Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Decentralized Task Assignment for Unmanned Aerial Vehicles

Mehdi Alighanbari and Jonathan P. How

Aerospace Controls Laboratory
Massachusetts Institute of Technology
{mehdi_a, jhow}@mit.edu

Abstract— **This paper investigates the problem of decentralized task assignment for a fleet of UAVs. Centralized task assignment for a fleet of UAVs is often not practical due to communication limits, robustness issues, and scalability, and using a distributed approach can mitigate many of these problems. One recently proposed decentralized approach is to replicate the central assignment algorithm on each UAV. The success of this *implicit coordination* strongly depends on the assumption that all UAVs have the same situational awareness. Examples are presented in this paper to show that this consensus in the information is both necessary and potentially time consuming. This paper extends the basic implicit coordination approach to achieve better performance with imperfect data synchronization. The resulting robust decentralized task assignment method assumes some degree of data synchronization, but adds a second planning step based on sharing the planning data. The approach is analogous to closing a synchronization loop on the planning process to reduce the sensitivity to exogenous disturbances. Further simulations are presented to show the advantages of this method in reducing the conflicts in the assignments, resulting in improved performance compared to implicit coordination. These results also clearly demonstrate the effect of communication at the different stages of the planning algorithm on the overall mission performance.**

I. Introduction

To ensure scalability and flexibility of high-level control systems, various distributed architectures have been developed for the task assignment process [1], [2], [3]. These include centralized, distributed, and hierarchic. Within specific distributed frameworks (*i.e. implicit coordination*), each vehicle determines their own mission by simultaneously choosing tasks for all vehicles in the fleet using a centralized planning algorithm [4]. It is typically assumed that each vehicle then executes its own plan. To ensure consistency, information is shared to update the situational awareness (SA) and to negotiate on the designed plans [5], [6]. Note that the list of vehicles included in this calculation could be severely constrained to reduce the computation/communication required to plan for all other vehicles.

Hierarchic approaches typically assume the formation of sub-teams that use locally dense communication networks to share information (states, measurements, and plans). Communication between sub-teams would be limited, although it is assumed to be available if necessary to exchange resources. Tasks can be selected by the sub-teams or allocated to the group by a coarse scheduling algorithm run at a higher-level and then assigned within the sub-team by a leader. To maintain flexibility, the sub-teams are assumed to be "dynamic" and assets can be assigned to other sub-teams by the scheduling algorithm.

These two approaches reduce the reliance on a central planner system, thereby increasing the rate that the system can react to pop-up threats and/or targets of opportunity, increasing the robustness to failure, and ensuring that the control system degrades gracefully. However, it is essential that these distributed control decisions be well coordinated to maintain good overall performance. A key problem is that achieving tight coordination typically requires that the vehicles exchange large quantities of information about the environment, their current states, and their future intentions. Communication on this scale will not always be possible and it also increases the visibility of the vehicles to threats.

Constraining the communication limits situational awareness, which raises two key issues: first, that decisions must be made based on incomplete information and second, that information may be inconsistent across the fleet, potentially leading to less cooperative behavior. Thus one of the primary challenges is that these high-level algorithms must be modified to make them much less reliant on having "perfect, global" situational awareness while still obtaining reasonable performance. For example, a UAV may be uncertain of the distant terrain but able to plan anyway, since another UAV has greater awareness of that region and will be responsible for tasks within it. While it is intuitive that such a scheme could perform very well with limited communication and global awareness, the exact nature of the resulting performance degradation is not well understood. This paper will investigate this question and tackle the underlying problem of algorithmically identifying the relative significance of information.

