Battery Level Estimation of Mobile Agents Under Communication Constraints

Jonghoek Kim, Fumin Zhang, and Magnus Egerstedt
Electrical and Computer Engineering, Georgia Institute of Technology, USA
jkim37@gatech.edu,fumin, magnus@ece.gatech.edu

Abstract

We consider a team of mobile agents where a leader has to monitor battery levels of all other agents. Only the leader is capable to transmit information to other agents. Every now and then, the leader commands other agents to move toward or against the leader with speed proportional to battery level of ~~the~~ agent. The leader then simultaneously estimate the battery life of other agents from measurements of the relative distances between itself and other agents. We propose a nonlinear system model that integrates ~~a particle~~ motion model and a dynamic battery model that has demonstrated high accuracy in battery capacity prediction. The extended Kalman filter (EKF) is applied to this nonlinear model to estimate the battery level of each agent. One improvement we have made to the EKF is that in addition to gain optimization, the motion of agents can also be controlled to minimize estimation error. Simulation results are presented to demonstrate effective of the proposed method.

1 Introduction

A key idea for cyber-physical systems research is to co-design different controlling mechanisms to balance performance in both physical systems and computing systems [17]. In mobile sensor networks, each sensor is a mobile robot that is able to reconfigure the structure of the network [9, 16, 15, 4, 14]. Due to the requirement for mobility, most mobile sensor networks are powered by batteries. Until now, the majority of batteries used are electrochemical batteries.

Many battery models have been developed with varying degrees of complexity. Analytical battery models are abstract models that are derived by simplifying physical models using mathematical analytic methods. Simple empirical equations or stochastic methods are used to model battery behaviors [17, 7, 10, 2]. Although many results exist in battery literature, to our knowledge, our paper is unique in designing the motion of a mobile agent jointly with battery level estimation algorithms.

Consider a team of mobile agents where a leader has to monitor battery levels of all other agents (followers)[1]. Only the leader is capable of transmitting information to other agents. To estimate battery levels of followers that are under communication constraints, a simple coordinated motion is developed. Every now and then, the leader commands other agents to move toward or away from the leader with speeds proportional to their battery levels. This motion is modeled as a particle motion. The leader then simultaneously estimates battery levels of other agents from measurements of the relative distances between itself and other agents.

The Rakhmatov-Vrudhula-Wallach (RVW) model was derived from solving the diffusion equations governing the electrolytes motion within a battery [11]. The RVW model has been shown to agree with experimental results and has demonstrated high accuracy in battery capacity prediction and battery life estimation. Authors of [17] developed a dynamic battery model based on the RVW model. The dynamic battery model is input-output equivalent to the RVW model when the number of state variables goes to infinity.

We propose a nonlinear system model that integrates the particle motion model and the dynamic battery model [17]. Based on this nonlinear model, the leader computes the extended Kalman filter (EKF) to estimate the battery level of every follower.

One improvement we have made to the EKF is that, in addition to gain optimization embedded in the EKF, motions of agents are controlled to minimize the mean square estimation error. At every N steps under the EKF, the leader adjusts motions of other agents to minimize the error covariance predicted N steps forward in time. This is a co-design approach for cyber-physical systems where the motion of an agent and battery level estimation algorithms are jointly designed. Simulation results are presented to demonstrate effectiveness of the proposed method.

[1] Leader-based formation control is discussed in [5, 13, 3, 6]

This paper is organized as follows : Section 2 formulates the problem. Section 3 introduces the nonlinear system model that integrates the particle motion model and the dynamic battery model. Battery level estimation is discussed in Section 4. In addition, Section 5 demonstrates MATLAB simulation results. Section 6 provides conclusions.

2 Problem Formulation

Consider a team of agents (agent 1,..., agent n) where a leader (agent 1) has a role to monitor battery levels of followers (agent 2,..., agent n). Suppose agent 1 is the leader. Only agent 1 is capable of transmitting information to other agents. To estimate the battery level of agent i ($\forall i \neq 1$) which is under communication constraints, we allow agent 1 to control motions of other agents. Every now and then, agent 1 commands other agents to move toward or away from agent 1 with speeds proportional to their battery levels. Agent 1 then simultaneously estimates the battery levels of other agents from measurements of the relative distances between itself and other agents.

