Decentralized Control of Unmanned Aerial Vehicle Collaborative Sensing Missions

Allison Ryan†, John Tisdale†, Mark Godwin†, Daniel Coatta†,
David Nguyen†, Stephen Spry†, Raja Sengupta§, and J. Karl Hedrick†

Abstract— **Cooperative unmanned aerial vehicle (UAV) teams can serve as a mobile sensor networks to autonomously execute sensing tasks in uncertain and dynamic environments. We have implemented a UAV system that performs collaborative sensing missions under the supervision of a single user. Decentralized task allocation and autonomous mission execution are enabled by onboard computing and ad-hoc wireless communication and provide robustness to communication and resource losses in quickly evolving scenarios.**

The collaboration algorithm combines shared and local information to produce multi-step plans with the goal of minimizing the global cost of the mission. Missions are constructed in real time from tasks such as patrolling an area or searching for an intruder, and may include constraints such as a restricted airspace. Algorithms for decentralized planning, guaranteed search and avoiding a restricted airspace were demonstrated using a team of four UAVs. Experimental data shows that tasks were allocated to the appropriate UAVs and successfully executed.

I. Introduction

As unmanned aerial vehicles (UAVs) become more sophisticated and widely deployed, there is an increasing need for control systems that allow a single user to effectively utilize a team of UAVs to gather required information. We present a system developed by the Berkeley Center for Collaborative Control of Unmanned Vehicles (C3UV) through which a single operator manages a team of UAVs performing a cooperative sensing mission. The user controls the UAV team by creating cooperative sensing tasks in real time. UAVs decompose and allocate tasks using onboard computation and aircraft-to-aircraft communication. The system is designed with a focus on robust performance in order to perform with limited and lossy communication.

The goal of cooperating agents optimally completing geographically separated tasks borrows from well-established areas such as optimal path planning, theories of cooperation and combinatorial optimization. Much of the theoretical work (with the notable exception of multi-agent planning in the artificial intelligence community [1]) focuses on optimally allocating “visit” tasks in which a UAV must pass through a given point [2], [3]. Auction-based methods for task allocation are based on bid computations that provide theoretical bounds related to optimal performance [4].

Methods for dealing with a variety of task types and heterogeneous UAV teams must incorporate constraints and problem formulations that are difficult to characterize within the optimal path planning and routing literature. Algorithms implemented on experimental multi-UAV teams often specialize to a very specific mission objective such as locating the origin of a signal [5]. More generalized techniques such as decentralized model predictive control or robust decentralized task assignment [6] are able to represent a variety of cost functions and constraints, but tend to come at a high computational cost and with fewer guarantees regarding optimality.

The cooperative control system presented here relates to both scenarios. Given a set of “visit” tasks, the system solves a general vehicle routing problem that can be compared to theoretical performance bounds from the literature. Additional task types of increasing complexity can be allocated using the same methods, allowing the system to perform complex sensing missions based on the same verified planning algorithms. Similarly to the robust decentralized task assignment method, we explicitly acknowledge limited communication and information consensus issues and allow UAVs to exchange plans in addition to state estimates. Sensing missions with UAV teams have also been demonstrated at the University of Pennsylvania GRASP Laboratory [7] and the Australian Center for Field Robotics [8].

In section II we describe the infrastructure that supports UAV collaboration, including communication systems and user interfaces. Section III will describe how tasks are allocated to UAVs, and section IV will describe the types of tasks and how they are executed. In section V we present results of a flight experiment and analyze both the execution of individual tasks and the effectiveness of multi-agent collaboration, followed by conclusions in section VI.

II. System Infrastructure

The system infrastructure is based on a Sig Rascal 110 airframe and the Piccolo II autopilot from Cloud Cap Technology [9]. An onboard PC104 computer executes high-level control and collaboration. These aspects of the system are very similar to the infrastructure that supported the precursor system demonstrated in 2005 [10]. The most important onboard software processes are “collaboration” and “task execution”.