Distributed planning is a challenging problem with uncertainty in the vehicles' SA, and it is made even harder when each vehicle has limited knowledge of the SA of the other vehicles in the team. This uncertainty can be reduced to some extent by communicating to share information. Important questions here are to determine which vehicles to communicate with, what data to exchange, and how to balance the effort between communicating input data (SA) or output data (control plans). These questions are driven by the conjecture, based on observations, that much of the information that could be exchanged does result in small changes in the control solution, but does not significantly impact the actual performance. The goal is to avoid this type of inefficiency and focus on only exchanging the data that will have the

0-7803-9568-9/05/$20.00 ©2005 IEEE

largest impact on the performance of the closed-loop system. Refs. [7] investigate a similar reduction in the information flow using *channel filters* for *distributed estimation*. Our problem is very similar, but based on the observation above, a more **control-centric** view of the information exchange must be developed to establish what information will have the largest impact on the closed-loop performance.

The outline for the paper is as follows: Section II provides background on the UAV task assignment problem and the petal algorithm, which is the basis for the distributed planning algorithm introduced in this paper. It also introduces the implicit coordination method, which is used as a benchmark. Section III presents the new decentralized task assignment algorithm, and simulation results are given in Section IV.

II. Background

This section defines the UAV task assignment problem and establishes the basis for the new Robust Decentralized Task Assignment (RDTA) approach to this problem. The RDTA is based on the petal algorithm [2], [8].

A. Petal Algorithm

In using these algorithms (petal and RDTA), several assumptions are made. First, the set of tasks have been identified for each team of UAVs. Second, the tasks have been divided between the team of UAVs and the waypoints for each team have been identified. The location of the waypoints are represented by a $N_w \times 2$ matrix B. Waypoint scores (values) are represented by the vector $S = [s_1 \ \ldots \ s_{N_w}]$. Each team is made up of N_v UAVs with known starting points, speed, and capability (*i.e.,* strike, reconnaissance, etc.). The UAV capabilities are represented by the $N_v \times N_w$ binary matrix K; $K_{vw} = 1$ represents a UAV v capable of performing the task associated with waypoint w. It is also assumed that there are polygonal "No Fly Zones" in the environment.

Given this information, the problem is to assign the UAVs to the waypoints to optimally fulfill a specific objective. There are several possibilities for specifying this objective, depending on the problem type. The most common objectives are to minimize mission completion time or to maximize the accumulated time-discounted score. This paper uses the time-discounted score formulation, which captures the notion of target score and is more realistic for real world problems. The score-based objective function can be written as

$$\max J_1 = \sum_{w=1}^{N_w} \lambda^{t_w} s_w \tag{1}$$

where s_w is the score associated with the task at waypoint w, and t_w is the time when waypoint w is visited. $0 \leq \lambda \leq 1$ is a discount factor that accounts for the decrease in target score with time. This factor is included in the objective function to better represent the real-world problems in which the score of visiting a target decreases proportional to the time in which it is visited. For example, consider the case of mobile Surface to Air Missile (SAM) sites. Once the enemy finds out that their SAM sites have been identified, they will typically start moving the missiles to a new location and, as a result, the task at the original waypoint loses score over time.

The algorithm developed for this approach can be explained as follows. First, a list of all un-ordered feasible task combinations are enumerated for every UAV, given its capabilities. Next, the length of the shortest path made up of straight line segments between the waypoints and around obstacles is calculated for all possible order-of-arrival permutations of each combination (these permutations are referred to as petals). The construction of these paths can be performed extremely rapidly using graph search techniques [2]. The time of visit for each waypoint w, t_w is estimated by dividing the length of the shortest path to that waypoint by the UAV's maximum speed. The time-discounted score is consequently calculated for each waypoint in each petal. The time-discounted score for each petal is the sum of the discounted score of its waypoints.

The algorithm produces two matrices whose p^{th} columns, taken together, fully describe one permutation of waypoints. These are matrix $\mathbf{V}$, whose V_{ip} entry is 1 if waypoint i is visited by petal p and 0 if not and vector $\mathbf{S}$, whose element S_p is the time-discounted score of the petal p. This procedure is described in detail in [2].

