

Optimal Multi-Agent Search and Rescue Using Potential Field Theory

John R. Cooper*

NASA Langley Research Center, Hampton, VA, 23681

This paper presents an algorithm for efficient search and rescue using a multi-agent system of vehicles. The algorithm uses an artificial potential field combined with a time-varying reward function for visiting various points within the search area. The reward function is used to weight the attractiveness of these points in the potential field, and collision avoidance terms are used to repel vehicles from each other, which has the additional effect of reducing duplication of searching efforts. The algorithm generates velocity commands in real-time based on communication with the other vehicles. This framework allows vehicles to react in a dynamic environment, which is a significant advantage to simply following a-priori defined trajectories. Simulation results are presented to demonstrate the ability of the algorithm to cover the search area effectively. The algorithm is also compared to an exhaustive lawn-mower search pattern. This comparison is done via a Monte Carlo simulation with randomized target initial conditions and trajectories. The time to find the target improved by 16 and 30% in the mean and median, respectively. Additionally, this paper presents a method for analyzing the upper bound for time to find a target under the potential field guidance algorithm assuming a radially expanding search area.

I. Nomenclature

V_s	=	search component of potential
i, k	=	vehicle indices
W_s	=	search weight
j	=	Gaussian radial basis function index
m	=	number of Gaussian radial basis functions
r	=	reward
ξ	=	position
p	=	Gaussian radial basis function center
δ	=	Gaussian radial basis function width
K_{r+}, K_{r-}	=	reward rate gains
α	=	boolean that indicates whether a point is observed
V_c	=	collision avoidance component of potential
W_c	=	collision avoidance weight
n	=	number of vehicles
V	=	total potential
v_{max}	=	maximum speed
v	=	velocity command
K	=	gradient descent step size
μ	=	measurement radius
r_0	=	initial reward
A_s	=	search area
ρ	=	search radius
v_s	=	target maximum speed
A_c	=	covered area
ρ_0	=	initial search radius

*Research Aerospace Technologist, Navigation, Guidance, and Control Systems, Autonomous Integrated Systems Research Branch, Mail Stop 233.

II. Introduction

This paper presents an algorithm for optimal area coverage by a system of multiple aircraft applied to a search and rescue mission. The main contribution herein is the development of a potential field guidance algorithm constructed to efficiently guide each vehicle towards an unobserved portion of the search area while avoiding collisions with other vehicles in the group and minimizing duplicated efforts. The guidance algorithm runs in real-time such that vehicles may react to a changing environment. This characteristic has obvious advantages over a-priori trajectory planning methods. The target of the search and rescue mission may be moving, and thus it is insufficient for vehicle positions to converge to some steady state. This paper introduces a time-varying reward function associated with visiting certain locations within the search area which is then used to optimize vehicle guidance commands. Simulation results are presented to showcase the efficacy of the algorithm. Simulation results are compared between this guidance algorithm and a baseline exhaustive search trajectory.

Distributed control of multi-agent systems has been an area of continued research interest for many years [1–3]. This paper specifically addresses the search and rescue scenario, but distributed control of multiple aerial vehicles can be applied to many other domains including cartography, reconnaissance, and package delivery. Some of the earliest work on distributed multi-agent control involved the consensus problem [4, 5], where each agent updates its own estimate of some value based on its interactions with other agents in the system. Consensus is achieved when all agents' estimations converge to the same value. In [6–10], consensus algorithms were developed for various constraints on information flow. A more complex problem in multi-agent control is formation control, where rather than converge to a single value, each agent's state may converge to a unique value to optimize a global objective function [11–13]. Formation control finds applications in formation flying, cooperative transportation, sensor networks, combat intelligence, surveillance, and reconnaissance [14–16]. Other extensions to multi-agent control include game-theoretic approaches [17–21] and adaptive control methods [22–24].

In this paper, a potential field is constructed for each agent from two competing objectives: searching and collision avoidance. These goals are analogous to two of the three basic rules of flocking developed in [11] and presented in [25]:

- 1) Flock centering - ensure that all agents remain close together
- 2) Obstacle avoidance - ensure that no agents collide with each other
- 3) Velocity matching - causes agents to move in the same direction and at the same speed

The obstacle avoidance rule is readily adapted. However, rather than drive vehicles towards the flock center, they are driven towards points in the search space which represent (recently) unexplored locations. Additionally, velocity matching does not need to be considered for the distributed search and rescue mission. The control input is computed by finding the direction of steepest descent of the potential field. Potential field gradient algorithms were developed for flocking in the body of work comprised of [25–29]. [28] introduced a differentiable, nonnegative, and radially unbounded potential field to represent the distance between two agents. Potential field methods were previously applied to multi-rotor aircraft guidance in large flocks in [30].

