Bioinsp. Biomim. 1 (2006) 1–11 doi:10.1088/1748-3182/1/1/001

Cooperative optimal control: broadening the reach of bio-inspiration

Cheng Shao[1,3] **and Dimitrios Hristu-Varsakelis**[2]

[1] Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA
[2] Department of Applied Informatics, University of Macedonia, Thessaloniki 54006, Greece

E-mail: cshao@umd.edu and dcv@uom.gr

Received 25 November 2005
Accepted for publication 10 March 2006
Published 6 April 2006
Online at stacks.iop.org/BB/1/1

Abstract
Inspired by the process by which ants gradually optimize their foraging trails, this paper investigates the cooperative solution of a class of free final time, partially constrained final state optimal control problems by a group of dynamical systems. We propose an iterative, pursuit-based algorithm which generalizes previously proposed models and converges to an optimal solution by iteratively optimizing an initial feasible trajectory/control pair. The proposed algorithm requires only short-range, limited interactions between group members, avoids the need for a 'global map' of the environment in which the group evolves, and solves an optimal control problem in 'small' pieces, in a manner which will be made precise. The performance of the algorithm is illustrated in a series of simulations and laboratory experiments.

1. Introduction

In recent years, problems in cooperative control and optimization have been increasingly capturing the attention of researchers, fueled both by a rising interest in understanding what can be accomplished by cooperation *vis-à-vis* decision and control as well as by advances in technology that make it possible to experiment with 'collectives' of robots, unmanned air vehicles (UAVs) and other electromechanical systems. Central to that program of research is the idea that cooperation can transform a group into 'more than the sum of its parts', enabling its members to collectively perform tasks that are beyond their abilities as individuals, and reaping various other benefits with regard to cost, robustness and performance.

The difficulty in constructing effective cooperative control laws is often matched by the elegance of the results when those laws are put into action. This becomes most apparent when observing natural systems [1]. For example, schools of fish can move in tight formations and respond to predators almost as quickly as a single organism; worker honey bees can share information by 'dancing' and distribute themselves among nectar sources in accordance with the profitability of each source; ants can utilize pheromone secretions for recruiting nest-mates and for optimizing their foraging trails [2].

Observations of these and other biological groups, combined with mathematical models that aim to link individual with group behavior, indicate that local interactions are apparently sufficient to produce an impressive array of complex behaviors, and that minor changes in the rules that govern individual decision-making can result in qualitative changes in group behavior. From a system's viewpoint this seems rather fortunate, and has given rise to efforts [2–4] aimed at describing or explaining the movement of animal groups, while seeding a variety of research in topics ranging from 'collective' motion control [5–7] to distributed covering and searching [8, 6], remote exploration and information gathering by swarms of small autonomous robots [9], cooperative estimation [10, 11], robotic teams [12–14] and biologically inspired optimization [15, 16].

1.1. Ant colonies and bio-inspired optimal control

This work is a continuation of recent efforts [17–19] to formulate cooperative optimal control strategies based on simple models of ant movement. Ants have made excellent test subjects for mathematical models that link local to collective behavior [20] and have inspired a significant amount of work

[3] Present address: SAC Capital Advisor, LLC, 540 Madison Avenue, New York, NY 10022, USA.

Figure 1. A trail formed by marching ants. In this case, the ants appear to follow the terrain in a way that minimizes the trail length. Courtesy of Nick Lyon, copyright `cockroach.org.uk`, 2005. Used with permission.

on control and optimization [3, 16], including the development of cooperative optimal control laws [17, 18], robust adaptive algorithms for optimal traversal of unknown regions [8] and distributed search methodologies for the traveling salesman problem [15].

The foraging activity of ant colonies offers a particularly interesting example of cooperation in animal aggregates. Ants are known to recruit their coworkers to convey newly discovered food back to the nest. Finding efficient (short) paths between the nest and a food source appears to be too complex for individuals to accomplish, given their limited cognition and small size relative to obstacles in their environment (see figure 1). Nonetheless, ant colonies exhibit a high degree of competence in such tasks.

Attempts to capture the organizing principle by which ants optimize their trails have included [2], which advanced a model based on the use of pheromonal secretions that ants lay down to recruit nest-mates and to indicate the frequency of use for a particular path. Earlier work [3] introduced a particularly simple—but elegant—ant colony organizing rule which will be central to our approach; specifically, it was shown that if the ants 'pursue' one another on $\mathbb{R}^2$ (each pointing its velocity vector toward its predecessor) then they produce progressively 'straighter' trails. This idea was initially used to solve cooperative path optimization problems involving simple kinematic vehicles in non-Euclidean environments [17]. Subsequent work [18] showed that the earlier results could also apply to collectives whose members had non-trivial dynamics, and proposed an algorithm termed 'local pursuit' (using the term coined in [3]) which guided members of a group of autonomous systems toward the solution of an optimal control problem, starting from an initial feasible trajectory.

The contribution of this paper is to propose a general bio-inspired pursuit strategy of which the algorithms in [3, 17, 18] are special cases, and which addresses the larger and more useful class of optimal control problems with free final times and partially constrained final states. Our work shows that the optimizing effects of pursuit-based cooperation apply in a far broader setting than initially conceived, once the notion of pursuit is properly defined, and includes the first (to the authors' knowledge) laboratory demonstration of pursuit-based optimal control using systems with non-trivial dynamics. For engineering systems, the strategy described here could be a useful alternative in the cases where an optimal control problem simply cannot be solved by an individual system due to lack of information about its surroundings and the global geometry. In particular, we will consider groups of dynamical systems whose members are not assumed to know the geometry of their environment beyond their (limited) sensor range; members do not necessarily use the common coordinate system for their environment, so they may not be able to communicate meaningful instructions to one another; they may not even know the coordinates of their destination, but instead may rely on a set of directions (e.g., an open-loop control input, or landmark-based instructions) to get there. Under such severe restrictions, solving the optimal control problem under consideration is difficult, if not impossible, for an individual. Remote exploration (e.g., planetary surfaces) by teams of robotic vehicles is one example of the setting where those constraints would likely apply (e.g., small vehicles with limited sensing range, operating on unknown terrain with limited computing power). The algorithms presented here could also be useful as a way to enhance the power of existing numerical optimization techniques; we will have a little more to say about this in section 4.1.

As we shall see, pursuit-based cooperation can be effective in overcoming the difficulties mentioned above, especially in the case of trajectory optimization problems which—because of the members' computational or sensing limitations—are easier to solve when boundary conditions are 'close' to one another, with the term 'close' taken to include not only geographical separation but also distance on the manifold on which copies of a dynamical system evolve. In addition, the algorithm discussed here is interesting as a mathematical model for some of the complex patterns that are observed in the movement of ant colonies. Although we do not claim that the cooperation strategies proposed here are in fact used by ants during trail formation, we find it noteworthy that, phenomenologically at least, ant-trail optimization can be described by a few simple instructions that are effective in a rich variety of environments (geometry) and vehicles (equations of motion).

Compared to the existing literature, this work differs in two important ways. First, the algorithms proposed here generalize those in [18] and prior works, by including a new, second 'stage', which requires each team member to eventually 'ignore' its predecessor and attempt to finish the optimization problem on its own. Second, the two-stage algorithm makes it possible to address a class of problems that is significantly broader over what was possible up to now. To the authors' knowledge, the pursuit-based optimization strategies presented to date have been restricted to problems where a dynamical system is to be steered (optimally) to a *fixed* final state. The proposed approach can address problems where the final state, and the final time, are unknown or partly known and are to be optimized over. The latter category includes important applications which were outside the scope of [3, 17, 18], such as the following: (i) satellite orbit transfer problems, where the goal is to attain a given altitude, but the precise longitude and latitude where the desired orbit is achieved are unimportant; (ii) optimal (e.g., minimum-fuel) intercept problems where the coordinates at which the target will be intercepted are not known *a priori*; (iii) remote exploration by teams of autonomous vehicles, e.g., 'find the shortest path from 'here' to the lake due East', where the 'target' is expansive and the final state on that target is something to be optimized jointly with the rest of the vehicles' trajectories. This enlargement of the domain of applicability and the use of our two-stage pursuit algorithm pose some technical difficulties which were not present in the previous works on pursuit-based cooperative control and which will be explained and dealt with in what follows.

