2017 IEEE Conference on Control Technology and Applications (CCTA)
August 27-30, 2017. Kohala Coast, Hawai'i, USA

Dynamic Grouping of Cooperating Vehicles using a Receding Horizon Controller for Ground Target Search and Track Missions

Cameron K. Peterson

Abstract— **Teams of unmanned vehicles are capable of accomplishing a wide variety of mission objectives, such as searching for and tracking targets. In this paper, a receding horizon control is utilized with information based reward measures to accomplish these two competing mission objectives. This approach for cooperatively searching and tracking has proven to be effective in past work. However, it is not generally scalable for large numbers of vehicles due to the computational expense required when generating joint path decisions. This paper proposes a method to dynamically group vehicles with neighbors that have intersecting decision spaces, thus reducing computational cost while still maintaining reasonable performance. Each vehicle also decides its ideal event horizon based upon inferred knowledge of the operational environment, further reducing cost.**

I. Introduction

Teams of unmanned vehicles are capable of accomplishing a wide variety of mission objectives such as, boundary detection [1], border and convoy protection [2], and target search and tracking [3]. In this paper flight planning algorithms are developed for a group of vehicles tasked to accomplish multiple mission level objectives. Two specific missions, target tracking and area search, are implemented. However, the framework presented is valid for a larger range of applications.

Cooperative path planning problems have been solved using a variety of methods, including by using genetic algorithms [4], potential fields [5], and sequential convex programming [6]. This work follows the paradigm presented in [3], [7], [8] and [9] that uses information rewards to plan vehicle paths. The information-based approach provides adaptive algorithms capable of handling dynamic environments and incorporating a wide variety of mission objectives. In [7] the work provides an algorithm for cooperatively tracking ground targets. This approach was expanded in [3] to include searching for and classifying targets. We follow the work of [3] and [9], both of which present an information based reward in combination with a receding horizon controller (RHC). Although this approach provides good results, it can be computationally expensive since all cooperating vehicles must jointly plan their paths.

To jointly optimize with a standard RHC each potential vehicle path must be evaluated in combination with all the other possible vehicle paths. This approach becomes intractable with large event horizons and when using a large number of vehicles. However, joint optimization is critical in determining the best control commands for the vehicles as a group. By cooperating the total reward will be greater than the sum of rewards gained through individual optimization.

Several approaches have been proposed to reduce the time needed to complete the RHC joint optimization. In [3] a Rollout Policy was used to limit the number of routes that need evaluation. This approach scales linearly with an increase in time horizon. A grouping algorithm was also proposed to limit the number of cooperating vehicles. However, the approach, based on the nearest neighbor algorithm with a maximum group capacity, could uncouple vehicles with dependencies.

Another approach presented in [10] and [11] is to individually optimize a single vehicle while holding the decision variables of all peer vehicles constant. This process is repeated for each cooperating vehicle and iterated upon until convergence of all decisions is reached. A particle swarm optimization was used with suffix tries in [12]. The suffix tries identify the most common starting sequence thereby restricting the number of branches that need to be searched. These approaches primarily focus on reducing the internal complexity of the RHC joint optimization algorithm. The goal of this work is to eliminate external cost through a more judicious application of the algorithm.

This paper most closely follows the work of [9] and uses the same reward functions for a RHC that jointly drives a group of UAVs to search for and track targets. We expand the previous work by introducing a method for reducing the computational complexity of the controller. This is accomplished in two steps, first by determining an ideal event horizon for the RHC (potentially limiting the number of look-ahead steps). Secondly by grouping vehicles together that would benefit from jointly optimizing their paths. Groups are adaptive in deciding when their path decisions will be decoupled from other vehicles. This results in a balance between computational efficiency and the RHC defined optimal routes.

The paper proceeds as follows. Section II describes the aircraft model and sensor used in this work. In Section III the target tracking and area search mission objectives and reward functions are presented. The main contributions of the paper are presented in Section IV, where the path planning control and joint optimization techniques are described. Results from numerical simulations are given in Section V. And finally, Section VI provides a summary and conclusions.

