

# Minimizing Mobility and Communication Energy in Robotic Networks: an Optimal Control Approach

H. Jaleel, Y. Wardi, and M. Egerstedt<sup>†\*</sup>

**Abstract**—This paper concerns the problem of minimizing the sum of motion energy and communication energy in a network of mobile robots. The robotic network is charged with the task of transmitting sensor information from a given object to a remote station, and it has to arrange itself in a serial (tandem) configuration for point-to-point transmission, where each robot acts as a relay node. The problem is formulated in a dynamic setting where the robots move and communicate at the same time, and it is cast in the framework of optimal control. The paper proposes an effective algorithm for solving this problem and demonstrates its efficacy on a simulation example. In order to highlight the salient features of the algorithm the network is assumed to be one-dimensional, and the case of planar movement with obstacles is deferred to future research.

## I. INTRODUCTION

Power-aware mobile sensor networks have been investigated extensively in the past few years, where one of the main issues is how to use mobility to reduce the energy required for sensing and communications; see [13] and references therein. Most of these papers either ignore the cost of energy needed for mobility of the sensors, or assume unlimited sources of such energy. However, since the pioneering work of Goldenberg et al. [5], the question of balancing the energy costs of mobility and communication has attracted considerable attention; see, e.g., [9], [14], [15], [12], [6] and references therein.

The development of inexpensive mobile, wireless sensing devices in the past few years (e.g., [4], [8]) has suggested the eventual massive deployment of mobile sensor networks in communication and control applications [6]. In many such applications the devices (agents) are tasked with transmitting data from one or more source objects to a remote station (controller), and to this end they have to arrange themselves in a network configuration. However, the agents often are powered by on-board, limited-energy sources such as batteries, which cannot be replenished during the application's lifetime. Therefore, the network has to be configured in a way that balances, optimally, the energy required for communication and mobility.

Reference [5] devised a distributed motion control law for steering multiple mobile relays into a position of minimum communication energy. Although it does not explicitly include the motion energy in the problem formulation, it observes from extensive simulation studies that the computed

trajectories are close to linear and hence their motion energies are almost minimal. However, situations of relatively high motion energy could justify its explicit inclusion in the problem formulation. Reference [12] considers a robot tasked with transmitting a given number of bits while in motion on a predetermined trajectory with variable degrees of channel fading. Using a realistic, detailed, probabilistic model of the channel's fading, that paper determines the robot's speed, transmission rate, and stopping times that minimize the total energy required for mobility and communication. Reference [6] considers the task of distributing wireless mobile agents so as to provide transmission of sensor data from one or more objects to a remote station, and doing it in a way that minimizes the total required energy. That paper uses graph-theoretic techniques to compute an optimal strategy comprised of the sequential scheduling of motion followed by transmission. Reference [15] addresses the combined-energy minimization problem through the dynamic setting of optimal control of the agents' trajectories. Having a quadratic cost function the problem is cast in the framework of LQR, where complexity reduction is obtained first via model-predictive control and then by having a distributed algorithm. A similar approach was used to maximize the lifetime of the various sensors in [14].

This paper also considers the total energy-minimization problem in the dynamic setting of optimal control, where the agents carry out their communication tasks while in motion. It is different from [15] in that it assumes a general, convex quadratic energy function and hence does not fall in the category of LQR. Furthermore, it does not use an existing algorithm but rather develops a new computational technique. It is different from [6] in that the latter reference first computes the agents' trajectories and then minimizes their communication energy, while this paper considers the problem in a dynamic setting of optimal control where the agents carry out their communication tasks while in motion.

Consider a scenario in which a supervisory controller instructs a collection of wireless mobile agents to form a tandem, point-to-point connection for transporting sensory data from a given object to a remote station (controller). Sensing and communication must commence immediately and be maintained for a given amount of time. Meanwhile the agents are arranging themselves dynamically in a network configuration where each one of them acts as a relay between a single downstream node and a single upstream node, and they determine their trajectories in a way that minimizes the energy spent on both communication and motion. The power required for transmission on a link is related to

<sup>†</sup>School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. Email: hassanjaleel@gatech.edu, ywardi@ece.gatech.edu, magnus@ece.gatech.edu.

