

Optimizing Sensor Movement Planning for Energy Efficiency *

Guiling Wang, Mary Jane Irwin, Piotr Berman, Haoying Fu, and Tom La Porta
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802, USA
{guiwang, mji, berman, hfu, tlp}@cse.psu.edu

ABSTRACT

Conserving the energy for motion is an important yet not-well-addressed problem in mobile sensor networks. In this paper, we study the problem of optimizing sensor movement for energy efficiency. We adopt a complete energy model to characterize the entire energy consumption in movement. Based on the model, we propose an optimal velocity schedule for minimizing energy consumption when the road condition is uniform; and a near optimal velocity schedule for the variable road condition by using continuous-state dynamic programming. Considering the variety in motion hardware, we also design one velocity schedule for simple microcontrollers, and one velocity schedule for relatively complex microcontrollers, respectively. Simulation results show that our velocity planning may have significant impact on energy conservation.

Categories and Subject Descriptors

C.4 [Performance of Systems]: Design studies

General Terms

Algorithms, Design

Keywords

Mobile Sensor, Energy-Efficiency

1. INTRODUCTION

As an emerging class of wireless sensor networks, mobile sensor networks have been attracting increasing research interest recently [4, 10, 12, 20] since they can solve or alleviate many design challenges in sensor networks. To name just a few, mobile nodes can self-deploy to guarantee required

coverage, relocate themselves when network failure happens, and aggregate around critical areas to increase sensing quality.

Bringing many benefits, augmenting sensor nodes with motion capability makes the core challenge of sensor networks, *energy efficiency*, even more crucial. In a mobile node, mechanical actuation has much higher power consumption than communication, sensing and computation [6]. Therefore, energy efficient movement is very important for mobile nodes to increase their lifetime when energy recharge is difficult.

Unfortunately, there is few work on conserving the actuation energy for mobile sensors. In the robotics field, there have been works on energy efficient motion planning. However, most of them consider only mechanical energy required for movement, instead of entire energy consumption for actuation from the battery. For example, in [15, 16, 18], only energy loss due to friction and gravity is considered. Mei *et al.* [14], modeled the power consumption from the battery as a polynomial of motor's angular velocity, but neglected the effect of acceleration. In addition, they did not consider the effect of the road conditions on the motion planning. This is a valid approximation in robotics field. But in mobile sensor networks, it is quite possible that mobile nodes move a short distance, stop and move again. Further, since they are likely to be deployed outdoors, mobile nodes may experience different road conditions during the course of movement. Both the effect of acceleration and the variable road condition must be taken into account for a motion planning of sensors.

In this paper, we study the problem of minimizing the entire actuation energy consumption in mobile sensors. To accurately characterize this energy consumption, we adopt a complete energy model, which includes not only the required mechanical energy for accomplishing the movement command, but also various energy dissipation due to acceleration, heating, viscous damping, internal motor friction, etc. Different from previous models, this model considers the entire energy consumption from the energy source, e.g., battery, to accomplish the moving requirement of the application. Based on this model, we propose an optimal velocity schedule for mobile nodes when the road condition is uniform. For sensors moving under variable road condition, we propose to use continuous-state dynamic programming to calculate a near optimal velocity schedule. We can postulate a desired approximation level. We propose a method to minimize the computation complexity to achieve the de-

*This work was supported in part by a grant from MARCO/DARPA Gigascale Systems Research Center.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'05, August 8–10, 2005, San Diego, California, USA
Copyright 2005 ACM 1-59593-137-6/05/0008 ...\$5.00.

sired approximation. We also study the variety in motion hardware. Specifically, we design one velocity schedule for simple microcontrollers, in which the acceleration can not be specified and one velocity schedule for relatively complex microcontrollers, in which acceleration can be specified. Simulation results show the effectiveness of our velocity planning.

The rest of the paper is organized as follows. Section 2 describes the related work. In section 3, we introduce a technical preliminary on actuation. We present the energy efficient motion planning in section 4 and draw conclusions in section 5

2. RELATED WORK

There have been a lot of research efforts on mobile sensor networks. The deployment of mobile sensors has been addressed in [11, 20]. In [10], we proposed to utilize sensor networks composed of both static and mobile sensors to achieve a balance between coverage and sensor cost. Using mobile sensors to do network repair was addressed in [12]. Butler *et al.* [4], proposed an algorithm to aggregate sensors around sensitive areas to increase the sensing quality. Dantu *et al.* [6], have developed a mobile sensor prototype, *Robomote*, for experimental usage. All these works utilize the motion capability of sensors, but none of them addresses the issue of conserving energy by controlling the kinetic aspects of motion. On the other hand, it is recognized that conserving energy in sensor networks is very important. Low power design in static sensor networks has been intensively studied [5, 9, 19].