The remainder of this paper is organized as follows. In the next section, we describe the optimal control problem to be addressed. Section 3 describes a bio-inspired iterative algorithm that is well suited to groups of cooperating dynamical systems operating with limited information. Section 4 contains the main results regarding the group's trajectories when its members evolve under the proposed strategy. Section 5 presents a series of simulations and laboratory experiments that illustrate our approach.

2. Problem statement and notation

We are interested in the solution of optimal control problems using a group of cooperating agents, where the term 'agent' refers to a copy of the control system,

$$\dot{x}_k = f(x_k, u_k), \quad x_k(t) \in \mathbb{R}^n, \quad u_k(t) \in \Omega \subseteq \mathbb{R}^m, \quad x_k(0) = x_S \text{ given} \tag{1}$$

for $k = 0, 1, 2 \ldots$, where x_k is the state of the kth agent. Here, u_k is an exogenous input via which we can affect the evolution of the corresponding agent's state, and Ω is a set that reflects any feasibility constraints that the control u_k may be subject to. Physically, each instance of (1) could stand for a robot, unmanned air vehicle (UAV) or other autonomous system.

Let $x(t)$ denote the state of a particular agent of (1). The problem under consideration is as follows:

Problem 1. *Find a control-trajectory pair $(u^*(t), x^*(t))$ of (1), a final time $\Gamma^* > 0$ and a final state $x^*(\Gamma^*)$ that minimize*

$$J(x, \dot{x}, t_0) = \int_{t_0}^{t_0+\Gamma} g(x, \dot{x}, u)\,\mathrm{d}t + G(x(t_0+\Gamma)), \tag{2}$$

subject to $x(t_0) = x_S$ and $Q(x(t_0+\Gamma)) = 0$,

where it is assumed that $g(x(t), \dot{x}(t), u) \geqslant 0$, $G(x(t_0+\Gamma)) \geqslant 0$ and that $Q(\cdot)$ is an algebraic function of the state. As formulated, problem 1 captures a broad class of optimal control problems, including those with free or fixed final time, and free or fixed final state. Examples include optimal trajectory tracking, and minimum-time or minimum-fuel intercept problems, to name a few (see, for example, [21, 22]). The special case $g(x, \dot{x}, u) = \|\dot{x}\|$ corresponds to the trail-length minimization problem. In the following, we will take $g(\cdot,\cdot,\cdot)$ to be continuously differentiable and will assume that the problem data (1), (2) are such that an optimal solution exists. For a fuller discussion regarding the existence of solutions to problem 1 we refer the reader to [23].

Problem 1 is a difficult optimal control problem which, in some instances, cannot be solved by an individual system. For example, in the case of a team of autonomous vehicles exploring unknown terrain (to make an analogy with foraging ants), finding an optimal trajectory to a distant location of interest might be beyond the capabilities of an individual vehicle; the vehicle would need to have a sufficiently long sensing range as well as a 'global' map of the terrain in order to proceed with the necessary computations. On the other hand, it is more likely that the available vehicles will be able to sense and communicate locally, i.e., within a small area around them, and may lack knowledge of the terrain. In addition, finding optimal trajectories may require considerably more computation when boundary conditions are 'far' apart. We will elaborate on some of these issues in section 4.1.

We would like to know whether a pursuit-like strategy can be used effectively by a group of cooperating dynamical systems to solve instances of problem 1 where final times and states are unconstrained, and if so, what is the appropriate mathematical interpretation of the word 'pursuit' in that setting. In the remainder of this paper we will make use of the following notation.

Definition 1. *Given the final state constraint $Q(x) = 0$, the constraint set of x is*

$$S_Q \triangleq \{x \mid Q(x) = 0\}.$$

We will assume that $\partial Q/\partial x$ has constant rank in the neighborhood of the set S_Q. For simplicity, we take Q to be scalar, so that S_Q defines an $(n-1)$-dimensional manifold in $\mathbb{R}^n$. This assumption can easily be lifted and will not change the results that follow.

The function $G(x)$ in (2) will be taken to be of the form

$$G(x) = \begin{cases} F(x) & \text{if } x \in S_Q \\ 0 & \text{if } x \notin S_Q, \end{cases}$$

with $F(x) \geqslant 0, \forall x \in S_Q$. Problem 1 involves optimal control with free final time and partially constrained final state. Fixed final state problems, where S_Q is a single point [18], are special cases of what are considered here.

For any pair of fixed states $a, b \in \mathbb{D} \subset \mathbb{R}^n$, let $x^*(t)$ denote the optimal trajectory of (1) from a to b with free final time (minimizing J with respect to x and Γ only). We will write $\Gamma^*(a, b)$ for the corresponding optimal final time. The cost of following x^* will be denoted by

$$\eta(a, b, t_0) \triangleq \int_{t_0}^{t_0+\Gamma^*} g(x^*, \dot{x}^*)\,\mathrm{d}t + G(x^*(t_0+\Gamma^*)) = \min_{u,\Gamma} J(x, \dot{x}, t_0), \tag{3}$$

where the minimum is taken subject to (1) with $x(t_0) = a$, $x(t_0+\Gamma) = b$.

Now, let $x^*(t)$ be the optimal trajectory from an initial state a to the constraint set S_Q, and let $\Gamma^*_Q(a, S_Q)$ be the corresponding optimal final time from a to S_Q. The cost of following x^* will be denoted by

$$\eta_Q(a, t_0) \triangleq \int_{t_0}^{t_0+\Gamma_Q^*} g(x^*, \dot{x}^*)\,dt + G(x^*(t_0 + \Gamma_Q^*)) = \min_{u,\Gamma_Q} J(x, \dot{x}, t_0) \quad (4)$$

subject to $x(t_0) = a$, $Q(x(t_0 + \Gamma_Q)) = 0$.

The cost of following a generic trajectory $x(t)$ of (1) during $[t_0, t_0 + \sigma)$ will be denoted by

$$C(x, t_0, \sigma) \triangleq \int_{t_0}^{t_0+\sigma} g(x, \dot{x})\,dt + G(x(t_0 + \sigma)). \quad (5)$$

The following facts can be derived easily from the properties of optimal trajectories and will be helpful in the development that follows:

Fact 1. *Let η, η_Q and C be as defined in (3)–(5), and let $x_k(t)$ be a trajectory of (1). Then, the following holds:*

(i) $\eta(a, b, t_0) \leqslant C(x_k, t_0, \Gamma)$ *for any* $x_k(\cdot)$ *such that* $x_k(t_0) = a$, $x_k(t_0 + \Gamma) = b$.
(ii) $\eta(a, c, t_0) \leqslant \eta(a, b, t_0) + \eta(b, c, t_0 + \sigma)$ *with* $\sigma = \Gamma^*(a, b)$.
(iii) $\eta_Q(a, t_0) \leqslant \eta(a, b, t_0)$ *for any* $b \in S_Q$.

3. A bio-inspired algorithm for optimal control

For the group of agents (1), we will assume that there is an initial feasible (but suboptimal) control/trajectory pair $(u_0(t), x_0(t))$ available, obtained through a combination of *a priori* knowledge about the problem and/or random exploration. The optimization strategy to be considered shortly can be described intuitively as follows. If one agent is able to follow a feasible trajectory to the target set, a second agent can reach the target set more efficiently (i.e., with a lower cost, as defined by (2)) by always moving along a trajectory that is optimal from its current state to that of its predecessor. Following the idea in [3, 18], the agents x_k leave the initial state x_S sequentially and pursue one another toward the set S_Q, in a way which will be made precise shortly and which mimics a line of marching ants. The sequence is initiated with the first agent following the trajectory x_0 to reach a point in S_Q. Each subsequent agent attempts to intercept its predecessor—along optimal trajectories defined by (3)—as long as the predecessor has not reached the constraint set S_Q. If the predecessor has already reached S_Q, then the pursuer is to ignore the preceding agent and instead evolve along the optimal trajectory defined by (4). We will use the term *generalized continuous local pursuit* (GCLP) to describe the strategy outlined above, in order to distinguish it from the special case (CLP) in [18]. The precise rules that govern the movement of each agent are as follows:

