A Cooperative Receding Horizon Controller for Multivehicle Uncertain Environments

Wei Li and Christos G. Cassandras, *Fellow, IEEE*

***Abstract*—We consider a setting where multiple vehicles form a team cooperating to visit multiple target points and collect rewards associated with them. The team objective is to maximize the total reward accumulated over a given time interval. Complicating factors include uncertainties regarding the locations of target points and the effectiveness of collecting rewards, differences among vehicle capabilities, and the fact that rewards are time-varying. We propose a receding horizon (RH) controller suitable for dynamic and uncertain environments, where combinatorially complex assignment algorithms are infeasible. The control scheme dynamically determines vehicle trajectories by solving a sequence of optimization problems over a *planning* horizon and executing them over a shorter *action* horizon. This centralized scheme can generate *stationary* trajectories in the sense that they guide vehicles to target points, even though the controller is not explicitly designed to perform any discrete point assignments. This paper establishes conditions under which this stationarity property holds in settings that are analytically tractable, quantifies the cooperative properties of the controller, and includes a number of illustrative simulation examples.**

***Index Terms*—Cooperative control, optimization, potential field, receding horizon.**

I. Introduction

COOPERATIVE control refers to settings which involve multiple controllable agents cooperating toward a common objective. The information based on which an agent takes control actions may differ from that of the rest. In addition, operating in an uncertain environment requires that agents be capable of reacting to unexpected events. Cooperative control has its roots in team theory [1] and it has recently emerged as a framework for coordinating a team of autonomous vehicles whose task it is to perform a mission with a common goal, e.g., [2] and [3].

In this paper, we consider a setting which involves a team of M *vehicles* and a set of N *target points* in a two-dimensional space (as we will see, N and M may vary with time). Associated with the a target point there is a reward. A *mission* is defined as the process of controlling the movement of the vehicles and ultimately assigning them to target points so as to maximize the total reward collected by visiting points in the set $\mathcal{T}$ within a given mission time T. It is assumed that at least one target point is present in order for a mission to be initiated, but other targets may be added (or deleted) during the course of the mission. The problem is complicated by several factors: i) Target point rewards may be time-dependent, typically decreasing in time; thus, the order in which target points are visited by vehicles may be critical; ii) Different vehicles have different capabilities in terms of collecting the reward associated with a target point; thus, assigning specific vehicles to specific points can also be critical; iii) The exact location of target points may not always be known in advance; iv) There may be *obstacles* (also known as *threats*) in the two-dimensional space (referred to as the *mission space*), which constrain the feasible trajectories of vehicles or may cause their elimination when they are encountered; and v) The collection of information about the state of the mission space is an important aspect of the problem, since knowledge of target point and obstacle locations clearly affects the ultimate objective of the mission.

This setting gives rise to a complex stochastic optimal control problem. In principle, one can invoke dynamic programming as a solution approach, but this is computationally intractable even for relatively simple mission control settings [4], [5]. Because of the complexity of the overall problem, it is natural to decompose it into various subproblems at different levels—from detailed motion control of the vehicles to higher-level path planning and assignment of vehicles to target points. For example, [6] and [7] address issues of dynamically allocating resources, while [8] formulates the problem of cooperative path planning as a mixed-integer linear program (MILP) that incorporates task timing constrains, vehicle capability constraints, and the presence of obstacles. At the level of generating trajectories, key issues involve formation control, obstacle avoidance, and stabilization [9]–[12].

An alternative to this "functional decomposition" approach is one based on *time decomposition*. This is aimed at developing online controllers suitable for uncertain environments where combinatorially complex assignment algorithms are infeasible. The main idea is to solve an optimization problem seeking to maximize the total *expected* reward accumulated by the team over a given time horizon, and then continuously extend this time horizon forward (either periodically or in purely event-driven fashion). This idea is in the spirit of receding horizon (RH) schemes, which are associated with model-predictive control and used to solve optimal control problems for which feedback solutions are extremely hard or impossible to obtain; see [13]–[15] and, more recently, [16]–[19]. In [20], we introduced this approach in the multivehicle, multitarget point setting described previously. The resulting cooperative control scheme

Manuscript received April 5, 2004; revised January 27, 2005. Recommended by Associate Editor at Large P. Antsaklis. This work was supported in part by the National Science Foundation under Grants DMI-0330171 and EEC-00-88073, in part by the Air Force Office of Scientific Research under Gants F49620-01-0056 and F49620-01-1-0348, and in part by the Army Research Office under grant DAAD19-01-0610.

The authors are with the Department of Manufacturing Engineering and the Center for Information and Systems Engineering, Boston University, Brookline, MA 02446 USA (e-mail: wli@bu.edu; cgc@bu.edu).

Digital Object Identifier 10.1109/TAC.2005.861685

dynamically determines vehicle trajectories by solving a sequence of optimization problems over a *planning* horizon and executing them over a shorter *action* horizon. We should emphasize that the optimization problem involved *does not attempt to make any explicit vehicle-to-target assignments*, but only to determine headings that, at the end of the current planning horizon, would place vehicles at positions such that a total expected reward is maximized. Thus, it is a relatively simple problem to solve. A surprising observation in early simulation experiments [20] was the fact that vehicle trajectories actually converge to target points, despite the fact that our approach, by its nature, was never intended to perform any such discrete vehicle-to-target-point assignment. With this seemingly salient property, an advantage of this approach is that it integrates the three tasks of: i) vehicle assignment to target points; ii) routing of the vehicles to their assigned target points; and iii) real-time trajectory generation, all in one function: Controlling vehicle headings in real time.

Motivated by this observation, the contribution of this paper is to present and analyze the RH control scheme outlined above and to explore the generality of the "stationarity" property observed, i.e., the fact that vehicles converge to target points without explicitly being assigned to them. We first study the general M-vehicle N-target point case and identify a condition that ensures a "stationary" trajectory in the sense that a vehicle always ends up visiting a target. For the 1-vehicle N-target point case, we show that this condition reduces to a simple test. For 2-vehicle problems, despite the loss of cost function convexity compared to a single-vehicle case, we are able to show that this condition is satisfied and RH controller trajectories are still stationary. The verification of this condition for $M \geq 3$ vehicles remains an open issue.

From a practical standpoint, the proposed RH control scheme translates into on-line task assignment and path planning for multi-vehicle cooperative missions and its event-driven nature is specifically suited to uncertain environments by naturally adjusting trajectories in response to random events. The scheme is centralized, i.e., it requires a controller (possibly one of the vehicles) with adequate computation and communication capacity to solve an optimization problem whenever invoked as a result of an event occurrence reported by any one vehicle. Commands are then issued from the controller to all team vehicles. The centralized scheme in this paper has formed the basis for developing a distributed RH controller, presented in [21] where the RH controller is also extended to include trajectory constraints such as obstacle avoidance and limits on the vehicle headings allowed.

In Section II, we present the cooperative RH (CRH) control scheme and include some simulation examples to illustrate its properties and cooperative characteristics. In Section III, we place this scheme in the context of a potential field and examine the stationarity properties of the CRH control scheme.

II. CRH Control Scheme

We consider a two-dimensional *mission space*, in which there is a set $\mathcal{T}$ of N *target points* indexed by $i = 1, \ldots, N$. The location of the ith target point is $y_i \in \mathbb{R}^2$. There are also initially M vehicles indexed by $j = 1, \ldots, M$ that define a set $\mathcal{A}$, and let $x_j(t) \in \mathbb{R}^2$ denote the position of the jth vehicle at time t. The vehicles' intial positions are given by x_{j0}, $j = 1, \ldots, M$. For simplicity, we assume a vehicle travels at constant velocity throughout the team mission, i.e.,

$$\dot{x}_j(t) = V_j \begin{bmatrix} \cos u_j(t) \\ \sin u_j(t) \end{bmatrix} \quad x_j(0) = x_{j0} \tag{1}$$

where $u_j(t) \in [0, 2\pi]$ is the controllable heading of vehicle j and V_j is the corresponding velocity. We note that M, N and y_i may change in time.

Vehicles complete tasks in the mission space by visiting target points. To distinguish the relative importance of tasks at time t, each target point has an associated *reward function* denoted by $R_i\phi_i(t)$, where R_i is the maximal reward and $\phi_i(t) \in [0, 1]$ is a discounting function which describes the reward change over time. Note that by appropriately selecting $\phi_i(t)$, it is possible to capture timing constraints imposed on target points, as well as precedence constraints (e.g., if point i must be visited before point k, then $\phi_k(t) = 0$ for all t prior to i being visited by some vehicle). By allowing $R_i < 0$ for some i and properly selecting $\phi_i(t)$, we may also model an obstacle or threat in the mission space (see also [21]). A simple discounting function we can use is the linear one

$$\phi_i(t) = 1 - \frac{\alpha_i}{T} t \quad \alpha_i \in (0, 1]. \tag{2}$$

When a deadline is associated with a paricular target point, we can use

$$\phi_i(t) = \begin{cases} 1 - \frac{\alpha_i}{D_i} t & \text{if } t \leq D_i \\ (1 - \alpha_i) e^{-\beta_i (t - D_i)} & \text{if } t > D_i \end{cases} \tag{3}$$

where D_i is a deadline assigned to target point i, and $\alpha_i \in (0, 1]$, $\beta_i > 0$ are parameters which may be target-specific and chosen to reflect different cases of interest.

In order to distinguish the effectiveness of vehicles relative to a target point i, we define a *vehicle capability factor* $p_{ij}(t) \in [0, 1]$, which reflects the probability that a vehicle j visiting point i at time t will complete the task and collect the reward $R_i\phi_i(t)$. We say that vehicle j *visits* target point i at time t if $\|x_j(t) - y_i\| \leq s_i$ ($\|\cdot\|$ is the usual Euclidean norm) and $p_{ij}(t) > 0$. Thus, $s_i > 0$ can be viewed as the *size* of a target point. If during a visit the vehicle successfully completes its task, it will collect the corresponding reward and, at the same time, target point i is no longer of interest to the mission and it is removed from the set $\mathcal{T}$. Since a visit at point i is related to the consumption of vehicle j's resources, the vehicle capability factor $p_{ij}(t)$ may decrease after the visit, i.e. $p_{ij}(t^+) \leq p_{ij}(t^-)$. In addition, if target i is a threat, a visit may result in the elimination of vehicle j, a negative reward is collected, and j is removed from the set $\mathcal{A}$.

Cooperative structure: Vehicles cooperate by dynamically partitioning the mission space and implicitly allocating regions of it among themselves. Given an arbitrary point $y \in \mathbb{R}^2$ in the mission space (not necessarily a target point), we would like to assign this point to vehicles at time t so that y is assigned to the closest vehicle with the highest probability. To formalize this idea, we first define a *neighbor set* $\mathcal{B}^b(y, t)$ to include the b closest vehicles to $y \in \mathbb{R}^2$ at time

t, where $b \in \{1,\ldots,M\}$. Let $B^l(y,t)$ be the lth closest vehicle to y, i.e. $B^l(y,t) = \arg\min\{\|x_k(t)-y\| : k \in \mathcal{A}, k \neq B^1(y,t),\ldots,B^{l-1}(y,t)\}$, $l = 1,\ldots,M$, so that

$$\mathcal{B}^b(y,t) = \left\{B^1(y,t),\ldots,B^b(y,t)\right\}.$$

We then define the *relative distance function*, $\delta_j(y,t)$, as follows:

$$\delta_j(y,t) = \begin{cases} \frac{\|x_j(t)-y\|}{\sum_{k\in\mathcal{B}^b(y,t)}\|x_k(t)-y\|}, & \text{if } j \in \mathcal{B}^b(y,t) \\ 1, & \text{otherwise} \end{cases}. \quad (4)$$

Of particular interest is the case $b = 2$, so we set $\mathcal{B}(y,t) \equiv \mathcal{B}^2(y,t)$. Moreover, when y coincides with a target point y_i, we write $\mathcal{B}_i(t) \equiv \mathcal{B}(y_i,t)$ and similarly $\delta_{ij}(t) \equiv \delta_j(y_i,t)$.

Next, we define a *relative proximity function* $q_j(y,\delta_j)$ to be any nonincreasing function of δ_j such that $q_j(y,0) = 1$, $q_j(y,1) = 0$. An example of such a function when $b = 2$ is

$$q_j(y,\delta_j) = \begin{cases} 1, & \text{if } \delta_j \le \Delta \\ \frac{1}{1-2\Delta}\left[(1-\Delta)-\delta_j\right], & \text{if } \Delta < \delta_j \le 1-\Delta \\ 0, & \text{if } \delta_j > 1-\Delta \end{cases} \quad (5)$$

where $\Delta \in [0,1/2)$ is an adjustable parameter which can be interpreted as a "capture radius": if a target point i happens to be so close to vehicle j as to satisfy $\delta_j(y_i,t) \le \Delta$, then at time t vehicle j is committed to visit target i. Again, when y coincides with a target point y_i, we write $q_{ij}(\delta_{ij}) \equiv q_j(y_i,\delta_j)$. We can now view $q_{ij}(\delta_{ij})$ as the probability that target point i is assigned to vehicle j at time t, based on the value of $\delta_{ij}(t)$, and observe that $\sum_j q_{ij}(\delta_{ij}) = 1$. Fig. 1(a) illustrates (5) when $\Delta = 0.4$. In Fig. 1(b), the mission space regions which are within the "capture radius" Δ of vehicles are indicated; as will be shown in Section III, the boundaries of these regions are made up by arcs (see Lemma 5). As Δ increases, the regions expand and when $\Delta \to 1/2$ they cover the entire mission space. It is worthwhile noting that when $\Delta = 1/2$ these regions reduce to the Voronoi partition of the mission space (see [22]), as a special case of this cooperative structure construction.

