49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Cooperative Receding Horizon Control for Multi-agent Rendezvous Problems in Uncertain Environments

Chen Yao, **Xu Chu Ding** and **Christos G. Cassandras**
Division of Systems Engineering
and Center for Information and Systems Engineering
Boston University
Brookline, MA 02446
cyao@bu.edu, xcding@bu.edu, cgc@bu.edu

Abstract— **We consider the problem of controlling vehicles to visit multiple targets and obtain rewards associated with each target with the added requirement that two or more vehicles are required to be present in the vicinity of each target in order to collect the associated reward. The environment is uncertain and targets may appear or disappear in real time. In view of the problem complexity, we focus on an optimization-based algorithm that makes decisions driven by available information. We develop a cooperative receding horizon controller to maximize the total reward obtained in a finite mission time horizon. We control the motion so as to maximize the expected rewards over a planning window, and show that the resultant trajectory is stationary in the sense that vehicles converge to targets without explicit target assignment.**

I. Introduction

Multi-agent cooperative control and coordination has become an important and growing field in recent years. Advances in robotics, communication, sensors and computer hardware have fueled this trend, and provide the enabling technologies for achieving cooperative control in multiple-vehicle systems. Cooperation among vehicles (more generally, agents) allows complex tasks to be solved more efficiently by a team, and generally results in a more robust solution for vehicles operating in an uncertain environment. Heterogeneity in terms of capability among agents allows a team to complete tasks otherwise impossible for a single agent, and is key to success for many unmanned missions (see, for example, [4], [6], [8])

One class of problems in cooperative control involves M vehicles which must visit N targets and collect the rewards that are associated with each target. The rewards are time-varying, and targets may emerge or disappear in real time. One can treat this problem as a stochastic optimal control problem [15], which leads to dynamic programming based approaches [5], [18]. Since this approach quickly becomes computationally intractable, it is common to separate the problem into two sub-problems: target assignment which assigns targets to vehicles, and path planning which generates trajectories for individual vehicles. The first problem can be viewed as a version of the vehicle routing problem, which is prominent in operations research [16]. This problem is NP-complete, and as such, there is much ongoing research to address it in an uncertain environment where information is provided in real time (for a survey, see [7]). The second problem involves cooperative path planning, which is an important problem in motion planning of robotic agents (see for example, [2], [12]).

A different approach was proposed in [10] (with a distributed version in [11]). Instead of "functional decomposition" by separating the problem into sub-problems, "time decomposition" was used. This approach aims at developing an optimization-based controller that maximizes the total expected reward accumulated by the agents over a planning window, then continuously extending this window forward. This method constitutes a receding horizon (RH) approach, and is similar to approaches used in the model predictive control community [13]. It avoids the combinatorial explosion in other approaches because there is never explicit vehicle-to-target assignment. Rather, in order to deal with random events, the controller employs a "hedge-and-react" strategy, in contrast to the traditional "estimate-and-plan". An interesting advantage of this approach is that the optimal headings generated by the receding horizon controller were shown in [11] to drive vehicles to targets without explicit assignments.

The previous setting does not model cooperation of vehicles that might be required to jointly achieve collection of reward. In many applications, reward collection translates to accomplishing some tasks at a fixed location, such as providing surveillance, servicing a customer or destroying a target. In many cases, due to heterogeneous capabilities among the vehicles, these tasks require multiple vehicles to be present near a location at the same time. From a resource point of view, each vehicle may carry different resources, and achievement of the task depends on resources from different vehicles. For example, in a search and rescue mission, Unmanned Aerial Vehicles (UAVs) may provide aerial views of the terrain and interact with Unmanned Ground Vehicles (UGVs) that traverse the terrain and complete the mission.

To address this problem, in this paper we aim to control a team of agents to collect rewards associated with targets,

The authors' work is supported in part by NSF under Grant EFRI-0735974, by AFOSR under grants FA9550-07-1-0361 and FA9550-09-1-0095, by DOE under grant DE-FG52-06NA27490, and by ONR under grant N00014-09-1-1051.

and each target requires two vehicles to be within a certain range from the target when the "reward harvesting" action is performed. This problem is called "multi-agent rendezvous" because agents are required to rendezvous at targets. We can again decompose this problem into sub-problems consisting of an assignment problem that requires rendezvous and a trajectory planning problem that achieves simultaneous meeting at a target point (addressed in [1],[14],[17]). When the information of the rewards becomes uncertain due to randomly emerging targets, computational complexity associated with the assignment problem quickly drives this approach infeasible.