Energy efficient motion planning has been addressed in the robotics field. Sun and Reif [18] worked on finding the most energy efficient path from a source point to a destination point. They assumed there was no acceleration and turning during the movement and adopted the energy model in which only energy loss due to friction and gravity was considered. A similar model is also used in [15, 16]. This model considers only the output mechanical energy of the motor, instead of the real energy consumption from the battery. In [14], power consumption is modeled as a polynomial of motor's angular velocity, neglecting the effect of acceleration. Unlike mobile robots, which may move continuously performing certain tasks like carpet cleaning, mobile sensor may move only a short distance, stop and move again for application's needs. In this way, acceleration and deceleration must be taken into consideration. Energy efficient motion planning has also been addressed for walking robots with legs[8]. But this moving mechanism is not likely to be adopted for mobile sensor for their relatively high cost.

3. TECHNICAL PRELIMINARY

Generally, the motion base of a mobile node is composed of three main parts: (1) driving/moving devices, e.g., the wheels; (2) motors, which transfer the electrical energy to mechanical energy and drive the wheels; (3) microcontroller, which controls mobile sensors to move as desired.

The driving devices include wheels, caterpillars and walking legs. The current mobile sensor prototype [17] is wheel-based. Many small mobile robot platforms, which are likely to be adopted in mobile sensors, use wheels as well for their low cost and easy manipulation. In this paper, we consider wheeled mobile sensors.

Motors transform electrical energy supplied by the bat-

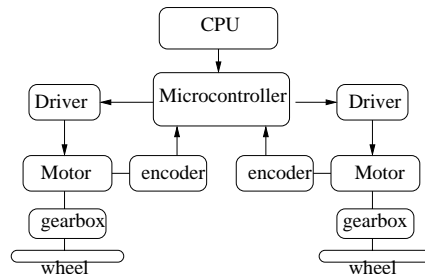


Figure 1: Architecture of the Motion Base

tery to mechanical energy needed to rotate the wheels and run the mobile sensors. The angular velocity of a motor determines that of the wheel it drives, and consequently determines the velocity of the mobile sensor. The output power of a motor is the product of its angular velocity and the torque it applies to run the wheel. The higher the torque, the heavier the load that a motor can support. For example, in a rough ground, a higher torque is needed to run a mobile sensor than that in a smooth ground. Normally, there is a gearbox between a motor and the wheel it drives, which can reduce the angular velocity and increase the torque supplied. In this way, with the same output power, more load can be supported.

By adjusting the voltage applied to a motor, it can be accelerated or decelerated. To make a motor rotate at a desired speed or with a desired acceleration, a feedback PID (Proportional Integration Derivation) controller is normally used to calculate the voltage which should be applied to the motor. The feedback controller takes the current velocity of the motor as input, computes its difference to the desired velocity, and calculates the best voltage to minimize this difference. This best voltage is expressed as a logical signal and will be amplified by the motor driver. Then the real desired voltage will be supplied to the motor. The current velocity of a motor is monitored by an encoder. This whole feedback control process is taken rather frequently, depending on the accuracy of the encoder. The relationship between these parts in the motion base is shown in Figure 1.

3.1 Movement Mechanism and Kinematics

We consider mobile sensors with differential drives, the same kind as Robomote [6], the mobile sensor prototype mentioned above. Small mobile robot platform, khepera [1] and FIRA [13] are also of this kind. Differential drive moves two coaxial wheels with two independent electric motors. The moving direction and velocity of a mobile sensor is determined by the rotation speed of these two drive wheels. Let ν be the velocity in the forward direction and ω_s be the angular velocity around the ICR (Instantaneous Center of Rotation) of the mobile sensor, ω_1 and ω_2 be the angular velocity of two wheels, r be the radius of the wheel, and d be the distance between the two wheels. ν and ω_s has the following relationship with ω_1 and ω_2 [13]:

$$\begin{cases} \nu = r \left(\frac{\omega_1 + \omega_2}{2} \right) \\ \omega_s = r \left(\frac{\omega_1 - \omega_2}{d} \right) \end{cases} \quad (1)$$

There are other kinds of wheeled movement mechanisms, including omnidirectional movement and car-like nonholonomic mechanism. For those kinds of motion mechanism, ν

and ω_s can also be written as a linear combination of the angular velocities of the wheels, as in equation (1). Our method of energy efficient motion planning can be easily extended for the usage in those cases.