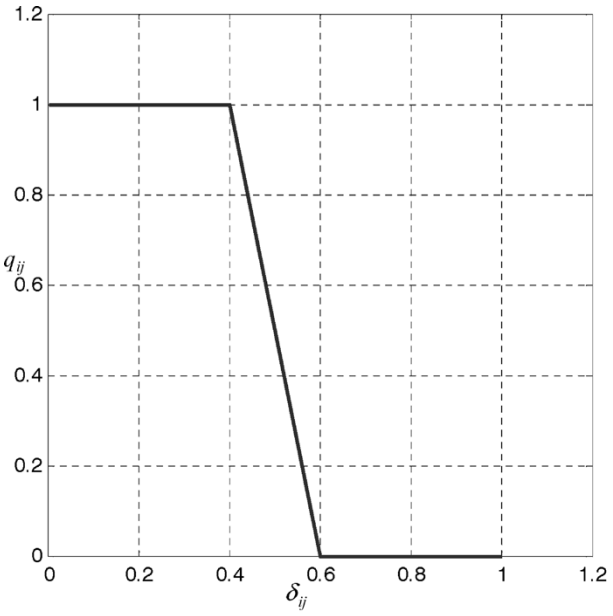
CRH trajectory construction: The objective of the mission is to collect the maximal total reward by the end of some mission time T (or some specific event that defines the end of the mission). To meet this goal, we design a cooperative controller which generates a set of trajectories for each vehicle in the team $\mathcal{A}$ during $[0,T]$. This on-line controller is applied at time points denoted by t_k, $k = 0,1,\ldots$, during the mission time. At t_k, the controller operates by solving an optimization problem $\mathbf{P}_k$, whose solution is the control vector $\mathbf{u}_k = [u_1(t_k)\ldots u_M(t_k)]$. Next, we explain how $\mathbf{P}_k$ is formulated.

Suppose that vehicles are assigned headings $u_1(t_k),\ldots,u_M(t_k)$ at time t_k, intended to be maintained for a *planning horizon* denoted by H_k. Then, at time $t_k + H_k$ the *planned* positions of the vehicles are given by

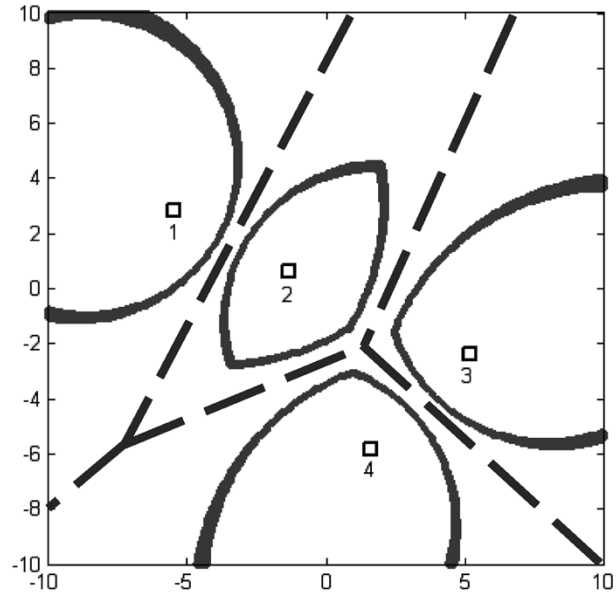
$$x_j(t_k+H_k) = x_j(t_k) + \dot{x}_j(t_k)H_k. \quad (6)$$

Define

$$\tau_{ij}(\mathbf{u}_k,t_k) = (t_k+H_k) + \frac{\|x_j(t_k+H_k)-y_i\|}{V_j} \quad (7)$$



(a)



(b)

Fig. 1. (a) q_{ij} function when $\Delta = 0.4$. (b) Full responsibility regions when $\Delta = 0.4$ and Voronoi partition (dotted line) obtained when $\Delta = 0.5$.

and note that $\tau_{ij}(\mathbf{u}_k,t_k)$ is the earliest time that vehicle j can reach point i under the condition that it starts at t_k with control dictated by $\mathbf{u}_k$ and then proceeds directly from $x_j(t_k+H_k)$ to the target point y_i. We are interested in the maximal reward that vehicle j can extract from target i if it reaches the target at time $\tau_{ij}(\mathbf{u}_k,t_k)$. Clearly, this is given by $R_i\phi_i[\tau_{ij}(\mathbf{u}_k,t_k)]$. For convenience, define

$$\tilde{\phi}_{ij}(\mathbf{u}_k,t_k) = \phi_i\left[\tau_{ij}(\mathbf{u}_k,t_k)\right] \quad (8)$$

where it is worth pointing out that $\tilde{\phi}ij(\cdot)$, unlike $\phi_i(\cdot)$, depends on both i and j. It is also clear that the probability of extracting

this reward, evaluated at time t_k, is given by $p_{ij}[\tau_{ij}(\mathbf{u}_k, t_k)]$. For convenience, we set

$$\tilde{p}_{ij}(\mathbf{u}_k, t_k) = p_{ij}\left[\tau_{ij}(\mathbf{u}_k, t_k)\right]. \tag{9}$$

Returning to the function $q_{ij}(\delta_{ij}) \equiv q_j(y_i, \delta_j)$ defined earlier, we are interested in its value at $t = t_k + H_k$ and define

$$\tilde{q}_{ij}(\mathbf{u}_k, t_k) = q_{ij}\left[\delta_{ij}(t_k + H_k)\right]. \tag{10}$$

Using the notation $\mathcal{A}_k$ and $\mathcal{T}_k$ to denote dependence of these sets on t_k, we can now present the optimization problem $\mathbf{P}_k$, formulated at time t_k, as follows:

$$\max_{\mathbf{u}_k} \sum_{i=1}^{N} \sum_{j=1}^{M} R_i \tilde{\phi}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{p}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{q}_{ij}(\mathbf{u}_k, t_k) \tag{11}$$

with $\tilde{\phi}_{ij}(\mathbf{u}_k, t_k)$, $\tilde{p}_{ij}(\mathbf{u}_k, t_k)$, and $\tilde{q}_{ij}(\mathbf{u}_k, t_k)$ as defined in (8)–(10), respectively. The expression $R_i \tilde{\phi}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{p}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{q}_{ij}(\mathbf{u}_k, t_k)$ in (11) can be seen as the expected reward that vehicle j collects from target point i, evaluated at time t_k using a planning horizon H_k. Note, once again, that we may have $R_i < 0$ in the case of threats in the mission space. Moreover, we can model the elimination of vehicles by including negative terms of the form $C_j \tilde{r}_{ij}(\mathbf{u}_k, t_k)$ where C_j is the cost of vehicle j and $r_{ij}(t) \in [0, 1]$ is the probability that a vehicle j visiting target point i at time t will be eliminated (see also [21]). We should also point out that $\mathbf{P}_k$ is readily extended to incorporate additional features of a team mission environment. For example, the formulation in (11) assumes that a vehicle is free to arbitrarily change its heading at no cost. Introducing such a cost [i.e., a negative reward in (11)] results in a tradeoff between changing direction and the incremental expected benefit of such a control action. This tradeoff is formalized in [21] by introducing a *heading change cost function* $\Delta(u, u')$, where u is the current heading of a vehicle and u' is a new heading, determined as the solution of a problem of the form $\mathbf{P}_k$ at some t_k.

Problem $\mathbf{P}_k$ is parameterized by the planning horizon H_k, which is critical in obtaining desirable properties for this CRH controller. In particular, we will set

$$H_k = \min_{j \in \mathcal{A}_k, i \in \mathcal{T}_k, p_{ij}(t_k) > 0} \left\{ \frac{||y_i - x_j(t_k)||}{V_j} \right\} \tag{12}$$

i.e., the smallest "distance" (in time units) between any target point and any capable vehicle at time t_k. We will show in Section III (see Lemma 1) why this choice is appropriate. Moreover, from our definition of a vehicle "visiting" a target point, it follows that $H_k > \min_{j \in \mathcal{A}_k, i \in \mathcal{T}_k, p_{ij}(t_k) > 0}\{s_i / V_j\}$ where $s_i > 0$ is the size of i. At the time of a visit, either the task at the target point is completed and the target is removed from the set $\mathcal{T}_k$ or the vehicle depletes its resources (i.e., $p_{ij}(t_k) = 0$); the planning horizon is also re-evaluated at this point.

Upon getting the optimal $\mathbf{u}_k$ for (11) based on H_k and all state information available at t_k, all vehicles follow this control for an *action horizon* $h_k \leq H_k$. The process is then repeated at time $t_{k+1} = t_k + h_k$, $k = 0, 1, \ldots$ The value of h_k is determined by two factors. First, if an unexpected event takes place (e.g., a new target is detected or a vehicle fails) at some $t_e \in (t_k, t_k + h_k)$, then we set $h_k = t_e - t_k$. Otherwise, we simply update the control after a prespecified amount of time. Thus, in general, $\{t_k\}$ is a random sequence. The CRH controller terminates when: i) all the target rewards are collected; ii) all vehicles are eliminated; iii) vehicles deplete all their resources; or iv) the mission time expires.

Looking at (11), observe that the solution of this problem is some heading $u_j(t_k)$ for every vehicle $j \in \mathcal{A}_k$, but there is no express constraint imposed on $\mathbf{P}_k$ to *assign* a vehicle to a target location: There is no constraint of the form $x_j(t_k + H_k) = y_i$, $i \in \mathcal{T}_k$ or $y_i - x_j(t_k)/||y_i - x_j(t_k)|| = x_j(t_k + H_k - x_j(t_k)/||x_j(t_k + H_k) - x_j(t_k)||$ forcing a vehicle to either be at a some point y_i by a certain time or to set a heading for it. The intent of designing the CRH controller was originally to simply direct vehicles toward points in the mission space in a way that the team maximizes an *expected* reward as represented by the objective function in (11); a lower level controller might then be responsible for making final *discrete* assignments. However, the empirical observation made (as seen in the next section) was that in fact vehicles always end up visiting a target point, as long as the planning horizon is appropriately selected, specifically as in (12) above. The remainder of this paper is largely intended to explore the validity and potential limitations of this "stationarity" property.

Complexity of problem $\mathbf{P}_k$: The motivation for our CRH approach comes from: i) the stochastic nature of the mission space, and ii) an effort to bypass the combinatorial complexity of vehicle assignment problems. In deterministic discrete optimization, there exist several well-known algorithms to address the problem of assigning M vehicles to sequentially visit N points and maximize a total reward (e.g., Vehicle Routing Problems or, if $M = 1$, Traveling Salesman Problems [23], [24]). Using such algorithms in our problem setting is infeasible, since random events necessitate that this type of combinatorially hard problem is repeatedly solved on line; in addition, these algorithms are typically very sensitive to the problem setting (e.g., target locations and rewards). This raises the issue of whether solving $\mathbf{P}_k$ is significantly simpler than any such approach and whether it is scalable with respect to N and M.

In order to provide a reasonable comparison of our approach to a deterministic assignment problem (e.g., as in [8]), it is common to consider the mission space as a two-dimensional grid. Depending on the grid design, the control $u_j(t_k)$ can take G values and in the case of M vehicles, the solution of (11) reduces to comparing G^M cost values. An immediate observation is that this is independent of the number of target points N [except for some arithmetic to add terms in (11)], so that *the complexity of this approach is independent of* N. The problem of evaluating assignments of M vehicles to sequentially visit N points can be mapped onto a "ball and urn" problem (ignoring the extra complexity induced by constraints that make some assignments infeasible) as follows: given a labeling of N balls, there are C_{M-1}^{N+M-1} groupings into M urns (where C denotes the number of combinations), therefore the total number of assignments is $(N!) \cdot C_{M-1}^{N+M-1} = (N + M - 1)!/(M - 1)!$. In a stochastic setting, if K random events occur each one requiring resolving a problem, the complexity of this approach is of the order of $K \cdot (N + M - 1)!/(M - 1)!$ compared to

the complexity of the CRH scheme which is $K \cdot G^M$. As an example, for $N = 10$, $M = 4$, $G = 8$ (a standard rectangular grid), $(N + M - 1)!/(M - 1)! = 1.0378e + 009$, whereas $G^M = 4096$, a complexity reduction of the order of 10^6 *for each of the* K *events*. We should also point out that when the *stochastic* optimization problem is considered, neither approach can be guaranteed to yield a global optimum because the associated stochastic discrete dynamic program is intractable. In practice, problem $\mathbf{P}_k$ is an unconstrained nonlinear program which may be solved very efficiently by a variety of standard techniques without resorting to any discretization of the mission space; naturally, in general, only local optimality can be guaranteed in this case.

A. Simulation Examples

Before proceeding with the analysis of the CRH controller's properties, we provide in this section a few examples in a simulated mission space environment. We do so at this point in order to illustrate these properties and motivate the analysis carried out in the rest of this paper, as well as to highlight some issues that deserve further scrutiny.

The implementation of the controller requires selecting the planning and action horizons H_k and h_k, respectively. In what follows, H_k is given by (12) and h_k is selected so that

$$h_k = \begin{cases} \frac{1}{2}H_k, & \text{if } H_k > r \\ H_k, & \text{if } H_k \leq r \end{cases} \tag{13}$$

where, for any target point, r is a small distance from it (in the sequel, $r = 2$). The reward discounting function $\phi_i(t)$ is given by (3), $q_{ij}(\delta_{ij}) \equiv q_j(y_i, \delta_j)$ is given by (5) with $\Delta = 0.4$, and $\delta_{ij} \equiv \delta_j(y_i, t)$ is given by (4) with $b = 2$. Finally, the simulated environment allows for time-dependent target point values (modeled through $\phi_i(t)$, including deadlines), different vehicle capabilities (modeled through $p_{ij}(t)$), and random events such as new target points being detected or vehicle failures.

An example of a mission executed under CRH control is shown in Fig. 2. In this case, there are 4 vehicles (A,B,C,D) and 10 target points (shown as indexed triangles). Each target point is assigned a maximal reward R_i, with $R_i = 10$ for $i = 1, 4, 5, 6, 7, 9, 10$, $R_2 = R_8 = 20$, and $R_3 = 30$. Targets are also assigned deadlines D_i, with $D_1 = D_9 = 10$, $D_i = 20$ for $i = 3, 5, 6, 8$, and $D_i = 30$ for $i = 2, 4, 7, 10$. The initial vehicle positions (randomly chosen) are indicated by the squares. The notation $(a : b)$ on the top right represents the time of the snapshot shown and the remaining total reward for the team. For simplicity, in this example we set all vehicle capability factors to 1 (the controller's basic behavior is not affected by changing these values). It is also assumed that all target points are known in advance and that information is shared by all vehicles.