Once the approximate scores for the petals are calculated, a mathematical method is developed for allocating the waypoints to each UAV based on these scores and other constraints. The base of the task allocation problem is formulated as a Multidimensional Multiple-Choice Knapsack Problem (MMKP) [9]. The "knapsack" in this case is the complete mission plan. The $\mathbf{V}$ column corresponding to each of the N_M petals makes up the multi-dimensional weight. The "multiple-choice" comes from choosing which petal to assign to each of the N_v different UAVs (sets). The objective is to assign one petal (element) to each vehicle (set) that is combined into the mission plan (knapsack), such that the score of the mission (knapsack) is maximized and the waypoints visited (weight) meet the constraints for each of the N_w dimensions. The problem is given by

$$\begin{aligned} \max \ J_2 = & \sum_{p \in \mathcal{M}} S_p x_p \\ \text{subject to} \qquad & \forall i \in \mathcal{W} : \sum_{p \in \mathcal{M}} V_{ip} x_p \leq 1 \\ & \forall v \in \mathcal{V} : \sum_{p \in \mathcal{M}_v} x_p = 1 \end{aligned} \tag{2}$$

where $\mathcal{M} = \{1, \ldots, N_M\}$, $\mathcal{M}_v \subseteq \mathcal{M}$ are the petals that involve UAV v, and $\mathcal{W} = \{1, \ldots, N_w\}$ is the list of waypoints. The binary decision variable $x_p = 1$ if petal p is selected, and 0 otherwise. The objective in this problem formulation maximizes the sum of the scores to perform each selected petal. The first constraint enforces that each waypoint i is visited at most once. The second constraint prevents more than one petal being assigned to each vehicle. The solution to the MMKP selects a petal for each vehicle.

The problem is now a Mixed-Integer Linear Programming (MILP) problem that can be solved using CPLEX [10]. The solution to the task allocation problem is a set of ordered sequences of waypoints for each vehicle, which ensure that each waypoint is visited the correct number of times while maximizing the mission expected score.

B. Implicit Coordination Assignment

This section discusses the implicit coordination method and points out some of its shortcomings. A new methodology is further developed to overcome these shortcomings.

The idea of implicit coordination is to replicate the centralized assignment in each UAV [1]. In this method, each UAV, plans for all the UAVs in its team based on its own information and the map of the environment. It then implements its own plan. The premise is that UAVs have the same information and use the same algorithms and objectives to plan. As a result, the plans are the same and similar to the case of centralized planning. Hence each UAV can argue that it has the optimal, feasible plan for itself and this plan is consistent with the other UAVs. With these assumptions, one can assume that there will be no conflicts between the plans executed. In reality, however, reaching consensus and having exact information and consistent map of the environment is not always possible. The environment can change rapidly and UAVs update their map and information set and this causes the mismatch in the information. UAVs must communicate in order to keep the information consistent. Relying on a perfect high bandwidth communication structure makes the implicit coordination method very fragile. Examples in Section IV demonstrate this fragility. Even with no limit on the amount of data that could be communicated between UAVs, the system could still fail as a UAV loses its communication with the team. The lack of robustness in the implicit coordination comes from the assumption of consistent information. In order to resolve this shortcoming, an algorithm has to produce consistent plans without the need for perfect consistency of information. In the next section, the implicit coordination is modified to remove this constraint and produce a robust distributed planning algorithm for UAVs with imperfect communication structure.

III. Robust Decentralized Task Assignment

In the implicit coordination method, each UAV assumes that once it generates the plan, it is consistent with the other UAVs and therefore it is executed. If the plans are not consistent, then there could be conflicts and the overall plan might be infeasible. Of course, further communication of the information can be performed to develop consensus across the UAV fleet. However, with the sensitivity of the planning process to the input data, this process can take a large number of iterations and still does not guarantee reaching a feasible plan. To avoid the conflicting cases, the UAVs need to communicate their plans and resolve any possible infeasibilities. This can be interpreted as adding a "feedback loop" to the planning phase. By a similar analogy, the implicit coordination is essentially an "open-loop" control system that can be strongly influenced by exogenous disturbances. As with standard systems, closing a feedback loop can help improve the overall performance and robustness.

