

Motion-Communication Co-optimization with Cooperative Load Transfer in Mobile Robotics: an Optimal Control Perspective[†]

U. Ali,* H. Cai,** Y. Mostofi,** and Y. Wardi*

Abstract—This paper considers the co-optimization of motion and communication in mobile robotic networks operating under energy constraints. More specifically, we consider a team of robots tasked with collectively transmitting a given amount of data to a remote station, while operating in realistic communication environments that experience path loss, shadowing, and multipath fading. We are interested in designing the load distribution, paths, and transmission power/rate schedules of the robots in a way that minimizes the total energy required for motion and communication. We use realistic models to quantify the motion and transmission power. We then show how this multi-agent problem can be efficiently solved using an optimal control framework and mathematically characterize properties of the optimal solution. We further extend the problem to an online adaptation setting where the robots need to keep adjusting their communication and motion decisions (e.g., paths, loads to transfer, transmission rates/power) as more information on the channel quality becomes available during the operation. We show how this problem can be effectively solved in a distributed manner and prove that this online distributed approach provides performance guarantees. Extensive simulations with real channel parameters demonstrate the efficacy of the proposed approach and validate the theoretical results.

I. INTRODUCTION

The area of networked robotics and control has received considerable attention in the past decade, with several major issues such as coordination, control, decentralized decision making, and task coordination heavily addressed in the literature [1], [2]. More recently, the field of communication-aware robotics has started to attract attention [3]–[5]. Leveraging the potential of the framework of networked control systems, communication-aware robotics aims at optimizing a balance between communication and control aspects of robotic systems under energy constraints. More specifically, the two main forms of energy consumption are due to robot motion and communication transmission, and the problem of optimally balancing them has been the focus of recent research [5]–[11]. Ref. [6] considers motion planning for mobile relays to minimize their transmission energy. While it does not explicitly consider motion power, the provided simulation results show that the computed paths come close to optimizing the combined motion/communication energy. Ref. [7] extends the algorithm of [6] to online maximize the lifetime of wireless sensor networks by considering transmission costs as well as motion energy costs. Ref. [8] presents an approximate path planning technique based on Dijkstra’s algorithm for minimizing the combined motion and communication en-

ergy. Ref. [9] proposes an approach for minimizing the motion/communication energy costs in a relay network. First the robots’ trajectories are computed for minimum motion energy, then the transmission schedule is computed to minimize the communication cost. Ref. [10] considers the combined energy minimization in the framework of LQR model predictive control, and develops a distributed algorithm for solving it. A similar approach is followed in [11] for maximizing the lifetime of wireless sensor networks.

All these works use approximate, deterministic disk models (or path-loss models) for characterizing the channel quality. In contrast, Ref. [5] considers a communication-aware problem under realistic channel models for shadowing and multipath fading. They pose the problem in a discrete-time setting, and apply nonlinear programming to solving it. Ref. [12] develops a specialized algorithm for optimal control problems and tests it on a simple communication-aware problem. In this paper, we consider a robotic team tasked with cooperatively transmitting a given amount of data (load) to a remote station while in motion. The robots start from the same initial point, but have individual destinations to go to.

Statement of contributions: Using a realistic channel model, and realistic motion and transmission power models, we show how the robots can efficiently find the optimal load distribution (how much data each robot should send), optimal path, and transmission power/rates along the paths. More specifically, we pose the problem in an optimal control framework and use our algorithm developed in [12]. We mathematically characterize properties of the optimum solution, which shows the coupling between the load distribution, motion decisions, and communication decisions.

We then extend the setting to the case of online adaptation and optimization scenario, where the robots have to adapt their load distribution, and motion and communication decisions as more channel information becomes available. As the robots have an update on the channel quality during the operation, their optimum load distribution and paths/transmission rates may need to change. For instance, if the new channel knowledge reveals that the route of a robot to its destination will experience low quality links, then this robot may have to revise its path and/or reduce its load, by transferring some of its load to its neighbors so it can transmit the remaining load with the required performance guarantee. We then show how this online motion/communication/load co-adaptation problem can be solved efficiently and in a distributed manner, and prove that the proposed distributed solution has guarantees of performance. Simulation results are also provided to explain our approach and validate the theoretical results.

In the conference version of this paper, we only considered a single agent [13]. As such, joint load planning and motion/communication co-optimization for multi-agent

[†]Research supported in part by the NSF under Grant Numbers CNS-1239225, NeTS-1321171, and RI-1619376.

*School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332. Email: usmanali@gatech.edu, ywardi@ece.gatech.edu.

**Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106. Email: hcail@ece.ucsb.edu, ymostofi@ece.ucsb.edu.

network was not considered there, and is dealt with in this paper. Moreover, in this paper we show how to solve the distributed online co-adaptation of load planning, motion, and communication. To the best of our knowledge, no existing literature has addressed multi-robot motion and communication co-optimization and online joint planning in realistic communication environments.

The rest of the paper is organized as follows. Sec. II develops the system model, formulates the problem of interest to this paper, and presents the algorithm to be used in later sections. Sec. III concerns the centralized multi-agent problem and Sec. IV considers the distributed online multi-agent problem. Sec. V concludes the paper.

II. PROBLEM FORMULATION

The motion-transmission power co-optimization problem in mobile robotics concerns the path planning and transmission scheduling of a robotic team, designed (computed) for an optimal balance of transmission power and motion power. This paper casts the problem in the optimal control framework, whose control variables consist of the robots' trajectories and transmission schedules. The performance criterion (cost functional) is a weighted sum of the robots' motion energy and transmission energy. To demonstrate our optimal control framework for solving such motion-communication co-optimization problems, we consider a problem where a team of robots is required to perform coordinated tasks.

Consider the case where a group of agents starts from the same point and are tasked with transmitting a given amount of data collectively to a remote base station within a prescribed time interval. Every agent needs to go to a different destination and thus experiences a different channel quality along its path. Hence the problem is to design the motion/transmission controls for each robot in such a way that the data transmission load is optimally divided among the robots thereby minimizing the joint motion-communication cost, while satisfying the constraints on each robot's position, velocity, and controls.

Suppose we have N identical robots and use subscript i to denote Robot i 's state variables, control variables, and constraint parameters. The motion of each robot along with boundary boundary constraints can be described as follows.

$$\begin{aligned} \dot{x}_{1,i}(t) &= x_{2,i}(t), & x_{1,i}(0) &= S_i, & x_{1,i}(t_f) &= D_i, \\ \dot{x}_{2,i}(t) &= u_i(t), & x_{2,i}(0) &= 0, & x_{2,i}(t_f) &= 0, \end{aligned} \quad (1)$$

where the operation time interval is $[0, t_f]$ for a given $t_f > 0$, $S_i \in \mathcal{R}^2$ is the starting position, $D_i \in \mathcal{R}^2$ is the destination while $x_{1,i}(t)$, $x_{2,i}(t)$, $u_i(t) \in \mathcal{R}^2$ denote the robot's position, velocity, and acceleration, respectively, at time t .

For the energy consumed during the motion, we use the power model derived from first principles in [14] for a DC-powered robot. Given non-negative constants k_1, \dots, k_6 , which depend on the terrain traversed by the robot, the motion power, denoted by $P_{m,i}(t)$, is defined as

$$\begin{aligned} P_{m,i}(t) &= k_1 \|u_i(t)\|^2 + k_2 \|x_{2,i}(t)\|^2 + k_3 \|x_{2,i}(t)\| + k_4 \\ &+ k_5 \|u_i(t)\| + k_6 \|u_i(t)\| \cdot \|x_{2,i}(t)\|. \end{aligned} \quad (2)$$

Next, we model the robot's communication transmission module. Let Q_i denote the required number of bits that Robot i

has to transmit to the remote station and let B denote the communication bandwidth, which is a constant. Denoting by $R_i(t)$ the spectral efficiency of the channel, the relationship between these quantities is given by the following equation.

$$\int_0^{t_f} R_i(t) dt = \frac{Q_i}{B} := c_i, \quad (3)$$

An equivalent representation of this equation, which will be used in the sequel, is obtained by defining $x_{3,i}(t)$ as follows:

$$\dot{x}_{3,i}(t) = R_i(t), \quad x_{3,i}(0) = 0, \quad x_{3,i}(t_f) = c_i. \quad (4)$$

The transmission power, $P_{c,i}(t)$ is modeled using the probabilistic framework of [5] as

$$P_{c,i}(t) = \frac{2^{R_i(t)} - 1}{K} E \left[\frac{1}{\Gamma_{\text{rv}}(x_{1,i}(t))} \right], \quad (5)$$

where K is a constant depending on the minimum acceptable Bit Error Rate (BER), $\Gamma_{\text{rv}}(x_{1,i}(t))$ is a random variable characterizing the Channel-to-Noise Ratio (CNR) at location $x_{1,i}(t)$ and $E[\cdot]$ denotes expectation. The expectation term in Eq. (5) can be computed based on a small number of a priori channel measurements, as described in [15] and summarized in Sec. II-A. We note that in the real-time setting discussed below, such channel measurements are taken periodically and the aggregated results are used to predict the channel quality for a future horizon, as will be explained in the sequel.

