Low Power Management for Autonomous Mobile Robots Using Optimal Control

Wei Zhang and Jianghai Hu
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907, USA.
{zhang70, jianghai}@purdue.edu

***Abstract*— This paper studies an approach of minimizing the energy consumption of a mobile robot by controlling its traveling speed and its on-board processor's frequency simultaneously. The problem is formulated as a discrete-time optimal control problem with a random terminal time and probabilistic state constraints. A general solution procedure suitable for arbitrary power functions of the motor and the processor is proposed. Furthermore, for a class of practically important power functions, the optimal solution is derived analytically. Simulation result shows that the proposed method can save up to 30% energy compared with some heuristic schemes.**

I. Introduction

Autonomous mobile robots are finding numerous applications, such as surveillance, transportation, environment sensing, and rescue [1], [2], [3]. In these applications, energy conservation is a crucial issue due to the limited energy that a mobile robot can carry. Without demanding computations, the energy consumptions in a mobile robot are dominated by the driving motor. However, often times heavy computations, such as those required by path planning algorithms and complex sensing algorithms, must be performed during the motion of the robot; thus the energy consumed by the processor can not be ignored [4], [5], especially for small robots [1], [6], [7].

The power management of the processor and the speed control of the motor are usually studied separately. For the processor, one can schedule its frequency to reduce its power consumption for a given task. For the motor, one can adjust its speed to optimize the energy cost for traveling a certain distance. Since each optimization is performed independently, the result may not correspond to the global optimal solution for the overall system. For a mobile robot, the deadline for its on-board processor to finish a job usually depends on the speed of its motion. Thus the two optimization problems are actually coupled with each other. Due to this coupling, a joint power management scheme that optimizes the processor's frequency and the motor's speed simultaneously may result in a better power performance for the whole system. We call this coupled optimization problem for mobile robots the *joint speed control and power scheduling (JSP)* problem.

The JSP problem is a new and challenging problem for the low power design of a class of real-time systems. It is originally proposed in [5], where the best scheduling of the CPU's frequency and the motor's speed are obtained through an exhaustive search algorithm. The result is further extended in [8], where the generic algorithm is employed to expedite the search for the optimal solutions. The main drawbacks of the previous studies are: (a) the optimal solution can only be obtained heuristically without any guarantee on its optimality; and (b) the computation for finding the optimal solution is excessive. This paper studies the JSP problem from a different perspective. We perform a rigorous analysis on the problem, and derive the optimal solution analytically with guaranteed optimality and little computation effort. Specifically, the contributions of this paper are the following: (a) The JSP problem is formulated as a discrete time optimal control problem with a random terminal time and probabilistic state constraints; (b) The problem is transformed to a deterministic nonlinear optimization problem and a general procedure is proposed to solve it; (c) For a class of practically important power functions of the motor and the processor, the optimal solution to the JSP problem is obtained analytically. In summary, this paper gives a complete formulation and solution to the JSP problem and represents an important application of the optimal control theory in the low power design of real-time systems.

This paper is organized as follows. The JSP problem is introduced and formulated in Section II. The general procedure for solving the JSP problem is discussed in Section III. The optimal solution of a class of practically important JSP problem is derived analytically in Section IV. The effectiveness of our results are demonstrated through a numerical simulation in Section V. Finally, some conclusion remarks are given in Section VI.

II. Problem Formulations

Before formally formulating the JSP problem, we first illustrate its basic idea through an example. Suppose that a robot moves in an unknown environment to search for certain objects. It must recognize the objects before traveling a certain distance to avoid missing the targets or colliding with the obstacles. Different motor speeds correspond to different times to travel the given distance, thus resulting in different deadlines for the recognition task. On the other hand, for a given task, the optimal operation frequency and its corresponding optimal energy consumption of the processor usually depend on the given deadline. Hence, the motor's speed not only determines the motor's energy, but

This work was partially supported by the National Science Foundation under Grant CNS-0643805

also affects the optimal energy for the processor to finish the task. Thus optimizing the motor's speed and the processor's frequency simultaneously may result in a better performance for the overall system.

