2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Optimality of Static Control Policies in Some Discrete Event Systems

Lei Miao and **Christos G. Cassandras**
Dept. of Manufacturing Engineering and Center for Information and Systems Engineering
Boston University
Brookline, MA 02446
leimaio@bu.edu,cgc@bu.edu

Abstract— **We consider a class of Discrete Event Systems (DES) that involves the control of resources allocated to tasks under real-time constraints. This is motivated by power-limited wireless environments such as sensor networks, where the objective is to minimize energy consumption while guaranteeing that task deadlines are always met. In obtaining optimal off-line controllers for such systems, we prove that simple static control gives the unique optimal solution. The result is of interest because it asserts the optimality of a simple controller that does not require any data collection or processing in environments where the cost of such actions is high.**

Index Terms— **Discrete event system, hybrid system, power-limited system, optimization**

I. INTRODUCTION

A large class of Discrete Event Systems (DES) involves the control of resources allocated to tasks according to certain operating specifications (e.g., tasks may have real-time constraints associated with them). The basic modeling block for such DES is a single-server queueing system operating on a first-come-first-served basis, whose dynamics are given by the well-known max-plus equation

$$x_i = \max(x_{i-1}, a_i) + s_i \tag{1}$$

where a_i is the arrival time of task $i = 1, 2, \ldots,$ x_i is the time when task i completes service, and s_i is its (generally random) service time. Traditionally, once a task begins service, its processing rate is kept fixed, i.e., s_i is independent of the system state. However, as performance requirements increase and DES are expected to operate in heavily constrained environments, an interesting question that arises is the following: *what is the benefit of varying the processing rate depending on the information available to a controller that can regulate this rate?* Examples arise in manufacturing systems, where the operating speed of a machine can be controlled to trade off between energy costs and requirements on timely job completion [1]; in computer systems, where the CPU speed can be controlled to ensure that certain tasks meet specified execution deadlines [2]; and in sensor networks where severe battery limitations call for new techniques aimed at maximizing the lifetime of such a network [3]. In such a setting, the physical process taking place when task i is served is characterized by its own dynamics

$$\dot{z}_i = g_i(z_i, u_i, t), \quad z_i(x_{i-1}) = z_i^0, \;\; t \in [x_{i-1}, x_i) \tag{2}$$

where $z_i(t)$ is the *physical state* of task i over $[x_{i-1}, x_i)$ and $u_i(t)$ is some control defined over $[x_{i-1}, x_i)$. Therefore, we can rewrite (1) as

$$x_i = \max(x_{i-1}, a_i) + s(z_i, u_i), \quad i = 1, 2, \ldots \tag{3}$$

where x_i is the *temporal state* of task i and $s(z_i, u_i)$ is its processing time which now depends on some control u_i; for notational ease, we write u_i to denote a function $u_i(t)$ defined over $[x_{i-1}, x_i)$, and similarly for z_i. This, in turn, transforms the DES into a hybrid system, where (3) represents the *event-driven* and (2) the *time-driven* component.

Our goal is to study the question raised above in the general context of (3) and (2), given a specific performance objective for the system. In this paper, we restrict ourselves to a particular family of problems motivated by power-limited wireless systems such as sensor networks, where the objective is to minimize energy consumption while satisfying some operating constraints. The processing of tasks at a typical node of such a system can be modeled by (3). The control, in this case, is the voltage of the node processor and the idea of *Dynamic Voltage Scaling* (DVS) is to adjust this voltage depending on the state of the system [4],[5],[6],[7],[8],[9]. Since energy is generally related to voltage through a relationship of the form $E = CV^2$ for some constant C, scaling down the processing voltage will decrease the processing rate but it will also quadratically decrease the energy per operation. The physical state of task i is the number of operations left, given that the task starts out with a given number of operations to execute. The objective is to minimize the total energy consumed over some given number N of tasks, subject to the event and time-driven dynamics and possibly additional constraints on the state and control processes.

The design of the controller depends on the mode of operation of the system. In an *off-line* scheme, the sequence

The authors' work is supported in part by the National Science Foundation under Grants EEC-00-88073 and DMI-0330171, by AFOSR under grants FA9550-04-1-0133 and FA9550-04-1-0208, and by ARO under grant DAAD19-01-0610.

