TECHNICAL RESEARCH REPORT

Bio-Inspired Optimal Control with Partially-Constrained Final State

by Cheng Shao, Dimitrios Hristu-Varsakelis

TR 2005-76

Bio-Inspired Cooperative Optimal Control with Partially-Constrained Final State ⋆

Cheng Shao [a], Dimitrios Hristu-Varsakelis [a,*]

[a] *Department of Mechanical Engineering and Institute for Systems Research University of Maryland, College Park, MD 20742 USA*

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 a cooperative, 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.

Key words: Co-operative control, Optimization, Algorithms, Agents, Group work, Trajectories, Minimum-time control

1 Introduction

In recent years, problems in cooperative control are increasingly capturing the attention of researchers, fueled by the development of decentralized control systems with cost and performance advantages. The rising interest in cooperative systems also stems from their potential to perform tasks which are not feasible for individuals. Examples include remote exploration and information gathering[13] by swarms of small autonomous robots [1], and satellite arrays, to name a few.

Members of such "engineered collectives" usually have limited sensing, communication, and computing capabilities, just like their natural counterparts. This suggests that each member can only perform relatively simple tasks. However, individual limitations can often be overcome by cooperation, if one can identify an effective way to organize the group into "more than the sum of its parts". Doing so may be difficult because it requires decomposing a desired group behavior into individual behaviors. The results however, can be spectacular, as is often demonstrated by biological collectives. For example, a school of fish can coordinate their movement in a tight formation and respond almost as fast as a single organism to evade encountering dangers; worker honey bees share information by "dancing" and distribute themselves among nectar sources in accordance with the profitability of each source; ants are known to utilize pheromone secretions for recruiting nest-mates and for optimizing their foraging trails [4]. Observations of such activities in nature have already seeded a variety of research, from modeling of animal group behaviors [4,2,17,10], to distributed collective covering and searching [18,13], cooperative estimation [14,11], cooperative robotic teams [6,19,12] and biologically-motivated optimization [5,3].

A particularly interesting example of cooperation in animal aggregates has to do with the foraging activity of ant colonies. Ants are able to recruit their co-workers to convey food back to the nest when they find it. Finding an efficient (short) path between the nest and the food source appears to be too complex for individual ants to accomplish, considering their limited cognition and size relatively to the obstacles in the environment, including stones, sticks and crevices. Nonetheless, ant colonies exhibit a high degree of competence in such tasks [4].

⋆ 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).

* Corresponding author. Tel: +1-301-405-5283, Fax: +1-301-314-9477.

Email addresses: cshao@glue.umd.edu (Cheng Shao), hristu@umd.edu (Dimitrios Hristu-Varsakelis).

Several models have been proposed in the attempt to capture the organizing principle by which ants find shortest paths when foraging. For example, [4] described a model based on the use of pheromonal secretions that ants lay down to recruit nestmates and to indicate the frequency of use for a particular path. Inspired by that model, [18] developed robust adaptive algorithms to perform tasks requiring the traversal of an unknown region, such as cleaning the floor of an unmapped building; [5] introduced a search methodology based on the "distributed autocatalytic process" to solve a classical optimization problem, the traveling salesman problem.

A particularly simple – but elegant – ant colony organizing rule was presented in [2], where it was shown that ants that "pursued" one another on $\mathbb{R}^2$ (each pointing its velocity vector towards a predecessor) had the effect of producing progressively "straighter" trails. That idea was later extended to path optimization problems involving kinematic vehicles in non-Euclidean environments [7–9].

In particular, [2,7] dealt exclusively with the "discovery" of geodesics, meaning that the autonomous system-members of the group had simple dynamics ($\dot{x} = u$), with no drift terms. In [8,9], it was shown that the earlier work could be generalized to a much broader class of optimal control problems, and collectives whose members have non-trivial dynamics. The proposed algorithm, termed "local pursuit" (to use the term coined in [2]), guides members of a group toward the solution of an optimal control problem. To the authors' knowledge, the pursuit algorithms presented to date (e.g., [8,9] and references therein) have been restricted to problems with fixed final time and fixed final states. The contribution of this paper is a modified version of local pursuit for solving a broader and more interesting class of optimal control problems, with free final time and partially-constrained final states.

As in the special cases treated in [8,9,2,7]), the proposed pursuit strategy does not require members of the collective to possess a global map of their environment, or even an agreed-upon common coordinate system. This reduces the sensing, communication and computational demands on the collective. As we shall see, the proposed algorithm is most useful in trajectory optimization problems which are easier to solve when boundary conditions are "close" to one another (e.g., because of the members' computational or sensing limitations), 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.