\* Research supported in part by NSF under Grant CNS-1239225.

the link's length, and the motion energy is related to the distance traveled. We define the problem in the setting of optimal control, and devise a highly-efficient algorithm for its solution, that eventually may lend itself to a natural distributed implementation.<sup>1</sup>

The contribution of the paper is twofold: a new algorithm that may have a wide scope in optimal control of dynamical systems, and its application to the aforementioned power-aware problem. Therefore we present the application problem in fairly general terms and we analyze the algorithm in this setting. Since the paper comprises an initial investigation, we consider only the case of a single stationary object, a stationary remote station, and a one-dimensional movement of the agents. As we shall see, the obtained simulation results are quite encouraging and suggest the potential viability of our approach in a wider context.

The algorithm that we propose is a descent technique whose direction is computed by minimizing the Hamiltonian at each time  $t$ . Obviously this is often impossible, and hence impractical in the general setting of optimal control, but the special structure of our problem makes it possible and even simple, and hence yields effective descent directions. The step size of the algorithm is determined via the Armijo procedure [11], [1] which, though having linear asymptotic convergence, often has the practical advantage of rapid progress at the initial phases of an algorithm's run. This point, demonstrated via simulations, will be argued to suggest the eventual use of the algorithm in real-time tuning of the agents' trajectories.

Section II describes the problem in the setting of optimal control, and Section III introduces the algorithm in a general setting and discusses its convergence. Section IV presents a simulation example, and Section V concludes the paper and suggests directions for future research.

## II. PROBLEM FORMULATION

Consider the network shown in Figure 1, consisting of  $N$  mobile agents,  $A_1, \dots, A_N$ , moving between an object and a remote controller station, indicated by  $O$  and  $C$  in the figure. Let  $x_k(t)$ ,  $k = 1, \dots, N$ , denote the relative position of  $A_k$  with respect to the object, and let  $d$  denote the relative position of the controller with respect to the object. Since we only consider motion in the line adjoining  $O$  to  $C$ , we have that  $x_k(t) \in \mathbb{R}$  and  $d \in \mathbb{R}$  as well. To simplify the notation we define  $x_0 = 0$  and  $x_{N+1} = d$ , and we note that these are the positions of  $O$  and  $C$ ; assuming that both the object and the controller station are stationary,  $x_0$  and  $x_{N+1}$  are constants and not functions of time. We define the vector

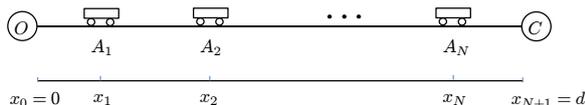


Fig. 1. Tandem network

<sup>1</sup>The issue of distributed implementation is deferred to a future study and is not considered in this paper.

notation  $x(t) := (x_1(t), \dots, x_N(t))^T \in \mathbb{R}^N$  to denote the position of the agents, and assume that  $x(0)$  is given and fixed. Furthermore, we define  $u(t) = (u_1(t), \dots, u_N(t))^T \in \mathbb{R}^N$  to be the vector of velocities of the agents, namely

$$\dot{x} = u, \quad (1)$$

where the notational dependence on  $t$  is suppressed. The problem that we consider is to determine the control  $u(t)$  and the related state trajectory  $x(t)$  (via (1)) for a given time-interval  $t \in [0, t_f]$ , in a way that minimizes a weighted sum of the agents' transmission energy and communication energy, subject to magnitude constraints on the controls.

The power required for moving an agent arguably is proportional to its speed [6], and hence the associated performance-functional term is  $J_{mobility} := \sum_{k=1}^N \int_0^{t_f} |u_k(t)| dt$ . For the transmission energy cost, let  $\psi(z) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  be a non-decreasing, continuously-differentiable function representing the transmission power of each agent over a link of length  $z$ . Commonly  $\psi(z) = a + bz^2$  for given constants  $a \geq 0$  and  $b > 0$  [6], but we consider a more general function  $\psi$ . Note that the transmission down the line is from  $A_n$  to  $A_{n+1}$ ,  $n = 0, \dots, N$ , and hence the total transmission energy can be represented by the cost functional  $J_{trans} := \sum_{k=1}^{N+1} \int_0^{t_f} (\psi(x_n(t) - x_{n-1}(t))) dt$ . The performance function that we consider is a weighted sum of  $J_{mobility}$  and  $J_{trans}$ , namely, for a given  $C > 0$ ,

$$J = \sum_{k=1}^{N+1} \int_0^{t_f} \psi(x_k(t) - x_{k-1}(t)) dt + C \sum_{k=1}^N \int_0^{t_f} |u_k(t)| dt. \quad (2)$$

The constraints that we consider are  $|u_k(t)| \leq 1$  for every  $k = 1, \dots, N$  and for all  $t \in [0, t_f]$ . The problem that we solve is to minimize  $J$  subject to these constraints.