4. ENERGY EFFICIENT MOTION PLANNING

We deal with the problem of energy-efficient motion planning of a mobile sensor given a movement task from the application: “move to target location D ”. To fulfill this task, the mobile sensor needs to determine the path to the target location and the velocity and acceleration along the path.

Suppose the distance between a sensor’s current location and its target location is L , and the heading position of this mobile sensor is θ with respect to its target location. To make a short moving path, the mobile sensor can first turn to the direction of its target location and then move straightly to its target location. According to equation (1), a mobile sensor can turn by making its two drive wheels spin in the opposite directions with the same rate and can move straight by making its wheels spin in the same directions with the same rate. Since the wheels spin at the same rate when turning, the energy consumed is the same as moving straight by a distance $\theta * d/2$. Therefore, we concentrate on how to determine a velocity schedule of motors (which is proportional to that of the wheels), such that the energy consumption is minimized. In the following sections, we first introduce the power model of a motor. Then we describe how to determine the velocity schedule under a constant load (uniform road condition). Finally, we present the continuous-state dynamic programming method to calculate the velocity schedule under variable load.

4.1 Power Model of Motion

To accurately model the energy consumption for motion, we consider not only the mechanical energy required for movement, but also the energy dissipation inside the motor due to various reasons, such as internal friction, etc. We choose the energy model described in [7]:

$$P(t) = V(t)I(t); \quad (2)$$

$$V(t) = RI(t) + K_e\omega(t); \quad (3)$$

$$I = \frac{1}{K_T}[(J_m + J_L)\frac{d\omega(t)}{dt} + T_L(t) + T_f + D\omega(t)] \quad (4)$$

Here, $P(t)$ is the power consumption. $V(t)$ is the voltage supplied to the motor and $I(t)$ is the current flow through the rotor of the motor. R is the armature resistance. K_e is the back EMF (electromotive force) constant. $\omega(t)$ is the angular velocity of the motor at time t . K_T is the torque constant of the motor. J_m and J_L is the inertia of motor and load, respectively. $T_L(t)$ is the load torque at time t . T_f is the friction torque of the motor. D is the viscous damping, which represents the coefficient of speed-dependent power dissipation.

4.2 Energy Efficient Velocity Planning with Constant Load

As shown in our power model, the energy consumed in a motion is determined by many factors. Among them, K_e , K_T , D , T_f and J_m are the characteristics of the motor. J_L

is a characteristic of the mobile sensor; load torque T_L is determined by the environments in which sensor moves, e.g, the surface friction; angular velocity is specified by the users; depending on the complexity of the microcontroller, accelerations may also be determined by the users. We aim to calculate a velocity schedule optimal for energy consumption in case (a) only velocity can be specified and case (b) both velocity and acceleration can be specified.

In this section, we consider a constant load torque during the course of movement, which corresponds to the assumption that the sensor moves over a flat ground with uniform friction. In the next section, we address the velocity planning under variable load torque.

Since unnecessary accelerations cost more energy, the optimal velocity schedule should be accelerating to ω , moving uniform at ω , and decelerating to zero at the target location. We will find the best ω for minimizing the energy consumption if we are not allowed to set the acceleration α_a and deceleration α_d ; and find the best triple $\langle \alpha_a, \omega, \alpha_d \rangle$ if we are allowed to do so.

The energy consumption for a mobile sensor to move distance L is the sum of the input energy of the two drive motors, which can be expressed as $E(L) = 2 \int_0^T P(t) dt$. T is the time for movement, which is a function of ω , α_a , α_d and L . Let T be $f_t(\omega, \alpha_a, \alpha_d, L)$. Then the energy consumption is

$$E(L) = 2 \int_0^{f_t(\omega, \alpha_a, \alpha_d, L)} P(t) dt. \quad (5)$$

4.2.1 Calculating Optimal ω

By computation, $E(L)$ can be expressed as a polynomial of ω , α_a , α_d . The detailed calculation and expression of $E(L)$ is shown in Appendix A. In case we can not specify α_a and α_d to the microcontroller, we want an ω that minimizes $E(L)$. The necessary condition for ω to be a local minimizer of $E(L)$ is $\frac{dE(L)}{d\omega} = 0$. By taking the derivative of $E(L)$ with respect to ω , we get a polynomial equation of degree four. Polynomial equations of degree four can be solved analytically in constant time.