A similar framework to that of this paper is the coverage control problem first discussed in [31]. There, it's assumed that there are enough vehicles to cover the entire space. Coverage control was extended to systems with insufficient sensor range to measure all points of interest in [32, 33] from a data fusion perspective. In [34], a method for a-priori vehicle trajectory planning was developed for this problem. In [21], the coverage control problem was addressed using graph theory (i.e., discrete state and action spaces). The present paper extends this research area further by presenting a real-time, continuous, deterministic guidance law that allows vehicles to react based on information about each other and their environment as opposed to simply following a pre-planned trajectory.

The paper is organized as follows: Section III describes the potential field, section IV presents the guidance algorithm, section V gives example simulation results, section VI compares the potential field guidance algorithm to a baseline exhaustive search trajectory method, section VII gives results for analyzing the time to find a target in a search and rescue mission, and conclusions are drawn in section VIII.

III. Potential Field Construction

The multi-agent search problem is composed of two objectives: efficiently searching the space and avoiding collisions with other vehicles. The component of the potential field that drives each vehicle to efficiently search the space is defined using a set of Gaussian radial basis functions (GRBFs). The GRBF centers are distributed over the search area. A simple design choice is create a grid of the search area and use the grid vertices as the GRBF centers. Another approach may be to use a higher density of GRBFs in regions of greater interest. The search component of the

potential of vehicle i is then

$$V_{s_i} = -\frac{W_s}{2} \sum_{j=1}^m r_j \exp\left(-\frac{\|\xi_i - p_j\|^2}{\delta}\right) \quad (1)$$

where W_s is the weight prescribed to the searching goal, j is an index for each GRBF center, m is the number of GRBFs in the search area, r_j is a time-varying reward for being within measurement range of the center of GRBF j , p_j is the position of the center of GRBF j , ξ_i is the position of vehicle i , and δ is the GRBF width.

The purpose of the reward variables, r_j are to increase the weights in the overall potential for areas of the search space which haven't been observed recently and decrease the weights for locations which have been observed recently. Each r_j is updated according to

$$\dot{r}_j = K_{r+}(1 - \alpha_j) - K_{r-}\alpha_j r_j \quad (2)$$

where K_{r+} and K_{r-} are reward rate gains and α_j is given by

$$\alpha_j = \begin{cases} 0 & \text{if } \min_k \|\xi_k - p_j\| > \mu \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

where $k = 1, \dots, n$ is the vehicle index and μ is the vehicle measurement range. (3) means that α_j is 0 if no vehicle is within measurement range of GRBF j and 1 otherwise. In conjunction with (2), this means that the reward for visiting p_j increases at a rate of K_{r+} until a vehicle goes there. Once any vehicle is within measurement range of p_j , this reward begins to decrease as a first-order linear system with a steady-state value of 0 and bandwidth K_{r-} .

The component of the potential field that drives each vehicle away from others is defined using the inverse of the distance between the two agents and is given by

$$V_{c_i} = \frac{W_c}{2} \sum_{k=1, k \neq i}^n \|\xi_i - \xi_k\|^{-2} \quad (4)$$

where W_c is the weight prescribed to the collision avoidance goal, and n is the number of vehicles involved in the search. As the distance between agents approaches 0, V_{c_i} approaches infinity. Thus, the collision avoidance component will always dominate the potential when two vehicles get close enough to each other.

The collision avoidance term of the potential field may also be viewed as a deterrent for each agent against duplicating the efforts of others. As a result of (4), potential will be higher in areas which other vehicles are closer to. In seeking low potential, vehicles will be driven away from areas of the search space that can be more efficiently searched by others.

IV. Guidance Algorithm

Each vehicle's overall goal is to minimize its own potential, $V_i = V_{s_i} + V_{c_i}$. Therefore, a velocity command, v_i , is chosen to be in the direction of steepest descent of the potential function and saturated by a maximum speed, v_{max} ,

$$v_i = \begin{cases} -K \frac{\partial V_i}{\partial \xi_i} & \text{if } \left\| K \frac{\partial V_i}{\partial \xi_i} \right\| \leq v_{max} \\ -v_{max} \frac{\frac{\partial V_i}{\partial \xi_i}}{\left\| \frac{\partial V_i}{\partial \xi_i} \right\|} & \text{otherwise} \end{cases} \quad (5)$$

where K is a control gain. The partial derivative of potential with respect to the agent's own position is used instead of considering all agents in the system because each agent only has authority over its own position and velocity. Thus, the guidance algorithm is decentralized.