Looking at Fig. 2, we make several observations. First, as pointed out in the previous section, the RH controller guides vehicles to eventually visit discrete target points. This is not at all obvious, since the controller sets headings with the aim of maximizing expected rewards, but has no explicit mechanism for forcing vehicles to reach any discrete point in the mission space. For instance, vehicle A initially heads for a position between points 1, 6, and 9, but then eventually converges to target

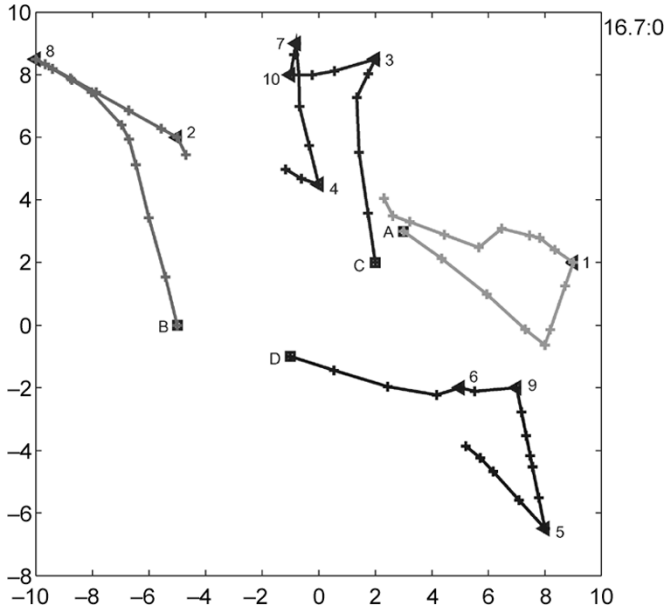


Fig. 2. Example of the CRH controller operation with four vehicles, ten target points.

point 1. This "stationarity" property of the CRH controller is analyzed in the next section. Note that such flexibility in a vehicle's behavior is highly desirable in an uncertain environment: The vehicle does not commit to a specific target point until the last possible time instant. In this example, target point 1 has a tight deadline $D_1 = 10$, so it is not surprising that A selects it first in its trajectory. The cooperative nature of the controller is also seen in the joint behavior of vehicles A and D: while A heads for 1, vehicle D heads toward 6 and 9 and visits 9 in time to make the deadline $D_9 = 10$ while also visiting 6 which happens to be along the way. Another manifestation of cooperative control in this example is the way that the four vehicles implicitly divide the mission space into respective areas of responsibility. Finally, note that the control updates are not made uniformly over time: control updates are indicated by dots in Fig. 2, and one can see that some are made more frequently than others. This is a result of the asynchronous event-driven nature of the controller, where an event such as "vehicle j visits target point i" triggers a re-evaluation of the controls obtained by solving $\mathbf{P}_k$ at that event time.

Fig. 3 shows another example with 4 vehicles (P, G, BK, BL, all initially located at the same point), but in this case *only a single target point is initially known* (labeled 1). The remaining points indicated are unknown to the team, but can be detected if a vehicle happens to be within a "sensor detection area" (in this case, a circle of radius 10/3 units). Thus, an essential aspect of the mission is the search for target points. In this case, the CRH controller serves to disperse the four vehicles so that they are headed in different directions. In Fig. 3(a), because targets 4, 8, 9, 11 are located within the detection radius defined by the initial location, they become known to the team. Vehicles BK, G, and BL are headed for these targets, while vehicle P is heading for target 1. In Fig. 3(b), all but one target points have been detected. An example of the controller's cooperative nature is seen in the behavior of BK: it was originally accompanying P toward

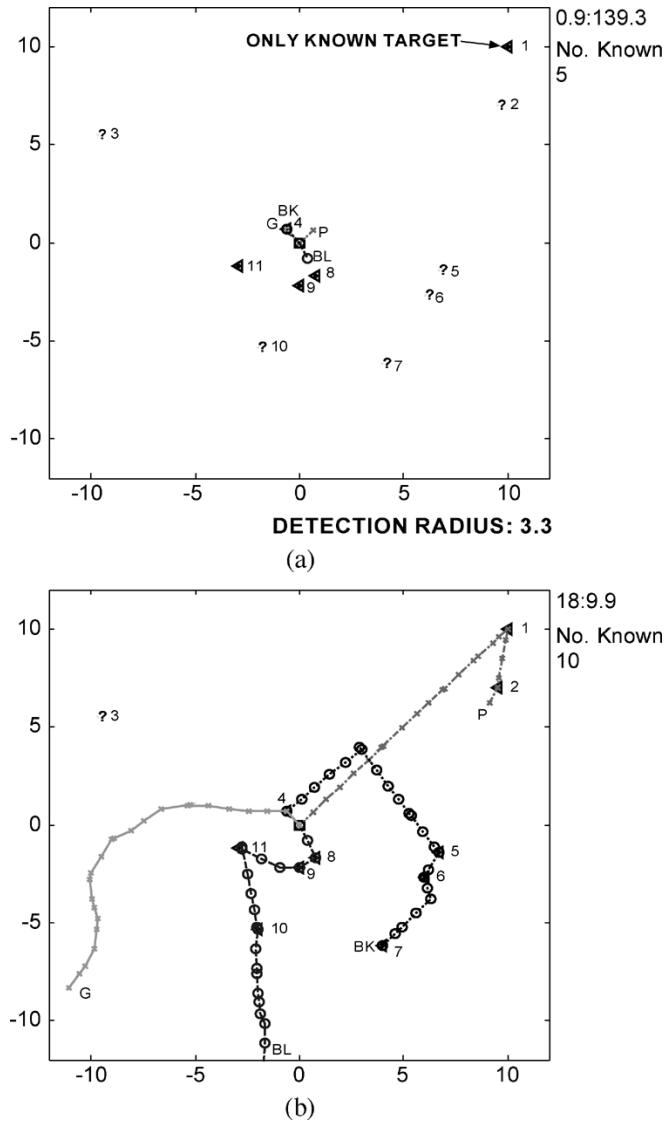


Fig. 3. Example of a mission with unknown targets.

1, up to a point where the effect of $q_{ij}(\delta_{ij})$ in (11) is such that the expected reward is maximized by having vehicle BK be "repelled," which is tantamount to this vehicle searching for unknown targets and in fact detecting some as seen in Fig. 3(b) relative to Fig. 3(a). This occasional repelling action automatically induced by the cooperative controller will be discussed in detail in Section III-C; see (28). Along the same lines, vehicle G, after visiting target 4, is directed away from the areas that the other three vehicles are covering. Although this vehicle fails to detect any of the remaining unknown targets, it will continue in "search mode" during the mission. On the other hand, vehicle BL visits a number of targets that it happens to detect along its trajectory.

Additional simulation results were reported in [20] where the performance of the controller was also compared to an upper bound provided by performing an exhausitive search over all possible straight-line (minimum-time) trajectories. The results indicate that the CRH controller matches the reward upper bound, except in some cases where the behavior of vehicles under CRH control exhibits instabilities in the form of headings that oscillate between two or more values corresponding to nearby target points. This is intuitively understandable, since a vehicle located in the vicinity of two or more target points with comparable expected rewards may not be able to clearly discriminate between these target choices. This oscillatory behavior is overcome by introducing heading change costs as described in the previous section and further discussed in [21]. Naturally, the fact that the CRH controller was observed to match the reward upper bound does not prove its "optimality" in any way, but it is a strong indication that it possesses attractive properties, some of which we discuss next.

III. Stationarity Properties of the CRH Control Scheme

We will now study in some detail the "stationarity" property illustrated in Section II-A, i.e., the fact that vehicle trajectories generated by the CRH controller converge to the discrete target points, even though there is no explicit mechanism forcing them to do so through the solution of problems $\mathbf{P}_k$, $k = 1, 2, \ldots$ To facilitate the analysis, let us first define a *stationarity trajectory* for a setting involving fixed sets $\mathcal{T}$ and $\mathcal{A}$. Recall that associated with every target point i is its size s_i and when vehicle j visits target point i at time t we have $\|x_j(t) - y_i\| \leq s_i$.

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 $\|x_j(t_f) - y_i\| \leq s_i$, $i \in \mathcal{T}$, $j \in \mathcal{A}$, then $\mathbf{x}(t)$ is a stationary trajectory, and we say that the trajectory $x_j(t)$ converges to target i. Otherwise, $\mathbf{x}(t)$ is a nonstationary trajectory.

In what follows, we will concentrate on problems with reward discounting functions given by (2) with $\alpha_i = \alpha$ and $R_i > 0$ for all $i \in \mathcal{T}$, and with relative proximity functions given by (5) where $b = 2$. We will also set $h_k = H_k$ and choose the planning horizon H_k as in (12) when this is necessary to establish some of our results. For simplicity, we also set $V_i = V$ for all $i \in \mathcal{T}$, and $p_{ij}(t) = 1$ for all $i \in \mathcal{T}$, $j \in \mathcal{A}$ and $t \in [0, T]$; these assumptions simplify the analysis but do not alter the nature of the results. For this problem, in the objective function in (11), we make use of the fact that $\sum_{j \in \mathcal{A}} q_{ij} = 1$ and omit the argument of $q_{ij}[\cdot]$ for notational simplicity. Maximizing this function over $\mathbf{u}_k$ is then equivalent to the minimization problem

$$\min_{\mathbf{u}_k} \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{A}} R_i \, \|x_j(t_k + H_k) - y_i\| \, q_{ij}. \tag{14}$$

Let us define $\mathcal{F}_k$ to be the feasible set for $\mathbf{x}_k = [x_1(t_k + H_k), \ldots, x_M(t_k + H_k)]$

$$\mathcal{F}_k = \{\mathbf{w} : \|w_j - x_j(t_k)\| = V H_k\}$$

where $\mathbf{w} = [w_1, \ldots, w_M] \in \mathbb{R}^{2M}$. Because of the one-to-one relationship between $\mathbf{u}_k$ and $\mathbf{x}_k$ [see (1) and (6)], (14) becomes

$$\min_{\mathbf{x} \in \mathcal{F}_k} \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{A}} R_i \|x_j - y_i\| q_{ij}.$$

It is convenient at this stage to introduce a *potential function* (see [25]) for this N-point system as follows:

$$J(\mathbf{x}) = \sum_{i \in \mathcal{T}} J_i(\mathbf{x}) = \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{A}} R_i \|x_j - y_i\| q_{ij} \tag{15}$$

with

$$J_i(\mathbf{x}) = \sum_{j\in\mathcal{A}} R_i \|x_j - y_i\| q_{ij}. \tag{16}$$

The optimization problem (14) can then be rewritten as

$$\min_{\mathbf{x}\in\mathcal{F}_k} J(\mathbf{x}) \quad \mathcal{F}_k = \{\mathbf{w} : \|w_j - x_j(t_k)\| = VH_k\}. \tag{17}$$

Looking at (17), we can see that the CRH controller operates by iteratively exploring the potential function $J(\mathbf{x})$ and leading vehicles toward the minimum points. Thus, its stationarity property is related to two questions: i) under what condition does the CRH controller lead a vehicle to a (local) minimum of $J(\mathbf{x})$, and ii) under what condition does such a minimum of $J(\mathbf{x})$ satisfy $x_j = y_i$ for some $i \in \mathcal{T}$, $j \in \mathcal{A}$ (i.e., the minima coincide with target points).

In Section III-A, we establish a general condition for the stationarity of an arbitrary M-vehicle, N-target point system. In Section III-B, we show that this condition can be reduced to a simple test in the case of a single vehicle. In Section III-C, we perform stationarity analysis to two-vehicle systems, which provides insights to the cooperative behavior of vehicles under CRH control.

A. M-Vehicle, N-Target Point Analysis—A General Result

For an arbitrary M-vehicle, N-target point system, question i) posed previously is answered by Theorem 3, the main result in this section. According to this theorem, under the condition that *every local minimum in the potential field corresponds to at least one vehicle located at some target point*, a trajectory generated by the CRH controller is stationary by leading a vehicle to one such local minimum. This condition can be formally written as follows.

Condition (C): Let $\mathbf{x}^l \in \mathbb{R}^{2M}$, $l = 1, \ldots, L$, be any local minimum of $J(\mathbf{x})$. Then, for all $\mathbf{x}^l = [x_1^l, \ldots, x_M^l]$, $x_j^l = y_i$ for some $i \in \mathcal{T}$, $j \in \mathcal{A}$.

We begin by defining the circular decision space $\mathcal{F}_k^j \subset \mathbb{R}^2$ for any vehicle j at step k

$$\mathcal{F}_k^j = \{x : \|x - x_j(t_k)\| = VH_k\}, \qquad j \in \mathcal{A}.$$

Recalling that $\mathbf{x} = [x_1(k), \ldots, x_M(k)] \in \mathbb{R}^{2M}$, the overall decision space $\mathcal{F}_k \subset \mathbb{R}^{2M}$ is defined as

$$\mathcal{F}_k = \left\{\mathbf{x} : x_j \in \mathcal{F}_k^j,\ j \in \mathcal{A}\right\}. \tag{18}$$

In addition, we define the interior of the region bounded by $\mathcal{F}_k^j$ at step k as

$$\mathcal{I}_k^j = \{x : \|x - x_j(t_k)\| < VH_k\}, \qquad j \in \mathcal{A}$$

and the region $\mathcal{I}_k \subset \mathbb{R}^{2M}$

$$\mathcal{I}_k = \left\{\mathbf{x} : x_j \in \mathcal{I}_k^j, j \in \mathcal{A}\right\}.$$

Finally, we define the closed regions $\mathcal{G}_k^j = \mathcal{I}_k^j \cup \mathcal{F}_k^j$ and $\mathcal{G}_k = \mathcal{I}_k \cup \mathcal{F}_k$, and note that they are convex sets in $\mathbb{R}^2$ and $\mathbb{R}^{2M}$, respectively.

We establish two useful results. The first asserts that under condition (**C**) every CRH control step leads vehicles to points that do not increase the potential function value.

Lemma 1: Under condition (**C**), let $\mathbf{x}_k = [x_1(t_k), \ldots, x_M(t_k)] \in \mathbb{R}^{2M}$ denote the position of vehicles at the beginning of the kth CRH control step. Then, if $H_k = \min_{j\in\mathcal{A}, i\in\mathcal{T}}\{\|y_i - x_j(t_k)\|/V\}$, the solution of the CRH control problem $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathcal{F}_k} J(\mathbf{x})$ satisfies $J(\mathbf{x}_{k+1}) < J(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{I}_k$ and $J(\mathbf{x}_{k+1}) \le J(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{G}_k$.

Proof: See the Appendix. ■

This lemma provides a key step toward showing the convergence of CRH trajectories to target points: since $\mathbf{x}_k \in \mathcal{I}_k$ and $J(\mathbf{x}_{k+1}) < J(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{I}_k$, we have $J(\mathbf{x}_{k+1}) < J(\mathbf{x}_k)$ for all k. However, the lemma does not offer information about the speed of convergence, an issue we address next. The following lemma gives a lower bound for $\|\nabla J(\mathbf{x})\|$ when vehicles are in the neighborhood of any local minimum $\mathbf{x}^l$ and provides information about the speed of convergence when vehicles approach a local minimum.