Algorithm 1 (generalized continuous local pursuit)**.** *Let x_S be a starting state in $\mathbb{D}$ and let S_Q be the constraint set in which the final state must lie. Let $x_0(t)(t \in [0, T_0])$ be an initial trajectory satisfying (1) with $x_0(0) = x_S$, $Q(x_0(T_0)) = 0$. Choose $0 < \Delta \leqslant T_0$.*

(i) *For $k = 1, 2, 3 \ldots,$*
let $t_k = k\Delta$ be the starting time of the kth *agent*[4] *so that $x_k(t_k) = x_S$. Let $t_k + T_k$ be the time of arrival of the kth agent at S_Q.*
(ii) *For all $t \in [t_k, t_k + T_k]$,*
let $u_t^*(\tau)$ be a control input that achieves

$$\begin{cases} \eta(x_k(t), x_{k-1}(t), t) & \text{if} \quad x_{k-1}(t) \notin S_Q \\ \eta_Q(x_k(t), t) & \text{if} \quad x_{k-1}(t) \in S_Q \end{cases}$$

for the system $\dot{x}_t(\tau) = f(x_t(\tau), u_t^*(\tau))$, where

$$\tau \in \begin{cases} [t, t + \Gamma^*(x_k(t), x_{k-1}(t))] \\ \quad \text{if} \quad x_{k-1}(t) \notin S_Q \\ [t, t + \Gamma_Q^*(x_k(t), S_Q)] \\ \quad \text{if} \quad x_{k-1}(t) \in S_Q. \end{cases}$$

(iii) *Apply $u_k(t) \triangleq u_t^*(0)$ to the k^{th} agent.*
(iv) *Repeat from step (ii), until the* kth *agent reaches S_Q.*
(v) $k \leftarrow k + 1$.

We will refer to Δ as the *pursuit interval*. When discussing pairs of agents during pursuit, the $(k-1)$th agent will be designated as the *leader* and the kth agent as the *follower*. As step 2 of the algorithm indicates, there are two types of follower movement, which could be termed loosely as 'catching up' and 'free running', depending on whether the leader has reached the final constraint set S_Q. The former lets agents 'learn' from their leaders, while the 'free running' stage enables them to find the optimal final state within S_Q once they are close enough to that set. Both stages will be essential in order for the group to solve problem 1.

Note that the GCLP algorithm requires each follower to continuously re-plan its trajectory (by sensing $x_k(t)$ and computing the 1-parameter family of control inputs $u_t^*(\tau)$) to catch up with its leader during the pursuit process. Continuous pursuit may imply a significant computational burden for each agent, especially in the cases where the optimal trajectories 'linking' the follower and leader cannot be written in the closed form. For instances of problem 1 where, for each follower, the optimal time to reach the leader is lower bounded for all times (as is the case for the examples in section 5), it is possible to alter the previous algorithm so that each agent performs only a *finite* number of measurements and trajectory updates as it evolves from x_S to S_Q. This can be accomplished by defining a sampled version of GCLP, termed *generalized sampled local pursuit* (GSLP), of which the algorithm in [18, 19] is a special case:

Algorithm 2 (generalized sampled local pursuit)**.** *Let x_S be a starting state in $\mathbb{D}$ and let S_Q be the constraint set in which the final state must lie. Let $x_0(t)$, $t \in [0, T_0]$ be an initial trajectory satisfying (1) with $x_0(0) = x_S$, $Q(x_0(T_0)) = 0$. Choose the pursuit interval Δ such that $0 < \Delta \leqslant T_0$.*

(i) *For $k = 1, 2, 3 \ldots,$*
let $t_k = k\Delta$ be the starting time of the kth *agent, i.e., $x_k(t_k) = x_S$.*

[4] One simple way to ensure that all agents are at the initial state when it is time to begin their pursuit is to require that the pair $(u = 0, x = x_S)$ be an equilibrium of (1).

(ii) *For $i = 0, 1, 2, 3, \ldots,$ define $t_k^i = t_k^{i-1} + \delta_i, t_k^0 = t_k$ to be the times when the kth agent will update its trajectory, where $0 < \delta_i < \min(\Delta, \Gamma_{i-1}^*)$, and Γ_{i-1}^* is the optimal final time (defined by (3) or (4)) of the trajectory which x_k planned to follow at t_k^{i-1} $\left(\Gamma_{-1}^* \triangleq \Delta\right)$. Let $u_{t_k^i}^*(\tau)$ be a control policy that achieves*

$$\begin{cases} \eta\left(x_k\left(t_k^i\right), x_{k-1}\left(t_k^i\right), t_k^i\right) & \text{if} \quad x_{k-1}\left(t_k^i\right) \notin S_Q \\ \eta_Q\left(x_k\left(t_k^i\right), t_k^i\right) & \text{if} \quad x_{k-1}\left(t_k^i\right) \in S_Q \end{cases}$$

subject to (1), where $\tau \in$

$$\begin{cases} \left[t_k^i, t_k^i + \Gamma^*\left(x_k\left(t_k^i\right), x_{k-1}\left(t_k^i\right)\right)\right] \\ \quad \text{if} \quad x_{k-1}\left(t_k^i\right) \notin S_Q \\ \left[t_k^i, t_k^i + \Gamma_Q^*\left(x_k\left(t_k^i\right), S_Q\right)\right] \\ \quad \text{if} \quad x_{k-1}\left(t_k^i\right) \in S_Q. \end{cases}$$

(iii) *Apply $u_k(t) = u_{t_k^i}^*(t - t_k^i)$ to the* kth *agent during $t \in \left[t_k^i, t_k^{i+1}\right)$.*

(iv) *Repeat from step (ii) until the* kth *agent reaches S_Q.*

(v) $k \leftarrow k + 1$.

According to GSLP, each agent x_k periodically samples the state $x_{k-1}\left(t_k^i\right)$ of its predecessor and computes the control $u_{t_k^i}^*(\tau)$ that will allow it to reach that state (or the target set S_Q) with minimum cost. The pursuing agent then applies that control until the sampling time.

We will refer to the δ_i as the *updating intervals*. To simplify the discussion, we assume that the Γ_i^* are lower bounded, so that we may choose δ to be a constant. Under GSLP, each agent executes a finite number of updates of its trajectory, once every $\delta < \Delta$ time units. The reduced computational demands of GSLP make it attractive in the cases where the complexity of the agents' dynamics (as well as that of the environment they evolve in) necessitate the use of numerical methods for finding optimal trajectories. We will have more to say about this in section 4.1

As defined above, the GCLP and GSLP algorithms assume that followers do not intercept their leaders. If an interception does occur, one can simply prescribe that the follower 'join' its leader by reproducing the leader's trajectory after the time of interception. Because the initial agent travels along its trajectory for T_0 units of time and the pursuit interval Δ is finite, there will be a finite number of such events, whose existence will not affect the results discussed below.

4. Main results

In this section, we explore the behavior of the group (1) under GCLP. Similar results can be derived for GSLP, along the lines of the discussion below. We will begin by considering the sequence of trajectories $\{x_k(t)\}$ produced by GCLP. We will first investigate the convergence of the corresponding cost sequence, and then that of the trajectories themselves. On a technical level, the fact that the time to reach the target set S_Q and the terminal point within that set are to be optimized (in contrast to [18] for example) has some non-trivial implications

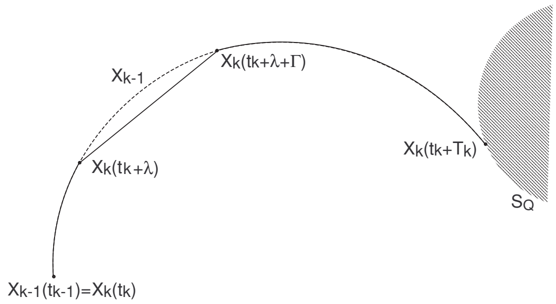

Figure 2. Illustration of the trajectory obtained by a single update when $\lambda < T_{k-1} - \Delta$.

for establishing the desired results. In particular, when taking into account the pursuit algorithm's second stage, we will have to show that each agent improves over its predecessor overall even if it abandons the pursuit after some time (as our algorithm requires) and heads to the target set on its own. Because pursuer and leader are at different locations when they 'enter' the algorithm's second stage, and finish their trajectories at different points on the target area it is not obvious that the pursuer's path should be closer to optimal than the leader's.