To calculate the partial derivative in (5), we express V_i in terms of (1) and (4) as

$$V_i = -\frac{W_s}{2} \sum_{j=1}^m r_j \exp\left(-\frac{\|\xi_i - p_j\|^2}{\delta}\right) + \frac{W_c}{2} \sum_{k=1, k \neq i}^n \|\xi_i - \xi_k\|^{-2} \quad (6)$$

Then the partial derivative with respect to ξ_i is

$$\frac{\partial V_i}{\partial \xi_i} = \frac{W_s}{\delta} \sum_{j=1}^m r_j \exp\left(-\frac{\|\xi_i - p_j\|^2}{\delta}\right) (\xi_i - p_j) - W_c \sum_{k=1, k \neq i}^n \frac{\xi_i - \xi_k}{\|\xi_i - \xi_k\|^4} \quad (7)$$

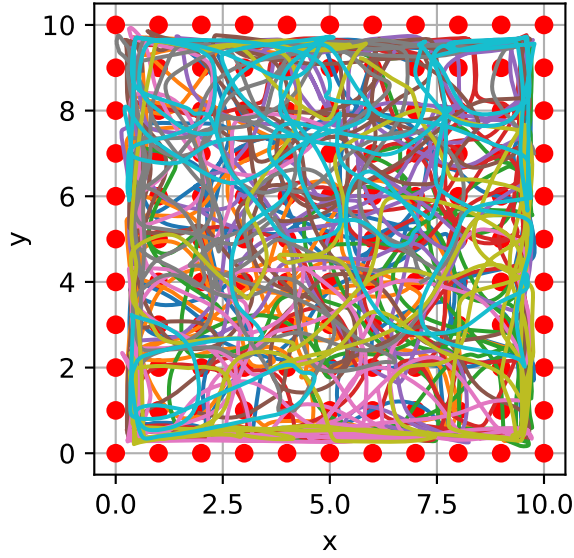


Fig. 1 Example simulation vehicle trajectories

The guidance algorithm operates similarly to stochastic gradient descent (SGD) [35], which is widely used to efficiently optimize cost functions for fitting models to large amounts of data. In SGD, each gradient descent step is done using a randomly sampled subset of the data to be fit. Because of this, the actual function being minimized changes by a small amount each step. Despite this change, the gradient direction from the previous step is often very close to optimal for the current step. In the guidance algorithm presented here, gradient descent of the potential field happens continuously, and the potential field also changes continuously. Over the course of one infinitesimal time-step, the change in potential the potential field is small, and thus the direction of the velocity command for any vehicle at the previous time-instant is very close to the direction that would have been prescribed by the current gradient. By continuously making these small adjustments, we ensure that vehicle trajectories over the course of a mission remain close to optimal.

V. Example Simulation

This section presents the results of an example simulation of 10 vehicles with randomized initial conditions in a 10×10 search area grid. The grid is evenly spaced in both axes, resulting in 222 vertices used as GRBF centers. Each vehicle has a measurement radius, μ of 1 and a maximum speed, v_{max} , of 2. Figure 1 shows the trajectories of all vehicles over the search area. The GRBF centers are represented as red circles. Figure 2 shows how the potential values and reward values change throughout the simulation. Empirically, the average reward and average potential values converge to a stable, steady-state value. Since neither the potential values nor the GRBF rewards grow unbounded, the search area is well covered for the duration of the simulation. The parameters for this simulation are given in table 1.

VI. Baseline Comparison

An exhaustive lawn mower search pattern is considered as a baseline multi-agent guidance law to compare against. The same environment and parameters from section V are used. Based on the GRBF distribution, the minimum distance between GRBFs is 1 which is also the vehicle measurement radius value. So if a vehicle coincides with a GRBF center, it can only observe that GRBF. The lawn mower pattern trajectories are generated as cubic splines that pass through each GRBF center. Cubic splines are used because the vehicle dynamics are assumed to perfectly track velocity commands, and the cubic spline trajectories offer a continuous velocity. The trajectories are scaled temporally with the vehicles' maximum speed. Once the end of the trajectory is reached, it is traversed again in reverse. This process repeats until the target is found. A system of four agents was used in the test scenario. The total search areas was divided into quadrants, and each agent is initialized in the northwest corner of its quadrant.