To counter this, we use a similar idea as in [10],[11]. In this framework, we propose a cooperative receding horizon controller that computes the optimal headings of the vehicles at the end of each *planning horizon*, such that the total *expected reward* obtained by the team is maximized (assuming no target emerges on the way during the planning horizon.) This heading is then executed for a shorter *action horizon*, unless a new target is detected, in which case the optimization is performed again. If there is no new target, then the optimal headings will be recomputed at the end of the action horizon. Even though no explicit task assignment is performed, we show in this paper that the proposed controller drives vehicles to targets. Hence, the trajectory generated by the controller is "stationary" in the sense that the vehicles converge to some targets even though the control decision is made in real time with no explicit assignment involved. Furthermore, this controller integrates the problems of task assignment and trajectory generation, both in an uncertain environment. In this paper we prove the stationarity property for the case of 2-vehicles and 1-target, leaving the general case for future work.

The rest of the paper is structured as follows. Section II formulates the setting and presents the cooperative receding horizon controller for the rendezvous problem. Section III provides some simulation results of the controller in action, and compares them to an approximate reward upper bound achieved by solving the assignment problem. Section IV defines a stationarity property and proves the stationarity of the CRH controller. Section V concludes the paper.

II. COOPERATIVE RECEDING HORIZON (CRH) CONTROLLER FOR THE RENDEZVOUS PROBLEM

We consider vehicles to be operating in a two-dimensional mission space. Assume that the mission is to collect rewards from N targets using M vehicles. Let $x_i \in \mathbb{R}^2, i = 1,2,...,M$, and $y_j \in \mathbb{R}^2, j = 1,2,...,N$ denote the positions of vehicles and targets respectively. Note that a target may change its location during operation of the system, or new targets may show up, hence N and y_j may not be constant. At the same time, a vehicle may malfunction, and thus M may also change in time.

We assume that the vehicles operate with constant velocity following the dynamics:

$$\dot{x}_i(t) = V_i \begin{bmatrix} \cos u_i(t) \\ \sin u_i(t) \end{bmatrix} \qquad x_i(0) = x_{i0} \tag{1}$$

Each vehicle is controlled by the heading $u_i(t)$, and V_i is the velocity.

A. Model of Reward Collection for Rendezvous

Each target is associated with a reward function, which takes the form $R_j^{\max}\phi_j(t)$, where $R_j^{\max}$ is the initial amount of reward and $\phi_j(t)$ is a non-increasing discounting function taking values from $[0,1]$. The reward function captures the time-varying aspect of the mission. We say that vehicle i *visits* target point j at time t if $\|x_i(t) - y_j\| \le s_j$ ($||\cdot||$ is the usual Euclidean norm), for all $i = 1,...,M$ and $j = 1,...,N$, where $s_j > 0$ can be viewed as the *size* of a target.

One example of reward function is given below:

$$R_j(t) = \begin{cases} R_j^{\max}(1 - \frac{\alpha_j}{D_j}t) & t \le D_j \\ R_j^{\max}(1-\alpha_j)e^{-\beta_j(t-D_j)} & t > D_j, \end{cases} \tag{2}$$

in which D_j represents a deadline. When the deadline passes, the reward drops exponentially. $R_j^{\max}$ is the initial reward for target j, and $\alpha_j \in (0,1]$, $\beta_j > 0$ are discounting parameters that modulate the speed of decrease.

Now we describe our model for target reward acquisition. Let $\kappa_j^l(t)$ be the index of the lth closest vehicle to target j at time t, based on the time it takes to reach the target if the vehicle used the maximum velocity, i.e., for all $l = 1,...,M$

$$\kappa_j^l(t) = \arg\min\left\{\frac{d_{ij}}{V_i} = \|x_i - y_j\| : i \ne \kappa_j^1,...,\kappa_j^{l-1}\right\}, \tag{3}$$

We then define a *proximity set* $\mathbb{Z}_j^b(t)$ for target j as the closest b vehicles from target j at time t, i.e.,

$$\mathbb{Z}_j^b(t) = \left\{\kappa_j^1(t),...,\kappa_j^b(t)\right\} \tag{4}$$

To simplify notation, we use $d_{k,l}(t)$ to denote the distance between vehicle k and vehicle l at time t. With a slight abuse of notation, we also use $d_{k,j}(t)$ to denote the distance between vehicle k and target j. When there is no possibility of confusion, we omit the t dependencies in the $\kappa_j^l(t)$ function.

