Energy-Constrained Coordination of Multi-Robot Teams

Tina Setter, Student Member, IEEE, and Magnus Egerstedt, Fellow, IEEE

Abstract—This paper discusses the problem of making multiple robots coordinate their movements subject to energy constraints imposed by their available battery levels. This problem is instantiated in the context of energy-aware rendezvous, whereby a collection of mobile robots must decide where and when to meet, with the objective of doing so in the least amount of time, given varying available initial battery levels. As the movements of the individual robots directly affect their energy consumption, the robots must coordinate their movements in such a way that they do not run out of energy prior to the completion of the task. This problem is formulated as a constrained optimization problem, where the constraints themselves result from a lower-level optimal control problem that ensures that all of the robots have non-negative battery levels when they meet. The energy-constrained coordination problem is solved in a distributed manner and is implemented on a team of differential-drive mobile robots.

Index Terms—energy-aware coordination, mobile robots, multi-robot systems, optimal control, rendezvous.

I. INTRODUCTION

One driving application for multi-agent robotics is sustained environmental monitoring, where deployments are envisioned to take place in environments where it may not be easy to replace batteries or refill the fuel, such as hostile or inaccessible environments, e.g., [1], [2]. As a result, energy consumption must be taken into account already at the design phase to ensure that the robot team can complete its mission. And, what makes this different from the types of energy-constrained problems encountered, for example, in the sensor network literature is that mobility is the main culprit from an energy-consumption point-of-view.

In fact, in the sensor networks community, a number of energy-aware algorithms have been developed to maximize the lifetime of a network, while still being able to satisfy some target level of coverage, e.g., [3]–[8]. These strategies come in to play once the network has been assembled. However, there is a broad class of problems in which mobility – both as a means of network deployment and for sustained coverage – becomes the bottleneck. As an observation, mobility is the most “expensive” when it comes to energy consumption, whereas communications are less costly, and computations and sensing are the least expensive, in comparison [9].

In this paper we make energy consumption a central aspect of the coordination problem and we instantiate this idea in the context of the rendezvous problem [10], in which a group of mobile robots are to meet at a common location using only relative displacement information. This problem should be thought of as a canonical, coordinated mobility problem, and it builds on the results in [9] and [11], where mobility-driven battery depletion has repercussions on the communication footprints in the context of a particular control strategy, as opposed to an energy-aware (mobility only) design strategy, as is the focus of this paper.

Efforts have also been made to minimize both motion energy and communication energy in mobile robot networks and in some cases, sensing energy is considered as well. For example, [12] minimizes motion and communication energy in the context of transmitting information from a stationary object to a stationary remote station. In [13], mobility, sensing and communication costs were all taken into account from an energy-consumption point-of-view, in the context of a dynamic coverage problem.

Sometimes we may not have control over the available energy of the robots when they are deployed, and in these cases, energy minimization may not be enough to ensure that the robots can complete their desired task. In contrast to the majority of the previous work, this paper views the available battery levels as a hard constraint and connects this constraint to the minimum-time rendezvous problem. What this means is that the robots must take their battery levels into account while agreeing on both where and when to meet. Once the meeting location and time is decided upon, an optimal control strategy is employed that achieves the rendezvous
objective while minimizing energy consumption. Initial work on this problem was done in [14] in terms of an initial characterization of the optimal motions. The work in this paper extends that of [14] by adding an implementation of the algorithm on a robot team.

The outline of the paper is as follows: In Section II, the spatio-temporal rendezvous problem is introduced, whereby the robots need to not only decide where to meet, but also when. This is to be achieved subject to a hard constraint on the individual robots’ motions that corresponds to never draining the batteries completely. The form of this constraint is the subject of Section III, where the optimal motions associated with the individual robots are computed, resulting in a hard bound on what is achievable given an initial battery level. Section IV presents the solution to the spatio-temporal rendezvous problem using energy-optimal motions, which, in turn, is implemented on a team of differential-drive mobile robots, as presented in Section V.