The robust decentralized method addresses this issue by dividing the planning into three stages. The first and second stages are similar to the implicit coordination method - each UAV communicates to other UAVs to reach a degree of consensus and then solves the assignment problem for all of the UAVs, as is done in the centralized assignment. But instead of generating one single optimal plan for itself, it generates a set of good (including the optimal one) plans. Each UAV then communicates its set of plans to other UAVs. After receiving the plans from other UAVs, in the third stage, each UAV has a set of plans for all of the UAVs in the fleet, which can be used to generate the best feasible plan by solving the task assignment again. The key difference here is that the set of information that forms the basis of the final planning is the communicated set of good plans. Therefore all of the UAVs have the same set of information and hence if they execute the same task assignment algorithms (same criteria and objectives), they would all generate consistent plans for the fleet. The following describes each stage of the RDTA in more detail.

A. First Stage–Updating Information (Reaching Consensus)

In the first stage of the planning, the UAVs communicate their information and iterate to try to reach consensus [5], [6]. This section presents a simple methodology for the UAVs to reach consensus in the information phase. The effectiveness of this method (in general communicating raw data) is compared to the effectiveness of communication in the planning phase in section IV-B.

Formulation: If the information, I_i is updated continuously, then the update law can be written as

$$\dot{I}_i = f_i(I_1, \cdots, I_{N_v}, G(t)) \tag{3}$$

where $G(t)$ represents the communication network. With the assumption that the communication network is not time-varying, a linear update scheme can be written as

$$\dot{I}_i = \sum_{j=1}^{N_v} \sigma_{ij} G_{ij} (I_j - I_i) \tag{4}$$

Where σ_{ij} are positive constants that represent the relative confidence of UAV$_i$ to UAV$_j$ about their information. G_{ij} is 1 if there is a direct communication link from UAV$_i$ to UAV$_j$ and zero otherwise. To implement this idea, a discrete equivalent of this formulation is needed. The simple linear discrete form of the filter can be implemented as

$$I_i(t+1) = I_i(t) + \sum_{j=1}^{N_v} \sigma_{ij} G_{ij} (I_j(t) - I_i(t)) \tag{5}$$

The simulations in this paper assume that the communication network is strongly connected, which means there is a path from every UAV to every other UAV.

B. Second Stage–Generating the Set of Best Plans

In this stage, the UAVs use their updated information to generate a set of feasible plans. The petal algorithm described in section II-A is a good candidate here, since it performs an optimization based on pre-generated feasible plans. A modified version of this algorithm is used in this stage, where each UAV, instead of generating one plan for itself, must generate a set of best plans. This can be implemented in different ways, as outlined in the followings.

In the first method, one optimization is solved to create all the plans at once. Assume for UAV$_i$, the set of generated petals for each UAV$_j$ is called P_{ij}, where

$$P_{ij} = \{p_{ij}^1, p_{ij}^2, \cdots, p_{ij}^{N_{ij}}\} \tag{6}$$

Similar to the petal algorithm, the idea is to pick a petal from each set, $P_{ij}, j \neq i$ in order to optimize the objective function while satisfying all of the constraints. For the set P_{ii} instead of one petal, a total of ρ_i petals are needed. The new optimization then can be written as:

$$\max_{x_{ij}^k} J_3 = \sum_{j=1}^{N_v} \sum_{k=1}^{N_{ij}} \quad x_{ij}^k . S_{ij}^k \tag{7}$$

$$\text{subject to} \quad \sum_{k=1}^{N_{ij}} x_{ij}^k = 1 \quad ; \quad \forall j \neq i \tag{8}$$

$$\sum_{k=1}^{N_{ii}} x_{ii}^k = \rho_i \tag{9}$$

$$(\sum_{j\neq i} \sum_{k=1}^{N_{ij}} x_{ij} p_{ij}^k) + x_{ii}^r p_{ii}^r \leq \mathbf{1}; \qquad \forall r \in \{1, \cdots, N_{ii}\} \tag{10}$$

