2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Dynamical Queue-based Task Management Policies for Human Operators

Ketan Savla Emilio Frazzoli

Abstract— **Formal methods for task management for human operators are gathering increasing attention to improve efficiency of human-in-the-loop systems. In this paper, we consider a dynamical queue approach to task management for human operators. We consider the model of dynamical queue proposed in our earlier work [1], in which the service time depends on the server utilization history. The focus of the paper is to characterize the throughput of the dynamical queue and design corresponding maximally stabilizing task release control policies, assuming deterministic arrivals. We focus extensively on threshold policies that release a task to the server only when the server state is less than a certain threshold. When every task brings in the same deterministic amount of work, we give an exact characterization of the throughput and show that an appropriate threshold policy is maximally stabilizing. When the amount of work associated with the tasks is an i.i.d. random variable with finite support, we show that the maximum throughput increases in comparison to the case where the tasks have deterministic amount of work.**

I. Introduction

Recent years have witnessed great technological advancements in automation, which in turn have marginalized the role of humans in many engineering applications. Nevertheless, the role of humans for critical tasks remains indispensable. With scientific and technological advances in modeling human performance, there has been an increasing interest in formal methods for task management for human operators to increase the overall efficiency of human-in-the-loop systems.

In this paper, we consider applications where human operators have to persistently perform similar tasks, generated over time by some arrival process. Typical examples for such settings include remotely located human operators processing continuous stream of information from unmanned vehicles in a persistent surveillance mission, e.g., see [2], or workers processing jobs in a production line. We consider a queueing framework for such settings. Queueing theory is a framework to study systems with waiting lines, and it is used to model several scenarios in commerce, industry, health-care, public service and engineering domains. An extensive treatment of queueing systems can be found in several texts, e.g., see [3], [4].

Queueing methods for task management in the context of call centers and job floors have attracted a great deal of attention, e.g., see [5]. A typical feature of such *static* queue models is that, as long as the tasks are independent of each other, the performance of the operator on the tasks are also independent. However, it is reasonable to expect that, even if the tasks are independent of each other, the performance of the server on those tasks could be correlated. For example, the cumulative performance of a human operator on two tasks serviced back to back would be different than the case when the same tasks are assigned to the operator with a break in between.

In this paper, we consider the dynamical queue model first proposed in [1], in which service times depend on the utilization history of the server. In other words, we consider the server as a dynamical system, and model the service time as a function of its state. Given this model, we consider the case in which new tasks arrive at a deterministic rate, and propose a task release control architecture that schedules the beginning of service of each task after its arrival. The model for state-dependent service times is inspired by a well known empirical law from psychology—the Yerkes-Dodson law [6]—which states that human performance increases with mental arousal up to a point and decreases thereafter. Our model in this paper is in the same spirit as the one in [7], [8], where the authors consider a state-dependent queueing system whose service rate is first increasing and then decreasing as a function of the amount of outstanding work. However, our model differs in the sense that the service times are related to the utilization history rather than the outstanding amount of work. A similar model has also been reported in the human factors literature, e.g., see [9]. Recently, there has also been interest in incorporating error rates into the performance metric for humans in a queueing setup, e.g., see [10], [11].

The control architecture considered in this paper falls under the category of *task release control*, which has been typically used in production planning to control the release of jobs to a production system in order to deal with machine failures, input fluctuations and variations in operator workload (see, e.g., [12], [13]). The task release control architecture is different from an *admission control* architecture, e.g., see [14], [8], where the objective is, given a measure of the *quality of service* to be optimized, to determine criteria on the basis of which to accept or reject incoming tasks. In the setting of this paper, no task is dropped and the task release controller simply acts like a switch regulating access to the server and hence effectively determines the schedule for the beginning of service of each task after its arrival. We extensively focus on threshold based task release control policies that release task to the operator only if the server state is below a certain fixed value.

While this paper discusses the use of dynamical queues and task release control policies for human-in-the-loop systems, such a framework is finding increasing application in

K. Savla is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 (ksavla@mit.edu).

E. Frazzoli is with the Laboratory for Information and Decision Systems, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 (frazzoli@mit.edu).

a variety of other domains, where the queue parameters are strongly dependent on some state. Examples include ramp metering congestion control of motorways, e.g., see [15], and air traffic control of national airspace systems, e.g., see [16].

The contributions of the paper are as follows. First, we adapt earlier results from [1] to exactly characterize the throughput of the dynamical queue for the special case when all the tasks are homogeneous and show that the task release control policy that releases tasks to the server only when the server state is below an appropriately chosen threshold value is maximally stabilizing. Second, for the heterogeneous task case, we provide bounds on the throughput of the dynamical queue, where we prove a surprising result that the throughput of the queue strictly increases with the introduction of heterogeneity.