II. ENERGY-CONSTRAINED SPATIO-TEMPORAL RENDEZVOUS

Consider a network of mobile robots that are to meet at some spatial location, where each robot starts with some non-negative battery level. At some point, the batteries will run out, and they must ensure that rendezvous is achieved before that happens, i.e., it matters when the robots meet as well, which is why we denote this the Spatio-Temporal Rendezvous problem. As an illustration, suppose there are two robots that have the same initial battery level. In order to meet in an equitable way, the correct approach would be to have the robots meet at the average of their initial positions, as shown in Fig. 1(a). Instead suppose that the battery levels are different. Where should they meet? It seems intuitive that they should meet closer to the robot with a lower available energy (b).

Not only do we want to solve the spatio-temporal rendezvous problem, but we also want to know how the robots should move to achieve rendezvous while minimizing the total energy consumption. This can be formulated as an optimal control problem, which differs from the standard notion of minimum energy control, where the $L_2$-norm of the control input is taken as a proxy for energy. The difference is due to the fact that mobility drives energy consumption in a way that is a function of more than just the control input.

Our approach to the energy-constrained spatio-temporal rendezvous problem is to separate it into two sub-problems, with the first being to determine where and when the mobile robots should meet, while satisfying the requirement that each robot must end up at the meeting location with a non-negative battery level. The construction of energy-optimal motions that realize the solution to the first sub-problem becomes the second sub-problem that must be addressed, and we start with a formulation of the first problem and leave the second problem to the subsequent section.

Suppose we have $N$ planar robots, whose task it is to meet at a common location. Let $p_i \in \mathbb{R}^2$, $i = 1, \ldots, N$, denote the opinions of the individual robots as to where this location should be. Similarly, let $\tau_i \in \mathbb{R}_+$ be the time at which robot $i$ thinks the team should meet. Since the robots should agree on when and where to meet, we would like to find a distributed protocol that makes $p_i = p_j$ and $\tau_i = \tau_j$, $\forall i, j \in \{1, \ldots, N\}$.

Now, what makes this a non-standard consensus problem is that the robots need to meet without draining their batteries and we would moreover like to achieve rendezvous quickly. Thus, the corresponding, constrained optimization problem becomes

$$\min_{\langle p, \tau \rangle} \sum_{i=1}^{N} \left( \frac{1}{2} \rho \tau_i^2 + \sum_{j \in N_i} (\|p_i - p_j\|^2 + \|\tau_i - \tau_j\|^2) \right)$$

s.t. $h_i(p_i, \tau_i) \leq 0$, $\forall i = 1, \ldots, N$, \hspace{1cm} (1)

where $p = [p_1^T, \ldots, p_N^T]^T$, $\tau = [\tau_1, \ldots, \tau_N]^T$, and $N_i$ is the set representing robot $i$’s neighborhood – the robots with whom robot $i$ can share information. Finally, $h_i(p_i, \tau_i)$ is the constraint (to be determined) that ensures that robot $i$ can indeed reach $p_i$ in time $\tau_i$ without running out of energy. And even though the robots are assumed to be similar, they will start at different positions and with different initial battery levels, which we encode through the subscript $i$ in the individual constraints.

The first term in the cost ensures that the robots meet quickly, where the weighting factor $\rho > 0$ determines how strongly this term affects the cost. The second term

1 One could additionally introduce weights $w_p$ and $w_\tau$ in front of the terms $\|p_i - p_j\|^2$ and $\|\tau_i - \tau_j\|^2$, respectively.
ensures that the decision variables, $p$ and $\tau$, across all robots get “close” in the least-squares sense. As such, what we consider is not exact rendezvous, but rather that the robots end up sufficiently close together at roughly the same time.

Now, in order to find a mathematical expression for $h_i$, we need to understand how the robots are moving and how that movement in turn affects the battery life of the robots. To this end, decisions must be made about the dynamics of the robots and their energy-consumption model. This is the topic of the next section. Once an expression for $h_i(p_i, \tau_i)$ has been derived, we will return to the spatio-temporal rendezvous problem.

III. SINGLE ROBOT CONTROL FOR MINIMUM ENERGY CONSUMPTION

The coupling between the different robot movements arises from the cost associated with the spatio-temporal rendezvous problem. However, the battery constraint is not coupled – robot $i$’s battery level dynamics does not depend on robot $j$’s battery levels. As such, in order to arrive at the constraint $h_i(p_i, \tau_i)$, it is sufficient to consider a single agent in isolation, which will be done in this section.

