Distributed Coverage Control for Networked Multi-Robot Systems in Any Environment

Pham Duy Hung[1], Tran Quang Vinh[1], and Trung Dung Ngo[2]
[1] University of Engineering and Technology, Hanoi, Vietnam; [2] University of Brunei Darussalam, Brunei Darussalam

INTRODUCTION

- Multi-robot coverage has been recently received a particular attention due to its potential applications, e.g., surveillance and reconnaissance, security and patrolling, environmental monitoring.
- Principles of distributed coverage control for multi-robot systems are classified into two categories: artificial potential force-based control and geometry-based control.
- In previous works, none of method can guarantee the global network integrity for collective decision making and deal with any environments including both the obstacle-free, and convex and non-convex structured environments.
- In this paper, the distributed coverage control is developed by combining the inter-connected hexagon-lattices based coverage and the hierarchical distributed control of networked multi-robot systems. The distributed coverage control is capable of driving mobile robots to move in the network to cover any environments in the configuration of inter-connected hexagonal-lattices.

Background

1. Hierarchical Distributed Control for Swarm Movement and Network Preservation

- To deploy a multi-robot system into an unknown environment while preserving a global network integrity.
- Synthesised by the distributed node control keeping all the robots connected in the network (levels 1 and 2), and the distributed connectivity control removing redundant critical connectivities formed in triangle or k-connected topology preventing the network expansion for exploration applications (levels 3 and 4).
- Composed of four levels event-trigged control in hierarchy.

Level 1: Behavioural Control

$$\vec{v}_{bi} = \alpha \vec{\nabla}_i^c + \beta \vec{\nabla}_i^s + \gamma \vec{\nabla}_i^a$$

- **Cohesion:** $\vec{\nabla}_i^c$
- **Separation:** $\vec{\nabla}_i^s$
- **Alignment:** $\vec{\nabla}_i^a$
- **Adjustable factors: α, β, γ**

Level 2: Network Preservation

$$\Delta x_i \le \begin{cases} \frac{\varepsilon}{2} & C_i = \emptyset \\ \frac{\varepsilon_i}{2} & C_i \neq \emptyset \end{cases}$$

Tolerances: $\varepsilon \triangleq r_c - r_f$

$\varepsilon_i \triangleq \min_{j \in C_i}(r_c - r_{ij}) \le \varepsilon$

Minimisation of Topologies:

Level 3: Triangle Topologies

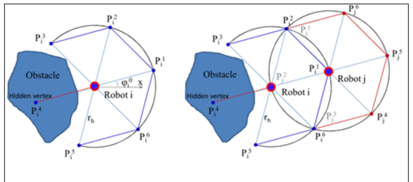
Level 4: K-Connected Topologies

2. Coverage

- **Coverage area of multi-robot system**: is synthesised by the coverage area of every robot in network.

$$\mathbf{C} = \bigcup_{i \in N} \mathbf{S}_i \iff \mathbf{C} = \mathbf{C}_{in} \cup \mathbf{C}_o$$

- **Hexagonal lattice:** created by occupying vertices by robots. Each occupied robot, so-called as a vertex, has six vertices uniformly located around itself with radius r_h. A Hexagonal lattice can be formed by hidden vertices existing on obstacles domains. Thanks to this feature, it can represent the convex or non-convex shape of an environment.



(a) Five virtual targets (b) Three virtual targets

Fig. 1. Virtual Targets Generation.

- **Virtual Target Generation:** when a robot has occupied a vertex, it generates maximum N_v=6 other vertices uniformly located around itself within radius r_h.

$$P_i = \{P_i^j, j \in I \mid P_i^j \notin (P_i^o \cup P_i^{N_i^o} \cup P_i^{ob})\}$$

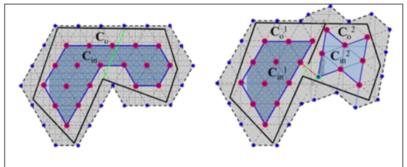
where: $P_i^j = x_i + (r_h cos\varphi_i^j, r_h sin\varphi_i^j)$

$\varphi_i^j = \varphi_i^0 + (j-1)\varphi_H$ and $j \in I = [1, ..., N_v]$

Penalty node as virtual target: created to connect two contiguous sub-environments by a passage with $d \le r_h$.

$$P_i^p = \{P_i^{p_{jk}}, j \in I, k \in I^+ \mid P_i^j, P_i^k \notin E_c\}$$

Thus, virtual target set: $P_i^+ = P_i \cup P_i^p$



(a) $d > r_h$ (b) $d < r_h$

Fig. 2. Multi-Robot coverage including the inner boundary area (blue) and the outer boundary area (grey). The passage between two contiguous sub-environments (green dashed line) can be either $d > r_h$ or $d < r_h$.

Distributed Coverage Control

- **Three states:**
  - ✓ **Occupied:** robot has occupied a virtual target. It becomes a landmark for attracting free robots.
  - ✓ **Assigned:** robot has been assigned for a virtual target.
  - ✓ **Unassigned:** robot is a static node waiting for an assignment. It is a free robots.
- **Two phases:**
  - ✓ **Phase 1 – Assignment**
    - A landmark $l_c \in L_m$ is active. It broadcasts a message to attract free robots to occupy its virtual targets.
    - Free robots receive such a message and find the shortest routes to l_c. If it is assigned, it will receive route≠∅. $N_{P_{\ell_i}^+}$ free robots are confirmed and assigned for $N_{P_{\ell_i}^+}$ virtual targets.
    - When $t_c > N_{P_{\ell_i}^+}$, the next landmark in L_m is active: $\ell_c \leftarrow \ell_c + 1$
  - ✓ **Phase 2 – Occupation**
    - Assigned robot moves along the route toward the landmark by using algorithm 3.
    - When robot has reached the target, it generate new virtual targets for itself.

