Quantitative Modeling and Analytical Calculation of Anelasticity for a Cyber-Physical System
Hongfang Gong, Renfa Li, Senior Member, IEEE, Jiyao An, Member, IEEE, Yang Bai, and Keqin Li, Fellow, IEEE

Abstract—This paper investigates resource provisioning in cyber-physical systems (CPSs) by developing a new definition of anelasticity. A flat semi-dormant multicontroller (FSDMC) model is established on a special type of CPS platform named arbitrated networked control system with dual communication channels. A novel, quantitative, and formal definition of anelasticity for the FSDMC is proposed. A new finite capacity $M/M/c$ queuing system with $N$-policy and asynchronous multiple working vacations of partial servers is established, and the FSDMC is modeled as a quasi-birth-and-death process to obtain the stationary probability distribution of the system. Based on the queueing model, we quantify various performance indices of the system to build a nonlinear cost-performance ratio (CPR) function. An optimization model is presented to minimize the CPR. A particle swarm optimization (PSO) algorithm is used to find the optimum solution of the optimization model and obtain the optimal configuration values of the system parameters under stability condition. By changing the system parameters, the sensitivity of the system performance indices and the CPR are analyzed, respectively. The unexpected workload varies randomly over time. Thus, an $M/M/1/K$ queue is constructed in a Markovian environment by employing a three-state, irreducible Markov process. In this queue, the conditional average queue length and the probabilities of the three-state process are calculated. Then, the anelasticity value of the system is precisely determined. When the average arrival rate exceeds the average service rate in the queueing system, an optimal CPR unchanged adaptive algorithm based on PSO is designed to dynamically adjust the controller service rate. Extensive numerical results show the usefulness and effectiveness of the proposed techniques and exhibit that the system can maintain elastic invariance in adaptive adjustment parameters.

Index Terms—Anelasticity, cost-performance ratio (CPR), cyber-physical system (CPS), queuing system, semi-dormant controller cluster.

I. INTRODUCTION

A. Background

UTOMOTIVE, avionics, and industrial automation systems are often distributed embedded systems (DESs), such as CAN, LIN, and FlexRay, with a large number of processing units (PUs) that communicate via shared buses. Such architectures are used to run a certain number of distributed control applications under multiple quality-of-control constraints [1]. In such a complex cyber-physical system (CPS), resource constraints, cost sensitivity, and time sensitivity are its important features [2], [3]. The CPU and network resources must be allocated to competing applications subject to the constraints on resources imposed by physical phenomena [4]. To achieve the minimum cost and optimal performance of the systems, the limited computational resources must be used as efficiently as possible. Hence, multiple distributed control applications in such systems require dynamic provisioning of resources based on application demand. An embedded computing platform relies on the mechanisms and capabilities for allocating/deallocating computing resources on demand to acquire a large number of PUs for handling workload surges or releasing PUs to avoid over-provisioning of resources. Such a dynamic resource provision and management feature is called elasticity [5]. Dustdar et al. [6] introduced the principle of elastic processes with cost, quality, and resources as the basic elasticity dimensions, thereby forming the foundations of elastic systems.

The problem of resource provisioning in CPSs has drawn research attention [7]. In [8], an elastic CPS (eCPS) was defined as an adaptive system that could further add/remove components at run-time, from computing resources to physical devices, to align their costs, quality, and resource usage to load and owner requirements. Moldovan et al. [9] introduced...
elasticiy to cyber-physical ecosystems to integrate computing processes, people, and things. They also outlined the vision over a new computing field, elastic computing, which pertains to the study of elastic systems. Candra et al. [10] presented a modeling tool named elasticity profile, which specifies constructs for modeling the elastic behavior of mixed systems with respect to tradeoffs among cost, quality, and resources. Schmidt et al. [3] identified challenges, opportunities, and benefits of large-scale CPSs that support cloud computing and elastic infrastructure. Palensky et al. [11] described the main challenges, tools and methods of continuous time-based and discrete event-based models of large scale cyber-physical energy systems under resource constraints. In [12], an elastic computation middle-ware was developed to federate computation resources on wearable, mobile, and connected devices in CPSs. However, these efforts neither involve the quantitative modeling of elasticity on CPS nor study the changes in the cost and performance of the system under different workloads.

In cloud computing, Wang et al. [13] proposed a computing resource allocation approach based on decentralized multiagent to address the problem of energy-aware concentration or migration resource provisioning. In [14], based on the Lyapunov optimization technique combined with the technique of weight perturbation, Fang et al. introduced a new stochastic control algorithm that dynamically allocates limited computing resources on mobile cloudlet platforms to reduce energy consumption and improve application performance. However, these efforts do not consider the problem of quantitative modeling of system elasticity and dynamic resource provisioning. Ai et al. [5] proposed an elasticity measurement model of the cloud platform by using a continuous-time Markov chain. Li [15] treated a cloud platform as a queueing system to analytically and comprehensively study elasticity, performance, and cost in cloud computing. However, this measurement model cannot be directly used in the elastic computing of DESs because of the time-varying delay caused by limited system resources.

B. Motivation

A special type of CPS named arbitrated networked control system (ANCS) was first presented in [16]. Gong et al. [17] extended the ANCS and proposed an ANCS with dual communication channels (see [17, Fig. 2]). In this novel special CPS, a flat semi-dormant multicontroller (FSDMC) model was proposed to consider load balancing and global consistency. To achieve a tradeoff between system performance and its implementation cost, the FSDMC was modeled as an $N/(d,c)$-M/M/c/K/AMWV queueing system to configure the system computing resources dynamically. However, in the FSDMC, two main problems need to be solved, namely, over-provisioning of resources and overloading of systems. In the $N/(d,c)$-M/M/c/K/SMWV system, synchronized vacations lead to over-provisioning of resources and incur unnecessary costs. In some cases, unexpected workload spikes are likely to occur due to unpredictable changes in the physical environment. In this paper, we focus on scaling resources dynamically in the FSDMC in accordance with dynamic external environment to improve resource utilization. We consider a typical distributed embedded architecture proposed in [1] (see [1, Fig. I(a)]).

In the ANCS, the main problem that can be attributed to the gap between high-level control models and their actual implementations is the end-to-end delay between the signal sensed at the plant output and the signal sent to the control actuator [18]. In an elastic system, we call the time-dependent elastic behavior an anelastic behavior and consider this time-sensitive elastic system an anelastic system. The current Wikipedia definition of anelasticity in physics states that “an anelastic material is a special case of a viscoelastic material. Viscoelastic materials have elements of both viscous and elastic characteristics and, as such, exhibit time-dependent strain. An anelastic material will fully recover to its original state on the removal of load.” When applied to computing, anelasticity naturally reflects the on-demand nature of time-related resource provisioning in DESs. Resource provisioning indicates making the resources seamlessly available to users as and when they are required [7]. In the FSDMC, when external forces (e.g., the number of packet-in message requests) are applied, the strain (e.g., increased resource utilization and average task response time) increase nonlinearly with time, that is, the FSDMC platform is deformed. A hysteresis occurs in the stress–strain due to the resource constraints of the shared bus. This hysteresis is the main difference between elastic and anelastic systems.

In this paper, we model the FSDMC as an $N/(d,c)$-M/M/c/K/AMWV queue to define auto-scaling mechanism for the resource management of FSDMC platform and obtain various performance measures to establish the nonlinear cost-performance ratio (CPR) function. When the workload increases in this queuing system, the system cost increases, whereas the system performance decreases. Thus, the strain (i.e., CPR) will change as the stress (i.e., arrival rate) increases. When the average arrival rate exceeds the average service rate, the controller service rate should be adjusted adaptively to maintain the CPR invariance of the system in a stable state. According to the match between the resource provisioning and workload, the FSDMC platform is in three states (three-state hereinafter), namely, over-provisioning, normal, and under-provisioning states [15]. We describe a Markovian environment using a three-state Markov process and construct an M/M/1/K queueing model with three phases in a Markovian environment [19] to obtain the quantitative model of anelasticity of the system under unexpected workload varying randomly over time. To our knowledge, we are the first to establish a quantitative and analytical model of anelasticity for a special CPS.

C. Our Contributions

The contributions of this paper are summarized as follows. 1) A novel, quantitative, and formal definition of anelasticity for the FSDMC on the special CPS is proposed. The quantization model is a two-tuple consisting of the average queue length in the normal state and the probability that the computing resources match the current workload.
2) We establish a finite capacity $\text{M}/\text{M}/c$ queuing system with $N$-policy and asynchronous multiple working vacations (AMWV) of partial servers, namely, $N/(d,c)-\text{M}/\text{M}/c/K/\text{AMWV}$, where $c$, $d$, and $K$ are described in [17]. This queuing system is used to quantify the various performance measures of the system, and to build an optimization model that minimizes the nonlinear CPR function. A particle swarm optimization (PSO) [20] algorithm is used to generate the optimum solution of the optimization model and obtain the optimal configuration values of the system parameters under stability condition.