The Sig Rascal airframe shown in Fig. 1 has a wingspan of 2.8 meters and takeoff weight of 11.8 kilograms and is capable of flights over one hour. The 32 cc two-stroke gasoline engine is mounted to reduce vibration transmission, as is the PC104 system. With low-level stability and control

This work was supported by the Office of Naval Research

†Department of Mechanical Engineering, University of California, Berkeley. Berkeley, CA, 94720, USA.

§Department of Civil and Environmental Engineering, University of California, Berkeley. Berkeley, CA, 94720, USA.

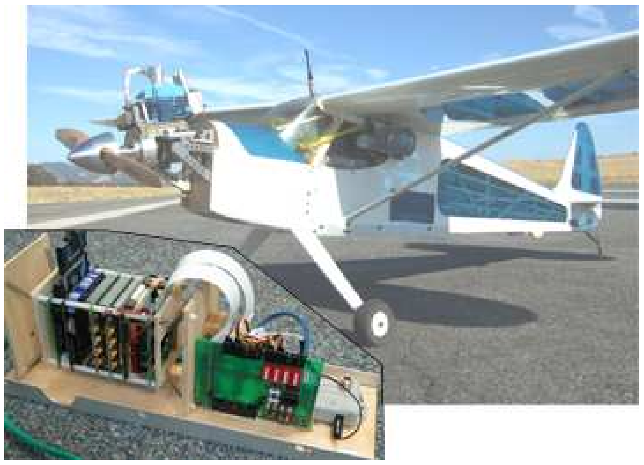
Fig. 1. Sig Rascal airframe and payload tray with vibration isolated PC104 computer

provided by the Piccolo autopilot, the UAV is represented as a constant velocity, constant altitude kinematic model. The yaw rate is a bounded control input.

The communication infrastructure consists of the 900 MHz UHF radio integrated in the Piccolo system and amplified 2.4 GHz ad-hoc wireless ethernet (802.11b). The Piccolo system is commanded and monitored by the operator over the 900 MHz link as well as by the PC104 onboard control. The 900 MHz radio also provides long-range, low-bandwidth communication between the Mission Commander Interface and onboard software. Wireless ethernet is used as a separate link for aircraft to aircraft communication. A single user directs the team of UAVs through the graphical Mission Commander Interface by creating a mission composed of tasks and constraints defined on a terrain map.

III. Collaboration

The system for collaboration is designed to produce robust near-optimal plans to execute a rapidly changing mission. The collaboration scheme must react quickly to changes in the mission or resources (such as UAV faults) and must performance should degrade gracefully with limited communication. The auction strategy for task allocation is designed to approximate a solution to the Min-Max Vehicle Routing Problem (VRP). For vehicles numbered $j = 1$ to N traveling paths with cost c_j, the goal of Min-Max VRP is to choose an allocation of vehicles to tasks that solves

$$\min \max_j c_j. \tag{1}$$

When cost is defined based on time, this optimization corresponds to minimizing the time until the last vehicle has completed its task. Rather than directly solve the NP-hard Min-Max VRP, the system presented here plans allocations in which the UAV tour costs are roughly equal. It is straight forward to show that balanced tours, within the some factor, are a necessary condition for a solution of the Min-Max VRP, and so solutions to the Min-Max VRP may be approximated by creating balanced tours.

Tasks are allocated using a decentralized auction algorithm. The algorithm has the effect of creating greedy multi-step plans for the team, in which each UAV performs the set of tasks that it may accomplish faster than all other UAVs. In the absence of communication, each UAV will plan to perform the entire mission; as the UAVs communicate, conflicts are resolved so that each task is assigned to exactly one UAV and performance improves. Given a static mission definition, the team will converge on a greedy allocation in which each task is assigned to the UAV which can do it at the lowest cost. A rich set of task descriptors enables a variety of missions. The mission parameters are taken into account in the planning process through cost calculations and allocation constraints. Information regarding plans and costs is communicated among the team, in addition to the *mission definition*.