In the discussion that follows, we will often cast matters in terms of trajectories (optimal or not) of (1), instead of the control inputs that produce them. Of course, it is the optimal input that we are ultimately seeking; the reference to trajectories is made only for the sake of convenience, as each of them is uniquely determined (after choosing an initial state $x(0)$) by a corresponding input $u(t)$.

Lemma 1. *Consider a leader–follower pair evolving under GCLP with pursuit interval Δ. Let the leader's trajectory be $x_{k-1}(t)$ ($t \in [t_{k-1}, t_{k-1}+T_{k-1}]$), where T_{k-1} is the leader's time of arrival at the final set S_Q, and fix $\lambda \in [0, T_{k-1})$. Suppose the follower updates its trajectory only once during $[t_k, t_k + T_k]$, as described next:*

- *If $\lambda < T_{k-1} - \Delta$, the follower moves along the optimal trajectory (in the sense of (3)) joining $x_k(t_k + \lambda)$ and $x_{k-1}(t_k + \lambda)$, with optimal final time $\Gamma = \Gamma^*(x_k(t_k + \lambda), x_{k-1}(t_k + \lambda))$. During other times, the follower replicates the leader's trajectory, i.e.,*

$$x_k(t) = \begin{cases} x_{k-1}(t - \Delta) & t \in [t_k, t_k + \lambda] \\ x_{k-1}(t - \Gamma) & t \in [t_k + \lambda + \Gamma, t_k + T_k]. \end{cases}$$

- *If $\lambda \geqslant T_{k-1} - \Delta$, the follower chooses to evolve along the optimal trajectory (in the sense of (4)) from $x_k(t_k + \lambda)$ to the constraint set S_Q. During other times*

$$x_k(t) = x_{k-1}(t - \Delta) \qquad t \in [t_k, t_k + \lambda].$$

Then the cost along the follower's trajectory will be no greater than that along the trajectory of its leader.

Proof. First, consider the case $\lambda < T_{k-1} - \Delta$. For $t \in [t_k + \lambda, t_k + \lambda + \Gamma]$, the follower moves on the locally optimal trajectory $x_k(t)$ (see figure 2). The cost along x_k

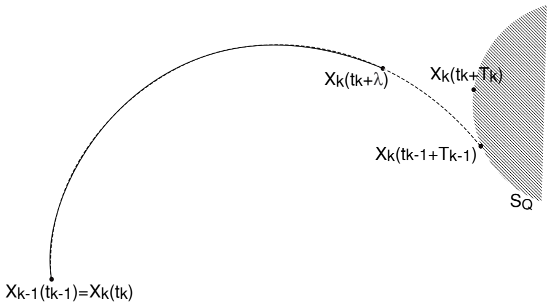

Figure 3. Illustration of the trajectory obtained by a single update when $\lambda \geqslant T_{k-1} - \Delta$.

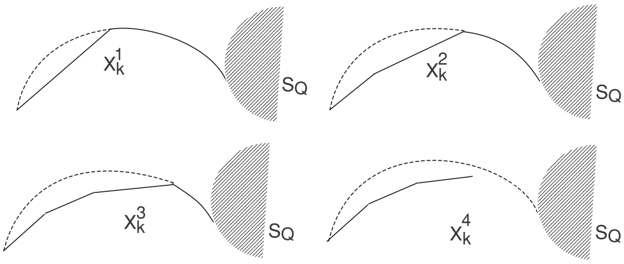

Figure 4. Illustration of the trajectory sequence $x_k^i(t)$. Each trajectory is obtained by a single update upon its predecessor.

satisfies

$$\begin{aligned} C(x_k, t_k, T_k) &= C(x_k, t_k, \lambda) + C(x_k, t_k + \lambda + \Gamma, T_k - \lambda - \Gamma) \\ &\quad + \eta(x_k(t_k + \lambda), x_{k-1}(t_k + \lambda), t_k + \lambda) \\ &\leqslant C(x_{k-1}, t_{k-1}, \lambda) + C(x_{k-1}, t_{k-1} + \lambda, \Delta) \\ &\quad + C(x_{k-1}, t_{k-1} + \lambda + \Delta, T_{k-1} - \lambda - \Delta) \\ &= C(x_{k-1}, t_{k-1}, T_{k-1}), \end{aligned} \tag{6}$$

where $\Gamma = \Gamma^*(x_k(t_k + \lambda), x_{k-1}(t_k + \lambda))$.

If $\lambda \geqslant T_{k-1} - \Delta$ (see figure 3), the cost along x_k is

$$\begin{aligned} C(x_k, t_k, T_k) &= C(x_k, t_k, \lambda) + \eta_Q(x_k(t_k + \lambda), t_k + \lambda) \\ &\leqslant C(x_{k-1}, t_{k-1}, \lambda) + C(x_{k-1}, t_{k-1} + \lambda, T_{k-1} - \lambda) \\ &= C(x_{k-1}, t_{k-1}, T_{k-1}). \end{aligned}$$

Therefore the cost along the follower's trajectory is no greater than the leader's. □

Now, the cost of the iterative trajectories can be shown to converge under GCLP:

Lemma 2 (convergence of cost)**.** *If the agents (1) evolve under GCLP, the cost of the iterated trajectories converges.*

Proof. Let C_{k-1} be the cost along the leader's trajectory $x_{k-1}(t)$ $(t \in [t_{k-1}, t_{k-1} + T_{k-1}])$. Define a trajectory sequence $x_k^i(t)$ $\left(t \in \left[t_k, t_k + T_k^i\right]\right)$, $i = 0, 1, 2 \ldots$, whose corresponding costs and final times are C_k^i and T_k^i, respectively, as follows. Let $x_k^0(t) = x_{k-1}(t)$ (the trajectory of a 'leader') and let $x_k^i (i > 0)$ be the trajectory of an agent that pursues x_k^{i-1} by performing only a *single trajectory update*, as described in lemma 1, with $\lambda = (i-1)\delta, \delta > 0$ (see figure 4).

From lemma 1, the cost of each follower's trajectory will be no greater than the leader's. Also, the sequence C_k^i is bounded below for any fixed k. Thus, $C_k^i \leqslant C_k^{i-1}$ and $\lim_{i\to\infty} C_k^i = C_k^\infty$ exists for each k. Consequently,

$$C_k^\infty \leqslant C_k^0 = C_{k-1}.$$

Now, take $\delta = T_{k-1}/i$, so that $\delta \to 0$ as $i \to \infty$. At the limit, the trajectory $x_k^\infty(t)$ is precisely what would be obtained by an agent that pursues its leader x_{k-1}, using GCLP. Hence, the follower's cost is $C_k = C_k^\infty \leqslant C_{k-1}$. Because the sequence $\{C_k\}$ is non-increasing and bounded below (there exists a minimum for (2)), it has a limit. □

To proceed to the main theorem, we will require that the optimal cost of (2) change 'little' for small changes to the endpoints of a trajectory:

Condition 1. $\forall\, a, b_1, b_2 \in \mathbb{D}, \Omega > 0$ *and for any* $x_1(t)$, *with* $x_1(0) = a, x_1(T) = b_1, \exists\, \varepsilon > 0$ *and* $x_2(t)$, *a trajectory of (1) with* $x_2(0) = a, x_2(T) = b_2$, *such that the costs of* x_1 *and* x_2 *satisfy*

$$\|b_1 - b_2\|_\infty < \varepsilon \Rightarrow \|C(x_1, 0, T) - C(x_2, 0, T)\|_\infty < \mathcal{L}\Omega$$

for some constant $\mathcal{L}$, *independent of* Ω.

Under condition 1, the next lemma tells us that optimal trajectories of (1) that 'overlap' (to be made precise below) are locally optimal:

Lemma 3. *Let* $x^*(t)$ *be a trajectory of (1) such that*

(i) $x^*(t)$ $(t \in [0, t_1 + \Delta_1])$ *is optimal (in the sense of (3)) from* $x^*(0)$ *to* $x^*(t_1 + \Delta_1)$, *and*

(ii) $x^*(t)$ $(t \in [t_1, T^*])$ *is optimal (in the sense of (4)) from* $x^*(t_1)$ *to the constraint set* S_Q. *Assume also that condition 1 is satisfied, and* $0 < t_1 < t_1 + \Delta_1 < T^*$. *Then,* $x^*(t)$ $(t \in [0, T^*])$ *is a local minimum of (4) from* $x^*(0)$ *to* S_Q.