II. Dynamical Queue Model

Consider the following single-server queue model. Tasks arrive periodically, at rate λ, i.e., a new task arrives every $1/\lambda$ time units. The tasks are identical and independent of each other and each task brings w units of work, where w is an i.i.d. random variable whose probability distribution is f_W with bounded support $[\mathcal{W}_1, \mathcal{W}_2]$ for some $\mathcal{W}_1 > 0$ and $\mathcal{W}_2 \geq \mathcal{W}_1$. In the rest of the paper, we shall assume this bounded support assumption on f_W without explicitly repeating it. Let $\bar{w}$ be the mean of w with respect to f_W. Let $\delta_{\bar{w}}$ be the Dirac delta distribution centered at $\bar{w}$. We shall use the δ distribution to denote the case with homogeneous tasks. Note that we assume that the task arrival process is deterministic. We briefly discuss the implications of stochastic inter-arrival times in Section V. The tasks need to be serviced in the order of their arrival. We next state the dynamical model for the server, which specifies the state-dependent rate of performing work by the server.

A. Server Model

Let $x(t)$ be the server state at time t, and let $b : \mathbb{R} \to \{0, 1\}$ be such that $b(t)$ is 1 if the server is busy at time t, and 0 otherwise. The evolution of $x(t)$ is governed by a simple first-order model:

$$\dot{x}(t) = \frac{b(t) - x(t)}{\tau}, \qquad x(0) = x_0, \tag{1}$$

where τ is a time constant that determines the extent to which past utilization affects the current state of the server, and $x_0 \in [0, 1]$ is the initial condition. Equation (1) is closely related to the moving window average model with time window τ. In particular, the two models coincide in the limit as $\tau \to \infty$. For other models of human mental workload, we refer the reader to [17].

The service times are related to the state $x(t)$ through a map $\mathcal{S} : [0, 1] \to \mathbb{R}_{>0}$. If a task is allocated to the server at state x, then the amount of time required to perform unit work is given by $\mathcal{S}(x)$. Therefore, if the amount of work associated with a task allocated to the server at state x is w, then the service time on that task is $w\mathcal{S}(x)$. Since the controller cannot interfere the server while it is servicing a task, the only way in which it can control the server state is by scheduling the beginning of service of tasks after their arrival. Such controllers are called task-release controllers and will be formally characterized later on. In this paper we assume that: $\mathcal{S}(x)$ is positive valued, continuous and convex. Let $\mathcal{S}_{\min} := \min\{\mathcal{S}(x) \mid x \in [0, 1]\}$, and $\mathcal{S}_{\max} := \max\{\mathcal{S}(0), \mathcal{S}(1)\}$.

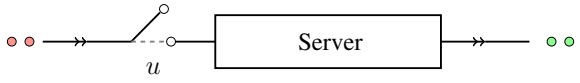

Fig. 1. Task release control architecture

A loose experimental justification of this server model in the context of humans-in-loop systems is included in our earlier work [18], where $\mathcal{S}(x)$ for that setup was found to have a U-shaped profile. We shall use that particular $\mathcal{S}(x)$ from [18] for various numerical illustrations in this paper. Further experimental evidence for this model are presented in [19].

B. Task Release Control Policy

We now describe task release control policies for the dynamical queue. Without explicitly specifying its domain, a task release controller u acts like an on-off switch at the entrance of the queue, e.g., see Figure 1. In short, u is a task release control policy if $u(t) \in \{\text{ON}, \text{OFF}\}$ for all $t \geq 0$, and an outstanding task is assigned to the server if and only if the server is idle, i.e., when it is not servicing a task, *and* when $u = \text{ON}$. Let $\mathcal{U}$ be the set of all such task release control policies. Note that we allow $\mathcal{U}$ to be quite general in the sense that it includes control policies that are functions of λ, $\mathcal{S}$, $x(t)$, f_W, τ, etc.

C. Objectives of the paper

We now formally state the problem. For a given $\tau > 0$ and f_W, let $n_u(t, \tau, \lambda, f_W, x_0, n_0)$ be the queue length, i.e., the number of outstanding tasks, at time t, under task release control policy $u \in \mathcal{U}$, when the task arrival rate is λ and the server state and the queue length at time $t = 0$ are x_0 and n_0 respectively. Define the maximum stabilizable arrival rate for policy u as:

$$\lambda_{\max}(\tau, f_W, u) = \sup\{\lambda | \limsup_{t \to +\infty} n_u(t, \tau, \lambda, f_W, x_0, n_0) < +\infty \quad \forall x_0 \in [0, 1],\, n_0 \in \mathbb{N} \quad \text{a.s.}\}.$$

The quantity $\lambda_{\max}(\tau, f_W, u)$ will also be referred to as the throughout under policy u. The maximum stabilizable arrival rate over all policies, or the throughput, is defined as $\lambda^*_{\max}(\tau, f_W) = \sup_{u \in \mathcal{U}} \lambda_{\max}(\tau, f_W, u)$. For a given $\tau > 0$ and f_W, a task release control policy u is called *maximally stabilizing* if, for any $x_0 \in [0, 1]$, $n_0 \in \mathbb{N}$, $\limsup_{t \to +\infty} n_u(t, \tau, \lambda, f_W, x_0, n_0) < +\infty$ for all $\lambda \leq \lambda^*_{\max}(\tau, f_W)$ almost surely. The primary objective in this paper is to compute the throughput and design a corresponding maximally stabilizing task release control policy for the dynamical queue whose server state evolves according to

Equation (1), and where $\mathcal{S}(x)$ is positive, continuous and convex.