The remainder of this paper is organized as follows: Section 2 describes the optimal control problem to be addressed, and proposes an iterative algorithm that is well-suited to groups of cooperating dynamical systems. Section 3 contains the main results regarding the group's trajectories when its members evolve under the proposed strategy. Section 4 presents a series of simulations and laboratory experiments that illustrate our approach.

2 A bio-inspired algorithm for optimal control

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 a control system:

$$\dot{x}_k = f(x_k, u_k), \ x_k(t) \in \mathbb{R}^n, u_k(t) \in \Omega \subset \mathbb{R}^m \tag{1}$$

for $k = 0, 1, 2 \ldots$. Physically, each instance of (1) could stand for a robot, UAV or other autonomous system.

2.1 Problem Statement and Notation

The problem under consideration is:

Problem 1 *Find a control $u^* \in \Omega$ and a final time Γ^* that minimize*

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

subject to the dynamics (1), with $x(t_0) = x_I$ given, and the final state constraint $Q(x(t_0 + \Gamma)) = 0$,

where it is assumed that $g(x(t), u(t)) \geq 0$, G is bounded below, and $Q(\cdot) : \mathbb{R}^n \to \mathbb{R}^q$ is an algebraic function of the state.

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

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

We will assume that $\partial Q/\partial x$ has constant rank in a neighborhood of the set 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 state [15,8,9], are special cases of what are considered here.

In the sequel we will make use of the following notation. 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 u and Γ only). We will write $\Gamma^*(a, b)$ for the corresponding optimal final time. The cost of following x^* will be denoted by:

$$\begin{aligned} \eta(a, b) &\triangleq \int_{t_0}^{t_0+\Gamma^*} g(x^*, u^*)dt + G(x^*(t_0 + \Gamma^*)) \\ &= \min_{u,\Gamma} J(u, x, \Gamma) \end{aligned} \tag{3}$$

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

$$\begin{aligned}\eta_Q(a) &\triangleq \int_{t_0}^{t_0+\Gamma_Q^*} g(x^*, u^*)dt + G(x^*(t_0+\Gamma_Q^*)) \\ &= \min_{x,\Gamma_Q} J(u, x, \Gamma_Q) \end{aligned} \tag{4}$$

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

We note that because the agent dynamics have no explicit dependence on time, $\eta(\cdot,\cdot)$, $\eta_Q(\cdot)$ and $J(\cdot,\cdot,\cdot)$, are independent of t_0. In the sequel, it will sometimes be convenient in some cases to discuss the solution of Problem 1 in terms of the *trajectories* of (1). Of course, given an initial condition $x(t_0)$, the trajectory $x(t)$ of (1) is uniquely determined by the control u, so, as statement of Problem 1 reflects, we are optimizing over inputs.

The cost of following a trajectory $x(t)$ of (1) generated by a generic control u during $[t', t'+\sigma]$ will be denoted by:

$$C(x, t', \sigma) \triangleq \int_{t'}^{t'+\sigma} g(x, u)dt + G(x(t'+\sigma)), \tag{5}$$

where $x(t), u(t)$ are defined on an interval containing $[t', t'+\sigma]$. The following facts can be derived easily from the properties of optimal trajectories and will be helpful in the sequel:

Fact 1 *Let eta, η_Q and C as defined in (3),(4),(5), and let $x_k(t)$ be a generic trajectory of (1). Then, the following hold:*

(1) $\eta(a, b) \le C(x_k, t_0, \Gamma)$ for any $x_k(\cdot)$ such that $x_k(t_0) = a,\ x_k(t_0+\Gamma) = b$.
(2) $\eta(a, c) \le \eta(a, b) + \eta(b, c)$ with $\sigma = \Gamma^*(a, b)$.
(3) $\eta_Q(a) \le \eta(a, b)$ for any $b \in S_Q$.

2.2 Algorithm

Assume that there is available an initial feasible (but suboptimal) control/trajectory pair $(u_{feas}(t), x_{feas}(t))$ for (1), obtained through a combination of a-priori knowledge about the problem and/or random exploration. Following the idea in [2,8,9], the agents x_k will leave the initial state x_I sequentially [1] and pursue one another towards the set S_Q, in a way which will be made precise shortly. The sequence is initiated with the first agent following x_{feas} to reach a point in S_Q. Each subsequent agent will attempt to intercept its predecessor – along optimal trajectories defined by (3) – as long as the predecessor has not reached S_Q. If the predecessor has already reached the constraint set S_Q, then the pursuer will ignore the preceding agent and instead evolve along the optimal trajectory defined by (4). The precise rules that govern the movement of each agent are:

Algorithm 1 *(Modified Continuous Local Pursuit): Let $x_0(t)$ ($t \in [0, T_0]$) be an initial trajectory satisfying (1) with $x_0(0) = x_I,\ Q(x_0(T_0)) = 0$. Choose $0 < \Delta \le T_0$.*