For the rendezvous problem, the reward collection requires two vehicles to be close to the target. We assume that the closest two vehicles to the target (i.e., vehicles in the set $\mathbb{Z}_j^2(t)$) are the vehicles performing the reward collection task. Furthermore, vehicles could be heterogeneous, and each one may have a varying amount of success cooperating with other vehicles to achieve reward collection. To capture this, we define $p_{k,l}^{max}$ as the probability of the reward being captured, if vehicles k and l are both close enough to the target. Furthermore, we define $d_{k,l}^*$ as the minimum distance required for vehicles k and l to achieve maximum likelihood (probability $p_{k,l}^{max}$) of successfully acquiring the reward. If the distance between either k or l to the target is below this threshold, then the probability decreases as a function of distance between the target and the vehicle further from the target, until a minimum probability of success $p_{k,l}^{min}$. Generally, these two values defined above also depend on the target being visited, i.e., for target j, the maximum and

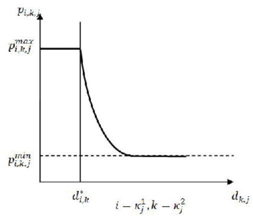

Fig. 1. A Typical Reward Capture Probability Function

minimum reward capture probabilities by vehicles k and l should be written as $p_{k,l,j}^{max}$ and $p_{k,l,j}^{\min}$ respectively.

Suppose at time t, vehicle $\kappa_j^1(t)$ is at location y_j trying to collect the reward of target j. The probability of it successfully capturing the reward depends on the distance of other vehicles from the target at time t. Specifically, we let the probability of the reward captured by $\kappa_j^1(t)$ for target j be a function of distance between the second closest vehicle $\kappa_j^2(t)$ and the target j, which is $d_{\kappa_j^2,j}(t)$:

$$p_{\kappa_j^1(t),\kappa_j^2(t),j}\left(d_{\kappa_j^2(t),j}(t)\right) = \quad (5)$$
$$\begin{cases} p_{\kappa_j^1(t),\kappa_j^2(t),j}^{\max} & 0 \le d_{\kappa_j^2,j}(t) < d^*_{\kappa_j^1(t),\kappa_j^2(t)} \\ g(d_{\kappa_j^2,j}(t)) & d_{\kappa_j^2,j}(t) \ge d^*_{\kappa_j^1(t),\kappa_j^2(t)} \end{cases}$$

where $g(\cdot)$ is a monotonically non-increasing function, $g(d^*_{\kappa_j^1(t),\kappa_j^2(t)}) = p^{\max}_{\kappa_j^1,\kappa_j^2,j}$, and $g(\infty) = p^{\min}_{\kappa_j^1,\kappa_j^2,j}$. An example of such a function is shown in Fig. 1, and typically we want to approximate the step function obtained by replacing $g(\cdot)$ in (5) by the fixed value $p^{\min}_{\kappa_j^1,\kappa_j^2,j}$. For simplicity, we will omit the dependencies on vehicles in the reward capture probability and only write $p_j\left(d_{\kappa_j^2(t),j}(t)\right)$ when no confusion arises.

In previous work on CRH [10],[11], a *relative proximity function* $q_{ij}(t)$ is defined for every pair of targets and vehicles, indicating the probability of assigning vehicle i to target j at time t. This function partitions the mission space into responsibility regions with respect to vehicles, and targets in the "cooperative region" of two vehicles will "attract" both vehicles toward them instead of committing vehicles explicitly to assignments. However, in the problem setting of this paper, where each target requires a team of vehicles to collect its reward, it is impossible to have similar partitions since each individual vehicle may be in several teams that are responsible for different targets. In addition, the reward capture function in (5) actually encompasses the $q_{ij}(t)$ function in the way that it implicitly creates "attraction forces" between each target and a team of vehicles in its proximity set. Therefore, we will not use the $q_{ij}(t)$ function in the CRH controller for the rendezvous problem.

