

AIPL による Swarm Robotics の研究

型付き Actor と局所通信による群ロボット制御の設計

Yasushi Kodama Repository Report

2026 年 5 月 4 日

概要

本稿は、AIPL を用いて Swarm Robotics を記述・実装する研究構想を述べる。Swarm Robotics は、多数の比較的単純なロボットが局所通信と局所観測に基づいて協調し、探索、搬送、被覆、隊列形成、災害対応、環境監視などを行う分散ロボットシステムである。AIPL は actor、message、future、select、capability、protocol を持つ actor-first 言語であり、各ロボット、基地局、環境センサ、タスク管理者を actor として表現できる。

本稿では、AIPL による Swarm Robotics の基本モデル、型付き message による安全な局所通信、群行動 protocol、シミュレータ設計、実機展開への段階的移行を論じる。特に、ロボットの位置、速度、電池、センサ値、近傍情報、タスク割当を型で区別し、誤った制御 message や capability 違反を防ぐ設計を提案する。

目次

1	はじめに	3
2	AIPL と Swarm Robotics	3
3	Actor モデル	3
4	型付き局所通信	4
5	群行動の基本戦略	4
5.1	探索	5
5.2	被覆	5
5.3	隊列形成	5
5.4	タスク割当	5
6	protocol と安全性	6
7	シミュレータ設計	6
8	評価指標	6

9	実機展開	7
10	課題	7
11	結論	8

1 はじめに

Swarm Robotics は、多数のロボットが中央集権的な制御なしに協調する分散システムである。個々のロボットは限定されたセンサ、通信範囲、計算資源、電池を持つが、群全体としては広域探索、地図作成、物体搬送、監視、災害現場調査などを実行できる。

このような system の難しさは、個々のロボット制御だけでなく、群全体の相互作用にある。通信遅延、通信断、局所観測、衝突回避、タスク競合、電池切れ、環境変化が同時に発生する。そのため、Swarm Robotics の記述言語には、並行性、局所通信、非同期 message、capability、protocol、simulation を自然に扱う能力が必要である。

AIPL は actor-first 言語であり、各ロボットを actor として表現できる。ロボット間通信は message passing として記述され、群行動は protocol または局所規則として表現される。型システムは、位置、速度、センサ値、タスク、制御命令を区別し、群制御 program の接続ミスを減らす。

2 AIPL と Swarm Robotics

定義 1 (AIPL Swarm). *AIPL Swarm* とは、複数の *RobotActor* が環境 *actor*、基地局 *actor*、*task manager actor* と *message* を交換しながら、群行動を実現する *AIPL program* である。

基本構成は次である。

$$Swarm = RobotActor^N + EnvironmentActor + TaskManagerActor + BaseStationActor$$

RobotActor は局所状態を持つ。

$$State = \{position, velocity, battery, sensors, task, neighbours\}$$

各状態要素は型で区別する。

$$Position \neq Velocity \neq BatteryLevel \neq TaskId$$

3 Actor モデル

AIPL では、ロボットを class として定義する。

```
class Robot {
  var id = "";
  var battery = 100.;
  var task = "idle";

  method sense() {
    var obs = sensor_read(id);
```

```

    reply(obs);
}

method assign(new_task) {
    task = new_task;
    reply("accepted");
}

method move(dx, dy) {
    motor_move(dx, dy);
}
}

```

この例では、sense、assign、move が RobotActor の interface である。実際の型付き設計では、dx,dy は単なる float ではなく Velocity または MotionCommand として扱う。

4 型付き局所通信

Swarm Robotics では、ロボットは近傍ロボットとだけ通信できることが多い。AIPL では、通信範囲を capability として扱う。

$$LocalComm(r_i, r_j) \Rightarrow send\ r_j.message(...)$$

通信 message の型を表 1 に示す。

表 1: Swarm Robotics の message 型

Message	型
hello	RobotId * Position * BatteryLevel -i Unit
task_offer	TaskId * Region * Priority -i Response
obstacle	Position * Confidence -i Unit
formation	FormationId * RelativePose -i Unit
rendezvous	Position * Time -i Unit
help	RobotId * FailureReason -i Unit

型付き message により、障害物情報を task id と取り違える、battery level を位置として扱う、formation command を通常移動命令として扱う、といった誤りを防げる。