3) We construct an $\text{M}/\text{M}/1/K$ queue in a Markovian environment to calculate the anelasticity value of the FSDMC platform accurately. With the aid of the equilibrium probability distribution of this queue, the conditional average queue length and the probabilities of three-state are yielded, and the anelasticity value of the system is derived. When the average arrival rate exceeds the average service rate in the $\text{N}/(d,c)-\text{M}/\text{M}/c/K/\text{AMWV}$ queuing system, we propose an optimal CPR unchanged adaptive algorithm based on PSO to adjust the controller service rate dynamically. As a result, the proposed queuing model is transformed into a classic $\text{M}/\text{M}/c/K$ queue where the anelasticity remained unchanged.

The rest of this paper is organized as follows. Section II describes the related works. In Section III, we define an anelasticity CPS and the measurement of anelasticity in CPSs. In Section IV, we propose a novel queuing model, namely, $\text{N}/(d,c)-\text{M}/\text{M}/c/K/\text{AMWV}$ queue, and compute the equilibrium distributions and the performance indices of the system by using matrix-geometric method based on a Markovian chain. In Section V, we construct an optimization model to minimize the average CPR in the system, and analyze the sensitivity of the system performance and the CPR using numerical results. In Section VI, we establish an $\text{M}/\text{M}/1/K$ queue in a Markovian environment to calculate the anelasticity value of the FSDMC platform accurately, and we discuss the problems of the system adaptive and anelastic invariance. Finally, Section VII concludes this paper.

II. RELATED WORKS

In CPSs, provisioning the required resource is extremely challenging in composing a new service because of inherent heterogeneity in the resources and thus in the services. Therefore, the issues related to service composition and thus resource provisioning in CPSs must be investigated [7]. Resource provisioning in resource management and modeling has been rarely investigated. Gunes et al. [21] presented a detailed survey on concepts, applications, and challenges in CPSs, and indicated that a highly scalable system should provide scatter and gather mechanisms for workload balancing and effective communication protocols to improve the performance.

Ravindran and Li [22] presented two proactive resource allocation algorithms, namely, deadline-driven proactive and laxity-driven proactive, which proactively allocate resources to maximize the aggregate deadline-satisfied ratio for the future time interval in asynchronous real-time distributed systems. Marti et al. [23] developed a dynamic resource allocation system for control tasks, namely, Draco, based on feedback information from the plants. Tan et al. [24] developed utility-based resource configuration algorithms to allocate the limited resources among users to satisfy their specific quality of service requirements in wireless networks. To improve the utilization of computing resources for self-triggered control applications, Samii et al. [25] proposed a software-based middleware component for scheduling and optimization of control performance and CPU usage of multiple self-triggered control loops on a uniprocessor platform. Lozoya et al. [26] presented a performance evaluation framework that permits assessment whether recent state-of-the-art resource/performance-aware policies can be implemented in practice. The application of this framework showed that online management of computing resources is the key for allowing adaption of embedded control systems to varying demands while delivering specific control performance with tunable resource utilization. Chantem et al. [27] focused on the elastic task model to select task periods for adapting periodic real-time systems in the presence of uncertainty. However, these efforts only consider resource allocation performance optimization problems but ignore cost-effectiveness issues.

Xie et al. [28] investigated the cost minimization of effective use of resources on heterogeneous embedded systems without using fault tolerance. Chang and Chakraborty [29] developed approaches in automotive control systems design that consider implementation resources to improve the control performance and reduce the system costs for a given amount of resources. Liu et al. [30] considered task scheduling problem at computing center in CPSs, and analyzed the performance indicators of the computing center which is regarded as a multipriority multiserver queuing system. A comprehensive resource scheduling strategy was proposed to balance cost and system performance. In [31], an elastic thread management scheme based on external coordinator was implemented and provided in a middleware to avoid under-utilization of server resources. This scheme suits the dynamic behavior of systems that require supporting varying numbers of clients in a cost-effective manner. In [32], the joint optimization problem with energy efficiency and effective resource utilization was investigated for heterogeneous and distributed multicore embedded systems. To address fast energy equipartition for cyber-physical network systems, Zeng et al. [33] developed three novel hybrid stabilization techniques, and established an optimization-based network topology design framework. A good review of design techniques and applications on CPSs was described in [34]. These works only focused on the system cost problem but ignored the system performance under resource utilization and the quantitative analysis in resource provisioning. Sheng et al. [35] considered workflow system performance and resource cost balancing issues and used an $\text{M}/\text{M}/c$ queuing network to model multidimensional workflow net (MWF-net) to calculate the probability density function of transaction instance.
to reach a certain maturity. Development technology of CPS clouds still needs considerable time
An overview of research efforts on the integration of CPSs
tic technologies for cloud-based real-time stream data analytics systems.
queueing networks and can handle arbitrary operator topologies
A dynamic resource scaling framework that included an accurate performance model based on the theory of Jackson open
et al. [38] focused on the workload scheduling issue in CPS and proposed an approach for solving
the limitations caused by heterogeneous cloud computing. In [39], a level value density task scheduling algorithm for
CPSs on cloud platform was proposed to schedule and distinguish the high-volume and multiclass tasks more efficiently.
The algorithm not only adds attributes of time and value to the tasks, but also describes the weighted relationship between these attributes by mathematical means. Fu et al. [40] proposed a dynamic resource scaling framework that included an accurate performance model based on the theory of Jackson open queueing networks and can handle arbitrary operator topologies for cloud-based real-time stream data analytics systems. An overview of research efforts on the integration of CPSs with cloud computing is presented in [41]. However, the development technology of CPS clouds still needs considerable time to reach a certain maturity.

III. MEASUREMENT OF ANELASTICITY

In this paper, we still consider the DES proposed in [1] (see [1, Fig. 1(a)]), where sensor task $T_s$, controller task $T_c$, and actuator task $T_a$ are performed in the respective ECUs, and the messages $m_1$ and $m_2$ are transmitted by the communication bus (e.g., FlexRay).

We consider the ANCS platform architecture with dual communication channels proposed in [17] (see [17, Fig. 2]). Multiple control applications are divided into a number of tasks that are mapped onto different PUs. All the controller nodes of the PUs are connected to the two channels of $A$ and $B$, but other nodes are only connected to the channel $A$ (see [17]). In addition to interacting with the corresponding sensor and actuator via the shared channel $A$, each controller needs to address the overload of other controllers via the channel $B$. The shared bus communication protocol is a hierarchical flexible TDMA/FP bus scheduling policy with an event-triggered protocol proposed in [17] (see [17, Fig. 5]).

According to the definition of eCPS in [8], we define an anelastic CPS as follows.

Definition 1: An anelastic CPS is an eCPS where strain (e.g., cost and performance) is delayed in stress (e.g., workload).

In [15], the elasticity of the system with dynamically variable workload is defined as the percentage of time (or probability) that the system is in the normal state. We use $p_{over}$, $p_{normal}$, and $p_{under}$ to represent the probabilities that the system is in over-provisioning, normal, and under-provisioning states, respectively, and $p_{over} + p_{normal} + p_{under} = 1$. In the $M/M/1/K$ queue in a Markovian environment, the conditional queue length densities are often utilized in the study of the time-dependent behavior of queues described by Markovian models [19]. AvgQL$_{normal}$ denotes the conditional average queue length when the $M/M/1/K$ queue in a Markovian environment is in the normal state.

Definition 2: In an anelastic CPS, the anelasticity is calculated as follows:

$$\text{Anelasticity} = \left( \text{AvgQL}_{normal} \cdot p_{normal} \right). \quad (1)$$

In the $N(d,c)-M/M/c/K/AMWV$ queue, a state is:
1) an over-provisioning state if $0 \leq k < c$;
2) a normal state if $c \leq k < N$;
3) an under-provisioning state if $N \leq k \leq K$, where $k$ represents the number of packet-in messages in the system.

In the $M/M/1/K$ queue in a Markovian environment, we use $(k, i)$ to describe a state, where $k \geq 0$ is the number of packet-in messages in the system, and $i$ takes the value from three states (or three phases, three-state or three-phase hereinafter), namely, over-provisioning, normal, and under-provisioning states; for simplicity, the three states are represented as 1, 2, and 3. The transitions among the three-state are described as follows.