B. Cooperative Receding Horizon (CRH) Controller

Now we propose the Cooperative Receding Horizon (CRH) control mechanism for the rendezvous problem. In this CRH framework, control is applied at time points t_k, $k = 1, 2, ...$ during mission time, and control headings $\mathbf{u}_k = [u_1(t_k), ..., u_M(t_k)]$ are assigned by solving an optimization problem P_k. Then, the vehicles move according to the headings assigned until time t_{k+1}, when either the current action horizon expires or a new event occurs, and then this process is repeated. It is apparent that the key element in the CRH mechanism is how we define the optimization problem P_k, which we will describe below.

Suppose vehicles are assigned headings $\mathbf{u}_k = [u_1(t_k), ..., u_M(t_k)]$ at time t_k, which are intended to be maintained over $[t_k, t_k + H_k]$. We denote H_k as the *planning horizon.* We will elaborate on how we choose H_k later in this section.

At time $t_k + H_k$, the projected positions of the vehicles are

$$\mathbf{x}(t_k + H_k) = [x_1(t_k + H_k), ..., x_M(t_k + H_k)]$$

where $x_i(t_k + H_k)$ is determined by $x_i(t_k)$ and the heading $u_i(t_k)$. Denote by $\digamma_k$ the set of all feasible projected positions $\mathbf{x}(t_k + H_k)$ given the current location at t_k, i,.e.,

$$\digamma_k = \{\mathbf{w} = [w_1, ..., w_M] : \|w_i - x_i(t_k)\| = V_i H_k, \; i = 1, ..., M\} \quad (6)$$

We would like to maximize the total *expected reward* over the projected position set $\digamma_k$, given our definition of probability of capturing the reward. Now we explain how we define the expected reward. First, at $t_k + H_k$, for given projected positions $\mathbf{x} \in \digamma_k$ and target j, vehicles in the set $\mathbb{Z}_j^2(t_k + H_k)$ are identified, i.e., vehicles $\kappa_j^1(t_k + H_k)$ and $\kappa_j^2(t_k + H_k)$. Next, we assume vehicle $\kappa_j^1(t_k + H_k)$ goes directly towards target j and vehicle $\kappa_j^2(t_k + H_k)$ also moves towards it to increase the reward capture probability. A reward capture event occurs as soon as vehicle $\kappa_j^1(t_k + H_k)$ reaches target j, which is at time $\tau_{\kappa_j^1(t_k+H_k),j} = t_k + H_k + \frac{d_{\kappa_j^1(t_k+H_k),j}}{V_{\kappa_j^1(t_k+H_k)}}$. In this case, the reward capture event is binary, with outcomes as either successfully capturing all the reward of target j at time $\tau_{\kappa_j^1(t_k+H_k),j}$ or failing to get any reward from target j. The probability of success is determined by the reward capture probability $p_{\kappa_j^1(t_k+H_k),\kappa_j^2(t_k+H_k),j}$ defined in (5), which depends on the location of the vehicle $\kappa_j^2(t_k + H_k)$ at that time. Then, we define the *expected reward* for target j as the expectation of reward collected from the reward capture event at time $\tau_{\kappa_j^1(t_k+H_k),j}$. The timeline clarifying the expected reward evaluation is shown in Fig. 2. Notice that the definition of expected reward depends on the assumed behaviors (strategies) of vehicles after $t_k + H_k$. We can also consider strategies other than the one adopted here, such as driving the closest vehicle to the target, and attempt to wait (by driving around the target) for the second vehicle to arrive. An interesting problem we leave for future research is to compare strategies and optimize expected rewards over them.

Let $J_{k,j}$ be the expected reward of target j which depends on the locations of members of the set $\mathbb{Z}_j^2(t_k + H_k)$. We also let J_k be the total expected reward, and it is the sum over

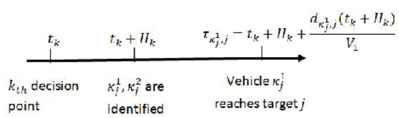

Fig. 2. Timeline of Expected Reward Evaluation

all targets, i.e.,

$$J_k(\mathbf{x}) = \sum_{j=1}^{N} J_{k,j}(\mathbf{x}) \tag{7}$$

Using the definition of expected reward above, for target j, at time $t_k + H_k$, the two closest vehicles $\kappa_j^1(t_k + H_k)$ and $\kappa_j^2(t_k + H_k)$ both move directly towards target j with their maximum speed, and a reward capture event occurs when vehicle $\kappa_j^1(t_k + H_k)$ reaches target j at time $\tau_{\kappa_j^1(t_k+H_k),j} = t_k + H_k + \frac{d_{\kappa_j^1(t_k+H_k),j}}{V_{\kappa_j^1(t_k+H_k)}}$, while the vehicle $\kappa_j^2(t_k + H_k)$ is now at a distance