II. Vehicle Model

All vehicles are modeled as fixed wing aircraft capable of commanding a bank angle. It is assumed that the vehicles

Cameron K. Peterson is with the Department of Electrical Engineering, Brigham Young University Provo, UT USA, cammy.peterson@byu.edu

978-1-5090-2182-6/17/$31.00 ©2017 IEEE

move in a plane of constant altitude with a constant (energy efficient) speed, V_a. The aircraft maneuvers following a coordinated turn given by $\dot{\psi} = (g \tan \phi)/V_a$, where g is the gravitational constant, $\dot{\psi}$ is the turn-rate, and ϕ is the roll angle. Each vehicle is limited in the amount it may roll, $abs(\phi) < \phi_{max}$.

Note that this formulation is similar to modeling each vehicle as a self-propelled particle that travels at constant speed and turns with gyroscope motion. This modeling approach for UAV path planning is abundant in the literature, see for example [13] and [14] which use it for target tracking and distributed monitoring respectively.

Each vehicle is equipped with a sensor that measures the ground range and azimuth of objects within its sensing radius. Vehicle movement is driven to maximize what it learns about the environment (i.e. the vehicle will move to put objects of interest in its field of regard).

The sensor's azimuth is the angle from north (y-axis) moving positively in the clockwise direction towards east (x-axis). The measurement noise covariance, given the range and azimuth standard deviations σ_r and σ_θ, is $R = diag[\sigma_r^2, \sigma_\theta^2]$. The measurement matrix relating measurement space to state space is given by the Jacobian

$$H_{v,m}(k) = \begin{bmatrix} \frac{x_m(k)-x_v(k)}{r_{m,v}(k)} & \frac{y_m(k)-y_v(k)}{r_{m,v}(k)} & 0 & 0 \\ \frac{y_m(k)-y_v(k)}{r_{m,v}(k)^2} & -\frac{x_m(k)-x_v(k)}{r_{m,v}(k)^2} & 0 & 0 \end{bmatrix}, \quad (1)$$

where $(x_v(k), y_v(k))$ represent the vth vehicle's planar position at time step k, $(x_m(k), y_m(k))$ is the mth target's position, also at time step k, and $r_{m,v}(k) = \sqrt{(x_m(k) - x_v(k))^2 + (y_m(k) - y_v(k))^2}$ is the distance between them.

Sensors share detections (azimuth and range measurements) and their own state belief with peer vehicles who are within their communication range.

III. Mission Objectives

This section describes the process of target tracking and the information-based reward functions used to balance tracking and searching.

A. Target Tracking

In this section the procedure for target tracking and its reward function are described. The state for target m is given by $\boldsymbol{x}_m(k) = [x_m(k), y_m(k), \dot{x}_m(k), \dot{y}_m(k)]$. When describing an estimated state the same variable are used, but are accented with a $\hat{(\cdot)}$.

Target motion is tracked assuming a discrete, constant velocity motion model $\boldsymbol{x}_m(k) = F\boldsymbol{x}_m(k-1) + w(k)$ where $w(k)$ is zero-mean, white process-noise with covariance matrix

$$Q = \begin{bmatrix} \frac{1}{3}\Delta t^3 q & 0 & \frac{1}{2}\Delta t^2 q & 0 \\ 0 & \frac{1}{3}\Delta t^3 q & 0 & \frac{1}{2}\Delta t^2 q \\ \frac{1}{2}\Delta t^2 q & 0 & \Delta t q & 0 \\ 0 & \frac{1}{2}\Delta t^2 q & 0 & \Delta t q \end{bmatrix} \quad (2)$$

and spectral density q. Δt is the discretized time. The state transition matrix F is chosen to be a constant velocity motion model for the targets. A Kalman filter,

$$\begin{aligned} \hat{\boldsymbol{x}}_m(k|k-1) &= F\hat{\boldsymbol{x}}_m(k-1|k-1) \\ P_m(k|k-1) &= FP_m(k-1|k-1)F^T + Q, \end{aligned}$$

is used to estimate the predicted state for each target. The predicted information matrix can be determined by taking its inverse of the error covariance, $\hat{I}_m(k|k-1) = P_m(k|k-1)^{-1}$.

