Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

On-line Optimal Control of a Class of Discrete Event Systems with Real-Time Constraints

Jianfeng Mao and **Christos G. Cassandras**
Division of Systems Engineering
and Center for Information and Systems Engineering
Boston University
Brookline, MA 02446
jfmao@bu.edu, cgc@bu.edu

***Abstract*— We consider Discrete Event Systems involving tasks with real-time constraints and seek to control processing times so as to minimize a cost function subject to each task meeting its own constraint. It has been shown that the off-line version of this problem can be efficiently solved by the *Critical Task Decomposition Algorithm* [9]. The on-line version has been dealt with to date using worst-case analysis so as to bypass the complexity of random effects. This approach, however, does not make use of probability distributions and results in an overly conservative solution. In this paper, we develop a new on-line algorithm without relying on worst-case analysis, in which a "best solution in probability" can be efficiently obtained by estimating the probability distribution of the off-line optimal control. We introduce a condition termed "non-singularity" under which the best solution in probability leads to the on-line optimal control. Numerical examples are included to illustrate our results and show substantial performance improvements over worst-case analysis.**

Keywords: **on-line optimal control, discrete event system, real-time constraints**

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

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

where a_i is the arrival time of task i, x_i is its completion time, and $s_i(u_i)$ is its service time which may be controllable through u_i. 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 [12]; in computer systems, where the CPU speed can be controlled to ensure that certain tasks meet specified execution deadlines [2],[6]; and in wireless networks where severe battery limitations call for new techniques aimed at maximizing the lifetime of such a network [3],[10]. A particularly interesting class of problems arises when such systems are subject to *real-time constraints*, i.e., $x_i \le d_i$ for each task i with a given "deadline" d_i. In order to meet such constraints, one typically has to incur a higher cost associated with control u_i. Thus, in a broader context, we are interested in studying optimization problems of the form

$$\begin{aligned} &\min_{u_1,\dots,u_N} \sum_{i=1}^{N} \theta_i(u_i) \\ \text{s.t.} \quad & x_i = \max(x_{i-1}, a_i) + s_i(u_i), \quad i = 1, ..., N; \qquad (2) \\ & x_i \le d_i, \ s_{\min,i} \le s_i(u_i) \le s_{\max,i}, \ i = 1, ..., N. \end{aligned}$$

where $s_{\min,i}, \ s_{\max,i} > 0$ are the lower and upper bound on the service time of task i respectively and $\theta_i(u_i)$ is a given cost function. Such problems have been studied for preemptive tasks [1],[13], nonpreemptive periodic tasks [4],[5] , and nonpreemptive aperiodic tasks [3] ,[10],[9]. The latter case is of particular interest in wireless communications where nonpreemptive scheduling is necessary to execute aperiodic packet transmission tasks which also happen to be highly energy-intensive; in such cases, the cost function in (2) represents the energy required for a packet transmission. One of the key challenges in dealing with (2) is to develop computationally efficient solution approaches that can be used in real-time settings and can be implemented in wireless devices with very limited computational power.

In general, this is a hard nonlinear optimization problem, complicated by the inequality constraints $x_i \le d_i$ and the nondifferentiable max operator involved. Nonetheless, it was shown in [9] that when $\theta_i(u_i)$ is convex and differentiable the solution to such problems is characterized by attractive structural properties leading to a highly efficient algorithm termed *Critical Task Decomposition Algorithm* (CTDA) [9] and Generalized CTDA [10]. The CTDA does not require any numerical optimization problem solver, but only needs to identify a set of "critical" tasks. The efficiency of the CTDA is crucial for applications where small, inexpensive devices are required to perform on-line computations with minimal on-board resources.

The on-line version of problem (2) arises when arrival times of tasks and task characteristics (e.g., their deadlines and sizes) are random and not known in advance. One way to bypass the complexity of such random effects is by using worst-case analysis (as in [9], [11]). However, there are

The authors' work is supported in part by NSF under Grants DMI-0330171 and EFRI-0735974, by AFOSR under grants FA9550-04-1-0133 and FA9550-04-1-0208, and by DOE under grant DE-FG52-06NA27490.

several disadvantages of a worst-case analysis approach: **(i)** the probability distribution information cannot be utilized; **(ii)** the decisions made based on worst-case analysis are too conservative, especially when the time horizon is very short (available future information is limited); **(iii)** only arrival times of tasks are assumed to be uncertain. If other task information is also uncertain, such as their deadlines and sizes, worst-case analysis will become more complicated and conservative; **(iv)** it fails when the release time jitter [6] is hard or impossible to estimate in advance (e.g., when inter-arrival time probability distributions have infinite support as in the common exponential distribution case).

In this paper, we develop a new optimal control approach to solve the on-line problem without relying on worst-case analysis. In this approach, all task information may be uncertain and real-time constraints are imposed so that the probability that $x_i \leq d_i$ in (2) is greater than a prespecified value. Since this probability cannot be analytically calculated when task characteristics are all random, this is a hard stochastic optimization problem. It is necessary to invoke simulation-based methods to estimate various quantities of interest, which renders highly questionable the feasibility of a realistic on-line algorithm. In fact, we show that a typical such approach is of complexity $O(IMN^2)$ where N is the number of tasks involved, M is the number of sample paths simulated for estimation purposes, and I is the number of iterations required for the optimization algorithm to converge. We introduce a condition termed *non-singularity*, under which the solution to the on-line problem is obtained in $O(MN + M\log M)$ complexity, leading to a much faster process amenable to on-line control.