where p_{ij}^k and S_{ij}^k are the k^{th} petal and its associated score for UAV$_j$ generated by UAV$_i$. x_{ij}^k is a binary variable, which is one if the associated petal is selected and zero if not. **1** in constraint (10) is a one vector. The result of this optimization is a set of ρ_i petals for UAV$_i$,

$$\mathbf{V}_i = \{p_{ii}^{k_1^*}, \ldots, p_{ii}^{k^*_{\rho_i}}\}$$

and a single petal for UAV$_j$, $p_{ij}^{k^*}, j \neq i$. In this formulation, each of the ρ_i petals in $\mathbf{V}_i$ can be combined with the petals selected for other UAVs, $p_{ij}^{k^*}$ to create a feasible plan for the fleet (constraints (8)-(10) ensure the feasibility of these plans).

In the second method, instead of solving one optimization to create the complete set, one optimization is solved for each petal. This method is essentially solving the original petal problem in (2), ρ_i times. After generating the set of all feasible petals for all the UAVs, the first optimization will generate the optimal plan for all the UAVs. This plan includes one petal for UAV$_i$, $p_{ii}^{k_1^*}$. In the second optimization, the same sets of feasible petals for all the UAVs except for UAV$_i$ is used. For UAV$_i$ the set

$$P_{ii} - \{p_{ii}^{k_1^*}\} \tag{11}$$

is used, which has all the original petals except the one that was selected in the previous optimization. The same optimization problem is then solved to generate a new set of plans. This process is repeated ρ_i times to generate a set of ρ_i good plans for UAV$_i$. This method is used in the simulations presented in this paper.

Each UAV then communicates this set of ρ plans to all other UAVs. The communicated information includes the selected petals,$\mathbf{V}_i$ (binary vectors) and the scores associated with these petals.

C. Third Stage- Generating the Final Feasible Plans

Each UAV$_i$ having the set of good plans for all the UAVs, $\mathbf{V}_j$'s, solves a simple petal algorithm to generate the final feasible plan for the fleet. Each UAV$_i$ then implements the petal p_{ii}^* that results from the optimization

$$\max_{x_{ij}^k} J_4 = \sum_{j=1}^{N_v} \sum_{k \in \mathcal{M}_j} \quad x_{ij}^k . S_{ij}^k \tag{12}$$

$$\text{subject to} \quad \sum_{k=1}^{N_{ij}} x_{ij}^k = 1; \forall j \tag{13}$$

$$\sum_{j=1}^{N_v} \sum_{k \in \mathcal{M}_j} x_{ij}^k p_{ij}^k \leq \mathbf{1} \tag{14}$$

where $\mathcal{M}_j = \{k_1^*, \ldots, k_{\rho_j}^*\}$.

The RDAT algorithm has several important features that improve the performance. First, the third phase is done based on consistent, pre-generated, plans, which ensures that there are no conflicts in the final plans selected independently. Second, since each UAV will execute a plan from the candidate set that it created, it is guaranteed to be feasible for that vehicle. Furthermore, communicating a small set of candidate plans helps overcome any residual disparity in the information at the end of stage 1 (consensus). This improves the possibility of finding a group solution that is close to the optimal, while avoiding the communication overload that would be associated with exchanging all possible candidates.

IV. Simulation Results

A. Advantages of RDTA Over the Implicit Coordination

The first set of simulations were designed to demonstrate the shortcoming of the implicit coordination and advantages of robust coordination over the implicit coordination. A simple example of 3 UAVs with 8 targets where each UAV is capable of visiting at most 2 targets is used. For simplicity, all targets are assigned the same score. It is also assumed that all the UAVs are capable of visiting all the targets. In the first run, the implicit coordination method is implemented. In this case, all the UAVs have consistent information and the result is the same as the centralized assignment (Figure 1).