Motivated by the above example, we consider a general problem where a mobile robot needs to finish a given computation task T before traveling a certain distance D. Let w_m be the maximum number of CPU cycles needed for any computation task performed by the robot. Suppose that w_m is evenly divided into n bins, each of which contains $b = \frac{w_m}{n}$ cycles. Let L be the number of bins that the given task T demands. Then L must be an integer between 1 and n. During the execution of the k^{th} bin, assume that the processor operates at frequency $f(k)$ and the robot moves at speed $s(k)$. It is required that

$$s(k) \geq 0 \text{ and } f(k) > 0, \quad k = 1, \ldots, L. \tag{1}$$

In other words, the robot may stop during some bins to avoid running over the given distance D; whereas the frequency of the processor must be positive for each bin, as otherwise it will never complete the task T. The goal of the JSP problem is to find the best f and s so that the robot can finish the task T before crossing the distance D, and at the same time its energy consumption for traveling the whole distance D is minimized.

Since both s and f are assumed to be constant during the execution of the k^{th} bin, the time spent for computing the k^{th} bin is $\frac{b}{f(k)}$, and the distance traveled over this period is $\frac{bs(k)}{f(k)}$. Introduce a state variable $x(k)$ to represent the total distance that the robot has traveled up to the completion time of the k^{th} bin. Then the state evolves according to the following difference equation

$$x(k+1) = x(k) + \frac{bs(k+1)}{f(k+1)}, \quad \text{with } x(0) = 0. \tag{2}$$

To finish the task before crossing the distance D, we must have:

$$x(k) \leq D, \quad \forall k \leq L \quad \Leftrightarrow \quad x(L) \leq D. \tag{3}$$

Since we allow that the processor finishes the task strictly before the deadline, the whole process can be divided into two stages: 1) the processor is computing while the robot is moving; 2) the computation task is finished but the robot is still traveling for the remaining distance. Let $\alpha(c)$ be the power consumption of the processor at frequency c and $\beta(d)$ be the power consumption of the motor at speed d. Then in the first stage, the total energy consumed by both the processor and the motor is:

$$J_1(L) = \sum_{i=1}^{L} \frac{b}{f(k)} \left[\alpha(f(k)) + \beta(s(k))\right]. \tag{4}$$

The total distance traveled during the first stage is $x(L)$. According to the constraint (3), $x(L) \leq D$. Thus the remaining distance for the second stage is $D - x(L)$. Upon the completion of the task T, the processor can be turned off and consumes a constant power $\alpha(0)$. Since the speed no longer affects the frequency scheduling in the second stage, the robot can be assumed to travel at a constant speed s_0 consuming a constant power $\beta(s_0)$. Hence, the energy consumed in the second stage is

$$J_2(L) = \frac{1}{s_0}(D - x(L))[\alpha(0) + \beta(s_0)]. \tag{5}$$

For a known and fixed work load L, the JSP problem can be formulated as the following discrete time optimal control problem.

Problem 1 (Deterministic Work Load Problem): Find the controls s_0, $f(k)$, and $s(k)$, for $k = 1, \ldots, L$ so that the total energy $J_1(L) + J_2(L)$ is minimized subject to the system equation (2), the control constraint (1) and the state constraint (3).

In Problem 1, we assume that L is known a priori. However, in practice it is very difficult to know exactly how many cycles are needed before running the task. On the other hand, the statistics of L is usually available beforehand. Thus it is more appropriate to assume that the number of needed bins L is a random variable with some given probability measure $P(\cdot)$. In this case, instead of minimizing the energy for a fixed value of L, we shall minimize the expected total energy over all the possible values of L. Note that only the terminal time L is random; all the control variables are still deterministic. Thus the system dynamics is still governed by the deterministic difference equation (2). To guarantee that the constraint (1) is always satisfied, it is required that $P(x(L) \leq D) = 1$. Hence, Problem 1 can be extended to the following problem.

Problem 2 (Random Work Load Problem): Find the optimal controls s_0, $f(k)$ and $s(k)$ for $k = 1, \ldots, n$ that

$$\begin{aligned} \text{minimize} \quad & E[J_1(L) + J_2(L)], \\ \text{subject to} \quad & (1),\ (2) \text{ and } P(x(L) \leq D) = 1, \end{aligned} \tag{6}$$

where $E[\cdot]$ denotes the expectation with respect to the probability measure $P(\cdot)$.