Lemma 2: Under condition (**C**), let $x_j^l = y_i$ for some $i \in \mathcal{T}$, $j \in \mathcal{A}$. Then

$$\left.\frac{\partial J(\mathbf{x})}{\partial x_m}\right|_{\mathbf{x}=\mathbf{x}^l} = 0, \qquad \text{for all } m \neq j$$

and

$$\lim_{\varepsilon\to 0}\left\| \left.\frac{\partial J(\mathbf{x})}{\partial x_j}\right|_{\mathbf{x}=[x_1^l,\ldots,x_j=y_i+\varepsilon,\ldots,x_M^l]}\right\| \ge \gamma(\mathbf{x}^l)$$

for all $\varepsilon \in \mathbb{R}^2$, where $\gamma(\mathbf{x}^l) > 0$ is a constant.

Proof: See the Appendix. ■

The fact that $\|\nabla J(\mathbf{x})\|$ has a positive lower bound when $\mathbf{x}$ is in the neighborhood of a local minimum $\mathbf{x}^l$ implies that under condition (**C**), the solutions of $\|\nabla J(\mathbf{x})\| = 0$ do not include any local minima of $J(\mathbf{x})$. Thus, when $\|\nabla J(\mathbf{x})\| = 0$, $\mathbf{x}$ must either be a local maximum or a saddle point. These two lemmas will help us establish Theorem 3 after making two additional technical assumptions.

Assumption 1: $J(\mathbf{x})$ has a finite number of points such that $\|\nabla J(\mathbf{x})\| = 0$.

Let us denote these points (which, by Lemma 2, can only be maxima or saddle points) by $\mathbf{x}^p$, $p = 1, \ldots, P$. Then, a circular region $\mathcal{D}_p(\eta)$ with radius $r_p(\eta)$ can be defined for each $\mathbf{x}^p$ and positive constant η so that

$$\mathcal{D}_p(\eta) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}^p\| \le r_p(\eta)\} \tag{19}$$

$$r_p(\eta) = \arg\min\{r : \|\nabla J(\mathbf{x})\| \ge \eta \text{ for all } \mathbf{x} \text{ s.t. } \|\mathbf{x} - \mathbf{x}^p\| = r\}. \tag{20}$$

Thus, $\mathcal{D}_p(\eta)$ defines the smallest circular region around $\mathbf{x}^p$ which contains all neighborhood points of $\mathbf{x}^p$ with $\|\nabla J(\mathbf{x})\| < \eta$. Since $J(\mathbf{x})$ is continuous, we must have $\lim_{\eta\to 0} r_p(\eta) = 0$ and $\lim_{\eta\to 0} \mathcal{D}_p(\eta) = \mathbf{x}^p$. In fact, if Assumption 1 is relaxed to require only that there is a finite number of regions which contain maxima or saddle points with the same value of $J(\mathbf{x})$, the same technique used in the proof of

Theorem 3 can be applied, at the expense of added notational complexity; we will therefore make use of Assumption 1.

Assumption 2: There exists some $\bar{\eta} > 0$ such that $\|\nabla J(\mathbf{x})\| \geq \bar{\eta}$ for all $\mathbf{x} \in \mathbb{R}^{2M}$ and $\mathbf{x} \notin \mathcal{D}_p(\bar{\eta})$, $p = 1, \ldots, P$.

Under this assumption, for all $0 < \eta \leq \bar{\eta}$, we have $\|\nabla J(\mathbf{x})\| \geq \eta$ for all $\mathbf{x} \in \mathbb{R}^{2M}$ except for a finite number of regions $\mathcal{D}_p(\eta)$, $p = 1, \ldots, P$. It follows from this discussion that we should select some $\bar{\eta}$ so that $\bar{\eta} < \gamma(\mathbf{x}^l)$, as defined in Lemma 2, for all $l = 1, \ldots, L$.

Theorem 3: Under Assumptions 1,2 and condition (**C**), let $\mathbf{x}_k = [x_1(t_k), \ldots, x_M(t_k)] \in \mathbb{R}^{2M}$ denote the position of vehicles at the beginning of the kth CRH control step and $\mathbf{x}_k \neq \mathbf{x}^l$ for all $l = 1, \ldots, L$. Then, $J(\mathbf{x}_k) - J(\mathbf{x}_{k+1}) > \gamma$ for all k, where $\gamma > 0$ is a constant independent of k, and the CRH trajectory converges to a local minimum in a finite number of steps.

Proof: See the Appendix. ∎

Theorem 3 offers a sufficient condition for stationarity based on Definition 3.1. If condition (**C**) holds, i.e., every local minimum $[x_1^l, \ldots, x_M^l]$ in the potential field corresponds to at least one vehicle located at a certain target point ($x_j^l = y_i$ for some $i \in \mathcal{T}$, $j \in \mathcal{A}$), the CRH controller is guaranteed to generate a stationary trajectory. The importance of this general result is that it translates testing the stationarity property of the CRH controller into a test for the structure of the potential function $J(\mathbf{x})$: At least through numerical methods, one can explore $J(\mathbf{x})$ to identify the function's minima and see if condition (**C**) applies. Naturally, it is also desirable to further simplify this task by determining simpler verifiable conditions under which condition (**C**) indeed holds, which is precisely question ii) as posed in the end of the last section. In what follows, we address this for the case where $M = 1, 2$.

B. One-Vehicle, N-Target Point Analysis

We begin by considering the single vehicle case, i.e., $M = 1$ and $q_{ij} = 1$ in (14). Aside from being a necessary first step before attempting to deal with multi-vehicle settings, this case is also the basis for eventual *distributed* cooperative control in which each vehicle can solve a problem relying only on its own information set (see [21]). As we will also see, even in this simple case where there is no cooperation involved we can only guarantee trajectory stationarity under a condition that involves the locations and rewards of target points.

Since we are now dealing with a single vehicle, there is a single control variable u_k for a vehicle whose velocity is V and position $x(t_k)$. Problem (17) becomes:

$$\min_{x \in \mathcal{F}_k} J(x) \quad \mathcal{F}_k = \{w : \|w - x(t_k)\| = VH_k\} \tag{21}$$

where $J(x) = \sum_{i \in \mathcal{T}} R_i \|x - y_i\|$ and $\mathcal{F}_k$ is the feasible set for $x(t_k + H_k)$. Moreover, note that $J(x)$ as defined above is a convex function on $\mathbb{R}^2$, since it is the sum of N convex functions $\|x - y_i\|$, $i = 1, \ldots, N$. This convexity greatly simplifies the one-vehicle stationarity analysis. The following theorem locates the global minimum of $J(x)$ and presents the stationarity properties of a one-vehicle N-target system.

Theorem 4: The convergence of the RH trajectory $x(t)$ satisfies the following.

i) **Necessary condition for nonstationary trajectory**: If $x(t)$ is a nonstationary trajectory, then there exists some $y_e \in \mathbb{R}^2$, s.t. $y_e \neq y_i$ for all $i = 1, \ldots, N$ and

$$\nabla J(y_e) = \sum_{i=1}^{N} R_i \frac{y_e - y_i}{\|y_e - y_i\|} = 0. \tag{22}$$

ii) **Sufficient condition for stationary trajectory**: $x(t)$ is a stationary trajectory if there exists a y_i, such that

$$\gamma \equiv R_i - \left\| \sum_{j=1, j \neq i}^{N} R_j \frac{y_i - y_j}{\|y_i - y_j\|} \right\| > 0 \tag{23}$$

and, letting $x(0) = x_0$, the RH trajectory converges by time $t_f \leq [J(x_0) - J(y_i)]/\gamma V$.

Proof: See the Appendix. ∎

Condition (23) in Theorem 4 fails to be necessary because in some settings where it does not hold $x(t)$ may still be a stationary trajectory. For example, suppose that $x(0) = y_i$ for some $i \in \mathcal{T}$; in this case, $x(t)$ is trivially a stationary trajectory, even though condition (23) may not hold. Also note that condition (23) is not as restrictive as it might appear at first glance. First of all, it only requires that *any one* target point satisfies this inequality. Second, when taking the sum of $N - 1$ vectors on the right-hand-side of (23), these vectors can often counteract each other, allowing some not necessarily large value R_i to satisfy the inequality.

We can give a simple interpretation to this theorem by viewing the vehicle as a particle with maximum velocity V that follows the motion dynamics

$$\dot{x} = \begin{cases} -\frac{\nabla J(x)}{\|\nabla J(x)\|} V, & \text{if } \|\nabla J(x)\| > 0 \\ 0, & \text{if } \|\nabla J(x)\| = 0 \end{cases}. \tag{24}$$

Let us define forces F_i applied to this particle as follows:

$$F_i = \begin{cases} -R_i \frac{x - y_i}{\|x - y_i\|}, & \text{if } x \neq y_i \\ 0, & \text{otherwise} \end{cases}, \qquad i = 1, \ldots, N \tag{25}$$

where x is the current position of the particle. In other words, each target point i applies an attraction force on the particle, the magnitude of which is R_i and its direction is toward i. The total force is given by

$$F = \sum_{i=1}^{N} F_i = -\nabla J(x) = -\sum_{i=1}^{N} R_i \frac{x - y_i}{\|x - y_i\|}.$$

Under these attraction forces F_i, $i = 1, \ldots, N$, the particle's motion is the same as applying the motion dynamics (24). In part i) of Theorem 4, we can see that at y_e the total force is $F = 0$. Therefore, the particle will stop once it reaches the equilibrium point y_e. In part ii) of the theorem, if there exists some i satisfying (23), then the attraction force of target point i is so large that it exceeds the total attraction force generated by all other nodes. In such cases, the particle will be "captured" by i and the motion trajectory will converge to this point. This "attraction force" point of view of the problem provides some insights that are particularly helpful in multivehicle settings.

Some additional insight is provided by the theorem regarding the observation made in [20] that the RH controller can generate trajectories exhibiting oscillatory behavior. If (23) is not satisfied, then the trajectory generated by our RH scheme might indeed be unstable, in which case additional actions are required to prevent that (e.g., including a heading change cost, as mentioned in Section III). However, in situations where case ii) applies, we can ensure that the trajectory will be stationary.

C. Two-Vehicle Analysis

A crucial difference between multi-vehicle and one-vehicle settings is the presence of the function $q_{ij}(\delta_{ij})$ in (14). As already mentioned, the interpretation of $q_{ij}(\delta_{ij})$ is the probability that target i is assigned to vehicle j, given the relative distance $\delta_{ij}(t)$ between vehicle j and target i at time t. For $M=2$, since $\sum_{j=1}^{M}\delta_{ij}(t)=\delta_{i1}(t)+\delta_{i2}(t)=1$, depending on the distance of the two vehicles from target i, there are three cases.

1) If $\delta_{i1}(t)\leq\Delta$, we have $\delta_{i2}(t)=1-\delta_{i1}(t)\geq 1-\Delta$, and $q_{i1}(\delta_{i1})=1$, $q_{i2}(\delta_{i2})=0$. This implies that vehicle 1 is expected to collect all the remaining reward of target i as if vehicle 2 did not exist. On the other hand, vehicle 2 is so far from target i compared to vehicle 1 that it expects no reward from the target as if the target did not exist.
2) Conversely, if $\delta_{i1}(t)\geq 1-\Delta$, $\delta_{i2}(t)=1-\delta_{i1}(t)\leq\Delta$, then $q_{i1}(\delta_{i1})=0$ and $q_{i2}(\delta_{i2})=1$ and vehicle 1 expects no reward from target i, while vehicle 2 expects the total remaining reward.
3) If $\Delta<\delta_{ij}\leq 1-\Delta$ for $j=1,2$, then $0<q_{ij}(\delta_{i1})<1$. In this case, both vehicles expect some fraction of the reward from target i.

These three cases characterize distinct cooperative relationships between vehicles. Since the evaluation of $q_{ij}(\delta_{ij})$ and of the cost function depends on the relative distances, a question that arises is: given the position of the two vehicles, can we find a target's positions in $\mathbb{R}^2$ which will lead to cases 1–3 respectively? This motivates the following definitions.

1) Given the positions of the two vehicles, all $y_i\in\mathbb{R}^2$, $i=1,\ldots,N$, such that $\delta_{i1}(t)\leq\Delta$ [case 1) above] define a set $S_1\subset\mathbb{R}^2$ which we call the *full responsibility region* for vehicle 1. This set is identical to the *invisibility region* for vehicle 2, denoted by I_2.
2) Similarly, all $y_i\in\mathbb{R}^2$, $i=1,\ldots,N$, such that $\delta_{i2}(t)\leq\Delta$ [case 2) above] define a set $S_2\subset\mathbb{R}^2$ which we call the *full responsibility region* for vehicle 2 and *invisibility region* set I_1 for vehicle 1.
3) We define the *cooperative region* for vehicles 1 and 2 to be $C_1=\mathbb{R}^2\setminus(S_1\cup I_1)=\mathbb{R}^2\setminus(S_2\cup I_2)=C_2$. This set contains all target positions which satisfy case 3) above.

Lemma 5: Suppose that by translating and rotating coordinates we set the position of vehicle 1 to $x_1=(0,0)$ and the position of vehicle 2 to $x_2=(c,0)$. Then $S_1=I_2$ is a circular region with center $(-(\Delta^2/(1-2\Delta))c,0)$ and radius $(\Delta(1-\Delta)/(1-2\Delta))c$. Similarly, $S_2=I_1$ is a circular region with center $((1+(\Delta^2/(1-2\Delta)))c,0)$ and radius $(\Delta(1-\Delta)/(1-2\Delta))c$.

Proof: See the Appendix. ∎

This result characterizes the regions S_1, S_2 as circular, nonoverlapping, and dependent only on the parameter Δ and the distance c between the vehicles.

A simple example can illustrate the fact that when $M=2$, the potential function $J(\mathbf{x})$ is not convex. This implies that multiple minima may be present in $J(\mathbf{x})$. If we can prove that every local minimum in the potential field corresponds to at least one vehicle located at a certain target, then by applying Theorem 3 we can conclude that a trajectory generated by the CRH controller leads to one such local minimum and, therefore, at least one vehicle converges to one of the targets. We remind the reader that after this visit the target point is removed from the set $\mathcal{T}$ and the process repeats with one fewer targets.