Given the motion and transmission models, the cost functional to be minimized is defined as

$$J := \int_0^{t_f} \sum_{i=1}^N (P_{c,i}(t) + \gamma P_{m,i}(t)) dt, \quad (6)$$

for a given constant $\gamma > 0$ and with $P_m(t)$ and $P_c(t)$ defined by Eqs. (2) and (5), respectively. The control variable for each robot is the pair $(u_i(t), R_i(t)) \in \mathcal{R}^2 \times \mathcal{R}^+$, $t \in [0, t_f]$.

As the robots perform a coordinated transmission task, the data sent by each robot must sum up to the total required amount of data, c_{total} , resulting in the following constraint

$$\sum_{i=1}^N c_i = c_{\text{total}}. \quad (7)$$

where c_i as defined in Eq. (3) is the amount of data to be sent by Agent i , that needs to be optimized and we will refer to it as the data load of Agent i . Additionally, there are pointwise constraints on the control variables of the form

$$0 \leq \|u_i(t)\| \leq u_{\text{max}}, \quad 0 \leq R_i(t) \leq R_{\text{max}}, \quad (8)$$

for given $u_{\text{max}}, R_{\text{max}} > 0$. The first problem considered in this paper is the centralized multi-agent optimization problem, which is stated as follows.

Problem 1. *Minimize the cost functional (6) over the interval $[0, t_f]$ subject to the constraints (1), (4), (7), and (8).*

Solving the centralized offline optimization problem 1 relies on the channel quality predicted a priori based on channel measurements collected in advance. During the operation, it is possible for the robots to receive additional channel measurements, which can be utilized to improve the channel prediction. Thus, the second problem considered in this paper is the online multi-agent co-optimization problem. More specifically, the robots obtain additional channel measurements during the operation and use them to update the predicted channel quality, as well as recompute (adapt) the load distribution (c_i 's) and the

motion/transmission controls to minimize the cost-to-go from a given time $t_0 \in (0, t_f]$ to the final time t_f . The online multi-agent co-optimization problem can then be stated as follows.

Problem 2. *Minimize the cost functional (6) over the entire horizon $[t_0, t_f]$ subject to the constraints (1), (4), (7) and (8) with updated boundary conditions, every few seconds.*

Before we set out to solve these two problems, we first briefly describe the online channel prediction framework and the optimal control algorithm employed to solve them.

A. Online Channel Prediction

In order to plan the trajectory and communication parameters, the robots need to assess the required transmission power at any unvisited location, which depends on the channel quality, as shown in Eq. (5). In this section, we briefly summarize our earlier results [15] on channel prediction, which will be utilized for the rest of this paper. Assuming that the robots adopt the common MQAM modulation when communicating to the remote station, the required transmission power at time t can be characterized by the following equation [16]:¹

$$P_{c, \text{actual}}(t) = \frac{2^{R(t)} - 1}{K\Gamma(x_1(t))}, \quad (9)$$

where $K = -1.5/\ln(5p_{b,\text{th}})$, $p_{b,\text{th}}$ is the desired Bit Error Rate (BER) threshold at the receiver, $R(t)$ is the spectral efficiency at time t and $\Gamma(x_1(t))$ is the instantaneous Channel-to-Noise Ratio (CNR) at location $x_1(t)$. It has been shown in the literature that the CNR can be modeled as a random process with three major components: path loss, shadowing, and multipath fading [16]. As shown in [15], based on a few a priori channel measurements, the CNR (in dB) at an unvisited location q can be best characterized by a Gaussian random variable, $\Gamma_{\text{rv, dB}}(q)$, whose mean and variance are given by

$$\begin{aligned} \bar{\Gamma}_{\text{rv, dB}}(q) &= H_q \hat{v} + \psi^T(q) \omega^{-1} (Y - H_{\mathcal{Q}} \hat{v}), \\ \Sigma(q) &= \hat{\xi}_{\text{dB}}^2 + \hat{\rho}_{\text{dB}}^2 - \psi^T(q) \omega^{-1} \psi(q), \end{aligned}$$

where Y is the stacked vector of m a priori collected CNR measurements, $\mathcal{Q} = \{q_1, \dots, q_m\}$ denotes the measurement locations, $H_q = [1 - 10 \log_{10}(\|q - q_b\|)]$, q_b is the location of the remote station, $H_{\mathcal{Q}} = [H_{q_1}^T \dots H_{q_m}^T]^T$, $\omega = \Omega + \hat{\rho}_{\text{dB}}^2 I_m$ with $[\Omega]_{i,j} = \hat{\xi}_{\text{dB}}^2 \exp(-\|q_i - q_j\|/\hat{\eta})$, for $i, j \in \{1, \dots, m\}$, and $\psi(q) = [\hat{\xi}_{\text{dB}}^2 \exp(-\|q - q_1\|/\hat{\eta}) \dots \hat{\xi}_{\text{dB}}^2 \exp(-\|q - q_m\|/\hat{\eta})]^T$. The variables $\hat{v} = [\hat{K}_{\text{PL}} \hat{n}_{\text{PL}}]^T$, $\hat{\xi}_{\text{dB}}$, $\hat{\eta}$, and $\hat{\rho}_{\text{dB}}$ are the estimated channel parameters based on the a priori channel measurements. See [15] for more details on channel parameter estimation and the channel prediction performance. Based on this framework, the CNR at an unvisited location $x_1(t)$ can be characterized as a lognormal random variable. The expected transmission power $P_c(t)$ is then given by

$$P_c(t) = \frac{2^{R(t)} - 1}{K} E \left[\frac{1}{\Gamma_{\text{rv}}(x_1(t))} \right]. \quad (10)$$

Note that for the lognormally-distributed $\Gamma_{\text{rv}}(x_1(t))$, we have

$$E \left[\frac{1}{\Gamma_{\text{rv}}(x_1(t))} \right] = \exp \left(\frac{(\ln 10)^2 \Sigma(x_1(t))}{200} \right) \frac{1}{\bar{\Gamma}_{\text{rv}}(x_1(t))}, \quad (11)$$

where $\bar{\Gamma}_{\text{rv}}(x_1(t)) = 10^{\bar{\Gamma}_{\text{rv, dB}}(x_1(t))/10}$. Eq. (11) provides a predicted metric of the channel quality at $x_1(t)$ and we let

¹We drop the subscript i in this section for conciseness. The computation of transmission power is associated with a single robot.

$s(x_1(t)) = E[1/\Gamma_{\text{rv}}(x_1(t))]$. Note that $s(x_1(t))$ can be computed for any unvisited location x_1 . Thus, we can compute the required transmission power (Eq. (10)) for any point in the environment, which can then be used in the optimization.

This framework is also applicable to the setting where the channel prediction is updated as additional channel measurements become available to the robot. For instance, the robot may have a few a priori channel measurements (e.g., by static sensors in the field), based on which it computes an initial prediction of the channel quality over the workspace. The robot then travels along the path obtained from minimizing J as defined in Problem 1 with this initial channel prediction. As the robot moves, it is provided with additional channel measurements (e.g., by gathering more samples along its path, through crowdsourcing and/or by other robots in the field), which enables it to improve the prediction accuracy of the channel quality. This allows for online control and path planning, which we shall see in Sec. IV.

