

Chapter 1

COOPERATIVE CONTROL FOR TARGET CLASSIFICATION

P. R. Chandler

Flight Control Division

Air Force Research Laboratory (AFRL/VACA)

Wright-Patterson AFB, OH 45433-7531

phillip.chandler@va.af.mil

M. Pachter

Department of Electrical and Computer Engineering

Air Force Institute of Technology (AFIT/ENG)

Wright-Patterson AFB, OH 45433-7765

mpachter@afit.af.mil

Kendall E. Nygard

Department of Computer Science and Operations Research

North Dakota State University

Fargo, ND 58105-5164

nygard@cs.ndsu.edu

Dharba Swaroop

Department of Mechanical Engineering

Texas A & M University

College Station, TX 77842-3123

dswaroop@mengr.tamu.edu

Abstract An overview is presented of ongoing work in cooperative control for unmanned air vehicles, specifically wide area search munitions, which perform search, target classification, attack, and damage assessment. The focus of this paper is the cooperative use of multiple vehicles to maximize the probability of correct target classification. Capacitated transshipment and market based bidding are presented as two approaches to team and vehicle assignment for cooperative classification. Templates are developed and views are combined to maximize the probability of correct target classification over various aspect angles. Optimal trajectories are developed to view the targets. A false classification matrix is used to represent the probability of incorrectly classifying nontargets as targets. A hierarchical distributed decision system is presented that has three levels of decomposition: The top level performs task assignment using a market based bidding scheme; the middle subteam level coordinates cooperative tasks; and the lower level executes the elementary tasks, eg path planning. Simulations are performed for a team of eight air

vehicles that show superior classification performance over that achievable when the vehicles operate independently.

Keywords: cooperative control, autonomous control

1. Introduction

The wide area search weapon system, as presently envisioned [1] has a number of air vehicles operating independently. The vehicles are released in a target area, and follow a set of waypoints that are preset at launch. If an object is detected in the sensor footprint, the vehicle tries to classify the object as a target. If the classification satisfies the criteria, the vehicle attacks the target and is destroyed. To maximize the probability of finding high value targets in a short period of time, cooperation among the vehicles has been proposed and work is ongoing in developing cooperative control algorithms. Cooperative search algorithms are being pursued in [2] where a cognitive map of threats, targets, and terrain is constructed using sensor inputs from all the vehicles. Cooperative classification algorithms are being developed by the authors that combine aspect angle dependent views of an object from multiple vehicles to maximize the probability of correct classification. Cooperative attack algorithms are being developed in [3] to ensure that sufficient weapons are engaged to ensure destruction of the target. The weapon target assignment problem is being addressed in [4] using dynamic stochastic programming and in [5] using dynamic network flow optimization models. Online optimal trajectory generation for cooperative rendezvous has been pursued by the authors [6] and others [7, 8].

Cooperative classification as discussed in Section 2 is the task of optimally and jointly using multiple vehicles to maximize the probability of correct target classification. This is shown in Fig. 1.1 where the arrows represent the velocity vectors of the vehicles. The vehicles can communi-

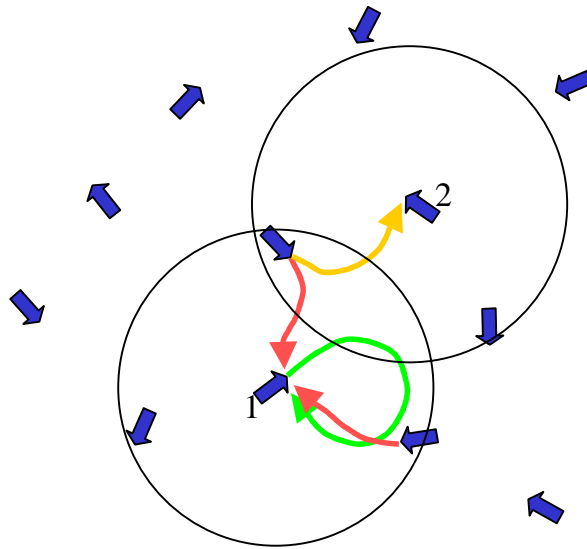


Figure 1.1. Target Classification Scenario

cate over some fixed range, which is represented by the large circles. The vehicle at 1 has detected a potential target, but in general the probability of classification $P_C < 1$ or some threshold. The

vehicle could perform a loopback maneuver or another vehicle could view the potential target at a different aspect angle. An approach to combining the views statistically is given in the next section. An important issue is choosing the optimal aspect angle for the second view. Initially, the second view was chosen to be orthogonal to the first.