In the second run, the same algorithm is used, but the data is perturbed so that the UAVs have inconsistent information to develop their plans. There are different attributes of targets that can be altered such as their type, score, and position. In this case only the positions of the targets are changed. A random number is added to the position of targets for each UAV. The random number is generated with a uniform

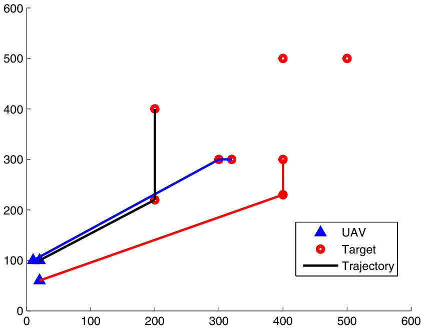

Fig. 1. Optimal Plan resulting from consistent information

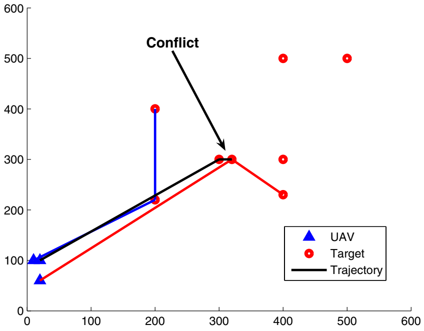

Fig. 2. Plan with conflicts resulting from inconsistent information

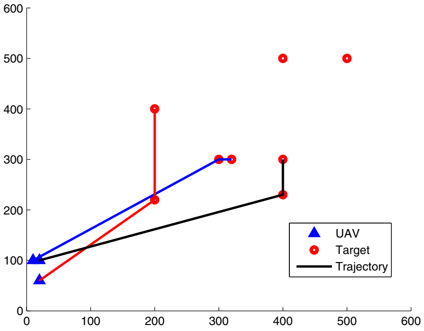

Fig. 3. Suboptimal plan with inconsistent information. No conflict result from RDTA

distribution in the interval of $[-30\%, +30\%]$ of the value of the position. Each UAV then has its own version of the information, which is inconsistent with other UAVs.

Figure 2 demonstrates the result for a case when the UAVs have conflicting assignments. Note that conflict here is defined as an assignment in which two or more UAVs are assigned to the same target. The same problem is also solved using the robust decentralized task assignment algorithm introduced in section III. Here, $\rho_i = 2, \forall i \in \{1, 2, 3\}$. The result is presented in Figure 3, which shows that using the RDTA for this example and only communicating two petals per UAV can eliminate the conflicts that appeared in the implicit coordination solution.

To better show the advantages of robust decentralized assignment over the implicit coordination, Monte Carlo simulations were used. In these simulations there are 5 UAVs that are supposed to visit 10 targets. The position of the UAVs are randomly generated for each simulation, but the knowledge of this location by the other UAVs is perturbed with noise to make it inconsistent. Figures 4 and 5 show the result of these simulations. Figure 4 shows the average number of conflicts versus the size of the communicated petal set, ρ. For the case $\rho = 1$, on average 2.1 of the UAVs have conflicts with other UAVs. The number of conflicts decreases as the ρ increases and drops to zero at $\rho = 7$. To understand the result of these conflicts on the overall performance of the plan, Figure 5 shows the performance versus the size of the communicated petal set, ρ. Increasing ρ can have a large impact on the performance – it is almost doubled for $\rho = 7$ compared to $\rho = 1$, which is the implicit coordination algorithm.

B. Effect of Communication on the Performance

A simple scenario of 5 UAVs and 10 targets is used to show the effect of communication on the planning stage (see Figure 6). Again, most of the data is kept similar and the only part of information that is not consistent between the UAVs is the position of the targets. To make the positions different for every UAV, similar to Section IV-A, a random number is added to the position of each waypoint. These random numbers are different for each target and each UAV. The RDTA algorithm is used to solve these problems. In the first stage, the algorithm in section III-A is used to improve the consensus of the information across the UAV team. This information is then used to produce a robust feasible plan for the fleet. In the simulations, the communication networks are randomly-generated, strongly connected networks. This means that in each communication network, there is a path from each UAV to any other UAV.