A. Energy Dissipation and Mobility

The pose of a differential-drive mobile robot comprises of position and orientation. If we let $(q_{i1}, q_{i2})$ be the two-dimensional position of robot $i$, moving in the direction $q_{i3}$ with linear velocity $v_i$ and angular velocity $\omega_i$, the kinematics become

\[
\begin{align*}
\dot{q}_{i1} &= v_i \cos q_{i3} \\
\dot{q}_{i2} &= v_i \sin q_{i3} \\
\dot{q}_{i3} &= \omega_i.
\end{align*}
\]

For the purpose of analysis, we will assume that the robots move along straight lines, i.e., $\omega_i = 0$. This assumption is justified by the fact that the shortest distance path between two planar points is the straight-line path between them and since the goal is to minimize energy consumption due to mobility, each robot should travel the shortest distance possible in order to reach its destination. In the robotic implementation, this assumption will be handled by initially having the robots turn in place until each is aligned with its target position. This initial re-orientation is deemed negligible in terms of energy consumption.

As a consequence, we can focus on this one-dimensional motion, where the distance the robot has traveled along its straight-line path is denoted $x_{i1}$, where $\dot{x}_{i1} = v_i$ is the robot’s velocity. However, to capture battery usage, we need to go beyond kinematic models, and we let $x_{i2} = u_i$, with dynamics $\dot{x}_{i2} = u_i$, with $u_i$ being the input, thus connecting to the rich literature on double integrator coordination, e.g., [15]–[18].

Finally, if we let $x_{i3}$ denote the robot’s available battery level, the dissipation of energy is given by quadratic functions of the velocity and acceleration of the robot, similar to the model used in [19], yielding the combined dynamics

\[
\begin{align*}
\dot{x}_{i1} &= x_{i2} \\
\dot{x}_{i2} &= u_i \\
\dot{x}_{i3} &= -x_{i2}^2 - \alpha u_i^2,
\end{align*}
\]

where $\alpha > 0$ determines how strongly acceleration affects the dissipation of energy in comparison to the effect of the velocity. Clearly, the battery level is decreasing while the robot is moving, hence we need that $\dot{x}_{i3} \leq 0$ at all times, which is satisfied by these dynamics. Also, if the robot is not accelerating, but still moving (i.e. has non-zero velocity), the battery level should still be decreasing.

B. Optimal Control Design

As the ambition is to move in a manner that conserves energy, we seek to find the control input $u_i$ that minimizes the total energy consumed, $x_{i3}(0) - x_{i3}(T)$. This optimal control problem can be formulated as

\[
\min_{u_i} \int_0^T -\dot{x}_{i3}(t) \, dt
\]

subject to the constraints that the robot starts and ends at rest and reaches its target position while ensuring that the battery level is non-negative at the end of the maneuver, i.e., such that

\[
\begin{align*}
x_{i1}(0) &= x_{i10} & x_{i1}(T) &= x_{i1T} \\
x_{i2}(0) &= 0 & x_{i2}(T) &= 0 \\
x_{i3}(0) &= x_{i30} > 0 & x_{i3}(T) &\geq 0.
\end{align*}
\]

Here $T$ is the time over which the maneuver is defined – later to be made an explicit part of the problem – and $x_{i10}, x_{i30}$ are the given initial positions and battery levels, respectively, and $x_{i1T}$ is the given final position. It should be noted already at this point that this problem may not have a solution, e.g., if $T$ is not large enough to ensure that the final battery level is non-negative.

Therefore, in order to solve this problem analytically, we will remove the constraint that the final battery level

\[\text{As already stated, this model does not consider energy consumed through sensing, communication, and computations as those are deemed to be negligible as compared to consumption due to mobility.}\]
be non-negative, which allows us to remove $x_{i3}$ as a state in the optimal control design altogether. We will later re-incorporate $x_{i3}$ and choose $T$ in order to ensure that the final battery level is non-negative, thus characterizing when a solution to the original problem exists. This relaxed problem that we first solve is

$$\min_{u_i} \int_0^T (x_{i2}^2(t) + \alpha u_i^2(t)) \, dt$$

subject to the dynamics

$$\dot{x}_{i1} = x_{i2},$$
$$\dot{x}_{i2} = u_i$$

and the fixed initial and final conditions

$$x_{i1}(0) = x_{i10}, \quad x_{i1}(T) = x_{i1T},$$
$$x_{i2}(0) = 0, \quad x_{i2}(T) = 0.$$  

We can now solve this optimal control problem for $u_i$, which is given in the following theorem.