Proof. Choose $0 < \sigma \leqslant \Delta_1$. From the principle of optimality, $x^*(t)$ $(t \in [0, t_1 + \sigma])$ and $x^*(t)$ $(t \in [t_1, T^*])$ are each locally optimal with respect to their corresponding endpoints. Suppose that $\|x^*(t_1 + \sigma) - s\|_\infty \geqslant \varepsilon_1$ for any $s \in S_Q$ and that $x^*(t)$ $(t \in [0, T^*])$ is not a local minimum. Then, there must exist $\epsilon < \min(\varepsilon, \varepsilon_1/2)$ (with ε as defined in condition 1) and another optimum $x(t) \in \mathbb{D} \times [0, T]$ satisfying $\|x(t) - x^*(t)\|_\infty < \epsilon$ and $C(x(t), 0, T) < C(x^*(t), 0, T^*)$. Note that $\|x(t_1 + \sigma) - s\|_\infty \geqslant \epsilon$ for any $s \in S_Q$. Construct two trajectories $y_1(t), y_2(t)$ $(t \in [t_1, t_1 + \sigma])$ that connect $x(t)$ and $x^*(t)$ (see figure 5) and satisfy condition 1 (with x^* or x playing the role of x_1, and y_1 or y_2 standing for x_2). In particular, let y_1, y_2 be such that $x^*(t_1) = y_2(t_1), x^*(t_1 + \sigma) = y_1(t_1 + \sigma), x(t_1) = y_1(t_1), x(t_1 + \sigma) = y_2(t_1 + \sigma)$. Now, condition 1 implies that

$$\begin{aligned} C(y_1(t), t_1, \sigma) &< C(x(t), t_1, \sigma) + \mathcal{L}\sigma, \\ C(y_2(t), t_1, \sigma) &< C(x^*(t), t_1, \sigma) + \mathcal{L}\sigma. \end{aligned} \tag{7}$$

Because $x^*(t)$ $(t \in [0, t_1 + \sigma])$ and $x^*(t)$ $(t \in [t_1, T^*])$ are each locally optimal, the following holds,

$$\begin{aligned} C(x^*(t), 0, t_1) + C(x^*(t), t_1, \sigma) &< C(x(t), 0, t_1) \\ &\quad + C(y_1(t), t_1, \sigma), \end{aligned} \tag{8}$$

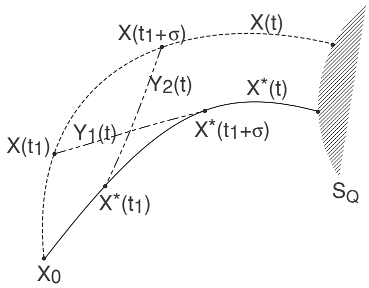

Figure 5. Illustration of the proof of lemma 3: 'overlapping' optimal trajectories form a locally optimal trajectory.

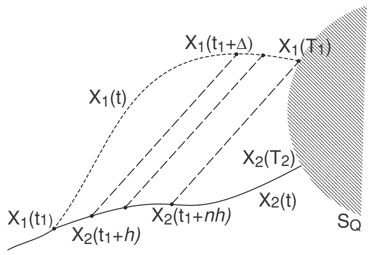

Figure 6. Illustration of the proof of lemma 4: pursuit between agents moving on two supposed 'limiting' equal-cost trajectories leads to the conclusion that the cost along the follower's trajectory is less than that along the leader's.

and

$$C(x^*(t), t_1, \sigma) + C(x^*(t), t_1 + \sigma, T^* - t_1 - \sigma) < C(x(t), t_1 + \sigma, T - t_1 - \sigma) + C(y_2(t), t_1, \sigma). \quad (9)$$

Combining (7) with (8), (9) leads to

$$C(x^*(t), 0, T) < C(x(t), 0, T) + 2\mathcal{L}\sigma. \quad (10)$$

We had assumed that $C(x(t), 0, T) < C(x^*(t), 0, T)$; but if σ is chosen so that

$$0 < \sigma < \frac{C(x^*(t), 0, T) - C(x(t), 0, T)}{2\mathcal{L}},$$

then (10) cannot hold. This is a contradiction, because σ could be chosen arbitrarily small. It follows that $x^*(t)$ $(t \in [0, T^*])$ must indeed be a local minimum. □

Assume now that the locally optimal trajectory from the follower to the leader (or to S_Q) is unique at all times. This assumption is generally satisfied if pursuit is restricted to take place within a 'small' region (setting Δ small), i.e., agents follow 'close' to one another. Then, convergence of the trajectories' cost also implies convergence of the trajectories themselves:

Lemma 4. *If at all times during GCLP, the locally optimal trajectory from the follower to leader (or to S_Q) is unique, then GCLP converges to a limiting trajectory $x_\infty(t)$.*

Proof. Suppose that the trajectory costs converge but that there exists more than one limiting trajectory. Let $x_1(t)$ $(t \in [0, T_1])$ and $x_2(t)$ $(t \in [0, T_2])$ be two such possibilities. Let $t_1 \in [0, T_1]$ be the earliest time that $x_1(t)$ differs from $x_2(t)$. From lemma 2, x_1 and x_2 must have the same cost, otherwise convergence of the cost is contradicted. Suppose that a leader $x_{k-1}(t)$ travels along $x_1(t)$, while a follower $x_k(t)$ travels along $x_2(t)$. Choose $h > 0$ small, and suppose that the follower is to perform a series of discrete updates to its trajectory, at $t_1 + ih (i = 1, 2 \ldots, n = (T_1 - t_1 - \Delta)/h)$, as figure 6 indicates.

At t_1 (the follower's first measurement of x_k), the follower continues to evolve along $x_2(t), t \in [t_1, t_1 + h)$. This means that the trajectory comprises the following: (i) $x_2(t), t \in [t_1, t_1 + h)$ and (ii) the optimal trajectory from $x_2(t_1 + h)$ to $x_1(t_1 + \Delta)$ (as indicated by the left dashed line in figure 6) either has a lower cost than $x_1(t), t \in [t_1, t_1 + \Delta)$, or it has the same cost as $x_1(t), t \in [t_1, t_1 + \Delta)$. The latter possibility contradicts the assumption that the locally optimal trajectory from the follower to leader is unique. Therefore, the locally optimal trajectory by which the follower at t_1 plans to reach the leader has a cost of $C(x_2, t_1, h) + \eta(x_2(t_1 + h), x_1(t_1 + \Delta), t_1 + h)$, and

$$C(x_2, t_1, h) + \eta(x_2(t_1 + h), x_1(t_1 + \Delta), t_1 + h) < C(x_1, t_1, \Delta). \quad (11)$$

Similarly, consider the follower's trajectory updates at $t_1 = ih, i = 2, 3, \ldots, n$ to obtain

$$C(x_2, t_1 + h, h) + \eta(x_2(t_1 + 2h), x_1(t_1 + \Delta + h), t_1 + 2h) < \eta(x_2(t_1 + h), x_1(t_1 + \Delta), t_1 + h) + C(x_1, t_1 + \Delta, h), \quad (12)$$

$$\vdots$$

$$C(x_2, t_1 + (n-1)h) + \eta(x_2(t_1 + nh), x_1(T_1), t_1 + nh) < \eta(x_2(t_1 + (n-1)h), x_1(T_1 - h), t_1 + (n-1)h) + C(x_1, T_1 - h, h). \quad (13)$$

Finally, at the last update the follower will choose to move along the locally optimal trajectory from its current state to S_Q:

$$C(x_2, t_1 + nh, T_2 - t_1 - nh) < \eta(x_2(t_1 + nh), x_1(T_1), t_1 + nh). \quad (14)$$

Note that the (strict) inequalities hold for arbitrarily small $h > 0$. Thus, from (11)–(14) we have that

$$C(x_2, t_1, T_2 - t_1) < C(x_1, t_1, T_1 - t_1). \quad (15)$$

If we take $h \to 0$, the trajectory produced by the process described above will approach the trajectory of a follower that evolves under GCLP, while (15) indicates that the cost along $x_2(t)$ must be strictly less than that of $x_1(t)$, contradicting the convergence of the trajectory costs under GCLP. □

In the subsequent discussion, it will be convenient to distinguish between the *planned trajectory*, denoted by $\hat{x}_t(\tau)$, that a follower computes at time *t* in order to reach its leader's state, and the *realized trajectory*, denoted by $x(t)$, along which the follower actually evolves.