We will proceed by showing that condition (**C**) in Theorem 3 is indeed satisfied. We begin with the necessary condition for local optimality. Recall that in $J_i(x_1,x_2)$ both x_1 and x_2 are vectors in $\mathbb{R}^2$ (which means that the decision variable is a vector in $\mathbb{R}^4$). Then, $\nabla J_i(x_1,x_2)$ is given by $\nabla J_i(x_1,x_2)=[(\partial J_i/\partial x_1),(\partial J_i/\partial x_2)]^T$, which we can evaluate from (16) and (5) to get

$$\frac{\partial J_i}{\partial x_j}=\begin{cases}R_i\frac{x_j-y_i}{d_{ij}}, & \text{if } \delta_{ij}\in(0,\Delta]\\ \frac{R_i\left[2(1-\delta_{ij})^2-\Delta\right]}{(1-2\Delta)}\frac{x_j-y_i}{d_{ij}}, & \text{if } \delta_{ij}\in[\Delta,1-\Delta]\\ 0, & \text{if } \delta_{ij}\in[1-\Delta,1)\end{cases}\tag{26}$$

with $x_j=(x_{j1},x_{j2})$ and δ_{ij} defined by (4) with $M=2$.

Let H_i be the Hessian matrix of J_i. In order to evaluate H_i we have three cases to consider.

i) When $\delta_{i1}\leq\Delta$, $\delta_{i2}\geq 1-\Delta$: By checking the determinants of all upper left sub-matrices of H_i, we can see that H_i is positive semidefinite.
ii) When $\Delta<\delta_{ij}<1-\Delta$ (for $j=1,2$): To determine the eigenvalues of H_i, set $|\lambda I-H_i|=0$, where I is a 4×4 identity matrix. Solving this equation yields four eigenvalues for H_i : $\lambda_1=-(4\delta_{i1}\delta_{i2}^2/d_{i1})$, $\lambda_2=-(\Delta-2\delta_{i2}^2/d_{i1})$, $\lambda_3=-(4\delta_{i1}^2\delta_{i2}/d_{i2})$, $\lambda_4=-(\Delta-2\delta_{i1}^2/d_{i2})$, where $\lambda_1<0$ and $\lambda_3<0$. Thus, H_i is not positive semidefinite.
iii) When $\delta_{i1}\geq 1-\Delta$, $\delta_{i2}\leq\Delta$: This case is the dual of i) and H_i is positive semidefinite.

To summarize cases i)–iii), we have $H_i\succcurlyeq 0$ if $\delta_{ij}\leq\Delta$ or $\delta_{ij}\geq 1-\Delta$ and $H_i\not\succcurlyeq 0$ if $\delta_{ij}\in(\Delta,1-\Delta)$. It is difficult, from this point on, to develop a generalized condition under which all local minima in an arbitrary 2-vehicle N-target setting lead to stationary trajectories. Thus, we will proceed by investigating the first two cases with $N=1,2$.

Two-Vehicle, One-Target Case: For this setting, we have $J(x_1,x_2)=J_1(x_1,x_2)$ and we can derive the following result, which, along with Theorem 3, establishes the desired trajectory stationary.

Theorem 6: In a cooperative control setting with $M=2$ and $N=1$, the potential function $J(x_1,x_2)$ has exactly two local minima

$$\text{i) } x_1=y_1,\ x_2\neq y_1$$
$$\text{ii) } x_1\neq y_1,\ x_2=y_1$$

and both lead to stationary trajectories.

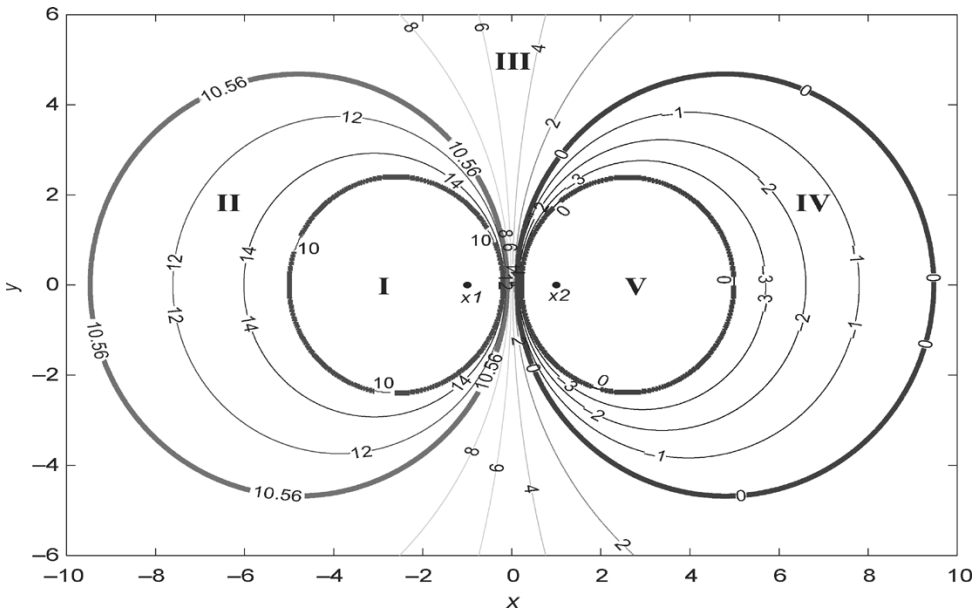


Fig. 4. Force exerted on a vehicle as a function of the target's position ($\Delta = 0.4$).

Proof: See the Appendix. ■

Two-Vehicle, Two-Target Case: For this setting, we have $J(x_1, x_2) = J_1(x_1, x_2) + J_2(x_1, x_2)$ and we derive the following result (the proof, which is omitted, is along the lines of that in the previous theorem, but requires checking nine different cases depending on the position of the two targets).

Theorem 7: In a cooperative control setting with $M = 2$ and $N = 2$ and $R_1 \neq R_2$, the potential function $J(x_1, x_2)$ has at most four local minima

$$\begin{aligned}
&\text{i) } x_1 = y_i \ s.t.\ i = \arg\max\{R_1, R_2\} \\
&\quad x_2 \in \{x : y_1, y_2 \in I_2(x)\} \\
&\text{ii) } x_2 = y_i \ s.t.\ i = \arg\max\{R_1, R_2\} \\
&\quad x_1 \in \{x : y_1, y_2 \in I_1(x)\} \\
&\text{iii) } x_1 = y_1,\ x_2 = y_2 \\
&\text{iv) } x_1 = y_2,\ x_2 = y_1
\end{aligned}$$

and all of these solutions lead to stationary trajectories.

Similar to the stationarity analysis of the 1-vehicle case, the partial derivative $\nabla_{x_j} J_i(x_1, x_2)$ can be viewed as the attraction force applied on vehicle j by target i. Thus

$$F_{ij} = c_i(x_j) \cdot f_i(x_j) \tag{27}$$

where

$$f_i(x_j) = \begin{cases} R_i \frac{x_j - y_i}{\|x_j - y_i\|}, & \text{if } x_j \neq y_i \\ 0, & \text{otherwise} \end{cases}$$

$$c_i(x_j) = \begin{cases} 1, & \text{if } y_i \in S_j \\ q_{ij} - \frac{(1-\delta_{ij})(2\delta_{ij}-1)}{(1-2\Delta)}, & \text{if } y_i \in C_j \\ 0, & \text{if } y_i \in I_j \end{cases}.$$

The attraction force exerted on vehicle j when target point i belongs to its full responsibility region S_j looks familiar, since it is exactly the same as if vehicle j were the only one in the mission space (see (25) in the 1-vehicle case). It is also easy to see why vehicle j has no attraction from target i if this target is in its invisibility region I_j. However, when $y_i \in C_j$, the expression for the force between vehicle j and target i needs some discussion.

Let us denote this force by F_{ij}^C and observe that it is composed of two separate forces, i.e., $F_{ij}^C = f_{ij}^c + f_{ij}^e$. The first force is an attraction force: $f_{ij}^c = -R_i q_{ij}(x_j - y_i/\|x_j - y_i\|)$. Recall that in the 2-vehicle case we have $q_{i1} + q_{i2} = 1$, which implies that $\|f_{i1}^c\| + \|f_{i2}^c\| = -R_i$. Thus, f_{ij}^c can be viewed as a partition of the total attraction force (whose magnitude is R_i) between two vehicles based on their relative proximity to i. As for the second force, i.e.,

$$f_{ij}^e = R_i \frac{(1-\delta_{ij})(2\delta_{ij}-1)}{(1-2\Delta)} \frac{x_j - y_i}{\|x_j - y_i\|} \tag{28}$$

note that it is an attraction force when $\delta_{ij} < 0.5$ and is otherwise a *repellent* force. It can be considered to be an *exclusive* force which is generated during the cooperation of two vehicles. The vehicle which is closer to target i will be further attracted to it and it therefore approaches it until the target enters its full responsibility region. Conversely, the vehicle which is further away from target i will experience a repellent force, and, on occasion, this repellent force directs the vehicle away from the target until it falls into its invisibility region. This type of behavior was observed in Fig. 3(b) involving vehicles P and BK.

In Fig. 4, we show the contours of the force exerted on vehicle 1 by target i as a function of the target's position y_i. In this figure, the two vehicles are positioned at $x_1 = (-1, 0)$, $x_2 = (1, 0)$ respectively and the target is in the rectangular region defined by $(-8, -5)$ and (8,5). In this contour graph, all target positions which generate the same force level are shown in the corresponding circles. The two small circular regions labeled I and V correspond to the full responsibility region S_1 and the invisibility region I_1, respectively. The remaining area is the cooperative region C_1. Note that if the target lies in the contour region labeled IV, the overall force vehicle 1 experiences is a repellent one, which pushes the vehicle further away from the target.

TABLE I
MISSION SPACE IS PARTITIONED INTO 5 REGIONS

No	Region Definition
I	$\delta_{i1} \in [0, \Delta]$
II	$\delta_{i1} \in \left(\Delta, \sqrt{\frac{\Delta}{2}}\right]$
III	$\delta_{i1} \in \left(\sqrt{\frac{\Delta}{2}}, 1-\sqrt{\frac{\Delta}{2}}\right)$
IV	$\delta_{i1} \in \left[1-\sqrt{\frac{\Delta}{2}}, 1-\Delta\right)$
V	$\delta_{i1} \in [1-\Delta, 1]$

TABLE II
COOPERATIVE BEHAVIORS IN DIFFERENT REGIONS

No	Force Description	Coop. Behavior Description
I	F_{i1} attractive, F_{i2} = 0	No cooperation, vehicle 1 takes care of target
II	F_{i1} attractive, F_{i2} repellent	Exclusive cooperation, vehicle 2 is excluded
III	F_{i1} attractive, F_{i2} attractive	Mutual cooperation, both vehicles proceed to target
IV	F_{i1} repellent, F_{i2} attractive	Exclusive cooperation, vehicle 1 is excluded
V	F_{i1} = 0 , F_{i2} attractive	No cooperation, vehicle 2 takes care of target

The different regions shown define a partition of the mission space. Depending on the location of a target at one of five regions, as shown in Fig. 4, the cooperative behavior of the vehicles present in the team is also different. Tables I and II summarize the five different types of cooperative behaviors and the corresponding region definition relative to a target point i.

One can see that by defining the function $q_{ij}(\delta_{ij})$ in different ways, the cooperative behavior of vehicles and the cooperative region definition may also change. However, regardless of the precise form of $q_{ij}(\delta_{ij})$ (as long as its defining monotonicity and normalization properties specified in Section II are not changed), an essential property of the relative proximity function $q_{ij}(\delta_{ij})$ is its ability to implicitly carry out dynamic resource allocation as illustrated in Fig. 4 for two vehicles and in Fig. 1(b) for four vehicles.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have considered a setting where multiple vehicles form a team cooperating to visit multiple target points in order to collect rewards associated with them and maximize the total reward accumulated over a given time interval. The CRH control scheme we have proposed dynamically determines vehicle trajectories by solving a sequence of relatively simple optimization problems over a planning horizon and executing them over a shorter action horizon. A noteworthy "stationarity" property of this scheme is that it guides vehicles to target points without any explicit assignment problem involved, thus bypassing the need to deal with combinatorially explosive problems. We have obtained conditions under which this property can be guaranteed in an arbitrary M-vehicle N-target point case and analyzed the 1-vehicle, N-target point case for which we can derive a simple condition for obtaining such a stationary trajectory. A similar analysis can be used for 2-vehicle settings and the q_{ij} function can be seen to act in a way that induces distinct cooperative behaviors. However, the analysis becomes extremely tedious for the $M > 2$ case and different techniques may be required to verify that the conditions derived still apply. The application of the proposed RH control scheme includes online task assignment and path planning for multivehicle cooperative missions where the event-driven nature of the RH controller well suits the uncertain mission environment. This centralized controller also acts as a basis for the development of a distributed version (see [21]), which has been successfully implemented in a multiple robot real-time testbed (see [26]).