The *mission definition* consists of a set of tasks and constraints created by the user, and possibly agent-created tasks. UAVs communicate *mission definitions* whenever a task has been added or deleted. The *mission state estimate* (MSE) consists of the information necessary for collaboration, specifically information about plans, associated costs and the status of tasks. Given communication, each UAV's local MSE will converge to reflect a set of plans where each task is assigned to a single UAV. The procedure for fusing local and incoming MSEs depends on data time stamps and a set of logical rules which account for the relevancy of a piece of data [11].

A. Task Allocation using a Mission State Estimate

The aim of the multi-step planning algorithm is to assign each task to exactly one UAV such that the cost to complete the mission approximates the desired objective function (1) and all constraints are satisfied. This is accomplished by iterations of planning and communication at periodic intervals. After the plans converge and while the mission remains constant, communication and planning iterations continue to check for new tasks, a better allocation or an agent failure.

An agent forms a plan consisting of only feasible tasks using its local knowledge and the information contained in the MSE. A task is feasible for a UAV if it can be accomplished without violating constraints at a lower cost than published by any other UAV. Each task has an associated priority, which may be *hard* or *soft*. A soft priority appears as a weight in the cost calculation, but a hard priority is a constraint that a task is infeasible until all tasks with a higher hard priority have been assigned to plans.

A UAV that can complete a task at a lower cost than published by any other team member is allowed to add that task to its own plan, editing the agent assignment and cost in the MSE and publishing the update as required. Similarly, if a UAV plans to do a task but sees a lower cost published, it removes that task from its own plan. In effect, the negative of a task cost corresponds to a bid in a task-allocation auction framework [4].

It is noteworthy that each UAV forms its own plan which is influenced by plans published by others, but no UAV makes an assignment for another UAV or makes assumptions about another UAV's plan in addition to published information. This distinguishes the algorithm from both centralized planners and implicit coordination methods [12] and leads to increased autonomy (due to local decision making) and decreased sensitivity to accurate team consensus.

The multi-step planning algorithm (Algorithm 1) consists of an outer loop to deconflict plans and an inner loop to form them. After processing a new MSE, the UAV calculates the cost to complete each task, eliminates infeasible tasks, and adds the lowest-cost task to the first step of its plan. Costs are computed by the task execution process based on the associated control algorithm.

Algorithm 1 Multi-step planning

```
1: while MSEs conflict do
2:     while feasible tasks remain unassigned do
3:         calculate costs
4:         eliminate infeasible tasks
5:         add best task to plan
6:         project state forward
7:     update and broadcast MSEs
8:     process incoming MSEs
```

After adding a task to its plan, the UAV estimates its location and the elapsed time upon completing the task. It uses this projection into the future to recalculate costs and adds the lowest cost feasible task to the next step of the plan. This process continues, with successive projections of the future UAV state, until all feasible tasks have been added to the plan. At this point the MSE is updated and broadcast, triggering another cycle of deconfliction.

B. Task decomposition

A task such as patrolling a large area may have more than one UAV assigned to it if necessary to provide a desired search frequency. In this case, the cycle of planning and deconfliction will allow more than one UAV to claim the task. If N UAVs are assigned to a task, the MSE states that the task should be decomposed into N individual roles. The task decomposition algorithm executes identically on each UAV based only on the task definition and N. Roles are then allocated to individual UAVs using the same decentralized bidding process as for allocation of tasks. The decomposition processes is reactivated in response to any change in the UAV assignment, providing dynamic reallocation in response to changing mission definitions or resources.

IV. Task Execution

The task execution process applies an appropriate control algorithm to execute the current task. It also estimates the costs to complete tasks, which are required for allocation, and reports when a task is complete or has failed. The implemented task types are

- visit point
- patrol segment
- patrol area
- guaranteed search.

A *safe flight filter* is required in addition to the set of task execution algorithms in order to enforce motion constraints such as a restricted airspace or in the future, collision avoidance.