1) $(k - 1, i) \xrightarrow{\lambda_i} (k, i + 1)$. This transition happens when a new packet-in message arrives in phase $i$.
2) $(k, i + 1) \xrightarrow{\mu_i} (k - 1, i)$. This transition happens when a packet-in message is completed in phase $i$, where $k = c$, $i = 1$ or $k = N$, $i = 2$; and $\lambda_i$ and $\mu_i$ represent the arrival and service rates when the system is in phase $i$ (or state $i$), respectively.

Under stability condition (see Section V-A), when the number of packet-in messages in the system is larger than or equal to the threshold $N$ and the current state is under-provisioning, the $d$ semi-dormant controllers return from the working vacations and proceed to serve the waiting packet-in messages. When the number of packet-in messages in the system is less than the number $c$ of the controllers and the current state is over-provisioning, the idle controller immediately starts to take working vacation until the number of the working vacation controllers is equal to $d$.

IV. QUEUEING MODEL

A. Quasi-Birth-and-Death Process Model

In this section, we establish the $N(d,c)-M/M/c/K/AMWV$ queuing system, and analyze the implement cost and performance of the FSDMC model. For convenience, the packet-in message, controller, and semi-dormant state are referred to as customer, server, and working vacation state, respectively. In the FSDMC, it is assumed that the interarrival time, the service time during a normal busy period, and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.
the service time during a working vacation follow exponential distributions with parameter \( \lambda, \mu_B, \) and \( \mu_V (\mu_V < \mu_B) \) in
the single global queue, respectively, [17].

The other features of the queueing system proposed in this paper are exactly the same as those of the queueing system presented in [17], except for the different ways of working vacations. Particularly, the former is an asynchronous way of multiple working vacations of partial servers, whereas the latter is a synchronous way. In the \( N/(d,c)-M/M/c/K/AMWV \) queue model, when the number of customers in the system is less than the number \( c \) of the servers, the idle server immediately starts to take working vacation with exponential rate \( \theta \) until the number of the working vacation servers is equal to \( d \). When the vacation of a server has been completed and if the number of customers in the system is less than the threshold \( N \), then the vacation server takes another independent and identically distributed working vacation; otherwise, the vacation server returns and proceed to serve the waiting customers with service rate \( \mu_B \) until the number of the customers is less than or equal to \( c - d \).

In this paper, we assume that all interarrival, service, and vacation times are mutually independent, and first-come, first-served discipline is applied in our queueing model. Suppose that a server can serve only one customer at a time [17].

Let \( L_v(t) \) and \( J(t) \) be the number of packet-in messages and the number of multiple working vacation controllers in the system at time \( t \), respectively. Then, \( \{L_v(t), J(t)\} \) is a quasi-birth-and-death (QBD) process with the state space
\[
\Omega = \{(k, d) : 0 \leq k \leq c - d\} \\
\cup \{(k, j) : c - d < k \leq c - 1, c - k \leq j \leq d\} \\
\cup \{(k, j) : c \leq k \leq K, 0 \leq j \leq d\}.
\]

The state-transition-rate diagram for the queueing system is shown in Fig. 1. The lexicographical sequence for the states indicates that the infinitesimal generator \( Q \) of the process is the same as that of the \( N/(d,c)-M/M/c/K/SMVW \) queueing system (see [17, Sec. IV-B]).

Let \( \sigma_j = (c - j)\mu_B + j\mu_V, \delta_{k,j} = (k - j)\mu_B + j\mu_V, 0 \leq j \leq d, 0 \leq k \leq c, \) and \( q = k - c + d, \) then \( \delta_{c,j} = \sigma_j \). The submatrices in \( Q \) are written as follows:

\[
A_k = \begin{cases}
-(\lambda + \delta_{k,0}), & 0 \leq k \leq c - d \\
-\lambda I - \text{diag}(\delta_{k,q}, \delta_{k,q-1}, \ldots, \delta_{k,1}, \delta_{k,0}) & c - d < k \leq c
\end{cases}
\]

\[
B_k = \begin{cases}
-\lambda I - \text{diag}(\sigma_d, \sigma_d-1, \ldots, \sigma_1, \sigma_0) & c < k \leq N - 1 \\
-\text{diag}(\sigma_d, \sigma_d-1, \ldots, \sigma_1, \sigma_0) + \Lambda, & k = K
\end{cases}
\]

\[
\delta_{k,0}, \quad \begin{pmatrix}
\text{diag}(\delta_{k,q}, \delta_{k,q-1}, \ldots, \delta_{k,1}) \\
(0, 0, \ldots, 0, \delta_{k,0})
\end{pmatrix}_{(q+1) \times q},
\]

\[
\text{diag}(\delta_{k,q}, \delta_{k,q-1}, \ldots, \delta_{k,0}) & c - d < k \leq c \\
\delta_{k,0} & c < k \leq N - 1
\end{cases}
\]

\[
C_k = \begin{cases}
\lambda, & 0 \leq k < c - d \\
\lambda I_{(q+1) \times (q+1)} & c - d < k \leq c \\
\lambda I_{(d+1) \times (d+1)} & c \leq k \leq N - 1
\end{cases}
\]

and

\[
A = -\lambda I - \text{diag}(\sigma_d, \sigma_d-1, \ldots, \sigma_1, \sigma_0) + \Lambda, & N \leq k < K
\]

\[
B = \text{diag}(\sigma_d, \sigma_d-1, \ldots, \sigma_1, \sigma_0), & N \leq k \leq K
\]

\[
C = \lambda I_{(d+1) \times (d+1)}, & N \leq k < K
\]

where \( I \) is an appropriate order identity matrix, \( 0_{(q+1) \times 1} \) is a \((q + 1)\)-dimensional column vector of zeros, \( \text{diag}(\delta_{k,q}, \delta_{k,q-1}, \ldots, \delta_{k,1}, \delta_{k,0}) \) and \( \text{diag}(\sigma_d, \sigma_d-1, \ldots, \sigma_1, \sigma_0) \)
denote diagonal matrices, and
\[ \Lambda = \begin{pmatrix} -d_0 & d_0 & & & & \\
 & -d_1 \theta & d_1 \theta & & & \\
 & & \ddots & \ddots & & \\
 & & & -\theta & \theta & \\
 & & & & 0 & \cdots \\
 & & & & & \cdots \\
 & & & & & & d_{d+1} \end{pmatrix}. \]

This QBD process can be analyzed by solving the minimal non-negative solution of the matrix quadratic equation
\[ R^2 B + RA + C = 0 \tag{2} \]
and \( R \) is called the rate matrix \([19]\).

**Theorem 1:** If \( \rho = \lambda \sigma_0^{-1} < 1 \), then the matrix (2) possesses the minimal non-negative solution
\[ R = \begin{pmatrix} r_d & r_{d,d-1} & \cdots & r_{d,1} & r_{d,0} \\
 r_{d-1} & r_{d,d-2} & \cdots & r_{d,1} & r_{d,0} \\
 \vdots & \vdots & \ddots & \vdots & \vdots \\
 & & & r_1 & r_{1,0} \\
 & & & & r_0 \end{pmatrix} \tag{3} \]

where 1) the diagonal element \( r_k \) is a real root of the quadratic equation
\[ \sigma_k z^2 - (\lambda + \sigma_k + k \theta) z + \lambda = 0, 0 \leq k \leq d \tag{4} \]
and \( r_k = (1/2\sigma_k)(\lambda + \sigma_k + k \theta - \sqrt{\Delta}) \), where \( \Delta = (\lambda + \sigma_k + k \theta)^2 - 4\sigma_k \lambda, 0 < r_k < 1, r_0 = \rho \), and \( r_k \) satisfies the following relationship:
\[ \lambda + k \theta + \sigma_k (1 - r_k) = \sigma_k + k \theta \]
\[ 1 \]
\[ r_k = \frac{\lambda}{1 - r_k}. \tag{5} \]

2) the nondiagonal element satisfies
\[ r_{u,v} = \begin{cases} \frac{u v + 1}{[(\lambda + \sigma_u + \nu \theta - \sigma_v (r_u + r_v))]}, & u = v + 1 \\
\frac{\nu v + 1}{[(\lambda + \sigma_u + \nu \theta - \sigma_v (r_u + r_v))]} & v + 1 < u \leq d \end{cases} \tag{6} \]

where \( u = d - i, v = d - j, 0 \leq i < d, i < j \leq d \), and \( r_{k,k} = r_k, 0 \leq k \leq d \).

**Proof:** It is assumed that another real root of (4) is \( r_k^* \). Then, we have
\[ r_k \neq r_k^* = \frac{1}{2\sigma_k} \left( \lambda + \sigma_k + k \theta \pm \sqrt{\Delta} \right). \]

Obviously, the following inequalities hold:
\[ (\lambda - \sigma_k + k \theta)^2 < \Delta < (\lambda + \sigma_k + k \theta)^2, \lambda > \sigma_k \]
\[ (\sigma_k - \lambda + k \theta)^2 < \Delta < (\lambda + \sigma_k + k \theta)^2, \lambda > \sigma_k. \]

We can obtain \( 0 < r_k < 1, r_k^* > 1 \), and \( r_k = (1/2\sigma_k)(\lambda + \sigma_k + k \theta - \sqrt{\Delta}) \).