0-7803-9098-9/05/$25.00 ©2005 AACC

of task arrival times $\{a_i\}$, $i = 1, \ldots, N$, is known in advance. Often, an arrival is always constrained to occur in a known interval $[a_i^-, a_i^+]$, referred to as "release time jitter" [2]; this includes situations where if expected tasks are not received within a particular time interval, then they are considered useless and are never processed (e.g., expected data that arrive too late to a processing node in a sensor network). Thus, a "worst case controller" can be designed using the sequence $\{a_i^+\}$, $i = 1, \ldots, N$. On the other hand, in an *on-line* scheme, at time t the controller has at its disposal all *actual* arrival time data $a_i < t$, which allows it to be more flexible; if, for example, the set $\{i : a_i < t\}$ has a large cardinality, then a high processing rate is called for to prevent a further backlog of tasks.

The controller is *dynamic* when $u_i(t)$ is allowed to vary over all $t \in [x_{i-1}, x_i)$; it is called *static* when $u_i(t)$ is kept fixed over $[x_{i-1}, x_i)$. The main contribution of this paper is to show that a static control is the unique optimal control of a problem minimizing the total energy consumed subject to task deadline constraints $x_i \leq d_i$ for given d_i, $i = 1, \ldots, N$. The result is significant since it asserts the optimality of a simple controller that does not require any data collection or processing in environments where the cost of such actions is high. Moreover, a static controller requires no overhead that would otherwise be involved in making continuous control adjustments and, as we will see later, it is helpful in designing on-line dynamic controllers as well. As will become obvious from the analysis, our result is quite general and applies to all optimal control settings described above, as long as the cost function of interest is strictly convex and monotonically increasing (or decreasing, depending on the control variables we use).

In section II, we present our system model and formulate the optimization problem. Section III contains the main results. We conclude in Section IV by discussing the implications of our work and future directions.

II. System model and problem formulation

The system we consider is characterized by the event-driven dynamics (3), where a_i is the arrival time of task $i = 1, 2, \ldots, N$, and x_i is the time when task i completes service. For power-limited wireless devices which must maintain operational simplicity, we assume a first-come-first-served and nonpreemptive queueing model. Let us assume that arrival times are given, so that we consider an off-line control scheme in which all controls are evaluated in advance.

Considering first a static controller, let u_i be a control variable representing the processing time allocated to task $i = 1, \ldots, N$ and is kept fixed throughout $[x_{i-1}, x_i)$. We assume that $u_{i\min} \leq u_i \leq u_{i\max}$, $i = 1, ..., N$, where $u_{i\min}, u_{i\max}$ are given. We also assume that each task i is constrained to be completed by a given deadline d_i and consider the optimization problem:

$$\min_{u_1,\ldots,u_N} \sum_{i=1}^{N} \theta_i(u_i) \tag{4}$$

$$\text{s.t.} \quad u_{i\min} \leq u_i \leq u_{i\max}, \quad i = 1, \ldots, N$$

$$x_i = \max(x_{i-1}, a_i) + u_i \leq d_i, \quad i = 1, \ldots, N, \; x_0 = 0$$

where the cost function $\theta_i(u_i)$ represents the energy consumed in processing task i under control u_i. We assume that $\theta_i(u_i)$ is strictly convex, differentiable, and monotonically decreasing in u_i. An explicit form for $\theta_i(u_i)$ can be obtained for specific processor types. As an example, for CMOS processors [10], the energy consumption per operation E is related to the operating voltage V through

$$E = C_1 V^2 \tag{5}$$

and the processing frequency (clock speed) is given by

$$f = \frac{V - V_t}{C_2 V} \tag{6}$$

where C_1, C_2 are constants dependent on the physical characteristics of a device and V_t is the threshold voltage, so that $V \geq V_t$. If task i has μ_i operations and is processed with a constant rate $\frac{\mu_i}{u_i}$, then from (5) and (6), we can get:

$$\theta_i(u_i) = \mu_i E = \mu_i C_1 \left(\frac{V_t u_i}{u_i - \mu_i C_2} \right)^2 \tag{7}$$

Moreover, assuming the voltage is constrained so that $V_{\min} \leq V \leq V_{\max}$, the constraint on u_i becomes

$$u_{i\min} = \frac{\mu_i C_2 V_{\max}}{V_{\max} - V_t} \leq u_i \leq u_{i\max} = \frac{\mu_i C_2 V_{\min}}{V_{\min} - V_t} \tag{8}$$

We omit the amount of time it takes for the processor to reach steady state during voltage and frequency changing, since the transition time is very small compared to task processing times (e.g., the lpARM processor in [11] is designed to operate between 1.1V and 3.3V, resulting in speeds between 10MHz and 100MHz, and clock frequency transitions take approximately $25\mu s$ for a complete 10MHz to 100MHz transition).