Each vehicle is equipped with a range and azimuth sensor. The predicted range $t_r(k) = \sqrt{(\hat{y}_m(k|k-1) - y_v(k))^2 + (\hat{x}_m(k|k-1) - x_v(k))^2}$ and predicted azimuth $t_\alpha(k) = \tan^{-1}((\hat{y}_m(k) - y_v(k))/(\hat{x}_m(k) - x_v(k)))$ are measures from the vehicles position to the target's a priori state estimate. Note that the sensor's measured position is assumed to be known, but noisy.

Vehicles are controlled by commanding their roll angle. The desired roll angle is choosen by picking a vehicle position that will maximize the information gain of each target. Target information is measured with the information matrix

$$\hat{I}_m(k|k) = \hat{I}_m(k|k-1) + \sum_{v=1}^{V} \underbrace{\hat{H}_{v,m}(k|k-1)^T R^{-1} \hat{H}_{v,m}(k|k-1)}_{\hat{I}_{new}}.$$

The measurement matrix $\hat{H}_{v,m}$ of Equation (1) is now a function of the the mth target's predicted estimate-position, $(\hat{x}_m(k|k-1), \hat{y}_m(k|k-1))$. This is equivalent to the inverse of the Kalman filter error covariance, $\hat{I}_m(k|k) = P_m^{-1}(k|k)$.

For each potential heading direction the reward is the difference between the expected measurement updated information matrix and the expected information matrix after predicting forward the track state estimate. Target rewards are summed for all known target tracks. Thus the optimization function decides which heading to use by maximizing the information gain of the reward [9]

$$J_{target} = \sum_{m=1}^{\mathrm{M}} \left(log|\hat{I}_m(k|k)| - log|\hat{I}_m(k|k-1)| \right). \quad (3)$$

The determinant, $|\cdot|$, of the information matrices are used to represent a volume metric.

B. Area Search

The search mission is implemented by rewarding vehicles that enter unexplored regions of interest. To create the reward function the operational area is divided with equally spaced grid points. The position of grid point g in the east-north frame is (x_g, y_g). When the grid center is within the aircraft's field of view it acquires the information contained in the region. The reward for viewing a region grows exponentially beginning from the time it was last searched. For a single grid point, g, at time k the reward is [9]

$$J_g(k) = J_{g,max} - (J_{g,max} - J_g(k-1))\, e^{-\Delta t/\lambda} \quad (4)$$

where λ is a growth rate, k is the time step, $J_{g,max}$ is an upper probability limit that the grid cell cannot exceed. The reward for searching a region is the summation of the individual rewards of each grid point searched by the aircraft,

$$J_{search} = \sum_{g} J_g(k), \quad \forall g \in \Gamma \tag{5}$$

where Γ is the union of grid points that lie within the vehicles sensing radius, $r_g = \sqrt{(x_g - x_v)^2 + (y_g - y_v)^2} < r_s$.

IV. Joint Optimization with Variable Horizon Receding Horizon Controller

To allow vehicles to seek out information gains that may take time to realize a RHC is employed with N look ahead steps. The RHC prevents vehicles from only seeking the closest reward when long term benefits may direct the vehicle into a different direction.

The RHC is implemented by projecting the vehicles and environment forward through a series of potential control commands until it reaches its event horizon N. At every step, k, the application of all allowable control commands divide the number of path options, increasing the total number of possible routes. Each route is evaluated to determine which one maximizes the combined search and track reward functions. Then the process is repeated. For this work the control decision is a discrete set of roll commands that translate to a change in vehicle heading as described by the equation for a coordinated turn.