Since all real sensors have finite range, let the maximal distance at which two agents can be separated and still sense each other be given by Δ. To estimate the battery level of agent i ($\forall i \neq 1$) based on the relative motion, we need the following assumption :

(A1) Every agent, other than agent 1, has an unique identifier to be distinguishable by agent 1. All agents are within the maximum sensing range Δ of agent 1, and LOS (line of sight) from agent 1 to any other agent is guaranteed.

3 Nonlinear System Model

3.1 Particle Motion Model

This subsection presents a particle motion model for the coordinated motion.

Agent 1 initiates the motion by sending triggering signals to agent i for all $i \neq 1$. After receiving triggering signals, agent i first stops. Then, agent i begins to move away from agent 1 with the speed $k_c\Gamma_i$ where k_c is a positive constant, and Γ_i denotes the battery level (remaining concentration of electrolyte) of agent i. The discrete-time control law that makes agent i move away from agent 1 is

$$s_{i,k+1} = s_{i,k} - k_c\Gamma_{i,k}\epsilon(s_{1,k} - s_{i,k}), \tag{1}$$

where $s_{i,k} \in R^2$ denotes the state (position) of agent i at the kth sampling in the discrete-time system, and $\Gamma_{i,k}$ is the battery level of agent i at the kth sampling. Furthermore, $\epsilon \ll 1$ is related to the sample period of the discrete-time system.

Agent i should be detectable by sensors of agent 1. Thus, once the distance between agent 1 and agent i reaches the maximum sensing range ($\Delta - \epsilon_2$ where $\epsilon_2 \ll 1$), then agent i begins to move toward agent 1 with the speed $k_c\Gamma_i$. The discrete-time control law that makes agent i move toward agent 1 is

$$s_{i,k+1} = s_{i,k} + k_c\Gamma_{i,k}\epsilon(s_{1,k} - s_{i,k}). \tag{2}$$

While agent i moves toward agent 1, it must not collide with agent 1. Thus, once the distance between agent 1 and agent i is smaller than a certain threshold ($\epsilon_2 \ll 1$), then (1) is applied again.

Fig.1 illustrates states (positions) of agent 1 and agent i ($i \in \{2, ..., n\}$). In this figure, s_i denotes the state of agent i. Bidirectional arrow shows the feasible position of agent i under the switching control.

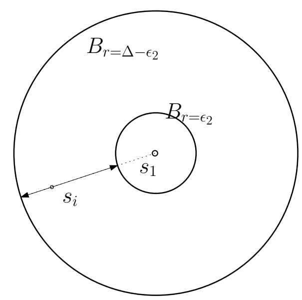

Figure 1. States (positions) of agent 1 and agent i ($i \in \{2, ..., n\}$). In this figure, s_i denotes the state of agent i. Bidirectional arrow shows the feasible position of agent i under the switching control.

3.2 System Modeling Incorporated with the Dynamic Battery Model

We present the nonlinear system model, which integrates the particle motion model and the dynamic battery model in [17]. To obtain the specific relation between the loss of battery level and the movement of each agent, the following assumption is required :

(A2) Discharge current of agent i at time step k is proportional to the speed of the agent at time step k.

Let $I_{i,k}$ denote the discharge current of agent i at time step k. Recall that the speed of agent i at time step k is $k_c\Gamma_{i,k}$. Using assumption (A2), we get

$$I_{i,k} = gk_c\Gamma_{i,k}, \tag{3}$$

where $g > 0$ is a constant.

To simplify the analysis, we use only two state variables in the dynamic battery model (See [17] for details of the dynamic battery model). Define the state variables for agent i as $x_{i,k} = [x^0_{i,k}, x^1_{i,k}]^T$ where $x_{i,0} = [0,0]^T$. Let $C = [1,1]_{1\times 2}$. Then the output $y_{i,k} = Cx_{i,k}$ represents the normalized remaining concentration of electrolyte for agent i. As time goes on, $y_{i,k}$ increases from 0 to 1. We get the system model as

$$\begin{aligned} x_{i,k+1} &= x_{i,k} + \epsilon(Ax_{i,k} + BI_{i,k}), \\ y_{i,k} &= Cx_{i,k}, \\ x_{i,0} &= [0,0]^T, \end{aligned} \tag{4}$$

where

$$A = \operatorname{diag}[0, -\lambda_1], B = \begin{bmatrix} \frac{1}{\alpha} \\ \frac{2}{\alpha} \end{bmatrix},$$