Let us now define $\tau_i = \frac{1}{f_i} = \frac{u_i}{\mu_i}$ where f_i is the processing rate of task i, and τ_i is the processing time per operation. Then, from (7), we get

$$\theta_i(u_i) = \mu_i C_1 \left(\frac{V_t \tau_i}{\tau_i - C_2} \right)^2$$

Thus, for cost functions of this form we can write

$$\theta_i(u_i) = \mu_i \theta(\tau_i) \tag{9}$$

where $\theta(\tau_i)$ does not depend on the specific task and represents the energy consumption per operation as a function of τ_i. Treating $\tau_1, ..., \tau_N$ as the control variables in the static controller setting and assuming a cost function that satisfies

(9), we can rewrite (4) as follows and refer to it as problem **P1**:

$$\min_{\tau_1,\ldots,\tau_N} \sum_{i=1}^{N} \mu_i \theta(\tau_i)$$
$$\text{s.t.} \quad \tau_{\min} \le \tau_i \le \tau_{\max}, \quad i = 1,\ldots,N$$
$$x_i = \max(x_{i-1}, a_i) + \tau_i \mu_i \le d_i, \quad i = 1,\ldots,N, \ x_0 = 0$$

where $\tau_{\min} = \frac{u_{i\min}}{\mu_i}, \tau_{\max} = \frac{u_{i\max}}{\mu_i}$. From (8), $\tau_{\min}$ $\tau_{\max}$ are constants, and they are independent of i. In what follows, we will remove the constraint $\tau_i \le \tau_{\max}$, $i = 1,\ldots,N$ from our problem formulation. We will soon show that this does not affect the optimal solution. The relaxed problem, which we shall refer to as $Q(1,N)$, becomes

$$\min_{\tau_1,\ldots,\tau_N} \sum_{i=1}^{N} \mu_i \theta(\tau_i)$$
$$\text{s.t.} \quad \tau_i \ge \tau_{\min}, \quad i = 1,\ldots,N$$
$$x_i = \max(x_{i-1}, a_i) + \tau_i \mu_i \le d_i, \quad i = 1,\ldots,N, \ x_0 = 0.$$

This problem formulation was used in [12] in addressing the DVS problem. $Q(1,N)$ is a special case of the more general class of problems of this type studied in [13], where a decomposition algorithm termed the Forward Algorithm (FA) was derived. As shown in [13], instead of solving this complex nonlinear optimization problem, we can decompose the optimal sample path to a number of busy periods. A *busy period* (BP) is a contiguous set of tasks $\{k,...,n\}$ such that the following three conditions are satisfied: $x_{k-1} < a_k$, $x_n < a_{n+1}$, and $x_i \ge a_{i+1}$, for every $i = k,\ldots,n-1$. The FA decomposes the entire sample path into BPs and replaces the original problem by a sequence of simpler convex optimization problems, one for each BP; as shown in [13], the solution is identical to that of the original problem. In [12], it is shown that the additional structure of $Q(1,N)$ leads to an efficient algorithm that decomposes the sample path even further and does not require solving any convex optimization problem. In what follows, we shall make use of some results in [12]. We will use $\{x_i^*\}$, $i = 1,\ldots,N$, to denote an optimal solution of $Q(1,N)$.

Lemma 1: $Q(1,N)$ has a unique solution.

Proof: Invoking Lemma 3 in [12], $x_i^* < a_{i+1}$ iff $d_i < a_{i+1}$.Therefore, the BP structure of the optimal sample path is uniquely characterized by the known $a_1,\ldots,a_N$ and $d_1,\ldots,d_N$. In addition, suppose we have a BP starting with task k and ending with task n. The set of all feasible controls $\{\tau_k,\ldots,\tau_n\}$ is a convex set and the cost function in $Q(1,N)$ is strictly convex. Therefore, the solution of the optimization problem pertaining to this BP is unique and it follows that $Q(1,N)$ has a unique solution. ■

Lemma 2: Suppose $\tau_1^*,\ldots,\tau_N^*$ is the unique solution to $Q(1,N)$. Let $\tau_i' = \tau_i^*$ if $\tau_i^* < \tau_{\max}^*$ and $\tau_i' = \tau_{\max}^*$ otherwise, for all $i = 1,\ldots,N$. Then, $\tau_1',...,\tau_N'$ is the unique solution to problem **P1**.