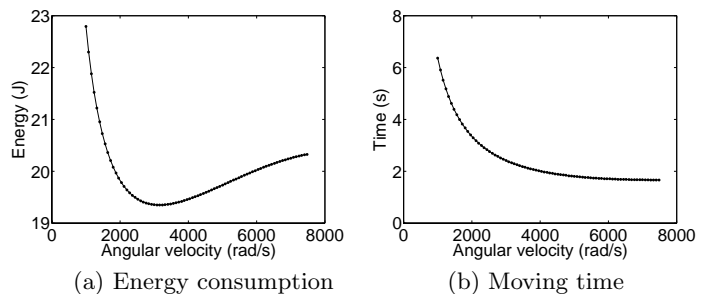


Figure 2: Impact of ω

In Figure 2, we plot the energy consumption and the moving time for a mobile sensor to move 5 meters as the angular velocity varies. The acceleration/deceleration is set to $9000rad/s^2$. The load torque is 10 times the friction torque. For simulating the behavior of a motor, we use the data of a DC Micromotor (Serious 1319 012S) from Micromo Electronics [2]. For other parameters, as gear ratio, we use those compatible with with *Robomote* and *Khepera*. The detailed value of each parameter is shown in Appendix B. From the figure, we can see that, by choosing an appropriate velocity,

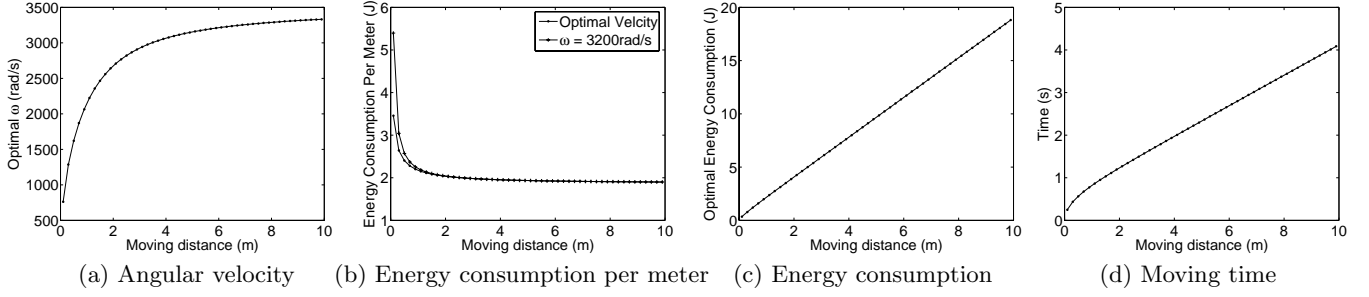


Figure 3: Impact of moving distance

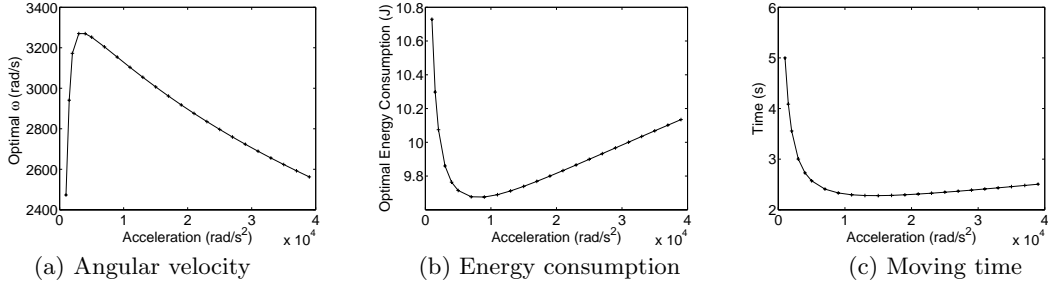


Figure 4: Impact of Setting Acceleration/Deceleration

energy can be saved significantly. For example, by setting ω at its optimal value, $0.7J$ can be saved compared with setting it to $6000rad/s$, and the saved energy can be used to send hundreds of messages.