Problem 2 is an optimal control problem with random terminal time L and probabilistic state constraint. Different realizations of the random variable L correspond to different control horizons. For example, if $l < n$ is a realization of L for a particular task, then we only need the controls $\{s(k), f(k)\}$ up to $k = l$ for this task. Since L could be any integer between 1 and n, in Problem 2, we need to find a sequence of control variables over the largest possible control horizon ($k = 1, \ldots, n$) that can handle all the possibilities of L.

Problem 1 is a special case of Problem 2 with $P(L = l) = 1$, In the rest of this paper, we will first derive the optimal solution to Problem 2 and then compare this solution with the solution of Problem 1 to gain more insights about the JSP problem.

III. General Solution Procedure

Before solving Problem 2, we first transform it to a simpler version. Note that the solution to the difference equation (2) is simply

$$x(l)=\sum_{k=1}^{l}\frac{bs(k)}{f(k)}. \tag{7}$$

Since $L\le n$ with probability one, the probabilistic state constraint in (6) can be transformed to a deterministic control constraint as

$$P(x(L)\le D)=1\Leftrightarrow x(n)\le D\Leftrightarrow\sum_{k=1}^{n}\frac{bs(k)}{f(k)}\le D. \tag{8}$$

Since s_0 only affects $J_2(L)$ and does not appear in any constraint in (6), the optimal value of s_0 can be obtained as

$$s_0^*=\arg\min_{s_0\ge0}J_2(L)=\arg\min_{s_0\ge0}\frac{\alpha(0)+\beta(s_0)}{s_0}. \tag{9}$$

Let $p_l=P(L=l)$ and $P_l=P(L\ge l)=\sum_{i=l}^{n}p_i$. Introduce an auxiliary control variable

$$g(k)=s(k)/f(k),\quad\text{for}\quad k=1,\dots,n. \tag{10}$$

Since $s(k)\ge0$ and $f(k)>0$, the new variable $g(k)$ is well defined and is nonnegative. With these definitions, the expected energy with $s_0=s_0^*$ reduces to

$$\begin{aligned}&E[J_1(L)+J_2(L)]=\sum_{l=1}^{n}p_l\cdot(J_1(l)+J_2(l))\\&=\sum_{l=1}^{n}p_l\left[\sum_{k=1}^{l}\frac{b}{f(k)}(\alpha(f(k))+\beta(g(k)f(k)))\right.\\&\qquad\left.+\frac{\alpha(0)+\beta(s_0^*)}{s_0^*}\cdot(D-x(l))\right]\\&=\sum_{k=1}^{n}\left[\frac{P_kb}{f(k)}(\alpha(f(k))+\beta(g(k)f(k)))\right.\\&\qquad\left.-c^*P_kg(k)b\right]+c^*D\\&\triangleq J_P(f,g),\end{aligned} \tag{11}$$

where $c^*\triangleq\frac{\alpha(0)+\beta(s_0^*)}{s_0^*}$. Considering (7), (10) and (11), Problem 2 can be simplified as

$$\begin{aligned}&\min_{f,g}\quad J_P(f,g)\\&\text{subject to}\quad\sum_{i=1}^{n}g(k)b\le D,\\&f(k)>0\quad\text{and}\quad g(k)\ge0.\end{aligned} \tag{12}$$

Now Problem 2 has been transformed into a nonlinear optimization problem without any dynamic state equations constraints. To solve Problem (12), define the Lagrangian as

$$\begin{aligned}L(f,g,\lambda,v)&=J_P(f,g)+c^*\cdot D\\&+\sum_{k=1}^{n}\lambda(k)\cdot(-g(k))+v(\sum_{k}^{n}g(k)b-D),\end{aligned}$$

where $\lambda(k)\ge0$, for $k=1,\dots,n$ and $v\ge0$ are Lagrangian multipliers. Note that we do not need to assign a Lagrangian multiplier for any $f(k)$ as it must be strictly greater than zero. By a standard result of optimization theory [9], the optimal solution (f^*,g^*) to Problem (12) must satisfy the following KKT conditions with certain Lagrangian multipliers $\lambda^*(k)$, $k=1,\dots,n$ and v^*,

$$\begin{aligned}&\frac{\partial}{\partial f^*(k)}\left\{\frac{P_kb}{f^*(k)}\big[\alpha(f^*(k))+\beta(g^*(k)f^*(k))\big]\right\}=0\\&\frac{\partial}{\partial g^*(k)}\left[\frac{P_kb}{f^*(k)}\beta(g^*(k)f^*(k))\right]=c^*P_kb+\lambda^*(k)-v^*b\\&\lambda^*(k)(-g^*(k))=0,\quad v^*(\sum_{i=1}^{n}g^*(k)b-D)=0,\\&f^*(k)>0,\ s^*(k)\ge0,\ \lambda^*(k)\ge0,\ v^*\ge0.\end{aligned} \tag{13}$$

When the objective function and the constraint set are convex, the KKT conditions in (13) characterize the global optimal solution to Problem (12). However, when convexity is not available, some further examinations should be carried out to exclude the possibilities of a maximum, local minimum and saddle point. These examinations are usually simple when a particular problem is being considered.