**Theorem III.1.** The control signal $u_i$ that solves the problem in (3), subject to the dynamics in (4) and to the boundary conditions in (5), is given by:

$$u_i = -\frac{1}{2\alpha} \left( c_{i1} e^{\frac{T}{\sqrt{\alpha}}} + c_{i2} e^{-\frac{T}{\sqrt{\alpha}}} \right),$$

where

$$c_{i1} = \frac{1}{2} e^{-\frac{T}{\sqrt{\alpha}}} \left( \nu_{i2} - \nu_{i1} \sqrt{\alpha} \right),$$

and

$$c_{i2} = \frac{1}{2} e^{\frac{T}{\sqrt{\alpha}}} \left( \nu_{i2} + \nu_{i1} \sqrt{\alpha} \right),$$

with

$$\nu_{i1} = \frac{2(x_{i10} - x_{i1T})}{T - 2\sqrt{\alpha} \tanh \left( \frac{T}{2\sqrt{\alpha}} \right)}$$

and

$$\nu_{i2} = -\sqrt{\alpha} \tanh \left( \frac{T}{2\sqrt{\alpha}} \right) \nu_{i1}.$$  

**Proof.** Following the Pontryagin Maximum Principle for fixed-endpoint control problems, given, e.g., in [20], the Hamiltonian associated with the cost in (3) and dynamics in (4) is

$$H = x_{i2}^2 + \alpha u_i^2 + \lambda_{i1} x_{i2} + \lambda_{i2} u_i,$$

where the costates $(\lambda_{i1}, \lambda_{i2})$ satisfy

$$\dot{\lambda}_{i1} = \frac{\partial H}{\partial x_{i1}} = 0,$$
$$\dot{\lambda}_{i2} = -\frac{\partial H}{\partial x_{i2}} = -2x_{i2} - \lambda_{i1}.$$  

We let $\lambda_{i1} = \nu_{i1}$, for some constant $\nu_{i1}$ on the entire interval since $x_{i1}$ is determined at time $T$. This yields $\lambda_{i2}(t) = -2x_{i2}(t) - \nu_{i1}$ and we set $\lambda_{i2}(T) = \nu_{i2}$ for some constant $\nu_{i2}$, leaving us with the two unknowns, $\nu_{i1}$ and $\nu_{i2}$.

The optimality condition on $u_i$ is

$$\frac{\partial H}{\partial u_i} = 2\alpha u_i + \lambda_{i2} = 0 \implies u_i = -\frac{\lambda_{i2}}{2\alpha}$$

and the second costate $\lambda_{i2}$ can be found by differentiating $\lambda_{i2}$, plugging in $u_i$ from (7) and then solving the resulting second-order differential equation

$$\ddot{\lambda}_{i2} = \frac{1}{\alpha} \lambda_{i2},$$

with boundary conditions $\lambda_{i2}(T) = \nu_{i2}$ and $\lambda_{i2}(0) = \lambda_1(T) = -\nu_{i1}$ due to the boundary conditions on $x_{i2}$. The solution is

$$\lambda_{i2} = c_{i1} e^{\frac{T}{\sqrt{\alpha}}} + c_{i2} e^{-\frac{T}{\sqrt{\alpha}}},$$

where

$$c_{i1} = \frac{e^{-\frac{T}{\sqrt{\alpha}}} \sqrt{\alpha}}{2 (\nu_{i2} - \nu_{i1} \sqrt{\alpha})}$$

$$c_{i2} = \frac{e^{\frac{T}{\sqrt{\alpha}}} \sqrt{\alpha}}{2 (\nu_{i2} + \nu_{i1} \sqrt{\alpha})}.$$  