In this paper, we extensively focus on a specific class of task release control policies – threshold policies. For a given $x^* \in [0,1]$, the x^*-threshold policy is defined as

$$u_{x^*}(t) = \begin{cases} \text{ON} & \text{if } x(t) \le x^*, \\ \text{OFF} & \text{otherwise.} \end{cases}$$

We prove that an appropriate threshold policy is maximally stabilizing when the tasks are homogeneous and utilize them to prove bounds on the throughput when the tasks are heterogeneous.

D. Simple bounds on the throughput

We start by deriving simple bounds on the throughput.

Proposition 2.1: For any $\tau > 0$ and f_W, we have that $\lambda^*_{\max}(\tau, f_W) \in \left[(\bar{w}\mathcal{S}(1))^{-1}, (\bar{w}\mathcal{S}_{\min})^{-1}\right]$.

Proof: The time between the start of service of successive tasks consists of two parts: the time to actively service a task, and the time when the server is idle, as governed by the task release control policy. The upper bound on the throughput is obtained by neglecting the idle times and by assuming that the server gives optimal performance for every task. The lower bound is proven by considering the trivial policy $u(t) \equiv$ ON as follows. Assume, by contradiction, that the queue length grows unbounded under this policy for some initial condition for an arrival rate $(\bar{w}\mathcal{S}(1))^{-1} - \epsilon$ for some $\epsilon > 0$. For a queue length growing unbounded, the server state eventually exceeds $1 - \eta$ for any given $\eta > 0$. After this time, all the service times per unit work are upper bounded by $\mathcal{S}(1) + \theta$ where θ depends on η through continuity of $\mathcal{S}(x)$. One can select η and hence θ such that $(\bar{w}\mathcal{S}(1) + \bar{w}\theta)^{-1} > (\bar{w}\mathcal{S}(1))^{-1} - \epsilon$, i.e., the average service time is less than the average inter-arrival time. Therefore, with probability one, the queue length will not grow unbounded. ∎

The bounds obtained in Proposition 2.1 can be shown to be tight in some simple cases. Consider first the case when $\mathcal{S} \equiv c$ for some constant $c > 0$. In this case, $\mathcal{S}(1) = \mathcal{S}_{\min} = c$ and hence Proposition 2.1 implies that $\lambda^*_{\max}(\tau, f_W) = (\bar{w}c)^{-1}$ for all $\tau > 0$. Additionally, the trivial policy $u(t) \equiv$ ON is maximally stabilizing. Another simple case is when $\mathcal{S}(x)$ is non-increasing. In this case, $\mathcal{S}(1) = \mathcal{S}_{\min}$ and hence Proposition 2.1 implies that $\lambda^*_{\max}(\tau, f_W) = (\bar{w}\mathcal{S}(1))^{-1}$ for all $\tau > 0$. One can show that the trivial policy $u(t) \equiv$ ON is maximally stabilizing in this case too.

We now derive tighter bounds on the throughput and design corresponding maximally stabilizing task release control policies.

III. Homogeneous Tasks

In this section, we consider the special case when the arriving tasks are homogeneous, i.e., every task brings in exactly the same amount of work with it. Formally, we let $f_W(w) = \delta_{\bar{w}}(w)$ for some $\bar{w} \in [\mathcal{W}_1, \mathcal{W}_2]$. We start by studying a specific class of equilibria that are associated with the trivial policy $u(t) \equiv$ ON. We only outline the key ideas here; the details can be found in [1].

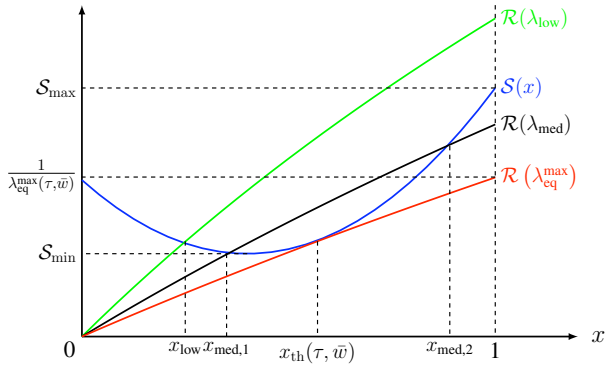

Fig. 2. A typical $\mathcal{S}(x)$ along with $\mathcal{R}(x,\tau,\bar{w},\lambda)$ for three values of λ: λ_{low}, λ_{med} and $\lambda^{\max}_{\text{eq}}(\tau,\bar{w})$ in the increasing order. Here, $x_{\text{low}} = x_{\text{eq},1}(\tau,\bar{w},\lambda_{\text{low}})$, $x_{\text{med},1} = x_{\text{eq},1}(\tau,\bar{w},\lambda_{\text{med}})$, $x_{\text{med},2} = x_{\text{eq},2}(\tau,\bar{w},\lambda_{\text{med}})$ and $x_{\text{th}}(\tau,\bar{w}) = x_{\text{eq},1}(\tau,\bar{w},\lambda^{\max}_{\text{eq}}(\tau,\bar{w}))$. Note that, since $x_{\text{th}}(\tau,\bar{w}) < 1$, $\lambda^{\max}_{\text{eq}}(\tau,\bar{w})$ is the value of λ at which $\mathcal{R}(x,\tau,\bar{w},\lambda)$ is tangent to $\mathcal{S}(x)$.