IV. Analytical Solutions for a Class of Power Functions

In this section we study in depth an important class of Problem (12), where the power functions adopt the following general structures:

$$\alpha(f)=\alpha_0f^3+\alpha_1,\qquad\beta(s)=\beta_0s^2+\beta_1s+\beta_2. \tag{14}$$

In other words, we use a particular class of third-order polynomial and a general quadratic function to represent the power functions of the processor and the motor, respectively. In practice, many variable-speed processors and motors can be effectively modeled using the above functions [4], [10]. To exclude the degenerate case, we assume that $\alpha_0\ne0$ and $\beta_0\ne0$. Furthermore, in order for $\alpha_0(f)$ and $\beta_0(s)$ to be valid power functions, they must be positive for any positive f and s, which implies that

$$\alpha_0>0,\quad\alpha_1>0,\quad\beta_0>0,\quad\text{and}\quad\beta_2>0. \tag{15}$$

In the following discussions, $\alpha(f)$ and $\beta(s)$ are always assumed to take the forms specified in (14) with parameters satisfying (15).

Now we derive analytical solutions of Problem 2 under the conditions (14) and (15). First by (9), the optimal speed for the second stage is

$$s_0^*=\arg\min_{s_0\ge0}\frac{\beta_0s_0^2+\beta_1s_0+\beta_2+\alpha_1}{s_0}=\sqrt{\frac{\alpha_1+\beta_2}{\beta_0}}. \tag{16}$$

The constant c^* in (11) is thus given by

$$c^*=\frac{\alpha(0)+\beta(s_0^*)}{s_0^*}=2\sqrt{\beta_0(\beta_2+\alpha_1)}+\beta_1. \tag{17}$$

Plugging (14) into equation (13), the KKT conditions reduce to

$$P_k b\left(2\alpha_0 f^*(k)+\beta_0 g^*(k)^2-\frac{\beta_2+\alpha_1}{f^*(k)^2}\right)=0, \quad (18a)$$

$$P_k b(2\beta_0 g^*(k)f^*(k)+\beta_1-c^*)+v^*b-\lambda^*(k)=0, \quad (18b)$$

$$\lambda^*(k)(-g^*(k))=0, \quad v^*(\sum_{k=1}^{n} g^*(k)b-D)=0, \quad (18c)$$

$$f^*(k)>0,\ g^*(k)\geq 0,\ \lambda^*(k)\geq 0,\ v^*\geq 0. \quad (18d)$$

By condition (18b), the optimal speed $s^*(k)$ can be expressed in terms of the Lagrangian multipliers as:

$$s^*(k)\triangleq f^*(k)g^*(k)=\frac{c^*-\beta_1}{2\beta_0}-\frac{v^*}{2\beta_0 P_k}+\frac{\lambda^*(k)}{2\beta_0 P_k b}.$$

According to (16) and (17), $\frac{c^*-\beta_1}{2\beta_0}$ is exactly s_0^*. Thus

$$s^*(k)=s_0^*-\frac{v^*}{2\beta_0 P_k}+\frac{\lambda^*(k)}{2\beta_0 P_k b}\triangleq Q_1(v^*,\lambda^*(k),P_k). \quad (19)$$

By (18a) and (19), the optimal frequency $f^*(k)$ can be expressed in terms of the Lagrangian multipliers as:

$$f^*(k)=\left(\frac{\beta_2+\alpha_1-\beta_0(s^*(k))^2}{2\alpha_0}\right)^{1/3} \quad (20)$$

$$\triangleq Q_2(s^*(k))=Q_2(Q_1(v^*,\lambda^*(k),P_k)). \quad (21)$$

To find the optimal solutions, we now characterize the Lagrangian multipliers through a series of lemmas.