A key drawback of this method is the computational cost inherent in evaluating all the routes. Since, the number of routes increases as a power to the number of decision options. A vehicle with d discrete decision options must evaluate d^{N-1} potential routes for each time iteration. To put this in perspective, a vehicle with an $N = 5$ event horizon and a $d = 5$ decision space would need to evaluate 625 potential routes. But if a longer event horizon of $N = 15$ is needed, then more than 6 billion evaluations would be necessary. This complexity is further compounded under joint optimization when all the paths of all the vehicles must be evaluated in combination. The sub-sections that follow present a new algorithm for joint optimization which reduces the computational complexity inherent in the process.

A. Joint Reward

In this section the effect of peer vehicle paths on the total reward gained is considered. For target tracking a covariance intersection algorithm [15] is used to ensure that vehicles will not be over rewarded for tracking identical targets.

To determine the combined contribution of vehicle measurements the improved fast covariance intersection algorithm [16] is employed. In this approach the weights associated with vth vehicle at a given time are determined by (replicated from [16] for convenience)

$$\tilde{\omega}_v = \frac{|\boldsymbol{I}| - |\boldsymbol{I} - I_v| + |I_v|}{V|\boldsymbol{I}| + \sum_{v=1}^{V} (|I_v| - |\boldsymbol{I} - I_v|)}$$

where $\boldsymbol{I}$ is the sum of all information gathered from V sensors simultaneously, $\boldsymbol{I} = \sum_{v=1}^{V} I_v$. And the final information across all sensors (detections) is given by $\hat{\boldsymbol{I}}_{tot} = \sum_{v=1}^{V} \tilde{\omega}_v I_v$. The target tracking reward for each RHC path incorporates the weighted sum of information across all sensors. The joint tracking reward for all vehicles at time step k is [9]

$$J^V_{target} = \sum_{m=1}^{M} \left(\log |\hat{I}_{tot}(k|k)| - \log |I(k|k-1)| \right). \tag{6}$$

Note that these equations reduce to Equation (3) if only one vehicle views the target at that time step.

Information gained by vehicles searching in the same area is accounted for by a joint-search reward function [9]

$$J^V_{search} = \sum_{g} J_g(k), \quad \forall g \in \Gamma^V \tag{7}$$

where $\Gamma^V = \{\bigcup_v^V (r_g < r_s)\}$ is the set of grid points that lie within every vehicle's sensing radius. Reward for searching a grid point is given if it is contained within the field of regard of any vehicle. The simultaneous viewing by multiple vehicles provides no additional reward.

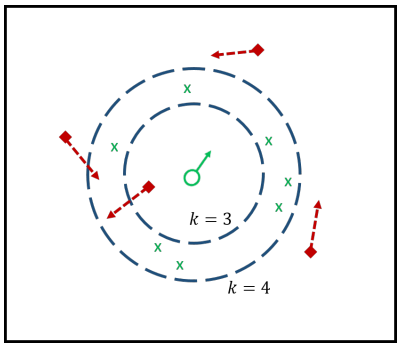


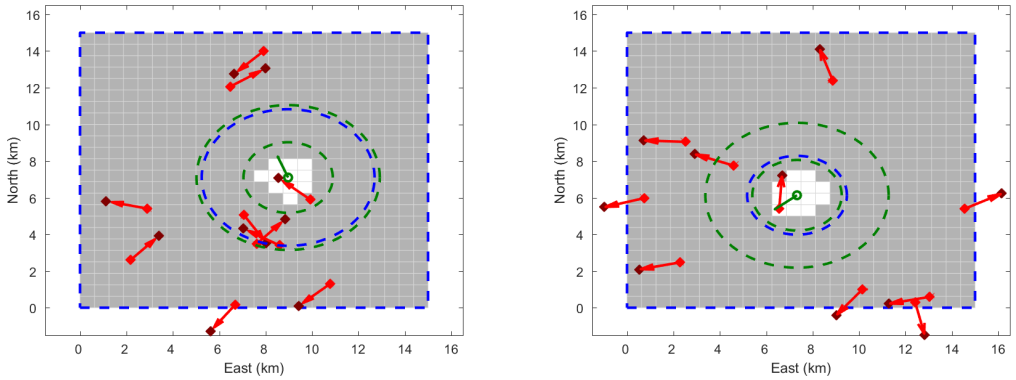
Fig. 1. Sampling points used to estimate the potential information gain for look-ahead step $k = 4$.