If \( k = 0 \), then \( r_0 = \rho \).

Substituting \( r_0 \) into (4) yields (5).

In (2), each of matrices \( A, B, \) and \( C \) is an upper triangular matrix. Matrix \( R \) is also an upper triangular matrix. Hence, matrix \( R^2 \) can also be represented as an upper triangular matrix. Let \( R \) be denoted as (3). Then,
\[ (R^2)_{ij} = \begin{cases} r_{d-j}^2, & i = j, 0 \leq j \leq d \\\n\sum_{l=i}^d r_{d-l,i,d-h,d-h-d-j}, & 0 \leq i < d, i < j \leq d. \end{cases} \]

Set \( u = d - i, v = d - j, g = d - h, \) and \( r_{k,k} = r_k, 0 \leq k \leq d \).

Substituting \( R^2 \) and \( R \) into the matrix (2), yields
\[ \begin{aligned}
\sigma_k r_k^2 - (\lambda + \sigma_v + \nu \theta) r_k + \lambda &= 0, 0 \leq v \leq d \\
\sigma_k \sum_{r_{u,v} > 0} r_{u,v} r_{u,v} + (v + 1) \theta r_{u,v} + (\lambda + \sigma_v + \nu \theta) r_{u,v} &= 0 \\
0 &\leq u \leq d, 0 \leq v < u. \tag{7} \end{aligned} \]

In (7), the first equation is (4). We take \( r_k \) to be the root of (4) over the interval \((0, 1), \) where \( r_k = (1/2\sigma_k)(\lambda + \sigma_v + \nu \theta - \sqrt{\Delta}), \) and \( \Delta = (\lambda + \sigma_v + \nu \theta)^2 - 4\sigma_k \lambda. \) In the second equation, if \( u = v + 1 \), then \( \sum_{r_{u,v} > 0} r_{u,v} r_{u,v} = r_{u,v} r_{u,v} + r_{u,v} r_{u,v} \); if \( u > v + 1 \), then \( \sum_{r_{u,v} > 0} r_{u,v} r_{u,v} = r_{u,v} r_{u,v} + r_{u,v} r_{u,v} + \sum_{v<u} r_{u,v} r_{u,v}. \)

Accordingly, (6) can be easily obtained. Obviously, \( \text{SP}(R) = \max_{0 \leq k \leq d} \{ r_k \} < 1 \) if and only if \( \rho < 1 \).

**B. Equilibrium Probability Distribution**

If \( \rho < 1 \), then let \( (L_v, J) \) be the equilibrium limit of the QBD process \((L_v(t), J(t))\). Let
\[ \pi_{k,j} = \lim_{t \to \infty} P(L_v(t) = k, J(t) = j), (k,j) \in \Omega. \]

Then
\[ \pi_{k,j} = \begin{cases} \pi_{k,d}, & 0 \leq k \leq c - d \\
\pi_{k,d-1}, \ldots, \pi_{k,c-k+1}, \pi_{k,c-1} \end{cases} \times \begin{cases} 1 \times (c-k+1), & c - d < k \leq c \\
\pi_{k,c-k}, \ldots, \pi_{k,c-1} \end{cases} \times \begin{cases} 1 \times (d+1), & c - k < N, 1 \leq j \leq d \\
\pi_{c,j} \end{cases} \times \begin{cases} 1 \times (c - k), & c - k < N, 1 < j \leq d \\
\pi_{c,j} \end{cases} \times \begin{cases} 1 \times (d+1), & c - k < N, j = 0 \end{cases} \tag{8} \]

and
\[ \pi_{N,j} = \begin{cases} r_d \pi_{c,d} \left( \frac{\lambda}{\sigma_d} \right)^{N-1}, & j = d \\
\frac{r_j}{\lambda} \sum_{i=j+1}^d \pi_{N,i} r_{i,j} + (j+1) \theta \pi_{N,j+1} \end{cases} \times \begin{cases} 1 \times (d+1), & 1 \leq j \leq d - 1 \\
\pi_{c,j} \end{cases} \times \begin{cases} 1 \times (c - k), & c - k < N, j = 0 \end{cases} \times \begin{cases} 1 \times (c - k), & c - k < N, j = 0 \end{cases} \tag{9} \]
where

\[
\pi_{c,j} = \begin{cases}
\frac{G^{\lambda c}}{\prod_{j=0}^{d} \lambda_{j}^{\alpha} \prod_{i=0}^{d} \lambda_{i}^{\alpha + d - i - j}} 
\left(1 + (\lambda - \sigma_d r_d) \left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}\Phi(d)\right) 
\left(\frac{\lambda}{\sigma_d}\right)^{\lambda j - \pi_k/j} 
\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c} 
\left(\frac{\lambda}{\sigma_d}\right)^{\lambda j - \pi_k/j} 
\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}
\right)
\end{cases}
\]

where \( \Phi(u) = \sum_{n=0}^{u-1} \lambda^{n} \prod_{i=n+1}^{\infty} \lambda_{i}^{\alpha} \), and the constant factor \( G \) can be yielded by the normalization condition.

Proof: \( \alpha \)-dimensional row vector \((\pi_0, \pi_1, \pi_2, \ldots, \pi_N)\) satisfies the set of equations

\[
(\pi_0, \pi_1, \pi_2, \ldots, \pi_N)B[R] = 0
\]

where \( \alpha = (1/2)d^2 + (N - c - 1/2)d + N + 1 \), and \( B[R] \) is an \( \alpha \times \alpha \) square matrix shown in [17, Th. 2].

Substituting \( B[R] \) into the above equation yields \( \alpha \) linear equations, as shown as follows:

\[
-\lambda \pi_{0,d} + \delta_{1,0} \pi_{1,d} = 0, \quad k = 0, \quad j = d
\]

\[
\lambda \pi_{k-1,d} - (\lambda + \delta_{k,0}) \pi_{k,d} + \delta_{k+1,0} \pi_{k+1,d} = 0
\]

\[
\quad 1 \leq k \leq c - d - 1, \quad j = d
\]

\[
\lambda \pi_{c-d-1,d} - (\lambda + \delta_{c-d,0}) \pi_{c,d-1} + \delta_{c-d+1,0} \pi_{c,d} = 0
\]

\[
\quad k = c - d, \quad j = d
\]

\[
\lambda \pi_{k-1} - (\lambda + \delta_{k,0}) \pi_{k,d} + \delta_{k+1,0} \pi_{k+1,d} = 0
\]

\[
\quad c - d < k \leq c - 1, \quad j = j - 1
\]

\[
\lambda \pi_{k-0} - (\lambda + \delta_{k,0}) \pi_{k,d} + \delta_{k+1,0} \pi_{k+1,d} = 0
\]

\[
\quad k = c, \quad j = 0
\]

\[
\lambda \pi_{k-1} - (\lambda + \delta_{k,0}) \pi_{k,d} + \delta_{k+1,0} \pi_{k+1,d} = 0
\]

\[
\quad c + 1 \leq k \leq N - 1, \quad j = 0
\]

\[
\lambda \pi_{k-1,j} - (\lambda + \delta_{k-1,j}) \pi_{k,j} + \delta_{k+1,j} \pi_{k+1,j} = 0
\]

\[
\quad c - d - 2 \leq k \leq c - 1, \quad c - k - 1 \leq j \leq d
\]

\[
\lambda \pi_{k-1,j} - (\lambda + \delta_{k-1,j}) \pi_{k,j} + \delta_{k+1,j} \pi_{k+1,j} = 0
\]

\[
\quad c + 1 \leq k \leq N - 1, \quad 1 \leq j \leq d
\]

\[
\lambda \pi_{N-1,j} + \sum_{i=j+1}^{d} \pi_{N,i} \delta_{i,j} \sigma_{j} + (j + 1) \theta \pi_{N,j+1}
\]

\[
\quad c - d - 1 \leq k \leq c, \quad 0 \leq j \leq d - 1
\]

\[
\lambda \pi_{N-1,d} + \left\lfloor r_d \sigma_{d} - (\lambda + \sigma_d d + \theta) \right\rfloor \pi_{N,d} = 0
\]

\[
\quad k = N, \quad j = d.
\]

From (20), we have

\[
\delta_{k,0} \pi_{k,d} - \lambda \pi_{k-1,d} = \delta_{1,0} \pi_{1,d} - \lambda \pi_{0,d}, \quad 1 \leq k \leq c - d.
\]

From (19) and (29), we obtain (8).