Let us denote by  $p(t) = (p_1(t), \dots, p_N(t))^T \in \mathbb{R}^N$  the costate (adjoint) variable. Then by Equation (1), the costate equation is

$$\dot{p}_k = \frac{d\psi}{dx}(x_{k+1} - x_k) - \frac{d\psi}{dx}(x_k - x_{k-1}), \quad (3)$$

$k = 1, \dots, N$ , with the boundary condition  $p_k(t_f) = 0$ . The Hamiltonian has the following form,

$$H(x, u, p) = \sum_{k=1}^N (p_k u_k + C|u_k|) + \sum_{k=1}^{N+1} \psi_k(x_k - x_{k-1}), \quad (4)$$

and this particular form is especially suitable for the algorithm that we present in the next section.

## III. ALGORITHM

Our algorithm is described in this section in a general setting of optimal control, and its detailed implementation for the problem described in Section II will be presented in Section IV.

Consider a system defined by the following differential equation,

$$\dot{x} = f(x, u), \quad (5)$$

where  $x \in R^n$ ,  $u \in R^k$ , the function  $f : R^n \times R^k \rightarrow R^n$  is continuously differentiable, the initial condition  $x_0 := x(0)$  is given, and  $t \in [0, t_f]$  for a given final time  $t_f$ . Let  $L(x, u) : R^n \times R^k \rightarrow R$  be a continuously-differentiable cost function, and let

$$J := \int_0^{t_f} L(x, u) dt \quad (6)$$

be the associated cost-performance functional. Let  $U \subset R^k$  be a compact, convex set, and consider the optimal control problem of minimizing  $J$  subject to the dynamics in (5), the initial state  $x(0) = x_0$ , and the constraint that  $u(t) \in U$  for every  $t \in [0, t_f]$ . We say that a control  $u$  is admissible if it is piecewise continuous with bounded variation in  $t$  and  $u(t) \in U \forall t \in [0, t_f]$ . All of the controls mentioned in the sequel are implicitly assumed to be admissible unless stated otherwise.

Recall [2] that the costate  $p(t) \in R^n$  is defined via the equation

$$\dot{p} = -\left(\frac{\partial f}{\partial x}(x, u)\right)^\top p - \left(\frac{\partial L}{\partial x}(x, u)\right)^\top \quad (7)$$

with the boundary condition  $p(t_f) = 0$ . Given an admissible control  $u$ , let  $x$  and  $p$  denote the associated state trajectory and costate trajectory derived via Equations (5) and (7), and recall the Hamiltonian  $H(x(t), u(t), p(t)) := p(t)^\top f(x(t), u(t)) + L(x(t), u(t))$ . Let  $w(t)$  be an admissible control that minimizes the Hamiltonian at every time  $t$ , namely  $w(t) \in \arg \min \{H(x(t), w, p(t)) | w \in U\}$ .

Define the functional  $\theta(u)$  by

$$\theta(u) = \int_0^{t_f} (H(x, w, p) - H(x, u, p)) dt. \quad (8)$$

Then it is readily seen that  $\theta(u) \leq 0$  for every control  $u$ , and if  $\theta(u) = 0$  then  $u$  satisfies the maximum principle [2]. Moreover, the magnitude of  $\theta(u)$ ,  $|\theta(u)|$ , can be viewed as a measure of the extent to which  $u$  fails to satisfy the maximum principle. Such functionals  $\theta(\cdot)$  are said to be *optimality functions*, and they have been used to characterize and prove convergence of optimization algorithms for infinite-dimensional problems including optimal control [10], [11]. Typically, it is required of a nonlinear-programming algorithm that every accumulation point of an iterate-sequence it computes, satisfies an optimality condition such as stationarity or a Kuhn-Tucker point. However, in infinite-dimensional problems iteration-point sequences often do not have accumulation points, and hence convergence of an algorithm is defined via the following limit,

$$\lim_{i \rightarrow \infty} \theta(u_i) = 0, \quad (9)$$

where  $u_i$ ,  $i = 1, 2, \dots$ , is a sequence of iteration points that is computed by the algorithm (see [11]).