B. Specialized Optimal Control Algorithm

Optimization algorithms typically are based on two computed objects at a given iteration: a direction (e.g., of descent), and a step size along it. Recently, we have developed an algorithm [12] suitable for a class of power-aware optimal control problems [13]. Cumulative experience with it reveals some favorable computational properties including fast approach towards a local minimum. This does not mean fast asymptotic convergence, which characterizes an algorithm's behavior close to a local minimum, but rather large strides towards a region of a (local) minimum. A key innovation in the algorithm is its choice of a descent direction, which is not based on explicit gradient descent but rather follows an alternative approach which, for a class of problems, requires fewer computations. For completeness, the algorithm in [12] is included in this paper and we next briefly explain its structure.

Consider the abstract Bolza optimal control problem where the system's dynamics are defined by the equation.

$$\dot{x} = f(x, u), \quad x(0) := x_0,$$

where $x \in \mathcal{R}^n$, $u \in \mathcal{R}^k$, and $f: \mathcal{R}^n \times \mathcal{R}^k \rightarrow \mathcal{R}^n$ is Lipschitz continuous in x and continuous in u . Given a final time $t_f > 0$, a cost function $L: \mathcal{R}^n \times \mathcal{R}^k \rightarrow \mathcal{R}$, and a terminal-state cost function $\phi: \mathcal{R}^n \rightarrow \mathcal{R}$, define the cost functional as

$$J := \int_0^{t_f} L(x, u) dt + \phi(x(t_f)).$$

The optimal control problem considered is to minimize J subject to the pointwise constraints $u(t) \in \mathcal{U}$, where $\mathcal{U} \subset \mathcal{R}^k$ is an input constraint set. We make the following assumptions on the considered optimal control problem, under which our optimal control algorithm is designed.

Assumptions. *The function $f(x, u)$ is affine in $u \in \mathcal{U}$ for every $x \in \mathcal{R}^n$, and the function $L(x, u)$ is convex in $u \in \mathcal{U}$ for every $x \in \mathcal{R}^n$. The set \mathcal{U} is compact and convex.*

Let $p(t)$, for $t \in [0, t_f]$, denote the costate (adjoint) trajectory defined by the equation

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

with the boundary condition $p(t_f) = \nabla\phi(x(t_f))$, and let

$$H(x, u, p) := p^T f(x, u) + L(x, u)$$

denote the Hamiltonian function (see, e.g., [17]). The kind of problems for which this algorithm (outlined below) is suitable have the property that, for given $x \in \mathcal{R}^n$ and $p \in \mathcal{R}^n$, a minimum value of the Hamiltonian $H(x, w, p)$, over $w \in \mathcal{U}$, can be computed via a simple, explicit formula.

Regarding the implementation, we assume that a finite grid will be used. The computations of the state trajectory, costate trajectory, and various integrals are assumed to be performed via numerical approximations. We refer to the control function $u(t)$, $t \in [0, t_f]$, by the boldface notation \mathbf{u} . We note that Algorithm 1 is an iterative method. The steps below are performed in each iteration until the update step size is smaller than a prescribed value.

Algorithm 1: Specialized Optimal Control Algorithm

Parameters: Constants $\alpha \in (0, 1)$ and $\beta \in (0, 1)$.

Given a control \mathbf{u} , compute the next control \mathbf{u}_{next} as follows:

Step 1 (*Direction from \mathbf{u}*): Compute the state and costate trajectories $x(t)$ and $p(t)$, $t \in [0, t_f]$. For every $t \in [0, t_f]$, compute a pointwise (t -dependent) minimizer of the Hamiltonian $u^*(t) \in \mathcal{U}$ satisfying

$$u^*(t) \in \operatorname{argmin}_{w \in \mathcal{U}} \left\{ H(x(t), w, p(t)) \right\}.$$

Define \mathbf{u}^* to be the function $u^*(t)$, $t \in [0, t_f]$.² Define the direction from \mathbf{u} to be $d(t) := u^*(t) - u(t)$, namely, in functional notation, $\mathbf{d} = \mathbf{u}^* - \mathbf{u}$.

Step 2 (*Step size along the direction \mathbf{d}*): Define

$$\theta(\mathbf{u}) = \int_0^{t_f} (H(x(t), u^*(t), p(t)) - H(x(t), u(t), p(t))) dt.$$

Compute $\Delta k \in \{0, 1, \dots\}$ defined as

$$\Delta k = \min\{j = 0, 1, \dots \mid J(\mathbf{u} + \beta^j \mathbf{d}) - J(\mathbf{u}) \leq \alpha \beta^j \theta(\mathbf{u})\},$$

and set the step size, $\lambda = \beta^{\Delta k}$.

Step 3 (*Update*): Set \mathbf{u}_{next} to be $\mathbf{u}_{\text{next}} = \mathbf{u} + \lambda \mathbf{d}$.

As we pointed out, the main innovation of the algorithm is in the choice of the direction in Step 1, while Step 2 describes a standard Armijo step size; See [18] for details.

III. MULTI-AGENT MOTION-COMMUNICATION CO-OPTIMIZATION

Consider the multi-agent motion-communication optimal control problem (Problem 1). To put the problem in a form that is more amenable to Algorithm 1, we treat the terminal constraints via a quadratic penalty function. We append the terminal constraints in Eq. (1) and (7) to the cost J in Problem 1 via penalty terms $C_{1,i}$, $C_{2,i}$, and C_3 , $\forall i = \{1, \dots, N\}$, which results in the cost functional

$$J_c := \sum_{i=1}^N \left(\int_0^{t_f} (P_{c,i}(t) + \gamma P_{m,i}(t)) dt + C_{1,i} \|x_{1,i}(t_f) - D_i\|^2 + C_{2,i} \|x_{2,i}(t_f)\|^2 \right) + C_3 \left\| \sum_{i=1}^N x_{3,i}(t_f) - c \right\|^2. \quad (12)$$

The problem then becomes minimizing J_c , subject to the dynamical constraints and the pointwise constraints on the

²There may arise measurability issues due to the explicit characterization of $u^*(t)$ for all t in the uncountable set $[0, t_f]$. However, in a grid-based implementation, these issues will be avoided since t will lie in a finite set.

control inputs as described in Problem 1. We use Algorithm 1 to solve this multi-agent optimal control problem. For each agent i , if we let $p_i = p_i(t) := (p_{1,i}(t), p_{2,i}(t), p_{3,i}(t)) \in \mathcal{R}^2 \times \mathcal{R}^2 \times \mathcal{R}$ denote the costate associated with each agent, then the costate (adjoint) equations are given by

$$\begin{aligned} \dot{p}_{1,i}(t) &= -\frac{2^{R_i} - 1}{K} \frac{\partial s(x_{1,i})}{\partial x_{1,i}}, \\ \dot{p}_{2,i}(t) &= -p_{1,i} - \gamma \left(2k_2 x_{2,i} + k_3 \frac{x_{2,i}}{\|x_{2,i}\|} + k_6 \|u_i\| \frac{x_{2,i}}{\|x_{2,i}\|} \right), \\ \dot{p}_{3,i}(t) &= 0, \end{aligned}$$

with boundary conditions

$$\begin{aligned} p_{1,i}(t_f) &= 2C_{1,i}(x_{1,i}(t_f) - D_i), \\ p_{2,i}(t_f) &= 2C_{2,i}x_{2,i}(t_f), \\ p_{3,i}(t_f) &= 2C_3 \left(\sum_{j=1}^N x_{3,j}(t_f) - c \right), \end{aligned} \quad (13)$$

for $i \in \{1, \dots, N\}$. Furthermore, the Hamiltonian is

$$H(x, [u, R], p) = \sum_{i=1}^N (p_{1,i}^T x_{1,i} + p_{2,i}^T u_i + p_{3,i} R_i + \frac{2^{R_i} - 1}{K} s(x_{1,i}) + \gamma (k_1 \|u_i\|^2 + k_2 \|x_{2,i}\|^2 + k_3 \|x_{2,i}\| + k_4 + k_5 \|u_i\| + k_6 \|u_i\| \|x_{2,i}\|)). \quad (14)$$