$$\begin{aligned} & d_{\kappa_j^2(t_k+H_k),j}\left(\tau_{\kappa_j^1(t_k+H_k),j}\right) \\ = \;& d_{\kappa_j^2(t_k+H_k),j}(t_k + H_k) - V_{\kappa_j^2(t_k+H_k)} \frac{d_{\kappa_j^1(t_k+H_k),j}}{V_{\kappa_j^1(t_k+H_k)}} \end{aligned} \tag{8}$$

away from the target. Then, based on (5), the reward capture probability is $p_j\left(d_{\kappa_j^2(t_k+H_k),j}(\tau_{\kappa_j^1(t_k+H_k),j})\right)$ at this time. With this probability, the reward capture event succeeds and gives the full reward of target j at that time, otherwise it fails with no reward captured. Therefore, for any given $\mathbf{x} \in F_k$, the expected reward $J_{k,j}$ is given by

$$\begin{aligned} & J_{k,j}(\mathbf{x}) \\ = \;& R_j\left(\tau_{\kappa_j^1(t_k+H_k),j}\right) \cdot p_j\left(d_{\kappa_j^2(t_k+H_k),j}(\tau_{\kappa_j^1(t_k+H_k),j})\right) \end{aligned} \tag{9}$$

We would like to maximize the total expected reward summed over all agents, and use the solution as the control we apply to agents for the time horizon $[t_k, t_{k+1})$. Therefore, we solve the following optimization problem

$$P_k: \qquad \max_{\mathbf{x} \in F_k} J_k(\mathbf{x}) \tag{10}$$

at the beginning t_k of time horizon $[t_k, t_k + H_k]$. Once we have solved problem P_k for an optimal $\mathbf{x}^*$ among all possible *planned positions* in F_k, the optimal heading $\mathbf{u}_k$ that corresponds to $\mathbf{x}^*$ will be obtained and applied to all vehicles for a time $h_k \leq H_k$, which we define as the *action horizon*, unless an event occurs. This event can be either the detection of a new target, or malfunction of an agent. In either case, the sets of targets and agents are updated, and the optimization problem is solved again.

Clearly, the solution of P_k depends on the choice of H_k and this choice is critical for obtaining desired properties for the CRH controller. It seems natural to choose H_k as the smallest time required for any agent to reach any target point. Hence, we set

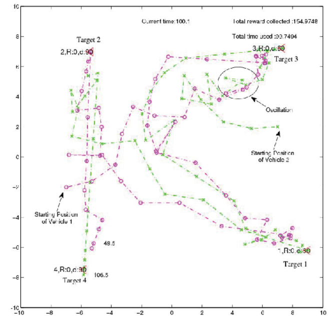

Fig. 3. A Two-Vehicle, Four-Target Example

$$H_k = \min_{\substack{k=1,\dots,M \\ j=1,\dots,N}} \left\{ \frac{\|y_j - x_k(t_k)\|}{V_k} \right\} \tag{11}$$

In section IV we show that this choice of H_k is appropriate for the stationarity property to be established later.

Finally, the performance of the CRH controller depends on the choice of the action horizon h_k. This parameter affects how often we re-evaluate the optimization problem P_k. High frequency evaluations provide smoother vehicle trajectories at the cost of computational burden.

III. Simulation Study

A. Simulation Examples

In this section, we will give some simulation examples using a MATLAB-based simulation environment. Figure 3 shows a mission problem example and the solution based on the CRH controller we developed in the previous sections.

In this example, there are 2 vehicles, and 4 targets. Each target is associated with an initial reward $R_j^{\max}, i = 1, .., 4$, and a deadline D_j, with the discounting function given in (2). All 4 targets in this example are present from the beginning, and the mission time T is 60 sec. For simplicity, $d_{i,j}^*$ and $p_{i,,k}^{\max}$ in (5) are all set to be constant, i.e., $d_{i,j}^* = d^* = 2$, $p_{i,,k}^{\max} = p^{\max} = 1$, for all $i, j = 1, 2, k = 1, .., 4$, and $g(d_{\kappa_j^2,j})$ in (5) is given by