Consider such an iterative algorithm for our optimal control problem defined by (5)-(6), and suppose that it

computes a sequence of admissible controls,  $u_i$ ,  $i = 1, 2, \dots$ . Furthermore, given a control  $u$ , denote by  $T(u)$  the next iteration point that the algorithm computes from  $u$ , and thus,  $u_{i+1} = T(u_i)$ . We say that the algorithm is a descent method if for every control  $u$ ,  $J(T(u)) \leq J(u)$ , and we define a stronger property, called *uniform sufficient descent*, as follows:

*Definition 1:* An algorithm is of *uniform sufficient descent* with respect to the optimality function  $\theta(\cdot)$  if, for every  $\delta > 0$  there exists  $\eta > 0$  such that, for every admissible control  $u$ , if  $\theta(u) < -\delta$  then

$$J(T(u)) - J(u) < -\eta. \quad (10)$$

The following proposition is obvious.

*Proposition 1:* Suppose that a descent algorithm that is of uniform sufficient descent with respect to  $\theta$ , computes a sequence of iteration points (controls),  $u_i$ ,  $i = 1, 2, \dots$ . Then the algorithm converges in the sense that Equation (9) is satisfied.

*Proof:* By (5) and (6),  $|J(u)|$  is upper-bounded over the space of admissible controls. By assumptions, and by Definition 1, for every  $\delta > 0$ ,  $\theta(u_i) < -\delta$  for at most a finite number of controls  $u_i$ , and hence (9) follows. ■

We point out that more general versions of this proposition have been derived, and systematically used in [11] to prove convergence of algorithms in a general setting of optimization. More recently, References [3], [7] applied the sufficient-descent principle to optimal control problems defined on switched-mode systems.

We next describe our algorithm for the general optimal control problem described above. It moves from a given control  $u$  in the direction towards the control  $w$  that minimizes the Hamiltonian, namely it attempts to close the optimality gap. Moreover, it uses the Armijo step-size procedure to compute the step size in that direction. For every  $\lambda \geq 0$ , define the function  $\tilde{J}_{u,w}(\lambda) : R^+ \rightarrow R$  as

$$\tilde{J}_{u,w}(\lambda) = J((u + \lambda(w - u))). \quad (11)$$

The formal computation of  $T(u)$ , for a given control  $u$ , is as follows.

*Algorithm 1: Parameters:*  $\alpha \in (0, 1)$ ,  $\beta \in (0, 1)$ .

*Step 1:* Compute the state trajectory  $x(t)$  and costate trajectory  $p(t)$ , associated with the control  $u$ .

*Step 2:* Compute a control  $w := (w(t))|_{t \in [0, t_f]}$  such that  $w(t) \in \arg \min \{H(x(t), \cdot, p(t))\} \forall t \in [0, t_f]$ .

*Step 3:* Compute

$$k'(u) := \min \{k' = 0, 1, \dots, : \tilde{J}_{u,w}(\beta^{k'}) - \tilde{J}_{u,w}(0) \leq \alpha \beta^{k'} \theta(u)\}. \quad (12)$$

*Step 4:* Set the step size to be  $\lambda(u) := \beta^{k'(u)}$ , and set  $T(u) = u + \lambda(u)(w - u)$ .

A few remarks are due.

1. The algorithm does not move from  $u$  in the steepest-descent direction, namely against the Fréchet derivative

$dJ/du$ , but rather in a projected direction that aims at closing the optimality gap. Our experience with this algorithm indicates fast convergence towards a minimum, as we shall see in the next section. Furthermore, this direction is always feasible as long as  $U$  is convex, thus avoiding potential computational issues associated with evaluating steepest descents in the event that  $u$  lies on the boundary of the feasible set.

2. The requirement of computing  $w(t)$  for every  $t \in [0, t_f]$  can be a tall order. However, it is possible and even straightforward to compute it for several classes of problems, including those where the Hamiltonian is linear in  $u$  as well as our energy-minimization problem. The latter point will be explained in the next section.

3. The Armijo step size, commonly used in gradient-descent algorithms, is a form of approximate line minimization; see, e.g., [11]. That reference also contains practical guidelines including the following two: (i). Effective values of  $\alpha$  and  $\beta$  are  $\alpha = \beta = 0.5$  (ii). The search for  $k'(u_i)$  in Step 3 need not start at  $k' = 0$ ; rather, at  $\beta^{-2}k'(u_{i-1})$ . This can expedite the computation of Step 3 when the step sizes are small.