From (24), (25), and \( \delta_{c,0} = \sigma_0 \), we get

\[
\pi_{k+1,0} = \rho \pi_{k,0} + \pi_{c,0}, \quad c < k < N.
\]

When (30) is iterated, we obtain (12).

Therefore, (26) can be satisfied by (11).

From (5), (11), and (28), we obtain (13), that is,

\[
\pi_{N,d} = r_d \pi_{N-1,d} = r_d \pi_{c,d} \left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}.
\]

Substituting (11) and (12) into (27) yields (14) and (15), respectively. From (26), we have

\[
\sigma_j \pi_{N-j} - \lambda \pi_{N-j-1} = \sigma_j \pi_{c,j} - \lambda \pi_{c-1,j}, \quad 1 \leq j \leq d.
\]

From (23), (31), and \( \delta_{c,j} = \sigma_j \), we obtain

\[
\delta_{k,k+j-c} \pi_{k,j} = \lambda \pi_{k-1,j} + \delta_{c,j} \pi_{N-j} - \lambda \pi_{N-1,j},
\]

\[
c - d < k \leq c, \quad c - k - 1 \leq j \leq d.
\]

Iterating (32) yields (9). In (9), let \( k = c, j = d \). Substituting (8) and (13) into (9) yields (16).

From (20) and (21), we obtain

\[
\pi_{c-d+1,d-1} = \lambda \pi_{N-1,d} - \delta_{c,d} \pi_{N,d}
\]

\[
= \lambda - \delta_{c,d} r_d \left(\frac{\lambda}{\sigma_d}\right)^{N-1-c} \pi_{c,d}.
\]

From (22), we obtain

\[
\pi_{k,j} = \frac{\delta_{k,0}}{\delta_{k,0} \left[\pi_{c,j+1} \left(\frac{\lambda}{\sigma_j}\right)^{N-c} - \pi_{N,j+1}\right]}.
\]

Substituting (9) into (34) yields (35)

\[
\pi_{k,j} = \frac{1}{\delta_{k,0}} \sum_{j=c-k}^{d} \sigma_{j} \left[\pi_{c,j} \left(\frac{\lambda}{\sigma_j}\right)^{N-c} - \pi_{N,j}\right]
\]

\[
c - d + 2 \leq k \leq c - 1, \quad j = c - k.
\]

Therefore, (33) satisfies (36) when \( k = c - d + 1 \) and \( j = d - 1 \) in (36). Thus, (10) can be derived from (33) and (36).

In (9), let \( 1 \leq j \leq d - 1 \). Then, \( s = c - j \). From (9), (10), and (14), we obtain (17).

Equation (18) can be expressed from (12) and (15).

Finally, \( G \) satisfies the normalization condition as follows:

\[
\sum_{k=0}^{c-d-1} \pi_{k,d} + \sum_{j=0}^{d} \sum_{k=c-j}^{K} \pi_{k,j} = 1.
\]

Theorem 3: If \( k > N \), then the equilibrium probability distribution of \((L, J)\) is

\[
\pi_{k,d} = \pi_{N,d} \left(\frac{k-N}{d}\right)^{N-1}. \quad N < k < K, \quad j = d
\]
\[ \psi_k^{(d)} = \pi_{N,d} \sum_{i=1}^{k-N} r_{d-1} \eta H^{k-N-i} + \sum_{j=1}^{d} K_{d} \] (38)

\[ N < k < K, \quad 0 \leq j \leq d - 1 \]

\[ \pi_{K,d} = \frac{\lambda r_{d}^{K-1-N}}{\sigma_d + d \theta} \pi_{N,d}, \quad j = d \] (39)

\[ \pi_{K,j} = \lambda \sum_{h=j}^{d-1} \left[ \frac{1}{(h+1) \theta} \prod_{i=j}^{h} \frac{(i+1) \theta}{\sigma_i + i \theta} \pi_{K-1,h} \right] + \frac{\lambda r_{d}^{K-1-N}}{\sigma_d + d \theta} \sum_{j=1}^{d} (i+1) \theta \pi_{K,d}, \quad 0 \leq j \leq d - 1 \] (40)

where \( \sigma_d + d \theta \).

From (40), we obtain (39), that is,

\[ \pi_{K,d} = \frac{\lambda r_{d}^{K-1-N}}{\sigma_d + d \theta} \pi_{N,d}. \]

We can obtain the k power of the rate matrix R as follows:

\[ R^k = \begin{pmatrix} r_{d}^k & \sum_{i=1}^{k} r_{d-1} \eta H^{k-i} \\ 0 & H^k \end{pmatrix}. \]

Substituting (42) into (41) yields (37) and (38), respectively. When \( k = K \), based on the equilibrium equations, we have

\[ \lambda \pi_{K-1,d} - (\sigma_d + d \theta) \pi_{K,d} = 0, \quad j = d \]

\[ \lambda \pi_{K-1,j} - (\sigma_j + j \theta) \pi_{K,j} + (j+1) \theta \pi_{K,j+1} = 0 \]

\[ 0 \leq j \leq d - 1. \]

From (43), we obtain (39), that is,

\[ \pi_{K,d} = \frac{\lambda}{\sigma_d + d \theta} \pi_{K-1,d} = \frac{\lambda r_{d}^{K-1-N}}{\sigma_d + d \theta} \pi_{N,d}. \]

From (44), we have

\[ \pi_{K,0} = \lambda \sum_{h=0}^{j-1} \frac{h! \eta^h}{(\sigma_i + i \theta)^h} \pi_{K-1,h} + \frac{j! \eta^j}{(\sigma_i + i \theta)^j} \pi_{K,j}, \quad 1 \leq j \leq d. \]

In (45), let \( j = d \). Then, obtain the next expression

\[ \pi_{K,d} = \lambda \sum_{h=j}^{d-1} \left[ \frac{1}{(h+1) \theta} \prod_{i=j}^{h} \frac{(i+1) \theta}{\sigma_i + i \theta} \pi_{K-1,h} \right] + \sum_{i=j}^{d-1} (i+1) \theta \pi_{K,d}, \quad 0 \leq j \leq d - 1. \] (46)

Substituting (39) into (46) yields (40).

\[ \text{C. Performance Measures} \]

Using the above equilibrium distribution, the performance indices of the \( N(d,c)-\text{M/M/}c/\text{K/AMWV} \) queueing system can be numerically evaluated as follows.

1) The mean number of customers in the system is

\[ E[L_s] = \sum_{i=0}^{c-d-1} i \pi_{i,d} + \sum_{j=0}^{d} K_j \sum_{i=c-j}^{K_j} i \pi_{i,j}. \]

2) The average number of customers in the queue is

\[ E[L_q] = \sum_{j=0}^{d} K_j \sum_{i=0}^{d} (i - c + j) \pi_{i,j}. \]

3) The expected number of servers during normal busy periods is

\[ E[NB] = c - E[WV] - E[I]. \]

4) The expected number of servers during working vacation periods is

\[ E[WV] = d \sum_{i=0}^{c-d-1} \pi_{i,d} + \sum_{j=1}^{d} \sum_{i=c-j}^{K_j} j \pi_{i,j}. \]

5) The mean number of servers during idle periods is

\[ E[I] = \sum_{i=0}^{c-d} (c - d - i) \pi_{i,d}. \]

6) The average waiting time that the customer sojourns in the queue is

\[ E[T_q] = \frac{E[L_q]}{\lambda (1 - P_{loss})}. \]

7) The loss probability in the system is

\[ P_{loss} = \sum_{j=0}^{d} \pi_{K,j}. \]

\[ \text{V. OPTIMIZATION AND SENSITIVITY ANALYSIS} \]

\[ \text{A. Minimizing Average CPR} \]

1) Cost: We utilize the concept of crew-service equipment by White et al. [42]. The average cost function per unit time is given by

\[ F(d, N, K, \mu_B, \mu_V, \lambda, \theta) = C_h E[L_s] + C_b E[NB] + C_v E[WV] + C_i E[I] + C_d E[T_q] + C_P_{loss} \]

where the cost parameters \( C_h, C_b, C_v, C_i, C_d, \) and \( C_p \) are defined in [17], and \( E[L_s], E[NB], E[WV], E[I], E[T_q], P_{loss} \) are given by (47) and (49)–(53), respectively.