Lemma 1: For optimal solutions, we must have $v^*>0$ and $\sum_{k=1}^{n} g^*(k)b-D=0$.

Proof: Assume that $v^*=0$. For an arbitrary k, if $g^*(k)=0$, then by (18b) and (17), $\lambda^*(k)=P_k b(\beta_1-c^*)=-2P_k b\sqrt{\beta_0(\beta_2+\alpha_1)}<0$, contradicting condition (18d). Thus we must have $g^*(k)>0$. In this case, by (18c), $\lambda^*(k)=0$. The optimal frequency as defined in (20) becomes $f^*(k)=Q_2(Q_1(0,0,P_k))=Q_2(s_0^*)$. Considering (16) and (20), we have $f^*(k)=0$, which again contradicts condition (18d). Thus $v^*\neq 0$ and the desired result follows from (18c) and (18d). ■

We now eliminate $\lambda^*(k)$ from equation (19) and (20) and express the optimal solutions only in terms of v^*. Note that when $g^*(k)=0$, the exact value of $\lambda^*(k)$ is not needed because in this case the optimal solution can be directly obtained as $s^*(k)=0$ and $f^*(k)=Q_2(0)$. If $g^*(k)>0$, then by (18c), we automatically have $\lambda^*(k)=0$. Therefore, to eliminate $\lambda^*(k)$, we only need to determine the sign of each $g^*(k)$.

Lemma 2: $g^*(k)>0$ if $s_0^*-\frac{v^*}{2\beta_0 P_k}>0$, and $g^*(k)=0$ if $s_0^*-\frac{v^*}{2\beta_0 P_k}\leq 0$.

Proof: If $g^*(k)=0$, then by (18b), $\lambda^*(k)=P_k b(\beta_1-c^*)+v^*b$. Considering (17) and (16), we have $\lambda^*(k)=b(v^*-2\beta_0 P_k s_0^*)$. It thus follows from the nonnegativeness of $\lambda^*(k)$ that $s_0^*-\frac{v^*}{2\beta_0 P_k}\leq 0$ when $g^*(k)=0$. If $g^*(k)>0$, then $\lambda^*(k)=0$. Since $f^*(k)$ is always positive, $s^*(k)=f^*(k)g^*(k)>0$. By (19), we have $s_0^*-\frac{v^*}{2\beta_0 P_k}>0$ when $g^*(k)>0$. The desired result follows easily by contraposition. ■

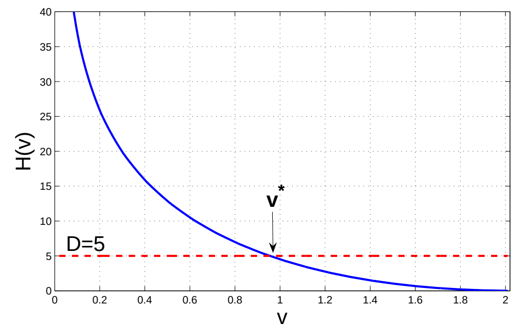


Fig. 1. An example of $H(v)$ with uniform distribution for L. The parameters are $\beta_0=\beta_1=\alpha_0=b=1$, $D=5$, and $\alpha_1=\beta_2=0.5$.

By Lemma 2, if $g^*(k)>0$, then $s_0^*>\frac{v^*}{2\beta_0 P_k}$. Since $P_{k-1}\geq P_k$ for any k, we also have $s_0^*>\frac{v^*}{2\beta_0 P_{k-1}}$, indicating that $g^*(k-1)>0$ as well. Therefore, if $g^*(n_s)>0$ for some $n_s>1$, then $g^*(k)>0$ for all $k\in[1,n_s]$. In other words, there exists an integer N such that the robot travels at a positive speed during the first N bins and stops all the time for the rest of bins. By (18c), we also have $\lambda^*(k)=0$ for $k\leq N$. This integer N depends on the value of v^* and cannot be zero since by Lemma 1, $g^*(k)$ cannot be zero for all the bins. It thus can be defined as

$$N(v^*)=\max\{1\leq k\leq n: Q_1(v^*,0,P_k)=s_0^*-\frac{v^*}{2\beta_0 P_k}>0\}. \quad (22)$$

With this definition, the total distance traveled during the n bins is

$$H(v^*)=\sum_{k=1}^{n}\frac{s^*(k)b}{f^*(k)}=\sum_{k=1}^{N(v^*)}\frac{s^*(k)b}{f^*(k)}=\sum_{k=1}^{N(v^*)}\frac{bQ_1(v^*,0,P_k)}{Q_2(Q_1(v^*,0,P_k))}, \quad (23)$$

where $Q_1(\cdot)$ and $Q_2(\cdot)$ are functions defined in (19) and (20), respectively. By Lemma 1, we must have $H(v^*)=D$. This sets up an equation for determining v^*.