We next establish the algorithm's convergence.

*Assumption 1:* 1) The functions  $f(x, u)$  and  $L(x, u)$  are continuous in  $(x, u)$ .

2) There is a finite set  $U_0 \subset U$  such that, for every  $u \in U \setminus U_0$ , the functions  $f(x, u)$  and  $L(x, u)$  are twice-continuously differentiable in  $x$ , and have bounded first - and second - order derivatives

*Proposition 2:* Algorithm 1 has the sufficient-descent property.

The proof can be found in the appendix.

#### IV. SIMULATION EXAMPLES

This section presents results of the application of Algorithm 1 to the problem described in Section II. The system under study has 6 agents, and the problem in question is to minimize  $J$  as defined in (2) subject to the dynamics in (1) and the constraints  $|u_i(t)| \leq 1$ . The distance of the object from the controller is  $d = 20$ , and the final time is  $t_f = 20$ . The transmission power over a link of length  $z$  is  $\psi(z) = z^2$  and hence, in Equation (2),  $\psi(x_k - x_{k-1}) = (x_k - x_{k-1})^2$ . The constant  $C$  in (2) is  $C = 7$ , and the initial condition for the state equation (1) is  $x(0) = (1, 2, 7, 9, 12, 19)^\top$  for every control  $u$ . Algorithm 1 was used with  $\alpha = \beta = 0.5$ .

This problem can be solved analytically due to the particular form of the function  $\psi(z)$ , but we use the algorithm in order to examine its performance. The algorithm was run for 200 iterations computing, recursively, controls  $u_i$ ,  $u = 1, \dots, 200$ . Note that each control is six-dimensional,  $u_i = (u_{i,1}, \dots, u_{i,6})^\top$ , and we chose, arbitrarily, the initial control to be  $u_{1,k}(t) = 1.0$  for every  $k = 1, \dots, 6$ . We used a uniform grid overlaying the time-interval  $[0, t_f]$  with  $\Delta t = 0.01$ , for the various computations in Equations (1)-(4) and (8), and for the differential equations we used the

forward Euler method. The minimizer of the Hamiltonian in (4),  $w_k$ , is known to be

$$w_k(t) = \begin{cases} -1, & p_k(t) > C \\ 0, & -C < p_k(t) < C \\ 1, & p_k(t) > -C, \end{cases} \quad (13)$$

for all  $k = 1, \dots, 6$ .

The 200-iteration run took 8.67 seconds of CPU time. The results are shown in Figure 2, whose parts (a) and (b) depict the graphs of the cost and optimality function as functions of the iteration count, while parts (c) and (d) show the the final input and corresponding state trajectories. Throughout the course of 200 iterations the cost came down from  $J(u_1) = 7,969.0$  to  $J(u_{200}) = 1,253.6$ , but this reduction is by no means linear in the number of iterations. In fact, the graph of the cost  $J(u_i)$  as a function of  $i$ , shown in part (a) of the figure, shows a rapid decrease in a few iterations at the early part of the algorithms run, followed by a relatively flat curve. Moreover, it took only 7 and 14 iterations to achieve 95% and 98% of the total cost reduction, respectively, obtained by the algorithm's run. Correspondingly, the optimality function rises from  $\theta(u_1) = -28,537.4$  to  $\theta(u_{200}) = -3.506$ . The proximity of  $u_{200}$  to the optimum, or at least a local minimum, was tested by various runs of 400 iterations starting from different initial inputs. The lowest value of  $J(u_{400})$  we obtained was 1,252.3 as compared to  $J(u_{200}) = 1,253.6$ , and the corresponding value of the optimality function was  $-0.805$  as compared to  $\theta(u_{200}) = -3.506$ . Thus, we believe that the algorithm practically converged, and as indicated by parts (a) and (b) of the figure, quite rapidly. We point out that the L-shaped graph of Figure 2(a) is not atypical of gradient-descent algorithms with Armijo step sizes, whose efficacy often is reflected in its rapid descent of the performance function during the early stages of its runs and not necessarily in its asymptotic convergence rate.