$x_{i,k} = [x^0_{i,k}, x^1_{i,k}]^T$. As $y_{i,k}$ increases from 0 to 1, the battery level decreases from Γ^*_i to 0 where Γ^*_i is an initial concentration of electrolyte for agent i. Thus, we obtain the equation for $\Gamma_{i,k}$ as follows :

$$\Gamma_{i,k} = \Gamma^*_i(1 - y_{i,k}) = \Gamma^*_i(1 - Cx_{i,k}). \tag{5}$$

Let's consider the system model when agent i moves away from agent 1 with the speed proportional to $\Gamma_{i,k}$. Since agent i can not transfer $\Gamma_{i,k}$ to agent 1, agent 1 needs to estimate $\Gamma_{i,k}$ based on the relative distance between itself and agent i.

Let d_i denote the relative displacement between agent 1 and agent i. Due to noise in the estimation, we get

$$d_{i,k+1} = d_{i,k} + \epsilon(k_c\Gamma_{i,k}) + v_k \tag{6}$$

where v_k is a zero mean gaussian noise. Further, $d_{i,k}$ denotes d_i at the kth sampling.

Using (6) and (5), we obtain

$$d_{i,k+1} = d_{i,k} + \epsilon(k_c\Gamma^*_i(1 - Cx_{i,k})) + v_k. \tag{7}$$

If we set $k = 0$ in (7), we get

$$d_{i,1} = d_{i,0} + \epsilon(k_c\Gamma^*_i) + v_0, \tag{8}$$

since $x_{i,0} = [0,0]^T$.

Note that we need to estimate an initial concentration of electrolyte Γ^*_i, since v_0 is unknown in (8). Using (3), (4), (5), and (7), we get

$$\begin{aligned} &x_{i,k+1} = x_{i,k} + \epsilon((A - Bgk_c\Gamma^*_{i,k}C)x_{i,k} + Bgk_c\Gamma^*_{i,k}) + w_k, \\ &\Gamma^*_{i,k+1} = \Gamma^*_{i,k}, \\ &d_{i,k+1} = d_{i,k} + \epsilon(k_c\Gamma^*_{i,k} - k_c\Gamma^*_{i,k}Cx_{i,k}) + v_k, \end{aligned} \tag{9}$$

where w_k is a zero mean gaussian noise. Letting $X_k = [x^T_{i,k}, \Gamma^*_{i,k}, d_{i,k}]^T$ and $q_k = [w^T_k, 0, v_k]^T$, we can express (9) as

$$X_{k+1} = F_1(X_k) + q_k, \tag{10}$$

where q_k is a gaussian noise with zero mean and the covariance matrix Q_k.

Since $d_{i,k}$ is measurable, we have

$$Y_k = HX_k + r_k, \tag{11}$$

where $H_{1\times 4} = [0,0,0,1]$, and r_k is a gaussian noise with zero mean and the covariance matrix R_k. In the case where agent i moves away from agent 1, (10) and (11) are state equation and measurement equation respectively.

Consider the system model when agent i moves toward agent 1 with the speed proportional to $\Gamma_{i,k}$. In this case, we need to change k_c in (6) to $-k_c$. Then the state equation (9) changes to

$$\begin{aligned} &x_{i,k+1} = x_{i,k} + \epsilon((A - Bgk_c\Gamma^*_{i,k}C)x_{i,k} + Bgk_c\Gamma^*_{i,k}) + w_k, \\ &\Gamma^*_{i,k+1} = \Gamma^*_{i,k}, \\ &d_{i,k+1} = d_{i,k} + \epsilon(-k_c\Gamma^*_{i,k} + k_c\Gamma^*_{i,k}Cx_{i,k}) + v_k. \end{aligned} \tag{12}$$

We can express (12) as

$$X_{k+1} = F_2(X_k) + q_k, \tag{13}$$

which is only slightly different from (10). We still use (11) as the measurement equation for this case.

4 Battery Level Estimation

4.1 Extended Kalman Filter

Since system models (10) and (13) are nonlinear systems, the extended Kalman Filter (EKF) can be applied to estimate X_k based on the measurement equation (11). Once X_k is estimated, we can obtain the battery level of agent i using (5).