In Section II, we formulate the on-line optimization problem. In Section III, we study the feasible control set for this problem, whose determination is complicated by the probabilistic real-time constraints. In Section IV, we introduce the non-singularity condition and its ramifications, leading to an algorithm for deriving a complete solution of the problem referred to as the "best solution in probability". Simulation results are given in Section V illustrating the on-line capability of the proposed approach and we close with conclusions presented in Section VI.

II. ON-LINE PROBLEM FORMULATION

In what follows, we concentrate on the control u_i being the processing rate and set $u_i = 1/\tau_i$ where τ_i is the processing time per operation in a task. If a task consists of μ_i operations (i.e., the size of the task), then we have $s_i(u_i) = \mu_i\tau_i$ and $\theta_i(u_i) = \mu_i\theta_i(\tau_i)$. Then, the off-line problem (2) becomes:

$$\begin{aligned} &\min_{\tau_1,\dots,\tau_N} \sum_{i=1}^{N} \mu_i\theta_i(\tau_i) \\ \text{s.t.}\quad & x_i = \max(x_{i-1}, a_i) + \mu_i\tau_i, \qquad i = 1,...,N; \\ & x_i \leq d_i,\ \tau_{\min} \leq \tau_i \leq \tau_{\max}, \qquad i = 1,...,N. \end{aligned} \tag{3}$$

where $\theta_i(\tau_i)$ is assumed to be monotonically decreasing and convex in τ_i and $\tau_{\min}$ and $\tau_{\max}$ are the minimal and maximal processing time per operation respectively. This off-line problem is deterministic because all task information a_i, d_i and μ_i, is assumed to be known. Thus, open-loop control is as good as closed-loop control for this case and we can obtain an optimal control for all tasks off line.

However, in practice, arrival times may be unknown. In fact, usually a_i and d_i cannot be known until task i arrives and μ_i cannot be acquired until task i completes. Only their probability distribution can be assumed known or estimated in advance from past history. Due to these uncertainties, closed-loop control is preferable, which necessitates an on-line optimization approach. In this manner, we have the opportunity to observe new information and update controls by solving on-line problems at a set of decision points. Generally, decision points can be arbitrarily selected and could be task departure times, arrival times or instants when some other specific events occur. From a practical standpoint, updating controls upon each arrival time can be problematic when arrivals are bursty, in which case it is even possible that the calculation of new controls takes longer than an inter-arrival time and this can lead to unstable behavior. In this paper, we choose task start times, i.e., $\max(x_{k-1}, a_k)$, $k = 1, 2, \dots$, to be these decision points.

Assume the current decision time is $\max(x_{k-1}, a_k)$ and the related on-line control is τ_k. The objective of the on-line problem is to minimize the expected cost of the current task k and all future incoming tasks, that is

$$\min_{\tau_k} E\{\mu_k\theta_k(\tau_k) + L(\tau_k, \mathcal{S}_k)\} \tag{4}$$

where $\mu_k\theta_k(\tau_k)$ is the cost of the current task k and $\mathcal{S}_k$ is a state vector defined to include all deterministic task information available at the current decision time. For example, assume there are Q tasks in queue and the arrival times $a_k, ..., a_{k+Q-1}$ and the deadlines $d_k, ..., d_{k+Q-1}$ can be observed. Then, the state vector is $\mathcal{S}_k = [a_k, \dots, a_{k+Q-1}, d_k, \dots, d_{k+Q-1}]^T$. Thus, $L(\tau_k, \mathcal{S}_k)$ is the *optimal cost* of all future incoming tasks under control τ_k when the state is $\mathcal{S}_k$. If the number of incoming tasks is infinite, then $L(\tau_k, \mathcal{S}_k)$ cannot be obtained and it is necessary to approximate it by the optimal cost of the next N (sufficiently large) tasks, denoted by $\hat{L}(\tau_k, \mathcal{S}_k)$. We will give a precise definition of $\hat{L}(\tau_k, \mathcal{S}_k)$ in Section IV.

In formulating the on-line problem, we also need to consider the effect of the control τ_k on the real-time constraints. A larger τ_k may result in a lower expected cost, but it may also cause a higher probability of violating the deadlines of some future tasks. To establish a guarantee for real-time constraints, we need to set up an acceptable lower bound p on the probability of satisfying all constraints. In other words, we need to quantify how likely is the existence of some $\tau_i \in [\tau_{\min}, \tau_{\max}]$ for $i > k$ such that $x_i \leq d_i$ for $i \geq k$ when the control τ_k is applied. Note that since $\tau_{\min}$ is the minimum processing time per operation, it follows that departure times x_i for $i > k$ are minimized when $\tau_i = \tau_{\min}$. Therefore, the event [there exists $\tau_i \in [\tau_{\min}, \tau_{\max}]$ such that $x_i \leq d_i$ for $i \geq k$] is equivalent to the event [$x_i \leq d_i$ for $i \geq k$ when $\tau_i = \tau_{\min}$]. The former obviously implies the latter by the previous observation and the latter implies the former by selecting $\tau_i = \tau_{\min}$ for all $i > k$. Based on the discussion above, let