$$g(d_{\kappa_j^2,j}) = p^{\max} \cdot e^{-\alpha \cdot (d_{\kappa_j^2,j} - d^*)} \tag{12}$$

where α is a discounting rate parameter which is set to $\alpha = 2$ in this case. The planning horizon H_k is given by (11), and the action horizon is specified through

$$h_k = \begin{cases} \frac{1}{2} H_k & \text{if } H_k \geq r \\ H_k & \text{if } H_k < r \end{cases} \tag{13}$$

where r is a constant range for all targets. If a vehicle is within that range from a target, then it will commit to that target and, in our example, $r = 0.3$. As we can see in Fig. 3, two vehicles start far away from each other, then they move closer together towards and finally rendezvous at target 3, and vehicle 2 leads to collect the reward. Then, as vehicle 2 moves towards target 1, vehicle 1 follows it in order to make sure when vehicle 2 reaches target 1 it is within the range d^* from target 1. These observations confirm that, although our CRH algorithm does not explicitly involve any procedures that drag vehicles together, vehicles under the algorithm do rendezvous at target points whenever there is reward collecting action at the target. Another important observation is that all targets are successfully visited, although vehicles do not usually follow straight lines toward target points. This can be both advantageous and disadvantageous: the advantage of not going straight toward targets is the ability to deal with potential uncertainties, e.g., new targets emerging, target location not being accurate, etc.; the disadvantage is that vehicles might oscillate in certain areas when there are multiple local optima in solving (10), thus causing delays in mission completion time. For example, in Fig. 3, the two vehicles oscillate at the upper right corner, because they are attracted to targets on both right and left side and, as the deadline of target 3 approaches, the two vehicles decide to visit target 3 first and then turn around to visit other targets. One way to avoid such oscillatory behavior (instabilities), is to introduce a *direction-change cost* [10], which will improve the time efficiency of our algorithm. However, we do not want this cost to be too large to influence our algorithm's ability to handle uncertainties, as shown in the next example.

Figure 4 shows another example, which involves 2 vehicles and 8 targets. What is interesting in this example is that there are two targets that emerge during the mission, and we can see that the vehicles adapt very well with respect to these random events. The two vehicles are on their way to target 4 after they finish collecting the reward of target 1, when target 7 emerges. Then, the vehicles turn right from their original path to target 7, which has larger reward. After they have collected the reward from target 7, they go back to their original path towards target 4. This example illustrates the robustness of our CRH algorithm in an uncertain environment, which is based on the fact that vehicles under our algorithm do not make early commitment to targets and is also the reason that we do not want the direction-change cost to be too large. Note that in both simulation examples all targets are eventually visited by vehicles, which shows the "stationarity" property of our CRH controller. We will analyze this property in detail in Section IV.

B. Comparison to an Approximate Reward Upper Bound

To evaluate the quality of our CRH algorithm as to the objective of reward collection maximization, it is compared to an upper bound on the maximal reward collected as in [9]. However, in our problem setting, even the exhaustive search method used in [9] will not work. For example, suppose we have specified a visiting sequence, and assume target j is set to be the kth target in the visiting sequence, and is visited by vehicle 1; however, since the behavior of vehicle 2 is not specified by the visiting sequence, we do not know where vehicle 2 is at the time when vehicle 1 visits target j, therefore the reward capture probability at this time is unknown and it is impossible to obtain a reward upper

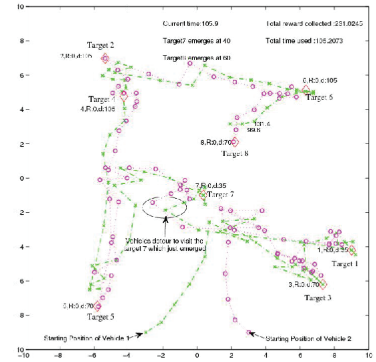

Fig. 4. A Two-Vehicle, Eight-Target Example

Case	ES Reward Upper Bound	CRH Reward	ES Time Lower Bound	CRH Time
1	199.2561	190.1775	55.1065	65.9
2	213.0371	197.3149	58.2331	65.3
3	158.0684	143.9583	70.5232	82.3
4	185.5948	174.7964	62.1479	70.6
5	183.0943	172.6958	64.8794	75.4