When the initial and final velocities are the same, then we have $k_5 = k_6 = 0$ in Eq. (2) and (14) (see [14]). The minimizer of this Hamiltonian, subject to the constraints on the control variables, can be seen after some algebra to be as follows.

$$u_i^* = \begin{cases} -\frac{p_{2,i}}{2\gamma k_1}, & \text{if } \frac{1}{2\gamma k_1} \|p_{2,i}\| \leq u_{\max} \\ -\frac{p_{2,i}}{\|p_{2,i}\|} u_{\max}, & \text{if } \frac{1}{2\gamma k_1} \|p_{2,i}\| > u_{\max}, \end{cases}$$

$$R_i^* = \begin{cases} \frac{1}{\ln(2)} \ln \left(\frac{-p_{3,i} K}{\ln(2) s(x_{1,i})} \right), & \text{if } p_{3,i} \leq -\frac{(\ln(2) s(x_{1,i}))}{K} \\ R_{\max}, & \text{if } \frac{1}{\ln(2)} \ln \left(\frac{-p_{3,i} K}{\ln(2) s(x_{1,i})} \right) > R_{\max} \\ 0, & \text{otherwise.} \end{cases}$$

Based on these expressions, our first result then reveals the relationship between the spectral efficiency and the channel quality. For analysis purposes, we assume no maximum spectral efficiency and assume that each agent needs to transmit a non-zero amount of data all the time.

Theorem 1. For Agent i and for all $i = 1, \dots, N$ along the optimal path, the relationship between the optimal spectral efficiency $R_i^*(t)$ and the channel quality $s(x_{1,i}(t))$ is given by

$$R_i(t)^* = a - b \ln(s(x_{1,i}(t))), \quad (15)$$

for some positive constants a and b .

Proof. From the expression of the minimizer of the Hamiltonian, we have when $p_{3,i}(t) \leq -\ln(2)s(x_{1,i}(t))/K$,

$$R_i(t)^* = \frac{1}{\ln(2)} \ln \left(\frac{-p_{3,i}(t) K}{\ln(2) s(x_{1,i}(t))} \right),$$

where

$$\dot{p}_{3,i}(t) = 0, \quad p_{3,i}(t_f) = 2C_3 \left(\sum_{j=1}^N x_{3,j}(t_f) - c \right).$$

³Since $s(x_{1,i})$ is continuously differentiable in the spatial variables, its partial derivatives exist and can be computed both in the x and y directions.

Along the optimal path and under the assumption that $R_i^*(t) > 0$ for all $t \in [0, t_f]$, the product $C_3(\sum_{j=1}^N x_{3,j}(t_f) - c)$ is some negative number, and represents the Lagrange multiplier in the limit as $C_3 \rightarrow \infty$. Since $\dot{p}_{3,i}(t) = 0$, the costate variable $p_{3,i}(t)$ is then constant for all $t \in [0, t_f]$ and equal to $\mu = 2C_3(\sum_{j=1}^N x_{3,j}(t_f) - c)$ along the entire optimal trajectory. $R_i(t)^*$ can then be expressed as

$$R_i(t)^* = a - b \ln(s(x_{1,i}(t))),$$

where $a = 1/\ln(2)\ln(\mu K/\ln(2))$ and $b = 1/\ln(2)$. \square

We next present application examples to demonstrate the application of our framework and validate the relationships presented in the above theorem.

A. Numerical Results

In this section, we show the performance of the proposed approach in a realistic simulation environment. We first evaluate the case where there is only one agent to analyze the optimization results in details. In the single-agent case, we solve Problem (12) with $N = 1$ and the single agent is responsible for transmitting all the required data. This is then followed by a case with multiple agents. We also provide numerical results that confirm the validity of Theorems 1. We note that all the differential equations are solved using the Trapezoidal method of integration.

1) *Single-Agent Case:* Suppose that there is a single agent that needs to transmit 160 bits/Hz to a remote station located at $x_b = (5, 5)$, while moving from the initial point $S = (35, 50)$ to the destination $D = (45, 25)$. The operation comprised of both motion and transmission has to be completed in 40 seconds. The motion parameters are $k_1 = 5.47$, $k_2 = 0.77$, $k_3 = 10.10$, and $k_4 = 4.24$ [14]. The agent's maximum acceleration and spectral efficiency are 1 m/s² and 6 bits/sec/Hz, respectively. The simulated channel parameters (estimated from real wireless measurements [15]) are $K_{PL} = -41.34$, $n_{PL} = 3.86$, $\xi_{dB} = 3.20$, $\eta = 3.09$ m and $\rho_{dB} = 1.64$. The receiver thermal noise is -110 dBm and the BER threshold is $p_{b,th} = 2 \times 10^{-6}$. The simulated channel is then predicted based on 500 (0.2%) random samples in the field by applying the framework summarized in Sec. II-A.

The balancing factor in Eq. (6) is set to $\gamma = 0.5$. The Armijo parameters in Algorithm 1 are set to $\alpha = \beta = 0.5$. The penalty coefficients in Eq. (12) are set to $C_1 = 5$, $C_2 = 2$, and $C_3 = 5$. The initial controls are set to $u(t) = 0$ and $R(t) = 0$ for all $t \in [0, t_f]$. The numerical integrations performed by Algorithm 1 has an integration time step size of $\Delta t = 0.05$ seconds. The algorithm is run until Δk in Algorithm 1 becomes greater than 50, indicating a descent step size in the order of 2^{-50} .

Results of a typical simulation are shown in Figs. 1-4. Fig. 1 shows the plot of the cost value J_c in Problem (12) w.r.t. the iteration number. The cost is reduced from the initial value of 1.3171×10^5 to the final value of 1.0937×10^3 in 870 iterations. However, the major portion of cost reduction occurs within the first 50 iterations, at which the cost is 6.0065×10^3 . Fig. 1 also shows a zoomed-in view of the tail of the cost, confirming that there is no significant decrease in the cost after the first 100 iterations. The algorithm was implemented on an Intel dual-core computer with an i5 processor running at 2.7 GHz, and it took 33.87 seconds of CPU time for the 870

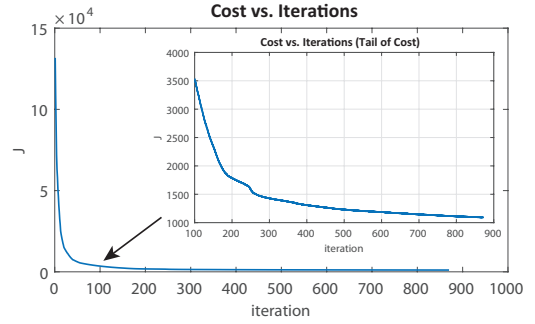


Fig. 1: Cost w.r.t. the iteration's index. Most of the cost reduction occurs during the first few iterations. The tail of cost after the 100th iteration is zoomed in for better visualization.

iterations. The fact that much of the cost reduction happens during the first few iterations allows for using Algorithm 1 in an online setting, which will be discussed in Sec. IV.

After 870 iterations, the total motion and communication cost in Problem 1 is $J = 1.0668 \times 10^3$ (not including the penalty terms). The final values of the state variables are $x_1(t_f) = (43.76, 25.3)$, $x_2(t_f) = (2.71, 0.1777)$, and $x_3(t_f) = 159.1$. We note a minor discrepancy from the desired final state values of $x_1(t_f) = (45, 25)$, $x_2(t_f) = (0, 0)$, and $x_3(t_f) = 160$. The accuracy can be improved by choosing larger values of C_1 , C_2 , and C_3 , and a smaller time step size Δt , at the expense of an increased CPU time.

The channel quality $s(x_1) := E[1/\Gamma_{rv}(x_1)]$ over the workspace and the path taken by the agent corresponding to the optimal control are shown in Fig. 2. Smaller values (corresponding to darker blue color) on the z-axis indicate a better channel quality as measured by $s(x_1)$. The agent's starting and ending points are marked by a brown square and a cyan disk, respectively. As can be seen, the agent avoids the peaks (bad channel quality regions) and steers towards regions with a relatively good channel quality (junction near point (30, 20)), before taking the final turn towards the goal.