$$J(\tau_k, \mathcal{S}_k) = \mu_k \theta_k(\tau_k) + \hat{L}\big(\tau_k, \mathcal{S}_k\big) \tag{5}$$

and we define the on-line problem as follows:

$$\begin{aligned} &\min_{\tau_k} E\big\{J(\tau_k, \mathcal{S}_k)\big\} \\ \text{s.t.} \quad & x_k = \max(a_k, x_{k-1}) + \mu_k \tau_k; \qquad (6) \\ & x_i = \max(a_i, x_{i-1}) + \mu_i \tau_{\min},\ i = k+1, ..., k+N; \\ & P[x_i \le d_i, \forall\ i = k, \ldots, k+N] \ge p. \end{aligned}$$

where the last two constraints capture the requirement $P[$there exists $\tau_i \in [\tau_{\min}, \tau_{\max}]$ such that $x_i \le d_i$ for $i \ge k] \ge p$ as explained above.

It should be noted that the optimal solution of problem (6) depends only on the state $\mathcal{S}_k$ when all stochastic processes (describing arrival times, deadlines, and task sizes) are stationary and corresponding probability distributions can be accurately obtained in advance. If the state space is finite, it is possible to compute the optimal solution of problem (6) for each state off line. We allow, however, the state space to be infinite because a_i and d_i are generally real-valued variables. Even if we can discretize the state space, observe that it still grows exponentially when more information is observed. Moreover, if probability distributions are not a priori available, they need to be estimated based on observed data, hence an on-line algorithm is necessary. Finally, we should mention that in the remainder of this paper we assume that problem (6) is feasible, i.e., the last constraint is satisfied. If that is not the case, our analysis is still valid but only after an admission control problem is first solved, where the objective is to ensure that as many tasks as possible meet their deadlines by rejecting some tasks; this problem is treated in [8].

III. Feasible Control Set

Before solving the on-line problem (6), we need to identify the feasible control set for τ_k. The difficulty in doing so comes from the last constraint where $P[x_i \le d_i, \forall\ i = k, ..., k+N]$ is a function of the control τ_k since x_i for $i = k, \ldots, k+N$ only depends on τ_k. For convenience, let $F(\tau_k)$ denote this probability when τ_k is selected. We establish a property of $F(\tau_k)$ in Lemma 1 below based on the following auxiliary problem:

$$\begin{aligned} &\max_{\tau_k}\ \tau_k \\ \text{s.t.} \quad & x_k = \max(x_{k-1}, a_k) + \mu_k \tau_k \le d_k; \qquad (7) \\ & x_i = \max(x_{i-1}, a_i) + \mu_i \tau_{\min} \le d_i,\ i \in [k+1, k+N]. \end{aligned}$$

The optimal solution of this problem is denoted by $\bar{\tau}_k$ and can be interpreted as the exact feasible upper bound of τ_k for a problem when all a_i, d_i and μ_i are known. In fact, if there are Q tasks in queue at the kth decision time (generally $Q \le N$), only information on these tasks is known. All remaining a_i, d_i and μ_i for $i > k+Q$ are random, so $\bar{\tau}_k$ is also a random variable. Strictly speaking, we should write $\bar{\tau}_k(\mathcal{S}_k)$ but omit this dependence for simplicity. In the following lemma, we show that $F(\tau_k)$ is the complementary cumulative distribution function of $\bar{\tau}_k$. (The proofs in this paper are omitted; the full proofs can be found in [7].)

Lemma 1: $F(\tau_k) = P[\bar{\tau}_k \ge \tau_k]$.

Let $F^{-1}(\cdot)$ denote the inverse function of $F(\cdot)$ and

$$\tau_k^p = \sup_{\tau} \big\{\tau : \tau = F^{-1}(p)\big\} \tag{8}$$

Lemma 1 implies that the probabilistic constraint $P[x_i \le d_i, \forall\ i = k, ..., k+N] \ge p$ in the on-line problem (6) is equivalent to the constraint $\tau_k \le \tau_k^p$. Therefore, the feasible control set becomes

$$\tau_{\min} \le \tau_k \le \min(\tau_{\max}, \tau_k^p). \tag{9}$$

To determine τ_k^p through (8) for any given p, we need $F(\tau_k)$. However, $F(\tau_k)$ is unknown and cannot be derived in closed form. One way to estimate it is through a Monte Carlo simulation method as follows. Suppose a sample path is generated based on tasks indexed by $i = k+1, \ldots, k+N$ with arrivals a_i, deadlines d_i, and number of operations μ_i. Given this information, (7) becomes a deterministic optimization problem. Note that, given the state $\mathcal{S}_k$ for Q tasks already in queue, only data for $N-Q$ future tasks need to be randomly generated; in addition, these data can be generated a priori when probability distributions are available, thus substantially reducing the burden of this process during on-line execution. Now, suppose there are M sample paths generated this way indexed by $j = 1, \ldots, M$ and let $\bar{\tau}_k^j$ be the solution of (7) in the jth sample path. Let $Z_j(\tau_k) = \mathbf{1}[\tau_k \ge \bar{\tau}_k^j]$, where $\mathbf{1}[\cdot]$ is an indicator function. Then, $F(\tau_k)$ can be estimated by $\hat{F}_M(\tau_k) = \frac{\sum_{j=1}^{M} Z_j(\tau_k)}{M}$. Let $\hat{\tau}_{k,M}^p = \sup_\tau \big\{\tau : \tau = \hat{F}_M^{-1}(p)\big\}$, where $\hat{F}_M^{-1}(\cdot)$ is the inverse function of $\hat{F}_M(\cdot)$. By the strong law of large numbers, $\hat{F}_M(\tau_k^p)$ converges to $F(\tau_k^p) = p$ w.p.1 as $M \to \infty$, as long as the M sample pathes are independently generated. Combining this with $\hat{\tau}_{k,M}^p = \hat{F}_M^{-1}(p)$ and $\tau_k^p = F^{-1}(p)$, we also conclude that $\hat{\tau}_{k,M}^p$ converges to τ_k^p w.p.1 as $M \to \infty$. Moreover, we can show that this convergence is such that $\hat{\tau}_{k,M}^p$ approaches τ_k^p exponentially fast as M increases.