Proof: Suppose there are $M \le N$ tasks whose optimal controls for $Q(1,N)$ are such that $\tau_1^* < \tau_{\max}^*$. Denote these optimal controls by $\tau_{L(1)}^*,\ldots,\tau_{L(M)}^*$, and denote the remaining optimal controls by $\tau_{R(1)}^*,\ldots,\tau_{R(N-M)}^*$. Therefore,

$$\tau_i' = \tau_i^*, \quad i = L(1),\ldots,L(M)$$
$$\tau_i' = \tau_{\max}^*, \ i = R(1),\ldots,R(N-M)$$

We first show that by removing those tasks with $\tau_i^* \ge \tau_{\max}^*$, there is no effect on the optimal control of task $i-1$ or task $i+1$ in $Q(1,N)$. If task i is the first or last task of a BP, we only need to consider task $i+1$ or task $i-1$ respectively. Therefore, we consider the more general case where task i neither starts nor ends a BP. There are five cases to consider:

1) $a_i < x_{i-1}^* < d_{i-1}$, and $a_{i+1} < x_i^* < d_i$. From Proposition 3 in [12], $\tau_{i-1}^* = \tau_i^* = \tau_{i+1}^*$. Therefore, both tasks $i-1$ and $i+1$ are also removed in this case.

2) $x_{i-1}^* = d_{i-1}$. Since task $i-1$ is done at its deadline, removing task i can have no effect on task $i-1$.

3) $x_{i-1}^* = a_i$. From Proposition 3 in [12], $\tau_{i-1}^* \ge \tau_i^*$. Since $\tau_i^* \ge \tau_{\max}^*$, we have $\tau_{i-1}^* \ge \tau_{\max}^*$ and task $i-1$ is also removed.

4) $x_i^* = d_i$. From Proposition 3 in [12], $\tau_i^* \le \tau_{i+1}^*$. Since $\tau_i^* \ge \tau_{\max}^*$, we have $\tau_{i+1}^* \ge \tau_{\max}^*$ and task $i+1$ is also removed.

5) $x_i^* = a_{i+1}$. Since task $i+1$ is processed right after its arrival, removing task i can have no effect on task $i+1$.

Since there is no improvement to the optimal controls of tasks adjacent to i for any $i = R(1),\ldots,R(N-M)$, it follows that removing all tasks $R(1),\ldots,R(N-M)$ can result in no improvement to the remaining tasks $L(1),\ldots,L(M)$ in $Q(1,N)$. This can be easily seen by applying a contradiction argument and using Lemma 1. Suppose there exists another solution to problem $Q(1,M)$ where the M tasks are those labeled $L(1),\ldots,L(M)$ above and this solution is $\{\bar{\tau}_{L(1)},\ldots,\bar{\tau}_{L(M)}\}$. By inserting tasks $R(1),\ldots,R(N-M)$, with corresponding controls $\tau_{R(1)}^*,\ldots,\tau_{R(N-M)}^*$, we obtain another solution to $Q(1,N)$ since we have shown that the inclusion of these tasks has no effect on the rest. This solution is given by $\{\bar{\tau}_{L(1)},\ldots,\bar{\tau}_{L(M)}\}$ for $L(1),\ldots,L(M)$ and $\{\tau_{R(1)}^*,\ldots,\tau_{R(N-M)}^*\}$ for the rest. This contradicts Lemma 1 where we established that $Q(1,N)$ has a unique solution.

Therefore, $\{\tau_{L(1)}^*,\ldots,\tau_{L(M)}^*\}$ is the unique solution to problem **P1** when this is solved for tasks $L(1),\ldots,L(M)$ instead of tasks $1,\ldots,N$. Now, let us add tasks $R(1),\ldots,R(N-M)$ into problem **P1** with control $\tau_{\max}^*$ for each one of them. This solution is just $\tau_1',\ldots,\tau_N'$ and it is the unique solution to problem **P1**. ■

Lemma 2 above provides the justification for removing the constraint $\tau_i \le \tau_{\max}$, $i = 1,\ldots,N$ in our problem formulation, without affecting optimality.

As already mentioned, we have formulated $Q(1,N)$ as a static control optimization problem (in the DVS setting, this is also referred to as "inter-task" control, i.e., the execution of control actions is at task departures only). A dynamic controller is one where voltage can be adjusted at any time instant. However, since the state of our queueing model can only change as a result of two event types (task arrivals and task departures), the only other possible times when a voltage change can be considered in any $[x_{i-1}, x_i)$ are arrival times a_k such that $x_{i-1} < a_k < x_i$. This is referred to as "intra-task" control, as illustrated in Fig. 1. For example, $a_2 < x_1$, therefore, the processing rate for task 1 is initially $1/\tau_{11}$ and then becomes $1/\tau_{12}$ at time a_2.