Lemma 5. *Let $\hat{x}_{k,t}(\tau)$ be the family of planned trajectories that the follower x_k computes via GCLP at time t, in order to reach $x_{k-1}(t)$ optimally from $x_k(t)$. If during GCLP*

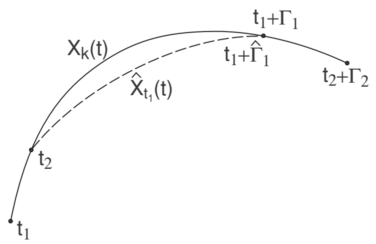

Figure 7. Assumed differences between the planned and realized trajectories contradict the convergence of trajectories under GCLP.

(i) *the locally optimal trajectory from follower to leader (or to S_Q) is unique, and*
(ii) $x_{k-1} = x_\infty$ *(see lemma 4),*
then $\hat{x}_{k,t_0}(t) = x_k(t) \forall\ t_0 \in [t_k, t_k + \Gamma_Q^*]$*, i.e., along the limiting trajectory produced under GCLP, the planned and realized trajectories overlap.*

Furthermore, if the locally optimal trajectories obtained at every updating time are smooth, then the limiting trajectory is also smooth.

Proof. Suppose that a leader x_{k-1} evolves along the limiting trajectory $x_\infty(t)$. Then, lemma 4 implies that $x_{k-1}(t) = x_k(t + \Delta)\ \forall\ t \in [t_k, t_k + T_k]$. Suppose also that at some time t_1, the leader is at $x_{k-1}(t_1 + \Gamma(t_1))$, where $\Gamma(t_1)$ is optimal, and the follower is at $x_k(t_1)$. Assume that the follower's planned trajectory $\hat{x}_{t_1}(t)$ $(t \in [t_1, t_1 + \hat{\Gamma}(t_1))$ (where, for convenience, we have dropped the subscript k in $\hat{x}_{k,t_1}$) differs from $x_k(t)$ $(t \in [t_1, t_1 + \Gamma(t_1))$, starting at some time $t_2 \geqslant t_1$. Furthermore, let $\hat{x}_{t_1}(t_1 + \hat{\Gamma}(t_1)) = x(t_1 + \Gamma(t_1))$. Because the planned trajectory $\hat{x}_{t_1}(t)$ is unique (by assumption) and optimal,

$$C(\hat{x}_{t_1}, t_2, \hat{\Gamma}(t_1) - (t_2 - t_1)) < C(x_k, t_2, \Gamma(t_1) - (t_2 - t_1)).$$

Now, construct the trajectory

$$\bar{x}(t) = \begin{cases} \hat{x}_{t_1}(t) & t \in [t_2, t_1 + \hat{\Gamma}(t_1)) \\ x_k(t - \hat{\Gamma}(t_1) + \Gamma(t_1)) & t \in [t_1 + \hat{\Gamma}(t_1), t_2 + \Gamma(t_2)]. \end{cases}$$

Clearly, $\bar{x}$ has lower cost than $x_k(t)$ $(t \in [t_2, t_2 + \Gamma(t_2)])$ (see figure 7). Thus, under GCLP, the follower would have chosen to evolve along $\bar{x}$ (or another trajectory with even lower cost) instead of $x_k(t)$ $(t \in [t_2, t_2 + \Gamma(t_2)])$. This contradicts the convergence to a limiting trajectory. The same argument can be applied at any other updating time, so that we may conclude that $\hat{x}_{t_1}(t) = x_k(t)$ $(t \in [0, T_k])$.

Finally, recall that $x_k(t)$ is smooth for $t \in [t_1, t_1 + \Gamma(t_1)]$, because the locally optimal trajectories linking the follower and leader are smooth by assumption. Similarly, $x_k(t)$ is smooth for $t \in [t_2, t_2 + \Gamma(t_2)]$ for any $t_1 < t_2 < t_1 + \Gamma(t_1)$. Therefore, $x_k(t)$ is smooth on $[t_1, t_2 + \Gamma(t_2)]$. Repeated applications of this argument lead to the conclusion that the entire trajectory $x_k(t)$ $(t \in [0, T_k])$ is smooth. □

The next theorem is an immediate consequence of lemmas 1–5:

Theorem 1. *Suppose that the group of agents (1) evolves under GCLP, and that at all times t, the locally optimal trajectories from the follower to leader are unique. Then, the limiting trajectory is unique and locally optimal. It is also smooth if the locally optimal trajectories calculated at every updating time are smooth.*

Proof. From lemma 4, the limiting trajectory is unique. It follows that $x_{k-1}(t - \Delta) = x_k(t)$ if $x_{k-1}(t) = x_\infty(t - t_{k-1})$. Choose δ_1, δ_2 such that $0 < \delta_1 < \delta_2 < \Gamma$ for all optimal final times Γ of the planned trajectories $\hat{x}_k$ generated during GCLP. The limiting trajectory x_∞ is piecewise smooth and locally optimal for $t \in [t_k + i\delta_1, t_k + i\delta_1 + \delta_2]$, $i = 0, 1, 2 \ldots$, because it coincides with the planned trajectories $\hat{x}_k(t)$. From lemma 3—in this case S_Q is a single point—we conclude that $x_k(t)$ $(t \in [t_k, t_k + \delta_1 + \delta_2])$ is optimal because it is the 'composition' of two overlapping locally optimal trajectories, $x_k(t)$ $(t \in [t_k, t_k + \delta_2])$ and $x_k(t)$ $(t \in [t_k + \delta_1, t_k + \delta_1 + \delta_2])$. From successive applications of this argument $(i = 2, 3, \ldots)$, it follows that $x_\infty(t)$ is locally optimal. Smoothness of x_∞ is proved in similar fashion, 'piece by piece'. □

4.1. Some remarks on the GCLP/GSLP algorithms

Local pursuit is a cooperative, decentralized algorithm for learning optimal controls/trajectories, starting from a feasible solution. The pursuit strategies discussed here are appropriate for systems with short-range sensors, and optimal control problems which are computationally easier to solve over 'short' distances. In settings where the available data are not sufficient for an individual to operate successfully, the algorithm places the solution within reach of a group of agents whose cooperation overcomes the lack of information. More specifically, agents are given only a set of 'instructions' (the initial, feasible control $u_0(\cdot)$) to reach the target set, but no description of a trajectory to that set and no map that describes the terrain or manifold on which they evolve. Each agent is only required to calculate optimal trajectories from its own state to that of its nearby leader. Because agents are separated by Δ time units as they leave x_S, each agent relies on local information only in order to follow its predecessor, and requires no knowledge of the global geometry. Therefore, there is no need for agents to exchange or 'fuse' local maps that they obtain individually.

In GCSP/GSLP, agents do not need to communicate their choices of coordinate systems as they evolve, nor do they need to know the coordinates of the target state. While it is possible that a group of agents could disperse and construct a global map from local information (in an attempt to solve the optimal control problem 'all at once'), such an approach might require significantly more computation and communication than local pursuit.