Equation (13) indicates that the optimal control for the problem is a bang-of-bang control, and this is evident from the state trajectory  $x_{200}$  that is shown in Figure 2(d). However, not all of the components of  $u_{200}$ , depicted in Figure 2(c) indicate a bang-of-bang control. This does not contradict our earlier statement that  $u_{200}$  is very close to being optimal. To explain this point, we point out that there is a great degree of insensitivity of  $J$  to certain large  $L^1$ -variations in  $u$ . Moreover, we recall that the chattering lemma implies that certain large  $L^1$  variations in  $u$  yield small  $L^\infty$ -variations in the corresponding state trajectory  $x$ . Consequently, the proximity of a given control  $u$  to an optimal control often is measured in the weak topology rather than in the  $L^1$  topology, and in our case this means by the difference between  $J(u)$  and the cost value of the optimal control.

Recall that the algorithms' run of 200 iterations reduced the cost from  $J(u_1) = 7,969.0$  to  $J(u_{200}) = 1,253.6$  in 8.67 seconds of CPU time. Further reduction in computing times can be obtained by running fewer iterations: it takes CPU times of 0.8703, 1.669, and 4.115 seconds to execute 20 iterations with  $J(u_{20}) = 1,335.0$ , 40 iterations with

$J(u_{40}) = 1,284.0$ , and 100 iterations with  $J(u_{100}) = 1,257.5$  respectively. More reductions can be obtained by increasing  $\Delta t$ . In fact, we ran the algorithm with  $\Delta t = 0.1$ , and the resulting values were  $J(u_{20}) = 1,335.5$  with CPU time of 0.178;  $J(u_{100}) = 1,259.4$  with CPU time of 0.5741; and  $J(u_{200}) = 1,255$  with CPU time of 1.077 seconds. These numbers are quite close to those obtained with  $\Delta t = 0.01$ , and further point to the effectiveness of the algorithm.

## V. CONCLUSIONS AND FUTURE RESEARCH

This paper concerns the problem of balancing motion-related energy with transmission power in a class of mobile sensor networks. The various sensors, mounted on mobile robots (agents) transmit information while in motion along a given segment, and the problem is to minimize a weighted sum of their energy expenditures. We address this problem in the framework of optimal control and devise an effective computational technique for its solution.

The algorithm that we propose is based on gradient projection with Armijo step sizes. The descent direction from a given control input aims at narrowing the optimality gap defined by the maximum principle. Though having a linear asymptotic convergence rate, this algorithm exhibits rapid descent towards a minimum at its initial stages from a given initial input. Simulation tests on a problem with six agents/sensors have shown to converge to a close proximity of their minimum points in a few seconds of CPU times. These results raise the possibility of on-line implementation of the algorithms and suggest a number of directions for future research, as follows.

First, there is the question of multiple targets and agents' planar motion with obstacles. We expect this to complicate the problem considerably, but its basic structure to yield efficacious extensions of our algorithm. Second, the problem can be extended from the case of stationary objects to the case where the targets move in unpredictable directions. In this case prediction or interpolation may have to be applied in conjunction with the algorithm, and this raises the question of real-time optimization. Third, the algorithm that we propose has a natural structure for decentralized implementation whose investigation will have to consider stability issues of distributed control.

## VI. APPENDIX

This appendix contains the proof of Proposition 2. It requires the following straightforward preliminary result.

*Lemma 1:* Let  $g(\lambda) : R \rightarrow R$  be a twice-continuously differentiable function. Suppose that  $g'(0) \leq 0$ , and there exists  $K > 0$  such that  $|g''(\lambda)| \leq K$  for every  $\lambda \in R$ . Fix  $\alpha \in (0, 1)$ , and define  $\gamma := 2(1 - \alpha)/K$ . Then for every positive  $\lambda \leq \gamma|g'(0)|$ ,

$$g(\lambda) - g(0) \leq \alpha\lambda g'(0). \quad (14)$$

*Proof:* Several variants of this result have been proved in [11] (e.g., Theorem 1.3.7). We provide a proof for this particular version in order to complete the presentation.

Recall the exact second-order approximation of  $C^2$  functions,

$$g(\lambda) = g(0) + \lambda g'(0) + \lambda^2 \int_0^1 (1-s)g''(s\lambda)ds. \quad (15)$$

Using this and the assumption that  $|g''(\cdot)| \leq K$ , we obtain that

$$\begin{aligned} & g(\lambda) - g(0) - \alpha\lambda g'(0) \\ &= (1-\alpha)\lambda g'(0) + \lambda^2 \int_0^1 (1-s)g''(s\lambda)ds \\ &\leq (1-\alpha)\lambda g'(0) + \lambda^2 K/2 \\ &= \lambda((1-\alpha)g'(0) + \lambda K/2). \end{aligned} \quad (16)$$

For every positive  $\lambda \leq \gamma|g'(0)|$ ,  $(1-\alpha)g'(0) + \lambda K/2 \leq 0$ , and hence, and by (16), Equation (14) follows. ■

We next prove Proposition 2.