2) Performance: We use the reciprocal of the average delay time that the customer stays in the system as the performance index (see [15, Sec. V]): \( P = (1/E[T]), \) where

\[ E[T] = (E[L_s]/(\lambda (1 - P_{loss}))). \]
3) CPR: In accordance with [15, eq. (16)] and (54), we can define the CPR of the FSDMC as CPR = \( \frac{F}{P} = F \times E[T] \). We consider the optimal model as follows:

\[
\begin{align*}
\text{Minimize} & \quad \text{CPR}(d, N, K, \mu_B, \mu_V, \lambda, \theta) \\
\text{subject to} & \quad 0 < d < c < N < K < K', \quad \mu_V < \mu_B \\
& \quad \mu_B \leq \mu_B \leq \mu_B'' \quad \mu_V \leq \mu_V'' \quad \lambda^\prime \leq \lambda \leq \lambda^\prime, \theta^\prime \leq \theta \leq \theta^\prime
\end{align*}
\]

(55)

where the lower and upper bounds of the parameters are described in [17, Sec. V].

We aim to determine the optimal values of parameters \( d, N, K, \mu_B, \mu_V, \lambda, \theta \) in the system to provide the configuration of the system in steady state. Thus, a PSO algorithm [20] is implemented to address the optimization problem numerically.

4) Optimal System Configuration: We solve the optimization model (55) to find the minimum value of the CPR in the system by using PSO algorithm. We select various parameter values of the system in [17] as follows: \( C_h = 1 \text{ mW}, C_b = 35 \text{ mW}, C_i = 15 \text{ mW}, C_t = 10 \text{ mW}, C_d = 8 \text{ mW}, C_{t'} = 10000 \text{ mW}, \text{ and } c = 10 \). The system parameters are selected as \( d \in [1, c], N \in [c + 1, 25], K \in [N + 1, 60], \mu_B \in [1.6, 3.4], \mu_V \in [0.5, 1.5], \lambda \in [4, 30], \) and \( \theta \in [0.01, 10], \) and they meet the constraint conditions in the optimization model (55).

In our experiment, the optimal values of the decision variables are yielded as follows: \( d = 4, N = 20, K = 42, \mu_B = 3.4, \mu_V = 1.5, \lambda = 16.9337, \theta = 6.6058 \). The minimum value of CPR is obtained, that is, CPR0 = 53.1134. We configure the system using the optimal values of parameters. At this point, we call this system a stable system. Notably, in the solving process, the optimal values of \( \mu_B \) and \( \mu_V \) are always taken from the upper bounds \( \mu_B'' \) and \( \mu_V'' \) of the respective constraints, because the CPR decreases with the increase of \( \mu_B \) or \( \mu_V \) (see Figs. 3 and 4 and Tables V and VI). Therefore, we adopt the values of \( \mu_B \) and \( \mu_V \) in [17] as the values of parameters \( \mu_B \) and \( \mu_V \) in our system, respectively.

B. Sensitivity Analysis of Performance Indices

Under stability condition, the numerical results show that changes in system parameters have an impact on system performance. Let \( K = 42 \) and consider that system parameters take different values in the following three cases.

Case 1: \( \lambda = 16.9, \theta = 6.6, d = 4, c = 10, N = 20, \) and changing the values of \( \mu_B, \mu_V \).

Case 2: \( \mu_B = 3.4, \mu_V = 1.5, d = 4, c = 10, N = 20, \) and changing the values of \( \lambda, \theta \).

Case 3: \( \lambda = 16.9, \theta = 6.6, \mu_B = 3.4, \mu_V = 1.5, \) and changing the values of \( d, c, N \).

The numerical results for the above three cases are shown in Tables I–III, respectively. Table I shows that, as the value of \( \mu_B \) or \( \mu_V \) increase, \( E[L_a], E[L_q], \) and \( E[NB] \) decrease and \( E[WV] \) slightly decreases but \( E[I] \) increases. As shown in Table II: 1) \( E[L_a] \) and \( E[NB] \) increase with the increase of \( \lambda \) but slightly increase with the increase of \( \theta \); 2) \( E[L_q] \) and \( E[WV] \) increase with the increase of \( \lambda \) but slightly decrease with the increase of \( \theta \); and 3) \( E[I] \) decreases with the increase of \( \lambda \) but does not change with the increase of \( \theta \). Table III shows that: 1) \( E[L_a], E[L_q], \) and \( E[WV] \) drastically increase as \( d \) increases; however, \( E[NB] \) and \( E[I] \) drastically decrease as \( d \) increases; 2) that as the values of \( c \) increase, \( E[L_a] \) and \( E[WV] \) decrease and \( E[L_q] \) drastically decreases; meanwhile, \( E[NB] \) increases and \( E[I] \) drastically increases; and 3) that similar to 1), \( E[L_a], E[L_q], \) and \( E[WV] \) slightly increase but \( E[NB] \) and \( E[I] \) slightly decrease with the increase of \( N \). These results are consistent with the actual situation for the \( N(d,c)-M/M/c/K/AMWV \) queue.

C. Sensitivity Analysis of CPR

Under stability condition, some results from our numerical experiment show that the CPR is impacted by the change in system parameters. We fix the values of the cost parameters as follows: \( C_h = 1 \text{ mW}, C_b = 35 \text{ mW}, C_i = 15 \text{ mW}, C_t = 10 \text{ mW}, C_d = 8 \text{ mW}, C_{t'} = 10000 \text{ mW}, \) controller number \( c = 10, \) and capacity \( K = 42 \). Consider that system parameters take different values in the following eight cases.

Case 4: \( N = 20, \mu_B = 3.4, \mu_V = 1.5, \theta = 6.6, \) and changing the values of \( \lambda \) in the interval \([1.0, 30.0]\) and \( d = 4, 7, 9, \)

Case 5: \( N = 20, \mu_V = 1.5, \lambda = 16.9, \theta = 6.6, \) and changing the values of \( \mu_B \) in the interval \([2.0, 20.0]\) and \( d = 4, 7, 9, \)

Case 6: \( N = 20, \mu_B = 3.4, \lambda = 16.9, \theta = 6.6, \) and changing the values of \( \mu_V \) in the interval \([0.0, 10.0]\) and \( d = 4, 7, 9, \)
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Case 7: \( N = 20, \mu_B = 3.4, \mu_V = 1.5, \lambda = 16.9, \) and changing the values of \( \theta \) in the interval \([2.0, 20.0]\) and \( d = 4, 7, 9. \)

Case 8: \( d = 4, \mu_B = 3.4, \mu_V = 1.5, \lambda = 16.9, \theta = 6.6, \) and changing the values of \( \mu_B, N. \)

Case 9: \( d = 4, \mu_V = 1.5, \lambda = 16.9, \theta = 6.6, \) and changing the values of \( \mu_V, N. \)

Case 10: \( d = 4, \mu_B = 3.4, \lambda = 16.9, \theta = 6.6, \) and changing the values of \( \mu_V, N. \)

Case 11: \( d = 4, \mu_B = 3.4, \mu_V = 1.5, \lambda = 16.9, \) and changing the values of \( \mu_B, N. \)

The experimental results of CPR are reported in Figs. 2–5 for cases 4–7 and in Tables IV–VII for cases 8–11, respectively. As shown in Fig. 2 and Table IV, CPR significantly fluctuates as \( \lambda \) and \( d \) increase, but increases with \( N. \) As shown in Figs. 3 and 4, CPR drastically decreases as \( \mu_B \) and \( \mu_V \) increase but shows fluctuations as \( d \) changes. Tables V and VI show that CPR slightly increases with the increase of \( N. \) From Fig. 5 and Table VII, CPR increases with \( d \) and \( N \) but slightly changes with the increase of \( \theta. \)

VI. QUANTITATIVE MODEL OF ANELASTICITY

A. MM/1/K Queue in Markovian Environment

In this section, we consider an \( M/M/1/K \) queue model in a Markovian environment, where \( K \) denotes the system capacity. The Markovian environment is described by a three-state, irreducible Markov renewal process with generator \( \bar{Q}. \) For simplicity, the three states are represented by states 1, 2, and 3.
When a transition from state $i$ to state $j$ occurs in process $Q$, the average Poisson arrival rate $\lambda_i$, and average service rate $\mu_j$ instantaneously change to new values of $\lambda_j$ and $\mu_j$, respectively, where $1 \leq i, j \leq 3$.

In accordance with the transitions among the three-state (see Section III), matrix $\check{Q}$ is written as

$$\check{Q} = \begin{pmatrix} -\mu_1 & \mu_1 & 0 \\ \mu_2 & -2\mu_2 & \mu_2 \\ 0 & \mu_3 & -\mu_3 \end{pmatrix}.$$ 

In $\check{Q}$, the mean service rate $\mu_i$ is computed by

$$\mu_i = E[NB_i] \mu_B + E[WV_i] \mu_V, \quad i = 1, 2, 3$$

(56)

where $E[NB_i]$ and $E[WV_i]$ represent the expected number of servers during normal busy and working vacation periods in the $i$th state, respectively. Their values can be obtained from (49)–(51).