In Figure 3, we plot the optimal value of ω for energy efficiency, the corresponding energy consumption, and moving time, as the moving distance varies. From 3(a), we can see that the optimal angular velocity is not linearly related to the moving distance. Instead, optimal velocity approaches a fixed value when the moving distance increases. We choose an ω , $3200rad/s$, which is compatible to the optimal velocity for moving distance longer than $2.5m$, and plot the energy consumption per meter under this velocity and the optimal velocity in Figure 3(b). As shown in the figure, the energy consumption per meter is almost constant for moving distance longer than 2 meters. We can see that, if the sensor moves on a uniform surface and the movements over very small distances are rare, we can obtain very satisfactory results with a fixed ω . Otherwise, calculating optimal velocity schedule can achieve significant savings of energy, on the order of even 40 – 50%.

As shown in Figure 3(c), the optimal energy consumption has a near-linear relationship with the moving distance. Also, when the moving distance approaches zero, the optimal energy consumption approaches a number greater than zero, which means there is always a “startup” energy. Figure 3(d) shows the moving time under optimal velocity. We can see that this moving time also has a near-linear relationship with the moving distance and there is a “startup” time.

4.2.2 Calculating Optimal $\langle \alpha_a, \omega, \alpha_d \rangle$

Some microcontrollers, such as the MCDC2805 motion controller from *Micromo Electronics*, allow users to specify acceleration and deceleration in addition to the angular velocity. In this case, we want to optimize triplet $\langle \alpha_a, \omega, \alpha_d \rangle$

to minimize $E(L)$. The necessary condition for this triplet to be a minimizer of $E(L)$ is as follows:

$$\begin{cases} \frac{\partial E(L)}{\partial \omega} = 0 \\ \frac{\partial E(L)}{\partial \alpha_a} = 0 \\ \frac{\partial E(L)}{\partial \alpha_d} = 0 \end{cases} \quad (6)$$

By taking the partial derivative of $E(L)$ with respect to ω , α_a , and α_d , we get a system of three equations. Using two of these three equations, α_a and α_d can be eliminated. In fact, α_a and α_d have the same magnitude. After the elimination, we get a polynomial equation of degree eight in ω . This equation can be solved by computing the eigenvalues of the companion matrix [3]. The computation complexity is $O(n^2)$, where n is the degree of the polynomial (here $n = 8$).

Figure 4 plots the optimal angular velocity for a sensor to move 5 meters when α varies (here we choose $\alpha_d = \alpha_a$), the corresponding energy consumption and moving time. From the figure, we can see that an optimally chosen α can not only reduce the energy consumption significantly, but also reduce the moving time.

Figure 5 shows the optimal ω and α ($\alpha_a = \alpha_d$ in the optimal setting) for energy efficiency, the achieved energy consumption under such setting, and the corresponding movement time, under different moving distance. We can see, by properly setting ω and α , the energy consumption and the moving time is approximately linear to the moving distance.

4.3 Energy Efficient Velocity Planning With Variable Load

In this section, we consider the situation that load torque changes during movement. When load torque changes, the format of the velocity schedule for constant load, *i.e.*, accelerating, moving uniformly, and decelerating, may not be an optimal solution as in the scenario of constant load torque. As shown in Figure 6, the optimal angular velocity and acceleration for energy consumption change when the load torque