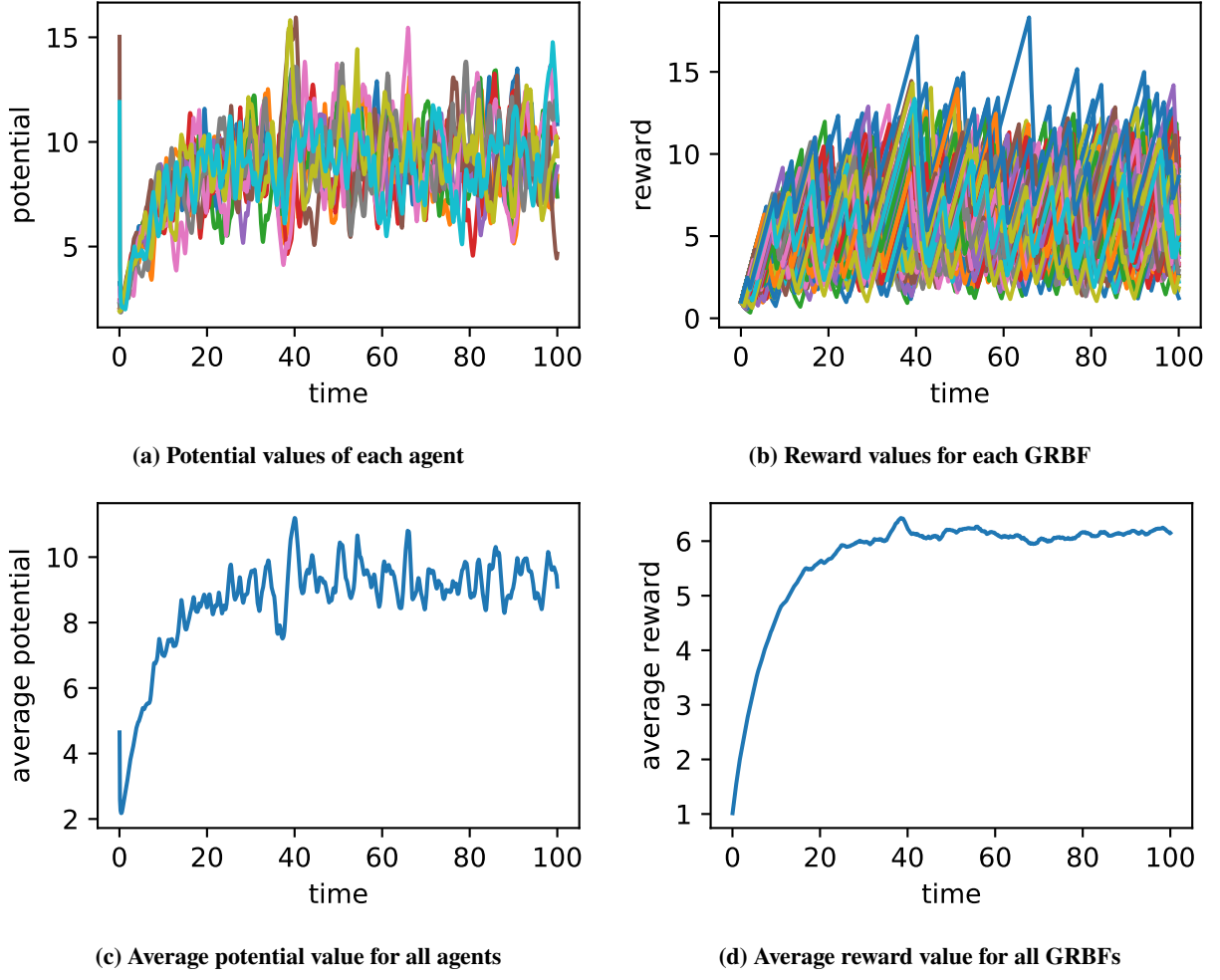
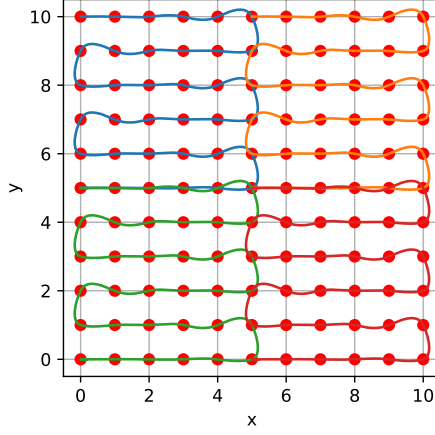


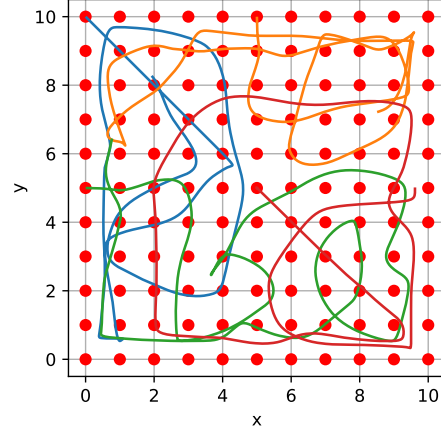
Fig. 2 Example simulation results

Table 1 Example simulation parameters

Parameter	Value
time-step	1/50
r_0	1
K_{r+}	1
K_{r-}	0.5
W_c	2
W_s	10
δ	2
K	5
μ	1