Lemma 2: For any $\epsilon > 0$, there always exists $C > 0$ such that $P\big[\,|\tau_k^p - \hat{\tau}_{k,M}^p| \ge \epsilon\big] \le 2e^{-CM}$.

Thus, we may use $\hat{\tau}_{k,M}^p$ as an estimate of τ_k^p in (9) so as to specify the feasible set of the on-line problem. To do so, however, we need an efficient solution of (7) which will provide us with $\bar{\tau}_k^j$, $j = 1, \ldots, M$, and hence $\hat{F}_M(\tau_k)$. This is accomplished through the simple algorithm in Table I in $O(N)$ complexity. Note that in Step **1)**, if $d_l - \mu_l \tau_{\min} \ge d_{l-1}$, then τ_k is independent of all tasks after task l and the solution involves only tasks prior to l.

TABLE I. Solution of the problem (7)

1) Find the first task l such that $d_l - \mu_l \tau_{\min} \ge d_{l-1}$ beginning from $l = k+1$. If there is no such l, then set $l = k+N$;
2) $\tau_k = d_l - \sum_{i=2}^{l} \mu_i \tau_{\min} - \max(a_k, x_{k-1})$.

Once we obtain the M exact feasible upper bounds $\bar{\tau}_k^1, ..., \bar{\tau}_k^M$, we sort them as $\bar{\tau}_k^{(1)} \le \cdots \le \bar{\tau}_k^{(M)}$ and $\hat{F}_M(\tau_k)$

is obtained as illustrated in Fig. 1 and $\hat{\tau}^p_{k,M}$ is immediately derived through $\hat{F}^{-1}_M(p)$. In summary, $\hat{\tau}^p_{k,M}$, and hence an estimate of the feasible control set in (9), is obtained through the algorithm in Table II in $O(MN + M\log M)$ complexity.

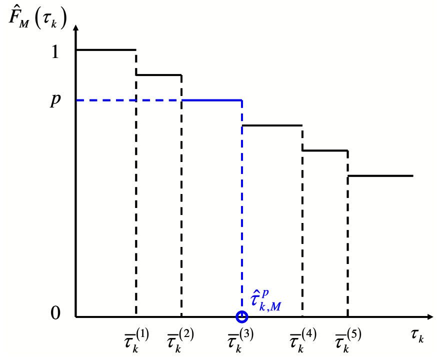

Fig. 1. An example of $\hat{F}_M(\tau_k)$ and $\tau^p_{k,M}$

TABLE II. DETERMINING $\hat{\tau}^p_{k,M}$

1)	Randomly generate M sample paths;
2)	Obtain $\bar{\tau}^j_k$ by applying the algorithm in Table I for $j = 1, ..., M$;
3)	Sort $\bar{\tau}^1_k, ..., \bar{\tau}^M_k$ to derive $F_M(\tau_k)$ and then $\tau^p_{k,M} = F^{-1}_M(p)$.

IV. BEST SOLUTION IN PROBABILITY

In this section, we provide a solution of the on-line problem (6) under a condition we term "non-singularity". To do so, we first need to define $\hat{L}(\tau_k, \mathcal{S}_k)$ in (5). As already mentioned, $\hat{L}(\tau_k, \mathcal{S}_k)$ is the optimal cost of the next N tasks under the control τ_k and the state $\mathcal{S}_k$, which can be obtained from the solution of the problem

$$\hat{L}(\tau_k, \mathcal{S}_k) := \min_{\tau_{k+1},\ldots,\tau_{k+N}} \sum_{i=k+1}^{k+N} \mu_i\theta_i(\tau_i) \qquad (10)$$

$$\text{s.t.} \quad x_i = \max(a_i, x_{i-1}) + \mu_i\tau_i,\ i = k, \ldots, k+N;$$
$$\tau_{\min} \le \tau_i \le \tau_{\max},\ x_i \le d_i,\ i = k, \ldots, k+N.$$

for any sample path specified by a_i, d_i and μ_i, $i = k, ..., k+N$. We note that this is of the same form as the off-line problem (2) and could be solved very efficiently through the CTDA [9] mentioned earlier if all a_i, d_i and μ_i were known.