*Proof:* Fix  $u$  and its associated  $w$ , and recall the definitions of  $\tilde{J}_{u,w}(\lambda)$  and  $\theta(u)$  in Equations (11) and (8), respectively. By Assumption 1, Theorem 5.6.10 in [11] implies that  $\tilde{J}_{u,w}(\lambda)$  is  $C^2$ . For every  $\lambda \in [0, 1]$ , define  $u_\lambda := (1-\lambda)u + \lambda w$ , and let  $x_\lambda$  and  $p_\lambda$  denote the associated state trajectory and costate trajectory defined via (5) and (7), respectively. Standard variational techniques yield that

$$\tilde{J}'_{u,w}(\lambda) = \int_0^{t_f} (H(x_\lambda, w, p_\lambda) - H(x_\lambda, u, p_\lambda))dt, \quad (17)$$

and therefore, and by Theorem 5.6.10 in [11],  $|\tilde{J}''_{u,w}(\lambda)|$  is bounded from above by a constant  $K$  that is independent of  $u$ ,  $w$ , and  $\lambda \in [0, 1]$ .

Since  $u_0 = u$ , we have that  $x_0 = x$  and  $p_0 = p$ , and therefore, and by (17),

$$\tilde{J}'_{u,w}(0) = \theta(u). \quad (18)$$

Consequently, and by Lemma 1, there exists  $\gamma > 0$ , independent of  $u$  or  $w$ , such that, for every positive  $\lambda \leq \gamma|\theta(u)|$ ,

$$\tilde{J}_{u,w}(\lambda) - \tilde{J}_{u,w}(0) \leq \alpha\lambda\theta(u). \quad (19)$$

Suppose, without loss of generality, that  $\beta\gamma|\theta(u)| \leq 1$ . Now (12) and (19) imply that

$$\lambda(u) \geq \beta\gamma|\theta(u)|, \quad (20)$$

and hence, and by Steps 2 and 3 of the algorithm,

$$J(T(u)) - J(u) \leq \alpha\lambda(u)\theta(u) \leq -\beta\gamma\theta(u)^2. \quad (21)$$

This proves the sufficient descent property of the algorithm. ■

*Corollary 1:* If Algorithm 1 computes a sequence of controls,  $u_i$ ,  $i = 1, 2, \dots$ , then  $\lim_{i \rightarrow \infty} \theta(u_i) = 0$ .