The *visit point, patrol segment* and *patrol area* tasks can be considered variations of a single task, to travel a path with varying dimensionality. For each case, the task can be assigned a timing property of *once*, *continuous* or *reactivated*. These tasks are executed by commanding a series of waypoints that correspond to the path. If the task is continuous, the UAV will continue to track the closed waypoint path until it is preempted by a new task. Otherwise, the task is complete after each waypoint has been visited.

If a task is to be reactivated, its status will change from DONE to TODO after the desired reactivation time. This is different from the usual notion of scheduling which guarantees a strict periodic timing constraint. Strict scheduling constraints cannot be guaranteed in the highly dynamic environment and mission scenario assumed here, so a reactivation-based scheduling concept is more appropriate.

These *visit point, patrol segment*, and *patrol area* tasks are useful for gathering data from a specified location and result in task allocation and routing problems that can be compared to benchmarks from the literature such as traveling salesman problems. However, they are trivial to execute by commanding waypoints. In contrast, the *guaranteed search* algorithm and *safe flight filter* described in the following sections incorporate the UAV's dynamic constraints into turn rate based control algorithms.

A. Guaranteed Search Task

The *guaranteed search* task produces a search pattern that will locate a target with an uncertain initial position and a known upper bound on velocity. The full algorithm and derivation have been published previously [13]. It is based on a deterministic analysis of worst-case target motion and a perfect sensor model.

The initial locus of target positions is a circle with radius $R_l(0)$. The radius of the locus of achievable target positions grows with velocity V equal to the maximum target velocity. The optimal path for the target to escape a searcher orbiting at constant velocity U is to travel at angle γ as shown in Fig. 2, which depends on the searcher's and target's radial coordinates r and R_t respectively (2). We assume the sensor is mounted on a downward pointing gimbal.

$$\gamma = \frac{Vr}{UR_t} \tag{2}$$

The optimal search strategy is a path that alternates between straight segments and outward spirals along the edge of the growing target locus. If the net change in r per encirclement of the target locus is negative, $R_l(t)$ will be reduced until the target is located. When the UAV sensor meets the edge of the target locus, the UAV orbit radius is given by $R_l(t_a)+R_s$, where the radius of the sensor footprint is R_s. The maximum value of $R_l(t_a)$ for which the UAV can

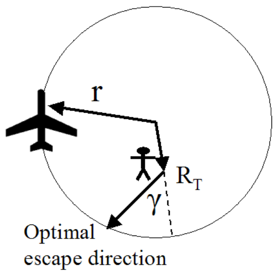

Fig. 2. Optimal travel direction for escape from orbiting searcher

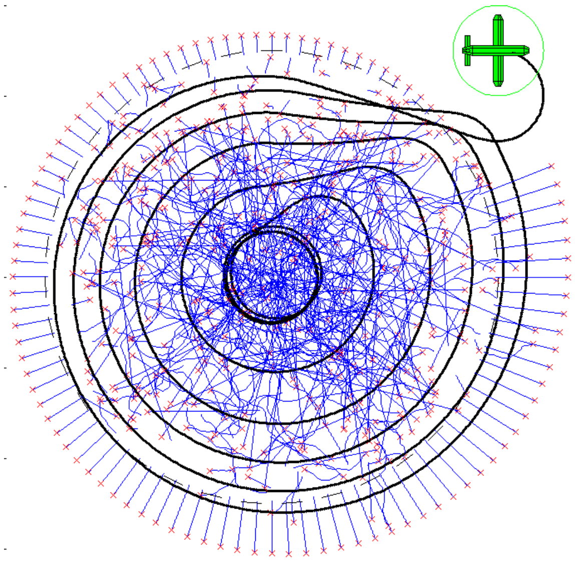

Fig. 3. Guaranteed search path tracked using sliding mode turn rate control, with random target paths

maintain a bounded target locus is given by R_l^* (3). If $R_l(t_a)$ when the UAV reaches the target locus is less than R_t^*, then $R_l(t)$ will decrease after each orbit and the search can be guaranteed.