Lemma 3: For any $D>0$, the Lagrangian multiplier v^* can be uniquely determined as $v^*=H^{-1}(D)$.

Proof: As discussed earlier, $N(v^*)$ is well defined. Thus $s^*(1)>0$. Considering that $P_1=1$, we have $v^*<2\beta_0 s_0^*$. This together with Lemma 1 indicates that $v^*\in(0,2\beta_0 s_0^*)$. As $v^*\to 0$, $s^*(1)\to s_0^*$ and by (16) and (20), $f^*(1)\to 0$; thus $H(0^+)=\infty$. On the other hand, as $v^*\to 2s_0^*\beta_0$, $N(v^*)\to 1$ and $s^*(1)\to 0$; thus $H((2s_0^*\beta_0)^-)=0$. It can be easily verified that for any parameters satisfying (15), $H(v^*)$ decreases monotonically from ∞ at $v^*=0^+$ to 0 at $v^*=(2s_0^*\beta_0)^-$. Thus $H(\cdot)$ is invertible and for any positive D, v^* can be uniquely determined by $v^*=H^{-1}(D)$. ■

In Fig. 1, we plot an instance of $H(\cdot)$ and the corresponding v^*. Note that v^* usually cannot be obtained analytically, but can be easily computed numerically. For a given distance D, v^* can be uniquely determined, so are $N(v^*)$, $s^*(k)$ and

$f^*(k)$. We thus can conclude that the following solution is the unique solution satisfying the KKT conditions (18):

$$s^*(k) = \begin{cases} s_0^* - \frac{v^*}{2\beta_0 P_k} & \text{if } i \le N(v^*) \\ 0 & \text{if } i > N(v^*) \end{cases}, \text{ and}$$
$$f^*(k) = \begin{cases} \left(\frac{\beta_2+\alpha_1-\beta_0(s^*(k))^2}{2\alpha_0}\right)^{1/3} & \text{if } i \le N(v^*) \\ (\frac{\beta_2+\alpha_1}{2\alpha_0})^{1/3} & \text{if } i > N(v^*) \end{cases}, \quad (24)$$

where $v^* = H^{-1}(D)$ and s_0^* is defined in (16).

Proposition 1: The solution defined in (24) is the unique global minimum to Problem 2.

Remark 1: As mentioned at the end of the last section, the KKT conditions (18) are only necessary conditions. It is still possible for solution (24) to be a maximum or local minimum or saddle point. However, these possibilities can be easily excluded by the following two obvious facts: (i) solution (24) is the unique one satisfying the KKT conditions; (ii) the cost function is infinite at any boundary point of the constraint set of Problem (12). Therefore, the solution (24) is indeed the global minimum to be sought.

V. Simulation

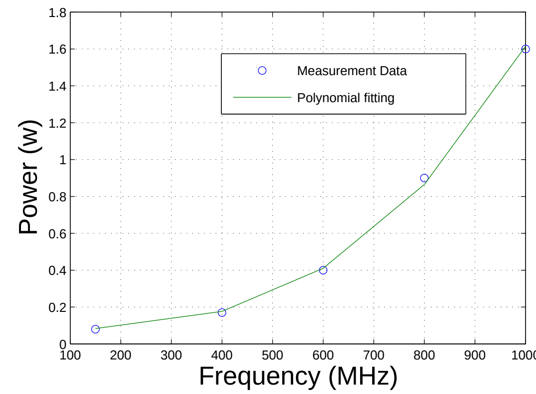
To demonstrate and verify our theoretical analysis, several simulations based on some real-world data are performed in this section. The data employed to identify the power functions $\alpha(f)$ and $\beta(s)$ are from [10] and [4], respectively. The identifications are accomplished using the standard least square method. The performances of the identified functions are shown in Fig. 2 and the resulting parameters are summarized in Table I. We assume that the worst-case execution takes 20 bins, i.e., $n = 20$, where each bin contains 0.1 billion cycles, i.e., $b = 0.1$ billion. The maximum distance D that the robot can travel before it finishes the given task is 100 meters. With these settings, the