By plugging (8) into the equation for $u_i$ in (7), $u_i$ is completely determined. And since $\dot{x}_{i1} = u_i$, we can integrate $u_i$ twice to obtain

$$x_{i1} = -\frac{1}{2} \left( c_{i1} e^{\frac{T}{\sqrt{\alpha}}} + c_{i2} e^{-\frac{T}{\sqrt{\alpha}}} \right) + x_{i10}.$$  

with terminal constraint $x_{i1}(T) = x_{i1T}$. By substituting the expressions for $c_{i1}$ and $c_{i2}$ from (9) and (10) into (11) and using the terminal constraint, we get

$$\nu_{i1} \left( T - \sqrt{\alpha} \sinh \left( \frac{T}{\sqrt{\alpha}} \right) \right)$$

$$+ \nu_{i2} \left( 1 - \cosh \left( \frac{T}{\sqrt{\alpha}} \right) \right) = 2 (x_{i10} - x_{i1T}),$$

which, together with (9) and (10) and the condition $\lambda_{i2}(0) = -\nu_{i1}$, gives

$$\nu_{i2} = -\sqrt{\alpha} \tanh \left( \frac{T}{2\sqrt{\alpha}} \right) \nu_{i1},$$

and

$$\nu_{i1} = \frac{2(x_{i10} - x_{i1T})}{T - 2\sqrt{\alpha} \tanh \left( \frac{T}{2\sqrt{\alpha}} \right)},$$

where everything on the right hand side of (14) is known. Hence we know $\nu_{i2}$, and can compute $c_{i1}$ and $c_{i2}$, which describe $\lambda_{i2}$ completely, and thus $u_i$ is also completely determined. □
A direct consequence of Theorem III.1 is that given $T$, $x_{i10}$, and $x_{i1T}$, the optimal control input that minimizes the total energy consumed throughout the move can be explicitly found. It should be noted, however, that the initial battery level does not matter to the control input and, as a consequence, we may indeed end up with a negative final battery level, which is impractical. But, the optimal control construction can be used to drive the system until the battery level is completely drained (i.e., $x_{i3} = 0$), which in turn corresponds to the shortest amount of time, $T_{\min,i}$, in which the move can be completed such that the final battery level is non-negative. Moreover, $T_{\min,i}$ certainly depends on the initial battery level, $x_{i30}$, even if $u_i$ does not.

In order to determine the shortest amount of time in which the motion can be completed, an expression for the battery level $x_{i3}(t)$ is needed. It is found by substituting the optimal control input $u_i$ from (6) and the corresponding velocity $x_{i2}$ into the dynamics for $x_{i3}$, given in (2), and integrating. $T_{\min,i}$ is then derived by computing $x_{i3}(T_{\min,i})$ and setting this expression equal to zero, i.e.,

$$
\frac{(x_{i10} - x_{i1T})^2}{T_{\min,i} - 2 \sqrt{\alpha} \tanh \left( \frac{T_{\min,i}}{2 \sqrt{\alpha}} \right)} - x_{i30} = 0
$$  \hspace{1cm} (15)

Although this expression is quite cumbersome to solve explicitly, numerical solutions are easy to come by, relating the minimum time in which the robot can achieve the total displacement, $|x_{iT} - x_{i10}|$, with the initial battery level, $x_{i30}$. As seen in Fig. 2, $T_{\min,i}$ increases with increasing distance traveled, as is to be expected. This is intuitive because in order to travel a further distance using the same amount of energy, the robot must travel slower, therefore taking longer. Also, a robot with a higher initial battery level can travel a specified distance in less time than a robot with a lower initial battery level since a higher initial battery level means that the robot is able to travel faster.

If we relax the constraint that the final battery level must be zero and instead let it be greater than or equal to zero, we get a similar condition on the initial battery level

$$
x_{i30} \geq \frac{(x_{i10} - x_{i1T})^2}{T - 2 \sqrt{\alpha} \tanh \left( \frac{T}{2 \sqrt{\alpha}} \right)},
$$  \hspace{1cm} (16)