To initialize the estimate in the EKF, we use (8) to get

$$\hat{\Gamma}^*_{i,0} = \frac{d_{i,1} - d_{i,0}}{\epsilon k_c}. \tag{14}$$

Note that $\hat{x}_{i,0} = [0,0]^T$ is known.

In the case where agent i moves away from agent 1, (10) and (11) are state equation and measurement equation respectively. Let μ_k be the estimate of X_k. The EKF is applied to update μ_k and P_k at each step k.

$$\mu^-_{k+1} = F_1(\mu_k). \tag{15}$$

Let $F_{1,k} = \frac{\partial F_1(x)}{\partial x}|_{x=\mu_k}$. Then we update the covariance matrix P_k.

$$P_{k+1}^- = F_{1,k} P_k F_{1,k}^T + Q_k, \quad (16)$$

where $F_{1,k}^T$ is the transpose matrix of $F_{1,k}$. The Kalman gain is obtained as

$$K_{k+1} = P_{k+1}^- H^T (H P_{k+1}^- H^T + R_{k+1})^{-1}. \quad (17)$$

We update μ_{k+1} which is the estimate of X_{k+1}.

$$\mu_{k+1} = \mu_{k+1}^- + K_{k+1}(Y_{k+1} - H\mu_{k+1}^-). \quad (18)$$

Finally, we update the covariance matrix.

$$P_{k+1} = ((F_{1,k} P_k F_{1,k}^T + Q_k)^{-1} + H^T R_{k+1}^{-1} H)^{-1}. \quad (19)$$

Since μ_{k+1}, and P_{k+1} are derived in (18) and (19) respectively, we can iterate equations from (15) to (19).

Next, consider the system where agent i moves toward agent 1. Let $F_{2,k} = \frac{\partial F_2(x)}{\partial x}|_{x=\mu_k}$. F_1 and $F_{1,k}$ in (15), (16), and (19) are replaced by F_2 and $F_{2,k}$ respectively to obtain the system where agent i moves toward agent 1.

Keep in mind that the transition of the EKF occurs if the relative distance between agent 1 and agent i at time step k, $d_{i,k}$, satisfies $d_{i,k} \leq \epsilon_2$ or $d_{i,k} \geq \Delta - \epsilon_2$.

4.2 Adjusting Motions of Agents to Minimize Estimation Errors

One improvement we have made to the EKF is that, in addition to gain optimization embedded in the EKF, motions of agents are controlled to minimize the mean square estimation error. At every N steps under the EKF, agent 1 adjusts the speed gain of agent i, k_c in $k_c\Gamma_i$, to minimize the error covariance predicted N steps forward in time.

Consider the system where agent i moves away from agent 1. A naive method to predict forward in time is to iterate equations from (15) to (19). Note that while predicting forward in time, we can not use measurement updates. Therefore, by setting $H = 0$ in EKF equations from (15) to (19), we obtain following equations.

$$\begin{aligned} \bar{\mu}_{s+1} &= F_1(\bar{\mu}_s), \\ \bar{F}_{1,s} &= \frac{\partial F_1(x)}{\partial x}|_{x=\bar{\mu}_s}, \\ \bar{P}_{s+1} &= \bar{F}_{1,s} \bar{P}_s \bar{F}_{1,s}^T + \bar{Q}_s, \end{aligned} \quad (20)$$

where predicted variables are denoted by a bar, and the index s is used to indicate prediction steps. Note that $\bar{\mu}_0 = \mu_k$ and that $\bar{P}_0 = P_k$ where k denotes the current time step in the EKF. To predict N steps forward in time, we iterate (20) N times.

Similarly, for the system where agent i moves toward agent 1, we obtain

$$\begin{aligned} \bar{\mu}_{s+1} &= F_2(\bar{\mu}_s), \\ \bar{F}_{2,s} &= \frac{\partial F_2(x)}{\partial x}|_{x=\bar{\mu}_s}, \\ \bar{P}_{s+1} &= \bar{F}_{2,s} \bar{P}_s \bar{F}_{2,s}^T + \bar{Q}_s. \end{aligned} \quad (21)$$

Using the measurement equation (11), we get $\bar{d}_{i,s} = H\bar{\mu}_s$ where $\bar{d}_{i,s}$ denotes the predicted estimate of the relative distance between agent 1 and agent i. Keep in mind that if $\bar{d}_{i,s} \leq \epsilon_2$ or $\bar{d}_{i,s} \geq \Delta - \epsilon_2$ during the prediction within N steps, then the propagation of error covariance must change, since (21) is different from (20).