The sampled version of our algorithm (GSLP) can be useful as a numerical method for computing optimal controls, in the following manner. Local pursuit allows a group to solve a large, difficult optimization problem in smaller, 'overlapping' pieces. This idea can be applied in numerical optimal control, where control/state trajectory pairs are typically sampled, and the location of the samples is then optimized using, for example, some type of a gradient method. For problems with long time horizons or dense sampling grids,

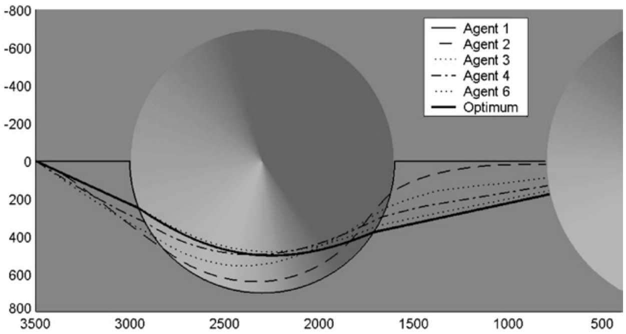

Figure 8. Continuous local pursuit in a complex environment. The initial trajectory (along the borders of the cones) is easily described but far away from optimal. The locally optimal trajectories were easier to compute than the global optimum because of the limited pursuit distance ($\Delta = 0.2T_0$). The iterated trajectories converged to the optimum.

the memory requirements can exceed the capacity of a typical PC. Local pursuit can circumvent this difficulty by solving many smaller (shorter time horizon) instances of the problem, which satisfy any storage constraints. There are also instances where numerical optimization methods may fail to converge because of singularities or numerical errors that arise when the problem is solved 'in one piece', but are well behaved when the problem is solved in smaller segments. A detailed discussion of GSLP's performance as a numerical optimization tool, together with examples, can be found in [24].

The GCLP and GSLP algorithms assume a countable infinity of agents; of course, such a collection cannot be realized. However, for some classes of problems, it is possible to achieve the same results with a finite number of agents that employ local pursuit to reach the final constraint set S_Q from x_S, then return to x_S by reversing course and exchanging the roles of S_Q and x_S. The required modifications are straightforward but will not be discussed here. An experiment that uses this technique is detailed in [18]. Finally, local pursuit is not guaranteed to converge to the global optimum. The choice of agent separation Δ can affect whether the limiting trajectory is a local or a global optimum. Some interesting cases involving spaces with 'holes' or 'obstacles' are discussed in [18].

5. Simulations and experiments

In this section, we describe a series of simulations and an experiment designed to illustrate the performance of GCLP.

5.1. Trail optimization with partially constrained final state

Consider the problem of finding shortest three-dimensional (3D) Euclidean paths in an environment consisting of a plane with two right cones, whose (partial) top view is shown in figure 8. The radii of the cones were 700 and 900 units of length, respectively. Each object (the plane and each cone) was parametrized with its own set of coordinate functions. The agents were governed by $\dot{x}_k = u_k$, $\|u_k\| = 1$ and were required to travel from $x_S = (3500, 0, 0)$ to the second cone.

Figure 8 shows the iterated trajectories generated when the agents implemented the GCLP policy with $T_0 = 3499$, $\Delta = 0.2T_0$. The initial trajectory was restricted to the plane, following the boundary of the first cone. The trajectory of the eighth agent was effectively optimal. Starting from x_S, each agent advanced by solving its own optimal control problem with boundary conditions given by the states of the agent and its follower (or the agent and the second cone). This was simpler than solving the 'global' problem, partly because of the fact that the globally optimal trajectory crosses multiple coordinate patches from the plane to the cone(s) and vice versa. When the leader and follower were both on the plane, or on the same cone, the computation of optimal trajectories was straightforward. In other cases, agents had to optimize trajectories that crossed at most two coordinate patches (plane-to-cone or cone-to-plane), selecting from a 1-parameter family of curves joining the leader and follower. On the other hand, computing the globally optimal trajectory at once would have required searching over a 3-parameter family of curves (there are a total of three 'crossings' between coordinate patches).

5.2. Minimum-time control with speed and acceleration constraints

Consider the minimum-time control of the second-order system

$$\ddot{x} = u, \text{ s.t. } |u| \leqslant 30, |\dot{x}| \leqslant 8$$

where we seek to minimize $J(x, \dot{x}, 0) = T$, with boundary conditions $\dot{x}(0) = \dot{x}(T) = x(0) = 0$, and $x(T)$ fixed. Here, the constraint set S_Q is a single point in the state space. The optimal policy for this problem is an instance of the well-known 'bang-off-bang' control: u switches at most once between 30 and -30, with $u = 0$ when the maximum for $|\dot{x}|$ has been reached. The initial, suboptimal input (agent 1 in figure 9), alternated between the maximum and minimum available acceleration. When using GCLP with $\Delta = 1.3$ s, the third agent's trajectory was optimal (see figure 9). Note that after $t > 2.7$ s the second agent intercepted the first and subsequently moved along the same trajectory, x_1. It is also interesting to note that in this case, optimality was achieved after a finite number of iterations.

5.3. An experiment in minimum-time control

We implemented the example of section 5.2 using a collection of three motors, pictured in figure 10. Each motor was equipped with position and rate sensors, which were sampled by a PC-based controller at the rate of 2000 Hz. The goal was to rotate the motors to a fixed final position in minimum time. Motor acceleration and speed were limited to 30 rad s^{-2} and 8 rad s^{-1}, respectively. The input to the first motor was a rectangular pulse with amplitude equal to the maximum acceleration (as in the simulation of section 2.4). Each of the remaining two motors tried to 'catch up' with its predecessor by measuring the predecessor's state and applying

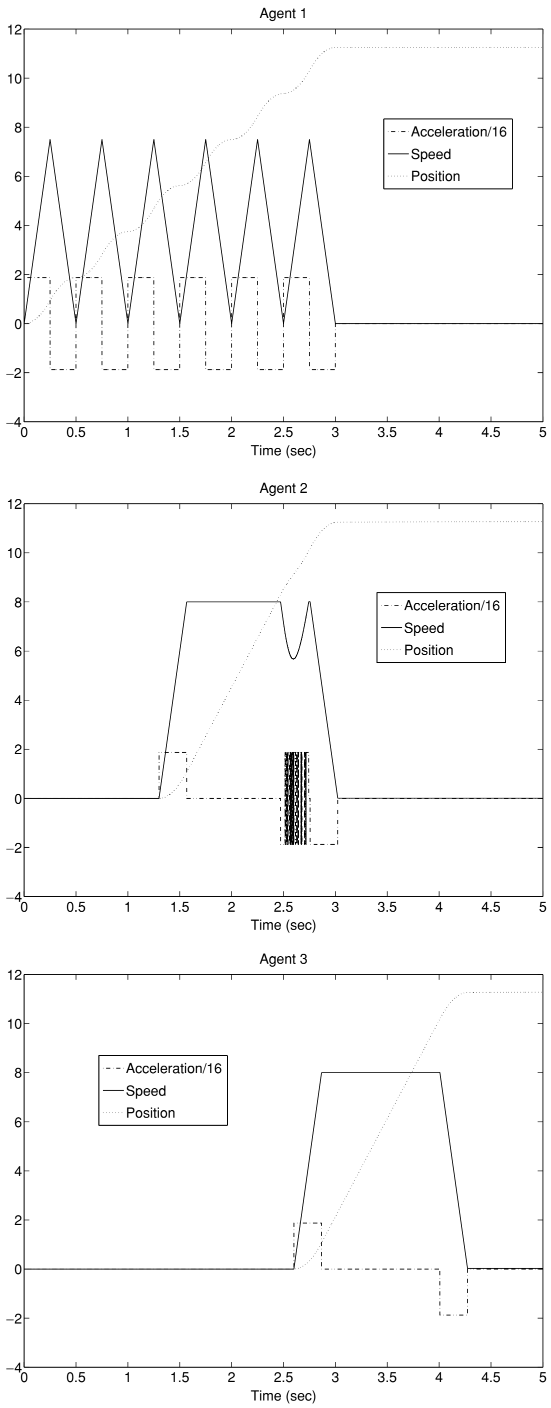

Figure 9. Iterative trajectories for minimum control with limited acceleration and speed. The pursuit interval was $\Delta = 1.3$. Units for acceleration, velocity and position are rad s^{-2}, rad s^{-1}, rad, respectively.

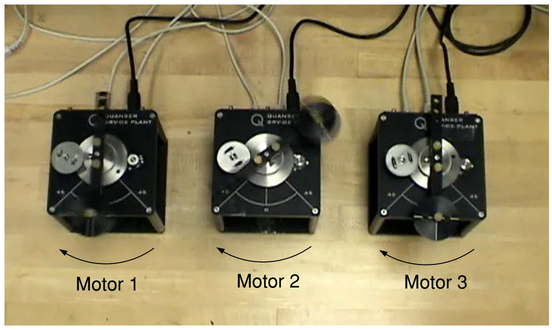

Figure 10. Application of local pursuit with a trio of motors to obtain minimum-time control with limited acceleration and speed.