*Proof:* Immediate by Proposition 1. ■

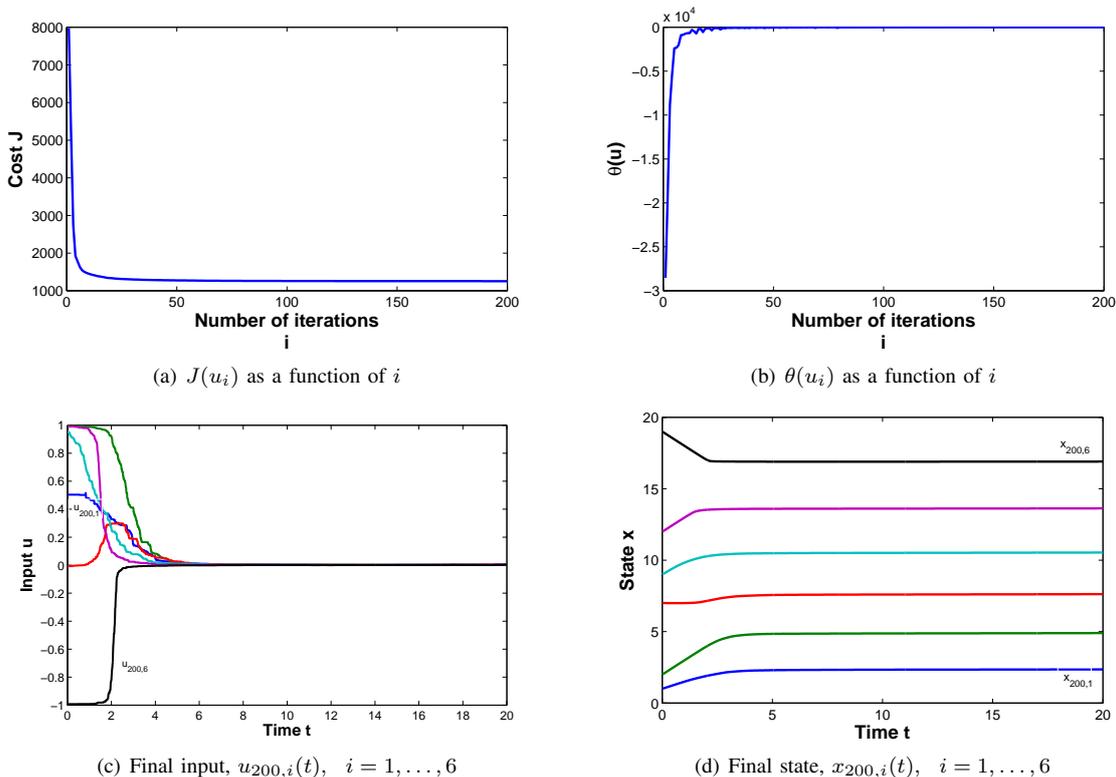


Fig. 2. Results of Algorithm 1: Six agents,  $\Delta t = 0.01$

## REFERENCES

- [1] L. Armijo, "Minimization of Functions Having Lipschitz Continuous First-Partial Derivatives," *Pacific Journal of Mathematics*, Vol. 16, pp. 1-3, 1966.
- [2] A.E. Bryson and Y.C. Ho, *Applied Optimal Control*, Hemisphere Publ. Co., 1975.
- [3] T. M. Caldwell and T. D. Murphey, "Projection-Based Switched System Optimization: Absolute Continuity of the Line Search," in *Proc. 51st CDC*, Maui, Hawaii, December 10-13, 3012.
- [4] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G.S. Sukhatme, "Robomote: Enabling Mobility in Sensor Networks," in *Proc. Fourth Intl. Conf. Information Processing in Sensor Networks (IPSN)*, 2005.
- [5] D.K. Goldenberg, J. Lin, A.S. Morse, B.E. Rosen, and Y.R. Yang, "Towards Mobility as a Network Control Primitive", in *Proc. 5th ACM Intl. Symposium on Mobile Ad Hoc Networking and Computing*, Roppongi Hills, Tokyo, Japan, pp. 163-174, 2004.
- [6] F. El-Moukaddem, E. Tornig, and G. Xing, "Mobile Relay Configuration in Data-Intensive Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, Vol. 12, pp. 261 - 273, 2013.
- [7] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S.S. Sastry, R. Bajcsy, and C. Tomlin, "A Numerical Method for the Optimal Control of Switched Systems", in *Proc. 49th CDC*, Atlanta, Georgia, pp. 7519-7526, December 15-17, 2010.
- [8] The K Team Mobile Robotics, Khepera, <http://www.k-team.com/robots/khepera/index.html>, 2012.
- [9] C. C. Ooi and C. Schindelbauer, "Minimal energy path planning for wireless robots," *Mobile Netw. Appl.*, Vol. 14, No. 3, pp. 309-321, 2009.
- [10] E. Polak and Y. Wardi, "A Study of Minimizing Sequences," *SIAM Journal on Control and Optimization*, Vol. 22, No. 4, pp. 599-609, 1984.
- [11] E. Polak, *Optimization Algorithms and Consistent Approximations*, Springer-Verlag, New York, New York, 1997.
- [12] Y. Yan and Y. Mostofi, "Co-Optimization of Communication and Motion Planning of a Robotic Operation under Resource Constraints and in Fading Environments," *IEEE Transactions on Wireless Communications*, Vol. 12, pp. 1562 - 1572, 2013.
- [13] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," in *Ad Hoc Networks*, vol. 6, issue 4, 2008, pp. 621-655.
- [14] S. Yu and C.S. George Lee, "Mobility and Routing Joint Design for Lifetime Maximization in Mobile Sensor Networks," in *Proc. 2011 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, San Francisco, California, September 25-30, 2011.
- [15] S. Yu, C.S. George Lee, and J. Hu, "Energy Optimal Control in Mobile Sensor Networks Using Hybrid Systems Theory", in *Proc. IEEE Intl. Cont. Systems, Man, and Cybernetics*, Anchorage, Alaska, October 9-12, 2011.