Clearly, a closed-form expression for $E\{J(\tau_k, \mathcal{S}_k)\}$ in (6) cannot be derived and has to be estimated. If we proceed via Monte Carlo simulation, there are three notable difficulties: **(i)** it is costly to evaluate $\hat{L}(\tau_k, \mathcal{S}_k)$ for each τ_k. Assume we randomly generate M sample paths (i.e., realizations of a_i, d_i and μ_i, $i = k, ..., k+N$) and solve problem (10) for each sample path. Since problem (10) can be solved in $O(N^2)$ by using the CTDA [9], the complexity of this process is $O(MN^2)$; **(ii)** both $dE\big(J(\tau_k, \mathcal{S}_k)\big)/d\tau_k$ and $dJ(\tau_k, \mathcal{S}_k)/d\tau_k$ are hard to compute because $J(\tau_k, \mathcal{S}_k)$ involves $\hat{L}(\tau_k, \mathcal{S}_k)$ that has no closed form. Only finite differences can be obtained, which costs two time-consuming evaluations; **(iii)** it may take many iterations to converge to the optimal solution of (6). Assuming the total number of iterations is I, the total complexity of solving the on-line problem is $O(IMN^2)$ where I, M and N are usually very large. Such huge complexity is not suitable for on-line control.

In the following, we will bypass much of this complexity by developing an efficient algorithm based on the condition defined below.

Non-singularity Condition (NSC):

$$P\left[J(\tau'_k, \mathcal{S}_k) \le J(\tau''_k, \mathcal{S}_k)\right] \ge 0.5$$
$$\Longrightarrow E\left[J(\tau'_k, \mathcal{S}_k)\right] \le E\left[J(\tau''_k, \mathcal{S}_k)\right]$$

The interpretation here is that if some control action τ'_k is more likely better than τ''_k (in the sense of resulting in lower cost), then the expected cost under τ'_k will be lower than the one under τ''_k. This is consistent with common sense in that any action A more likely better than B should result in A's expected performance being better than B's. Only "singularities" such as $J(\tau'_k) \gg J(\tau''_k)$ with an unusually low probability for some (τ'_k, τ''_k) can affect the corresponding expectations so that this condition may be violated. It is straightforward to verify this NSC for several common cases; for example, consider $\min_x E(x-Y)^2$, where Y is a uniform random variable over $[a, b]$. The optimal solution $(a+b)/2$ satisfies the NSC. Based on the NSC, we define the "Best Solution in Probability" below:

Definition 1: τ^*_k is the *Best Solution in Probability(BSIP)* if and only if τ^*_k satisfies, for all $\tau_k \in [\tau_{\min}, \min(\tau_{\max}, \tau^p_k)]$,

$$P\left[J(\tau^*_k, \mathcal{S}_k) \le J(\tau_k, \mathcal{S}_k)\right] \ge 0.5$$

A natural question that arises is whether it is possible for a better solution $\tau'_k \ne \tau^*_k$ to exist such that, for all $\tau_k \in [\tau_{\min}, \min(\tau_{\max}, \tau^p_k)]$,

$$P\left[J(\tau'_k, \mathcal{S}_k) \le J(\tau_k, \mathcal{S}_k)\right] \ge q, \quad \text{for some } q > 0.5. \quad (11)$$

The lemma below shows that any such τ'_k coincides with τ^*_k.

Lemma 3: If τ'_k satisfies (11), then $\tau'_k = \tau^*_k$.

Based on Lemma 3, we can define the BSIP as the one satisfying (16). Moreover, if there exists a BSIP τ^*_k, then based on the NSC, it satisfies $E\left\{J(\tau^*_k, \mathcal{S}_k)\right\} \le E\left\{J(\tau_k, \mathcal{S}_k)\right\}$ for all $\tau_k \in [\tau_{\min}, \min(\tau_{\max}, \tau^p_k)]$, that is, τ^*_k is also the optimal solution of the on-line problem (6). In the following, we will prove the existence of the BSIP using a construction method. Then, an algorithm is developed to determine the BSIP in $O(MN)$ complexity. To begin with, we exploit a property of $\hat{L}(\tau_k, \mathcal{S}_k)$ based on the convexity of $\theta_i(\tau_i)$.

Lemma 4: $\hat{L}(\tau_k, \mathcal{S}_k)$ is convex with respect to τ_k.

Next, recalling (5) and (10), we consider another auxiliary problem: $\min_{\tau_k} J(\tau_k, \mathcal{S}_k)$, whose optimal solution is denoted by $\tilde{\tau}_k$. The domain of the function $\hat{L}(\tau_k, \mathcal{S}_k)$ is $[\tau_{\min}, \min(\tau_{\max}, \bar{\tau}_k)]$, in which $\bar{\tau}_k$ is a random variable and its value depends on the related sample path. Thus, $\bar{\tau}_k$ may be less than τ^p_k, that is, a feasible solution of the on-line problem (6) may not guarantee all the real-time constraints in $\hat{L}(\tau_k, \mathcal{S}_k)$ for some specific sample path. In order to derive $E\{J(\tau_k, \mathcal{S}_k)\}$ for any τ_k such that $\tau_k > \bar{\tau}_k$ for some sample paths, we have to assign an appropriate value to $J(\tau_k, \mathcal{S}_k)$ when $\tau_k > \bar{\tau}_k$. The common way is to set $J(\tau_k, \mathcal{S}_k) =$