APPENDIX

A. Proof of Lemma 1

The fact that $J(\mathbf{x}_{k+1}) < J(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{I}_k$ is established by contradiction. We assume that $J(\mathbf{x}_{k+1}) \geq J(\mathbf{w})$ for some $\mathbf{w} \in \mathcal{I}_k$. Since $J(\mathbf{x}_{k+1}) = \min_{\mathbf{x}\in\mathcal{F}_k} J(\mathbf{x})$, it follows that $J(\mathbf{x}) \geq J(\mathbf{w})$ for all $\mathbf{x} \in \mathcal{F}_k$. Therefore, since $J(\cdot)$ is continuous, this further implies that in region $\mathcal{I}_k$ there must exist at least one local minimum, i.e., there exists some $\bar{\mathbf{x}} \in \mathcal{I}_k$ such that $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_M) = \mathbf{x}^l$ for some $l = 1, \ldots, L$. Since $H_k = \min_{j\in\mathcal{A}, i\in\mathcal{T}}\{\|y_i - x_j(t_k)\|/V\}$, we have

$$\|y_i - x_j(t_k)\| \geq V H_k, \qquad \text{for all } j \in \mathcal{A},\ i \in \mathcal{T}$$

thus $y_i \notin \mathcal{I}_k^j$ for all i, j. This further means that $\bar{x}_j \neq y_i$ for all i, j. But this is in contridiction with the given condition (**C**), i.e., since $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_M) \in \mathbb{R}^{2M}$ is a local minimum of $J(\mathbf{x})$, we must have $\bar{x}_j = y_i$ for some $i \in \mathcal{T}$ and $j \in \mathcal{A}$. This completes the proof of $J(\mathbf{x}_{k+1}) < J(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{I}_k$ and it immediately follows that $J(\mathbf{x}_{k+1}) \leq J(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{G}_k$. ■

B. Proof of Lemma 2

For $J(\mathbf{x}) = \sum_{n\in\mathcal{T}} \sum_{m\in\mathcal{A}} R_n \|x_m - y_n\| q_{nm}$,

$$\frac{\partial J}{\partial x_m} = \sum_{n\in\mathcal{T}, m\in\mathcal{B}_n} \frac{\partial}{\partial x_m} R_n \left(\|x_m - y_n\| q_{nm} + \|x_v - y_n\| q_{nv}\right),$$

where $\mathcal{B}_n = \{m, v\}$ is the neighbor set of target n (i.e., m and v are the vehicles which constitute $\mathcal{B}_n$). Since $q_{nv} + q_{nm} = 1$ [see (5)],

$$\frac{\partial J}{\partial x_m} = \sum_{n\in\mathcal{T}, m\in\mathcal{B}_n} R_n \left[\frac{x_m - y_n}{\|x_m - y_n\|} q_{nm} + \frac{\partial q_{nm}}{\partial x_m}\left(\|x_m - y_n\| - \|x_v - y_n\|\right)\right]. \quad (29)$$

Since $\mathbf{x}^l = (x_1^l, \ldots, x_M^l)$ as defined is a local minimum of $J(\mathbf{x})$ and $x_j^l = y_i$ for some $i \in \mathcal{T}$, $j \in \mathcal{A}$ (i.e., $\mathbf{x}^l$ corresponds to vehicle j located at target i), we have $(\partial J/\partial x_m)|_{x_m = x_m^l} = 0$ for all $m \neq j$. Since $x_j^l = y_i$, $\partial J/\partial x_j|_{x_j = x_j^l}$ cannot be defined by (29), so we concentrate on the neighborhood of x_j^l instead. Let $x_j = y_i + \varepsilon$ where ε is an arbitrary vector in $\mathbb{R}^2$ sufficiently small to ensure $\delta_{ij} \leq \Delta_i$ as defined in (4) and (5), so that from

the definition of q_{ij} in (5) we have $q_{ij}=1$ and $\partial q_{ij}/\partial x_j=0$. Thus

$$\left.\frac{\partial J}{\partial x_j}\right|_{x_j=y_i+\varepsilon}=\sum_{n\in\mathcal{T},j\in\mathcal{B}_n,n\neq i}R_n\left[\frac{y_i+\varepsilon-y_n}{\|y_i+\varepsilon-y_n\|}q_{nj}+\left.\frac{\partial q_{nj}}{\partial x_j}\right|_{x_j=y_i+\varepsilon}\left(\|y_i+\varepsilon-y_n\|-\left\|x_v^l-y_n\right\|\right)\right]+R_i\frac{\varepsilon}{\|\varepsilon\|}.$$

The inner product of $(\partial J/\partial x_j)|_{x_j=y_i}+\varepsilon$ with a unit vector $\varepsilon/\|\varepsilon\|$ gives

$$\left.\frac{\partial J}{\partial x_j}\right|_{x_j=y_i+\varepsilon}\cdot\frac{\varepsilon}{\|\varepsilon\|}=\left[\sum_{n\in\mathcal{T},j\in\mathcal{B}_n,n\neq i}R_n\left(\frac{y_i+\varepsilon-y_n}{\|y_i+\varepsilon-y_n\|}q_{nj}+\left.\frac{\partial q_{nj}}{\partial x_j}\right|_{x_j=y_i+\varepsilon}\left(\|y_i+\varepsilon-y_n\|-\left\|x_v^l-y_n\right\|\right)\right)\right]\cdot\frac{\varepsilon}{\|\varepsilon\|}+R_i.$$

Taking limits with $\varepsilon\to 0$, we obtain the equation shown at the bottom of the page. The inner product on the right-hand side of this equation is minimized at $-\|\cdot\|$, where $\|\cdot\|$ is the norm of the term in square brackets. Therefore, the minimum of the right-hand side is

$$\gamma(\mathbf{x}^l)\equiv R_i-\left\|\sum_{n\in\mathcal{T},j\in\mathcal{B}_n,n\neq i}R_n\left(\frac{y_i-y_n}{\|y_i-y_n\|}q_{nj}+\left(\|y_i-y_n\|-\left\|x_v^l-y_n\right\|\right)\left.\frac{\partial q_{nj}}{\partial x_j}\right|_{x_j=y_i}\right)\right\|$$

and it follows that

$$\lim_{\varepsilon\to 0}\left.\frac{\partial J}{\partial x_j}\right|_{x_j=y_i+\varepsilon}\cdot\frac{\varepsilon}{\|\varepsilon\|}\geq\gamma(\mathbf{x}^l). \tag{30}$$

Since $\mathbf{x}^l$ is a local minimum of $J(\mathbf{x})$, we must have $(\partial J/\partial x_j)|_{x_j=y_i+\varepsilon}\cdot(\varepsilon/\|\varepsilon\|)>0$ for all small ε so that x_j is in the neighborhood of y_i. Therefore, $\lim_{\varepsilon\to 0}(\partial J/\partial x_j)|_{x_j=y_i+\varepsilon}\cdot(\varepsilon/\|\varepsilon\|)>0$ for any ε. Since the above limit is minimized by $\gamma(\mathbf{x}^l)$, it follows that $\gamma(\mathbf{x}^l)>0$.

Finally, for any ε, the inner product $\partial J/\partial x_j\cdot\varepsilon/\|\varepsilon\|$ implies that $\|(\partial J/\partial x_j)\|\geq(\partial J/\partial x_j)\cdot\varepsilon/\|\varepsilon\|$, therefore

$$\lim_{\varepsilon\to 0}\left\|\frac{\partial J}{\partial x_j}\right\|\geq\lim_{\varepsilon\to 0}\frac{\partial J}{\partial x_j}\cdot\frac{\varepsilon}{\|\varepsilon\|}. \tag{31}$$

Combining (31), (30), and $\gamma(\mathbf{x}^l)>0$, we have $\lim_{\varepsilon\to 0}\|(\partial J(\mathbf{x})/\partial x_j)|_{x_j=y_i+\varepsilon}\|\geq\gamma(\mathbf{x}^l)>0$ which completes the proof. ■

C. Proof of Theorem 3

We will proceed by establishing the following two facts: i) $J(\mathbf{x})\geq 0$ for all $\mathbf{x}\in\mathbb{R}^{2M}$; and ii) before vehicles reach a local minimum, $J(\mathbf{x}_k)-J(\mathbf{x}_{k+1})$ is bounded from below by a positive constant γ independent of k. These two facts guarantee that the CRH trajectory converges to a local minimum in a finite number of steps.

i) $J(x)\geq 0$ for all $x\in R^{2M}$. The fact that

$$J(\mathbf{x})\geq 0 \tag{32}$$

easily follows from the definition of $J(\mathbf{x})$, i.e., $J(\mathbf{x})=J(x_1,\ldots,x_M)=\sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{A}}R_i\|x_j-y_i\|q_{ij}$, where $R_i\geq 0$, $\|x_j-y_i\|\geq 0$, $q_{ij}\geq 0$ for all $i\in\mathcal{T}$, $j\in\mathcal{A}$.

ii) Before vehicles reach a local minimum, $J(x_k)-J(x_{k+1})$ is bounded from below by a positive constant γ independent of k. To prove this statement, we build a lower bound for $J(\mathbf{x}_k)-J(\mathbf{x}_{k+1})$ by comparing the CRH trajectory to an *artificially constructed trajectory* which, in the kth step, terminates at $\mathbf{g}_{k+1}\in\mathcal{G}_k$ (see Fig. 5). Using Lemma 1, we have $J(\mathbf{x}_{k+1})\leq J(\mathbf{x})$ for all $\mathbf{x}\in\mathcal{G}_k$, which implies that

$$J(\mathbf{x}_k)-J(\mathbf{x}_{k+1})\geq J(\mathbf{x}_k)-J(\mathbf{g}_{k+1}). \tag{33}$$

Thus, a lower bound for $J(\mathbf{x}_k)-J(\mathbf{g}_{k+1})$ will also be a lower bound for $J(\mathbf{x}_k)-J(\mathbf{x}_{k+1})$. We will now proceed by dividing the proof into two steps: step ii.A) concentrates on the construction of a trajectory used as a baseline for the analysis, while in step ii.B), we analyze this baseline trajectory, which helps us determine a lower bound for $J(\mathbf{x}_k)-J(\mathbf{x}_{k+1})$.

ii.A) Construction of baseline trajectory. For each step k, the constructed trajectory starts at $\mathbf{x}_k$ and it is generated by letting vehicles follow carefully designed paths. If the locations of vehicles are such that $\mathbf{x}\notin D_p(\eta)$ for all $p=1,\ldots,P$, a gradient-based method is applied to generate vehicle trajectories. Otherwise, when $\mathbf{x}\in D_p(\eta)$ for some $p=1,\ldots,P$, vehicles switch to different motion dynamics. Specifically, when the locations of vehicles are such that $\mathbf{x}\notin\mathcal{D}_p(\eta)$ for all $p=1,\ldots,P$, letting

$$\nabla_j J(\mathbf{x})=\frac{\partial J(\mathbf{x})}{\partial x_j},\qquad\text{for all } j\in\mathcal{A}$$

the gradient-based vehicle trajectories are given for each $j\in\mathcal{A}$ by

$$\dot{x}_j(t)=\begin{cases}-\frac{\nabla_j J(\mathbf{x})}{\|\nabla_j J(\mathbf{x})\|}V, & \text{if } \mathbf{x} \text{ is not a local minimum}\\ 0, & \text{if } \mathbf{x} \text{ is a local minimum}\end{cases}. \tag{34}$$

On the other hand, when $\mathbf{x}\in\mathcal{D}_p(\eta)$ for some $p=1,\ldots,P$, under Assumption 2 vehicles must be in the neighborhood of a local maximum or saddle point, say $\mathbf{x}^p$. In this case, vehicles

$$\lim_{\varepsilon\to 0}\left.\frac{\partial J}{\partial x_j}\right|_{x_j=y_i+\varepsilon}\cdot\frac{\varepsilon}{\|\varepsilon\|}=R_i+\lim_{\varepsilon\to 0}\left\{\left[\sum_{n\in\mathcal{T},j\in\mathcal{B}_n,n\neq i}R_n\left(\frac{y_i-y_n}{\|y_i-y_n\|}q_{nj}+\left.\frac{\partial q_{nj}}{\partial x_j}\right|_{x_j=y_i}\left(\|y_i-y_n\|-\left\|x_v^l-y_n\right\|\right)\right)\right]\cdot\frac{\varepsilon}{\|\varepsilon\|}\right\}$$

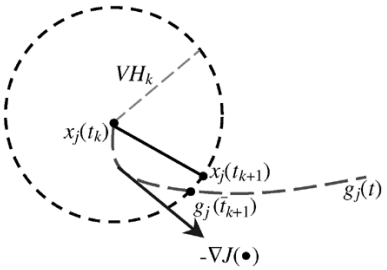


Fig. 5. Comparing the CRH trajectory and the constructed trajectory.

switch to different motion dynamics under which vehicles head in straight lines toward a point $\mathbf{w}_k^p = (w_{k,1}^p, \ldots, w_{k,M}^p) \in \mathbb{R}^{2M}$ which is obtained by

$$\mathbf{w}_k^p = \arg\min_{\mathbf{x}} J(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \mathcal{D}_p(\eta) \cap \mathcal{G}_k. \tag{35}$$

Formally, when $\mathbf{x} \in \mathcal{D}_p(\eta)$, vehicles trajectories are given by

$$\dot{x}_j(t) = \frac{w_{k,j}^p - x_j(t)}{\left\| w_{k,j}^p - x_j(t) \right\|} V. \tag{36}$$

The following observation is made to ensure that driven by (36), vehicles will depart from $D_p(\eta)$ and will not re-enter it again. By Lemma 2, the region $\mathcal{D}_p(\eta) \cap \mathcal{G}_k$ does not contain local minima of $J(\mathbf{x})$, therefore, $\mathbf{w}_k^p$ must be on the boundary of $\mathcal{D}_p(\eta) \cap \mathcal{G}_k$. If $\mathcal{D}_p(\eta) \cap \mathcal{G}_k = \mathcal{D}_p(\eta)$ (i.e., $\mathcal{G}_k$ contains the entire $\mathcal{D}_p(\eta)$), then $\mathbf{w}_k^p$ is on the boundary of $\mathcal{D}_p(\eta)$. Then, at point $\mathbf{w}_k^p$, when the vehicle motion dynamics are switched back to (34), under the effect of gradient descent, vehicles will move out of $\mathcal{D}_p(\eta)$ (otherwise, the value of $J(\mathbf{x})$ would increase). In addition, since $J(\mathbf{x})$ will keep decreasing thereafter, it is guaranteed that vehicles will not re-enter $\mathcal{D}_p(\eta)$ again. On the other hand, if $\mathbf{w}_k^p$ is on the boundary of $\mathcal{G}_k$ (i.e., $\mathbf{w}_k^p \in \mathcal{F}_k$), then by leading vehicles to $\mathbf{w}_k^p$, the constructed trajectory crosses the CRH decision space $\mathcal{F}_k$ at $\mathbf{w}_k^p$, and it terminates thereafter.

ii.B) Properties of baseline trajectory. Let us now examine the baseline trajectory $\mathbf{g}(t) = \{g_1(t), \ldots, g_M(t)\}, t \geq t_k$, constructed through the motion dynamics (34), (36). This will help us determine a lower bound for $J(\mathbf{x}_k) - J(\mathbf{x}_{k+1})$. We will proceed by first defining the termination point of $\mathbf{g}(t)$ in the kth step. Let $\bar{t}_{k+1}$ be the earliest time that any vehicle, say j, crosses $\mathcal{F}_k^j$, i.e.,

$$\bar{t}_{k+1} = \min \left\{ t : t > t_k,\ g_j(t) \in \mathcal{F}_k^j,\ \text{for some } j \in \mathcal{A} \right\} \tag{37}$$

and we define the termination point as $\mathbf{g}_{k+1} \equiv \mathbf{g}(\bar{t}_{k+1})$. We can see that, by this definition, $\mathbf{g}_{k+1} \in \mathcal{G}_k$. We will show later in (46) that $\bar{t}_{k+1}$ is finite. Another fact we should notice is that

$$\bar{t}_{k+1} \geq t_{k+1}, \qquad \text{for all } k. \tag{38}$$

This is because at step k, the CRH trajectory consists of straight lines between $x_j(t_k)$ and $x_j(t_{k+1})$, which is the fastest way vehicles can reach $\mathcal{F}_k$. A constructed trajectory for vehicle j is illustrated in Fig. 5, where this vehicle starts at $x_j(t_k)$ and by following the motion dynamics (34), (36), at time $\bar{t}_{k+1}$, it crosses $\mathcal{F}_k^j$ at $g_j(\bar{t}_{k+1})$. Let us now concentrate on the segment of $\mathbf{g}(t)$ where $t \in [t_k, \bar{t}_{k+1}]$ and partition it into two distinct sets. Along the trajectory $\mathbf{g}(t)$, we now define

$$\mathcal{U}_k(\eta) = \{ t : t \in [t_k, \bar{t}_{k+1}], \mathbf{g}(t) \notin \mathcal{D}_p(\eta), \text{ for all } p = 1, \ldots, P \}$$

and

$$\mathcal{V}_k(\eta) = \{ t : t \in [t_k, \bar{t}_{k+1}], \mathbf{g}(t) \in \mathcal{D}_p(\eta), \text{ for some } p = 1, \ldots, P \}.$$