In Fig. 1.1 the vehicle at 2 also has detected a potential target. The vehicle at 3 could view either potential target 1 or 2, or the vehicle at 4 could view potential target 1. Determining which vehicle could optimally provide the second view is the assignment problem. The optimization function could be, among other alternatives, to maximize the value of targets classified, or to minimize the time to classify. Two approaches to the assignment problem are given in Section 3.

The mission performance of wide area search munitions is quite sensitive to false target attack rate. This stems from the sensor used, the capability of the sensor processing algorithm, the number and type of objects in the search area, and whether the objects are partially hidden in clutter. The basic approach is to observe the object at the optimum aspect angle, as discussed in Section 2, as well as over the largest range of aspect angles, at minimum cost. Cost is defined as deduction from search time or attack tasks. The cooperative classification uses adjacent vehicles to maximize aspect angle ranges to achieve high probability of correct classification or low false target attack rate.

Section 4 discusses the development of a hierarchical architecture for cooperative search, classification, attack, and assessment. While many of these component functions are under development, the critical organizational theory of how to integrate the disparate and generally contradictory functions into a decision system has not been available. The three level hierarchy allows sub-teams to be formed dynamically at the midlevel. The top level uses a market analogy bidding procedure to assign vehicles and tasks to the sub-teams.

In Section 5, our Matlab/Simulink simulation is discussed. This high fidelity simulation has eight vehicles searching an area that has both targets and nontargets. The sensor processing is emulated, including false classification. While decisions are made concerning which task each vehicle is to perform, the coupling between the tasks is not completely accounted for. For example, the change in the search pattern if a vehicle is assigned a classification task.

Section 6 discusses many of the issues in cooperative classification that have yet to be addressed. The classification performance is also discussed. Section 7 presents the conclusions.

2. Joint Classification

The key technique for achieving a low false target attack rate is to use multiple views. A notional template is shown in Fig. 1.2 of probability of correct classification versus the aspect angle (θ) at which the object is viewed. This can also be looked at as a confidence level. False classification is addressed later in this section. To keep the occurrence of false classification low, the threshold is set high, in this case, $P_C > .9$ before the target can be attacked. As can be seen in Fig. 1.2, the threshold is achieved only over a narrow range of aspect angles (0 or 180 deg).

The objective is to combine the statistics from multiple views; in general, for two views:

$$P_C = P(\theta_1) + P(\theta_2) - P(\theta_1, \theta_2) \quad (1)$$

If the views are statistically independent:

$$P_C = P(\theta_1) + P(\theta_2) - P(\theta_1)P(\theta_2) \quad (2)$$

Initially it is assumed the views are uncorrelated. This assumption will be relaxed later in the section. From eqn. 2, one can see that if the object is viewed at the same θ , P_C increases.

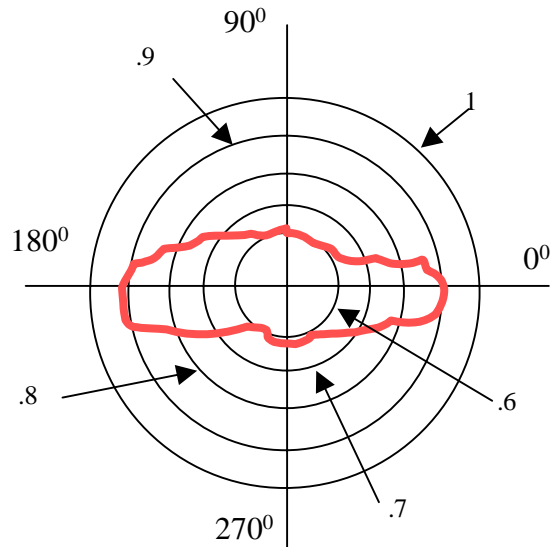


Figure 1.2. Notional Target Template

Intuitively, this is not reasonable, since there is no additional information. It is generally true that if the aspect angles are separated by 90 deg the views should be uncorrelated and the information content should be greater. Based on this insight, trajectories now need to be derived that result in views orthogonal to the first.

