ROB-104 — Leçon 2

Représentation de l’environnement et planification de trajectoire

Collège du Sacré-Cœur de Gatineau, Québec | Faculté d’informatique | Département de robotique, intelligence artificielle et systèmes autonomes

Auteur : Angelo Nibizi, recteur | Collection : Systèmes autonomes et navigation avancée | 2026



Figure 2.1 — Scène pédagogique illustrée : observation collective d’un robot et de sa carte de navigation.

Objectifs de formation

- Construire une carte d’occupation à partir d’observations de capteurs.
- Représenter un problème de navigation sous forme de graphe pondéré.

- Expliquer et appliquer Dijkstra et A*; choisir une heuristique admissible.
- Distinguer chemin géométrique, trajectoire temporelle et commande motrice.
- Tester une navigation sûre en simulation avec obstacles et replanning.

1 — Voir le robot et ses capteurs

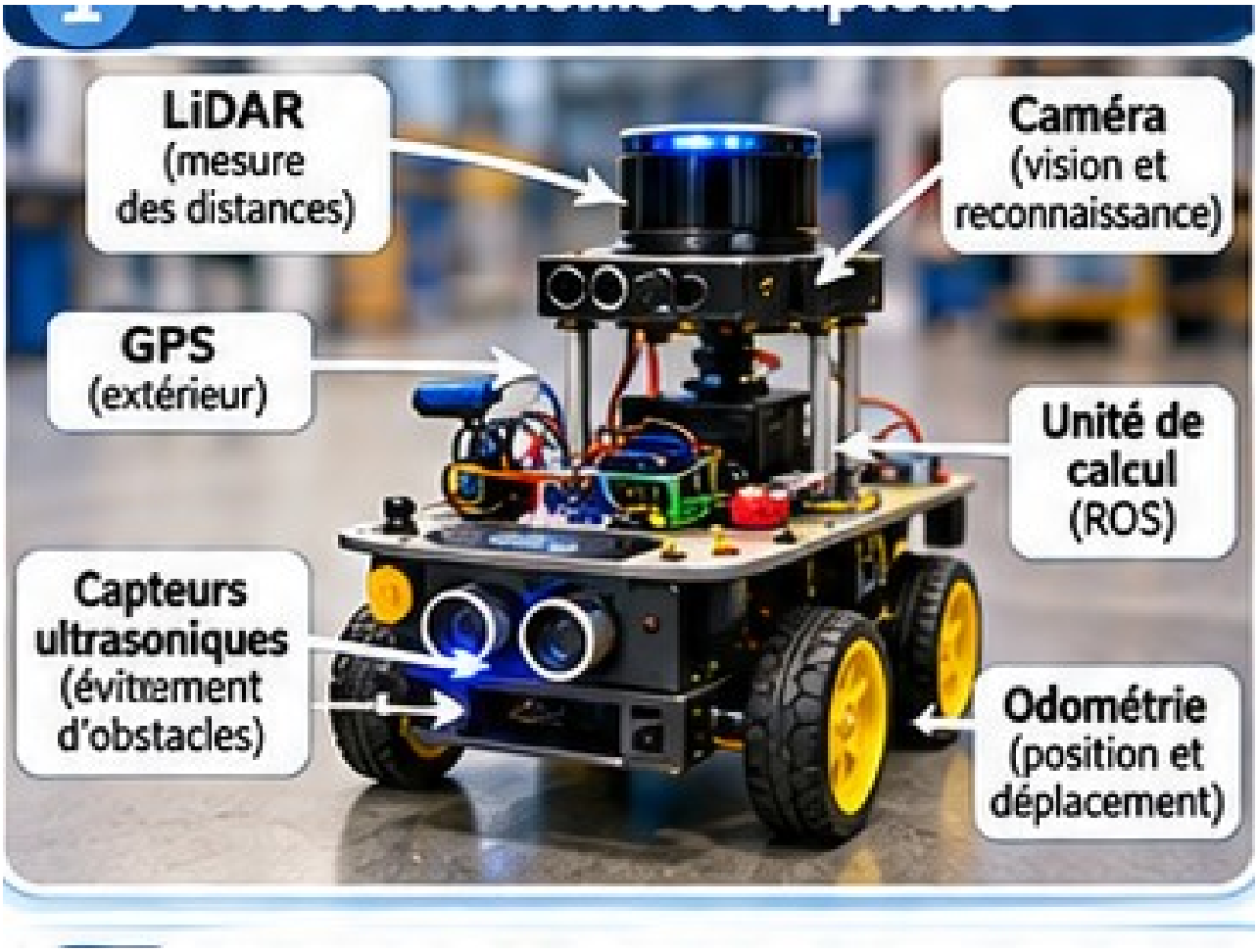


Figure 2.2 — Robot mobile illustré avec capteurs et actionneurs. Identifier physiquement les éléments avant de lire les équations.

Un robot reçoit des observations de LiDAR, caméra et odométrie. Il doit distinguer l'espace libre, les obstacles et les zones non observées. La carte n'est pas le monde : c'est une représentation partielle, datée et incertaine.