(1) For $k = 1, 2, 3\ldots$, let $t_k = k\Delta$ be the starting time of the k^{th} agent, i.e., $u_k(t) = 0, x_k(t) = x_I$ for $0 \le t \le t_k$.
(2) For all $t \in [t_k, t_k+T_k]$, let $u_t^*(\tau)$ be such that it achieves
$$\begin{cases} \eta(x_k(t), x_{k-1}(t)) & \text{if } x_{k-1}(t) \notin S_Q \\ \eta_Q(x_k(t)) & \text{if } x_{k-1}(t) \in S_Q \end{cases}$$
for the system $\dot{\hat{x}}_t(\tau) = f(\hat{x}_k(\tau), u_t^*(\tau))$, where
$$\tau \in \begin{cases} [t, t+\Gamma^*(x_k(t), x_{k-1}(t))] & \text{if } x_{k-1}(t) \notin S_Q \\ [t, t+\Gamma_Q^*(x_k(t), S_Q)] & \text{if } x_{k-1}(t) \in S_Q \end{cases}$$
(3) Apply $u_k(t) \triangleq u_t^*(t_k)$ to the k^{th} agent.
(4) Repeat from step 2, until the k^{th} agent reaches S_Q.

We will refer to Δ as the *pursuit interval.* When discussing pairs of agents during pursuit, the $(k-1)^{st}$ agent will be designated as the *leader* and the k^{th} 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 *modified continuous local pursuit* (mCLP) algorithm requires each follower to continuously re-plan its trajectory (by sensing $x_k(t)$ and computing $u_t^*(\tau)$) to catch up with its leader during the pursuit process. Continuous pursuit may imply a significant computational burden for each agent, especially in cases where the optimal trajectories "linking" follower and leader cannot be written down in closed form. For instances of Problem 1 where, for each follower, the optimal time to reach the leader is lower bounded for all time (as is the case

[1] It will be convenient (although by no means necessary) to assume that the initial state is an equillibrium point for (1), i.e., there exists a constant u_e such that $f(x_I, u_e) = 0$. If that is the case, then agents can "wait" at x_I until it is their time to begin pursuit. Without loss of generality, we will take $u_e = 0$.

for the examples in Section 4), it is possible to alter the previous algorithm so that each agent only performs a finite number of measurements and trajectory updates as it evolves from x_I to S_Q. This can be accomplished by defining a sampled version of mCLP, termed *modified sampled local pursuit* (mSLP), of which the algorithm in [8,9] is a special case:

Algorithm 2 *(Modified Sampled Local Pursuit): Let $x_0(t), t \in [0, T_0]$ be an initial trajectory satisfying (1) with $x_0(0) = x_I$, $Q(x_0(T_0)) = 0$. Choose the pursuit interval Δ such that $0 < \Delta \leq T_0$.*

(1) For $k = 1, 2, 3 \ldots$, let $t_k = k\Delta$ be the starting time of the k^{th} agent, i.e., $u_k(t) = 0$, $x_k(t) = x_I$ for $0 \leq t \leq t_k$.

(2) For $i = 0, 1, 2, 3, \ldots$, define $t_k^i = t_k^{i-1} + \delta_{i+1}, t_k^0 = t_k$ to be the times when the k^{th} agent will update its trajectory, where $\delta_i < \min(\Delta, \Gamma^*_{i-1})$, 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} ($\Gamma^*_{-1} = \Delta$). Let $u^*_t(\tau)$ be a control policy that achieves

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

subj. to (1), where

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

(3) Apply $u_k(t) = u^*_{t_k^i}(t)$ to the k^{th} agent during $t \in [t_k^i, t_k^{i+1})$.

(4) Repeat from step 2 until the k^{th} agent reaches S_Q.

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 mSLP each agent executes a finite number of updates of its trajectory, once every $\delta < \Delta$ time units. mSLP's reduced computational demands make it attractive in 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. In fact, the mSLP algorithm can itself be useful as a numerical method for computing optimal controls. The details of mSLP's numerical performance are outside the scope of this paper and will not be discussed here; see, however, [16].

As defined above, the mCLP and mSLP 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.

3 Main Results

In this section we explore the behavior of the group (1) under mCLP. Similar results can be derived for mSLP, along the lines of the discussion below.

We will begin by considering the sequence of trajectories $\{x_k(t)\}$ produced by mCLP. We will first investigate the convergence of the corresponding cost sequence, and then that of the trajectories themselves. In the subsequent discussion, it will be convenient to distinguish between the *planned trajectory*, denoted by $\hat{x}(t)$, that a follower computes at every point in time in order to reach its leader's state, and the *realized trajectory*, denoted by $x(t)$, along which the follower actually evolves.