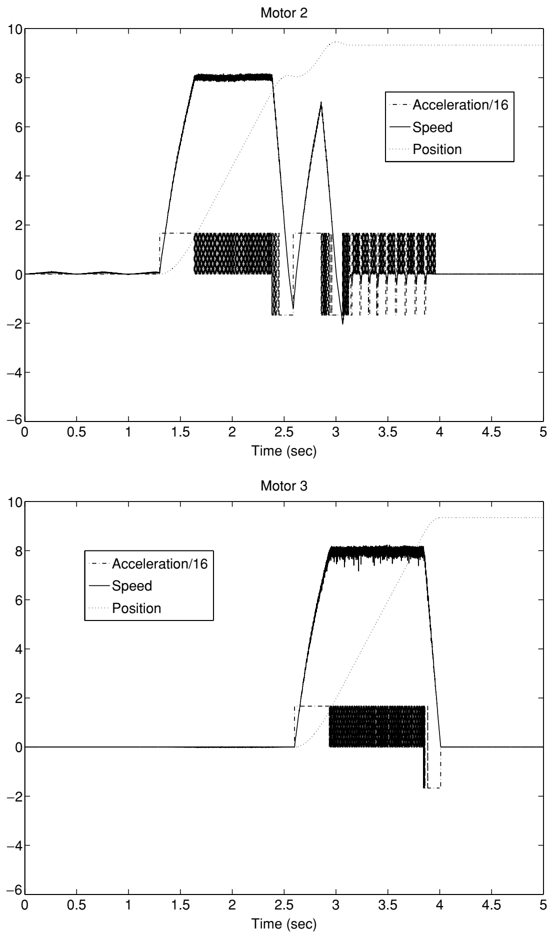

Figure 11. Iterative trajectories of motors under GCLP to attain minimum-time control with limited acceleration and speed. The pursuit interval $\Delta = 1.3$. The third motor evolved under essentially optimal control.

the control necessary to reach that state in the minimum time.

The trajectories of the second and third motors with $\Delta = 1.3$ s are shown in figure 11. The trajectory of the first motor was similar to that of the first simulated agent in figure 9 and has been omitted. The third motor evolved under essentially optimal control, and that the second motor 'intercepted' the first after $t \approx 2.3$ s. We note that because of unmodeled friction, the final position $\theta(T)$ was less than the nominal value (see $x(T)$ in the last simulation). The presence of friction also caused the motors to decelerate when a zero input was applied (once the motors had reached the maximum speed). That deceleration in turn caused the GCLP policy to try and 'catch up' by introducing a positive control input, resulting in the chatter observed in the velocity and acceleration curves of motors 2 and 3 in figure 11.

6. Conclusions and ongoing work

We discussed a biologically inspired cooperative strategy (termed 'generalized local pursuit') for solving a class of optimal control problems with free final time and partially constrained final state. The algorithm presented here generalizes the previously proposed models that mimic the foraging behavior of ant colonies and allow a collective to 'discover' optimal controls, starting from an initial suboptimal solution. Members of the collective are only required to obtain local information on their environment and to calculate optimal trajectories to their nearby neighbors. The local pursuit algorithm relies on cooperation to perform a task which would be difficult or impossible for a single system to perform, namely solving an optimal control problem with limited information (in terms of coordinate systems that describe the environment or the coordinates of the final state) and short-range interactions among agents. The algorithms' convergence rate and a rigorous description of the class of optimal control problems for which GCLP converges in a finite number of iterations are the subject of ongoing work.

Acknowledgments

This work was supported by the National Science Foundation under grant no EIA0088081 and by ARO ODDR&E MURI01 grant no DAAD19-01-1-0465 (Center for Communicating Networked Control Systems, through Boston University).

References

[1] Parrish J K and Hammer W M 1997 *Animal Groups in Three Dimensions* (Cambridge: Cambridge University Press)
[2] Camazine S, Deneubourg J-L, Franks N R, Sneyd J, Theraulaz G and Bonabeau E 2001 *Self-Organization in Biological Systems* (Princeton, NJ: Princeton University Press)
[3] Bruckstein A M 1993 Why the ant trails look so straight and nice *Math. Intell.* **15** 59–62
[4] Vicsek T, Czirok A, Jacob E B, Cohen I and Schochet O 1995 Novel type of phase transitions in a system of self-driven particles *Phys. Rev. Lett.* **75** 1226–9
[5] Jadbabaie A, Lin J and Morse A S 2003 Coordination of groups of mobile autonomous agents using nearest neighbor rules *IEEE Trans. Autom. Control* **48** 988–1001
[6] Passino K, Polycarpou M, Jacques D, Pachter M, Liu Y, Yang Y, Flint M and Baum M 2002 Cooperative control for autonomous air vehicles *Cooperative Control and Optimization* ed R Murphey and P M Pardalos (Dordrecht: Kluwer) pp 233–72
[7] Zhang F and Krishnaprasad P S 2004 Co-ordinated orbit transfer of satellite clusters *Astrodynamics, Space Missions, and Chaos, Ann. New York Acad. Sci.* **1017** 12–137
[8] Wagner I A, Lindenbaum M and Bruckstein A M 1999 Distributed covering by ant-robots using evaporating traces *IEEE Trans. Robot. Autom.* **15** 918–33
[9] Brooks R A and Flynn A M 1989 Fast, cheap and out of control: a robot invasion of the solar system *J. Br. Interplanet. Soc.* **42** 478–85
[10] Roumeliotis S I and Bekey G A 2002 Distributed multi-robot localization *IEEE Trans. Robot. Autom.* **18** 781–95
[11] Kurazume R and Hirose S 1998 Study on cooperative positioning system: optimum moving strategies for cps-iii *Proc. 1998 IEEE Int. Conf. in Robotics and Automation (Leuven, May)* vol 4, pp 2896–903
[12] Fierro R, Song P, Das A and Kumar V 2002 Cooperative control of robot formation *Cooperative Control and Optimization* ed R Murphey and P M Pardalos (Dordrecht: Kluwer) pp 73–93
[13] Yamaguchi H and Burdick J W 1998 Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations *Proc. IEEE Int. Conf. Robotics and Automation* vol 4, pp 3573–80
[14] Ögren P, Fiorelli E and Leonard N E 2002 Formation with a mission: stable coordination of vehicle group maneuvers *Proc. 15th Int. Symp. MTNS (Notre Dame, IL, August)*
[15] Dorigo M, Maniezzo V and Colorni A 1996 Ant systems: optimization by a colony of cooperating agents *IEEE Trans. Syst. Man Cybern.* B **26** 29–41
[16] Bruckstein A M, Mallows C L and Wagner I A 1997 Probabilistic pursuits on the grid *Am. Math. Mon.* **104** 323–43
[17] Hristu-Varsakelis D 2000 Robot formations: learning minimum-length paths on uneven terrain *Proc. 8th IEEE Mediterranean Conf. on Control and Automation*
[18] Hristu-Varsakelis D and Shao C 2004 Biologically-inspired optimal control: learning from social insects *Int. J. Control* **77** 1545–66
[19] Hristu-Varsakelis D and Shao C 2005 corrections to Biologically-inspired optimal control: learning from social insects *Int. J. Control* **78** 157
[20] Couzin I D and Franks N R 2003 Self-organised lane formation and optimised traffic flow in army ants *Proc. R. Soc.* B **270** 139–46
[21] Bryson A E and Ho Y-C 1979 *Applied Optimal Control* (New York: Wiley) (revised edition)
[22] Kirk D E 2004 *Optimal Control Theory: An Introduction* (New York: Dover)
[23] Sontag E 1998 *Mathematical Control Theory: Deterministic Finite-Dimensional Systems (Tests in Applied Mathematics)* 2nd edn (Berlin: Springer)
[24] Shao C and Hristu-Varsakelis D 2005 Local pursuit as a bio-inspired computational optimal control tool Technical Report TR2005-85, ISR, University of Maryland, College Park, MD 20742 (http://techreports.isr.umd.edu/reports/2005/TR_2005-85.pdf)