Algorithm 2: Distributed Area Coverage

```
Initialize: ℓc = 1, tc = 1
for All robots i do
  switch States do
    case unAssigned
      Stop
      Route = ShortestPath(ℓc)
      if Route ≠ ∅ then
        States ← Assigned
    case Assigned
      Call Algorithm 3
      if Flag_i = 1 then
        Get AssingedTarget
        Move to AssingedTarget
        if Reach AssingedTarget then
          States ← Occupied
    case Occupied
      Stop
      if FirstTime then
        Call Algorithm 1
        if P_i ≠ ∅ then
          Lm ← Lm + 1
      if Robot i is ℓc then
        MessageBroadcasting
        if Confirmed then
          ConfirmedRobot ← tc
          tc ← tc + 1
        if All P_i are assigned (tc > N_{P_i^+})
        then
          ℓc ← ℓc + 1
          tc ← 1
```

Algorithm 3: In-network Mobility Control

```
Input: Route_i, IDc_i, MaxID_i
Output: Flag_i
switch IDc_i do
  case MaxID_i
    Flag_i ← 1
  otherwise
    Flag_i ← 0
    Dest_1 ← Route(IDc_i)
    Dest_2 ← Route(IDc_i + 1)
    Move to Dest_1
    if Observe Dest_2 then
      IDc_i = IDc_i + 1
```

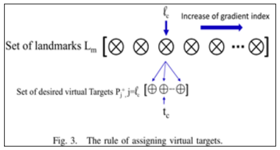


Fig. 3. The rule of assigning virtual targets.

Algorithm 1: Virtual Target Generation

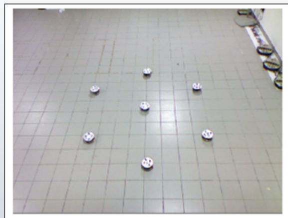
```
Input: φ_i^0, φ_H
Output: P_i
Get P_i^o, P_i^{N_i^o}
for j ∈ I do
  φ_i^j = φ_i^0 + (j − 1)φ_H
  P_i^j = x_i + (r_h cosφ_i^j, r_h sinφ_i^j)
  if P_i^j ∈ Ob then
    Get k ∈ I^+
    if (∃P_i^k ∈ P_i^ob) ∧ (Lp_jk ∩ E_p ≠ ∅) then
      φ_i^{p_jk} = φ_i^j + ρ_p (φ_i^j + φ_i^k)/2
      P_i^{p_jk} = x_i + (r_h cosφ_i^{p_jk}, r_h sinφ_i^{p_jk})
      P_i^p ← P_i^{p_jk}
    P_i^ob ← P_i^j
  if P_i^j ∉ (P_i^o ∪ P_i^{N_i^o} ∪ P_i^ob) then
    P_i ← P_i^j
P_i^+ = P_i ∪ P_i^p
```

CONCLUSIONS

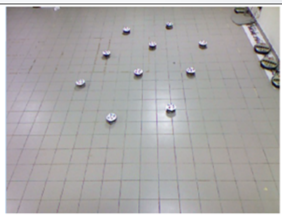
We have presented a new method of distributed coverage control for the multi-robot coverage in hexagonal lattice configuration. The coverage is generated by tracking and occupying desired virtual targets created by occupied robots. Using the hierarchical distributed control, the robots can move to occupy their assigned virtual targets while maintaining the global network integrity for in-network mobility and data communication. We validated experiments on up-to 14 real mobile robots in both the obstacle-free environments and the non-convex structured environment.

RESULTS

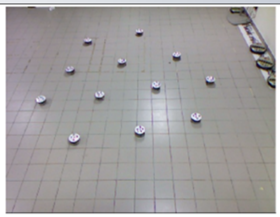
1. The multi-robot coverage in an obstacle – free - environment



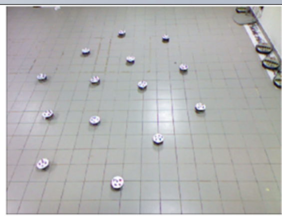
(a) 7 robots – 1 hexagonal lattice



(b) 10 robots – 2 hexagonal lattice

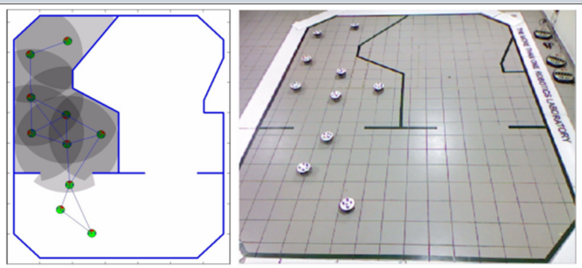


(c) 12 robots – 3 hexagonal lattice

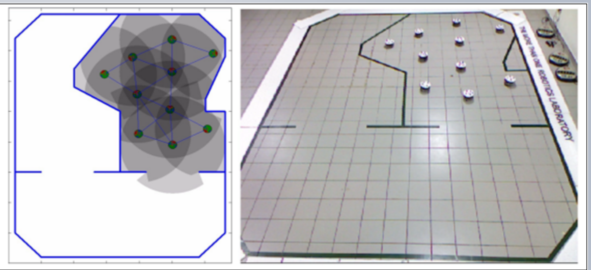


(d) 14 robots – 4 hexagonal lattice

2. The multi-robot coverage in a non-convex environment consisting of two rooms



(a) Using 7 robots to coverage the room



(b) Using 10 robots to coverage the room