To see the effect of communication in each stage, two parameters are varied in the simulations. In the first stage of the assignment (consensus), the convergence of the information is directly related to the number of iterations. The amount of communication is also a linear function of the number of iterations. Therefore by changing the number of iterations, the amount of information communicated in the first stage can be directly controlled. The second important parameter is the size of the petal set that is communicated in the planning phase (stage 2), ρ_i. The communication in this phase is also a linear function of this parameter and thus can be controlled as well. Figure 7 shows the result of these simulations. For each scenario the number of iterations was changed from 0 to 7 and for each case RDTA was run with different values of $\rho_i = 1, \ldots, 5$. The result shown is the average of 100 Monte Carlo simulations in which the position of the targets was chosen randomly. In this figure, the x-axis shows the number of iterations, the y-axis is the size of the petal set communicated, ρ_i and the z-axis is the performance of the algorithm (total score of the plan). The plot clearly shows that the performance of the algorithm increases as both parameters (number of iterations and the size of communicated petal set) increase.

To better show the relationship between performance and the communication in each phase, Figure 7 is transformed into Figure 8 in which the x- and y-axes are the communication in the information (consensus) and planning phases, respectively. Communication was measured using the following rules. In the information phase (stage 1), in each iteration, each UAV has to communicate its information about the position of all targets to other UAVs. Assuming that the position of each target has two dimensions, then two words of information must be communicated for each target. There are 10 targets in this example, so a total of $2 \times 10 = 20$ words must be communicated form each UAV. In the planning phase (stage 2), the petals and the score of each petal must be communicated. Each petal is a binary

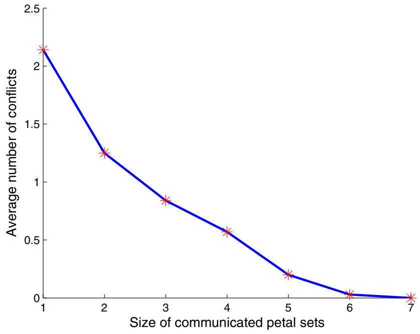

Fig. 4. Effect of communication in the planning phase (closing the loop in the planning phase) on the reduction of conflicts.

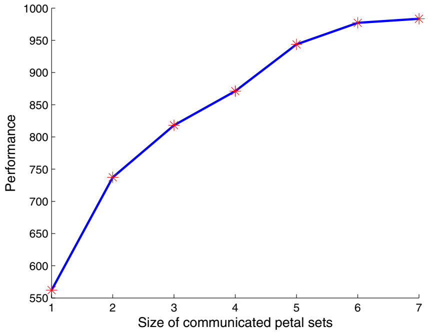

Fig. 5. Effect of communication in the planning phase (closing the loop in the planning phase) on the performance.

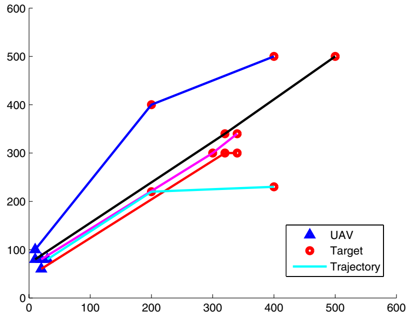

Fig. 6. Optimal plan for the scenario of 5 UAVs and 10 targets.

vector which can be interpreted to an integer number and transmitted as a word. Hence for each petal, each UAV must communicate a total of two words. And if the size of the communicated petal set is ρ, the total communication from each UAV in the planning phase is 2ρ.

The graph shows that increasing the communication in either axis (consensus or planning phase) improves the performance. However, the results also clearly show that communication in the planning phase is more efficient than in the information (consensus) phase in the sense that 8 words of communication in the planning phase has approximately the same effect on performance as 80 words in consensus phase. The plot also shows that to maximize the performance, some communication in both phases is needed.

