A Minimalist Solution to the Multi-Robot Barrier Coverage Problem

Thomas Green, Kevin Kamel, Siyuan Li, Christopher Shinn, Paolo Toscano, Xintong Wang, Yuchen Ye, and Roderich Groß

Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK
{tdggreen1, kkgkamel1, sli101, cshinn1, ptoscano1, xwang309, yye30, r.gross}@sheffield.ac.uk

Abstract. This paper addresses the multi-robot barrier coverage problem. It presents a group of memory-less robots that encircle a group of herd agents, by moving along a polygonal barrier. The results, produced from simulations in CoppeliaSim, demonstrate high retention of herd agents, and robust performance across a range of simulated scenarios.

Keywords: Coverage · multi-robot system · swarm robotics.

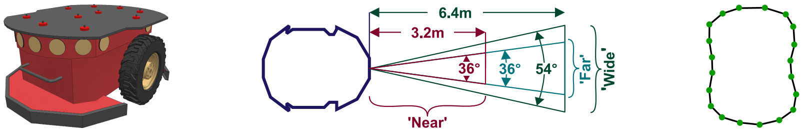
1 Introduction

This study addresses multi-robot barrier coverage, an extensively studied problem [8,5,7,6] with a variety of potential applications in defence, and beyond, such as developing robots as alternatives to sheepdogs [8,7], mine sweeping [5], as well as in understanding swarming behaviours in sheep and other organisms [7]. For large-scale applications, such as oil spill clean-ups [9], solutions with low hardware requirements could improve feasibility. Here, the computation-free swarming paradigm [1] is used in controlling a group of barrier coverage robots.

2 Methods

A herd of n simulated terrestrial agents is to be contained within a 2-D polygonal region, delineated by p vertices, represented by non-collidable green discs on the ground. A minimalist solution is realised using m simple, memory-less, identical barrier coverage robots (BCRs) whose aim is to minimise the number of herd agents (HAs) crossing the boundary. Simulations are carried out in CoppeliaSim Edu 4.1.0. The default setup comprises a regular polygon where $p = 20$ and side length is 1 m, $n = 10$ homogeneous HAs, and $m = 4$ BCRs. The HAs [BCRs] start from random positions and orientations within the polygonal region, less [more] than 1 m away from its centre. The robot controllers are updated at 2 Hz.

Barrier Coverage Robot Design: BCRs are simulated using the Pioneer P3-DX (see Fig. 1(a)), a differential wheeled robot weighing $\sim$ 9 kg. In this study, the motor velocities can take continuous values in a range equivalent to



(a) (b) (c)

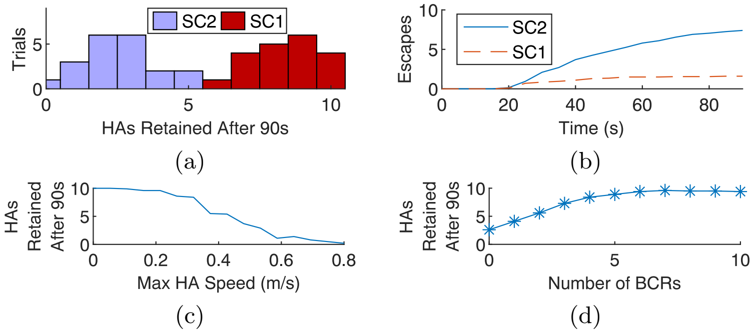
Fig. 1. (a) The barrier coverage robot (BCR), a Pioneer P3-DX. (b) BCR sensors (not to scale). The range of 'far' and 'wide' equals d, the distance between opposing boundary vertices. Range of 'near' is $d/2$. (c) Irregular polygonal barrier used in SC3.

$[-0.80, 0.80]$ m/s. Each BCR has three discrete-state, pyramidal sensors mounted in the forward-facing direction, detecting objects that are near, far, or in a wide range, respectively (see Fig. 1(b)). They take values $S_{near} \in \{0, G, BC, H\}$, $S_{far} \in \{0, G\}$, and $S_{wide} \in \{0, BC\}$, where $0, G, BC$, and H, respectively represent no object, green disc, other BCR, and HA detection. The BCR has hence $4 \cdot 2 \cdot 2 = 16$ sensing states. Let $\bar{v}_l, \bar{v}_r \in [-1, 1]$ represent the normalised left and right BCR wheel velocities, respectively, where a wheel velocity value of -1 [1] corresponds to the wheel turning backwards [forwards] at maximum velocity. The controller maps the sensor readings $S_{near}, S_{far}, S_{wide}$ onto the wheel velocities $\bar{v}_l, \bar{v}_r$: $\{0,1,2,3\} \times \{0,1\} \times \{0,1\} \rightarrow [-1,1]^{32}$. It outputs a tuple, $\boldsymbol{x} = (\bar{v}_{l0}, \bar{v}_{r0}, ..., \bar{v}_{l31}, \bar{v}_{r31}) \in [-1,1]^{32}$, where $\bar{v}_{l_i}$ and $\bar{v}_{r_i}$ are respectively the left and right wheel velocities for the i^{th} sensing state. The output represents one of four actions (see Table 1): spin clockwise (CW), move forward, turn left while moving forward, and turn left while very slowly moving forward. This allows the BCRs to establish a distributed formation in which they traverse the boundary counter-clockwise, while avoiding collisions.