Fig. 1.3 shows a sample configuration for two vehicles. The **X** marks the location of an

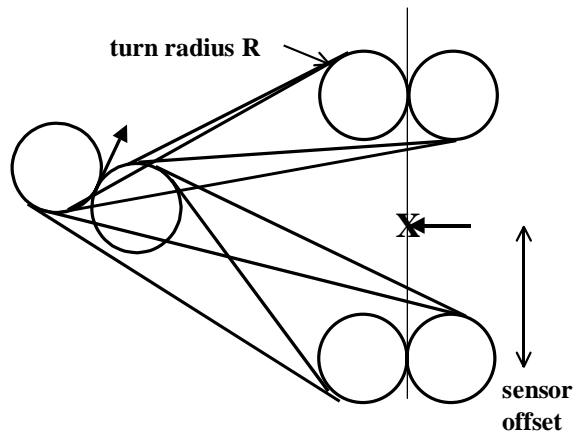


Figure 1.3. Minimum Time Trajectories

object and the arrow the velocity vector of the vehicle that detected the object. The other arrow represents the velocity vector of an adjacent vehicle that could provide a second view. As stated earlier, the simplification is that the second vehicle should come in ± 90 deg. Because the sensor looks ahead of the vehicle, the vehicle must be on the orthogonal line at least the distance of the sensor offset. The circles represent a specified minimum turn radius R .

It can be proven that the minimum time trajectory to a target consists of an initial turn through a circular arc, a straight line dash, and a turn through a final circular arc. The arcs are on the circles in Fig. 1.3. As can be seen, there are eight possible trajectories for the adjacent vehicle to place its sensor on the object. The approach pursued here, is to calculate the distances traveled for all eight trajectories. The shortest, of course, is the minimum time trajectory. It can be shown that this algorithm holds for any configuration of adjacent vehicle and object. Once the trajectories are defined, we return to the target templates.

A notional target is shown in Fig. 1.4. For simplicity, the target is rectangular with sides a , b

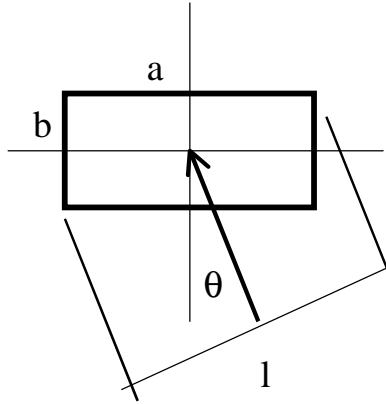


Figure 1.4. Target Projection Diagram

and θ is the aspect angle at which the target is viewed. The assumption is that the projected line length l is proportional to the probability of classification. Views at $-\pi, -\pi/2, 0, \pi/2, \pi$ contain projections from only one side, so that no estimate of aspect ratio can be made. The probability, or confidence level, is defined as being proportional to the length of the side that is viewed. The projected line length is normalized by the length $a + b$, where the maximum occurs at θ^* . The orientation of the target on the ground is defined by ψ and θ . The probability (projection) is:

$$P_c(\theta) = \begin{cases} \frac{a \cos \theta + b \sin \theta}{a+b} & \text{for } 0 \leq \theta \leq \pi/2 \\ \frac{-a \cos \theta + b \sin \theta}{a+b} & \text{for } \pi/2 \leq \theta \leq \pi \\ \frac{-a \cos \theta - b \sin \theta}{a+b} & \text{for } \pi \leq \theta \leq 3\pi/2 \\ \frac{a \cos \theta - b \sin \theta}{a+b} & \text{for } 3\pi/2 \leq \theta \leq 2\pi \end{cases}$$

The maximum projected line length occurs at $\theta^* = \arctan(b/a)$. The maximum value is:

$$P_c(\theta^*) = \frac{\sqrt{a^2 + b^2}}{a + b} \leq 1$$

Fig. 1.5 shows the periodic nature of $P_c(\theta)$, since a rectangle has 2 axes of symmetry. Finally, the plot is scaled by P_{max} , which in the figure is .8. If the threshold is .9, this means that it is not possible to classify the target from one view.

If n statistically independent views of the target have been taken, the probability of identifying the target is calculated as:

$$P_{CI} = 1 - \prod_{i=1}^n [1 - P_c(\theta_i)]$$

In the special case of $n = 2$, as before

$$P_{CI} = P_c(\theta_1) + P_c(\theta_2) - P_c(\theta_1)P_c(\theta_2)$$

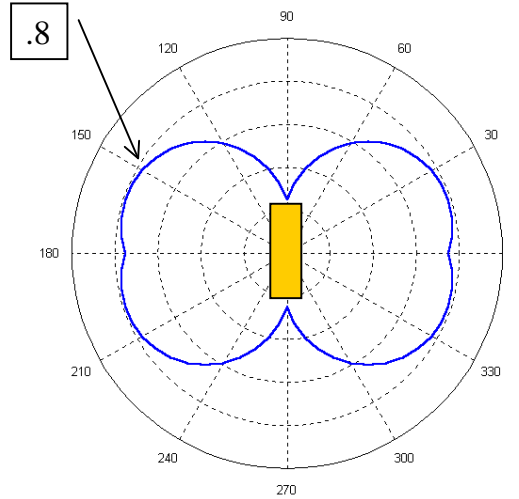


Figure 1.5. Probability versus aspect angle

The joint probability calculation given above is overly optimistic when the aspect angles are close. The exact joint probability for 2 views is not available, but it is reasonable that correlation $r = 1$ when $|\theta_2 - \theta_1| = \Delta\theta = 0$ and $r = 0$ when $\Delta\theta = 90^\circ$. Therefore, as an approximation, a blending function is defined as:

$$\rho(\Delta\theta) = 1 - e^{-.03|\Delta\theta|}$$

The modification to the 2 view probability for correlated views is as follows:

$$P_{CC} = P_{CI}(\theta_1, \theta_2) + (1 - \rho(\Delta\theta)) \left[P_c(\theta_1)P_c(\theta_2) - \frac{P_c(\theta_1) + P_c(\theta_2)}{2} \right]$$

This assumes the views are uncorrelated when $|\Delta\theta| \geq \pi/2$.

We now have an algorithm for generating classification probabilities, or more accurately, confidence levels, from two views of an object. For the autonomous munition, False Target Attack Rate is probably *the* critical factor in weapon system performance. Therefore, a reasonable emulation must include false classification as well. A notional false classification probability matrix is given in Table 1.1. When the vehicle detects a target, the class is selected according to the probabilities in

Table 1.1. False Classification Matrix

	C	L	A	S	S
		0	1	2	3
T	0	.91	.03	.03	.03
R	1	.04	.92	.02	.02
U	2	.04	.02	.92	.02
E	3	.04	.02	.02	.92

the table. The emulation enters the “selected” class template, not the “true” class template, at θ to get P_c . If $P_c < .9$, then another view is needed. If the second class is the same as the first, then proceed to combine views as above. If the combined statistic does not exceed the threshold, then the allocation process will determine if taking another view is cost effective. If the two classes are

not the same, then the view with the highest priority class could be retained for the assignment and the other view discarded. An alternative is to retain both views and let the assignment algorithm determine from the priorities if an additional view is warranted. Cooperative target classification is driven by inputs from the upper level of the hierarchical cooperative control system currently under development. In this overview paper, we outline in the following 2 sections the assignment algorithms and hierarchical control architecture.

3. Assignment

In general, the assignment problem involves not only classification, but also search, attack, and damage assessment. For purposes of illustration here, all of the vehicles are considered available to perform cooperative classification. The assignment algorithm then, is to select the optimal vehicle to provide the second view. An assignment method that includes the other tasks is addressed later in this section.

When an object is detected, the location, heading angle ψ , probability, and aspect angle θ is transmitted to all the other vehicles. The vehicles use Fig. 1.3 to determine distances and time to the object at, initially, angles perpendicular to ϕ . Later on, four angles to the object were used, these represent the best vectors to view the object from the template in Fig. 1.5. The calculated minimum time, distance, or cost to the object is then transmitted to the other vehicles. This is done for all the objects that need classification. The result is that all the vehicles have complete information and solution of the assignment problem is globally optimal. All the vehicles solve the same problem and therefore arrive at the same solution – conflicts are avoided and a degree of redundancy is achieved.

An example assignment matrix is given in Table 1.2. The columns are targets, the rows

Table 1.2. Assignment Matrix

	T ₁	T ₂	T ₃	T ₄
V ₁	C ₁	C ₂	C ₃	C ₄
V ₂	C ₅	C ₆	C ₇	C ₈
V ₃	C ₉	C ₁₀	C ₁₁	C ₁₂
V ₄	C ₁₃	C ₁₄	C ₁₅	C ₁₆
V ₅	C ₁₇	C ₁₈	C ₁₉	C ₂₀

are vehicles, and the entries are costs, for example: time to object; remaining life; distance; or a weighted target value. Each of these types of costs have been used in the simulations discussed in a later section. This is a straightforward linear assignment problem and can be put in an integer linear programming form. This is easily solvable, even on modest hardware, for many targets and vehicles. The matrix is completely dynamic. As new objects are found, all of the vehicles are optimally reassigned. Or, when classified, taken out of the assignment matrix. The objective could also be to maximize the vehicles remaining life or to maximize the value of objects classified. The next topic addresses assignment for all the tasks.