A. One-task equilibria

Define a function $\mathcal{R}$ as:

$$\mathcal{R}(x,\tau,\bar{w},\lambda) := \frac{\tau}{\bar{w}} \log\left(1 - (1 - e^{\frac{1}{\lambda\tau}})x\right). \tag{2}$$

For a given $\tau > 0$ and $\lambda > 0$, define the set of one-task equilibrium server states as:

$$x_{\text{eq}}(\tau,\bar{w},\lambda) := \{x \in [0,1] \mid \mathcal{S}(x) = \mathcal{R}(x,\tau,\bar{w},\lambda)\}. \tag{3}$$

The strict convexity of $\mathcal{S}(x) - \mathcal{R}(x,\tau,\bar{w},\lambda)$ implies that the cardinality of $x_{\text{eq}}(\tau,\bar{w},\lambda)$ can take on values 0,1 and 2. For a given $\tau > 0$, $\bar{w} > 0$ and $\lambda > 0$, let $x_{\text{eq},1}(\tau,\bar{w},\lambda)$ be the smaller element of $x_{\text{eq}}(\tau,\bar{w},\lambda)$ if it is not empty and let $x_{\text{eq},2}(\tau,\bar{w},\lambda)$ be the other element if the cardinality of $x_{\text{eq}}(\tau,\bar{w},\lambda)$ is 2. Figure 2 illustrates these definitions through an example. One can show that $x_{\text{eq},1}(\tau,\bar{w})$ is a stable equilibrium point and $x_{\text{eq},2}(\tau,\bar{w})$, if it exists, is an unstable equilibrium point. Formally, one can show that

(i) For any $\tau > 0$ and $\bar{w} > 0$, the set $(x_{\text{eq},2}(\tau,\bar{w}), 1]$ is invariant and is not in the region of attraction of $x_{\text{eq},1}(\tau,\bar{w})$ or $x_{\text{eq},2}(\tau,\bar{w})$,

(ii) There exists a $\tau^* > 0$ such that for all $\tau > \tau^*$, the set $[0, x_{\text{eq},2}(\tau,\bar{w}))$ is invariant for all $\tau > \tau^*$. Moreover, in the limit as $\tau \to +\infty$, the set $[0, x_{\text{eq},2}(\tau,\bar{w}))$ is the region of attraction of $x_{\text{eq},1}(\tau,\bar{w})$.

We introduce a couple of more definitions. For a given $\tau > 0$ and $\bar{w} > 0$, let

$$\begin{aligned} \lambda^{\max}_{\text{eq}}(\tau,\bar{w}) &:= \max\{\lambda > 0 \mid x_{\text{eq}}(\tau,\bar{w},\lambda) \neq \emptyset\}, \\ x_{\text{th}}(\tau,\bar{w}) &:= x_{\text{eq},1}\left(\tau,\bar{w},\lambda^{\max}_{\text{eq}}(\tau,\bar{w})\right). \end{aligned} \tag{4}$$

In the rest of the paper, we will restrict our attention on those τ, $\bar{w} > 0$ and $\mathcal{S}(x)$ for which $x_{\text{th}}(\tau,\bar{w}) < 1$. Loosely speaking, this is satisfied when $\mathcal{S}(x)$ is increasing on some interval in $[0,1]$ and the increasing part is *steep enough* (e.g., see Figure 2). It is reasonable to expect this assumption to be satisfied in the context of human operators whose performance deteriorates quickly at very high utilizations.

The implications of the case when $x_{\text{th}}(\tau,\bar{w}) = 1$ are discussed briefly at appropriate places in the paper.

B. Lower bound on the throughput

We start by analyzing the throughput under a specific task release control policy. In particular, we consider the $x_{\text{th}}(\tau,\bar{w})$-threshold policy, where $x_{\text{th}}(\tau,\bar{w})$ is as defined in Equation (4).

Theorem 3.1: For any $\tau > 0$, $\bar{w} > 0$, $x_0 \in [0,1]$, $n_0 \in \mathbb{N}$ and $\lambda \leq \lambda_{\text{eq}}^{\max}(\tau,\bar{w})$, if $x_{\text{th}}(\tau,\bar{w}) < 1$ then we have that $\limsup_{t\to+\infty} n_u(t,\tau,\lambda,\delta_{\bar{w}},x_0,n_0) < +\infty$ under the $x_{\text{th}}(\tau,\bar{w})$-threshold policy.

The proof of this result, which can be found in [1], follows along the lines of the proof of Theorem 4.1, where we analyze threshold policies for the heterogeneous task case.

C. Upper bound on the throughput

We now prove that the $x_{\text{th}}(\tau,\bar{w})$-threshold policy is indeed maximally stabilizing by showing that no other task release control policy gives more throughput.

Theorem 3.2: For any $\tau > 0$, $\bar{w} > 0$, $x_0 \in [0,1]$, $n_0 \in \mathbb{N}$, $\lambda > \lambda_{\text{eq}}^{\max}(\tau,\bar{w})$ and $u \in \mathcal{U}$, if $x_{\text{th}}(\tau,\bar{w}) < 1$ then we have that $\limsup_{t\to+\infty} n_u(t,\tau,\lambda,\delta_{\bar{w}},x_0,n_0) = +\infty$.