The magnitude and direction of the agent's acceleration at different points along the path, as well as the time evolution along the path, are shown in Fig. 3. The small vectors near the junction point (30, 20) indicates that the robot slows down in this region to transmit the data, taking advantage of the good channel quality. This is also demonstrated by the blue time curve. Near the location with a good channel quality (marked by the vertical dashed arrow), the time increases while the robot's position remains almost constant, indicating a close to zero velocity. The large deceleration at the end in Fig. 3 indicates that the robot makes a stop at the destination.

Fig. 4 shows the robot's spectral efficiency and the predicted channel quality along the optimal path. There is a significant correlation between the robots transmission spectral efficiency and the predicted channel quality, as can be seen in Fig. 4 (Top) and (Middle). The negative log of the predicted channel quality metric $s(x_1)$ is plotted in Fig. 4 (Bottom). As shown in Theorem 1, the robot's optimal spectral efficiency is an affine function of the negative log of $s(x_1)$. This can be verified by comparing the 1st and 3rd subplots of Fig. 4.⁴

⁴The obtained spectral efficiency is almost but not exactly an affine function of $-\ln(s(x_1))$ as indicated by Theorem 1 since the simulation is performed with a discretized time step of 0.05 seconds rather than in continuous time.

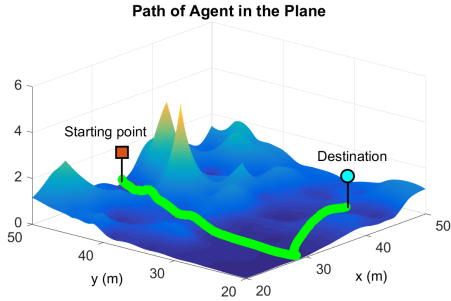


Fig. 2: The green curve represents the path followed by the robot, veering towards regions with better channel qualities. The brown square and the cyan disk represent the starting and end points, respectively. Smaller values on the z-axis indicate a higher predicted channel quality measured by $s(x_1)$. Note that the actual path is in the plane and the path is plotted in 3D for better visualization. Readers are referred to the color PDF for optimal viewing.

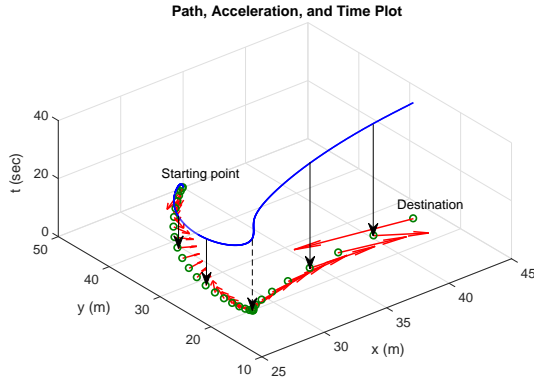


Fig. 3: The green circles represent the path followed by the robot. The red arrows represent the acceleration of the robot along its path. The evolution of mission time w.r.t. the robot path is shown on the z-axis. It can be seen that the acceleration is small near the junction of (30, 20), where the robot slows down to take advantage of the good channel quality for data transmission. Also, the robot position does not change much as time increases near the vertical dashed arrow, indicating a close to zero velocity at this point.

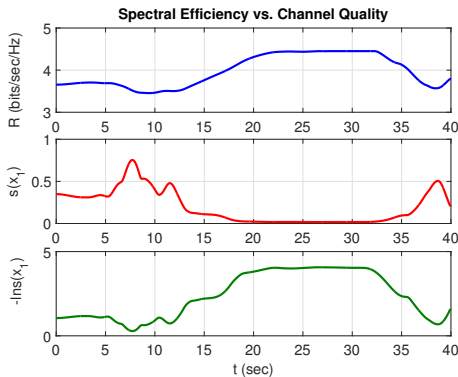


Fig. 4: (Top) The robot's spectral efficiency; (Middle) The predicted channel quality metric; (Bottom) The negative log of the predicted channel quality metric for a better comparison with the spectral efficiency. Overall, the figure confirms the results of Theorem 1.

2) *Multi-Agent Case*: To demonstrate the application of our approach to the multi-agent motion-communication co-optimization, we present an example of 6 agents starting at the same point (25, 30) and each having a different destination as shown in Fig. 5. The total amount of data to be sent is 600 bits/Hz and the maximum spectral efficiency of each agent is $R_{max} = 6$ bits/sec/Hz. The time horizon is 40 seconds. Thus,

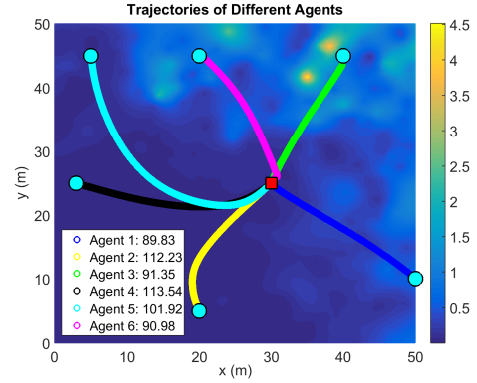


Fig. 5: Paths of the six robots starting from the same starting point and going towards different goals, obtained from solving the multi-agent optimal control problem (12). The red square denotes the common starting point and the cyan disks denote the agents' respective destinations. The optimal data assigned to each agent is annotated in the legend. Smaller values in the color scheme represent a higher predicted channel quality, as measured by $s(x_1)$. Readers are referred to the color PDF for optimal viewing.

Agent	Assigned Data	Ave. of $-\ln s(x_1)$ along Path
1	89.83	2.17
2	112.23	4.00
3	91.35	2.18
4	113.54	3.97
5	101.92	3.33
6	90.98	2.23

TABLE I: Assigned optimal amount of data and the predicated channel quality (in terms of $-\ln s(x_1)$) averaged over the optimal path for each agent.

there does not exist an agent which can send all the data in the given time horizon by itself. The maximum acceleration for each agent is $u_{max} = 1$ m/s². The channel parameters are the same as in the single-agent example of Section III-A1.

With the minimizers to the Hamiltonian available, we solve Problem (12) using Algorithm 1 described in Sec. II-B to compute the optimal controls. The resulting paths of the optimization are shown in Fig. 5, where the red square denotes the common starting point and the cyan disks denote the respective destinations. Figs. 6 shows the optimal spectral efficiency and the negative log of the predicted channel quality metric ($-\ln s(x_{1,i})$) along the optimal path for each agent. It can be seen that there is a significant correlation between the spectral efficiency and the predicted channel quality metric for each robot, as indicated by Theorem 1. Along the optimal path, the spectral efficiencies are higher in regions of good channel (higher values of $-\ln s(x_1)$) and vice versa.

The optimal data distribution among the agents in bits/Hz for the given application example is $c_1 = 89.83$, $c_2 = 112.23$, $c_3 = 91.35$, $c_4 = 113.54$, $c_5 = 101.92$, and $c_6 = 90.98$, respectively, summing up to 599.85. To better understand the data distribution, we compare the assigned amount of data with the average negative log of the predicted channel quality metric along the optimal path for each agent in Table I. It can be seen that for Agents 2 and 4, which have been assigned the largest amounts of data, their average predicted channel qualities (in terms of $-\ln s(x_1)$) along the optimal paths are the best among all agents. For Agents 1, 3, and 6, which have been assigned the least amounts of data, their average predicted channel qualities are the worst. For Agent 5, its

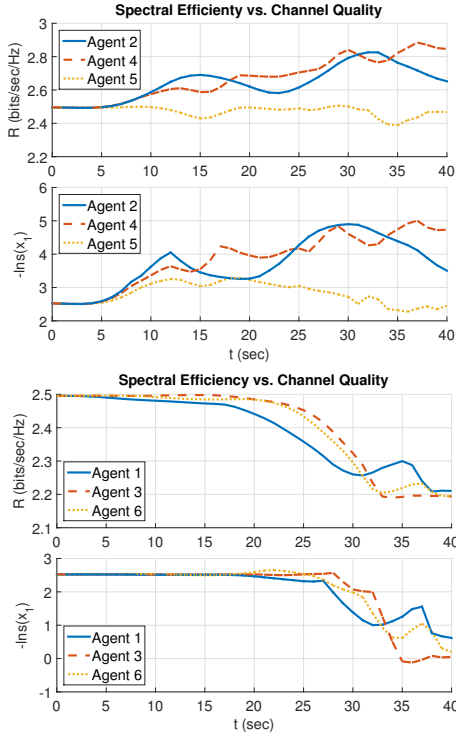


Fig. 6: The spectral efficiencies and the negative log of the predicted channel quality along the optimal paths. The top figure shows the results for Agents 2, 4, and 5. The bottom figure shows the results for Agents 1, 3, and 6. The results validate Theorem 1.

predicted channel quality is worse than Agents 2 and 4, but better than Agents 1, 3, and 6. As indicated by Theorem 1, if an agent experiences a good channel quality, its optimal spectral efficiency will be high and thus, it will be assigned more data. The observations from Table I are consistent with the results shown in Theorem 1.