The assignment of vehicles to search, classification, attack, and battle damage assessment is posed as a network flow optimization model. The model shown in Fig. 1.6 is described in terms of supplies and demands for a commodity, nodes which model transfer points, and arcs that interconnect the nodes and along which flow can take place. Arcs can have capacities that limit the flow along them. An optimal solution is the globally greatest benefit set of flows for which supplies flow through the network to meet the demands. In the model, vehicles are supplies and the tasks

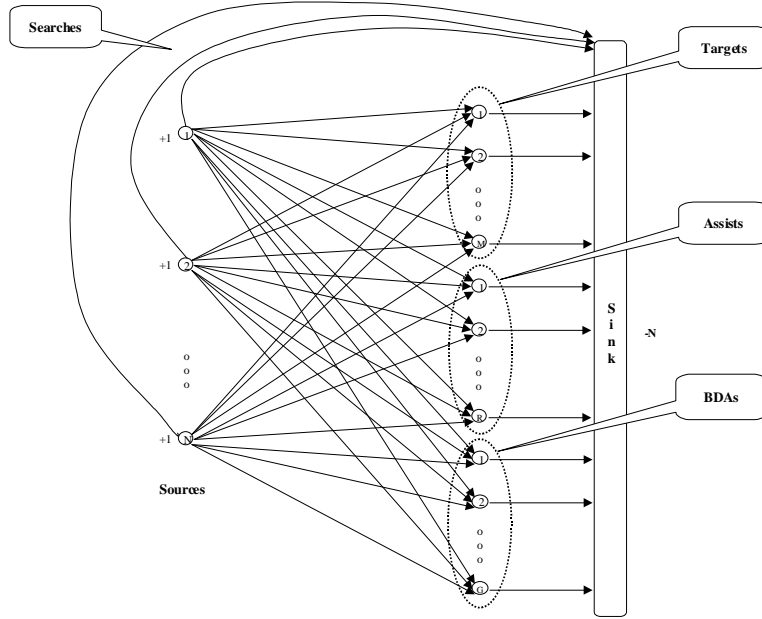


Figure 1.6. Network Flow Analogy

are demands. Since the vehicles are in only one mode at a time, the arcs have a flow of 0 or 1. Fig. 1.6 is also known as a Capacitated Network Trans-shipment Model and reduces to an integer (binary) linear programming problem. The linear program is formulated as follows:

$$Z = \max \sum_{i,j \in I, i \neq j} c(i,j)x(i,j) \quad (3)$$

Subject to:

$$\sum_{j \in I, i \neq j} x(i,j) - \sum_{k \in I, k \neq i} x(k,i) = 0 \quad i \in I \quad (4)$$

$$\sum_{(i,j) \in A} x(i,j) \leq b(i,j) \quad [(i,j)|i,j \in I, i \neq j] \quad (5)$$

$$x(i,j) \geq 0 \quad [(i,j)|i,j \in I, i \neq j] \quad (6)$$

where,

$c(i,j)$ = Expected value of vehicle i attacking target j ,

$c(i,k)$ = Expected value of vehicle i classifying target k ,

$c(i,g)$ = Expected value of vehicle i BDA target g ,

$c(i,s)$ = Expected value of vehicle i continuing to search,

the vector x is the binary decision variable, c are the benefits to be maximized, Eqn. 4 is the flow balance, and Eqn. 5 is the link or flow capacity.

As in the previous assignment problem, the solution is globally optimal. The LP problem has a specialized structure that is very fast to solve, is highly flexible, event driven, and dynamic. An important issue is the determination of the costs above. Determining the utility of continuing to search may be particularly difficult to calculate. Also, output of the nodes are restricted to 1 to maintain linear form, which means multi-vehicle attack of a target is not allowed. This restriction is relaxed in the next section, either by augmenting the matrix or by a process of “bidding”.