B. Variable Event Horizon

At every time step each vehicle calculates its desired RHC event horizon. The event horizon is upper bounded and will be shortened if the vehicle determines it can still make a judicious decision given a more restricted world view. This can significantly reduce the computational complexity of the joint action. To calculate the desired event horizon a potential reward value is predicted for each look-ahead step, then a balance is found between information gain and the number of look-ahead steps.

The potential reward is calculated using random samples drawn uniformly from the footprint of locations a vehicle might travel for that look-ahead step. The footprint is defined by a ring, as seen in Figure 1, which is lower bounded by the furthest distance the aircraft could travel for the prior look-ahead step and upper bounded by the maximum distance it could reach for the current look-ahead step. As a simplification it is assumed that the vehicle travels radially outward. The effect due to its initial heading is disregarded (i.e. turning constraints are not considered).

Figure 1 illustrates a sampling event where the reward for a $k = 4$ time step is determined. The environment (target



(a) Large desired event horizon encompassing high reward. (b) Small desired event horizon due to lack of reward.

Fig. 2. Numerical simulations showing the calculation of the desired event horizon.

and search states) are predicted forward to that step and random samples (represented by the green x values) are drawn uniformly from the region bounded by the two dashed green circles. This region is the furthest distance a vehicle could travel by time step $k = 3$ (inner ring) and $k = 4$ (outer ring). The samples represent possible future vehicle positions and are given by $v_s = (x_s, y_s)$, where $s = [1, 2, \ldots, S]$ is a sampling index. Also shown in the figure are the vehicle's position and heading (green circle and arrow) and the target's initial position and final location, for the $k = 4$ time step (red diamonds and arrows).

To ensure diversity, a minimum number of samples S_{min} are required for each step. However, the total number of samples increases beyond the minimum proportionally to the increased sampling space volume $S = \max(S_{min}, S_{min}(A_N/A_s))$, where A_s is the vehicle's sensing area and A_N defines the area of the sampling ring. The sampling area will increase with increased radial distance from the vehicle's current position.

At each sample position, v_s, the vehicle estimates the potential reward given its knowledge of the inferred environment. The look-ahead reward for step k is calculated by summing the reward values of all the samples for that time step,

$$J_k = \sum_{s=1}^{S} (J_{target}(v_s) + J_{search}(v_s)). \tag{8}$$

Note that these calculation are for a single vehicle and therefore use the reward functions defined by equations (3) and (5).

The reward value, J_k is computed for $k = N_{min}, \ldots, N_{max}$ and the resulting set of discrete values are fit to a spline curve and smoothed with a low pass filter to give function $f(t)$, where t is now a continuous value of time. The first and second derivatives of the spline function are used to define the curvature of the function as [17]

$$K(t) = \frac{f(t)''}{(1 + f(t)'^2)^{3/2}}.$$

The desired event horizon, $N_v = round(min(K))$, is set to be the location of minimum curvature, or the time step where the reward decreases the most.

Figure 2 shows two examples of a vehicle determining its ideal event horizon. In the figures the solid green circles show the vehicle with its initial heading direction. The boxes mark the search grid and the shading (white to dark gray) indicate the length of time since it has been searched. The red diamonds are the target initial and end (darker red) positions. The green dashed circles show the minimum and maximum sensing area for the look-ahead spaces under consideration. The blue dashed circle shows the approximate sensing boundary that a vehicle may travel for its chosen event horizon. In Figure 2(a) the chosen event horizon encompasses the cluster of vehicles close to the boundary, but does not extend beyond that point. This ensures that the reward of detecting those vehicles will be considered when optimizing the routes. The second example in Figure 2(b) shows a reward that will not increase significantly beyond the first step. In this case a more conservative event horizon is chosen.