$J(\bar{\tau}_k, \mathcal{S}_k) + \alpha(\tau_k) \cdot \mathbf{1}[\tau_k > \bar{\tau}_k]$, where $\alpha(\tau_k) > 0$ is a penalty function for $\tau_k > \bar{\tau}_k$, monotonically non-decreasing in τ_k, which ensures $J(\tau_k, \mathcal{S}_k) > J(\bar{\tau}_k, \mathcal{S}_k)$ for all $\tau_k > \bar{\tau}_k$. Thus, we have $\tilde{\tau}_k \leq \bar{\tau}_k$. As seen next, our approach is based on obtaining τ_k^* not by computing $E\{J(\tau_k, \mathcal{S}_k)\}$ but rather $\tilde{\tau}_k$ and its cumulative distribution function. Since $\tilde{\tau}_k \leq \bar{\tau}_k$, an advantage of this approach is that we need not concern ourselves with the penalty $\alpha(\tau_k)$, thus saving the effort of specifying an appropriate such function. We can obtain $\tilde{\tau}_k$ by solving the problem below

$$\min_{\tau_k,\dots,\tau_{k+N}} \left\{ \sum_{i=k}^{k+N} \mu_i \theta_i(\tau_i) \right\} \quad (12)$$

$$\text{s.t.} \quad x_i = \max(a_i, x_{i-1}) + \mu_i \tau_i, \ i = k, \dots, k+N;$$
$$\tau_{\min} \leq \tau_i \leq \tau_{\max}, \ x_i \leq d_i, \ i = k, \dots, k+N.$$

where $\tilde{\tau}_k$ is the optimal solution of (12) for task k. Problem (12) can be regarded as an off-line problem like (3) with a_i, d_i and μ_i, $i = k, ..., k+N$, the given arrival times, deadlines and number of operations. This can be efficiently solved by the CTDA [9]. Moreover, since only the optimal solution for task k is needed, we can obtain $\tilde{\tau}_k$ in $O(N)$ complexity without solving the whole problem (12) [9]. Since a_i, d_i and μ_i are random, $\tilde{\tau}_k$ is also a random variable. As in the case of the auxiliary problem (7), strictly speaking, we should write $\tilde{\tau}_k(\mathcal{S}_k)$ but omit this dependence for simplicity.

The solution $\tilde{\tau}_k$ has the following properties which are easily established as a corollary of Lemma 4:

Corollary 1: Assume $\tau_{\min} \leq \tau_k' < \tau_k'' \leq \tau_{\max}$. Then,

$$\tilde{\tau}_k \leq \tau_k' \Longrightarrow J(\tau_k', \mathcal{S}_k) \leq J(\tau_k'', \mathcal{S}_k);$$
$$\tilde{\tau}_k \geq \tau_k'' \Longrightarrow J(\tau_k'', \mathcal{S}_k) \leq J(\tau_k', \mathcal{S}_k).$$

By Corollary 1, we can obtain an additional property of $\tilde{\tau}_k$.

Lemma 5: Assume $\tau_{\min} \leq \tau_k' < \tau_k'' \leq \tau_{\max}$. Then,

$$P\left[J(\tau_k', \mathcal{S}_k) \leq J(\tau_k'', \mathcal{S}_k)\right] \geq P[\tilde{\tau}_k \leq \tau_k'];$$
$$P\left[J(\tau_k', \mathcal{S}_k) \geq J(\tau_k'', \mathcal{S}_k)\right] \geq P[\tilde{\tau}_k \geq \tau_k''].$$

Define $G(\tau_k) = P[\tilde{\tau}_k \leq \tau_k]$ and $\tau_k^h = \sup_\tau \{\tau : \tau = G^{-1}(0.5)\}$, where $G^{-1}(\cdot)$ is the inverse function of $G(\cdot)$. Using Lemma 5, we can establish the following result.

Theorem 1: For any $\tau_k \in [\tau_{\min}, \tau_{\max}]$,

$$P\left[J(\tau_k^h, \mathcal{S}_k) \leq J(\tau_k, \mathcal{S}_k)\right] \geq 0.5 \quad (13)$$

Based on Theorem 1, we can obtain τ_k^h through the cumulative distribution function of $\tilde{\tau}_k$, $G(\tau_k)$. Although τ_k^h satisfies (13), it still may not be the BSIP since the feasible control set in (9) also requires that $\tau_k \leq \tau_k^p$. Theorem 2 below provides the complete final solution.

Theorem 2: The BSIP τ_k^* satisfies $\tau_k^* = \min(\tau_k^h, \tau_k^p)$.