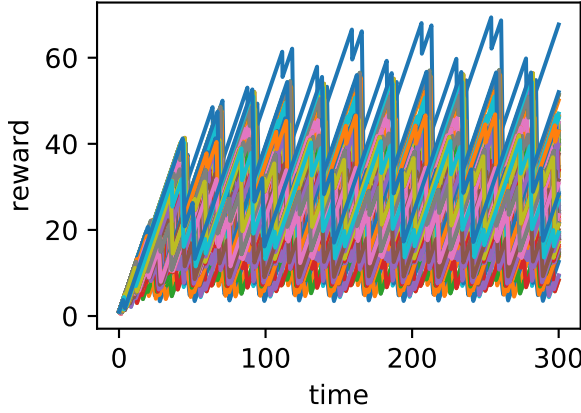


(a) Baseline lawn mower pattern trajectories

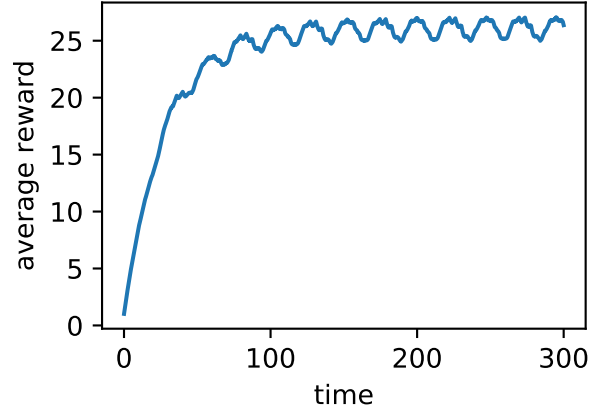


(b) Corresponding potential field-based trajectories

Fig. 3 Test scenario trajectories



(a) Reward values for each GRBF under lawn mower search pattern



(b) Average reward value for all GRBFs under lawn mower search pattern

Fig. 4 Reward values under lawn mower search pattern

Figure 3a shows the lawn mower pattern trajectories, and figure 3b shows the corresponding potential-field based trajectories for the first 24 seconds, given the same initial conditions.

Figure 4a shows the GRBF reward values under the lawn mower search pattern, and figure 4b shows the corresponding average over all GRBFs. The maximum GRBF reward reaches a value of about 65 before that GRBF is observed, and the average signal converges to about 25 with a small oscillation.

Figure 5 shows potential values and GRBF rewards for the potential field algorithm with the same initial conditions as the lawn mower search pattern. Since the average potential and reward converge to constant values without unbounded growth, full coverage of the search area can be inferred. In comparing figures 5b and 5d to figure 4a and 4b respectively, note that the average GRBF reward under the potential field algorithm converges to about 23 (lower than the corresponding value of 25 under the lawn mower pattern). This indicates that there is less time between observations under the potential field algorithm than under the lawn mower pattern since the reward does not grow as large before an observation occurs.

Also note that maximum individual reward values shown in figure 5b reach a value of about 85. While this is higher than the maximum of around 65 under the lawn mower pattern, note that there is a larger margin between these few outlier GRBF reward values and the rest of rewards. This phenomenon occurs because the potential field algorithm