Monte Carlo runs were used to compare the decisions chosen when using the ideal and maximum event horizons. In 100 simulations the final decision agreed between the two methods 84% of the time.

C. Vehicle Grouping for Joint Optimization

A second method for reducing complexity is employed to eliminate unnecessary cooperation between vehicles. Vehicles that may impact each other are grouped together and jointly optimize their paths. Vehicles outside of the group are not accounted for in the groups optimization process.

Groups are formed among vehicles who have intersecting decision spaces. The vehicle's decision space is the portion of the environment a vehicle may sense given its ideal event horizon, N_v. Therefore it is the area contained in the circle with radius $r_d = r_s + N_v V_a \Delta t$, and center location (x_v, y_v). Given two vehicles centered at (x_{v1}, y_{v1}) and (x_{v2}, y_{v2}) with decision radii r_{d1} and r_{d2}. If $(r_{d1} + r_{d2}) <$

$\sqrt{(x_{v1}-x_{v2})^2+(y_{v1}-y_{v2})^2}$ then the vehicle's decision space will intersect and they are grouped together.

Vehicles need only intersect with one other vehicle in the group to be clustered together. All vehicles not belonging to a group optimize their paths individually. Groups adopt the maximum lookahead step of the members to jointly optimize their paths.

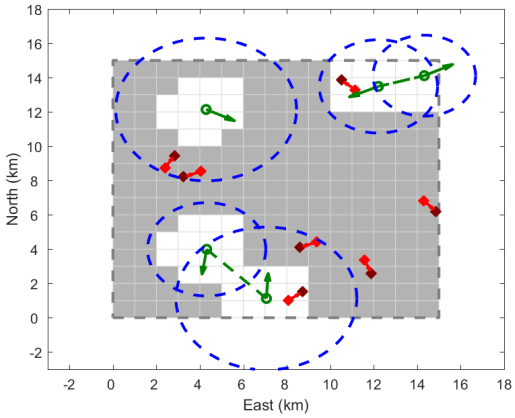


Fig. 3. Application of grouping algorithm to connect vehicles with intersecting decision spaces.

Figure 3 shows a simulation example of the grouping algorithm. In this example five vehicles and seven targets are initialized with random positions and headings and placed in an unsearched operational area. The vehicles each compute their desired event horizons, N_v, and use them to determine their decision space, a circle with radius r_d (shown with the blue-dashed circle). Vehicles are grouped with peers whose decision spaces overlap. The dashed green lines connect vehicles that belong in the same group.

D. Vehicle Routing

The algorithm is implemented in a discrete simulation with the main components shown in Figure 4. In the simulation each vehicle (sensor) garners the information in its sensing radius and incorporated it into its target state estimates. The new information is also shared with neighboring vehicles and every vehicle updates their operating picture. The vehicle's estimated environment (target states and grid cell values) and peer state information (position and velocity) is input into the routing algorithm which proceeds as follows:

1) Each vehicle determines its ideal event horizon, N_v, by the process described in Section IV-B.
2) Vehicles are grouped together when their decision space intersects. The decision space is estimated using the vehicles variable event horizon. (Section IV-C)
3) For each group a RHC is used to determine the best routes for all vehicles based off the joint reward functions defined in equations (6) and (7).
4) The vehicles are commanded to the roll angle necessary to follow their optimal route.
5) Each vehicle moves forward through the discrete simulation (Figure 4) by one time step and the process is repeated the next time the routing block is called.

V. Simulation Results

This section presents results that are obtained using a simulation structure outlined by Figure 4. These results illustrate the concepts of cooperatively searching and tracking.