The proof of Theorem 3.2 is a simple adaptation of a similar result in [1].

Theorems 3.1 and 3.2 imply that the throughput of the dynamical queue is $\lambda_{\text{eq}}^{\max}(\tau,\bar{w})$, and that the $x_{\text{th}}(\tau,\bar{w})$-threshold policy is maximally stabilizing.

Remark 3.3: (i) For a given $\bar{w}$, it is interesting to note the dependence of $\lambda_{\text{eq}}^{\max}$ on τ. For any $\bar{w} > 0$, one can show that $\lambda_{\text{eq}}^{\max}(\tau,\bar{w})$ is monotonically strictly decreasing in τ. Additionally, $\lim_{\tau\to 0^+} \lambda_{\text{eq}}^{\max}(\tau,\bar{w}) = (\bar{w}\mathcal{S}_{\min})^{-1}$, and $\lim_{\tau\to+\infty} \lambda_{\text{eq}}^{\max}(\tau,\bar{w}) = a$, where $a > 0$ is such that the line passing through the origin and having slope $\bar{w}/a$ is tangential to $\mathcal{S}$ in $(0,1)$. An example plot of $\lambda_{\text{eq}}^{\max}(\tau,\bar{w})$ is shown in Figure 3.

(ii) If $x_{\text{th}}(\tau,\bar{w}) = 1$, then one can show that, for any $\epsilon > 0$, there exists no stabilizing task release control policy for arrival rates greater than $\lambda_{\text{eq}}^{\max}(\tau,\bar{w}) + \epsilon$, i.e., $\lambda_{\max}^{*}(\tau,\bar{w}) \leq \lambda_{\text{eq}}^{\max}(\tau,\bar{w}) + \epsilon$.

(iii) For any $\lambda < \lambda_{\text{eq}}^{\max}(\tau,\bar{w})$, Theorem 3.1 holds true for any x-threshold policy with $x \in [x_{\text{eq},1}(\tau,\bar{w}), x_{\text{eq},2}(\tau,\bar{w})]$, if $x_{\text{eq},2}(\tau,\bar{w})$ exists or $x \in [x_{\text{eq},1}(\tau,\bar{w}), 1]$ otherwise. It is possible to exploit this flexibility to design a threshold policy with dynamically changing threshold values to ensure that, for any $n_0 \in \mathbb{N}$ and $x_0 \in [0,1]$, the queue length goes to zero in finite time for any $\lambda < \lambda_{\text{eq}}^{\max}(\tau,\bar{w})$.

IV. Heterogeneous Tasks

In this section, we return to the general case when f_W is not necessarily the delta distribution. For this general case, we are not able to compute the throughput exactly but we provide meaningful bounds.

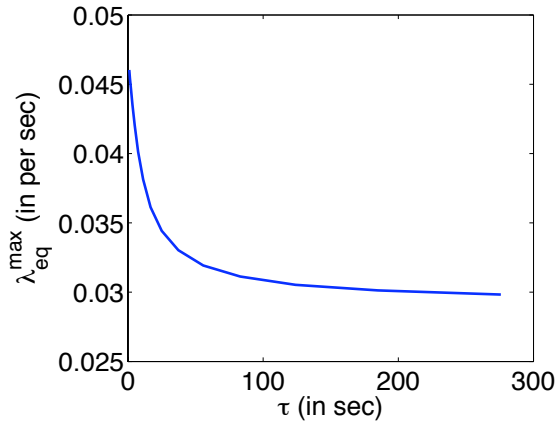

Fig. 3. Plot of $\lambda_{\text{eq}}^{\max}(\tau,\bar{w})$ versus τ for $\mathcal{S}(x) = 229x^2 - 267x + 99$ and $\bar{w} = 1$.

A. Lower bound on the throughput

We first prove a surprising lower bound.

Theorem 4.1: For any f_W and $\tau > 0$, we have that

$$\lambda_{\max}^{*}(\tau,f_W) \geq \lambda_{\max}^{*}(\tau,\delta_{\bar{w}}),$$

where the inequality is strict if and only if $f_W \neq \delta_{\bar{w}}$.

Proof: We first prove an upper bound on the inter-task time under a threshold policy, which will be critical in proving the main result. The time between servicing successive tasks under the x-threshold policy is the random variable $w\mathcal{S}(x) + \tau\log\left(\frac{1-(1-x)e^{-w\mathcal{S}(x)/\tau}}{x}\right)$. Due to the bounded support assumption of f_W, this random variable has a finite variance. The expected value can be written as

$$\begin{aligned} &E_{f_W}\left[w\mathcal{S}(x) + \tau\log\left(\frac{1-(1-x)e^{-w\mathcal{S}(x)/\tau}}{x}\right)\right] \\ &= \bar{w}\mathcal{S}(x) + \tau E_{f_W}\left[\log\left(1-(1-x)e^{-w\mathcal{S}(x)/\tau}\right)\right] - \tau\log(x). \end{aligned} \tag{5}$$