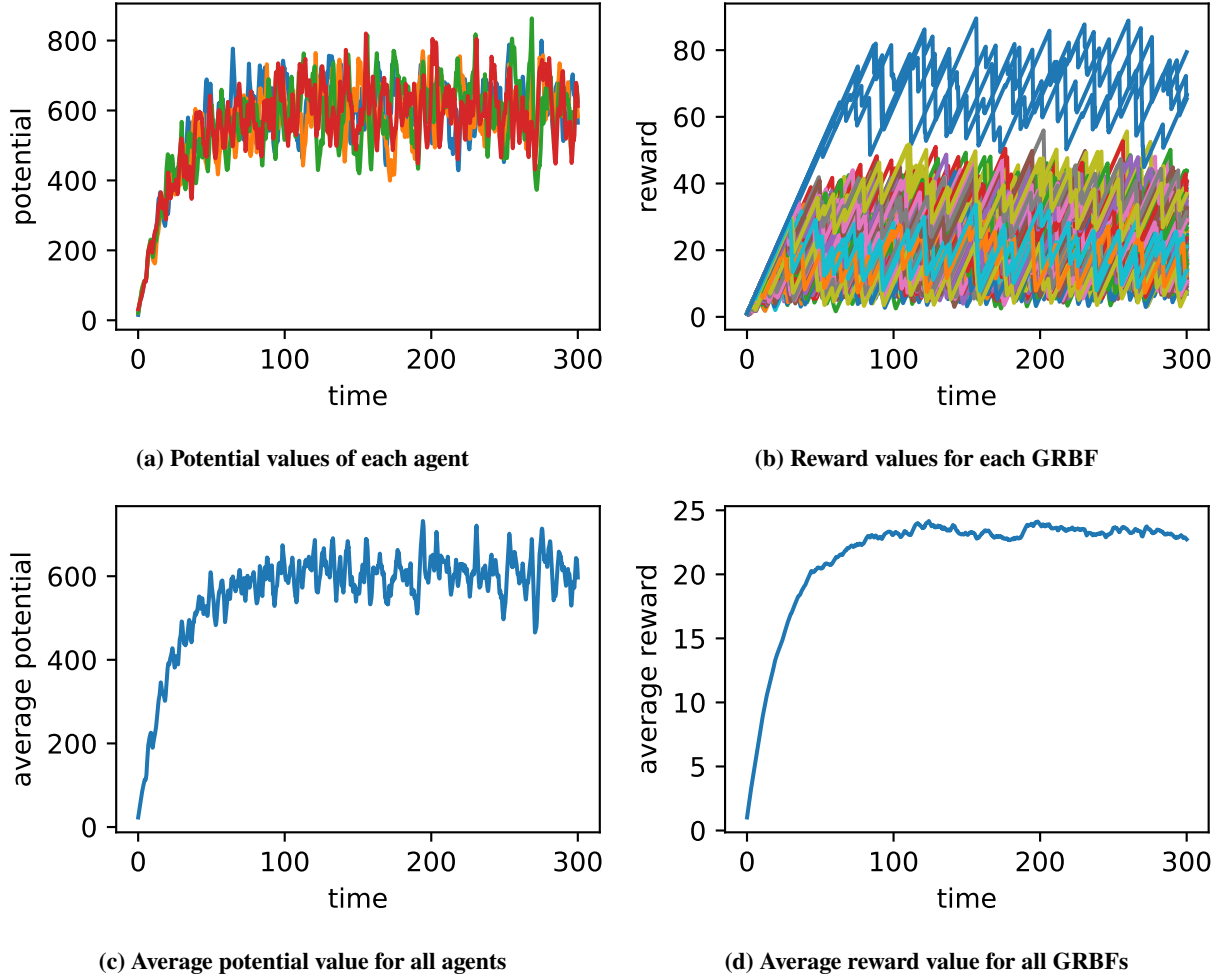


Fig. 5 Example simulation results

naturally explores the edges of the search space less than the center regions. Since center regions have more overlapping GRBFs, their attractive pull in the potential field is higher than the edge regions. This behavior is actually beneficial to search and rescue missions since the search area should be designed with the highest probability of finding the target at the center. The potential field algorithm allows the space to be searched in a way that puts more effort into places where the target is more likely to be whereas an exhaustive search weights all regions equally, which results in wasted efforts and lower efficiency.

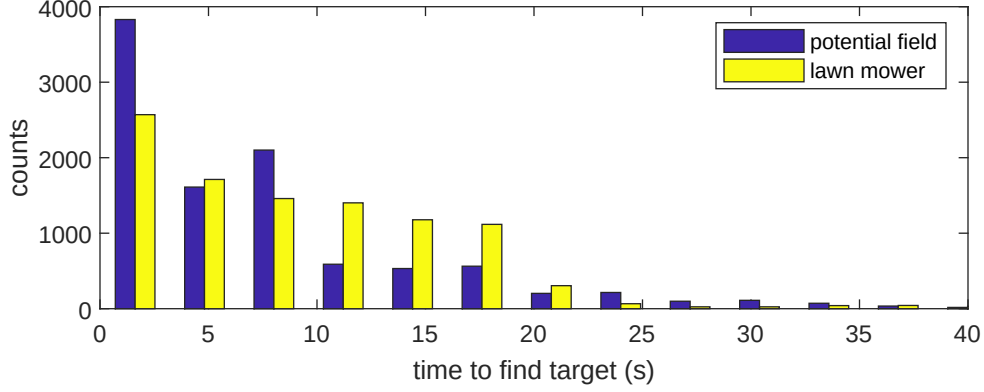
The benefits of the exhaustive search are maintained in the potential field algorithm by the time-varying reward function. As a region goes unsearched, the probability of finding the target there (and thus, the reward for searching there) increases until a vehicle is drawn there. Thus the potential field guidance algorithm eliminates the inefficiency of the exhaustive search while balancing the ability to search all regions.

A Monte Carlo simulation was carried out by randomizing target initial position within the 10×10 grid described in section V. At each time step, the target takes a positional step according to a zero-mean normal distribution with standard deviation of 2 for both x and y directions. Each target initial condition and trajectory was run for both the potential field guidance algorithm and the lawn mower search pattern. The simulation ended when the target fell within the measurement radius of a vehicle. Thus the simulated time for each run is the time to find the target. The Monte Carlo simulation contained 10,000 runs. Table 2 shows the mean and median time to find values for each guidance algorithm from the Monte Carlo simulation. The potential field algorithm exhibits a 16% improvement in the mean, and a 30% improvement in the median over the lawn mower pattern.

Figure 6 shows histograms for the time to find of each guidance algorithm in the Monte Carlo simulation. The lawn

Table 2 Time to find statistics from Monte Carlo simulation

	Potential Field	Lawn Mower
Mean	7.51 s	8.98 s
Median	5.72 s	8.23 s

**Fig. 6 Histogram of Monte Carlo results**

mower pattern times to find are more widely distributed between 0 and 20 seconds, while the potential field algorithm results were highly concentrated in the 0 to 10 second range.