Herd Agent Design: HAs are simulated using e-pucks [2], which are miniature, differential wheeled robots of mass ~ 150 g. For HA i, potential fields [4] are generated by other HAs within 1 m radius, BCRs within 3 m radius, and a random attraction point, unique per HA, and sampled every 15 s from a $20\,\text{m} \times 20\,\text{m}$

Table 1. BCR controller lookup ('*' = 'whichever state that sensor takes')

S_{near}	S_{far}	S_{wide}	$\bar{v}_l, \bar{v}_r$	Belief of Situation and Intended Behaviour
H	*	*	0.2,-0.2	Spins CW to avoid collision with HA in path
BC	*	*	0.2,-0.2	Spins CW to avoid collision with other BCR in path
G	*	BC	0.2,-0.2	Spins CW to avoid potential collision with BCR in wide range
G	*	0	1,1	Detects green disc and no obstacles in wide range, moves forward at max speed
0	G	*	0.2,-0.2	Inside the boundary but not facing a short exit path, spins CW to find a closer green disc
0	0	BC	0.05,1	While orbiting detects BCR in wide range, turns left while very slowly moving forward until other BCR moves out of detection range
0	0	0	0.5,1	Outside the boundary, turns left & forward until a green disc is detected



(a) (b)

(c) (d)

Fig. 2. (a) Histogram of HAs remaining after 90 s in SC1 and SC2, (b) Average escapes over time in SC1 and SC2 (20 trials each). Effect of maximum HA speed (2c, SC4) and number of BCRs (2d, SC5) on total HAs remaining after 90 s (10 trials per setting).

region in the centre. Let $\boldsymbol{x_{h_j}}$ be the position vector of HA j, $\boldsymbol{x_{bc_k}}$ be the position vector of BCR k, and $\boldsymbol{x_{rand}}$ be the position vector of the random point, and thus for HA i, define $\boldsymbol{r_{h_j}} = \boldsymbol{x_{h_j}} - \boldsymbol{x_{h_i}}$, $\boldsymbol{r_{bc_k}} = \boldsymbol{x_{bc_k}} - \boldsymbol{x_{h_i}}$, and $\boldsymbol{r_{rand}} = \boldsymbol{x_{rand}} - \boldsymbol{x_{h_i}}$. Thus, for HA i, defining distances $r_{h_j} = |\boldsymbol{r_{h_j}}|, r_{bc_k} = |\boldsymbol{r_{bc_k}}|$, and $r_{rand} = |\boldsymbol{r_{rand}}|$, the field due to repulsion from other HAs is $U_h = k_h \sum_{j \neq i} \frac{1}{r_{h_j}}$, the field due to repulsion from BCRs is $U_{bc} = k_{bc} \sum_k \frac{1}{r_{bc_k}}$, and the field due to the random point of attraction is $U_{rand} = k_{rand} r_{rand}$, where k_h, k_{bc}, and k_{rand} are the weights for HA repulsion, BCR repulsion, and random motion, set to 1.5, 3, and 1.5, respectively in the default scenario. HA i is thus subject to force $\boldsymbol{F}$ equal to the negative gradient w.r.t. $\boldsymbol{x_{h_i}}$ of the total potential field (adapted from [4]):

$$\boldsymbol{F} = -\nabla(U_h + U_{bc} + U_{rand}) = -\left(k_h \sum_{j \neq i} \frac{\boldsymbol{r_{h_j}}}{r_{h_j}^3} + k_{bc} \sum_k \frac{\boldsymbol{r_{bc_k}}}{r_{bc_k}^3} - k_{rand} \frac{\boldsymbol{r_{rand}}}{r_{rand}} \right)$$

The velocity at time step t, $\boldsymbol{v_t}$, is computed as $\boldsymbol{v_t} = \frac{\boldsymbol{F} - \nu \boldsymbol{v_{t-1}}}{m} \Delta t + \boldsymbol{v_{t-1}}$, where $\nu = 0.01$ is viscous friction, $\boldsymbol{v_{t-1}}$ is the previous velocity, initialised to $\begin{bmatrix}0 \ 0\end{bmatrix}^T$, m is the e-puck mass, and Δt is the simulation time step. The velocity of each HA is clipped to $[-M, M]$, where M is the maximum speed in m/s.