These are the time sets over which (34) or (36), respectively, is active. We then define

$$S_{\mathcal{U}}(\eta) = \int_{t \in \mathcal{U}_k(\eta)} dt \quad \text{and} \quad S_{\mathcal{V}}(\eta) = \int_{t \in \mathcal{V}_k(\eta)} dt.$$

Since, for any $t \in [t_k, \bar{t}_{k+1}]$, either $\mathbf{g}(t) \notin \mathcal{D}_p(\eta)$ or $\mathbf{g}(t) \in \mathcal{D}_p(\eta)$, we have

$$\bar{t}_{k+1} - t_k = S_{\mathcal{U}}(\eta) + S_{\mathcal{V}}(\eta). \tag{39}$$

We will now derive an upper bound for $S_{\mathcal{V}}(\eta)$, the amount of time spent by $\mathbf{g}(t)$ in some $\mathcal{D}_p(\eta)$. Let $\mathcal{P} \subseteq \{1, \ldots, P\}$ be the set of instances such that $\mathbf{g}(t)$ crosses $\mathcal{D}_p(\eta) \cap \mathcal{G}_k$ for all $p \in \mathcal{P}$ during $[t_k, \bar{t}_{k+1}]$. Without loss of generality, let vehicles enter $\mathcal{D}_p(\eta) \cap \mathcal{G}_k$ at time t_1^p for some $p \in \mathcal{P}$ and, under motion dynamics (36), vehicles reach $\mathbf{w}_k^p$ at time t_2^p. Thus

$$\mathcal{V}_k(\eta) = \cup_{p \in \mathcal{P}} [t_1^p, t_2^p]. \tag{40}$$

Under (36), we also have $t_2^p - t_1^p = \|\mathbf{w}_k^p - \mathbf{g}(t_1^p)\|/V$. Since both $\mathbf{w}_k^p$ and $\mathbf{g}(t_1^p)$ are in the circular region $\mathcal{D}_p(\eta)$, we have $\|\mathbf{w}_k^p - \mathbf{g}(t_1^p)\| \leq 2r_p(\eta)$, where $r_p(\eta)$ was defined in (20). Thus, $t_2^p - t_1^p \leq 2r_p(\eta)/V$. In the worst case, starting from $\mathbf{x}_k$, $\mathbf{g}(t)$ crosses $\mathcal{D}_p(\eta)$ for all $p = 1, \ldots, P$, and $\mathbf{g}(t)$ crosses each $\mathcal{D}_p(\eta)$ by following its diameter. This worst case gives an upper bound for $S_{\mathcal{V}}(\eta)$, that is

$$S_{\mathcal{V}}(\eta) \leq \frac{2}{V} \sum_{p=1}^{P} r_p(\eta).$$

Because of (39), this also offers a lower bound for $S_{\mathcal{U}}(\eta)$, that is

$$\begin{aligned} S_{\mathcal{U}}(\eta) &= (\bar{t}_{k+1} - t_k) - S_{\mathcal{V}}(\eta) \\ &\geq (\bar{t}_{k+1} - t_k) - \frac{2}{V} \sum_{p=1}^{P} r_p(\eta). \end{aligned} \tag{41}$$

After partitioning $[t_k, \bar{t}_{k+1}]$ into sets and deriving bounds for them, now let us consider the decrease of $J(\mathbf{x})$ along a trajectory $\mathbf{g}(t)$ over $[t_k, \bar{t}_{k+1}]$. By partitioning $[t_k, \bar{t}_{k+1}]$ into $\mathcal{U}_k(\eta)$ and $\mathcal{V}_k(\eta)$ and using (40), we obtain

$$\begin{aligned} J(\mathbf{x}_k) - J(\mathbf{g}_{k+1}) = &\sum_{p \in \mathcal{P}} \left[J\left(\mathbf{g}\left(t_1^p\right)\right) - J\left(\mathbf{g}\left(t_2^p\right)\right) \right] \\ &- \int_{\{\mathbf{g}(t) : t \in \mathcal{U}_k(\eta)\}} \nabla J\left(\mathbf{g}(t)\right) \cdot d\mathbf{g}(t). \end{aligned} \tag{42}$$

The first term on the right-hand side of the equation corresponds to the decrease of $J(\mathbf{x})$ when $t \in \mathcal{V}_k(\eta)$. Since $\mathbf{g}(t_1^p) \in \mathcal{D}_p(\eta) \cap$

$\mathcal{G}_k$, $\mathbf{g}(t_2^p) = \mathbf{w}_k^p$ and $\mathbf{w}_k^p$ minimizes $J(\mathbf{x})$ over $\mathcal{D}_p(\eta) \cap \mathcal{G}_k$ as defined in (35), we have

$$\sum_{p\in\mathcal{P}} \left[J\left(\mathbf{g}\left(t_1^p\right)\right) - J\left(\mathbf{g}\left(t_2^p\right)\right)\right] = \sum_{p\in\mathcal{P}} \left[J\left(\mathbf{g}\left(t_1^p\right)\right) - J\left(\mathbf{w}_k^p\right)\right] \geq 0$$

and it follows from (42) that

$$J(\mathbf{x}_k) - J(\mathbf{g}_{k+1}) \geq - \int\limits_{\{\mathbf{g}(t):t\in\mathcal{U}_k(\eta)\}} \nabla J\left(\mathbf{g}(t)\right)\cdot d\mathbf{g}(t) \equiv -\mathcal{L}. \tag{43}$$

Now, let us try to find a lower bound for $-\mathcal{L}$, the second term in the right-hand-side of (42). Recalling that $\nabla J(\mathbf{g}(t)) = [\nabla_1 J(\mathbf{g}(t)), \ldots, \nabla_M J(\mathbf{g}(t))]$ and $d\mathbf{g}(t) = [g_1(t), \ldots, g_M(t)]$, the right-hand side is

$$\begin{aligned} -\mathcal{L} &= - \int\limits_{\{\mathbf{g}(t):t\in\mathcal{U}_k(\eta)\}} \sum_{j\in\mathcal{A}} \nabla_j J\left(\mathbf{g}(t)\right)\cdot dg_j(t) \\ &= - \int\limits_{\{\mathbf{g}(t):t\in\mathcal{U}_k(\eta)\}} \sum_{j\in\mathcal{A}} \nabla_j J\left(\mathbf{g}(t)\right)\cdot \dot{g}_j(t)dt. \end{aligned}$$

Using the motion dynamics (34), we get

$$-\mathcal{L} = \int\limits_{t\in\mathcal{U}_k(\eta)} \sum_{j\in\mathcal{A}} \left\|\nabla_j J\left(\mathbf{g}(t)\right)\right\| V dt$$

and by applying the triangle inequality

$$-\mathcal{L} \geq \int\limits_{t\in\mathcal{U}_k(\eta)} \left\|\nabla J\left(\mathbf{g}(t)\right)\right\| V dt.$$

Since $\mathbf{g}(t) \notin \mathcal{D}_p(\eta)$ for $t \in \mathcal{U}_k(\eta)$ and all $p = 1, \ldots, P$, and under Assumption 2

$$\int\limits_{t\in\mathcal{U}_k(\eta)} \left\|\nabla J\left(\mathbf{g}(t)\right)\right\| V dt \geq \eta V \int\limits_{t\in\mathcal{U}_k(\eta)} dt = \eta V S_{\mathcal{U}}(\eta).$$

Thus

$$-\mathcal{L} \geq \eta V S_{\mathcal{U}}(\eta)$$

and, returning to (43), we obtain

$$J(\mathbf{x}_k) - J(\mathbf{g}_{k+1}) \geq \eta V S_{\mathcal{U}}(\eta). \tag{44}$$

This inequality implies two important facts. First, it leads to the conclusion that $\bar{t}_{k+1} < \infty$ (i.e., $\mathbf{g}(t)$ crosses $\mathcal{F}_k$ in finite time). This is because (32) implies that $J(\mathbf{g}_{k+1}) \geq 0$, therefore, $J(\mathbf{x}_k) \geq J(\mathbf{x}_k) - J(\mathbf{g}_{k+1})$. In addition, using (41), we have

$$\eta V S_{\mathcal{U}}(\eta) \geq \eta V \left[(\bar{t}_{k+1} - t_k) - \frac{2}{V}\sum_{p=1}^{P} r_p(\eta)\right] \tag{45}$$

so that (44) yields an upper bound for $\bar{t}_{k+1}$

$$\bar{t}_{k+1} \leq t_k + \frac{J(\mathbf{x}_k)}{\eta V} + \frac{2}{V}\sum_{p=1}^{P} r_p(\eta). \tag{46}$$

The second implication of (44) is that it offers a lower bound for $J(\mathbf{x}_k) - J(\mathbf{x}_{k+1})$, which is obtained as follows. Since $\bar{t}_{k+1} \geq t_{k+1}$ from (38), and recalling that $t_{k+1} - t_k = H_k$, it follows from (45) that

$$S_{\mathcal{U}}(\eta) \geq (t_{k+1} - t_k) - \frac{2}{V}\sum_{p=1}^{P} r_p(\eta) = H_k - \frac{2}{V}\sum_{p=1}^{P} r_p(\eta).$$

Since, by assumption, no vehicle is located at any target point at t_k, we have $H_k > s$, where $s = \min_{i\in\mathcal{T}} s_i$. Thus

$$S_{\mathcal{U}}(\eta) > s - \frac{2}{V}\sum_{p=1}^{P} r_p(\eta) \tag{47}$$

and by combining (33), (44), and (47), we get

$$J(\mathbf{x}_k) - J(\mathbf{x}_{k+1}) > \eta V \left[s - \frac{2}{V}\sum_{p=1}^{P} r_p(\eta)\right].$$

Since $r_p(\eta) \geq 0$, $\lim_{\eta\to 0} r_p(\eta) = 0$ and $\lim_{\eta\to 0}[s - (2/V)\sum_{p=1}^{P} r_p(\eta)] > 0$, we can select some $\eta^* > 0$ such that $s^* \equiv s - (2/V)\sum_{p=1}^{P} r_p(\eta^*) > 0$. Therefore

$$J(\mathbf{x}_k) - J(\mathbf{x}_{k+1}) > \eta^* s^* V > 0, \qquad \text{for all } k.$$

Now, we conclude the proof of ii). By setting $\gamma \equiv \eta^* s^* V$

$$J(\mathbf{x}_k) - J(\mathbf{x}_{k+1}) > \gamma, \qquad \text{for all } k. \tag{48}$$

Finally, by combining (32) and (48), we can derive an upper bound for the total number of steps required for a CRH trajectory to converge to a local minimum. If at $t = 0$ vehicles start at some $\mathbf{x}_0$, then clearly the CRH trajectory is guaranteed to converge to a local minimum in $K = \lceil J(\mathbf{x}_0)/\gamma \rceil$ steps. ■

D. Proof of Theorem 4

To prove i), we begin with the definition of $J(x)$ in (15) and (16), where we see that

$$J(x) = \sum_{i\in\mathcal{T}} R_i \|x - y_i\| = \sum_{i\in\mathcal{T}} R_i \left[(x - y_i)^T (x - y_i)\right]^{\frac{1}{2}}$$

from which it follows that

$$\nabla J(x) = \sum_{i\in\mathcal{T}} R_i \frac{x - y_i}{\|x - y_i\|}. \tag{49}$$

Since $\lim_{x\to\infty} J(x) = \infty$ and $J(x)$ is a convex function, a global minimum in $\mathbb{R}^2$ exists. Since we have assumed that $x_p(t)$ is nonstationary, there exists some $y_e \neq y_i$, $i = 1, \ldots, N$, such that y_e is the global minimum of $J(x)$ and we must have

$$\nabla J(y_e) = \sum_{i\in\mathcal{T}} R_i \frac{y_e - y_i}{\|y_e - y_i\|} = 0.$$

Next, to prove ii), assume that there exists some i that satisfies (23). We will show that y_i is the global minimum of $J(x)$. Since $\nabla J(x)$ is not analytic at such y_i, let us consider $\nabla J(y_i + \varepsilon)$ for some arbitrarily small $\varepsilon \in \mathbb{R}^2$. We have

$$\begin{aligned} \lim_{\varepsilon\to 0} \nabla J(y_i + \varepsilon) &= \lim_{\varepsilon\to 0} \sum_{j=1}^{N} R_j \frac{(y_i + \varepsilon) - y_j}{\|(y_i + \varepsilon) - y_j\|} \\ &= \sum_{j\in\mathcal{T},\ j\neq i} R_j \frac{y_i - y_j}{\|y_i - y_j\|} + \lim_{\varepsilon\to 0} R_i \frac{\varepsilon}{\|\varepsilon\|} \end{aligned}$$

where $\varepsilon/\|\varepsilon\| \equiv e_\varepsilon$ is a unit vector. Then

$$\lim_{\varepsilon\to 0} \nabla^T J(y_i+\varepsilon)\cdot e_\varepsilon = \sum_{j\in\mathcal{T},\, j\neq i} R_j \frac{(y_i-y_j)^T}{\|y_i-y_j\|}\cdot e_\varepsilon + R_i. \quad (50)$$

Note that for an arbitrary unit vector e_ε

$$-\left\|\sum_{j\in\mathcal{T},\, j\neq i} R_j \frac{y_i-y_j}{\|y_i-y_j\|}\right\| \leq \sum_{j\in\mathcal{T},\, j\neq i} R_j \frac{(y_i-y_j)^T}{\|y_i-y_j\|}\cdot e_\varepsilon \leq \left\|\sum_{j\in\mathcal{T},\, j\neq i} R_j \frac{y_i-y_j}{\|y_i-y_j\|}\right\|.$$

Therefore

$$\lim_{\varepsilon\to 0} \nabla^T J(y_i+\varepsilon)\cdot e_\varepsilon = \sum_{j\in\mathcal{T},\, j\neq i} R_j \frac{(y_i-y_j)^T}{\|y_i-y_j\|}\cdot e_\varepsilon + R_i \geq R_i - \left\|\sum_{j\in\mathcal{T},\, j\neq i} R_j \frac{y_i-y_j}{\|y_i-y_j\|}\right\|$$

If (23) holds, this implies

$$\lim_{\varepsilon\to 0} \nabla^T J(y_i+\varepsilon)\cdot e_\varepsilon \geq R_i - \left\|\sum_{j\in\mathcal{T},\, j\neq i} R_j \frac{y_i-y_j}{\|y_i-y_j\|}\right\| > 0$$

for every ε. This is a necessary and sufficient condition for $J(y_i)$ to be the unique global minimum.

