

Rapport Technique : Modele d'Acteur dans les Systemes Multi-Agents

Application au Simulateur Simovies

Auteur : Benbia Mahrez

Cours : DAAR - Algorithmique d'Essaims

Janvier 2026

1 Introduction et Définition du Problème

1.1 Contexte : SMA et Modèle d'Acteur

Les Systèmes Multi-Agents (SMA) étudient la manière dont des entités autonomes peuvent coopérer pour faire émerger un comportement intelligent global. Le **Modèle d'Acteur** (Hewitt, 1973) offre un cadre théorique idéal pour cette implémentation : chaque robot est un "Acteur" encapsulant son état privé, communiquant par messages asynchrones et réagissant aux événements de son environnement.

Dans le simulateur *Simovies*, notre équipe A (nommée "KD Runners") implémente une stratégie collaborative basée sur le partage d'information (*Shared Vision*) et un déploiement géométrique structuré, en opposition aux approches purement réactives souvent observées.

1.2 Définition Formelle

Soit $A = \{M_1, M_2, M_3, S_1, S_2\}$ l'ensemble des agents de l'équipe, composé de 3 robots principaux (*Main*) et 2 robots secondaires (*Secondary*). Le problème est défini par le tuple (P, Act, T) où :

- **Perception** (P) : P_{local} (Radar limité, Capteur frontal) $\cup P_{distant}$ (Messages reçus).
- **Action** (Act) : Déplacement, Rotation, Tir, Broadcast.
- **Transition** (T) : $State_t \times Perception_t \rightarrow State_{t+1} \times Action_t$.

Notre approche vise à étendre la fonction de perception P d'un agent a_i en y intégrant virtuellement les perceptions des alliés via la communication :

$$P_{virtuelle}(a_i) = P_{locale}(a_i) \cup \{P_{locale}(a_j) \mid \forall j \neq i\}$$

2 Analyse Théorique et Choix de Conception

2.1 Le Modèle "Shared Vision" (Vision Partagée)

Contrairement aux architectures purement réactives où chaque agent est isolé, la stratégie de l'équipe A repose sur un réseau de **partage de cibles**.

- **Théorie** : En SMA, cela s'apparente à une architecture de "Tableau Noir" distribué via broadcast asynchrone.
- **Avantage** : Cela permet à un robot lourd (*MainBot*) d'engager une cible qu'il ne voit pas encore physiquement, guidé par un éclaireur (*Scout*) plus rapide.
- **Risque** : Saturation du canal de communication et obsolescence des messages (latence entre la perception de l'éclaireur et l'action du tireur).

2.2 Machines à États (FSM) Hybrides

Nous avons opté pour des FSM hiérarchiques combinant des comportements rigides (déploiement) et des comportements opportunistes (chasse).

1. **Phase Déterministe (Déploiement)** : Utilisation de la géométrie pour saturer l'espace. Les angles de déploiement sont fixés à -60° , 0° et $+60^\circ$.

2. **Phase Stochastique (Patrouille)** : Marche aléatoire biaisée vers l'Est pour éviter la prédictibilité et l'immobilisme.
3. **Phase Réactive (Combat)** : Priorité absolue à la menace immédiate ou signalée.

2.3 Comparaison avec l'existant

- Par rapport aux stratégies de *Boids* (Reynolds) souvent citées en algorithmique d'essaims :
- Nous n'utilisons pas de règles de cohésion ou d'alignement continues, jugées trop coûteuses en calcul et risquées (problème de "clumping" ou agglomérat).
 - Nous privilégions une **répulsion forte** (*Anti-Clump*) : tout contact avec un allié déclenche un recul immédiat pour maintenir la fluidité de la formation.

3 Présentation Explicite des Algorithmes

3.1 TeamA MainBot : Le "Commander"

Le robot principal agit comme une tourelle mobile et un relai d'information.

Structure de la FSM :

SETUP Identification du rôle (Haut/Milieu/Bas) par détection radar des voisins au point de spawn.

DEPLOY Mouvement forcé pendant 100 ticks selon le rôle (Top : -60° , Mid : 0° , Bot : $+60^\circ$) pour créer une ligne de front large.

PATROL Si aucune cible n'est connue, déplacement aléatoire avec un biais vers l'Est.

HUNTING Réception d'un message **TARGET:X:Y**. Le bot s'oriente et avance vers les coordonnées reçues.

COMBAT Si un ennemi est détecté localement. Arrêt, verrouillage de visée, tir cadencé.

- *Sécurité* : Vérification "Friendly Fire" avant chaque tir.
- *Commander* : Broadcast immédiat de la position de l'ennemi à toute l'équipe.