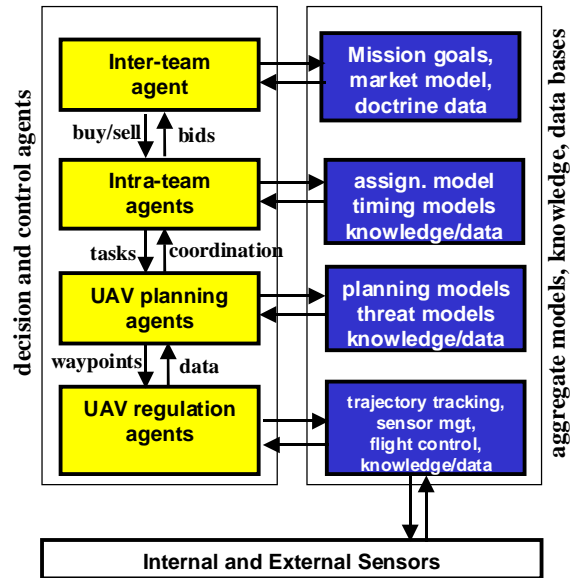


Figure 1.7. Hierarchical Decomposition

4. Hierarchical Architecture

Figure 1.7 illustrates a general architecture for cooperative control and resource allocation among multiple vehicles. The Inter-team Cooperative Planning agent is basically responsible for configuring teams of vehicles and providing them with their goals. This agent has visibility of the highest level goals for the overall mission, and its internal model codifies doctrinal information as it applies at this level. If teams are preconfigured before takeoff, this agent will primarily be responsible for determining if teams should be reconfigured as new information is received and situation awareness improves. The models it invokes are expected to request and receive information from the Intra-team Cooperative Control Planning Agents at the next lower level. Based on this information, this agent may autonomously abandon certain high-level goals in favor of others.

The domain of responsibility for an Intra-team Cooperative Control Planning Agent involves the division of responsibilities among the vehicles working as a configured team. Leadership responsibilities and coordination mechanisms depend on the mission, the models available to support accomplishing the mission, available data, and the current capabilities (eg fuel status) of the vehicles on the team.

The vehicle planning agents function specifically within the domain of an individual vehicle. These on-board planners accept a specific goal that is appropriate for a single vehicle, then invoke path planning and scheduling algorithms aimed at meeting the goal.

Finally, the vehicle Regulating Agents provide command sequences for the vehicle, in order to accomplish such tasks as following trajectories, activating sensors, executing maneuvers, changing speed, and releasing weapons.

At the Inter-team level is a high-level auction procedure [10] for determining which targets should be assigned to which team. We assume that an initial allocation of targets to vehicles has been made, and that each team has solved its own generalized assignment problem to determine which vehicles attack which targets, and the total expected value of the chosen decisions. Thus, a team derives value through its current assets, which are targets to strike, and vehicles to strike

them. To potentially improve the overall value among the teams, we now allow targets and vehicles to be “traded” from one team to another, in a way that simulates a stock exchange. When a team hypothetically gives up an asset, the following computations can be derived:

- For a specified target, the reduction in value to the team if the target is given to another team. This is the target “sell” value.
- For a specified vehicle, the reduction in value to the team if a vehicle is given to another team. This is the vehicle “sell” value.

Similarly, when a team acquires an asset, viz an additional target or vehicle, the following computations can be derived:

- The gain in value to the team if a specified target is received from another team. This is the target “buy” value.
- For a specified vehicle, the gain in value to the team if a vehicle is received from another team. This is the vehicle “buy” value.

The advantage of making a trade is guaranteed to be realized only if the trade is isolated from other trades involving the same teams, because the buy and sell values apply at the margin and the assigning of multiple vehicles to a target is inherently nonlinear.

The Intra-team level has agents that manage cooperative behavior, including: cooperative search, cooperative classification, cooperative attack, damage assessment, and rendezvous. Cooperative search consists of building maps of threats, targets, and terrain. As each of the vehicles uncovers information, it is transmitted to the other vehicles to build the maps. An optimization problem is solved to apportion individual vehicles to search areas that have the greatest probability of containing high value targets, while minimizing fuel and exposure. Cooperative classification has already been discussed. Cooperative attack stems from the probability of kill from an individual munition is less than one ($P_K \leq 1$). Multiple munitions may be needed to kill the target with sufficient confidence. Cooperative damage assessment is to ensure that high value targets have indeed been destroyed by viewing the target after attack. The rendezvous function is the time coordination of vehicles arrival at a target.

5. Simulation

A simulation was developed for up to eight vehicles cooperatively controlled in a wide area search and attack mission. The simulation is based on the Control Automation and Task Allocation (CATA) [9] simulation in C++. The simulation was converted to run under Matlab Simulink to expedite algorithm research. Much of the software is compiled C++ code that is incorporated into Simulink blocks. The research algorithms are coded using graphics or math script.