Let $\bar{\pi} = (\bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3)$ be the stationary probability vector of Markov process $Q$. Then,

$$\bar{\pi} \check{Q} = 0, \quad \bar{\pi} e = 1.$$ 

(57)

From (57), we obtain

$$\bar{\pi} = (\mu_2\mu_3, \mu_3\mu_1, \mu_1\mu_2)(\mu_1\mu_2 + \mu_2\mu_3 + \mu_3\mu_1)^{-1}.$$ 

By a standard construction, the $M/M/1/K$ queue may be studied as a QBD process with generator $\check{Q}$, given by

$$\check{Q} = \begin{pmatrix} \check{Q} - \Delta(\lambda) & \Delta(\lambda) & \Delta(\lambda) \\ \Delta(\mu) & \check{Q} - \Delta(\lambda + \mu) & \Delta(\lambda) \\ \cdots & \cdots & \cdots \\ \Delta(\mu) & \Delta(\lambda + \mu) & \check{Q} - \Delta(\mu) \end{pmatrix}$$

where $\Delta(\lambda)$ and $\Delta(\mu)$ represent diagonal matrices $\text{diag}(\lambda_1, \lambda_2, \lambda_3)$ and $\text{diag}(\mu_1, \mu_2, \mu_3)$, respectively, (see [19, Sec. VI-2]).

Matrix $\check{R}$ is the minimal non-negative solution of the matrix quadratic equation

$$\check{R}^2 \Delta(\mu) + \check{R} \Delta(\lambda + \mu) + \Delta(\lambda) = 0$$

(58)

where $\check{R}$ can be solved by employing the fast Newton–Shamanskii iteration algorithm [43].

Let $X = (x_0, x_1, x_2, \ldots, x_K)$ be the stationary probability vector of the stable queue, where $x_k = (x_{k,1}, x_{k,2}, x_{k,3}), 0 \leq k \leq K$. In $x_{k,j}, k$ and $j$ denote the queue length and the environmental state, respectively, $j = 1, 2, 3$.

**Theorem 4:** If $\bar{\rho} = \bar{\pi} \check{Q}(\bar{\pi} \mu)^{-1} < 1$, then the stationary probability distribution $x_k$ is

$$x_k = \begin{cases} \bar{G} \check{Q}(I - \check{R}) \bar{\pi}, & 0 \leq k < K \\ \bar{G} \check{Q}(I - \check{R}) \check{R}^{K-1} \Delta(\lambda)(\Delta(\mu) - \check{Q})^{-1}, & k = K \end{cases}$$

(59)

where $\bar{\lambda} = (\lambda_1, \lambda_2, \lambda_3)^T$, $\bar{\mu} = (\mu_1, \mu_2, \mu_3)^T$, and

$$\bar{G} = \left[ \check{Q}(I - \check{R}) \bar{\pi} + \check{Q}(I - \check{R}) \check{R}^{K-1} \Delta(\lambda) \times (\Delta(\mu) - \check{Q})^{-1} e \right]^{-1}.$$ 

(60)

**Proof:** From [19, Th. 6.2.1], we have

$$x_k = \bar{G} \check{Q}(I - \check{R}) \bar{\pi}, \quad 0 \leq k < K$$

(61)

where $\bar{G}$ denotes a pending constant.

From $x_k \check{Q} = 0$, we can obtain the equilibrium equation as follows:

$$x_{K-1} \Delta(\lambda) + x_K \check{Q} - \Delta(\mu) = 0.$$ 

(62)

From (61) and (62), we obtain the second expression in (59). The pending constant $\bar{G}$ can be obtained by the normalization condition

$$\sum_{k=0}^{K-1} \bar{G} \check{Q}(I - \check{R}) \check{R}^k \bar{\pi} + \bar{G} \check{Q}(I - \check{R}) \check{R}^{K-1} \Delta(\lambda) \times (\Delta(\mu) - \check{Q})^{-1} e = 1.$$ 

(63)

From (63), (60) can be obtained.

**B. Sensitivity Analysis of Elasticity**

From (59), the probabilities that the system is in the three states are calculated using (64), respectively,

$$P_{\text{over}} = \sum_{j=0}^{c-1} \sum_{i=0}^{N-1} x_{i,j}, \quad P_{\text{normal}} = \sum_{j=1}^{N} \sum_{i=c}^{N-1} x_{i,j}, \quad P_{\text{under}} = \sum_{j=1}^{3} \sum_{i=0}^{K} x_{i,j}.$$ 

(64)

In the case where the traffic intensity $\bar{\rho} = (\bar{\pi} \check{Q})(\bar{\pi} \mu)^{-1} < 1$, we use numerical data to demonstrate that elasticity is impacted by the change in the basic parameters of the system.

We fix the controller number $c = 10$, the semi-dormant controller number $d = 4$, the threshold value $N = 20$, and the capacity $K = 42$. Consider that system parameters take different values in the following four cases.

**Case 12:** $\mu_B = 3.4, \mu_V = 1.5, \theta = 6.6$, and changing the values of $\lambda$ in the interval $[5.0, 35.0]$.

**Case 13:** $\lambda = 16.9, \mu_V = 1.5, \theta = 6.6$, and changing the values of $\mu_B$ in the interval $[2.0, 20.0]$.

**Case 14:** $\lambda = 16.9, \mu_B = 3.4, \theta = 6.6$, and changing the values of $\mu_V$ in the interval $[0.0, 10.0]$.

**Case 15:** $\lambda = 16.9, \mu_B = 3.4, \mu_V = 1.5$, and changing the values of $\theta$ in the interval $[2.0, 20.0]$.

The numerical results of the elasticity are depicted in Figs. 6–8 for cases 12–14, respectively.
as \( \lambda \) increases, \( p_{\text{over}} \) drastically decreases and \( p_{\text{under}} \) drastically increases but \( p_{\text{normal}} \) changes, that is, initially increases and then slightly decreases. From Fig. 7, as \( \mu_B \) increases, \( p_{\text{normal}} \) and \( p_{\text{under}} \) decrease but \( p_{\text{over}} \) slightly increases. As shown in Fig. 8, as \( \mu_V \) increases, the change in the elasticity probabilities is the same as that in Fig. 7. However, for case 15, as \( \theta \) increases, the three probabilities remain unchanged, and \( p_{\text{over}} = 0.9945, p_{\text{normal}} = 0.5452e^{-02}, \) and \( p_{\text{under}} = 0.3038e^{-04}. \) Therefore, the elasticity of the system exhibits no correlation with the working vacations of the controller.

C. Adaptivity Analysis

In this section, we analyze the adaptive capability of the system. In the \( N/(d,c)-M/M/c/K/AMWV \) queueing system, when \( \rho = \lambda \sigma_0^{-1} < 1 \), the system changes the number \( d \) of semi-dormant controllers to adapt to the workload changes (see Fig. 2). However, for \( \rho \geq 1 \), we increase the mean service rate \( \mu_B \) of the controller in the system during a normal busy period and let \( d = 0 \) and \( \mu_V = \mu_B \). Accordingly, the queueing system becomes a classic queueing system \( M/M/c/K \).

As the average arrival rate decreases and when the number of customers in the system is less than \( c \), the idle controller starts to take working vacations until the number of the semi-dormant controllers is equal to \( d \). At this point, the system returns to a stable state again.

As the average arrival rate \( \lambda \) increases to \( \rho \geq 1 \), the system adaptively adjusts the controller service rate \( \mu_B \) during a normal busy period while keeping the optimal value CPR\(_0\) of CPR at a stable state. The optimal CPR unchanged adaptive algorithm based on PSO is summarized as Algorithm 1.

In Algorithm 1, rand() represents a random function in the closed interval \([0, 1]\). Algorithm 1 is essentially an extension of the PSO algorithm. Different from the PSO algorithm, Algorithm 1 sets the upper and lower bounds of the decision variables, and continuously changes the upper and lower bounds by the dichotomy to narrow the search space of the decision variable. Given the minimum value CPR\(_0\) of the objective function, the termination condition of the algorithm is that the global optimum value of the CPR is equal to the minimum value CPR\(_0\) within the given precision. The global optimal position of the particles is obtained by alternately reducing the range of values of the particles while the optimal value of CPR alternates close to the minimum value CPR\(_0\) on both sides. In Algorithm 1, we use a population of 100 particles and a mutation rate of 0.5. Let, high = 20, low = 1.6, and CPR\(_0\) = 53.1134.