Suppose that k_c is parameterized finitely, i.e., $k_c \in \{k_1, k_2, ..., k_M\}$[2]. To find k_c^* that minimizes the error covariance predicted N steps forward in time, we iterate equations (20) or (21) N times while changing k_c from k_1 to k_M. In other words, brute-force search is applied to find the adaptive gain $k_c^* \in \{k_1, k_2, ..., k_M\}$.

Once the adaptive gain $k_c^* \in \{k_1, k_2, ..., k_M\}$ is found using brute-force search, then agent 1 updates k_c to k_c^*. At the same time, agent 1 transmits k_c^* to agent i so that the speed of agent i changes to $k_c^*\Gamma_i$.

4.3 Implementation Issues

Prediction process to obtain k_c^* must be done in real-time, which implies that the overall prediction process is done within a single sampling period. However, as M increases, it takes longer to obtain k_c^* using brute-force search.

Note that iterating (20) or (21) N times while setting k_c as k_{m_1} is independent of iterating (20) or (21) while setting k_c as k_{m_2} ($\forall m_2 \neq m_1$). This implies that the prediction process to find an adaptive gain k_c^* in $\{k_1, k_2, ..., k_M\}$ can be performed in a parallel manner if M computing devices are utilized [12].

Suppose that agent 1 is equipped with M processing devices. Then each processing device, say $proc_m$ ($1 \leq m \leq M$), performs the prediction N steps forward in time while setting k_c as k_m.

Comparing predicted error covariances $\bar{P}_N$ obtained using M processing devices, we can find k_c^* that minimizes the error covariance predicted N steps forward in time.

5 Simulation Results

The EKF using the fixed k_c is compared with the EKF using adaptive k_c in MATLAB simulations.

[2]Parametrization of input can be found in literature on nonlinear model predictive control [1, 8].

In MATLAB simulations, we select the physical parameters of a battery to be $\alpha = 40375$ and $\lambda = 0.04$. These parameters were also used in [17, 11]. g in (3) is set to be 100. To estimate the battery level of agent i at each step k, (5) is applied to the states estimated using the EKF.

Fig.2 shows the EKF by fixing k_c as 50. The estimated battery level is shown in red, and the true battery level is shown in blue. Initially, the difference between the estimated battery level and the true battery level is 1.5. However, after running the EKF within 17 time steps, the estimated battery level converges to the true battery level.

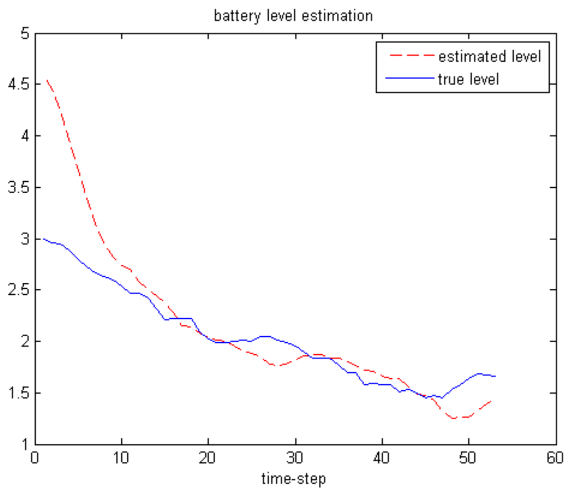

Figure 2. The EKF using fixed k_c.

Fig.3 shows the EKF by adjusting the gain k_c in $k_c\Gamma_i$. At every 5 steps under the EKF, agent 1 adjusts k_c to minimize the error covariance predicted 5 steps forward in time. k_c is parameterized as $k_c \in \{1, 1.1, 1.2, ..., 100\}$. Initially, the difference between the estimated battery level and the true battery level is 1.5. However, after running the EKF within 6 time steps, the estimated battery level converges to the true battery level. We can see that using adaptive k_c, the estimated battery level converges to the true battery level faster than the case where the fixed gain is used.

Fig.2 shows that the estimated battery level converges to the true battery level after the true battery level decreases by almost 0.8 from the initial battery level 3. In contrast, Fig.3 shows that the estimated battery level converges to the true battery level after the true battery level decreases by almost 0.2 from the initial battery level 3. This implies that using adaptive gain in the EKF decreases the power consumption required for battery level estimation.