V. Conclusions

The success of implicit coordination approach in which the central assignment algorithm is replicated on each UAV strongly depends on the assumption that all UAVs have the same situational awareness, and the examples showed that this consensus is necessary, but potentially time consuming. This paper presents an extension of the basic implicit coordination approach that assumes some degree of data synchronization, but adds a second planning step based on shared planning data. The resulting Robust Decentralized Task Assignment method uses these shared candidate plans to overcome any residual disparity in the information at the end of the (possibly truncated) consensus stage. The simulations demonstrated the advantages of this new method in generating feasible plans that reduced the conflicts in the assignments and improved the performance compared to implicit coordination. Further results demonstrated the effect of communication on the performance of assignment in different stages of the planning. A crucial part of the RDTA is choosing the best subset of plans to be communicated. Current research investigates different selection methods and the results will be presented in Ref. [11].

Acknowledgements

Research funded under AFOSR grant FA9550-04-1-0458.

References

[1] P. Chandler and M. Pachter, “Hierarchical Control for Autonomous Teams” proceedings of the AIAA *GNC,* Aug. 2001.

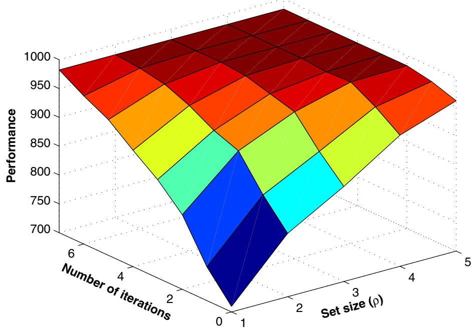

Fig. 7. The effect of two important parameters (iterations for consensus and size of communicated petal set) on the performance.

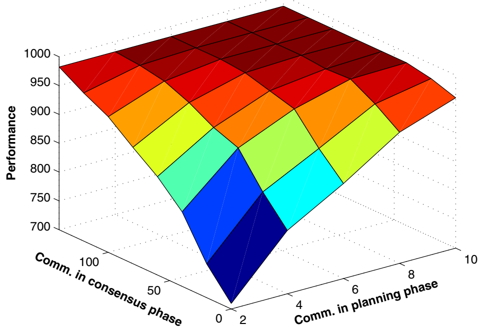

Fig. 8. The effectiveness of communication in the two phases of algorithm (consensus and loop closure in the planning) on the performance.

[2] J. Bellingham, M. Tillerson, A. Richards, J. How, “Multi-Task Assignment and Path Planning for Cooperating UAVs,” *CCCO,* Nov. 2001.

[3] E. King, M. Alighanbari, Y. Kuwata, and J. How, “Coordination and Control Experiments on a Multi-vehicle Testbed,” IEEE *ACC* 2004.

[4] P. Chandler, “Decentralized Control for an Autonomous Team,” *AIAA Unmanned Unlimited systems, Technologies, and Operations,* 2003.

[5] R.W. Beard and V. Stepanyan, “Synchronization of Information in Distributed Multiple Vehicle Coordinated Control,” IEEE *CDC,* 2003.

[6] R. Olfati-Saber and R.M. Murray, “Consensus Problems in Network of Agents With Switching Topology and Time-Delay ,” IEEE transaction on Automatic Control, Vol. 49, No. 9, Sept. 2004.

[7] B. Grocholsky, A. Makarenko, and Durrant-Whyte H. ”Information-Theoretic Coordinated Control of Multiple Sensor Platforms,” IEEE *International Conference on Robotics and Automation,* May 2003.

[8] G. Laporte and F. Semet, “Classical Heuristics for the Capacitated VRP,” in *The Vehicle Routing Problem,* edited by P. Toth and D. Vigo, SIAM, Philadelphia, 2002.

[9] M. Moser, D. Jokanovic, and N. Shiratori, “An Algorithm for the Multidimensional Multiple-Choice Knapsack Problem,” *IEICE Trans. Fundamentals*, vol. E80-A, pp. 582-589, Mar. 1997.

[10] ILOG, *ILOG CPLEX User’s guide*, 1999.

[11] M. Alighanbari and J. How, “Robust Decentralized Task Assignment for Cooperative UAVs” submitted to IEEE *ACC* 2006.