TABLE I

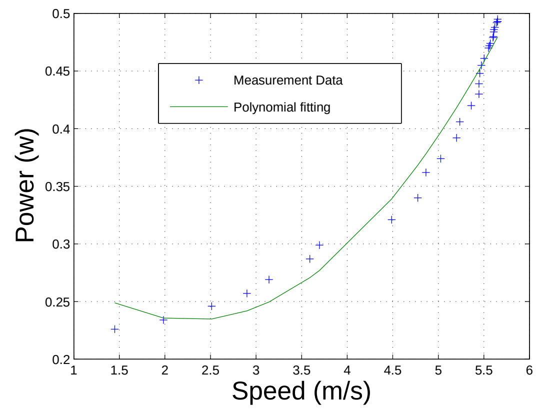
Identified Parameters

α_0	α_1	β_0	β_1	β_2
1.5373×10^{-9}	0.0780	0.0217	-0.0993	0.3471

JSP problem is studied for six different distributions of L. The first two are deterministic distributions where $L = 10$, and $L = 20$ with probability one, respectively. The last four are uniform, Gaussian, and two different exponential distributions as shown in Fig 3. Optimal solutions for these six distributions are computed and compared in Fig. 4. As we can see from Fig. 4, the optimal solution for any deterministic case is constant. On the other hand, in the random case, the optimal speeds decreases as the bin number increases until it reaches 0. Note that for the exponential2 distribution, the speed reaches 0 earlier than the other three random cases. This is consistent with our intuition since small tasks have larger probability in exponential2 distribution than in the other three nondeterministic distributions. Furthermore, the energy consumptions of four different schemes are compared in Fig. 5 for four different distributions of L as shown in



Identification of $\alpha(f)$



Identification of $\beta(s)$

Fig. 2. Identification results of the power functions

Fig. 3. The first three schemes are optimal solutions for the deterministic cases $P(L = 20) = 1$, $P(L = 10) = 1$ and $P(L = 1) = 1$, respectively. In other words, these schemes do not consider the underlying distributions of L and simply execute every task by assuming the "worst case" ($L = 20$), the "median case" ($L = 10$) and the "best case" ($L = 1$). On the other hand, the fourth scheme is the proposed optimal solution with respective to the corresponding distribution of L. To evaluate these schemes, for each distribution of L, we generate a set of large number of tasks with execution times distributed according to the given distribution. Then we use the four schemes to execute each set of tasks. The average energy consumptions of these schemes for each set of tasks are compared in Fig. 5. We can see that the optimal solution always perform the best and can save up to 30% energy compared with the worst scheme.

VI. Conclusion

The JSP problem is formulated as a class of optimal control problems with random terminal times and probabilistic state constraints. The problem is transformed to a deterministic nonlinear optimization problem. The first-order necessary conditions are derived for arbitrary power functions of the processor and motor. For a class of practically important power functions, the optimal solutions are derived analytically. Simulation shows that the optimal solution can save

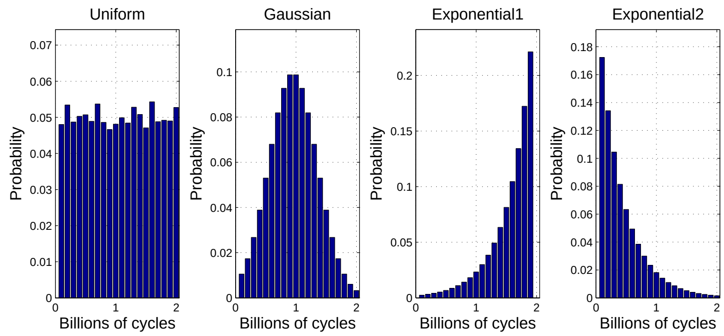
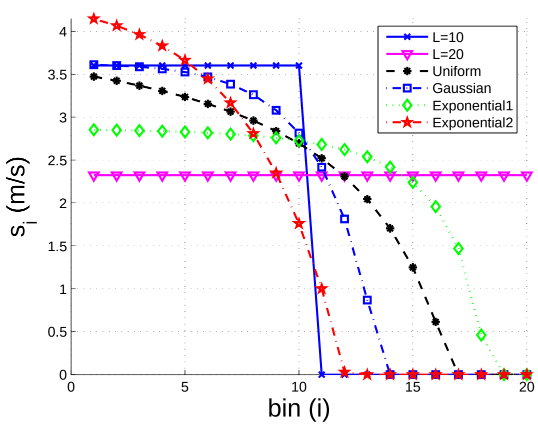
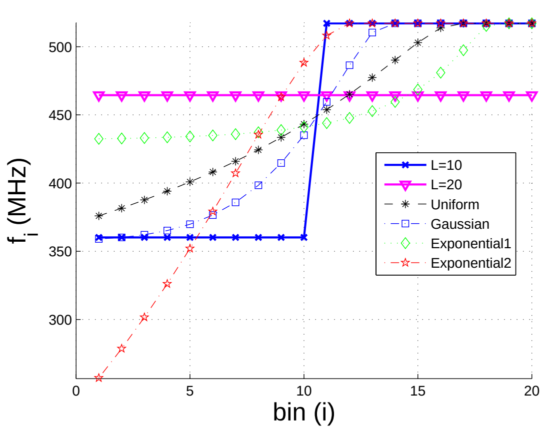