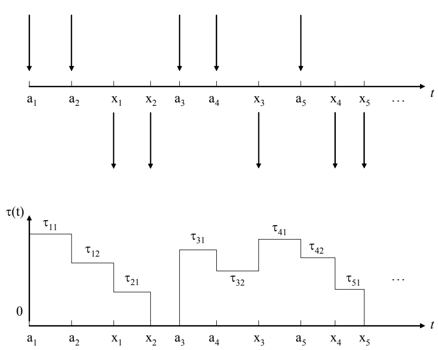

Fig. 1. Example of intra-task control.

We can formulate an optimization problem in which the controller is updated at both arrival and departure times as follows. Let c_i be the number of times the rate changes during the processing time of task i, i.e., number of arrivals during the processing time of task i. Moreover, let $\eta_{i,j}$ be the length of j-th interval in the processing time of task i over which the rate remains fixed, $j = 1, \ldots, c_i + 1$. The intra-task optimization problem **P2** can be formulated as follows:

$$\min_{\substack{c_i \ for\ all\ i \\ \tau_{i,j},\ \eta_{i,j}\ for\ all\ i,j}} \sum_{i=1}^{N} \sum_{j=1}^{c_i+1} \frac{\eta_{i,j}}{\tau_{i,j}} \theta(\tau_{i,j})$$

$$\text{s.t.} \quad \tau_{i,j} \geq \tau_{i,j\,\min}, \quad i = 1, \ldots, N, \ j = 1, \ldots, c_i + 1$$

$$\eta_{i,j} > 0, \quad 0 \leq c_i \leq N, \quad i = 1, \ldots, N, \ c_0 = 0$$

$$x_i = \max(x_{i-1}, a_i) + \sum_{j=1}^{c_i+1} \eta_{i,j} \leq d_i, \quad i = 1, \ldots, N,$$

$$x_0 = 0$$

$$\eta_{i,j} = a_{l+j} - \max(x_{i-1}, a_{l+j-1}), \quad \text{for } c_i > 0,$$

$$l = i + \sum_{k=0}^{i-1} c_k, \quad j = 1, \ldots, c_i, \quad i = 1, \ldots, N$$

$$\sum_{i=1}^{N} c_i = N, \qquad \sum_{j=1}^{c_i+1} \frac{\eta_{i,j}}{\tau_{i,j}} = \mu_i, \quad i = 1, \ldots, N$$

Note that c_i, $\tau_{i,j}$ and $\eta_{i,j}$ are all treated as control variables. However, not all $\eta_{i,j}$ are controllable: when $c_i > 1$, only $\eta_{i,1}$ and η_{i,c_i+1} are controllable, whereas all other $\eta_{i,j}$, $1 < j < c_i + 1$, are determined by task arrivals. A control $\tau_{i,j}$ is applied at the beginning of task i's processing time and may be updated at each arrival (if any) that occurs during the processing of task i. However, the control between any two adjacent events is still constant.

We shall now formulate a general dynamic control problem, **P3**, in which the controller may change at any time, so that **P2** will be a special case of it. We shall then analyze **P3**. To do so, we will view $f = 1/\tau$ as the controllable processing rate, so that in the cost function we replace $\theta(\tau)$ by $\theta(1/f) \equiv \gamma(f)$. Recalling the hybrid system framework of (3) and (2), we can define the physical state of task i as the number of operations left in the interval $[\max(x_{i-1}, a_i), x_i]$, i.e.,

$$\dot{z}_i = -f(t), \quad t \in [\max(x_{i-1}, a_i), x_i),$$
$$z_i(\max(x_{i-1}, a_i)) = \mu_i, \qquad z_i(x_i) = 0$$

while the temporal state satisfies

$$x_i = \max(x_{i-1}, a_i) + s(f(t)), \quad t \in [\max(x_{i-1}, a_i), x_i)$$

It is more convenient to treat the departure times x_i, $i = 1, \ldots, N$ as control variables as well and combine the two equations above to obtain a set of integral sample path constraints:

$$\int_{\max(x_{i-1}, a_i)}^{x_i} f(t)dt = \mu_i, \ i = 1, \ldots, N$$

The dynamic optimization problem **P3** is as follows:

$$\min_{f(t),\ x_i\ for\ i=1,\ldots N} \int_{a_1}^{d_N} f(t)\gamma(f(t))dt$$

$$\text{s.t.} \quad 1/f(t) \geq \tau_{\min}, \quad \text{for all } t \in [a_1, d_N]$$

$$x_i \leq d_i, \quad i = 1, \ldots, N, \ x_0 = 0$$

$$\int_{\max(x_{i-1}, a_i)}^{x_i} f(t)dt = \mu_i, \ i = 1, \ldots, N$$

Comparing this to **P2**, note that in the objective function of **P2** $\frac{\eta_{i,j}}{\tau_{i,j}}\theta(\tau_{i,j})$ is the energy consumed by the $\frac{\eta_{i,j}}{\tau_{i,j}}$ operations executed over the jth fixed-control interval in the processing time of task i. In the objective function of **P3**, $f(t)dt$ is the number of operations processed in time dt and $f(t)\gamma(f(t))dt$ is the energy consumed by these operations. In both cases, the objective function represents the total energy needed to process N tasks.