where the expression on the right-hand side is the amount of energy lost during the motion, which must be less than or equal to the initial battery level in order to have a non-negative battery level at time $T$. This inequality constraint is exactly what we set out to find, and, by recalling that $(q_{i1}, q_{i2})$ is the two-dimensional position of robot $i$, $p_i$ is robot $i$’s opinion of the location to meet, and $\tau_i$ is the time at which robot $i$ thinks the team should meet, we can connect this back to the two-dimensional problem by replacing $(x_{i10} - x_{i1T})^2$ with $\parallel (q_{i1}(0), q_{i2}(0))^T - p_i \parallel^2$ and $T$ with $\tau_i$. A slight rearrangement of terms gives the sought-after constraint for the spatio-temporal rendezvous problem from the previous section,

$$
h_i(p_i, \tau_i) = \parallel (q_{i1}(0), q_{i2}(0))^T - p_i \parallel^2$$

$$- (x_{i30}) \left[ \tau_i - 2 \sqrt{\alpha} \tanh \left( \frac{\tau_i}{2 \sqrt{\alpha}} \right) \right].
$$  \hspace{1cm} (17)

IV. ENERGY-AWARE COORDINATION

Now that we have an expression for the constraint, $h_i$, we can tackle the energy-constrained, spatio-temporal rendezvous problem in (1). This is an optimization problem with a global cost function and local constraints, where all robots are minimizing the same objective function, but each robot has its own constraint that only depends on its own decision variables, namely $p_i$ and $\tau_i$.

Since gradient-descent methods are particularly well-suited for distributed implementations, this problem was solved via the primal-dual gradient laws for constrained optimization\(^3\), as described in [21], using the update dynamics given by

$$\dot{p}_i = - \sum_{j \in N_i} (p_i - p_j) - \frac{\partial h_i}{\partial p_i} (p_i, \tau_i) \mu_i
$$

$$\dot{\tau}_i = - \sum_{j \in N_i} (\tau_i - \tau_j) - \rho \tau_i - \frac{\partial h_i}{\partial \tau_i} (p_i, \tau_i) \mu_i
$$

$$\dot{\mu}_i = \begin{cases} h_i(p_i, \tau_i) & \text{if } h_i(p_i, \tau_i) > 0 \text{ or } \mu_i > 0 \\ 0 & \text{otherwise} \end{cases}
$$

\(^3\)Because the constraints $h_i$ are convex in $p_i$, but not in $\tau_i$, only locally optimal solutions can be ensured.
where the dynamics associated with the Lagrange multiplier, $\mu_i$, ensure that the multipliers remain positive, which is necessary because of the Karush-Kuhn-Tucker conditions associated with the inequality constraints. Note that this is a decentralized algorithm since each robot needs only its own $p_i$, $\tau_i$, and $\mu_i$ values and the $p_j$ and $\tau_j$ values of its neighbors. Each robot’s $\mu_i$ dynamics only depend on its own values of $p_i$, $\tau_i$, and $\mu_i$, and thus these multipliers do not have to be shared among neighboring robots.

This algorithm was implemented in MATLAB for a network of three simulated robots with one-dimensional positions. The robots’ initial positions are 10, 20, and 30, with initial battery levels of 30, 30, and 5, respectively. The initial meeting point for robot $i$, i.e., $p_i$, was set to robot $i$’s initial position, $x_{i10}$, to indicate that robot $i$ would initially like to meet the other robots at its own starting position. Each robot’s initial meeting time, $\tau_i$, was set to 100, so that the energy constraint ($h_i(p_i, \tau_i) \leq 0$) would be satisfied initially for all robots. In this simulation as well as in the robotic implementation in the next section, the dissipation coefficient $\alpha$ in (2) was set to 5. This coefficient can be obtained through a calibration procedure, as discussed in [19].

The dynamics in (18) were executed in simulation using an ODE solver with a variable step Runge-Kutta method to obtain approximate solutions for $p_i$, $\tau_i$, and $\mu_i$, for $i = 1, 2, 3$. Fig. 3 shows the evolution of the meeting points, $p_i$, for $i = 1, 2, 3$, and Fig. 4 shows the evolution of the meeting times, $\tau_i$ for $i = 1, 2, 3$.

In this example, the robots do indeed end up agreeing to meet closer to the robot that has the lowest initial battery level, as was to be expected. In this simulation, the final meeting point values were $p_1 = 24.1925$, $p_2 = 24.002$, and $p_3 = 24.2079$, whereas the final meeting times were $\tau_1 = 11.1250$, $\tau_2 = 11.1169$, and $\tau_3 = 11.1203$. With the meeting locations and times computed, each robot can individually use (6) to execute the optimal input $u_i$ that allows it to travel to the meeting location using the least amount of energy.