Fig. 3. Distributions of L



Optimal speed sequences



Optimal frequency sequences

Fig. 4. Comparison of optimal solutions in different cases

up to 30% energy compared with some heuristic methods. Future research will focus on the case where the frequency can only take discrete values.

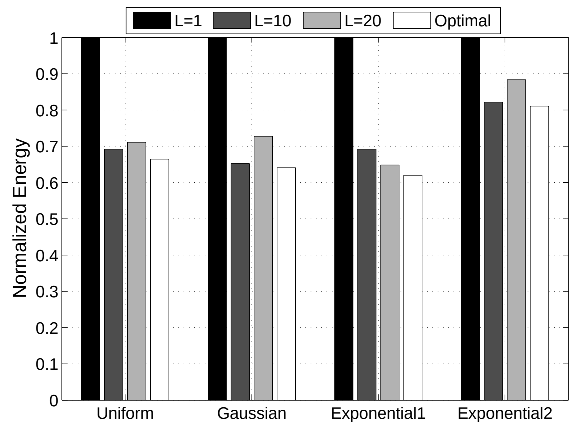


Fig. 5. Normalized average energy consumptions for uniform, Gaussian, and the two exponential type distributions. The first three methods are optimal solutions for the deterministic cases where $L = 1$, $L = 10$ and $L = 20$, respectively. The last method is the optimal solution of the corresponding distribution.

REFERENCES

[1] A. Drenner, I. Burt, T. Dahlin, B. Kratochvil, C. McMillen, B. Nelson, N. Papanikolopoulos, P. Rybski, K. Stubbs, D. Waletzko, and K. Yesin. Mobility enhancements to the scout robot platform. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002.

[2] A. Davids. Urban search and rescue robots: From tragedy to technology. *IEEE Intelligent Systems*, 17(2):81–83, 2002.

[3] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida. Cooperative sweeping by multiple mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996.

[4] Y. Mei, Y.-H. Lu, Y.C. Hu, and C.S.G. Lee. A case study of mobile robot's energy consumption and conservation techniques. In *International Conference on Advanced Robotics*, pages 492– 497, 2005.

[5] J. Brateman C. Xian and Y.-H. Lu. Energy-efficient scheduling for autonomous mobile robots. In *IFIP International Conference on Very Large Scale Integration (VLSI-SoC 2006)*, 2006.

[6] G. Sibley, M. Rahimi, and G. Sukhatme. Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington D.C, April 2002.

[7] A. Colot, G. Caprari, and R. Siegwart. Insbot: design of an autonomous mini mobile robot able to interact with cockroaches. In *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004.

[8] J. Brateman. Frequency and speed setting for energy conservation in autonomous mobile robots. Master's thesis, Purdue University, School of Electrical and Computer Engineering, 2006.

[9] D. G. Luenberger. *Linear and nonlinear programming (2nd Ed)*. Addison-Wesley, Reading,MA, 1984.

[10] R. Xu, C. Xi, R. Melhem, and D. Moss. Practical pace for embedded systems. In *EMSOFT '04: Proceedings of the 4th ACM international conference on Embedded software*, pages 54–63, New York, NY,, 2004. ACM Press.