The function $1-(1-x)e^{-y/\tau}$ is strictly concave in y (except for $x = 1$) and non-negative for every $\tau > 0$. Since every non-negative strictly concave function is also logarithmically strictly concave, e.g., see [20], applying Jensen's inequality to $E_{f_W}\left[\log\left(x-1+e^{w\mathcal{S}(x)/\tau}\right)\right]$ and using Equation (5), one gets that for all $f_W \neq \delta_{\bar{w}}$,

$$\begin{aligned} &E_{f_W}\left[w\mathcal{S}(x) + \tau\log\left(\frac{1-(1-x)e^{-w\mathcal{S}(x)/\tau}}{x}\right)\right] \\ &\qquad < \bar{w}\mathcal{S}(x) + \tau\log\left(\frac{1-(1-x)e^{-\bar{w}\mathcal{S}(x)/\tau}}{x}\right). \end{aligned} \tag{6}$$

We now use this bound to prove the main result. Let

$$T := \min_{x\in[0,1]} E\left[w\mathcal{S}(x) + \tau\log\left(\frac{1-(1-x)e^{-w\mathcal{S}(x)/\tau}}{x}\right)\right], \tag{7}$$

and let x^* be the minimizer. Using Equation (6) for $x = x_{\text{th}}(\tau,\bar{w})$, we get that $T < 1/\lambda_{\text{eq}}^{\max}(\tau,\bar{w}) = 1/\lambda_{\max}^{*}(\tau,\delta_{\bar{w}})$, where the equality follows from Theorems 3.2

and 3.1. Consider the evolution of the queue under the x^*-threshold policy for an arrival rate $1/T - \eta$ for any $\eta \in (0, 1/T - \lambda^*_{\max}(\tau, \delta_{\bar{w}}))$. We now prove the boundedness of the queue length, thereby proving the theorem.

Let x_i and t_i be the server state and time instants respectively at the beginning of service of the i-th task. For brevity in notation, let $n(t)$ be the queue length at time t. For any $x_0 \in [0,1]$ and $n_0 \in \mathbb{N}$, considering the possibility when $x_0 > x^*$ we have that $n(t_1) = \max\{0, n_0 - 1, n_0 - 1 + \lfloor \lambda\tau \log(x_0/x^*) \rfloor\}$. Consider the following two cases:

- **State 1:** $x_1 = x^*$. While $n(t_i) > 0$, we have that $x_{i+1} = x^*$ and $t_{i+1} - t_i = w\mathcal{S}(x^*)$, where w is independently and identically distributed according to f_W. Applying the Strong Law of Large Numbers to the sequence $t_i - t_{i-1}$, we have that, for every $\epsilon > 0$, there exists $n(\epsilon) > 0$ such that, for all $n \geq n(\epsilon)$,

$$\Pr\Big(\Big|\frac{t_{n+1} - t_1}{n} - E_{f_W}\left[w\mathcal{S}(x^*) + \tau \log\left(\frac{x^* - 1 + e^{w\mathcal{S}(x^*)/\tau}}{x^*}\right)\right]\Big| < \epsilon\Big) = 1.$$

  This implies that, with $\epsilon := \frac{1}{2}\left(\frac{1}{\lambda} - \bar{w}\mathcal{S}(x^*)\right)$, for all $n \geq \max\left\{n(\epsilon), \frac{2n_0}{1 - \lambda\bar{w}\mathcal{S}(x^*)}\right\}$, we have that

$$t_{n+1} - t_1 < n\left(\frac{1}{\lambda} - \frac{n_0}{n\lambda}\right) = \frac{n - n_0}{\lambda}.$$

  In other words, for any initial queue length n_0, with probability one, after service of finite number of tasks the queue length goes to zero and the server state drops below x^*. Thereafter, we appeal to the next case by resetting x_i and t_i as x_1 and t_1 respectively. Moreover, with these notations, $n(t_1)$ will be zero.
- **State 2:** $x_1 < x^*$. While the queue length is non-zero, the server is never idle. The maximum amount of continuous service time required for the server state to cross x^* starting from any $x_1 < x^*$ is upper bounded by $-\tau \log(1 - x^*) + \mathcal{W}_2\mathcal{S}_{\max}$. This is possibly followed by an idle time that is upper bounded by $-\tau \log x^*$, at the end of which the server state is x^*. Therefore, the maximum number of outstanding tasks when the server state reaches x^* is upper bounded by $n_1 + \lceil \tau \log(1 - x^*)\left((\mathcal{W}_2\mathcal{S}_{\max})^{-1} - \lambda\right)\rceil + \lceil -\lambda\tau \log x^* \rceil$. Thereafter, we appeal to the earlier case by resetting $x_1 = x^*$ and n_1 to be the number of outstanding tasks when the server state reaches x^*.

In summary, when the system is in State 1, the queue length decreases to zero with probability one at which point it enters State 2. When the system is in State 2, it stays in it for ever or eventually enters State 1 with bounded queue length. Collecting these facts, we arrive at the result. ∎

Remark 4.2: (i) Theorem 4.1 shows that, the throughput strictly increases with the introduction of stochasticity in service times. This is novel because not only does this imply that the throughput of the dynamical queue depends also on the second moment of the service times, but also that this dependence occurs in an unexpected way. To the best of our knowledge, such a phenomenon has not been reported for queueing systems so far.