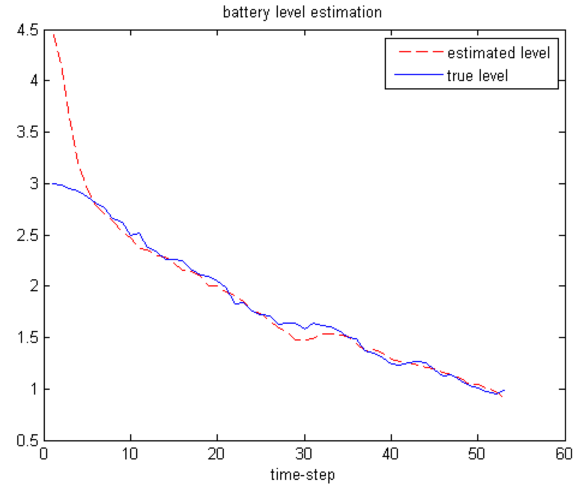

Figure 3. The EKF using adaptive k_c.

6 Conclusions

This paper introduces the nonlinear system model that integrates the particle motion model and the dynamic battery model. Then the EKF is applied to the nonlinear system model to estimate the battery level of every agent. One improvement we have made to the EKF is that, in addition to gain optimization embedded in the EKF, motions of agents are controlled to minimize the mean square estimation error. We have presented battery level estimation as an example where a co-design approach for cyber-physical systems is followed by integrating motion, battery level estimation, and communication.

Acknowledgment

The research is supported by ONR grants N00014-08-1-1007 and N00014-09-1-1074, and NSF grants ECCS-0841195 and CNS-0931576.

References

[1] F. Allgöwer, R. Findeisen, and Z. K. Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs.*, 35:299–315, 2004.

[2] C.-F. Chiasserini and R. Rao. Energy efficient battery management. *IEEE Journal on Selected Areas in Communications*, 19(7):1235–1245, 2001.

[3] T. Eren, W. Whiteley, B. D. O. Anderson, A. S. Morse, and P. N. Belhumeur. Information structures to secure control of rigid formations with leader-follower architecture. In *Proc. of American Control Conference*, pages 2966–2971, Portland, Oregon, 2005.

[4] M. Ji and M. Egerstedt. Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23:693–703, 2007.

[5] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa. Containment control in mobile networks. *IEEE Transactions on Automatic Control*, 53:1972–1975, 2008.

[6] M. Ji, A. Muhammad, and M. Egerstedt. Leader-based multi-agent coordination: Controllability and optimal control. In *Proc. of American Control Conference*, pages 1358–1363, Minneapolis, MN, 2006.

[7] D. Linden and T. Reddy. *Handbook of Batteries*. McGraw-Hill, 2001.

[8] L. Magni, D. M. Raimondo, and F. Allgöwer. *Nonlinear Model Predictive Control : Towards New Challenging Applications*. Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 2009.

[9] S. Martínez, J. Cortés, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4):75 – 88, 2007.

[10] D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, and K. Lahiri. Battery life estimation of mobile embedded systems. In *Proc. of VLSI Design*, pages 57–63, Bangalore, India, 2001.

[11] D. Rakhmatov., S. Vrudhula, and D. A. Wallach. A model for battery lifetime analysis for organizing applications on a pocket computer. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11:1019–1030, 2004.

[12] J. H. Reif. *Synthesis of Parallel Algorithms*. Morgan Kaufmann, 1993.

[13] H. Tanner, G. J. Pappas, and V. Kumar. Leader to formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, 2004.

[14] F. Zhang, D. M. Fratantoni, D. Paley, J. Lund, and N. E. Leonard. Control of coordinated patterns for ocean sampling. *International Journal of Control*, 80:1186–1199, 2007.

[15] F. Zhang and N. E. Leonard. Coordinated patterns of unit speed particles on a closed curve. *Systems and Control Letters*, 56:397–407, 2007.

[16] F. Zhang and N. E. Leonard. Cooperative control and filtering for cooperative exploration. *IEEE Transactions on Automatic Control, accepted June 2008 (in print)*, 2008.

[17] F. Zhang, Z. Shi, and W. Wolf. A dynamic battery model for co-design in cyber-physical systems. In *Proc. of 2nd International Workshop on Cyber-Physical Systems*, pages 51–56, Montereal, Canada, 2009.