In accordance with Algorithm 1, let \( d = 0, N = 20, K = 42, \) and vary the values of \( \lambda \) from 34.0 to 65.0 while \( \mu_B \) changes accordingly, \( p_{\text{over}}, p_{\text{normal}}, \) and \( p_{\text{under}} \) are influenced, as shown in Fig. 10. As \( \lambda \) increases, \( p_{\text{normal}} \) and \( p_{\text{under}} \) increase but \( p_{\text{over}} \) decreases.
The average arrival rate

Algorithm 1 Optimal CPR Unchanged Adaptive Algorithm Based on PSO

Input: The average arrival rate $\lambda$, and tolerance $\varepsilon = 10^{-7}$.
Output: The average arrival rate $\tilde{\mu}_B$ of the controller during a normal busy period.
1: Initialize the upper bounds $\text{high}$ and the lower bounds $\text{low}$, respectively, and $\text{low} < \text{high}$;
2: Set the initial value of the CPR is $CPR_0$;
3: while TRUE do
4:   for each particle $i$ do
5:      $x_i \leftarrow \text{high} + \text{rand()} \times (\text{high} - \text{low})$;
6:      $v_i \leftarrow \text{low} + \text{rand()} \times (\text{high} - \text{low})$;
7:      Evaluate particle $i$ using (55), and set $pBest_i = x_i$;
8:   end for
9:   $Best \leftarrow \min \{pBest_i\}$;
10:   for each particle $i$ do
11:      Update the velocity and position of particle $i$, and evaluate particle $i$;
12:      Update the global optimal position $gBest$, and $gCPR \leftarrow CPR(gBest)$;
13:   end for
14:   if $\text{abs}(gCPR - CPR_0) < \varepsilon$ then
15:      break;
16:   else
17:      $\text{mid} \leftarrow (\text{high} + \text{low})/2$;
18:      if $gCPR \leq CPR_0$ then
19:         $\text{temp} \leftarrow \text{high}$;
20:         $\text{high} \leftarrow \text{mid}$;
21:      else
22:         $\text{low} \leftarrow \text{high}$;
23:         $\text{high} \leftarrow \text{temp}$;
24:      end if
25:   end if
26: end while
27: return $gBest$.

In the $M/M/1/K$ queue model in a Markovian environment, the conditional queue length densities $q_k(j)$ are given by
$q_k(j) = P[L = k|j = j] = (x_k/j\bar{\pi}_j), 0 \leq k \leq K, 1 \leq j \leq 3$. From (1), we obtain that the anelasticity value of the system under stability condition is Anelasticity $= (1.4172, 5.4617e - 03)$.

Our experiment exhibits that as $\lambda$ increases, no linear relationship is found between the conditional average queue length AvgQLnormal of the queue in Phase 2 and elasticity $p_{\text{normal}}$. As shown in Fig. 11, for $\lambda \in [1, 34]$ (i.e., $\rho = \lambda \sigma_0^{-1} < 1$), as $\lambda$ increases, the conditional average queue length of the queue in Phase 2 increases but elasticity $p_{\text{normal}}$ initially increases and then slightly decreases after one value. However, for $\lambda \in [34, 65]$ (i.e., $\rho = \lambda \sigma_0^{-1} \geq 1$), after adaptively adjusting the service rate value of the controller as $\tilde{\mu}_B$ in the queueing system, we obtain $\rho = \lambda (\tilde{\mu}_B)^{-1} < 1$. Thus, when the conditional average queue length of the queue in Phase 2 increases, elasticity $p_{\text{normal}}$, which is denoted by the asterisk, still increases and is the same as that before the adaptive adjustment of the parameters in the system. Therefore, the system can maintain elastic invariance in adaptive adjustment parameters when the average arrival rate of the system exceeds the average service rate.

D. Comparison and Analysis

In [15], the models for calculating probabilities $p_{\text{over}}, p_{\text{normal}}$, and $p_{\text{under}}$ are expressed by (4), (5), and (6), respectively. We exhibit that the probabilities using the models in [15] are influenced by varying the values of $\lambda$ from 5.0 to 34.0, as shown in Fig. 12. As the average arrival rate $\lambda$ continues to increase, $\lambda$ is greater than or equal to $c\mu_B$ at some cases, where $\mu_B = 3.4$. Therefore, the system adaptively adjusts the controller service rate $\mu_B$ in accordance with Algorithm 1. Under the variation in arrival rate $\lambda$ from 34 to 65 and the change in service rate $\mu_B$, the plot of $p_{\text{over}}, p_{\text{normal}}$, and $p_{\text{under}}$ versus $\lambda$ is shown in Fig. 13.

As shown in Fig. 12, as $\lambda$ increases, the volatility of probability $p_{\text{normal}}$ is greater than that shown in Fig. 6 and that the probability value is larger than that shown in Fig. 6. Similarly, the comparison of Figs. 10 and 13 shows that the
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 11. Plot of $\text{AvgQL}_{\text{normal}}$ versus $p_{\text{normal}}$ in the normal state for $\lambda \in [1, 34) \cup [34, 65]$.

Fig. 12. Plot of $p_{\text{over}}$, $p_{\text{normal}}$, and $p_{\text{under}}$ versus $\lambda$ using models in [15].

Fig. 13. Plot of $p_{\text{over}}$, $p_{\text{normal}}$, and $p_{\text{under}}$ versus $\lambda$ while $\mu_B$ changes accordingly using models in [15].

probability $p_{\text{normal}}$ in Fig. 13 is larger than that in Fig. 10. Moreover, as $\lambda$ increases, $p_{\text{normal}}$ initially increases and then decreases (Fig. 13), whereas $p_{\text{normal}}$ continues to increase steadily (Fig. 10). Our elastic model is therefore in line with the actual CPS.

We compare the $N/(d,c)\cdot M/M/c/K/AMWV$ queueing system proposed in this paper with the $N/(d,c)\cdot M/M/c/K/SMWV$ queueing system in [17] by using the cost function $F$ and the nominal time delay function $\tau_{\text{nom}}$. In the current study, $F$ and $\tau_{\text{nom}}$ are given by (54) and (52), respectively. We use the values of the system parameters in [17] as the input of the two models. The results of the comparison are shown in Table VIII. From Table VIII, $F$ and $\tau_{\text{nom}}$ of the queue model in this paper are less than those in [17], respectively. The results show that our model exhibits low cost and high performance.

### VII. Conclusion

In this paper, we considered resource provisioning under uncertain workload condition on a special CPS. In the shared-bus DESs, an on-demand nature of time-related resource provisioning existed. The anelasticity naturally reflected the on-demand nature with the hysteresis characteristic between stress and strain. We proposed a novel, quantitative, and formal definition of anelasticity model for the special CPS. The optimal values of various parameters in such a system under stability condition were also solved. We established the FSDMC as an $N/(d,c)\cdot M/M/c/K/AMWV$ queue model to minimize the CPR of the system. Then, we obtained the respective service rates of three states, namely, over-provisioning, normal, and under-provisioning states. An $M/M/1/K$ queue model in a Markovian environment was developed to calculate the anelasticity value of the FSDMC platform accurately. When the average arrival rate exceeded the average service rate in the queueing system, an adaptive adjustment algorithm based on PSO was proposed to adjust the controller service rate dynamically while maintaining the minimum value of the system CPR unchanged under stability condition.

### References


Jiyao An (M’12) received the M.Sc. degree in mathematics from Xiangtan University, Xiangtan, China, in 1998 and the Ph.D. degree in mechanical engineering from Hunan University, Changsha, China, in 2012.

He was a Visiting Scholar with the Department of Applied Mathematics, University of Waterloo, Waterloo, ON, Canada, from 2013 to 2014. In 2000, he joined the College of Computer Science and Electronic Engineering, Hunan University, where he is currently a Full Professor. He has published over 50 papers in international and domestic journals and refereed conference papers. His current research interests include cyber-physical systems, Takagi–Sugeno fuzzy systems, parallel and distributed computing, and computational intelligence.

Dr. An is a member of the ACM and a Senior Member of CCF. He is an active reviewer of international journals.

Yang Bai received the B.S. degree in communication engineering and M.S. degree in computer technology from Hunan University, Changsha, China, in 2013 and 2016, respectively, where she is currently pursuing the Ph.D. degree in computer science and technology with the Key Laboratory for Embedded and Network Computing of Hunan Province.

Her current research interests include service computing, embedded systems, and cyber-physical systems.

Keqin Li (M’90–SM’96–F’15) received the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is a Distinguished Professor of computer science with the State University of New York, New Paltz, NY, USA. He has published over 570 journal articles, book chapters, and refereed conference papers. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU–GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of Things, and cyber-physical systems.

Mr. Li was a recipient of several best paper awards. He is currently serving or has served on the editorial boards for the IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, IEEE Transactions on Services Computing, and IEEE Transactions on Sustainable Computing.