Lemma 1 *Consider a leader-follower pair evolving under mCLP with pursuit interval Δ. Let the leader's trajectory be $x_{k-1}(t)$ ($t \in [t_{k-1}, t_{k-1} + T_{k-1}]$) 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 reproduces 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 $\lambda \geq T_{k-1} - \Delta$, 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*

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

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

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 Fig. 1). The cost along x_k satisfies

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

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

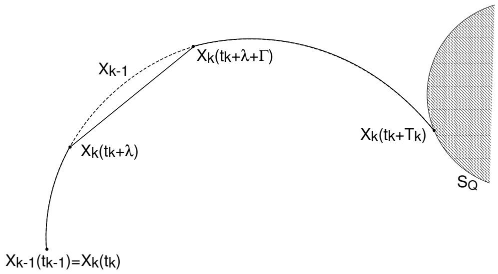

Fig. 1. Illustration of the trajectory obtained by a single update when $\lambda < T_{k-1} - \Delta$.

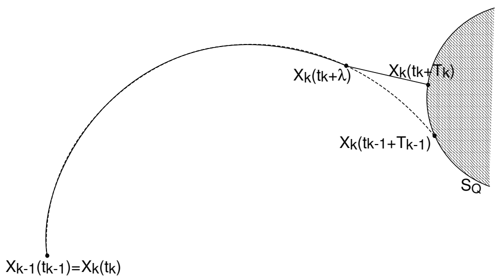

Fig. 2. Illustration of the trajectory obtained by a single update when $\lambda \geq T_{k-1} - \Delta$.

If $\lambda \geq T_{k-1} - \Delta$ (see Fig. 2), the cost along x_k is

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

Therefore the cost along the follower's trajectory is no greater than the leader's. $\square$

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

Lemma 2 *(Convergence of Cost) If the agents (1) evolve under mCLP, 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 $x_k^i(t)$ $(t \in [t_k, t_k + T_k^i]), i = 0, 1, 2\ldots,$ 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 Fig. 3).

From Lemma 1, the cost of each follower's trajectory will be no greater than the leader's. Also, the sequence C_k^i

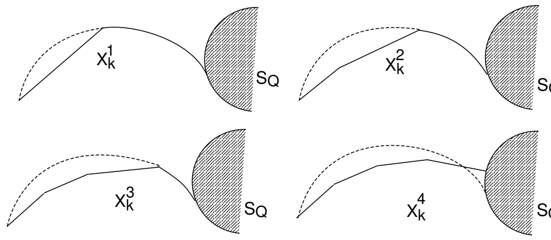

Fig. 3. Illustration of the trajectory sequence $x_k^i(t)$. Each trajectory is obtained by a single update upon its predecessor.

is bounded below for any fixed k. Thus, $C_k^i \leq C_k^{i-1}$ and $\lim_{k\to\infty} C_k^i = C_k^\infty$ exists for each k. Consequently,

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

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 mCLP. Hence, the follower's cost is $C_k = C_k^\infty \leq C_{k-1}$. Because the sequence $\{C_k\}$ is non-increasing and bounded below (there exists a minimum for (2)), it has a limit. $\square$

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

Condition 1 *$\forall\ a, b_1, b_2 \in \mathbb{D}, \Omega > 0$ and $\forall\ x_1(t)$ trajectory of (1) 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 $0 < \Delta \leq \Delta_1$. From the principle of optimality, $x^*(t)$ $(t \in [0, t_1 + \Delta])$ and $x^*(t)$ $(t \in [t_1, T^*])$ are each locally optimal with respect to their corresponding end points. Suppose that $\|x^*(t_1 + \Delta) - s\|_\infty \geq \varepsilon_1$ for any $s \in S_Q$ and that $x^*(t)$ $(t \in [0, T^*])$ is not a local minimum. 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 $C(x(t), 0, T) < C(x^*(t), 0, T^*)$.

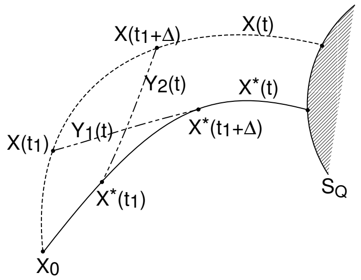

Fig. 4. Illustrating the proof of Lemma 3: "overlapping" optimal trajectories form a locally optimal trajectory.