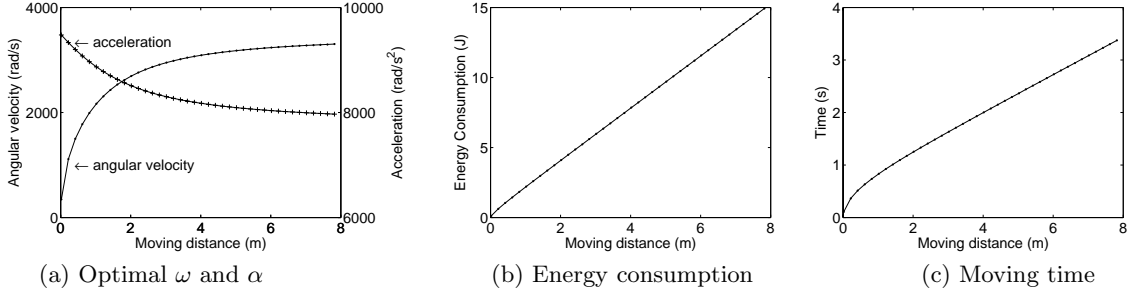


Figure 5: Optimal setting of $\langle \alpha_a, \omega, \alpha_d \rangle$

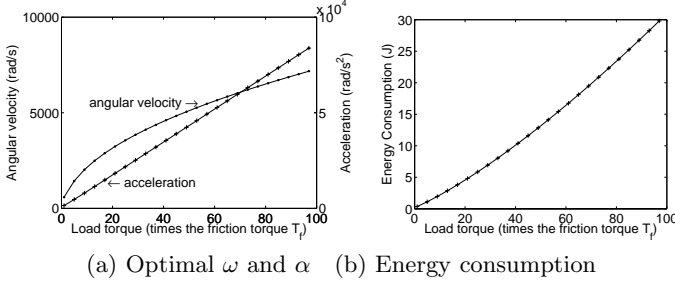


Figure 6: Impact of load torque

varies. In these situations, minimization of energy consumption may require additional accelerations in the middle of movement.

As explained in section 4.2, frequent accelerations should be avoided. In addition to starting and stopping, accelerations should only occur around the place where load torque changes. The whole path with variable condition can be segmented into S_1, S_2, \dots, S_m , such that each segment has uniform load condition. For example, a mobile node moves from a concrete road onto the dirt field. Its path can be segmented into two parts. In segment S_i with uniform load torque, mobile node will have velocity schedule $\langle \bar{\omega}_i, \alpha_i, \omega_i, \alpha'_i, \bar{\omega}_{i+1} \rangle$, where $\bar{\omega}_i$ and $\bar{\omega}_{i+1}$ are the initial and final speed, which are the transition speed between segment S_{i-1} and S_i , and between S_i and S_{i+1} , ω_i is the uniform velocity in segment S_i , α_i is the initial acceleration and α'_i is the final acceleration. Therefore, the velocity schedule for a path with m segments should have the following format:

$$\langle 0 = \bar{\omega}_1, \alpha_1, \omega_1, \alpha'_1, \bar{\omega}_2, \dots, \bar{\omega}_m, \alpha_m, \omega_m, \alpha'_m, \bar{\omega}_{m+1} = 0 \rangle$$

Certainly, it is quite possible that α_i or α'_i is equal to zero and $\bar{\omega}_i$ is equal to ω_{i-1} or ω_i in the optimal velocity planning.

An optimal velocity schedule can be obtained by calculating the total energy consumption expressed in terms of the variables described above, taking partial derivatives, and constructing and solving a system of equations. However, this method is not scalable since the system of equations changes when the number of segments changes.

To achieve scalability, we propose a dynamic programming to optimize velocity schedule. In our dynamic programming problem, the transition velocity between segments is the "state variable". Given the transition velocity of one segment, we can calculate the optimal velocity schedule for minimizing energy consumption in this segment by using a similar method as in section 4.2. Let the function to do

this calculation be G , where $\langle \alpha_i, \omega_i, \alpha'_i \rangle = G(\bar{\omega}_i, \bar{\omega}_{i+1}, S_i)$ (argument S_i provides the length and load torque of this segment), and $E(\bar{\omega}_i, \bar{\omega}_{i+1}, S_i)$ is the respective energy consumption. Let $Ec(\bar{\omega}, i)$ be the minimum energy consumption for segments S_1, S_2, \dots, S_i , assuming that the final angular velocity equals $\bar{\omega}$. We have the following recurrence:

$$Ec(\bar{\omega}, i) = \begin{cases} E(0, \bar{\omega}, 1), & \text{if } i = 1 \\ \min_{\bar{\omega}_p} (Ec(\bar{\omega}_p, i-1) + E(\bar{\omega}_p, \bar{\omega}, S_i)), & \text{otherwise.} \end{cases}$$

Calculating $Ec(\bar{\omega}, 1), Ec(\bar{\omega}, 2), \dots, Ec(\bar{\omega}, i-1)$ in sequence, we obtain the minimized $Ec(0, m)$ and the corresponding velocity schedule. This actually is a process of finding the transition velocity $\bar{\omega}_i$ (for $i = 2, \dots, m$) to minimize the total energy consumption. However, our problem is a *continuous-state* dynamic programming problem since velocity is a continuous variable and there are infinite possible values. We need to discretize the velocity space, such that the finite-state dynamic programming can be applied. In this way, we can obtain the approximately optimal solution.