2 — Cartes d’occupation

Une grille d’occupation découpe le sol en cellules. Chaque cellule peut porter une probabilité P(occupée) ou un état discret : libre, occupée, inconnue. Les cellules inconnues ne doivent pas être considérées automatiquement comme sûres. Une résolution de 0,10 m/cellule donne 100 cellules par mètre carré.

Exemple : une salle de 6 m × 4 m, à 0,10 m/cellule, exige 60 × 40 = 2 400 cellules. Pour un robot de rayon 0,25 m et une marge de 0,10 m, on peut gonfler les obstacles d’environ 0,35 m, soit quatre cellules par excès dans cette grille.

3 — Du plan au graphe

Chaque cellule libre devient un sommet. On relie les voisines par quatre mouvements (coût 1) ou huit mouvements (coût 1 en ligne droite et √2 en diagonale). Les coûts doivent être non négatifs pour Dijkstra. Le planificateur doit empêcher les diagonales qui coupent le coin d’un obstacle.

4 — Algorithme de Dijkstra

Dijkstra explore les sommets par coût cumulé g(n) croissant. Avec des coûts non négatifs, il garantit un plus court chemin sur le graphe si le but est atteignable. Il n’utilise pas d’information géométrique sur la direction du but.

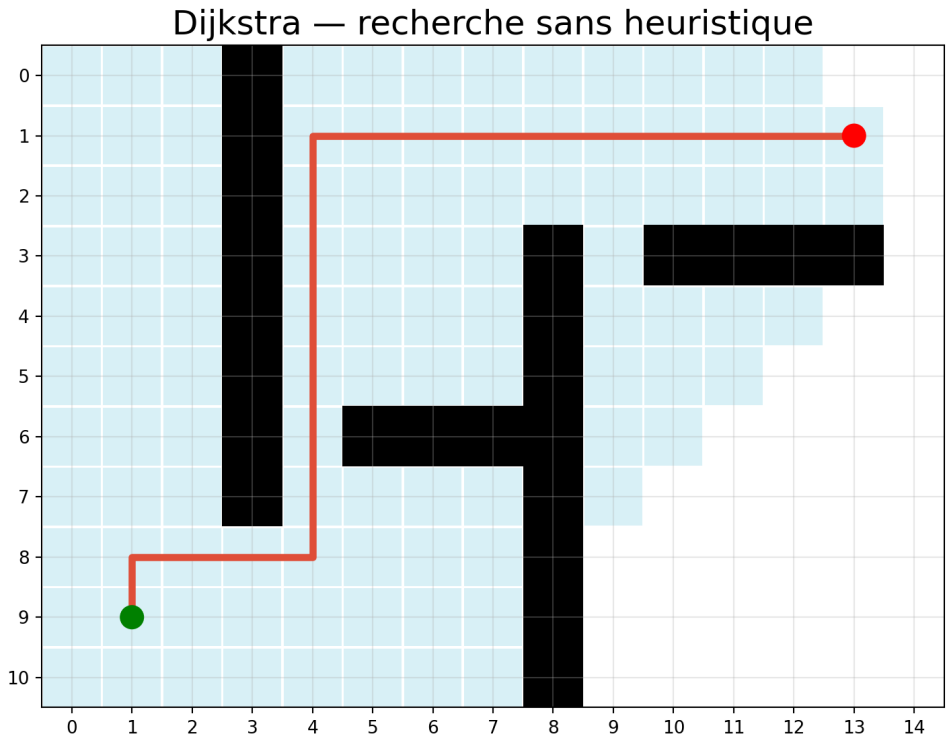


Figure 2.3 — Calcul reproductible sur une grille 15 × 11 : obstacles noirs, cellules explorées en bleu et chemin calculé en rouge.

5 — Algorithme A*

A* utilise f(n)=g(n)+h(n). Ici, g(n) est le coût depuis le départ et h(n) une estimation du coût restant. Avec des déplacements orthogonaux unitaires, la distance de Manhattan h=|x−xbut|+|y−ybut| est admissible et cohérente. A* conserve l’optimalité dans ces conditions et oriente la recherche vers le but.

