

# Architecture de contrôle hybride pour une planification robuste

**Réalisé par :** Koudjo AGBEDANOU

**Encadrants :** M. Halim Djerroud,



# Contexte

## Objectif

Faire de la manipulation robotique guidée par Intelligence Artificielle.

### Dispositif Expérimental :

- Bras robotique **xArm6 Lite**
- Vision stéréoscopique via caméra **ZED2i**.

### Fonctionnement attendu :

- **Fluidité** : Tri ordonné de cubes.
- **Résilience** : Capacité à poursuivre la mission malgré des imprévus.



Vue d'ensemble de l'installation

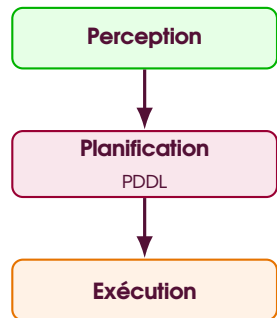
# Fonctionnement du modèle de base : La Boucle Ouverte

## Le cycle de décision séquentiel

- 1 **Perception initiale** : Capture de l'état du monde.
- 2 **Planification délibérative** : Calcul d'une séquence d'actions complète vers le but.
- 3 **Exécution aveugle** : Application des commandes sans vérification intermédiaire.

## Limite majeure

Exécution **statique** : Incapacité de détection des perturbations exogènes après le calcul du plan.



**Problématique : Comment rendre le système réactif à un environnement dynamique ?**

# Divergence modèle / réalité : L'hypothèse du monde clos

## Fonctionnement avec le PDDL standard (1)

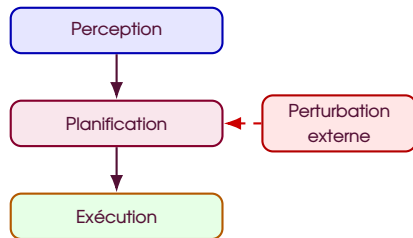
- **Modélisation** : Traduction du monde en fichiers *.pddl*.
- **Planification** : Calcul hors-ligne d'une séquence d'actions.
- **Exécution** : Application en boucle ouverte.

## Hypothèse critique

Le monde est considéré comme **statique** et **prédictible** durant l'exécution du plan.

## Limites du PDDL : (2)

- Principe du monde clos
- Actions déterministes



**Divergence modèle / réalité**

## Conséquences (Observation)

- Tentative de saisie dans le vide
- Mauvais positionnement final
- Plan partiellement invalide

# Panorama des planificateurs

## 1. Déterministe (3)

**Principe** : Recherche d'un chemin optimal dans un graphe d'états certains.

**Limites** : Monde figé

**Représentants** : *Fast Forward (FF)*, *Fast Downward*.

## 2. Temporel (1)

**Principe** : Intégration des durées d'actions et des contraintes de ressources.

**Limites** : Complexité de calcul ; difficulté à se resynchroniser en direct.

**Représentants** : *LPG-td*, *POPF*.

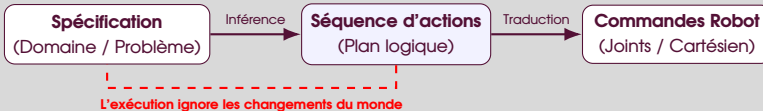
## 3. Probabiliste (4)

**Principe** : Modélisation des effets incertains via des MDP (Markov Decision Processes).

**Limites** : Explosion combinatoire ; latence de résolution.

**Représentants** : *PR-P*, *FF-Replan*.

## Flux d'information : De l'abstraction au mouvement physique



# Alternatives pour la réactivité I : Conditionnelle et Hybride

## 1. Planification Conditionnelle (5, 6)

**Principe :** Création d'un arbre de décision incluant des branches de secours (*Contingency Planning*).

- **Avantage :** Réaction instantanée sans recalcul.
- **Limite :** *Explosion combinatoire* ; anticipation exhaustive d'imprévus hors de portée.

## 2. Architectures Hybrides (BT) (7, 8)

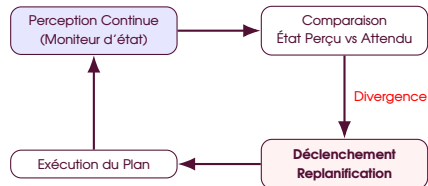
**Principe :** Couplage d'un planificateur (stratégie) et de *Behavior Trees* (réactivité locale).

- **Avantage :** Modularité et robustesse face aux micro-pannes (ex. grip faillure).
- **Limite :** Taux de rafraîchissement

# Alternatives pour la réactivité II : L'Architecture Réactive par Replanification

## Le paradigme de la Boucle Fermée (*Replanning*) (5)

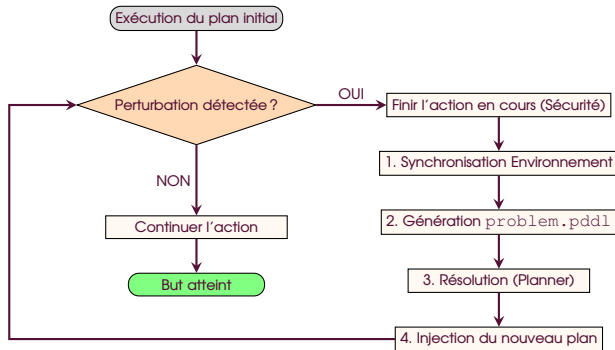
Consiste à traiter l'imprévu comme une nouvelle condition initiale pour un recalcul rapide du plan.