Figure 5 shows the results of two vehicles tracking three targets at time step $k = 100$ seconds. The targets and vehicles were initialized with random position and heading directions. The green lines show the UAV vehicle paths. The red dotted lines indicate target tracks, with the current position given by the diamond and heading arrow. The current state of the searched area is shown in gray. Darker gray indicates that it has been longer since that area was searched. The figure shows that at the current time a significant percentage of the operational area has been searched. It is also evident from the figure that the vehicles maintain paths which follow the targets.

This simulation was run for a total of 300 seconds and the tracking and search results are compared for four different routing algorithms. The first implementation is the grouping algorithm as presented in Section IV-D. The second is a joint implementation where all vehicles mutually decide their next decision step using the maximum allowed event horizon and an exhaustive search of all the RHC paths. Third is an individual optimization where each vehicle optimizes its path regardless of other vehicle behavior. And the fourth is a non-optimized implementation, used for comparison, where the control decisions are made randomly.

TABLE I
AVERAGE ERROR AND STANDARD DEVIATION OF THE TARGET POSITION (M) FOR THREE TARGETS WITH TWO SENSORS.

	Target 1	Target 2	Target 3	Mean
Random	81.4+/-95.2m	57+/-40.2m	262+/-202m	134m
Individual	12.9+/-14.2m	168.3+/-188m	16.4+/-33m	65.9m
Grouped	13.4+/-14.2m	13.3+/-23.6m	40+/-39.6m	22.2m
Joint	12.5+/-14.3m	12.6+/-23.7m	22+/-32.4m	15.7m

Table I shows the tracking error of all three targets for each routing implementation. As expected all optimization algorithms outperform randomly chosen directions. Also on average the joint implementation provides the best results by tracking all three targets with an average of 15.7m while the grouping algorithm does slightly worse at 22.2m. The individually optimization tracks target one and three well at the expense of losing target two. The grouping algorithm does the best at searching the operational space. It maintains an average of 67.2% searched with the joint optimization at 64.2%, individual optimization at 60.1%, and random at 53.7%. Though the joint optimization tracking results are slightly better, the grouping algorithm was able to achieve its results with an 86% increase in speed.

VI. Conclusion

This paper presented an improved method for cooperatively searching for and tracking ground targets using a

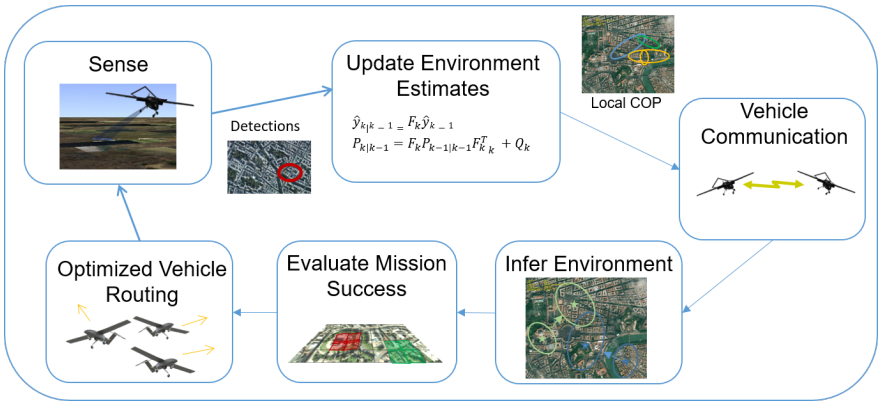


Fig. 4. The primary components of the search and track simulation. Each block represents a process that is executed in sequence.

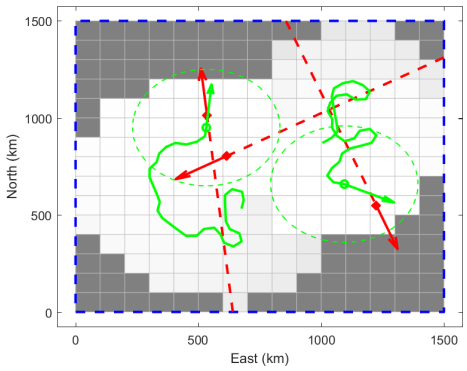


Fig. 5. Simulation of two vehicles and three targets and their tracks after 100 seconds.

combined RHC with variable event horizon. The vehicles were routed to positions of maximum reward for both reducing uncertainty in target estimates and exploring new areas. A grouping algorithm was presented which reduced the computation time without significantly decreasing the algorithm's effectiveness.