$$R_l^* + R_s = \left(R_l^* - R_s + 2V\sqrt{\frac{R_l^* R_s}{U^2 - V^2}}\right) * \exp\left(\frac{V(2\pi - \arccos((R_l^* - R_s)/(R_l^* + R_s)))}{\sqrt{U^2 - V^2}}\right) \tag{3}$$

The desired UAV path is defined in a polar coordinate system (r, θ) based at the center of the target locus where $r(t)$ is defined by the optimal search strategy and angular velocity follows from the constant airspeed assumption. A sliding mode turn rate control [14] provides stable tracking of the desired path.

Fig. 3 shows the trajectory of a simulated UAV tracking the desired path under sliding mode control. The circle around the UAV icon represents its sensor range and the broken black line shows the target set on arrival of the UAV. 500 simulated intruders followed randomly generated paths beginning within the target set. The UAV passed within sensor range of each target, representing a perfect detection rate under the assumption of a perfect sensor.

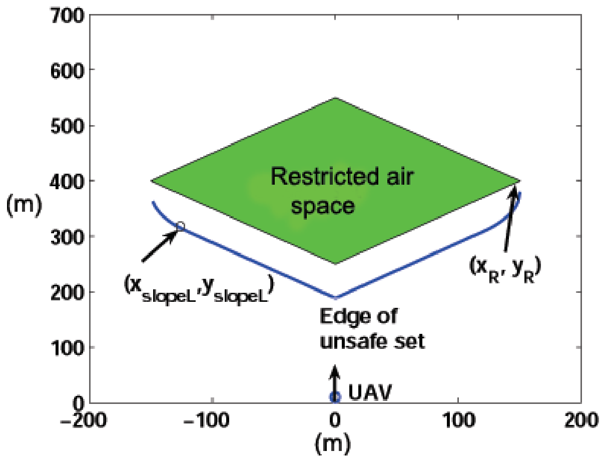

Fig. 4. Points associated with the restricted airspace for calculating the unsafe set

B. Safe Flight Filter

A UAV must execute tasks and travel between them without entering restricted airspace. The safe flight filter has two distinct modes of operation corresponding to the task execution process's turn rate and waypoint control modes. Both are based on the UAV's forward reachable set as determined by its kinematic constraints. The reachable set $\{\mathcal{T}\} = \mathcal{F}(\mathbf{x})$ is the set of achievable trajectories starting from state $\mathbf{x} = (x, y, \psi)$

1) Turn rate mode: Under turn rate control, we define the unsafe set $\mathcal{U} \subset \mathcal{R}^3$ as the set of aircraft states such that all trajectories beginning at $\mathbf{x} \in \mathcal{U}$ pass through the restricted airspace $\mathcal{X}$ (4). This method is identical to a safe flight calculation used to avoid an obstacle [15].

$$\mathcal{U} = \{\mathbf{x} : \mathcal{T} \cap \mathcal{X} \neq \emptyset \ \forall \ \mathcal{T} \in \mathcal{F}(\mathbf{x})\} \tag{4}$$

The unsafe set $\mathcal{U}$ has a closed form definition for a convex polygonal restricted airspace and the kinematic UAV model. A point $\mathbf{x} = (x, y)$ in the UAV's local aligned coordinate system is inside the unsafe set if y is greater than functions $z_L(x)$ and $z_R(x)$ which form the boundaries of $\mathcal{U}$. $z_L(x)$ is given in (5) and $z_R(x)$ is similarly defined, based on points $\mathbf{x_{slopeL}}$ and $\mathbf{x_{slopeR}}$ where the boundary of $\mathcal{U}$ has slope m_L or m_R. T_d is the assumed control delay and R is the UAV's minimum turn radius. The restricted airspace has left and right approaching corners at $\mathbf{x_L}$ and $\mathbf{x_R}$ and left and right nearest faces with slopes m_L and m_R as shown in Fig. 4, in which the arrow indicates the UAV's position and heading.