IV. DISTRIBUTED ONLINE MULTI-AGENT MOTION-COMMUNICATION CO-OPTIMIZATION

During the operation, additional channel measurements may become available to the agents (e.g., collected on their paths, provided by crowd-sourcing and/or communicated by other agents in the field), which can be utilized to increase the accuracy of the channel prediction. With a better predicted channel, the agents then need to adapt their previously-computed motion/transmission controls to reduce the overall energy consumption. More specifically, they need to redistribute the remaining load among themselves, revise their paths, and decide on the new optimum communication transmission rate/power along their paths.

One possible way of online adaptation is to solve the original centralized problem (12) every time. This approach, however, is suitable only initially when the agents are co-located in a depot for example, while distributed strategies are more suitable for online adaptation. Also, from the computational standpoint, although we have shown fast cost reduction w.r.t. the number of iterations of Algorithm 1, solving the entire original centralized problem in a real-time fashion, every few seconds, is not viable.

In this section, we propose a distributed online strategy for redistributing the agents' communication loads, and updating the motion/transmission controls, given new channel

measurements. For this, we first mathematically relate the optimal data load difference of any two agents to channel qualities over their respective optimal paths which allows the agents to solve for the new optimum data loads by solving a system of linear equations and these results are captured in Theorem 2 below. We then show how the agents can achieve the new optimum load distribution in a distributed manner, and further update their paths. The proposed distributed algorithm only requires each agent to solve a single-agent problem, as opposed to the N -agent centralized one. This drastically reduces the computation time by at least $(100 - 100/N)\%$ as the per-iteration computation time scales linearly with N .

Theorem 2. For any two agents $i, j \in \{1, \dots, N\}$, we have

$$R_i^*(t) - R_j^*(t) = b \ln \left(\frac{s(x_{1,j}(t))}{s(x_{1,i}(t))} \right) \quad (16)$$

and

$$c_i^* - c_j^* = b \int_0^{t_f} \ln \left(\frac{s(x_{1,j}(t))}{s(x_{1,i}(t))} \right) dt, \quad (17)$$

where $b = 1/\ln(2)$, $R_i^*(t)$ and $R_j^*(t)$ are the two agents' respective optimal spectral efficiencies at time t , $x_{1,i}(t)$ and $x_{1,j}(t)$ are points on the two agents' respective optimal paths, and c_i^* and c_j^* are the two agents' respective optimal loads.

Proof. From the costate equations for the multi-agent case, we see that, for Agent k , with $k \in \{1, \dots, N\}$,

$$\dot{p}_{3,k}(t) = 0, \quad p_{3,k}(t_f) = 2C_3 \left(\sum_{j=1}^N x_{3,j}(t_f) - c_{\text{total}} \right),$$

where c is the total amount of data to be sent. It can be seen that $p_{3,k}(t_f)$ is the same for each agent and $\dot{p}_{3,k}(t)$ being zero implies that $p_{3,k}(t)$ is constant over $[0, t_f]$, allowing for the use of Eq. (15) from Theorem 1. Thus, we have $R_i^*(t) = a - b \ln(s(x_{1,i}(t)))$ and $R_j^*(t) = a - b \ln(s(x_{1,j}(t)))$. Taking the difference between $R_i^*(t)$ and $R_j^*(t)$ results in Eq. (16). Integrating $R_i^*(t) - R_j^*(t)$ over $[0, t_f]$ gives the difference in data loads between Agents i and j in Eq. (17). \square

Corollary 1. Given the predicted channel quality along the optimal paths, the optimal data loads of the agents can be obtained from the solution to the following N linearly independent equations:

$$\sum_{i=1}^N c_i^* = c_{\text{total}},$$

$$c_i^* - c_j^* = b \int_0^{t_f} \ln \left(\frac{s(x_{1,i}(t))}{s(x_{1,j}(t))} \right) dt, \quad \forall i \in \{2, \dots, N\}.$$

Proof. In Theorem 2, Eq. (17) provides $N - 1$ independent linear equations on the optimal loads c_i^* . Together with the data constraint $\sum_{i=1}^N c_i^* = c_{\text{total}}$, we have N independent linear equations, which uniquely determine c_i^* , $\forall i \in \{1, \dots, N\}$. \square

Corollary 2. Suppose that at time $t_0 \in [0, t_f]$, the predicted channel quality is updated, which is denoted by $s_{\text{new}}(x_1)$. Given the paths, then the new optimal load for each agent c_i^+ can be computed as follows.

$$c_i^+ = \frac{c_{\text{total}, t_0}}{N} + \left(S_i - \frac{1}{N} \sum_{j=1}^N S_j \right), \quad \forall i = 1, \dots, N, \quad (18)$$

where c_{total, t_0} is the total remaining data to be sent at t_0 and $S_i = b \int_{t_0}^{t_f} -\ln(s_{\text{new}}(x_{1,i}(t))) dt$.

Proof. Give the updated channel prediction $s_{\text{new}}(x_1)$ and assuming that the initial paths are still optimal, we have the following equation for the new optimal loads of two agents i and j based on Eq. (17) of Theorem 2.

$$c_i^+ - c_j^+ = b \int_{t_0}^{t_f} \ln \left(\frac{s_{\text{new}}(x_{1,j}(t))}{s_{\text{new}}(x_{1,i}(t))} \right) dt. \quad (19)$$

A set of linear equations similar to those in Corollary 1 can be formed, where the new individual loads sum up to c_{total,t_0} and the pairwise load difference is given by Eq. (19). It can be easily verified that Eq. (18) provides the solution to these linear equations. Thus, Eq. (18) defines the new optimal loads. \square

From Corollary 2, it can be seen that to compute the new optimal loads, each agent needs to know $c_{\text{total},t_0}/N$ and $\sum_{j=1}^N S_j/N$. The agents can efficiently solve for these two parameters in a distributed manner by using average consensus methods [19] or by utilizing a simple flooding algorithm [20], both of which have been heavily studied in the literature. Once the agents have computed their new optimum loads (c_i^+), they need to redistribute the data among themselves to achieve the new optimum. In Algorithm 2, we show how the agents can achieve this in an efficient distributed manner, by only communicating to their neighbors. Our approach can be considered as a modified version of a general load balancing algorithm based on the literature [19], [21], where we have incorporated the updated channel information (encoded in the parameters c_i^+) in the load transfer operations.

Algorithm 2: Distributed Load Transfer (adapted from [19])

Parameters: c_i^+ is the target load for Agent i and $c_i(k)$ is Agent i 's load at iteration k .

For each agent $i \in \{1, \dots, N\}$, repeat the following steps.

Step 1: Agent i broadcasts its $c_i(k) - c_i^+$ to all its neighbors.

Step 2: Agent i finds $j = \operatorname{argmin}_{q \in Q_i} c_q(k) - c_q^+$, where

$Q_i = \{q \mid c_q(k) - c_q^+ \leq c_i(k) - c_i^+\} \cap \mathcal{N}_i$ and \mathcal{N}_i is the set of Agent i 's neighbors. If $Q_i \neq \emptyset$, Agent i makes an offer of $(c_i(k) - c_i^+ - c_j(k) + c_j^+)/3$ to Agent j .