VII. Radially Expanding Search Area

Consider a scenario where the target has some a-priori known maximum speed. In this situation, it follows that the search area should grow to accommodate the worst-case scenario of the target moving in a straight line at maximum speed, in an unknown direction. This translates to a circular search area with a radius that grows at a rate equal to the target's maximum speed. The search area growth rate can then be expressed as

$$\dot{A}_s = 2\pi\rho v_s \quad (8)$$

where A_s is the search area, ρ is the search radius, and v_s is the maximum target speed.

Conversely, the maximum rate of area coverage, $\dot{A}_c$, by the multi-agent system is

$$\dot{A}_c = 2\mu n v_{max} \quad (9)$$

This information can be used to establish a bound on the maximum time to find a target before the search area grows too large. As long as $\dot{A}_c \geq \dot{A}_s$, it is possible for the vehicles to cover the entire search space. Substituting (8) and (9) into this inequality yields

$$2\mu n v_{max} \geq 2\pi\rho v_s \quad (10)$$

as the condition for total coverage. (10) can be rearranged to give the maximum search area radius such that total coverage is possible:

$$\rho \leq \frac{\mu n v_{max}}{\pi v_s} \quad (11)$$

Note that ρ evolves over time according to

$$\rho = \rho_0 + v_s t \quad (12)$$

where ρ_0 is the initial search area radius, and t is time. Substituting (12) into (11) and solving for t yields

$$t \leq \frac{\mu n v_{max}}{\pi v_s^2} - \frac{\rho_0}{v_s} \quad (13)$$

The right-hand side of (13) represents the upper bound for time to find the target. Up until this time, the multi-agent system is able to completely cover the search area. Note that this bound is optimistic because the vehicles are not likely to travel at their maximum speed at all times. It is also pessimistic because the target is not likely to travel at its maximum speed in a straight line at all times. This bound can be used to select the number and types of vehicles used for a search depending on the time since the target was last seen.

VIII. Conclusion

The potential field guidance algorithm presented shows some key improvements when compared to an exhaustive search method. The Monte Carlo simulation performed for this paper shows an improvement in time to find a target in a multi-agent, cooperative search and rescue mission of 16-30%. The time-varying reward function provides a way for vehicles to dynamically weight the importance of searching certain regions, which keeps the majority of the search focused on the locations where the target is most likely to be found.

Furthermore, the potential field method offers a mechanism by which to analyze the maximum time to find a target via the radially expanding search area approach. This analysis can be used to help select the number and type of vehicles to use for the mission.

Overall, the potential field guidance algorithm has the advantage of allowing for decentralized, real-time control of a multi-agent system in a dynamic environment. This is in contrast to a-priori designed trajectories. In the a-priori trajectory case, if a vehicle fails to follow its trajectory, search efficiency is lost, and a collision may even occur. Under the potential field algorithm, when a vehicle fails, the remaining vehicles can dynamically distribute the workload to maintain efficiency.

Acknowledgments

This work was performed as part of the Autonomy Teaming & TRAjectories for Complex Trusted Operational Reliability (ATTRACTOR) research thrust under NASA's Convergent Aeronautics Solutions project. Thanks to Danette Allen and Natalia Alexandrov for steering the direction of this research.

References

- [1] Bullo, F., Cortés, J., and Martínez, S., "Distributed control of robotic networks: a mathematical approach to motion coordination algorithms," Tech. rep., PRINCETON UNIV NJ, 2008.
- [2] Murray, R. M., "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 129, No. 5, 2007, pp. 571–583.
- [3] Cao, Y., Yu, W., Ren, W., and Chen, G., "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, Vol. 9, No. 1, 2013, pp. 427–438.
- [4] Tsitsiklis, J. N., "Problems in decentralized decision making and computation." Tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND DECISION SYSTEMS, 1984.
- [5] Tsitsiklis, J., Bertsekas, D., and Athans, M., "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE transactions on automatic control*, Vol. 31, No. 9, 1986, pp. 803–812.
- [6] Jadbabaie, A., Lin, J., and Morse, A. S., "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on automatic control*, Vol. 48, No. 6, 2003, pp. 988–1001.
- [7] Fax, J. A., and Murray, R. M., "Information flow and cooperative control of vehicle formations," *IEEE transactions on automatic control*, Vol. 49, No. 9, 2004, pp. 1465–1476.
- [8] Olfati-Saber, R., and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, Vol. 49, No. 9, 2004, pp. 1520–1533.
- [9] Ren, W., and Beard, R. W., "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on automatic control*, Vol. 50, No. 5, 2005, pp. 655–661.
- [10] Moreau, L., "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on automatic control*, Vol. 50, No. 2, 2005, pp. 169–182.

