1 — Environnement

Créer un campus simulé, trois robots différenciés par namespaces et un corridor étroit.

Séance 2 — Planification

Attribuer trois missions, générer les chemins et mesurer les conflits potentiels.

Séance 3 — Réservation

Programmer une file de passage avec priorité et timeout; tester deux demandes simultanées.

Séance 4 — Pannes

Couper la communication d’un robot, introduire un piéton et vérifier la réaction locale sûre.

Séance 5 — Comparaison

Comparer coordination centralisée, distribuée et sans coordination sur 20 essais reproductibles.

10. Tableau de résultats à remplir

Scénario	Durée totale	Conflits / arrêts	Conclusion
Sans coordination	À mesurer	À mesurer	À analyser
Centralisée	À mesurer	À mesurer	À analyser
Distribuée	À mesurer	À mesurer	À analyser
Perte réseau	À mesurer	À mesurer	État sûr à vérifier

11. Évaluation corrigée

Question — Quelle différence entre décision tactique et contrôle opérationnel ?

Corrigé — La tactique choisit les objectifs intermédiaires et priorités; le contrôle calcule les commandes de mouvement.

Question — Pourquoi un modèle appris ne suffit-il pas à garantir la sûreté ?

Corrigé — Ses erreurs possibles et situations hors distribution exigent des contraintes et tests indépendants.

Question — Que faire lorsque deux réservations se chevauchent ?

Corrigé — Accorder une réservation selon une règle explicite et reporter ou replanifier l’autre.

Question — Quel risque présente un message de pose ancien ?

Corrigé — La position réelle peut avoir changé; une décision fondée sur cette pose peut être dangereuse.

Question — Que faire en cas de perte de communication de flotte ?

Corrigé — Limiter les actions à celles dont la sécurité est vérifiable localement, sinon arrêter.

12. Film pédagogique avec dialogues

Scène 1 : le professeur présente trois robots dans le laboratoire. Scène 2 : une étudiante attribue les missions. Scène 3 : deux robots demandent le même corridor; un étudiant explique le conflit. Scène 4 : le système réserve le passage et le second robot attend. Scène 5 : un piéton entre dans la zone; les robots ralentissent ou s’arrêtent. Scène 6 : une perte réseau est

simulée; le professeur vérifie le comportement sûr. Scène 7 : les étudiants présentent les graphiques et défendent leurs décisions.

Livrables : code de coordination, fichiers de simulation, traces horodatées, tableaux de comparaison, vidéo de démonstration et rapport d’analyse.