Step 3: Agent i finds $j = \operatorname{argmax}_{o \in O_i} c_o(k) - c_o^+$, where O_i is the set of agents making an offer to Agent i . If $O_i \neq \emptyset$, Agent i asks Agent j to transfer the offered load and rejects all the other offers.

Step 4: If Agent i accepts Agent j 's offer, then Agent j sends the offered load to Agent i .

The following proposition shows that Algorithm 2 converges to the new optimal loads, and further describes the convergence behavior w.r.t. the number of iterations.

Proposition 1. Consider Algorithm 3 below that is running over a connected graph. Define $V(k) = \sum_{i=1}^N (c_i(k) - c_i^+)^2$. Then there exists a constant μ such that given $\varepsilon > 0$,

$$V(k) \leq \varepsilon V(0), \quad \forall k \geq -\mu N^2 \ln(\varepsilon).$$

Proof. Consider a new state variable for each agent i , $x_i(k) = c_i(k) - c_i^+$. It can be easily verified that Algorithm 3 performs the balancing algorithm of Nedic et al. [19] on $x_i(k)$, $\forall i \in \{1, \dots, N\}$. By Theorem 12 of [19], $\sum_{i=1}^N (x_i(k) - \bar{x})^2 \leq \varepsilon \sum_{i=1}^N (x_i(0) - \bar{x})^2$, $\forall k \geq \mu N^2 \ln(1/\varepsilon)$, where \bar{x} is the mean of

$x_i(k)$. By substituting $x_i(k) = c_i(k) - c_i^+$ and $\bar{x} = 0$ into the inequality, we obtain the result. \square

We next summarize the steps of the distributed online motion-communication co-optimization in Algorithm 3. The agents deploy Algorithm 3 every so often to account for new channel learning.

Algorithm 3: Distributed Online Motion & Comm. Adaptation

Parameters: t_0 is the current time, $s_{\text{new}}(x_1)$ is the updated channel prediction, c_{total,t_0} is the total remaining load at t_0 , and $x_{1,i}(t)$ is the previously-computed optimal path for Agent i .

Step 1: Each agent computes $S_i = b \int_{t_0}^{t_f} -\ln(s_{\text{new}}(x_{1,i}(t))) dt$.

Step 2: The agents share with each other their respective S_i and their remaining loads in a distributed manner (using average consensus or flooding) to compute $c_{\text{total},t_0}/N$ and $\sum_{j=1}^N S_j/N$.

Step 3: The agents compute their respective new optimum load according to Eq. (18):

$$c_i^+ = \frac{c_{\text{total},t_0}}{N} + \left(S_i - \frac{1}{N} \sum_{j=1}^N S_j \right), \quad \forall i = 1, \dots, N,$$

and then transfer the loads, in a distributed manner, according to Algorithm 2 to achieve the new optimum loads.

Step 4: For all $i \in \{1, \dots, N\}$, given the updated load c_i^+ , Agent i solves a single-agent problem for the remaining time $[t_0, t_f]$.

Remark 1. In Algorithm 3, it is assumed that all the agents form a connected network, which, however, may not always be possible during the operation. Instead, the agents may form locally-connected networks or clusters with other nearby agents. The distributed adaptation of Algorithm 3 can then be run in each cluster to re-distribute the loads and update the paths, given updated channel knowledge.

In the following proposition, we show that the distributed online adaptation scheme of Algorithm 3 guarantees improvement to the agents' initial motion/transmission control decisions (obtained either from the initial centralized solution, or from the previous adaptation) and reduces the overall cost.

Proposition 2. Suppose that an initial solution to Problem (12) is obtained based on the previously-predicted channel $s_{\text{init}}(x_1)$ with cost J_{init} . Given an improved channel prediction $s_{\text{new}}(x_1)$, Algorithm 3 provides an adapted solution with cost J_{new} . We then have $J_{\text{new}} \leq J_{\text{init}}$, where both costs are based on Eq. (12) and the more accurate channel prediction $s_{\text{new}}(x_1)$.

Proof. Given the updated channel prediction, first consider a simplified optimal control problem,

$$\min_R \sum_{i=1}^N \int_{t_0}^{t_f} \frac{2^{R_i(t)} - 1}{K} s_{\text{new}}(x_{1,i}(t)) dt + C_3 \left\| \sum_{i=1}^N x_{3,i}(t_f) - c_{\text{total},t_0} \right\|^2$$

subject to the constraints (4) and (8). The control variables only consist of the agents' spectral efficiencies R_i , while the agents' motion controls are the same as in the initial solution. Thus, $u_i(t)$ and $x_{1,i}(t)$ are simply pre-defined functions of time. In this problem, there is only one state $x_{3,i}(t)$ for each agent, which is as defined in Eq. (4).⁵ For each agent, the costate equations and boundary conditions are given by

$$\dot{p}_{3,i}(t) = 0, \quad p_{3,i}(t_f) = 2C_3 \left(\sum_{i=1}^N x_{3,i}(t_f) - c_{\text{total},t_0} \right).$$

⁵The subscript index 3 is kept to be consistent with the earlier formulations.

The Hamiltonian associate with this problem is given by

$$H(x_3, R, p_3) = \sum_{i=1}^N \frac{2^{R_i(t)} - 1}{K} s_{\text{new}}(x_{1,i}(t)) + p_{3,i} R_i(t)$$

Based on a similar derivation as in the proof of Theorem 1, the optimal spectral efficiency R_i^* is given by

$$R_i^*(t) = a - b \ln(s_{\text{new}}(x_{1,i}(t))). \quad (20)$$

We can then construct a feasible solution to Problem (12) where we use the transmission controls given by Eq. (20) and the motion controls given by the initial solution. Denote the cost of this constructed feasible solution as J_1 , which is evaluated based on $s_{\text{new}}(x_1)$. We then have $J_1 \leq J_{\text{init}}$, as the communication related costs are reduced in J_1 , while the motion related costs stay the same as in J_{init} .

It can be verified that Eq. (17) holds for the constructed feasible solution above, indicating that its load distribution is the same as in Eq. (18). Thus, the constructed feasible solution provides a feasible solution to each agent's single-agent optimization in Step 3 of Algorithm 3. Since Step 3 further optimizes the motion/transmission controls of each agent given the adapted loads, we then have $J_{\text{new}} \leq J_1 \leq J_{\text{init}}$. \square

Remark 2. In Algorithm 3, it is assumed that the updated channel prediction $s_{\text{new}}(x_1)$ is the same for each agent. In practice, it is possible that each agent receives different additional channel measurements, resulting in a different updated channel prediction for each agent, $s_{\text{new},i}(x_1)$.⁶ In this case, Algorithm 3 can still be used by replacing the common $s_{\text{new}}(x_1)$ with the individual $s_{\text{new},i}(x_1)$ in Step 1, when evaluating S_i . In addition, a similar result to Proposition 2 holds with the cost evaluated by Eq. (12), and Agent i 's transmission power computed using $s_{\text{new},i}(x_1)$, $\forall i \in \{1, \dots, N\}$.

A. Numerical Results

We next demonstrate the efficacy of our proposed distributed online adaptation algorithm in a realistic wireless environment. We present an example with 2 agents to showcase the details of the distributed online adaptation process. In this simulation, Agents 1 and 2 start at the same point of (10,40), and their destinations are (42,45) and (5,10), respectively. The total amount of data to be transmitted is 195 bits/Hz. In this online adaptation scenario, the motion-communication co-optimization problem is first solved once in a centralized manner with an initial channel prediction based on 100 random measurements. The agents then execute the motion/transmission controls given by the initial solution. After 10 seconds, the agents obtain 100 additional random channel samples and update predicted channel based on the cumulative pool of measurements. With the updated channel prediction, the agents perform the distributed online adaptation of Algorithm 3. In Step 4 of Algorithm 3, each agent runs Algorithm 1 for the single-agent problem for 60 iterations (which takes about 2.5 seconds on average), where the optimization is initialized with the agent's previously-obtained solution. We only need such few iterations because: 1) Algorithm 1 can rapidly reduce cost, and 2) the previously-obtained solution provides a good starting point for the optimization with the

⁶For instance, an agent may collect some channel data on its own and/or receive some data from other data-collecting robots in the vicinity.