The simulation scenario entails eight vehicles searching a battle space that has six targets of various values and up to five nontargets. The vehicles are initially in a echelon formation and following a serpentine path. As targets are detected, vehicles are dynamically assigned to perform classification and attack. The search could be dynamically changed as vehicles are assigned so as to cover the areas that have the highest probability of containing a high value target. In this simulation, if a vehicle is reassigned back to search, it returns to the original serpentine path. All of the targets are found and attacked before the vehicles run out of fuel. No nontargets are attacked.

Fig. 1.8 shows a typical scenario. In Fig. 1.8 vehicle 2 detects target 2 first. Vehicle 5 is assigned to classify target 2. Vehicle 2 then detects target 6 and vehicle 3 detects target 5. Vehicle 6 is then assigned to classify target 5 and vehicle 7 is assigned to classify target 6. Then vehicle

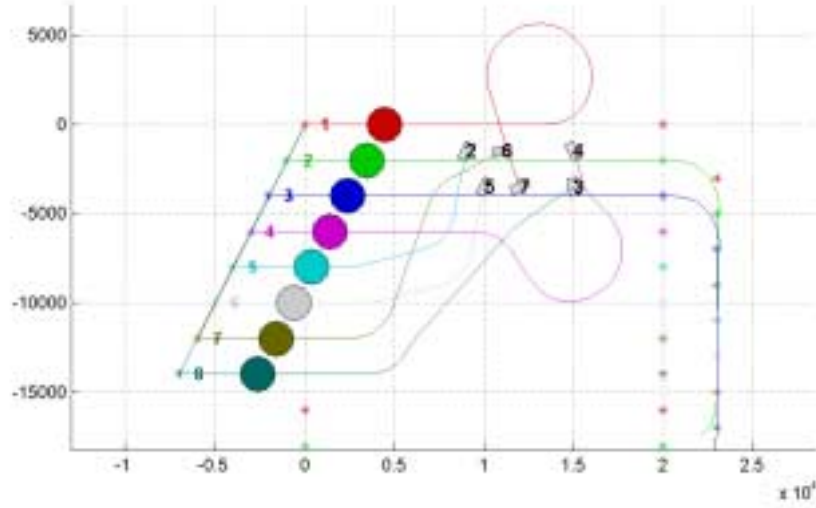


Figure 1.8. Scenario with 8 vehicles, 6 targets

3 detects target 7, which results in vehicle 8 being assigned to classify target 7. At this point, vehicle 2 detects target 4, which results in vehicle 4 being assigned to classify target 4. Vehicle 5 also detects target 5 on its way to classify target 2, but this does not trigger a reassignment. This is because vehicle 5 does not pass over the target close enough to the specified aspect angle. The fortuitious detections could be more optimally incorporated. Finally, vehicle 3 detects target 3, however, vehicle 8 is assigned target 3. This results in vehicle 1 being assigned to target 7, where vehicle 1 had not been previously assigned. Vehicles 2 and 3 continue on the serpentine path, while the other vehicles classify and attack their assigned targets. All of the vehicles cross over their assigned targets at the specified aspect angles and the classification threshold is crossed. All of the targets are of a high value, so the targets are attacked as soon as they are classified – there is no delayed attack.

Not shown, but if a false target attack rate and nontargets are introduced, all of the valid targets are attacked, but one of the nontargets is attacked. If the probability of correct classification threshold is raised, then the potential targets are viewed at more aspect angles. This prevents the nontargets from being attacked, but sometimes results in valid targets not being attacked. The developed simulation tool allows us to conduct a parametric study and thus optimally address this trade off situation.

6. Classification Issues

- 1 Aspect angle estimate $\hat{\theta}$. This estimate could be used to determine the 2nd optimum viewing angle. To date, 2nd view angles are based on the heading angle ψ of the first view, not the orientation of the object on the ground. The computation of the optimal 2nd view could be done offline for the finite set of templates. However, this does not mean that the classification

threshold will be crossed. Another offline optimization could be performed to determine the number and aspect angles of views to yield classification. Given that this information were available, the algorithms discussed previously could use it and allocate resources optimally.