V. IMPLEMENTATION ON A ROBOT TEAM

The energy-constrained, spatio-temporal rendezvous algorithm was implemented on a team of Khepera III differential-drive mobile robots, each having a 600MHz ARM processor with 128MB RAM, embedded Linux, and a wireless card for enabling communication over a wireless router. Ten Optitrack S250e motion capture cameras were used to obtain highly accurate position and orientation data for the robots, providing the information required for the algorithmic implementation and to project the visualization of the robot battery levels onto the floor. All code was written in MATLAB.
Three robots were used, with initial positions given by $[0, -1]$, $[-1.5, 1]$, and $[1, 0]$, and initial battery levels given by 2, 1, and 0.5, respectively. The robots’ initial positions and relative initial battery levels are depicted in Fig. 5, where the radii of the circles surrounding the robots are pictorial representations of the available battery levels. The meeting points and meeting times were obtained by solving the constrained optimization problem in a distributed manner, with solution $p_1 = [-0.0304, 0.4278]$, $p_2 = [-0.0318, 0.4283]$, $p_3 = [-0.0289, 0.4272]$, and $\tau_1 = 6.4886$, $\tau_2 = 6.4890$, $\tau_3 = 6.4889$. These decision variables were computed before the robots started moving.

The corresponding, optimal trajectories were then executed on the robots by having them travel along straight-line paths toward the meeting point with the corresponding, optimal linear velocities. The robots first turn in place until they are positioned in such a way that they can reach their goal position by traveling along straight-line paths. This initial turning is deemed negligible in terms of energy consumption. While the robots are moving, the modeled battery life of each is depicted by projected circles on top of each robot, where the radius of each circle is proportional to the respective robot’s remaining battery level.

Four snapshots of the robots driving to the meeting point are presented in Fig. 6. It can be seen that at the
end of the move, one of the robots has significant battery life remaining, while the other two robots have depleted their batteries. It can also be seen that instead of the robots meeting at the average of their initial positions \((-0.1667, 0)^T\), as would happen if they were running the standard consensus protocol, they instead meet closer to the robots with the lower initial battery levels, i.e., robots 2 and 3. The difference between the average of their initial positions and where they actually meet can be seen in Fig. 7. The modeled battery level trajectories are shown in Fig. 8.

VI. CONCLUSION

In this paper, we present an energy-constrained strategy for allowing a network of robots to achieve rendezvous in the shortest possible amount of time without depleting their batteries. We formulate the problem of finding the best location for the robots to meet given their initial positions and battery levels as a constrained optimization problem. The energy constraints arise by solving the problem of how to control a single robot that is moving from some given initial position to some desired final position such that the overall energy consumption (due to mobility) is minimized. This allows us to achieve rendezvous as quickly as possible while also minimizing the amount of energy that each robot consumes throughout the motion. The algorithms described in this paper were implemented on a team of Khepera III mobile robots to show that this is viable in practice.

REFERENCES


Tina Setter (S’11) received the B.S. degree in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2012, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2013. She is currently working toward the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology.

As a member of the Georgia Robotics and Intelligent Systems (GRITS) Laboratory under the direction of Dr. Magnus Egerstedt, her research interests are in control theory, multi-agent coordination algorithms, and human-swarm interaction.

Magnus Egerstedt (S’99–M’00–SM’05–F’12) received the B.A. degree in philosophy from Stockholm University, Stockholm, Sweden, in 1996, and the M.S. degree in engineering physics and the Ph.D. degree in applied mathematics both from the Royal Institute of Technology, Stockholm, Sweden, in 1996 and 2000, respectively.

He is currently the Schlumberger Professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, where he serves as Associate Chair for Research. He is the director of the Georgia Robotics and Intelligent Systems Laboratory and a recipient of a number of research and teaching awards, including the Ragazzini Award from the American Automatic Control Council and the CAREER Award from the U.S. National Science Foundation. His research interests include control theory and robotics, with particular focus on control and coordination of complex networks, such as multirobot systems, mobile sensor networks, and cyber-physical systems.