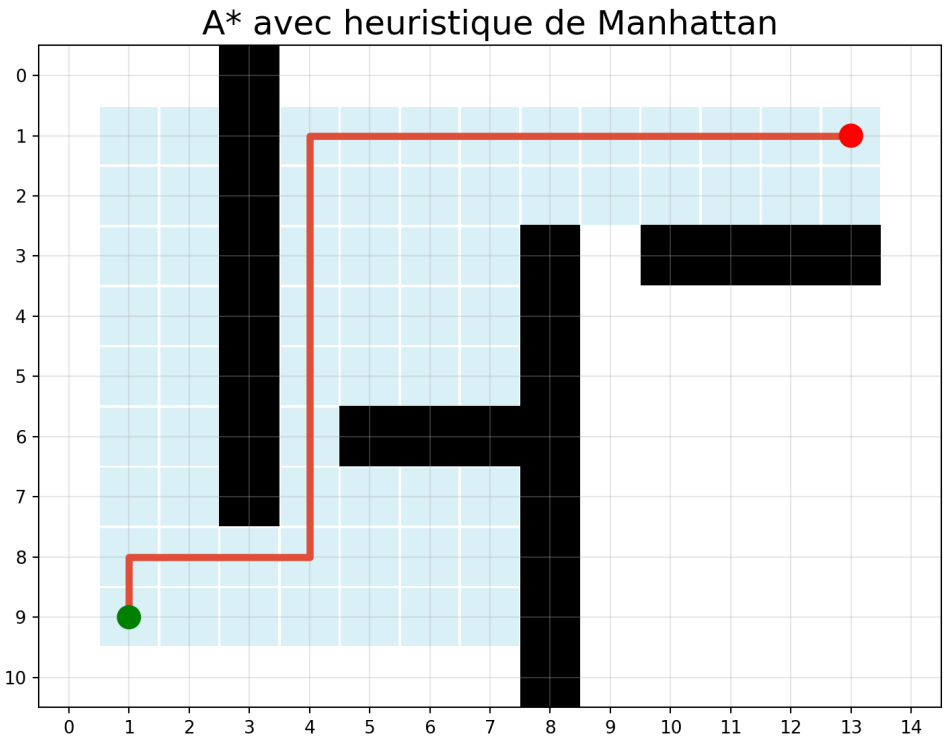


Figure 2.4 — Même carte et même coût, avec A*. Les zones bleues représentent les cellules explorées pendant la recherche.

Important : les deux algorithmes calculent un chemin sur le graphe, pas une trajectoire dynamique directement exécutable. Le robot doit encore vérifier la largeur de passage, sa cinématique, les accélérations, les obstacles mobiles et les limites de sécurité.

6 — Du chemin à la trajectoire



Figure 2.5 — Illustration de navigation parmi des obstacles. Les flèches et trajectoires représentées sont pédagogiques, non des résultats de test certifiés.

Un chemin est une suite de positions; une trajectoire associe à ces positions des temps, vitesses et éventuellement orientations. Un planificateur global produit le chemin général; un planificateur local ou contrôleur suit une trajectoire réalisable et réagit aux changements. Le système doit ralentir ou s’arrêter lorsque la perception ne garantit plus une marge suffisante.

7 — Simulation et laboratoire à distance



Figure 2.6 — Exemple illustré d’un environnement de simulation et d’une carte de navigation.

1. Créer une carte 15 × 11 et placer départ, arrivée et obstacles.
2. Implémenter Dijkstra puis A* avec déplacements à quatre voisins.
3. Afficher cellules visitées, coût du chemin et durée de calcul mesurée sur votre machine.
4. Gonfler les obstacles selon la taille du robot et recalculer.
5. Ajouter un obstacle sur le chemin et déclencher une nouvelle planification.
6. Faire suivre le chemin au robot simulé en limitant vitesse et accélération.

7. Produire un rapport : captures, paramètres, résultats et analyse des échecs.

8 — Étude de cas corrigée

Sur la grille de nos figures, le départ est (1,9) et le but (13,1). Les mouvements sont orthogonaux, de coût 1 par cellule. Le calcul donne un chemin de 20 déplacements. Le nombre de cellules explorées dépend de l'implémentation, de l'ordre de traitement des voisins et des critères d'arrêt : il doit être mesuré et non inventé.

Question : une carte est à 0,20 m/cellule. Quelle distance de déplacement représente un chemin de 24 pas orthogonaux ? Réponse : 24 × 0,20 = 4,80 m, avant lissage et erreurs de suivi.

9 — Évaluation et corrigés

Question — Pourquoi conserver un état « inconnu » ?

Corrigé — Parce que l'absence d'observation n'établit pas que la cellule est libre.

Question — Que représente g(n) dans A* ?

Corrigé — Le coût connu depuis le départ.

Question — Quelle heuristique convient à une grille à quatre voisins de coût unitaire ?

Corrigé — La distance de Manhattan.

Question — Dijkstra et A* produisent-ils directement une commande moteur ?

Corrigé — Non; un contrôleur et une trajectoire faisable sont nécessaires.

Question — Que faire si un obstacle nouveau bloque le chemin ?

10 — Scénario du film pédagogique (séance 2)

Scène 1 : une enseignante et deux étudiants observent un robot mobile dans un laboratoire. Scène 2 : le robot mesure des distances et une carte d’occupation apparaît. Scène 3 : un étudiant explique Dijkstra; l’autre propose A*. Scène 4 : les deux chemins sont calculés sur la même carte. Scène 5 : un obstacle nouveau apparaît; le robot ralentit et replanifie. Scène 6 : les étudiants analysent les résultats et répondent à un questionnaire. Dialogues naturels, voix différenciées, démonstrations visuelles et absence de narration continue.

Travail à remettre : carte, code source, deux captures de recherche, comparaison des algorithmes, essai de replanification et réflexion technique (2 à 4 pages).