- [11] Reynolds, C. W., "Flocks, Herds and Schools: A Distributed Behavioral Model," *ACM SIGGRAPH Computer Graphics*, Vol. 21, No. 4, 1987, pp. 25–34.
- [12] Leonard, N. E., and Fiorelli, E., "Virtual leaders, artificial potentials and coordinated control of groups," *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, Vol. 3, IEEE, 2001, pp. 2968–2973.
- [13] Ogren, P., Fiorelli, E., and Leonard, N. E., "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic control*, Vol. 49, No. 8, 2004, pp. 1292–1302.
- [14] Weisbin, C., Matthies, L., Rodriguez, G., Blitch, J., Krotkov, E., Lavery, D., and Shoemaker, C., "Miniature robots for space and military missions," *AIAA Defense and Civil Space Programs Conference and Exhibit*, 1998, p. 5309.
- [15] Rybski, P. E., Papanikolopoulos, N. P., Stoeter, S. A., Krantz, D. G., Yesin, K. B., Gini, M., Voyles, R., Hougen, D. F., Nelson, B., and Erickson, M. D., "Enlisting rangers and scouts for reconnaissance and surveillance," *IEEE Robotics & Automation Magazine*, Vol. 7, No. 4, 2000, pp. 14–24.
- [16] Curtin, T. B., Bellingham, J. G., Catipovic, J., and Webb, D., "Autonomous oceanographic sampling networks," *Oceanography*, Vol. 6, No. 3, 1993, pp. 86–94.
- [17] Marden, J. R., Arslan, G., and Shamma, J. S., "Cooperative control and potential games," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 39, No. 6, 2009, pp. 1393–1407.
- [18] Marden, J. R., Young, H. P., Arslan, G., and Shamma, J. S., "Payoff-based dynamics for multiplayer weakly acyclic games," *SIAM Journal on Control and Optimization*, Vol. 48, No. 1, 2009, pp. 373–396.
- [19] Chasparis, G. C., and Shamma, J. S., "Distributed dynamic reinforcement of efficient outcomes in multiagent coordination and network formation," *Dynamic Games and Applications*, Vol. 2, No. 1, 2012, pp. 18–50.
- [20] Marden, J. R., and Shamma, J. S., "Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation," *Games and Economic Behavior*, Vol. 75, No. 2, 2012, pp. 788–808.
- [21] Lim, Y., and Shamma, J. S., "Robustness of stochastic stability in game theoretic learning," *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 6145–6150.
- [22] Luo, J., Cooper, J., Cao, C., and Pham, K., "Cooperative adaptive control of a two-agent system," *International Journal of Control*, Vol. 86, No. 1, 2013, pp. 127–138.
- [23] Luo, J., "Adaptive Control for Distributed Multi-Agent Coordination," 2013.
- [24] Luo, J., and Cao, C., "Flocking for multi-agent systems with unknown nonlinear time-varying uncertainties under a fixed undirected graph," *International Journal of Control*, Vol. 88, No. 5, 2015, pp. 1051–1062.
- [25] Olfati-Saber, R., "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," *IEEE Transactions on Automatic Control*, Vol. 51, No. 3, 2006, pp. 401–420.
- [26] Tanner, H. G., Jadbabaie, A., and Pappas, G. J., "Stable Flocking of Mobile Agents, Part I: Fixed Topology," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Vol. 2, IEEE, 2003, pp. 2010–2015.
- [27] Tanner, H. G., Jadbabaie, A., and Pappas, G. J., "Stable Flocking of Mobile Agents Part II: Dynamic Topology," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Vol. 2, IEEE, 2003, pp. 2016–2021.
- [28] Tanner, H. G., Jadbabaie, A., and Pappas, G. J., "Flocking in Fixed and Switching Networks," *IEEE Transactions on Automatic Control*, Vol. 52, No. 5, 2007, pp. 863–868.
- [29] Su, H., Wang, X., and Lin, Z., "Flocking of Multi-Agents with a Virtual Leader," *IEEE Transactions on Automatic Control*, Vol. 54, No. 2, 2009, pp. 293–307.
- [30] Cooper, J. R., and Rothhaar, P. M., "Link!: Potential Field Guidance Algorithm for In-Flight Linking of Multi-Rotor Aircraft," *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 3604.
- [31] Cortes, J., Martinez, S., Karatas, T., and Bullo, F., "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, Vol. 20, No. 2, 2004, pp. 243–255.
- [32] Gupta, V., Chung, T. H., Hassibi, B., and Murray, R. M., "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, Vol. 42, No. 2, 2006, pp. 251–260.

- [33] Tiwari, A., Jun, M., Jeffcoat, D. E., and Murray, R. M., “Analysis of dynamic sensor coverage problem using Kalman filters for estimation,” *Proceedings of the 16th IFAC world congress*, 2005.
- [34] Ousingsawat, J., and Campbell, M., “Establishing trajectories for multi-vehicle reconnaissance,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 5224.
- [35] Robbins, H., and Monro, S., “A stochastic approximation method,” *The annals of mathematical statistics*, 1951, pp. 400–407.