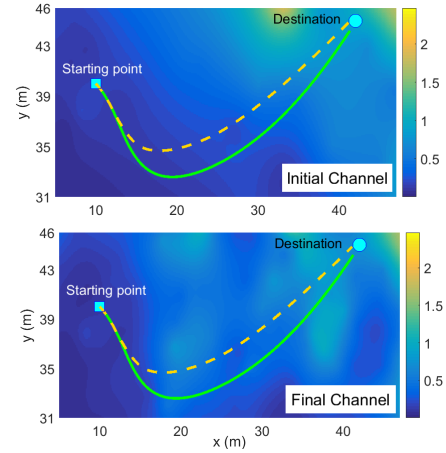


Fig. 7: Green solid line shows the initial path based on 100 channel measurements and yellow dashed line shows the path obtained from performing online channel prediction and the distributed adaptation of Algorithm 3 every 10 seconds. The square (disk) is the starting point (destination). The initial (final) channel prediction is shown in the top (bottom) figure with darker colors (lower values) indicating better channel quality. Only Agent 1 is shown here, which has large path updates. Readers are referred to the color PDF for optimal viewing.

updated channel prediction. This online update process is performed every 10 seconds in the simulation. The channel and optimization parameters are the same as in Sec. III-A1.

Fig. 7 shows Agent 1's initial (green solid) and actual (yellow dashed) paths, which has large path updates, with the initial/final predicted channel in the background. Darker colors (lower values) indicate better channel qualities. As we can see, Agent 1 initially plans to steer towards good channel quality regions (base station at (5,5)) before going to its destination. During the operation, however, as Agent 1 discovers that the channel quality near its initially-planned path is much worse than the initial prediction, it transfers some load to Agent 2 to reduce its communication burden, as shown in Table II. With much less transmission responsibility, Agent 1 focuses more on motion efficiency and moves directly towards its destination. Basically, Agent 1 realizes that it would incur too much motion energy to send its previously-assigned load itself so that it is better to have Agent 2 send most of its load instead.

Table II shows the initial load assignment as well as the actual final load transmitted after online updates. It can be seen that Agent 1 has transmitted 57.83% less data as compared to the initial load. On the other hand, as Agent 2 travels in regions which turn out to have a relatively good channel quality (after online learning), it takes on a large portion of Agent 1's initial assignment and transmits more data than initially planned. Additionally, Fig. 8 shows the two agents' initial and adapted spectral efficiencies w.r.t. time. As can be seen, Agent 1 keeps decreasing its spectral efficiency since it encounters worse channel quality. On the other hand, in order to transmit the additional load given by Agent 1, Agent 2 has to increase its spectral efficiency during the operation.

As shown in Table III, the total energy cost of the initial solution is 836.92 Joules (evaluated based on Eq. (6)) and that of the adapted final solution given by Algorithm 3 is 665.92 Joules, which is 20.43% less. It is noteworthy that the communication energy saving is very significant, which is 55.25%. This indicates that our proposed distributed

Agent	Initial Data Load	Final Data Load
1	83	35
2	112	160

TABLE II: The 1st column shows the agents' indices. The 2nd column shows the initially-assigned loads. The 3rd column shows the final transmitted load by each agent after online load adaptations of Algorithm 3 as new channel information becomes available.

Energy Cost	Comm.	Motion	Combined (Eq. (6))
Initial Sol.	286.29	550.62	836.92
Adapted Sol.	128.12	537.81	665.92
Savings	55.25%	2.33%	20.43%

TABLE III: The table shows the communication, motion, and combined (Eq. (6)) energy costs by using the initial controls and Algorithm 3, respectively, as well as the percentage energy savings by using Algorithm 3 as compared to the initial solution.

online adaptation method of Algorithm 3 can effectively reduce energy costs by adapting the agents' loads and motion/transmission controls to updated channel information.

V. CONCLUSIONS

In this paper, we considered the motion-communication co-optimization of a team of robots operating in realistic communication environments, whose decisions were coupled via a cooperative data transmission task. By utilizing realistic motion and communication models, we formulated and solved this problem in an optimal control framework. We showed how to optimally design the load distribution, paths, and transmission power/rate schedules for the robots such that the total motion and communication energy costs are minimized, and further mathematically characterized some properties of the optimal solution. We then extended this problem to an online adaptation setting where the robots need to cooperatively adjust their decisions (e.g., paths, loads to transfer, transmission power/rates) based on updated channel knowledge during the operation. We demonstrated how this problem can be efficiently solved in a novel distributed manner, and proved that this online distributed approach improves the performance. Extensive simulations with real channel parameters demonstrated the efficacy of the proposed approach and validated the theoretical results.

REFERENCES

- [1] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [2] L. E. Parker, "Multiple mobile robot systems," in *Springer Handbook of Robotics*, 2008, pp. 921–941.
- [3] J. Fink, A. Ribeiro, and V. Kumar, "Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams," *Proceedings of the IEEE*, vol. 100, no. 1, 2012.
- [4] Y. Kantaros and M. M. Zavlanos, "Communication-aware coverage control for robotic sensor networks," in *Proceedings of the IEEE Conference on Decision and Control*, 2014, pp. 6863–6868.
- [5] Y. Yan and Y. Mostofi, "To go or not to go on energy-aware and communication-aware robotic operation," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 218 – 231, July 2014.
- [6] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, "Towards mobility as a network control primitive," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2004, pp. 163–174.

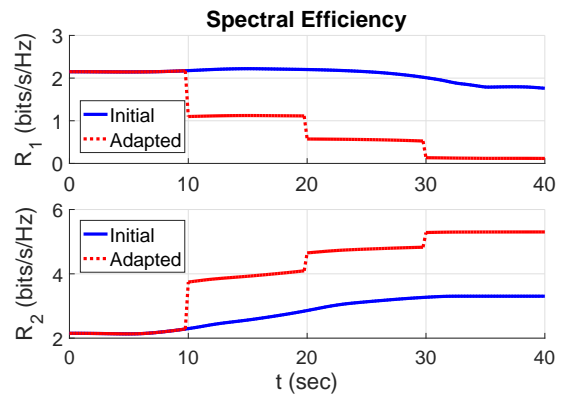


Fig. 8: Initial and adapted spectral efficiency w.r.t. time (top: Agent 1, bottom: Agent 2). Since Agent 1 transfers some data load to Agent 2 as shown in Table II, Agent 1 (Agent 2) reduces (increases) its spectral efficiency accordingly during the operation.

- [7] C. Tang and P. McKinley, "Energy optimization under informed mobility," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 947–962, 2006.
- [8] C. Ooi and C. Schindelhauer, "Minimal energy path planning for wireless robots," *Mobile Networks and Applications*, vol. 14, no. 3, pp. 309–321, 2009.
- [9] F. El-Moukaddem, E. Torng, G. Xing, and G. Xing, "Mobile relay configuration in data-intensive wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 261–273, 2013.
- [10] S. Yu, C. G. Lee, and J. Hu, "Energy optimal control in mobile sensor networks using hybrid systems theory," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2011.
- [11] S. Yu and C. G. Lee, "Mobility and routing joint design for lifetime maximization in mobile sensor networks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.
- [12] M. Hale, Y. Wardi, H. Jaleel, and M. Egerstedt, "Hamiltonian-based algorithm for optimal control," *arXiv preprint arXiv:1603.02747*, 2016.
- [13] U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, "Motion and communication co-optimization with path planning and online channel prediction," in *Proceedings of the American Control Conferenc*, 2016, pp. 7079–7084.
- [14] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, pp. 1–22, 2013.
- [15] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 964–978, 2012.
- [16] A. Goldsmith, *Wireless Communications*. Cambridge Univ. Press, 2005.
- [17] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*. CRC Press, 1975.
- [18] E. Polak, *Optimization: Algorithms and consistent approximations*. Springer-Verlag New York, Inc., 1997.
- [19] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506–2517, 2009.
- [20] A. Tanenbaum, *Computer Networks*. Prentice Hall, 2002.
- [21] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *Journal of Parallel and Distributed Computing*, vol. 7, no. 2, pp. 279–301, 1989.