Notice that $\|x(t_1 + \Delta) - s\|_\infty \geq \epsilon$ for any $s \in S_Q$. Construct two trajectories $y_1(t), y_2(t)$ $(t \in [t_1, t_1 + \Delta])$ that connect $x(t)$ and $x^*(t)$ (see Fig. 4) and satisfy Condition 1 (with x^* or x playing the role of x_1, and y_1 or y_2 standing in for x_2). In particular, let $y_1,\ y_2$ be such that $x^*(t_1) = y_2(t_1), x^*(t_1 + \Delta) = y_1(t_1 + \Delta), x(t_1) = y_1(t_1), x(t_1+\Delta) = y_2(t_1+\Delta)$. Now, Condition 1 implies that

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

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

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

and

$$\begin{aligned} &C(x^*(t), t_1, \Delta) + C(x^*(t), t_1 + \Delta, T^* - t_1 - \Delta) \\ &< C(x(t), t_1 + \Delta, T - t_1 - \Delta) + C(y_2(t), t_1, \Delta) \end{aligned} \tag{9}$$

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

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

We had assumed that the cost $C(x(t), 0, T)$ was less than $C(x^*(t), 0, T)$; but if Δ is chosen so that

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

then (10) cannot hold. This is a contradiction, because Δ 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 mCLP, the locally optimal trajectory from follower to leader (or to S_Q) is unique, then mCLP converges to a limiting trajectory $x_\infty(t)$.*

PROOF. Suppose that the trajectory costs converge but that there exist 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

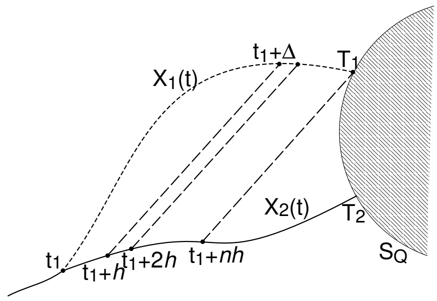

Fig. 5. Illustrating the proof of Lemma 4: pursuit between agents moving on two supposed "limiting" equal-cost trajectories, leads to the conclusion that the cost along the follower's trajectory is less than that along the leader's.

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 Fig. 5 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 composed of: 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 Fig. 5), 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 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 $C(x_2, t_1, h) + \eta(x_2(t_1 + h), x_1(t_1 + \Delta))$, and

$$\begin{aligned} &C(x_2, t_1, h) + \eta(x_2(t_1 + h), x_1(t_1 + \Delta)) \\ &< C(x_1, t_1, \Delta) \end{aligned} \tag{11}$$

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

$$\begin{aligned}&C(x_2, t_1 + h, h)\\&+\eta(x_2(t_1 + 2h), x_1(t_1 + \Delta + h))\\<\ &\eta(x_2(t_1 + h), x_1(t_1 + \Delta))\\&+C(x_1, t_1 + \Delta, h)\end{aligned} \tag{12}$$

......

$$\begin{aligned}&C(x_2, t_1 + (n-1)h)\\&+\eta(x_2(t_1 + nh), x_1(T_1))\\<\ &\eta(x_2(t_1 + (n-1)h), x_1(T_1 - h))\\&+C(x_1, T_1 - h, h)\end{aligned} \tag{13}$$

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

$$\begin{aligned}&C(x_2, t_1 + nh, T_2 - t_1 - nh)\\&< \eta(x_2(t_1 + nh), x_1(T_1))\end{aligned} \tag{14}$$

Notice that the (strict) inequalities holds for arbitrarily small $h > 0$. Thus, from (11)$\sim$(14) we have that

$$C(x_2, t_1, T_2 - t_1) < C(x_1, t_1, T_1 - t_1) \tag{15}$$

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

Lemma 5 *Let $\hat{x}_{k,t}(\tau)$ be the family of planned trajectories that the follower x_k computes via mCLP at time t, in order to reach $x_{k-1}(t)$ optimally from $x_k(t)$. If during mCLP:*
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 mCLP, 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 follower is at $x_k(t_1)$ and the leader is at $x_{k-1}(t_1)$, and that the leader reached that state $\Gamma(t_1)$ time units after being at $x_k(t_1)$, where $\Gamma(t_1)$ is optimal. Assume that the follower's planned trajectory is $\hat{x}_{t_1}(t)\quad (t \in [t_1, t_1 + \hat{\Gamma}(t_1))$ (where, for convenience,

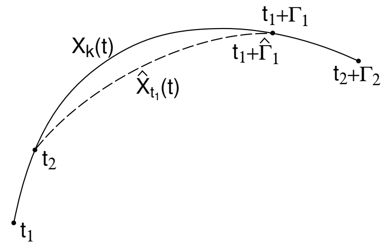

Fig. 6. Differences between the planned and realized trajectories contradict the convergence of trajectories under mCLP.

we have dropped the subscript k in $\hat{x}_{k,t_1}$) differs from $x_k(t)\quad (t \in [t_1, t_1 + \Gamma(t_1))$, starting at some time $t_2 \geq t_1$. 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 Fig. 6). Thus, under mCLP, the follower would 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 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. $\square$

The next theorem is an immediate consequence of Lemmas 1 $\sim$ 5:

Theorem 1 *Suppose that the group of agents (1) evolves under mCLP, that Cond. 1 holds, and that at all times t, the locally optimal trajectories from 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 mCLP. 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". □

3.1 Remarks

Local pursuit is a cooperative, decentralized algorithm for learning optimal controls/trajectories, starting from a feasible solution. 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_I, 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. Agents do not need to communicate their choice of coordinate systems as they evolve, nor do they need to know the coordinates of x_f. While it is possible that a group of agents could disperse and construct a global map from local information, such an approach might require significantly more computation and communication than local pursuit. The latter solves the optimal control problem in many "short pieces", which makes it no need to compute the optimum over the whole environment. Thus, local pursuit is appropriate for systems with short-range sensors (for example, in the case of a swarm of robots exploring unknown terrain), and optimal control problems which are easier to solve over "short" distances.