$$z_L(x) = \begin{cases} y_L - UT_d - \sqrt{R^2 - (x - R - x_L)^2} \\ \qquad x_L \leq x \leq x_{slopeL} \\ y_{slopeL} + m_L(x - x_{slopeL}) \\ \qquad x_{slopeL} \leq x \leq x_R \end{cases} \tag{5}$$

A reference point is placed directly ahead of the UAV at $(0, \Delta)$ in local coordinates. If the reference point is within the unsafe set, the UAV must execute an evasion maneuver. Its location determines whether the UAV should evade to the left or right, as in Algorithm 2. Currently, any task requiring turn rate control cannot be achieved if a restricted airspace interferes with the required trajectory. Therefore, the evasion maneuvers are to proceed at maximum turn rate to a safe waypoint either to the left or right of the restricted airspace and then proceed to another task. The commanded evasion maneuver is guaranteed to be safe because it forms the boundary of the UAV's reach set that is not intercepted by $\mathcal{U}$.

Algorithm 2 Evasion rules for turn-rate control

```
if $\Delta \geq z_L(0) \geq z_R(0)$ then
    evade left
if $\Delta \geq z_R(0) \geq z_L(0)$ then
    evade right
else
    continue
```

2) Waypoint mode: To avoid entering a restricted airspace en route to an otherwise safe waypoint, a UAV must begin an evasive maneuver before it enters $\mathcal{U}$. $\mathcal{U}$ will always be contained within a set $\mathcal{P}$ formed by extending the polygon $\mathcal{X}$ by distance d, where d is the maximum thickness of $\mathcal{U}$ (6).

$$d = UT_d + R \tag{6}$$

The safety calculation consists of placing a reference point $(0, \Delta)$ ahead of the UAV as in the previous section and calculating whether it lies within $\mathcal{P}$. If so, a detour is executed by visiting the vertices of $\mathcal{P}$ until the trajectory to the original waypoint no longer intersects $\mathcal{U}$. The direction of travel around $\mathcal{P}$ is chosen to minimize the detour distance.

V. Flight Experiment with 4 UAVs

A flight experiment with four UAVs was conducted at McMillan Army Airfield in Camp Roberts, California. The team of UAVs was controlled by a single operator using the interface described in section II to create sensing tasks.

A. Analysis of Task Execution

All of the tasks described in section IV were successfully executed. The guaranteed search task was tested by placing a GPS device on a human evader. The human attempted to evade the UAV and was detected when he passed within the UAV's simulated sensor range.

Fig. 5 shows a UAV avoiding a restricted airspace while executing a *guaranteed search* under turn rate control. The restricted airspace is defined by the large rectangle, which includes a buffer to account for the susceptibility of small aircraft to wind disturbances. The UAV's position and orientation are indicated by a small arrow located at the end of its traversed path. The blue curve adjacent to the restricted airspace is the border of the unsafe set, which can be observed to depend on the UAV's orientation. The UAV begins a detour maneuver between frames two and three, and in the fourth frame is en route to a safe detour waypoint, having avoided the restricted airspace.

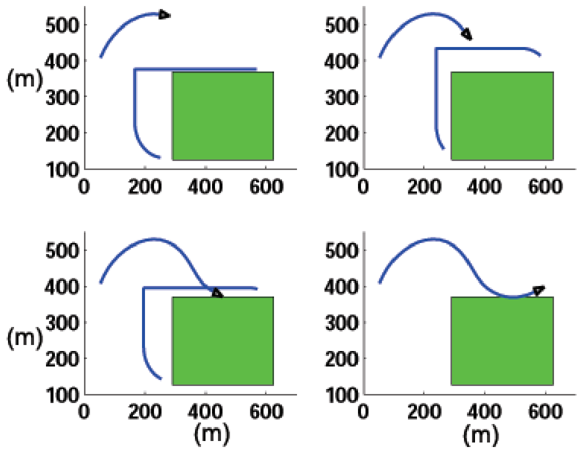

Fig. 5. UAV avoids restricted airspace under turn rate control and detours to safe waypoint

B. Analysis of Collaboration

The goals of collaboration were to provide robust task allocation and to execute decomposed tasks. With good communication and few constraints, we expect *visit* tasks to be completed in near-optimal time. When a restricted airspace interferes with the optimal path, or when communication fails, we expect that tasks will still be completed but at higher cost.