In the next section we will analyze **P3** and show that its solution is in fact a controller which is static over each task's processing time. Moreover, this solution is identical to the solution $\{\tau_1^*, \ldots, \tau_N^*\}$ of problem $Q(1, N)$, so that applying intra-task control as in **P2** provides no benefit.

III. Off-line optimality analysis

We begin with an auxiliary lemma, which will be used to establish the key result in this section, Lemma 4, which in turn will allow us to derive Theorem 5.

Lemma 3: If $g(s)$ is a strictly convex, differentiable and increasing function of $s \in \mathbb{R}$, $s > 0$, then $sg(s)$ is a strictly convex function.

Proof: See [14]. ■

Lemma 4: Suppose $\int_a^b \phi(t)dt = C$, where $\phi(t) > 0$ is bounded over $[a, b]$, $a, b \in \mathbb{R}$, $a < b$, C is a constant, and $g(\phi)$ is strictly convex, increasing, and differentiable. Then, $\int_a^b \phi(t)g(\phi(t))dt$ is minimized when $\phi(t) = \frac{C}{b-a}$ and this is the unique minimum.

Proof: We use a contradiction argument and assume that $\phi(t) = \frac{C}{b-a}$ is not the unique minimizer, i.e., either $\phi(t) = \frac{C}{b-a}$ is not a minimizer or it is a minimizer but is not unique. Let $\phi^{'}(t) \neq \frac{C}{b-a}$ for all $t \in [a, b]$ be a feasible minimizing function bounded over $[a, b]$. Because $\phi^{'}(t)$ is feasible,

$$\int_a^b \phi^{'}(t)dt = C$$

and since $\phi^{'}(t) \neq \frac{C}{b-a}$ for all $t \in [a, b]$, there must exist t_1 and t_2 s.t. $a < t_1 < t_2 < b$, $\phi^{'}(t_1) \neq \frac{C}{b-a}$, $\phi^{'}(t_2) \neq \frac{C}{b-a}$, and $\phi^{'}(t_1)+ \phi^{'}(t_2) = 2\frac{C}{b-a}$.

Consider a function $\phi^{''}(t)$ which is defined as follows:

$$\phi^{''}(t) = \phi^{'}(t) + (\frac{C}{b-a} - \phi^{'}(t_1))\mathbf{1}[t = t_1] \\ +(\frac{C}{b-a} - \phi^{'}(t_2))\mathbf{1}[t = t_2]$$

where $\mathbf{1}[\cdot]$ is the usual indicator function. Then,

$$\int_a^b \phi^{''}(t)dt = \int_a^{t_1^-} \phi^{'}(t)dt + \int_{t_1^-}^{t_1^+} \frac{C}{b-a}\mathbf{1}[t = t_1]dt+ \\ \int_{t_1^+}^{t_2^-} \phi^{'}(t)dt + \int_{t_2^-}^{t_2^+} \frac{C}{b-a}\mathbf{1}[t = t_2]dt + \int_{t_2^+}^{b} \phi^{'}(t)dt$$

and since $\phi^{'}(t_1)+ \phi^{'}(t_2) = 2\frac{C}{b-a}$, we get

$$\int_a^b \phi^{''}(t)dt = \int_a^b \phi^{'}(t)dt = C$$

Therefore, $\phi^{''}(t)$ is feasible. Then,

$$\int_a^b \phi^{''}(t)g(\phi^{''}(t))dt = \int_a^{t_1^-} \phi^{'}(t)g(\phi^{'}(t))dt+ \tag{10}$$
$$\int_{t_1^-}^{t_1^+} \frac{C}{b-a}g(\frac{C}{b-a})\mathbf{1}[t = t_1]dt + \int_{t_1^+}^{t_2^-} \phi^{'}(t)g(\phi^{'}(t))dt+ \\ \int_{t_2^-}^{t_2^+} \frac{C}{b-a}g(\frac{C}{b-a})\mathbf{1}[t = t_2]dt + \int_{t_2^+}^{b} \phi^{'}(t)g(\phi^{'}(t))dt$$