The mCLP and mSLP algorithms assumed a countable infinity of agents; of course, such a collection cannot be realized. It is however possible to achieve the same results with a finite number of agents that apply local pursuit to reach the final constraint set S_Q from x_I, then return to x_I along the obtained path. The required modifications are straightforward but will not be discussed here as they are beyond the scope of this paper. An experiment that uses this technique is detailed in [8,9]. 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 [8,9,15].

4 Simulations and Experiments

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

4.1 A trail optimization problem with free final states

Consider the problem of finding shortest paths in an environment consisting of a plane with two right cones, whose (partial) top view was shown in Fig. 7. The radii of the cones were 800 and 1000 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 $x_I = (3500, 0, 0)$ to the second cone.

Fig. 7 shows the iterated trajectories generated when the agents implemented the mCLP policy with $T_0 = 3499$, $\Delta = 0.2T_0$. For the computation of the optimal trajectory, each agent had to solve its own optimal control problem which was simpler than 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 vise 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 one-parameter family of curves joining leader and follower. On the other hand, computing the globally optimal trajectory at once would have required searching over a four-parameter family of curves (there are a total of four "crossings" between coordinate patches). A detailed accounting of the computational requirements and numerical performance of local pursuit can be found in [16].

4.2 Minimum-time control with speed and acceleration constraints

Next, consider the minimum-time control of the second-order system

$$\ddot{x} = u; \quad s.t. \;\; |u| \leq 30, \;\; |\dot{x}| \leq 8$$

where we seek to minimize $J(u, x, T) = T$, 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 control policy for this problem is similar to the well-known 'bang-bang" control: the control u switches at most once between 30 and -30, with $u = 0$ when the maximum or minimum speed $\dot{x}$ has been reached. The initial, suboptimal input (Agent 1 in Fig. 8), alternated between the maximum and minimum available acceleration. When using mCLP with $\Delta = 1.3$ sec, the third agent's trajectory was optimal (see Fig. 8).

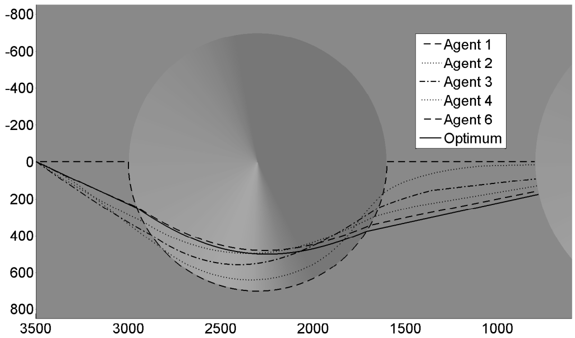

Fig. 7. Continuous local pursuit in a complex environment. The initial trajectory (along the borders of the cones) is easily described but far away from optimal. The locally optimal trajectories were easier to compute than the global optimum because of the limited pursuit distance ($\Delta = 0.2T_0$). The iterated trajectories converged to the optimum.

Notice that after $t > 2.7$ sec 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.

4.3 An experiment in minimum-time control

We implemented the example of Sec. 4.2 using a collection of three motors, pictured in Fig. 9. Each motor was equipped with position and speed sensors, which were sampled by a PC-based controller at a rate of $2000Hz$. The goal was to rotate the motors to a fixed final position in minimum time. Motor acceleration and speed were limited to 30 rad/sec^2 and 8 rad/sec, respectively. The input to the first motor was a rectangular pulse with amplitude equal to the maximum acceleration (same as in the simulation of Sec. 2.4). Each of the remaining two motors tried to "catch up" with its predecessor by measuring the predecessor's state and applying a control to reach that state in minimum time.