TABLE I
COMPARISON BETWEEN CRH ALGORITHM AND APPROXIMATE BOUNDS

bound in this way. Hence, we can only have an approximate reward upper bound, by assuming that the reward capture probability is a step function. Under this assumption, for the case where $N = 4, M = 2$ (the exhaustive search will become intractable for large N, M, see [9]) and a given visiting sequence, the best way to collect reward of a target is obviously to let the vehicle that is not specified to visit the target also move straight towards the target, in order to get close enough when the reward of the target is collected and to increase the reward capture probability. Based on the analysis above, and using a similar exhaustive search algorithm as in [9] to search over all possible visiting sequences in a fully deterministic environment, we can obtain approximate reward upper bounds as shown in Table I

In Table I, for five different cases (different targets locations, initial rewards, vehicle starting points, etc.), we compare the reward obtained through the CRH algorithm and the approximate upper bound obtained through exhaustive search, and also the mission completion time between the two methods. The reward collected based on the CRH algorithm is close to the approximate upper bound in all cases, while the mission completion time is longer. This is expected, since the CRH controller is designed to hedge against uncertainty by not following straight line paths to targets.

IV. STATIONARITY OF THE CRH CONTROL SCHEME

In this section, we will study in detail the stationarity property mentioned in the introduction and illustrated in simulation examples, i.e., the fact that vehicle trajectories generated by the CRH controller will converge to the target points even though there is no explicit mechanism forcing them to do so. To facilitate the analysis, let us first define a *stationary trajectory* for a setting involving a fixed set of vehicles and targets. Recall that associated with every target j is its size s_j, and when vehicle i visits target j at time t we have $\|x_i(t) - y_j\| \leq s_j$.

Definition 1: For a trajectory $\mathbf{x}(t) = [x_1(t) \ldots x_M(t)]$ generated by a controller, if there exists some $t_f < \infty$, such that $\|x_i(t_f) - y_j\| \leq s_j$, for some $i = 1, ..., M$ and some $j = 1, ..., N$, then $\mathbf{x}(t)$ is a stationary trajectory, and we say that the trajectory $x_i(t)$ converges to target j. Otherwise, $\mathbf{x}(t)$ is a nonstationary trajectory.

In the following, we will prove that for the two-vehicle one-target case, the vehicle trajectory under CRH control is stationary. The proof can be naturally extended, at the expense of more complicated notation, to the multiple-vehicle one-target case, because only the two closest vehicles are selected to join the mission. The proof of stationarity for the general case where we have M vehicles and N targets is still the subject of ongoing research. Nevertheless, all simulation results support the stationarity property, including the multiple vehicle and target scenarios shown in Fig. 3-4.

A. 2 Vehicle, 1 Target Analysis

In this section, we prove the stationarity property of the CRH controller for the two-vehicle one-target case, i.e. $M = 2, N = 1$. First, we make the following assumptions.

Assumption 1: Both vehicles have the same maximum speed, i.e. $V_i = V$ for $i = 1, 2$.

Assumption 2: The maximum reward capture probabilities for all vehicles are equal to 1, i.e., $p_{i,1}^{\max} = 1$, for $i = 1, 2$, and the threshold values in (5) are also the same for both vehicles i.e., $d_{1,2}^* = d_{2,1}^* = d^*$, $i = 1, 2$.

Assumption 3: The action horizon is the same as the planning horizon, i.e., $H_k = h_k$, for all $k = 1, 2.....$

Under these assumptions, the solution of the optimization problem P_k will be the positions of the vehicles at t_{k+1}.

Next we give a Lemma to characterize the solutions of the optimization problem in (10), and we use $\mathbf{x}^k = \left(x_1^k, x_2^k\right)$ to represent the locations of vehicles at t_k, and set $d_i^k = \left\|x_i^k - y_1\right\|$, $i = 1, 2$, where y_1 is the location of the target.

Lemma 1: If at time t_k, neither vehicle is visiting the target, i.e., $d_i^k = \left\|x_i^k - y_1\right\| > s_1, i = 1, 2$, then at t_{k+1}, we either have $x_i^{k+1} = y_1$ for some i, or $\left|d_1^{k+1} - d_2^{k+1}\right| < \left|d_1^k - d_2^k\right| - s_1$.

The stationarity property of the two-vehicles one-target case can be proved based on Lemma 1.

Theorem 1: If there are only two vehicles and one target, the vehicle trajectories under the CRH controller are stationary.

All proofs are omitted here due to space limitations, and can be found in [3].