This result provides the BSIP in terms of τ_k^h and τ_k^p. Just as τ_k^p in the previous section had to be estimated by estimating $F(\tau_k)$, similarly we need to estimate τ_k^h by estimating $G(\tau_k)$ which is not available in closed form. Once again, we can resort to a Monte Carlo simulation method, in which we generate M sample paths where a sample path is generated based on tasks indexed by $i = k, \dots, k+N$ with arrivals a_i, deadlines d_i, and number of operations μ_i. As in the estimation of $F(\tau_k)$ in the previous section, given the state $\mathcal{S}_k$ for Q tasks already in queue, only data for $N - Q$ future tasks are needed and could in fact be available from prior off-line generation. Suppose there are M sample paths indexed by $j = 1, \dots, M$ and let $\tilde{\tau}_k^j$ denote the solution of minimizing $J(\tau_k)$ in the jth sample path. Then, set $Z_j(\tau_k) = \mathbf{1}[\tilde{\tau}_k^j \leq \tau_k]$ and $G(\tau_k)$ can be estimated by $\hat{G}_M(\tau_k) = \frac{\sum_{j=1}^M Z_j(\tau_k)}{M}$. Let $\hat{\tau}_{k,M}^h = \sup_\tau \{\tau : \tau = \hat{G}_M^{-1}(0.5)\}$. Based on the strong law of large number, $\hat{G}_M(\tau_k^h)$ converges to $G(\tau_k^h)$ w.p.1 as $M \to +\infty$. Combining this with $\hat{\tau}_{k,M}^h = \hat{G}_M^{-1}(0.5)$ and $\tau_k^h = G^{-1}(0.5)$, $\hat{\tau}_{k,M}^h$ also converges to τ_k^h w.p.1 as $M \to +\infty$. Furthermore, using the Chernoff bound and an argument similar to that in Lemma 2, we can show that $\hat{\tau}_{k,M}^h$ approaches τ_k^h exponentially fast as M increases, that is,

Lemma 6: For any $\epsilon > 0$, there always exists $C > 0$ such that $P\left[|\tau_k^h - \hat{\tau}_{k,M}^h| \geq \epsilon\right] \leq 2e^{-CM}$.

The analysis above leads to the algorithm in Table III through which we can obtain the estimate $\hat{\tau}_{k,M}^h$ of τ_k^h. In Step **2)**, we only solve M off-line problems (for each sample path) without any iterative process which a traditional stochastic programming method would require, hence having to solve IM off-line problems where I is the number of iterations. Each such problem can be very efficiently solved in $O(N)$ complexity using the CTDA [9]. Note that CTDA's worst case complexity is in fact $O(N^2)$, but the problem at hand involves solving for τ_k only, i.e., the first task and not all $k, \dots, k+N$ tasks, which reduces to $O(N)$. In Step **3)**, we obtain $\hat{G}_M(\tau_k)$ by sorting $\tilde{\tau}_k^1, \dots, \tilde{\tau}_k^M$ similar to Fig. 1. Deriving $\hat{\tau}_{k,M}^h$ is accomplished in $O(MN + M\log M)$ complexity, which is clearly a vast improvement over $O(IMN^2)$.

Finally, combining Tables II and III we can obtain an estimate of the BSIP τ_k^* as $\min(\hat{\tau}_{k,M}^p, \hat{\tau}_{k,M}^h)$. Of course, it remains an open problem whether the NSC is satisfied in this particular problem. If so, the BSIP is an estimate of the optimal solution of the on-line problem (6) which we have seen converges to the true solution exponentially fast. Otherwise, the BSIP is a sub-optimal solution which we expect to be quite close to optimal.

TABLE III. DETERMINING $\hat{\tau}_{k,M}^h$

1) Randomly generate M sample paths;
2) Obtain τ_k^j minimizing $J(\tau_k, \mathcal{S}_k)$ in the jth sample path by applying CTDA for $j = 1, ..., M$;
3) Sorting $\tilde{\tau}_k^1, ..., \tilde{\tau}_k^M$ to derive $\hat{G}_M(\tau_k)$ and then $\hat{\tau}_{k,M}^h = \hat{G}_M^{-1}(0.5)$.

V. SIMULATION RESULTS

In this section, we compare the performance of our on-line algorithm to the method based on worst-case analysis, in which $\tau_{\min} = 1$, $\tau_{\max} = 10$, the arrival time a_i is uniformly distributed in some release jitter interval $[a_i^-, a_i^- + 4]$, the deadline d_i is uniformly distributed in $[a_i + 20, a_i + 40]$, μ_i is a random integer uniformly distributed in $\{1, \dots, 5\}$ and $\theta(\tau_i) = 1/(\tau_i - 0.5)^2$. We select $p = 1$, i.e., all tasks are required to meet their deadlines. A total of 1600 tasks are processed and we always look ahead $N = 100$ tasks at each decision point. All information on tasks that arrived before the current decision time is available, while only a probability distribution is known for future tasks. In the worst-case analysis, the probability distribution is not utilized and we set a_i, d_i and μ_i for all future tasks as the earliest arrival time a_i^-, the tightest deadline $a_i^- + 20$ and largest size 5 respectively to guarantee the real-time constraints. Then, the control for the current task k is obtained by solving an off-line problem with a_i^-, $a_i^- + 20$ and 5 as the values of arrival times, deadlines and sizes respectively for all tasks arriving after the decision time.

Define the following three costs: (i) $C^*(i)$ is the sum of the optimal costs from task 1 to task i for the ideal model, i.e., when all a_i, d_i and μ_i are known in advance, (ii) $C^w(i)$ is the sum of costs from task 1 to task i by applying worst-case analysis, and (iii) $C^b(i)$ is the sum of costs from task 1 to task i by applying our on-line algorithm where we choose $M = 500$. Based on these costs, we can define two relative performance ratios: $\lambda^b(i) = \frac{C^b(i)-C^*(i)}{C^*(i)}$ and $\lambda^w(i) = \frac{C^w(i)-C^*(i)}{C^*(i)}$. The comparison results are shown in Figs. 2 and 3. Observe that the relative ratio of the worst-case analysis method converges to $\lambda^w(1600) = 2.06$ and the one of our on-line algorithm to $\lambda^b(1600) = 0.13$, an order of magnitude better. In particular, the solution obtained by our on-line algorithm has a 13% larger cost than the ideal optimal cost, while the worst-case analysis method results in a much more conservative solution, whose cost is 206% larger than the ideal cost.