Most tasks were immediately claimed by the optimal UAV based on a Euclidian distance cost measure. The most outstanding case of delay in task allocation was caused by two UAVs publishing nearly identical costs for the same task, leading to a short burst of switching behavior before the final allocation was achieved.

By measuring the time between a task being created and the user being informed of its completion, we analyze the system from the user's perspective, and thereby include delays due to communication and negotiation. However, frequent loss of packets directed at the user interface meant that this measurement could be a severe over estimate of the actual cost. In accordance with a user-centered analysis, we also calculate the optimal cost to complete a task based on the states of the UAVs at the time the task was created rather than at the time of allocation.

The optimal path for tasks of the type *visit point once* consists of a maximum curvature turn followed by a tangent line, and so the optimal cost can be easily calculated. Five tasks of this type were completed at costs ranging from near the optimal to a maximum of three times the optimal, as shown in Fig. 6. The tasks completed at near-optimal cost suggest that waypoint control can provide near-optimal paths under favorable conditions. The remaining tasks were completed at costs much higher than the optimal cost for any of the team of UAVs. Therefore, the high cost was not due only to an incorrect assignment, but was more likely due to the high degree of packet loss observed on the user interface communication channel.

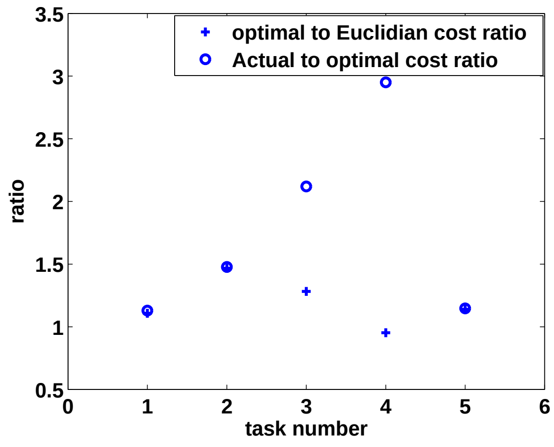

Fig. 6. Performance and cost estimates for *visit point* tasks

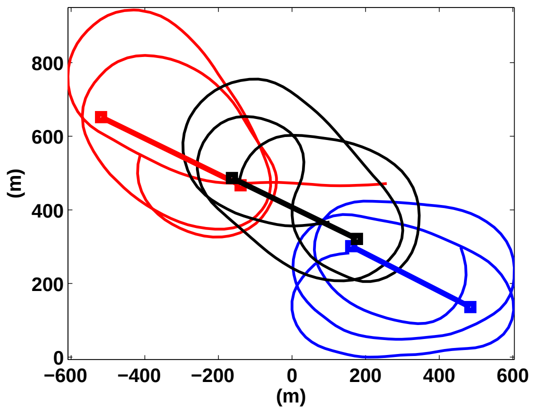

Fig. 7. Three UAVs cooperatively patrol a segment

Fig. 6 also shows that the ratio of Euclidean versus Dubins model [16] cost estimates ranged from approximately 1 to 1.5. This ratio indicates the error introduced by using a Euclidean cost estimate for task allocation (as is done in this system) instead of calculating the true cost along a feasible Dubins path. Assigning tasks by true rather than Euclidean cost would have effected the result in two of the five cases, showing that tasks were not dispersed widely compared to the minimum turn radius, and a Dubins cost estimate may improve performance.