3 Results and Discussions

Five scenarios are considered, referred to as SC1 to SC5, respectively (for a representative selection of video clips, see [3]).

SC1 analyses HA retention under the default setup with $M \approx 0.27\,\text{m/s}$. SC2 differs from SC1 only in that no BCRs are present. This comparison provides a baseline for the efficacy of the design. Fig. 2(a) shows a marked separation in the distributions. An average of 2.55 ($\sigma = 1.32$) HAs are retained after 90 s in SC2 versus 8.40 ($\sigma = 1.19$) retained in SC1. Fig. 2(b) shows the impact of the BCRs on the rate of HA escapes: no escapes occur before $\sim 20\,\mathrm{s}$ in either scenario (the approximate time for HAs to reach the boundary), and a distinct plateau after $\sim 25\,\mathrm{s}$ for SC1 (presumably, at that moment the BCRs achieved formation).

SC3 tests the robustness with respect to irregular barrier polygons. The selected polygon (Fig. 1(c)) has the most complex shape for which the design proved effective given the local nature of the sensing strategy. The area of the polygon is kept approximately equal to that of SC1 (i.e. $31.57\,\mathrm{m}^2$). A mean of 7.50 ($\sigma = 1.70$) HAs are retained over 20 trials. This is similar to the performance seen in SC1, with an expected slight decrease in retained HAs possibly due to the increased initial proximity of HAs to some sections of the barrier.

In SC4, the maximum HA speed is varied through $M \in [0, 0.80]$ m/s in 18 discrete steps, under otherwise the default setup. Fig. 2(c) shows that once a speed of $M \approx 0.31\,\mathrm{m/s}$ is surpassed, HA retention falls below the 8.40 of SC1. Furthermore, at $M \approx 0.48\,\mathrm{m/s}$, more than 70% of the HAs escape, linearly increasing to $\sim 100\%$ at $M \approx 0.80\,\mathrm{m/s}$, rendering the design ineffective. The performance could be improved by increasing BCR speed, thus more promptly establishing a barrier, however this could result in erratic behaviour as BCR reaction time becomes a restricting factor.

In SC5, the number of BCRs, m, is varied from 0 to 10. As shown in Fig. 2(d), increasing m from 0 to 7 yields improved performance, but with diminishing returns: significant improvements occur for $m = 1 \rightarrow 4$ ($\nabla > 1$ HA/BCR), with only marginal improvements ($\nabla < 1$ HA/BCR) for $m = 5 \rightarrow 7$. The proposed design ($m = 4$) therefore offers a good trade-off between cost and performance. Beyond $m = 7$, HA retention decreases, as long chains of slowly moving, closely packed BCRs temporarily result in large gaps in barrier coverage.

This paper proposed a simple solution to multi-robot barrier coverage, which does not require the barrier coverage robots to communicate, or store information during run-time. The solution was shown to perform robustly in a range of simulation scenarios. Future work will test more realistic conditions including environments with obstacles and porting the solutions to real robots.

References

1. Gauci, M., Chen, J., Li, W., Dodd, T.J., Groß, R.: Self-organized aggregation without computation. The Int. J. of Robot. Res. **33**(8), 1145–1161 (2014)
2. Gonçalves et al.: The e-puck, a robot designed for education in engineering. 9th Conference on Autonomous Robot Systems and Competitions **1**, 59–65 (2009)
3. Green, T., Kamel, K., Li, S., Shinn, C., Toscano, P., Wang, X., Ye, Y., Groß, R.: A minimalist solution to the multi-robot barrier coverage problem (supplementary video material) (2021), `http://doi.org/10.15131/shef.data.15082653`
4. Howard, A., Matarić, M.J., Sukhatme, G.S.: Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In: Distributed Autonomous Robotic Systems 5, pp. 299–308. Springer (2002)
5. Lien, J.M., Bayazit, O.B., Sowell, R.T., Rodriguez, S., Amato, N.M.: Shepherding behaviors. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004. vol. 4, pp. 4159–4164. IEEE (2004)

6. Nguyen, T.M., Li, X., Xie, L.: Barrier coverage by heterogeneous sensor network with input saturation. In: 11th Asian Control Conference. pp. 1719–1724 (2017)
7. Strömbom, D., Mann, R.P., Wilson, A.M., Hailes, S., Morton, A.J., Sumpter, D.J., King, A.J.: Solving the shepherding problem: Heuristics for herding autonomous, interacting agents. Journal of the Royal Society Interface **11**(100), 20140719 (2014)
8. Vaughan, R., Sumpter, N., Henderson, J., Frost, A., Cameron, S.: Experiments in automatic flock control. Robotics and Autonomous Systems **31**(1-2), 109–117 (2000)
9. Zahugi, E.M.H., Shanta, M.M., Prasad, T.: Design of multi-robot system for cleaning up marine oil spill. Int. J. of Advanced Information Technology **2**(4), 33–43 (2012)