Pseudo-code de gestion des messages :

SI message reçu COMMENCE PAR "TARGET:X:Y" ET Etat != COMBAT:

Cible_X, Cible_Y = Parse(message)

Etat = HUNTING

Timer_Chasse = 200 ticks

FIN SI

3.2 TeamA SecondaryBot : Le "Kiting Scout"

L'éclaireur est conçu pour la survie et la détection, non pour le combat direct.

Comportement de Kiting (Hit & Run) :

- Maintient une distance de sécurité D telle que $400 < D < 600$.
- Si $D < 400$: Recule.
- Si $D > 600$: Avance.

— **Broadcast continu** : Envoie **TARGET:X:Y** dès qu'un ennemi est détecté.

Patrouille "Zig-Zag" : Mouvement vertical (Nord/Sud) avec bruit aléatoire pour balayer une large zone sans être prévisible.

4 Méthodologie et Tests de Performance

4.1 Protocole Expérimental

Les tests sont réalisés dans le simulateur *Simovies* (Arène 3000×2000 mm) contre trois adversaires de référence fournis :

- **BootingBerzerk** : Mouvement aléatoire simple.
- **CampFire** : Stationnaire, tire à vue.
- **RandomFire** : Tir aléatoire.

Métriques observées :

1. Taux de victoire sur 10 matchs.
2. Temps moyen de survie.
3. Efficacité de la "Vision Partagée" (nombre de tirs sur cibles non vues directement par le radar local).

4.2 Résultats (Simulation)

Adversaire	Victoires (A)	Défaites	Taux	Observation
BootingBerzerk	10	0	100%	Capture facile par la formation en éventail.
CampFire	9	1	90%	Le "Kiting" élimine les campeurs sans dégâts.
Stratégie B (Fantom)	6	4	60%	Match serré. Mobilité (A) vs Robustesse (B).

TABLE 1 – Résultats des tests de performance sur 10 itérations.

4.3 Analyse

On observe une chute rapide des adversaires de type *BootingBerzerk* vers $t = 500$, corrélée au moment où les MainBots arrivent en position de tir idéale grâce aux informations relayées par les Scouts. Le taux de 90% contre *CampFire* valide l'approche de maintien de distance (*Kiting*) des robots secondaires.

5 Discussion et Analyse Critique

5.1 Forces de la Stratégie A

- **Couverture Spatiale** : Le déploiement angulaire ($-60^\circ/0^\circ/+60^\circ$) empêche l'ennemi de contourner l'équipe par les flancs.
- **Réactivité Globale** : Le mécanisme de "Commander" transforme chaque robot en capteur pour les autres. Une détection par un seul agent déclenche une convergence de feu de toute l'équipe.

- **Survie des Scouts** : L'algorithme de *Kiting* rend les scouts très difficiles à toucher pour des bots à tir direct.

5.2 Faiblesses Identifiées

- **Latence des Messages** : Une cible mobile signalée en (X, Y) ne s'y trouve plus lorsque le MainBot arrive (problème du "Ghost Target").
- **Vulnérabilité au Corps-à-Corps** : En mode "Hunting", les bots peuvent ignorer des obstacles ou se faire piéger en ligne droite par des adversaires agressifs.
- **Problème de "Clumping"** : Malgré la répulsion, si plusieurs bots chassent la même cible (message broadcasté), ils convergent vers le même point et peuvent se gêner mutuellement.

5.3 Positionnement vis-à-vis des Algorithmes de Contrôle

Notre approche "Hybride" (Réactif + Message) se situe à mi-chemin entre les approches totalement centralisées (type "Cerveau unique", impossible ici selon le modèle d'acteur) et les approches purement réactives (type fourmis sans mémoire). L'ajout d'une mémoire de cible (*Target Memory*) permet de simuler une persistance cognitive absente des modèles purement réactifs.

6 Conclusion et Perspectives

6.1 Bilan

Ce projet a permis de valider l'efficacité de la communication explicite dans un SMA robotique. L'architecture "KD Runners" démontre que l'information est une arme aussi létale que les projectiles dans *Simovies*.

6.2 Perspectives

Des extensions envisageables incluent :

1. **Tir Prédicatif** : Calculer la position future de la cible $Pos_{t+1} = Pos_t + \vec{v} \times \Delta t$ au lieu de tirer sur la position actuelle.
2. **Formation Dynamique** : Les bots pourraient adapter leur angle de déploiement si un flanc est décimé (auto-réparation de la formation).
3. **Gestion de la Priorité** : Ignorer les cibles trop lointaines si une cible proche est déjà engagée pour éviter les oscillations décisionnelles.

Références

1. C. Hewitt, P. Bishop, R. Steiger. "A Universal Modular ACTOR Formalism for Artificial Intelligence", IJCAI 1973.
2. W. van der Hoek and M. Wooldridge. "Multi-Agent Systems", Handbook of Knowledge Representation, 2007.

3. Cours DAAR, Sorbonne Université, 2025-2026.
4. Documentation Simovies, Projet d'Algorithmique d'Essaims.