- 2 Statistically combining 3 or more views. For a possible high value target, an arbitrary number of views may be desirable. The simplified joint probability approach presented earlier would have to be much more complex. Instead of including all the views at once, the calculation could be recursive. Calculate the joint probability for 2 views, use the best θ , add the 3rd view, etc.
- 3 How to account for clutter. To date, all the targets are assumed viewable from all angles with no objects obstructing the view. To emulate a target obstructed by a building on one side, one could scale the template on that side with a squashing function. For example, $P_C = P(\theta)(1 - \cos \theta)$, $-90 \leq \theta \leq 90$. Clutter, of course, has a large impact on the performance of the search algorithm to detect targets.
- 4 Classification mismatch. False Target Attack Rate (FTAR) stems most directly from the sensor and sensor processing. If on the 1st view the classification threshold is crossed for the wrong object, then cooperative classification cannot contribute. If the threshold is not crossed on 1st view, a 2nd view is then taken, and the classes are different, it is of course not possible to combine the statistics. However, the mismatch could be resolved by the optimization algorithm. Select one of the classes either based on probability of occurrence or target value. The optimization then determines whether resources should be assigned to classify the selected object. If so, then the class from the 3rd view should break the tie. The additional views contribute to reducing the FTAR.
- 5 Other issues – registration. Previous discussions assume the multiple views are of the same object. This is a function of the navigation precision versus object density for fixed targets. If the objects can move between views, then registration is more of a concern and contributes to classification mismatch. Finally, if a vehicle is pulled off of search to perform a classification or other task, what is the impact on the search strategy? This is especially critical in mission performance if the objects to classify are ultimately nontargets.

7. Conclusions

High fidelity simulations have been performed of eight vehicles in random and serpentine search patterns to detect, classify, and attack targets. Sensor processing is emulated using the target templates previously discussed. Work to date has focused on orthogonal 2nd views where the views are combined statistically. Minimum time maneuvers are used to view the potential target at the specified heading. The optimal assignment is based on minimizing the time to classify. Other metrics also used include: maximizing remaining life, and maximizing value of targets classified. The largest difference using these metrics is that maximizing remaining life resulted in delayed attacks until the vehicles were nearly out of fuel. Which of these functions are best in maximizing the probability of targets killed would come from a systems analysis study.

With communications and cooperative classification, fewer loop-back maneuvers are performed where the same vehicle performs the second view. This implies a more efficient utilization of resources. If the vehicles are in line formation where there is significant overlap in the sensor footprints, this results in extensive looping maneuvers to perform classification. Placing the vehicles in an echelon or staggered formation yields much more direct (efficient) classification trajectories.

The assignment techniques discussed are fast and globally optimal. The market approach to assignment becomes more useful as the number of vehicles increase; however, the benefit degrades as the transactions become more coupled.

Scenarios without coordination frequently result in valid targets not being found; cooperative classification successfully addresses this problem. Introduction of false classification can be countered with more emphasis on cooperative classification, but with some increase in the probability of not classifying valid targets.

Hierarchical cooperative control allows for near optimal solution of the large scale optimization problem. It is compatible with the prevailing information pattern in the air to ground attack scenario, and it is computationally efficient for dynamic replanning.

Acknowledgments

The authors wish to thank Lt. Col David Jacques and Dr. Robert Murphey for their contributions and support. The authors also wish to thank their colleague Steven Rasmussen for his technical support.

References

- [1] Low Cost Autonomous Attack System briefing, approved for public release, AFDTC/PA, 24 Mar 99.
- [2] Passino, Kevin M., Marios M. Polycarpu, David Jacques, and Meir Pachter, “Distributed Cooperation and Control for Autonomous Air Vehicles”, *ibid.*
- [3] Gillen, Daniel P., and David R. Jacques, “Cooperative Behavior Schemes for Improving the Effectiveness of Autonomous Wide Area Search Munitions”, *ibid.*
- [4] Murphy, Robert A., “An Approximate Algorithm for a Weapon Target Assignment Stochastic Program”, in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, Kluwer Academic Publishers, 1999.
- [5] Nygard, Kendall E., Phillip R. Chandler, and Meir Pachter, “Dynamic Network Flow Optimization Models for Air Vehicle Resource Allocation”, submitted to American Control Conference 2001.
- [6] Chandler, P., S. Rasmussen, M. Pachter, “UAV Cooperative Control”, AIAA GNC 2000, Denver, CO, Aug 2000.
- [7] McLain, T., “Cooperative Rendezvous of Multiple Unmanned Air Vehicles”, AIAA GNC 2000, Denver, CO, Aug 2000.
- [8] Bortoff, S., “Path Planning for UAVs”, AIAA GNC 2000, Denver, CO, Aug 2000.
- [9] “Control Automation and Task Allocation”, AFRL Final Report, Boeing, 1997.
- [10] M. Wellman, and P. Wurman, “Market-aware Agents for a Multiagent World”, *Robotics and Autonomous Systems*, 24:115-125, 1998.