V. CONCLUSIONS

We have developed a cooperative receding horizon controller to drive a team of vehicles to a set of targets, assuming that at each target there is a (possibly time-varying) reward that can be collected. We have focused on the case where only two vehicles are required to achieve the reward collection. For two vehicles and one target, we show that the controller provides stationary trajectories in the sense that it drives vehicles to discrete target points despite no explicit vehicle-to-target assignment process. In future work, we will study the generalization of the stationarity result and explore broader strategy-based algorithms.

REFERENCES

[1] R.W. Beard, T.W. McLain, M.A. Goodrich, E.P. Anderson, Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. *IEEE Transactions on Robotics and Automation*, Vol 18, No. 6, pp. 911-922.

[2] J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, Cooperative path planning for multiple UAVs in dynamic and uncertain environments, in *Proc. 41st IEEE Conf. Decision and Control*, 2002, pp. 2816-2822.

[3] C. Yao, X. C. Ding, and C. G. Cassandras. Cooperative Receding Horizon Control for Multiagent Rendezvous Problem in Uncertain Environments, Technical Report, Center of Information and Systems Engineering, Boston University, 2010, url: http://vita.bu.edu/cyao/YDCTR.pdf.

[4] B. T. Clough, Unmanned aerial vehicles: autonomous control challenges, a researchers perspective, in *Cooperative Control and Optimization*, R. Murphey and P. M. Pardalos, Eds. Norwell, MA: Kluwer, 2000, pp. 35-53.

[5] M. L. Curry, J. M. Wohletz, D. A. Castanon, and C. G. Cassandras, Modeling and control of a joint air operations environment with imperfect information, in *Proc. SPIE 16th Annual International Symposium*, 2002, pp. 41.

[6] A.R. Girard, A.S. Howell and J.K. Hedrick. Border Patrol and Surveillance Missions using Multiple Unmanned Air Vehicles, *IEEE Conference on Decision and Control*, pp. 620-625, Dec. 2004.

[7] G Ghiani, F Guerriero, G Laporte, R. Musmanno. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, Vol. 151, No. 1, pp. 1-11, 2002.

[8] B. Grocholsky, J. Keller, V. Kumar and G. Pappas. Cooperative Air and Ground Surveillance: A scalable approach to the detection and localization of targets by a network of UAVs and UGVs. *IEEE Robotic and Automation Magazine*, Vol. 13, No. 3, pp. 16-26, 2006.

[9] C.G. Cassandras, W. Li, A Receding Horizon Approach for Solving Some Cooperative Control Problems, *Proc. 41st IEEE Conference on Decision and Control*, pp. 3760-3765, 2002.

[10] W. Li, C.G. Cassandras, A Cooperative Receding Horizon Controller for Multi-vehicle Uncertain Environments, *IEEE Transactions on Automatic Control*, AC-51, 2, pp. 242-257, 2006.

[11] W. Li, C.G. Cassandras, Centralized and Distributed Cooperative Receding Horizon Control of Autonomous Vehicle Missions, *Mathematical and Computer Modelling*, Vol. 43, 9-10, pp. 1208-1228, 2006.

[12] F. Lian and R. Murray, Real-time trajectory generation for the cooperative path planning of multi-vehicle systems, in *Proc. 41st IEEE Conference on Decision and Control*, 2002, pp. 3766-3769.

[13] D. Q. Mayne and L. Michalska, Receding Horizon Control of Nonlinear Systems, in *IEEE Transactions on Automatic Control*, AC-35, 7, pp. 814-824, 1990.

[14] T. McLain, P. Chandler, S. Rasmussen, and M. Pachter, Cooperative control of UAV rendezvous, in *Proc. of American Control Conference*, June 2001, pp. 2309-2314.

[15] K. Passino, M. Polycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint, and M. Baum, Cooperative Control for Autonomous Air Vehicles, in *Cooperative Control and Optimization*, R. Murphey and P. M. Pardalos, Eds. Kluwer Academic Publishers, 2000, pp. 233-269.

[16] P. Toth and D. Vigo, Eds., *The Vehicle Routing Problem*. Philadelphia, PA: SIAM, 2002.

[17] J. Yu, S. M. LaValle and D. Liberzon, Rendezvous without Coordinates, in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 1803-1808.

[18] J. M. Wohletz, D. A. Castanon, and M. L. Curry, Closed-loop control for joint air operations, in *Proc. 2001 American Control Conference*, 2001, pp. 4699-4704.