The ability to cooperatively execute a task was demonstrated for the cases of patrolling a segment and an area. As the mission varied, the number of UAVs assigned to the cooperative tasks changed and the tasks were decomposed as described in section III-B. Fig. 7 shows the paths of three UAVs cooperatively patrolling a 1.4 kilometer segment. Each UAV's estimate of the endpoints of its own segment appear as squares.

VI. Conclusions

This system for collaborative control of a UAV team combines algorithms for multi-agent collaboration, vehicle motion controllers for a variety of sensing tasks, and a complete infrastructure for multiple aircraft flight experiments. The methods for collaboration are based on the exchange of limited information in the form of mission state estimates, allowing decentralized task allocation and information propagation without a single point of failure. The collaboration strategy is robust to losses of communication or agents and executes missions composed of a variety of tasks and constraints.

Future work includes expansion to an airframe with larger payload capacity to incorporate a variety of sensors such as lidar, infrared, and computer vision into the existing sensing mission framework. Further analysis of the collaboration algorithms should result in guarantees of convergence of mission state estimates and performance bounds for traveling salesman type missions.

References

[1] D. Pynadith and M. Tambe, “The communicative multiagent team decision problem: Analyzing teamwork theories and models,” *Journal of Artificial Intelligence Research*, vol. 16, 2002.

[2] K. S. J. Enright, E. Frazzoli and F. Bullo, “On multiple UAV routing with stochastic targets: Performance bounds and algorithms,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2005.

[3] V. Sharma, M. Savchenko, E. Frazzoli, and P. Voulgaris, “Time complexity of sensor-based vehicle routing,” in *Robotics: Science and Systems I*, S. Thrun, G. S. Sukhatme, and S. Schaal, Eds. Cambridge, MA: MIT Press, 2005, pp. 297–304.

[4] M. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Mayerson, and S. Jain, “Auction-based multi-robot routing,” in *Proceedings of the International Conference on Robotics: Science and Systems (ROBOTICS)*, 2005, pp. 343–350.

[5] G. Hoffmann, S. Waslander, and C. Tomlin, “Distributed cooperative search using information-theoretic costs for particle filters, with quadrotor applications,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006.

[6] M. Alighanbari and J. How, “Robust decentralized task assignment for cooperative uavs,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006.

[7] B. Grocholsky, R. Swaminathan, J. Keller, V. Kumar, and G. Pappas, “Information driven coordinated air-ground proactive sensing,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005.

[8] D. Cole, A. Goktogan, and S. Sukkarieh, “The demonstration of a cooperative control architecture for UAV teams,” in *Proceedings of the 10th International Symposium on Experimental Robotics*, July 2006.

[9] B. Vaglienti, R. Hoag, and M. Niculescu, “Piccolo system user's guide,” Cloud Cap Technology, September 2006, v. 1.3.2.

[10] A. Ryan, X. Xiao, S. Rathinam, J. Tisdale, D. Caveney, R. Sengupta, and J. Hedrick, “A modular software infrastructure for distributed control of collaborating unmanned aerial vehicles,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006.

[11] M. Godwin, S. Spry, and J. Hedrick, “Distributed collaboration with limited communication using mission state estimates,” in *Proceedings of the American Controls Conference*, June 2006.

[12] P. Chandler, “Decentralized control for an autonomous team,” in *Proceedings of AIAA Unmanned Unlimited systems, technologies, and operations*, 2003.

[13] T. G. McGee and J. K. Hedrick, “Guaranteed strategies to search for mobile evaders in the plane,” in *Proceedings of the American Controls Conference*, June 2006.

[14] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1990.

[15] E. Frew and R. Sengupta, “Obstacle avoidance with sensor uncertainty for small unmanned aircraft,” in *Proceedings of the 43rd IEEE Conference on Decision and Control*, December 2004.

[16] L. E. Dubins, “On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents,” *American Journal of Mathematics*, vol. 79, pp. 497–516, 1976.