5 群行動の基本戦略

AIPL Swarm で扱う基本的な群行動は次である。

1. 探索

2. 被覆
3. 隊列形成
4. タスク割当
5. 情報拡散
6. 故障回復

5.1 探索

探索では、各 RobotActor が未知領域を局所的に選び、観測結果を共有する。

$$Robot.sense \rightarrow Environment.update \rightarrow Map.merge$$

frontier-based exploration では、未探索領域の境界を frontier として表現し、TaskManager が frontier を RobotActor に割り当てる。

$$assign : RobotId * Frontier \rightarrow Unit$$

5.2 被覆

被覆では、群全体で領域を覆う。各 robot は近傍との距離を保ちつつ移動する。

$$Control = attraction(goal) + repulsion(neighbours) + obstacleAvoidance$$

AIPL では、各制御項を actor または pure primitive として分離できる。

5.3 隊列形成

隊列形成では、leader と follower の相対位置を保つ。

$$FormationSpec = \{leader, relativePose_i\}$$

formation message は FormationId と RelativePose を持つため、通常の移動 command と区別できる。

5.4 タスク割当

タスク割当は、robot の能力、距離、電池、現在 task に基づいて決める。

$$score(r, t) = w_a distance(r, t) + w_b battery(r) + w_c capability(r, t)$$

TaskManagerActor は score に基づき割当を行う。

$$task.assign : Task * RobotSet \rightarrow Assignment$$

6 protocol と安全性

群ロボットでは、局所 message の型だけでは十分でない。たとえば、搬送 task では approach、grasp、lift、move、release の順序が重要である。

$$approach \rightarrow grasp \rightarrow lift \rightarrow transport \rightarrow release$$

この順序を protocol として表現する。

$$\Gamma; \Delta \vdash action \triangleright \Delta'$$

protocol checker は、危険な順序違反を防ぐ。たとえば、lift の前に grasp が完了していなければ、lift command を拒否する。

7 シミュレータ設計

AIPL Swarm の研究にはシミュレータが不可欠である。シミュレータは次の actor で構成する。

表 2: AIPL Swarm シミュレータの actor

Actor	役割
WorldActor	時刻、地図、障害物、物理状態を管理する
RobotActor	個々のロボット制御 logic を実行する
SensorActor	simulated sensor reading を生成する
PhysicsActor	移動、衝突、速度制約を計算する
NetworkActor	通信範囲、遅延、packet loss を模擬する
TaskManagerActor	task 生成と割当を行う
MonitorActor	safety violation、coverage、energy、completion time を記録する

simulation loop は次である。

$$sense \rightarrow decide \rightarrow communicate \rightarrow move \rightarrow physics \rightarrow monitor$$

AIPL の future により、各 RobotActor の decision step を並列に実行できる。

8 評価指標

Swarm Robotics の評価指標は、単一 robot の性能ではなく群全体の性質である。

表 3: 評価指標

指標	内容
Coverage	指定領域の探索・監視率
Completion time	task 完了までの時間
Energy consumption	群全体および個体ごとの消費電力
Robustness	robot 故障や通信断への耐性
Scalability	robot 数を増やしたときの性能変化
Safety	衝突、危険 command、protocol violation の数
Communication cost	message 数、帯域、遅延

9 実機展開

シミュレータから実機へ移行するには、AIPL actor interface を保ったまま、SensorActor と MotorActor を実デバイス adapter に差し替える。

$$SimSensorActor \Rightarrow RealSensorActor$$

$$SimMotorActor \Rightarrow RealMotorActor$$

この差し替えを安全に行うため、device capability と型付き interface が必要である。実機 robot は、通信断、電池低下、センサ異常を message として報告する。

$$failure : RobotId * FailureReason \rightarrow Unit$$

10 課題

AIPL Swarm には以下の課題がある。

- 大量 actor 実行の性能。
- 実時間制約と garbage collection の調整。
- 連続値制御と離散 message protocol の統合。
- 通信断を前提とした protocol 設計。
- 安全性検証と物理環境の不確実性。

特に重要なのは、型が物理世界の不確実性を消すわけではない点である。型は command と data の構造を守るが、センサ誤差や滑り、衝突の可能性は monitor と controller によって扱う必要がある。

11 結論

本稿では、AIPL による Swarm Robotics の研究構想を示した。AIPL の actor、message、future、select、capability、protocol は、群ロボットの局所通信と分散制御を自然に表現できる。型付き message により、位置、速度、電池、task、障害物、formation command を区別でき、群制御 program の接続ミス減らせる。

今後の課題は、AIPL Swarm シミュレータを実装し、探索、被覆、隊列形成、タスク割当、故障回復の benchmark を整備することである。その後、sim-to-real の actor interface を設計し、実機 robot への段階的展開を行う。