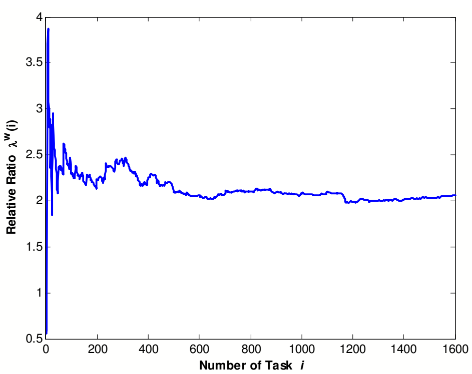

Fig. 2. Relative ratio $\lambda^w(i)$

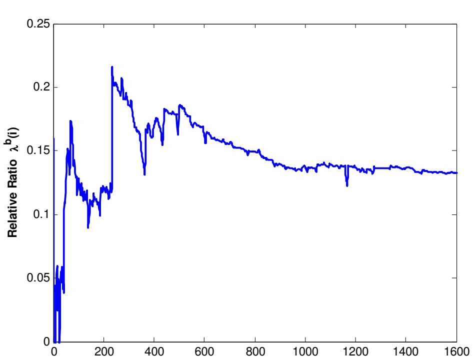

Fig. 3. Relative ratio $\lambda^b(i)$

VI. Conclusions

We have revisited the on-line version of optimization problems encountered in discrete event systems processing tasks with hard real-time constraints. In this case, arrival times of tasks and their deadlines and sizes are unknown in advance. Rather than a worst-case analysis (pursued elsewhere), we make use of probability distributions, which generally leads to less conservative solutions. We propose a condition termed "non-singularity condition" (NSC) based on which we obtain an algorithm that provides a "best solution in probability". This solution estimates the on-line optimal control (and converges to it exponentially fast) if the non-singularity condition holds and otherwise provides suboptimal solutions. Empirical results to date indicate significant performance improvements over worst-case analysis.

Future work is aiming at studying the validity and range of the NSC, that is, identifying the kinds of problem that satisfy it. Another natural direction is to develop an efficient way to improve the best solution in probability when the NSC is not satisfied. Moreover, we assume that information on probability distributions for task arrivals, deadlines, and sizes is known beforehand. However, in some applications only rough information of this type may be available. We plan to incorporate a learning algorithm to estimate these probability distributions based on past history and study their convergence properties when stationarity applies.

References

[1] H. Aydin, R. Melhem, D. Mossé, and P. Mejia-Alvarez. Power-aware scheduling for periodic real-time tasks. *IEEE Trans. on Computers*, 53(5):584 – 600, May 2004.

[2] G.C. Buttazzo. *Hard Real-time Computing Systems: Predictable Scheduling Algorithms and Applications*. Kluwer Academic Publishers, Norwell, MA, 1997.

[3] A. E. Gamal, C. Nair, B. Prabhakar, Elif Uysal-Biyikoglu, and S. Zahedi. Energy-efficient scheduling of packet transmissions over wireless networks. In *Proceedings of IEEE INFOCOM*, volume 3, 23-27, pages 1773–1782, New York City, USA, 2002.

[4] K. Jeffay, D.F. Stanat, and C.U. Martel. On non-preemptive scheduling of periodic and sporadic tasks. In *Proc. of the IEEE Real-Time Systems Symposium*, pages 129–139, 1991.

[5] J. Jonsson, H. Lonn, and K.G. Shin. Non-preemptive scheduling of real-time threads on multi-level-context architectures. In *Proceedings of the IEEE Workshop on Parallel and Distributed Real-Time Systems*, volume 1586, pages 363–374. Springer Verlag, 1999.

[6] J.W.S Liu. *Real - Time Systems*. Prentice Hall Inc., 2000.

[7] J. Mao and C.G. Cassandras. On-line optimal control of a class of discrete event systems with real-time constraints. *Technical Report, CODES, Boston University*, 2008. See also ftp://dacta.bu.edu:2491/TechReport/2008OLCTDA.pdf.

[8] J. Mao and C.G. Cassandras. Optimal admission control of discrete event systems with real-time constraints. *Journal of Discrete Event Dynamic Systems*, 2008. Accepted.

[9] J. Mao, C.G. Cassandras, and Q.C. Zhao. Optimal dynamic voltage scaling in power-limited systems with real-time constraints. *IEEE Trans. on Mobile Computing*, 6(6):678–688, June 2007.

[10] L. Miao and C. G. Cassandras. Optimal transmission scheduling for energy-efficient wireless networks. In *Proc. of INFOCOM*, 2006.

[11] L. Miao and C.G. Cassandras. Receding horizon control for a class of discrete event system with real-time constraints. In *Proc. of the 44th IEEE Conf. Decision and Control*, pages 7714 – 7719, 2005.

[12] D.L. Pepyne and C.G. Cassandras. Optimal control of hybrid systems in manufacturing. *Proceedings of the IEEE*, 88(7):1108–1123, 2000.

[13] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced cpu energy. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 374–382, 1995.