(ii) The concavity property that leads to Equation (6) is associated with the server dynamics and is independent of the convexity of $\mathcal{S}(x)$, and hence Equation (6) is valid for any $\mathcal{S}(x)$ that does not necessarily satisfy the convexity property assumed in this paper.

B. Upper bound on the throughput

In this section, we derive an upper bound on the maximum throughput possible using any task release control policy.

Theorem 4.3: For any f_W and $\tau > 0$, we have that

$$\lambda^*_{\max}(\tau, f_W) \leq \left(E_{f_W}\left[1/\lambda^{\max}_{\text{eq}}(\tau, w)\right]\right)^{-1}.$$

Proof: The proof is similar to that of Theorem 3.2. The difference is in the time required for the n-task static problem. Consider a set of n tasks associated with works $w_1, \ldots, w_n$, where each w_i is identically and independently sampled from f_W. It is desired to service these n tasks in the fastest possible way using a task release control policy under the constraint that the initial and final server state is x. The time required for the 1-task problem is $w_1\mathcal{S}(x) + \tau \log\left(\frac{x - 1 + e^{w_1\mathcal{S}(x)/\tau}}{x}\right)$ which is lower bounded by $1/\lambda^{\max}_{\text{eq}}(\tau, w_1)$. Using the same decomposition and rearrangement approach as before, the time required for the n-task problem can be lower bounded by $\sum_{i=1}^{n} 1/\lambda^{\max}_{\text{eq}}(\tau, w_i)$. Using Strong law of large numbers, one can show that, with probability one, as $n \to +\infty$, the average time required per task is lower bounded by $E_{f_W}\left[1/\lambda^{\max}_{\text{eq}}(\tau, w)\right]$. The rest of the proof follows similarly. ∎

Theorems 4.1 and 4.3 do not imply that a threshold policy is maximally stabilizing when the tasks are heterogeneous. However, the proof of Theorem 4.1 implies that best threshold policy would correspond to the x^*-threshold policy, where x^* is the minimizer in Equation (7).

C. Simulations

For $\tau = 300\,s$, f_W a uniform distribution over $[5, 45]$ and $\mathcal{S}(x) = (229x^2 - 267x + 99)/25\,s$, the lower bound, as given by Theorem 4.1 is computed to be about $0.031\,s^{-1}$ and the upper bound, as given by Theorem 4.3 is computed to be about $0.039\,s^{-1}$. Note that, these bounds are tighter than the bounds provided by Proposition 2.1.

Theorem 4.1 suggests that, for appropriate f_W, one could possibly increase $\lambda^{\max}_{\text{eq}}(\tau, f_W)$ up to $(\bar{w}\mathcal{S}_{\min})^{-1}$ for all $\tau > 0$ and, hence, by Proposition 2.1, one could achieve the maximum possible throughput. Figure 4 demonstrates that this is feasible through an illustrative extreme example. The solid curve in Figure 4, which represents the throughput curve, shows that for tasks with large heterogeneity, the throughput for a given threshold value x closely follows the inverse of $\bar{w}\mathcal{S}(x)$ which itself is the maximum possible throughput under the x-threshold policy. In particular, the throughput under the $\arg\min_{x \in [0,1]} \mathcal{S}(x)$-threshold policy is very close to $(\bar{w}\mathcal{S}_{\min})^{-1}$.

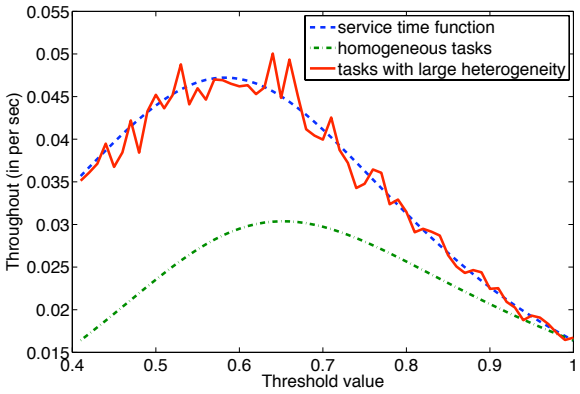

Fig. 4. Illustration of maximum possible throughput under extreme task heterogeneity. The solid curve represents the throughput curve under threshold policies when $f_W(w)$ is a binary random variable that takes values 0.01 and 5000 with probabilities 0.995 and 0.005 respectively, and $\mathcal{S}(x) = (229x^2 - 267x + 99)/25\,s$; the dash-dotted curve represents the throughput curve under threshold policies when $f_W(w) = \delta_{25}(w)$ and the same $\mathcal{S}(x)$; the dashed curve represents the inverse of the function $\bar{w}\mathcal{S}(x)$, with $\bar{w} = 25$ and the same $\mathcal{S}(x)$.

V. Conclusions

In this paper, we discussed a dynamical queue framework as a possible formal approach to task management of human operators under a task release control policy. Inspired by empirical laws, we proposed a model whereby the service times are dependent on the state of a simple underlying dynamical system. We studied the stability of such dynamical queues under deterministic task inter-arrival times. For homogeneous tasks, we proved that a task release control policy that releases a task to the server only when it's state is below an appropriately chosen threshold value gives the maximum throughput. For heterogeneous tasks, we showed that the throughput strictly increases with the introduction of heterogeneity. The deterministic task inter-arrival time assumption in our analysis is not binding and the results extend when the inter-arrival times are sampled identically and independently sampled from a common distribution with bounded variance.