Because $g(\phi)$ is convex, increasing and differentiable, and $\phi > 0$, from Lemma 3, $\phi g(\phi)$ is a strictly convex function. By definition, for all $\alpha \in [0, 1]$,

$$\alpha\phi^{'}(t_1)g(\phi^{'}(t_1)) + (1-\alpha)\phi^{'}(t_2)g(\phi^{'}(t_2)) > (\alpha\phi^{'}(t_1)+ \\ (1-\alpha)\phi^{'}(t_2))g(\alpha\phi^{'}(t_1) + (1-\alpha)\phi^{'}(t_2))$$

Choosing $\alpha = 1/2$ gives

$$\phi^{'}(t_1)g(\phi^{'}(t_1)) + \phi^{'}(t_2)g(\phi^{'}(t_2)) > \\ 2((\phi^{'}(t_1) + \phi^{'}(t_2))/2)g((\phi^{'}(t_1) + \phi^{'}(t_2))/2) \\ = 2\frac{C}{b-a}g(\frac{C}{b-a}).$$

and using this inequality in (10) we obtain:

$$\int_a^b \phi^{''}(t)g(\phi^{''}(t))dt < \int_a^{t_1^-} \phi^{'}(t)g(\phi^{'}(t))dt+ \\ \int_{t_1^-}^{t_1^+} \phi^{'}(t_1)g(\phi^{'}(t_1))\mathbf{1}[t = t_1]dt + \int_{t_1^+}^{t_2^-} \phi^{'}(t)g(\phi^{'}(t))dt+ \\ \int_{t_2^-}^{t_2^+} \phi^{'}(t_2)g(\phi^{'}(t_2))\mathbf{1}[t = t_2]dt + \int_{t_2^+}^{b} \phi^{'}(t)g(\phi^{'}(t))dt \\ = \int_a^b \phi^{'}(t)g(\phi^{'}(t))dt$$

This inequality contradicts the assumption that $\phi^{'}(t) \neq \frac{C}{b-a}$ is a minimizer. Since all feasible functions other than the fixed one $\frac{C}{b-a}$ cannot be minimizers, it follows that $\phi(t) = \frac{C}{b-a}$ must be the unique minimizer. ■

Using this result, let us now compare the solutions of $Q(1, N)$ and of problem **P3**.

Theorem 5: If $f_i^*(t)$ is the optimal control function during the processing of task i in **P3**, and τ_i^* is the corresponding optimal control in $Q(1, N)$, then $f_i^*(t) = 1/\tau_i^*$.

Proof: In Lemma 4, let $\phi(t) = f(t)$, $g(f(t)) = \gamma(f(t))$, and $a = max(x_{i-1}, a_i)$, $b = x_i$ where $\{x_i\}$, $i = 1, \ldots, N$, is any feasible solution of problem **P3**. Then, $f_i^*(t)$ is a constant. With $f_i^*(t)$ constant, look at **P3** and observe that when the cost function is minimized, $f(t) = 0$ in any idle period. Thus, the cost function can be rewritten as a summation over N tasks:

$$\sum_{i=1}^{N} \int_{\max(x_{i-1},a_i)}^{x_i} f_i(t)\gamma(f_i(t))dt = \\ \sum_{i=1}^{N}[x_i - \max(x_{i-1}, a_i)]f_i(t)\gamma(f_i(t))$$

With $f_i(t) = 1/\tau_i$ a constant over $[\max(x_{i-1}, a_i), x_i]$, the integral constraints in **P3** reduce to

$$(x_i - \max(x_{i-1}, a_i)) = \tau_i\mu_i, \quad i = 1, \ldots, N \tag{11}$$

and the cost function above becomes

$$\sum_{i=1}^{N} \mu_i\gamma(1/\tau_i) \equiv \sum_{i=1}^{N} \mu_i\theta(\tau_i)$$

Moreover, from (11) we can see that all x_i are explicitly determined from a_i and τ_i and the initial condition $x_0 = 0$. Therefore, x_i is no longer a control variable. Combining all these observations, problem **P3** can be rewritten as follows:

$$\min_{\tau_1,\ldots,\tau_N} \sum_{i=1}^{N} \mu_i \theta(\tau_i)$$
$$\text{s.t.} \quad \tau_i \geq \tau_{\min}, \quad i = 1, \ldots, N$$
$$x_i = \max(x_{i-1}, a_i) + \tau_i \mu_i \leq d_i, \quad i = 1, \ldots, N, \; x_0 = 0.$$

which is precisely problem $Q(1, N)$. Therefore, with $f_i(t)$ constant, problems $Q(1, N)$ and **P3** are identical and it follows that $f_i^*(t) = 1/\tau_i^*$. From Lemma 1, **P3** also has a unique solution. ■

Theorem 5 asserts that the optimal dynamic control for the off-line problem we have formulated is to keep the processor rate constant while processing the same task, i.e., a static control is globally optimal. In other words, the solution of $Q(1, N)$ (which can be obtained by the efficient algorithm presented in [12]) gives a lower bound for the off-line dynamic optimization problem.