The trajectories of all three motors with $\Delta = 1.3$ sec are shown in Fig. 10. We see that the third motor evolved under essentially optimal control, and the second motor "intercepted" the first after $t \approx 2.3$ sec. 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 maximum speed). That deceleration in turn caused the mCLP policy to try and "catch up" by introducing a positive control input, resulting in chatter observed in the velocity and acceleration curves of

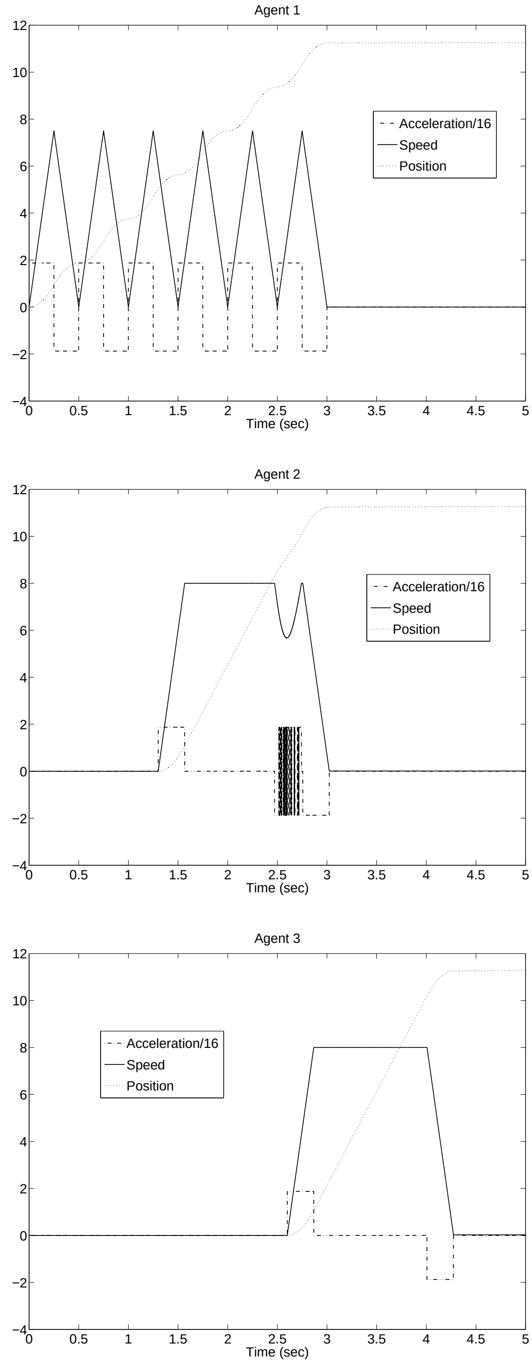

Fig. 8. Iterative trajectories for minimum control with limited acceleration and speed. The simulated control loop ran at a frequency of $2000Hz$ so that the control policy could be regarded as approximately mCLP. The pursuit interval was $\Delta = 1.3$. Units for acceleration, velocity and position are $Rad/s^2, Rad/s, Rad$, respectively.

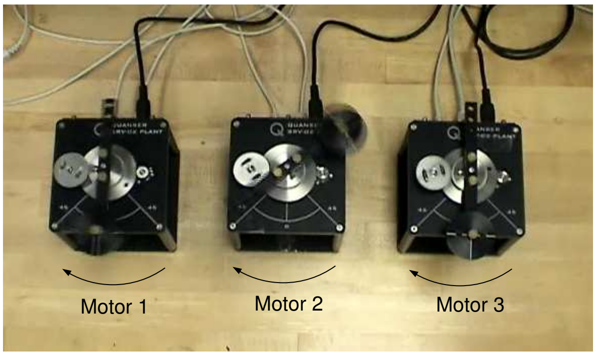

Fig. 9. Applying local pursuit with a trio of motors to obtain minimum-time control with limited acceleration and speed.

motors 2 and 3 in Fig. 10.

5 Conclusions and ongoing work

We discussed a biologically-inspired cooperative strategy (termed "Local Pursuit") for solving a class of optimal control problems with free final time and partially-constrained final state. The algorithms presented here generalize 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.

References

[1] R.A. Brooks and A.M Flynn. Fast, cheap and out of control: a robot invasion of the solar system. *Journal of the British Interplanetary Society*, 42:478–485, 1989.

[2] A.M. Bruckstein. Why the ant trails look so straight and nice. *The Mathematical Intelligencer*, 15(2):59–62, 1993.

[3] A.M. Bruckstein, C.L. Mallows, and I. A. Wagner. Probabilistic pursuits on the grid. *The American Mathematical Monthly*, 104(4):323–343, April 1997.

[4] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, New Jersey 08540, 2001.

[5] M. Dorigo, V. Maniezzo, and A. Colorni. Ant systems: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(1):29–41, 1996.