Suppose the space of the angular velocity is $\{0, \omega_{max}\}$. One possible discretization is to choose a number n and discretize the velocity space into $\{0, \omega_{max}/n, 2\omega_{max}/n, \dots, (n-1)\omega_{max}/n, \omega_{max}\}$. For this discretization, we need to run function G $m(n+1)^2$ times, for a fineness of ω_{max}/n . Suppose ω_{max} is 17000 rad/s and we want a fineness of 10 rad/s . Then n is 1701 and we need to run function G $1701^2 m$ times. To reduce the computation complexity, we discretize the velocity space iteratively. We use a small n in each step, discretize the velocity space into $(n+1)$ grid points and get the best transition velocities within these grid points. Then we reduce velocity space into $2/n$ of the original space centered at the calculated transition velocity. After l iterations, we obtain the fineness of $\omega_{max} * (2/n)^l$ and run the function G $m * l * (n+1)^2$ times.

The remaining task is to find a n such that we can minimize the computation cost for the same fineness. That is, we want to find a n which can get a finer discretization with the same computation. Let x be the number of evaluations of function G , and $f(x)$ be the fineness of the discretization. In each iteration, we do $m(n+1)^2$ computations, and reduce the velocity space into $2/n$ of the original. After l iterations, we will have executed function G $lm(n+1)^2$ times and reduced the velocity space into $(2/n)^l$ of the original space. Here, $x = lm(n+1)^2$ and $f(x) = (2/n)^l$. Then $f(x)$ can be expressed as:

$$f(x) = \left[\left(\frac{2}{n} \right)^{\frac{1}{(n+1)^2}} \right]^{\frac{x}{m}} \quad (7)$$

Therefore, we want to minimize $\left(\frac{2}{n} \right)^{\frac{1}{(n+1)^2}}$ to obtain small-

est velocity space with the same computation complexity. When $n = 4$, we get the best solution. For the example mentioned above, by setting n to four, we get a fineness of $8.3rad/s$ after 11 iterations, and run the function G 275m times, as compared with 1701^2m times execution of function G for a fineness $10rad/s$.

5. CONCLUSION

In this paper, we address the problem of minimizing movement-related energy consumption. We adopt a complete energy model. Based on the model, we propose approaches for calculating the velocity schedule to minimize the energy consumption for mobile nodes equipped with various kinds of motion controllers and moving in various load situations. Experimental results show that energy saving is significantly by using the velocity schedule.

6. REFERENCES

- [1] <http://www.k-team.com/robots/khepera/index.html>.
- [2] <http://www.micromo.com>.
- [3] A. Edelman and H. Murakami. Polynomial Roots from Companion Matrix Eigenvalues. *Mathematics of Computation*, 64:763–776, 1995.
- [4] Z. Butler and D. Rus. Event-Based Motion Control for Mobile Sensor Networks. *IEEE Pervasive Computing*, 2(4):34–43, 2003.
- [5] T. Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Network. *SensSys*, 2003.
- [6] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal and G. S. Sukhatme. Robomote: Enabling Mobility in Sensor Network. *CRES-04-006 CRES Tech Report*.
- [7] Electro-Craft Corporation. *DC Motors, Speed Controls, Servo Systems, An Engineering Handbook*. Pergamon Press, 1977.
- [8] Y. Fujimoto. Trajectory Generation of Biped Running Robot with Minimum Energy Consumption. *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.
- [9] S. Gregori, Y. Li, H. Li, J. Liu and F. Maloberti. 2.45 GHz power and data transmission for a low-power autonomous sensors platform. *In Proc. of ISLPED*, 2004.
- [10] Guiling Wang, Guohong Cao and Tom La Porta. A Bidding Protocol for Deploying Mobile Sensors. *The 11th IEEE International Conference on Network Protocols (ICNP)*, November 2003.
- [11] Guiling Wang, Guohong Cao and Tom La Porta. Movement-Assisted Sensor Deployment. *INFOCOM*, March 2004.
- [12] Guiling Wang, Guohong Cao, Tom La Porta and Wensheng Zhang. Sensor Relocation in Mobile Sensor Networks. *INFOCOM*, March 2005.
- [13] J. -H. Kim, D. -H. Kim, Y. -J. Kim and K. -T. Seow. *Soccer Robotics*. Springer, 2004.
- [14] Y. Mei, Y. H. Lu, C. Lee and Y. C. Hu. Energy-Efficient Motion Planning for Mobile Robots. *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.
- [15] N. C. Rowe. Obtaining Optimal Mobile-Robot Paths with Non-Smooth Anisotropic Cost Functions Using Qualitative-State Reasoning. *International Journal of Robotics Research*, 16(3):375–399, June 1997.
- [16] N. C. Rowe and R. S. Ross. Optimal Grid-Free Path Planning Across Arbitrarily Contoured Terrain with Anisotropic Friction and Gravity Effects. *IEEE Transactions on Robotics and Automation*, 6(5):540–553, Oct. 1990.
- [17] G. T. Sibley, M. H. Rahimi and G. S. Sukhatme. Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks. *2002 IEEE International Conference on Robotics and Automation*, 2002.