ACKNOWLEDGMENT

This research was supported by the NSF Center for Unmanned Aircraft Systems (C-UAS) and Brigham Young University.

REFERENCES

[1] B. A. White, I. Antonios Tsourdos, M. Ashokaraj, S. Subchan, and R. Zbikowski, "Contaminant cloud boundary monitoring using network of UAV sensors," *IEEE Sensors Journal*, vol. 8, no. 10, pp. 1681–1692, 2008.

[2] X. C. Ding, A. R. Rahmani, and M. Egerstedt, "Multi-UAV convoy protection: An optimal approach to path planning and coordination," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 256–268, 2010.

[3] X. Tian, Y. Bar-Shalom, and K. R. Pattipati, "Multi-step look-ahead policy for autonomous cooperative surveillance by UAVs in hostile environments," *Proceedings of the IEEE Conference on Decision and Control*, vol. 5, no. 1, pp. 2438–2443, 2008.

[4] H. Ergezer and K. Leblebicio, "3D Path Planning for Multiple UAVs for Maximum," *Journal of Intelligent and Robotic Systems*, pp. 737–762, 2014.

[5] B. Di, R. Zhou, and H. Duan, "Potential field based receding horizon motion planning for centrality-aware multiple UAV cooperative surveillance," *Aerospace Science and Technology*, vol. 46, pp. 386–397, 2015.

[6] D. Morgan, G. P. Subramanian, S.-j. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1261–1285, 2016.

[7] A. Sinha, T. Kirubarajana, and Y. Bar-shalomb, "Autonomous Ground Target Tracking by Multiple Cooperative UAVs," in *Aerospace Conference*, (Big Sky, Montana), pp. 1–9, 2005.

[8] R. R. Pitre, X. R. Li, and R. Delbalzo, "UAV route planning for joint search and track missions–an information-value approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2551–2565, 2012.

[9] A. J. Newman, S. R. Martin, J. T. DeSena, J. C. Clarke, J. W. McDerment, W. O. Preissler, and C. K. Peterson, "Receding Horizon Controller using Particle Swarm Optimization for Closed Loop Ground Target Surveillance and Tracking," *Signal Processing, Sensor Fusion, and Target Recogniton*, vol. 7336, no. 1, pp. 73360M–1–73360M–12, 2009.

[10] K. Hernández and J. C. Spall, "Cyclic stochastic optimization with noisy function measurements," *Proceedings of the American Control Conference*, pp. 5204–5209, 2014.

[11] C. K. Peterson, A. J. Newman, and J. C. Spall, "Simulation-based examination of the limits of performance for decentralized multi-agent surveillance and tracking of undersea targets.," in *SPIE Defense+ Security. International Society for Optics and Photonics*, 2014.

[12] S. R. Martin and A. J. Newman, "The application of particle swarm optimization and maneuver automatons during non-Markovian motion planning for air vehicles performing ground target search," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 2605–2610, 2008.

[13] A. V. Savkin and H. Teimoori, "Bearings-Only Guidance of a Unicycle-Like Vehicle Following a Moving Target with a Smaller Minimum Turning Radius," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 960–964, 2010.

[14] F. Morbidi, R. A. Freeman, and K. M. Lynch, "Estimation and control of UAV swarms for distributed monitoring tasks," *American Control Conference (ACC), 2011*, pp. 1069–1075, 2011.

[15] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, vol. 4, pp. 2369–2373 vol.4, 1997.

[16] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," *2005 7th International Conference on Information Fusion, FUSION*, vol. 1, pp. 154–160, 2005.

[17] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," *Proceedings - International Conference on Distributed Computing Systems*, pp. 166–171, 2011.