The ability of appropriately designed threshold based task control policies to stabilize an otherwise unstable queue, and the associated throughput optimality results proven in this paper, provide a formal methodology and justification for similar approaches commonly adopted in practice, e.g., see [13]. From a scientific point of view, the increase in throughput due to heterogeneity in tasks is a novel phenomenon for queueing systems. This result provides an additional dimension to improving throughput of dynamical systems by repackaging the tasks until one has maximum heterogeneity across the repackaged tasks. Moreover, if one has to decide upon a quantization technique for a work-intensive task to be completely by a human operator as quickly as possible, this result suggests that uniform quantization is the worst possible quantization.

In future, we plan to extend our analysis to characterize the average wait time of dynamical queues. We also plan to extend our formulation and analysis to incorporate accuracy of the job done by the operators in the performance metric. We intend to perform extensive experiments to develop a high fidelity dynamical model for human operators. Finally, we also plan to extend our framework to align it more closely to conventional state-dependent queues where the notion of server state is closely related to the amount of outstanding work rather than the past utilization.

Acknowledgments

This research was partially supported by the Michigan/AFRL Collaborative Center on Control Science, AFOSR grant no. FA 8650-07-2-3744.

References

[1] K. Savla and E. Frazzoli, "Maximally stabilizing admission control policy for a dynamical queue," *IEEE Trans. on Automatic Control*, vol. 55, no. 11, pp. 2655–2660, 2010. Available at `http://arxiv.org/abs/0909.3651`.

[2] K. Savla, T. Temple, and E. Frazzoli, "Human-in-the-loop vehicle routing policies for dynamic environments," in *IEEE Conf. on Decision and Control*, pp. 1145–1150, 2008.

[3] L. Kleinrock, *Queueing Systems I: Theory*. Wiley-Interscience, 1975.

[4] S. Asmussen, *Applied Probability and Queues*. Springer, 2003.

[5] G. Koole and A. Mandelbaum, "Queueing models of call centers: An introduction," *Annals of Operations Research*, vol. 113, pp. 41–59, 2002.

[6] R. M. Yerkes and J. D. Dodson, "The relation of strength of stimulus to rapidity of habit-formation," *Journal of Comparative Neurology and Psychology*, vol. 18, pp. 459–482, 1908.

[7] J. H. Dshalalow, ed., *Frontiers in Queuing Models and Applications in Science and Engineering*, ch. Queueing Systems with State Dependent Parameters. CRC press, Inc., 1997.

[8] R. Bekker and S. C. Borst, "Optimal admission control in queues with workload-dependent service rates," *Probability in the Engineering and Informational Sciences*, vol. 20, pp. 543–570, 2006.

[9] M. L. Cummings and C. E. Nehme, "Modeling the impact of workload in network centric supervisory control settings," in *2nd Annual Sustaining Performance Under Stress Symposium*, (College Park, MD), Feb. 2009.

[10] L. F. Bertuccelli, N. Pellegrino, and M. Cummings, "Choice tasks in modeling relooks in UAV search missions," in *American Control Conference*, (Baltimore, MD), pp. 2410–2415, 2010.

[11] V. Srivastava, R. Carli, F. Bullo, and C. Langbort, "Task release control for decision making queues," in *American Control Conference*, (San Francisco, CA), 2011. To appear.

[12] C. R. Glassey and M. G. C. Resende, "A scheduling rule for job release in semiconductor fabrication," *Operations Research Letters*, vol. 7, no. 5, pp. 213–217, 1988.

[13] J. W. M. Bertrand and H. P. G. V. Ooijen, "Workload based order release and productivity: a missing link," *Production Planning and Control*, vol. 13, no. 7, pp. 665–678, 2002.

[14] S. Stidham, "Optimal control of admission to queueing system," *IEEE Trans. Automatic Control*, vol. 30, pp. 705–713, Aug 1985.

[15] F. P. Kelly and R. J. Williams, "Heavy traffic on a controlled motorway," in *Probability and Mathematical Genetics: Papers in Honour of Sir John Kingman* (N. H. Bingham and C. M. Goldie, eds.), no. 378 in London Mathematical Society Lecture Notes Series, Cambridge University Press, 2010.

[16] J. Le Ny and H. Balakrishnan, "Distributed feedback control for an eulerian model of the national airspace system," in *Proceedings of the American Control Conference*, (St. Louis, MO), pp. 2891–2987, 2009.

[17] P. A. Hancock and N. Meshkati, eds., *Human mental workload*. No. 52 in Advances in Psychology, Elsevier Science Publishers B. V., 1988.

[18] K. Savla, C. Nehme, T. Temple, and E. Frazzoli, "Efficient routing of multiple vehicles for human-supervised services in a dynamic environment," in *AIAA Conf. on Guidance, Navigation, and Control*, (Honolulu, HI), 2008.

[19] K. Savla and E. Frazzoli, "A dynamical queue approach to intelligent task management for human operators," *Proceedings of the IEEE*, 2011. To appear.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.