IV. Conclusions and Future Work

We have considered the off-line optimal control problem for a class of DES encountered in power-limited wireless systems. In particular, we are interested in minimizing the energy consumption in such systems subject to some control constraints and the requirement that the execution of all tasks meets prespecified time deadlines. We have established the fact that a static control is the unique solution of this dynamic optimization problem, which ensures that such systems can be optimally controlled without the need to collect or process data. In addition, the fact that the control is fixed over each task further preserves the overhead that would otherwise be required to make processing rate (i.e., processor clock speed) adjustments at arrival event times in the formulation of **P2**.

Our next goal is to design on-line controllers for these systems. In this case, we no longer assume that task arrival information is known; instead, real-time event information obtained over the evolution of a sample path is used and one can no longer expect that a static controller would be optimal. Moreover, the absence of future event time information requires us to treat this as a stochastic control problem. To bypass the complexity that would result, one can resort to designing a Receding Horizon (RH) controller, based on the assumption that some future information over a limited time window is available or can be estimated with good accuracy; such controllers have been developed and analyzed in [15]. In general, we must seek on-line controllers which (i) guarantee the required task deadlines, (ii) attempt to minimize energy consumption, and (iii) if they are not optimal, it is possible to quantify their deviation from optimal performance. Our results in this paper are helpful in designing on-line RH controllers, since at each decision point where the receding horizon is updated we are actually solving an off-line problem that uses the limited future task information available within the RH "lookahead" window. Although optimal control will no longer be static from one decision point to another, the optimal control evaluated at a specific decision point will remain static. Therefore, based on our results, at any such decision point an on-line controller does not require any data collection or processing either, after it acquires the updated task information.

References

[1] D. L. Pepyne and C. G. Cassandras, "Optimal control of hybrid systems in manufacturing," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1108–1123, 2000.

[2] J. Liu, *Real - Time System*. Prentice Hall Inc., 2000.

[3] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. Srivastava, "Power optimization of variable-voltage core-based systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 12, pp. 1702–1714, 1999.

[4] A. Qadi, S. Goddard, and S. Farritor, "A dynamic voltage scaling algorithm for sporadic tasks," in *Real-Time Systems Symposium.*, pp. 52–62, 2003.

[5] G. Qu, "What is the limit of energy saving by dynamic voltage scaling?," in *Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on*, p. 560, Nov. 2001.

[6] T. Pering, T. Burd, and R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," in *Low Power Electronics and Design, 1998.Proceedings. 1998 International Symposium on*, pp. 76–81, Aug. 1998.

[7] W. Kim, J. Kim, and S. L. Min, "Dynamic voltage scaling algorithm for dynamic-priority hard real-time systems using slack time analysis," in *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings*, p. 788, Mar. 2002.

[8] W. Kim, D. Shin, H.-S. Yun, J. Kim, and S. L. Min, "Performance comparison of dynamic voltage scaling algorithms for hard real-time systems," in *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings.*, p. 219, Sept. 2002.

[9] J. Lorch and A. Smith, "Using user interface event information in dynamic voltage scaling algorithms," in *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*, pp. 46–55, Oct. 2003.

[10] T. Pering, T. Burd, and R. Brodersen, "Dynamic voltage scaling and the design of a low-power microprocessor system," in *Power Driven Microarchitecture Workshop*, ISCA98, 1998.

[11] T. Pering, T. Burd, and R. Brodersen, "Voltage scheduling in the iparm microprocessor system," in *ISLPED '00*, (Rapallo, Italy), p. 96, 2000.

[12] J. F. Mao, Q. C. Zhao, and C. G. Cassandras, "Optimal dynamic voltage scaling in power-limited systems with real-time constraints," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004.

[13] Y. Cho, C. Cassandras, and D. Pepyne, "Forward decomposition algorithms for optimal control of a class of hybrid systems," *Intl.J. of Robust and Nonlinear Control*, vol. 11(5), pp. 497–513, 1999.

[14] L. Miao and C. Cassandras, "Optimality of static control policies in some discrete event systems," *submitted to IEEE Transactions on Automatic Control*.

[15] C. G. Cassandras and R. Mookherjee, "Receding horizon control for a class of hybrid systems with event uncertainties," in *Proc. of 2003 American Control Conf.*, pp. 413–418, June 2003.