- [18] Z. Sun and J. Reif. On Energy-minimizing Paths on Terrains for a Mobile Robot. *IEEE International Conference on Robotics and Automation (ICRA)*, September 14-19 2003.
- [19] Y. Zou and K. Chakrabarty. Energy-Aware Target Localization in Wireless Sensor Networks. *Proc. IEEE Int. Conf. Pervasive Computing and Communications*, 2003.
- [20] Y. Zou and K. Chakrabarty. Sensor Deployment and Target Localization Based on Virtual Forces. *INFOCOM*, 2003.

Appendix A

First, we describe some notations. t_{acc} is the time used to accelerate, t_{con} is the time of moving uniformly at ω , and t_{dec} is the time of deceleration. d_{acc} , d_{con} and d_{dec} are the moving distances in these three phases, respectively. v is the speed of the mobile sensor and N_r is the gear ratio of the gear box. $v = r\omega/N_r$.

The time T to finish the movement can be expressed as

$$T = t_{acc} + t_{con} + t_{dec}, \text{ where}$$

$$t_{acc} = \omega/\alpha_a; d_{acc} = \frac{1}{2} \cdot \frac{\omega}{\omega_a} \cdot \frac{r\omega}{N_r}$$

$$t_{dec} = \omega/\alpha_d; d_{dec} = \frac{1}{2} \cdot \frac{\omega}{\omega_d} \cdot \frac{r\omega}{N_r}$$

$$t_{con} = d_{con}/v = (L - d_{acc} - d_{dec})/v = \frac{N_r * L}{r * \omega} - \frac{\omega}{2\alpha_a} - \frac{\omega}{2\alpha_d} \quad (8)$$

Plugging equation (3), (4) and (8) into equation (5), we get

$$E(L) = b_3 * w^3 + b_2 * w^2 + b_1 * w + b_0 + b_{-1} * w^{-1}, \text{ where}$$

$$b_3 = (-1/(6 * \alpha_a) - 1/6\alpha_d)((RD^2 + K_e K_T D)/K_T^2$$

$$b_2 = (K_e K_T + 2RD)(\alpha_a J + T_L + T_f)/(2\alpha_a K_T^2) +$$

$$(K_e K_T + 2RD)(\alpha_d J + T_L + T_f)/(2\alpha_d K_T^2) -$$

$$(K_e K_T + 2RD)(T_L + T_f)/(2\alpha_d K_T^2 + 2\alpha_a K_T^2)$$

$$b_1 = R(\alpha_a J + T_L + T_f)^2/(\alpha_a K_T^2) +$$

$$R(\alpha_d J + T_L + T_f)^2/(\alpha_d K_T^2) -$$

$$R(T_L + T_f)^2(1/(2\alpha_a) + 1/(2\alpha_d))/K_T^2 +$$

$$(RD^2 + K_e K_T D)N_r L/rK_T^2$$

$$b_0 = (K_T K_e + 2RD)(T_L + T_f)N_r L/rK_T^2$$

$$b_{-1} = N_r LR(T_L + T_f)^2/r$$

Appendix B: Experimental Setting

For simulating the behavior of motor, we use the data of a DC Micromotor (Serious 1319 012S) from *Micromo Electronics*, which are listed below.

Table 1: Parameters of the Motor

R	22 Ω
J_m	$3.824 * 10^{-6}$ z-in-sec ²
K_e	0.682 mv/rpm
K_T	0.923 oz-in/A
T_f	0.014 oz-in

The viscous damping D is not provided from the datasheet of this Micromotor. We set D to be $4.85 * 10^{-8}$ Nm/(rad/s), which is compatible to similar DC micromotors. We use the same gear ratio, 25 : 1, as the mobile sensor prototype developed by USC, Robomote. Since the load inertia after the gear reduction should be smaller than 5 times the load inertia, here we choose it to be three times the motor inertia, which is $1.1472 * 10^{-5}$ z-in-sec². The wheel radius is set to be $0.02m$ and the distance between the two drive wheels is $0.05m$, which are compatible with *Robomote*, *Khepera* and *Mica*.