It remains to determine an upper bound for the trajectory convergence time t_f. We proceed by finding a lower bound for $J(x_k) - J(x_{k+1})$. Toward this goal, we compare two trajectories: The trajectory $x(t)$ generated by the RH controller, which ends at x_{k+1} at $t_{k+1} = t_k + H_k$, and the trajectory $g(t)$ generated by a gradient method, which will terminate at g_{k+1} (see also Fig. 5). From Lemma 1, we have $J(x_k) - J(x_{k+1}) \geq J(x_k) - J(g_{k+1})$. Thus, a lower bound for $J(x_k) - J(g_{k+1})$ will also be a lower bound for $J(x_k) - J(x_{k+1})$.

The constructed trajectory is based on the motion dynamics

$$\dot{x} = \begin{cases} -\frac{\nabla J(x)}{\|\nabla J(x)\|}V, & \text{if } \|\nabla J(x)\| > 0 \\ 0, & \text{if } \|\nabla J(x)\| = 0. \end{cases} \quad (51)$$

This is similar to the proof of Theorem 3 without the need for a second mode in the motion dynamics. Now let us evaluate $J(x_k) - J(g_{k+1})$, where $\bar{t}_{k+1}$ as defined in the proof of Theorem 3 [see (37)], is the termination time of $g(t)$

$$J(x_k) - J(g_{k+1}) = \int_{\{g(t):t_k\leq t\leq \bar{t}_{k+1}\}} -\nabla J\,(g(t))\cdot dg(t). \quad (52)$$

Under (51), we see that for any point on $g(t)$, the steepest descent direction is always the tangent direction of $g(t)$, hence $-\nabla J(g(t))\cdot dg(t) = \|\nabla J(g(t))\| V dt$ and (52) becomes

$$J(x_k) - J(g_{k+1}) = \int_{\{g(t):t_k\leq t\leq \bar{t}_{k+1}\}} \|\nabla J\,(g(t))\|\, V dt$$

Since $J(x)$ is a convex function, we have: $\|\nabla J(x)\| \geq \|\nabla J(x^*)\|$ for every $x \in \mathbb{R}^2$ and x^* being the global minimum of $J(x)$. Under the condition in (23), we saw that $\|\nabla J(x^*)\| = \lim_{x\to x^*} \|\nabla J(x)\| = \gamma$. Thus

$$J(x_k) - J(g_{k+1}) \geq \gamma V \int_{\{g(t):t_k\leq t\leq \bar{t}_{k+1}\}} dt = \gamma V\,(\bar{t}_{k+1} - t_k).$$

Since $\bar{t}_{k+1} > t_{k+1}$ (see (38) in the proof of Theorem 3), then $\bar{t}_{k+1} - t_k > t_{k+1} - t_k = H_k$. We further obtain $J(x_k) - J(g_{k+1}) \geq \gamma V H_k$ and, from Lemma 1, $J(x_{k+1}) \leq J(g_{k+1})$. Therefore

$$J(x_k) - J(x_{k+1}) \geq J(x_k) - J(g_{k+1}) \geq \gamma V H_k$$

which gives a lower bound for the cost improvement the RH controller can provide in each iteration. Setting $s = \min_{i\in\mathcal{T}} s_i$, we have $H_k > s$ (otherwise the vehicle is already at a target point), and we get

$$J(x_k) - J(x_{k+1}) > \gamma V s.$$

It follows that if at $t = 0$ the vehicle is located at x_0, then the RH controller is guaranteed to converge in $n_f \leq \lceil (J(x_0) - J(y_i))/\gamma V s \rceil$ steps, or by time $t_f \leq ((J(x_0) - J(y_i))/\gamma V)$. ■

E. Proof of Lemma 5

For the full responsibility region S_1 of x_1, its boundary can be expressed by

$$\frac{\|x-x_1\|}{\|x-x_1\| + \|x-x_2\|} = \Delta.$$

Defining $x = (a, b)$, this becomes

$$\frac{\sqrt{a^2+b^2}}{\sqrt{a^2+b^2} + \sqrt{(a-c)^2+b^2}} = \Delta.$$

After squaring both sides and following some simplifications, we get

$$(1-2\Delta)a^2 + (1-2\Delta)b^2 + 2\Delta^2 ac - \Delta^2 c^2 = 0.$$

Dividing both sides by $(1-2\Delta)$ and adding $[\Delta^2 c/(1-2\Delta)]^2$ to both sides yields

$$\left(a + \frac{\Delta^2 c}{1-2\Delta}\right)^2 + b^2 = \left[\frac{\Delta(1-\Delta)}{1-2\Delta} c\right]^2$$

which is a circle with center $(-(\Delta^2 c/(1-2\Delta)), 0)$ and radius $(\Delta(1-\Delta)/(1-2\Delta))c$. Similarly, we can prove S_2 is a circular region with center $(1 + (\Delta^2 c/(1-2\Delta)), 0)$ and radius $(\Delta(1-\Delta)/(1-2\Delta))c$. ■

F. Proof of Theorem 6

There are three cases to consider in terms of the target point location y_1.

i) $y_1 \in S_1$. This means $\delta_{11} < \Delta$ and $\delta_{12} > 1-\Delta$. Thus, from (23)

$$\nabla J = \nabla J_1 = \begin{bmatrix} \frac{\partial J_i}{\partial x_1} \\ \frac{\partial J_1}{\partial x_2} \end{bmatrix} = \begin{bmatrix} R_1 \frac{x_1-y_1}{\|x_1-y_1\|} \\ 0 \end{bmatrix}$$

when $x_1 \neq y_1$. Moreover, $H_1 \succcurlyeq 0$. Since $J \geq 0$ for all (x_1, x_2) and $J(x_1, x_2) = 0$ for $x_1 = y_1$ and $x_2 \neq y_1$, this is the only possible local minimum.

ii) $y_1 \in C_1$. This means $\Delta < \delta_{1j} < 1 - \Delta$ for $j = 1, 2$. In this case, from (26)

$$\nabla J = \nabla J_1 = \begin{bmatrix} \frac{R_1}{(1-2\Delta)} \left[2(1-\delta_{11})^2 - \Delta\right] \frac{x_1 - y_1}{\|x_1 - y_1\|} \\ \frac{R_1}{(1-2\Delta)} \left[2(1-\delta_{12})^2 - \Delta\right] \frac{x_2 - y_1}{\|x_2 - y_1\|} \end{bmatrix}.$$

The solution for $\nabla J = 0$ is $\delta_{11} = \delta_{12} = 1 - \sqrt{\Delta/2}$. Then, from the definition of Δ, we have $\Delta \in [0, 1/2)$, thus $1 - \sqrt{\Delta/2} > 1/2$. Since, from the definition of δ_{ij}, we have $\delta_{11} + \delta_{12} = 1$, we can see that $\delta_{11} = \delta_{12} = 1 - (\sqrt{\Delta/2})$ is not feasible because $\delta_{11} + \delta_{12} = 2(1 - (\sqrt{\Delta/2})) > 1$. Since there is no feasible solution for $\nabla J = 0$, there is no local minimum in this case.

iii) $y_1 \in S_2$. This case is similar to i) and gives $x_2 = y_1$ and $x_1 \neq y_1$ as the only local minimum.

Combining i) and iii) gives the desired result. The second assertion follows from Theorem 3. ∎

REFERENCES

[1] J. Marschak and R. Radner, *Economic Theory of Teams*. New Haven, CT: Yale Univ. Press, 1972.

[2] P. R. Chandler, M. Pachter, and S. Rasmussen, "UAV cooperative control," in *Proc. 2001 Amer. Control Conf.*, 2001, pp. 50–55.

[3] B. T. Clough, "Unmanned aerial vehicles: autonomous control challenges, a researcher's perspective," in *Cooperative Control and Optimization*, R. Murphey and P. M. Pardalos, Eds. Norwell, MA: Kluwer, 2000, pp. 35–53.

[4] J. M. Wohletz, D. A. Castañón, and M. L. Curry, "Closed-loop control for joint air operations," in *Proc. 2001 Amer. Control Conf.*, 2001, pp. 4699–4704.

[5] M. L. Curry, J. M. Wohletz, D. A. Castañón, and C. G. Cassandras, "Modeling and control of a joint air operations environment with imperfect information," in *Proc. SPIE 16th Annu. Int. Symp.*, 2002, pp. 41–51.

[6] D. A. Castañón and J. M. Wohletz, "Model predictive control for dynamic unreliable resource allocation," in *Proc. 41st IEEE Conf. Decision and Control*, 2002, pp. 3574–3579.

[7] J. Finke, K. M. Passino, and A. G. Sparks, "Cooperative control via task load balancing for networked uninhabited autonomous vehicles," in *Proc. 42nd IEEE Conf. Decision and Control*, 2003, pp. 31–36.

[8] J. S. Bellingham, M. Tillerson, M. Alighanbari, and J. P. How, "Cooperative path planning for multiple UAV's in dynamic and uncertain environments," in *Proc. 41st IEEE Conf. Decision and Control*, 2002, pp. 2816–2822.

[9] J. Hu and S. Sastry, "Optimal collision avoidance and formation switching on Riemannian manifolds," in *Proc. 40th IEEE Conf. Decision and Control*, 2001, pp. 1071–1076.

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

[11] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms: combined effects of attractant/repellent profiles," in *Proc. of 41st IEEE Conf. Decision and Control*, 2002, pp. 2848–2853.

[12] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Proc. 41st IEEE Conf. Decision and Control*, 2002, pp. 112–117.

[13] D. Q. Mayne and L. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 35, no. 7, pp. 814–824, Jul. 1990.

[14] J. B. Cruz, M. A. Simaan, A. Gacic, and Y. Liu, "Moving horizon game theoretic approaches for control strategies in a military operation," in *Proc. 40th IEEE Conf. Decision and Control*, 2001, pp. 628–633.

[15] L. Singh and J. Fuller, "Trajectory generation for a UAV in Urban Terrain using nonlinear MPC," in *Proc. 2001 Amer. Control Conf.*, 2001, pp. 2301–2308.

[16] E. Franco, T. Parisini, and M. M. Polycarpou, "Cooperative control of discrete-time agents with delayed information exchange: A receding-horizon approach," in *Proc. 43rd IEEE Conf. Decision and Control*, 2004, pp. 4274–4279.

[17] W. B. Dunbar and R. M. Murray, "Receding horizon control of multi-vehicle formations: A distributed implementation," in *Proc. 43rd IEEE Conf. Decision and Control*, 2004, pp. 1995–2002.

[18] A. Richards and J. How, "Decentralized model predictive control of cooperating UAVs," in *Proc. 43rd IEEE Conf. Decision and Control*, 2004, pp. 4286–4291.

[19] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proc. 43rd IEEE Conf. Decision and Control*, 2004, pp. 3357–3363.

[20] C. G. Cassandras and W. Li, "A receding horizon approach for solving some cooperative control problems," in *Proc. 41st IEEE Conf. Decision and Control*, 2002, pp. 3760–3765.

[21] W. Li and C. G. Cassandras, "Centralized and distributed cooperative receding horizon control of autonomous vehicle missions," *J. Math. Comp. Model*, 2006, to be published.

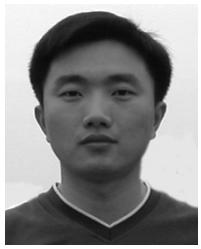
[22] Z. Drezner, Ed., *Facility Location: A Survey of Applications and Methods*. ser. Springer Series in Operations Research. New York: Springer-Verlag, 1995.

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

[24] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, Eds., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. New York: Wiley, 1985.

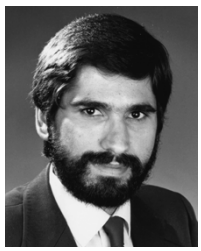
[25] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[26] W. Li and C. G. Cassandras, "Cooperative distributed control of networks with mobile nodes: Theory and practice," in *Proc. 43rd Annu. Allerton Conf. Communication, Control and Computation*, 2005.



Wei Li received the B.E. and M.E. degrees in control theory and control engineering from Tsinghua University, Beijing, China, in 1998 and 2001, respectively. He is currently working toward the Ph.D. degree in the Department of Manufacturing Engineering and the Center for Information and Systems Engineering (CISE), Boston University, Boston, MA.

His research interests include control of discrete-event and hybrid systems, stochastic optimization, distributed computation, and control of real-time systems, with applications to communication networks, sensor networks, manufacturing systems, intelligent vehicle systems, and robotics.



Christos G. Cassandras (S'82–M'82–SM'91–F'96) received the B.S. degree from Yale University, New Haven, CT, the M.S.E.E. degree from Stanford University, Stanford, CA, and the S.M. and Ph.D. degrees from Harvard University, Cambridge, MA, in 1977, 1978, 1979, and 1982, respectively.

From 1982 to 1984, he was with ITP Boston, Inc., Boston, MA, where he worked on the design of automated manufacturing systems. From 1984 to 1996, he was a Faculty Member in the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst. Currently, he is Professor of Manufacturing Engineering and Professor of Electrical and Computer Engineering at Boston University, Boston, MA, and a founding member of the Center for Information and Systems Engineering (CISE). He specializes in the areas of discrete-event and hybrid systems, stochastic optimization, and computer simulation, with applications to computer networks, sensor networks, manufacturing systems, transportation systems, and command-control systems.

Dr. Cassandras has published over 200 papers in these areas, along with two textbooks, one of which was awarded the 1999 Harold Chestnut Prize by the IFAC. He is currently Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and has served on several editorial boards and as Guest Editor for various journals. He is a member of the IEEE Control Systems Society Board of Governors and an IEEE Distinguished Lecturer. He was awarded a 1991 Lilly Fellowship and is also a member of Phi Beta Kappa and Tau Beta Pi.