### Possibilités offertes par cette architecture :

- **Adaptabilité immédiate** : Prise en compte immédiate des variations.
- **Simplicité logicielle** : Utilisation de planificateurs existants.
- **Robustesse opérationnelle** : Poursuite de mission malgré les perturbations.

# Algorithme de la Boucle de Replanification



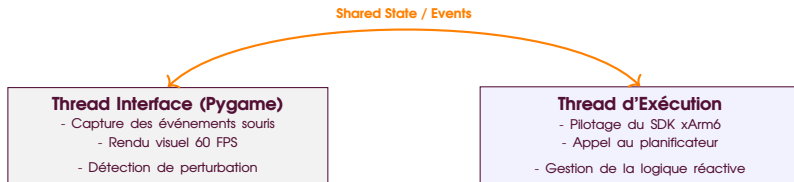
Architecture du système de planification réactive.

## Mécanismes clés :

- **Atomicité** : Garantit la sécurité mécanique en achevant l'action en cours avant toute modification du plan.
- **Synchronisation** : Aligne le modèle logique sur la réalité physique en mettant à jour les coordonnées des objets en temps réel..
- **Injection dynamique** : Mise à jour dynamique du plan sans interruption système.

# Architecture Logicielle : Le Superviseur Asynchrone

- **Problématique** : L'exécution d'un plan et le calcul PDDL sont bloquants pour l'interface utilisateur (GUI).
- **Solution** : Mise en œuvre du *Multi-threading*.



## Avantages

- **Fluidité** : L'interface reste interactive pendant que le robot calcule.
- **Isolation** : Une erreur dans le thread de planification ne fait pas planter l'UI.

# Limites et besoins évolutifs

## Limites : (3)

- **Latence décisionnelle** : Chaque imprévu déclenche un calcul global, ce qui peut nuire à la fluidité du comportement.
- **Dépendance à la visibilité** : Si un cube est masqué (occlusion), le planificateur échoue au lieu de chercher à le retrouver.
- **Perte d'identité** : Si un marqueur ArUco disparaît, le cube est considéré comme supprimé du monde.

## Le problème de l'incertitude sur la connaissance

Actuellement, le robot réagit aux changements d'état physique, mais il ne sait pas gérer le **manque d'information**.

# Planification Épistémique (4)

## Définition

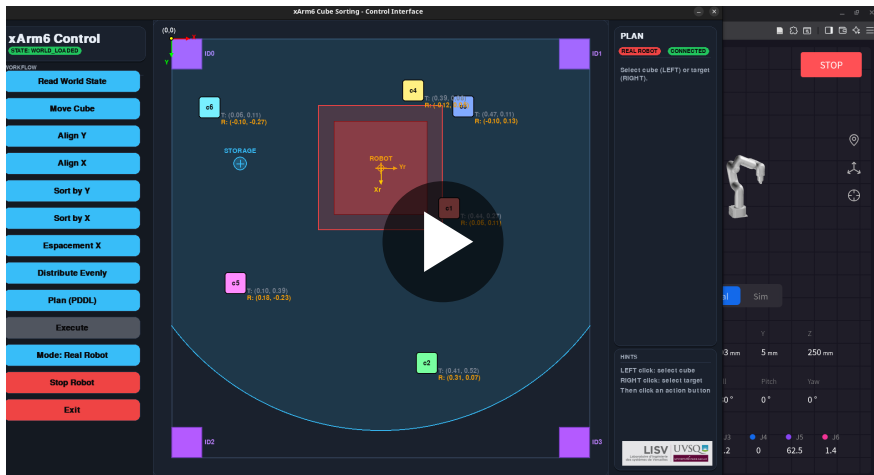
La planification épistémique permet au robot de raisonner non pas sur l'état du monde, mais sur son propre **état de connaissance** (Savoir que l'on ne sait pas).

## Le besoin d'un planificateur "Sur mesure" :

- **Complexité des solveurs actuels** : Les planificateurs épistémiques généraux (ex : MEP) souffrent d'une explosion combinatoire.
- **Spécificité métier** : Notre robot n'a besoin que d'une sous-partie de la logique épistémique (gestion des occlusions et recherche d'objets).



# Démo



# Références I

- (1) M. Fox and D. Long, "Pddl2.1 : An extension to pddl for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.
- (2) M. Fox and D. Long, "Modelling the behaviour of plans in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 25, pp. 97–123, 2006.
- (3) M. Helmert, "The fast downward planning system," in *Journal of Artificial Intelligence Research*, vol. 26, 2006, pp. 191–246.
- (4) L. Dupont and T. C. Son, "Probabilistic epistemic planning for robotic manipulation," in *Proceedings of the 37th International Conference on Automated Planning and Scheduling (ICAPS)*, 2026.
- (5) S. Macenski, F. M. Cano, J. R.-L. Francisco, and V. Matellan, "Plansys2 : A planning system for ros2," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- (6) C. Muise, S. McIlraith, and J. C. Beck, "Pddl3.1 : Explicit contingency planning," in *International Conference on Automated Planning and Scheduling (ICAPS)*, 2014, référence clé pour l'extension du PDDL vers les choix conditionnels.
- (7) M. Colledani and P. Ögren, *Behavior trees in robotics and AI : An introduction*. CRC Press, 2018.
- (8) M. Iovino, E. Scukins, J. Styrud, P. Ögren, and C. Smith, "A survey of behavior trees in robotics and ai," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 2025–2044, 2022, l'état de l'art le plus récent sur les BT en robotique.