[6] R. Fierro, P. Song, A. Das, and V. Kumar. Cooperative control of robot formation. In R. Murphey and P.M. Pardalos,

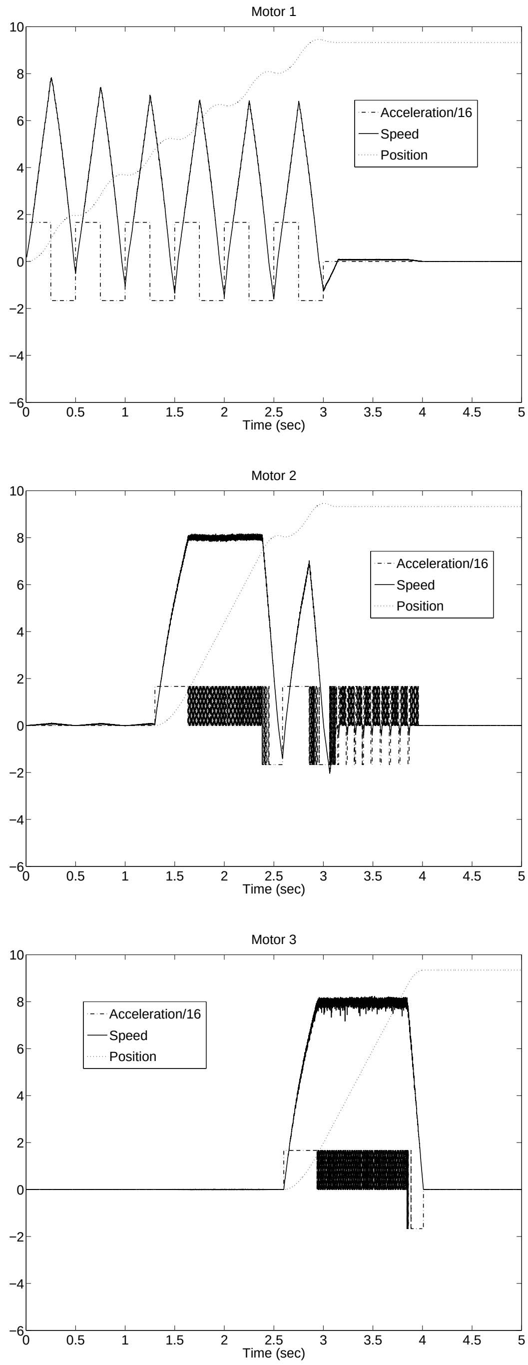

Fig. 10. Iterative trajectories of motors when applying local pursuit to attain minimum-time control with limited acceleration and speed. The pursuit interval $\Delta = 1.3$. The third motor evolved under essentially optimal control.

editors, *Cooperative control and optimization*, pages 73–93. Kluwer Academic Publishers, 2002.

[7] D. Hristu-Varsakelis. Robot formations: Learning minimum-length paths on uneven terrain. In *Proceedings of the 8th IEEE Mediterranean Conference on control and Automation*, 2000.

[8] D. Hristu-Varsakelis and C. Shao. Biologically-inspired optimal control: Learning from social insects. *International Journal of Control*, 77(18):1545–66, Dec. 2004.

[9] D. Hristu-Varsakelis and C. Shao. Corrections to: Biologically-inspired optimal control: Learning from social insects. *International Journal of Control*, 78(2):157, Jan. 2005.

[10] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), June 2003.

[11] R. Kurazume and S. Hirose. Study on cooperative positioning system: optimum moving strategies for cps-iii. In *Proceeding of the 1998 IEEE International Conference in Robotics and Automation*, volume 4, pages 2896–2903, Leuven, Belgium, May 1998.

[12] P. Ögren, E. Fiorelli, and N.E. Leonard. Formation with a mission: Stable coordination of vehicle group maneuvers. In *Proceedings Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, Notre Dame, IL, August 2002.

[13] K. Passino, M. Polycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint, and M. Baum. Cooperative control for autonomous air vehicles. In R. Murphey and P.M. Pardalos, editors, *Cooperative control and optimization*, pages 233–272. Kluwer Academic Publishers, 2002.

[14] S.I. Roumeliotis and G.A. Bekey. Distributed multi-robot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[15] C. Shao and D. Hristu-Varsakelis. Biologically inspired algorithms for optimal control. Technical Report TR2004-29, Institute for Systems Research, University of Maryland, College Park, MD 20742, 2004.

[16] C. Shao and D. Hristu-Varsakelis. The performance of a bio-inspired local pursuit strategy as a numerical method for optimal control. Technical Report TR2005-85, Institute for Systems Research, University of Maryland, College Park, MD 20742, 2005.

[17] T. Vicsek, A. Czirok, E.B. Jacob, I. Cohen, and O. Schochet. Novel type of phase transitions in a system of self-driven particles. *Physical Review Letters*, 75:1226–1229, 1995.

[18] I.A. Wagner, M. Lindenbaum, and A.M. Bruckstein. Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15(5):918–933, 1999.

[19] H. Yamaguchi and J.W. Burdick. Asymptotic stabilization of multiple nonholonomic mobile robots forming group formations. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3573–3580, 1998.