Scheduling Algorithms of Flat Semi-Dormant Multicontrollers for a Cyber-Physical System

Hongfang Gong, Renfa Li, *Senior Member, IEEE*, Jiyao An, *Member, IEEE*, Weiwei Chen, and Keqin Li, *Fellow, IEEE*

***Abstract*—Recently, the modeling and design of distributed controllers in cyber-physical systems (CPSs), which suffer from messages lost, delay variation, and jitter, has gained lots of research attentions. A special CPS, arbitrated networked control system (ANCS), has been designed for scheduling or arbitrating networks in a control system. In this paper, we propose a novel ANCS with dual communication channels. The proposed ANCS uses a hierarchical flexible time-division multiple access (TDMA)/fixed priority scheduling policy that is based on the event trigger protocol. A flat semi-dormant multicontrollers (FSDMC) model is developed for the proposed ANCS. We then model the FSDMC as an *N/(d,c)-M/M/c/K/SMWV* queue, and obtain various performance indices. Based on the model, a multiobjective optimization problem is then formulated to minimize the nonlinear energy consumption function and the nominal delay function presented in this study. To resolve the multiobjective optimization problem, a scheduling algorithm based on the multiobjective particle swarm optimization algorithm is proposed to generate the Pareto front and the corresponding nondominated vector sets. An optimal stopping algorithm is also designed to obtain the optimal value of the number of semi-dormant controllers. The optimal values of various parameters of the control system are obtained by using the above nondominated vector sets, and are applied to the proposed ANCS. Extensive numerical results are provided to illustrate the usefulness of the proposed algorithms and the effects of the control system parameters on the optimal policy.**

***Index Terms*—Control and architecture codesign, cost and performance evaluation, cyber-physical system (CPS), queuing system, semi-dormant controller clusters.**

I. Introduction

Automotive, avionics, and industrial automation systems are often distributed embedded systems (DES) with a large number of processing units (PUs), sensors, and actuators that communicate via shared buses such as controller area network (CAN), local interconnect network (LIN), and FlexRay. Such architectures are used to run distributed control applications, often with multiple quality-of-control (QoC) constraints [1]. Inherent heterogeneity and diverse nature of such cyber-physical systems (CPSs) make these systems very complex. The continually increasing data also bring about the information unbalanced problem in various application domains [2]. Such unbalanced load problem on control applications maybe lead to performance reduction and energy dissipation. In order to improve the system performance and reduce the energy consumption of system, it is necessary to consider new control design paradigms. Such design approaches need to accommodate architecture-level constraints, for example, the availability or cost of computation and communication resources [3], [4].

A special kind of CPS named as arbitrated networked control system (ANCS) is first proposed in [4] in order to emphasize that the control systems are to be designed for networks that are scheduled or arbitrated. In such a system, the question for the fixed delay generated by the time-triggered protocol is only investigated in [1], [3], [4], but the problem for varying delay not considered. Zeng *et al.* [5] investigated the problem of the electronic control unit (ECU) and FlexRay bus optimal scheduling synthesis from the perspective of the application designer. Dvorak and Hanzalek [6] proposed a heuristic algorithm that decomposes the scheduling problem to the ECU-to-channel assignment subproblem and the channel scheduling subproblem by two independent channels interaction with gateway. Hu *et al.* [7] studied a holistic scheduling problem for handling real-time applications in time-triggered invehicle networks, where practical aspects in system design and integration are captured. Kang *et al.* [8] presented a frame-packing algorithm called bandwidth consumption minimizing with various periodic signals to pack the signals with different periods into a message frame. However, these scheduling efforts are targeted only to the static segment.

Manuscript received August 9, 2016; revised December 5, 2016; accepted March 24, 2017. Date of publication April 4, 2017; date of current version August 1, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61173036, Grant 61370097, Grant 61370095, and Grant 11371074. Paper no. TII-16-0829. *(Corresponding author: Renfa Li.)*

H. Gong is with the College of Computer Science and Electronic Engineering, the Key Laboratory for Embedded and Network Computing of Hunan Province, and the National Supercomputing Center in Changsha, Hunan University, Changsa 410082, China, and also with the School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha 410114, China (e-mail: ghongfang@126.com).

R. Li, J. An, and W. Chen are with the College of Computer Science and Electronic Engineering, the Key Laboratory for Embedded and Network Computing of Hunan Province, and the National Supercomputing Center in Changsha, Hunan University, Changsa 410082, China (e-mail: lirenfa@hnu.edu.cn; jt_anbob@hnu.edu.cn; chen.ava.0012@gmail.com).

K. Li is with the College of Computer Science and Electronic Engineering, the Key Laboratory for Embedded and Network Computing of Hunan Province, and the National Supercomputing Center in Changsha, Hunan University, Changsa 410082, China, and also with the Department of Computer Science, State University of New York, New York, NY 12561 USA (e-mail: lik@newpaltz.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2690939

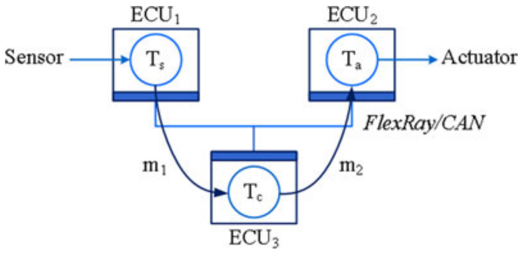


Fig. 1. Typical distributed embedded system (DES) [1].

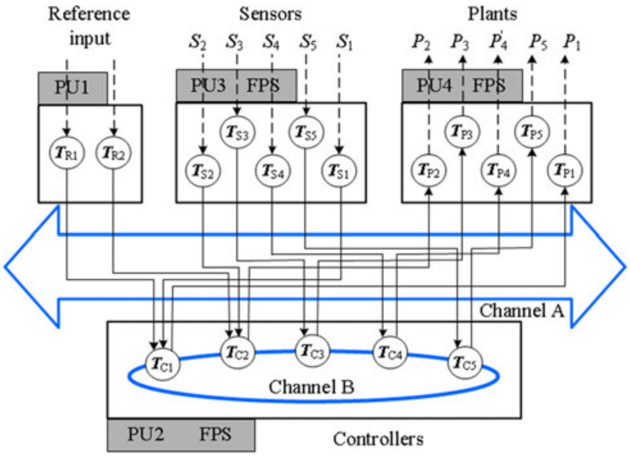


Fig. 2. System architecture with five distributed control applications.

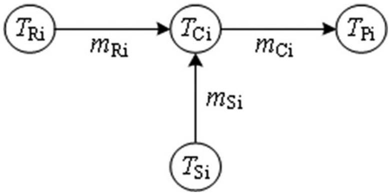


Fig. 3. Task graph for control application i, $i = 1, 2$.

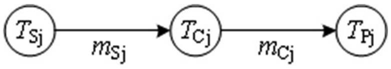


Fig. 4. Task graph for control application j, $j = 3, 4, 5$.

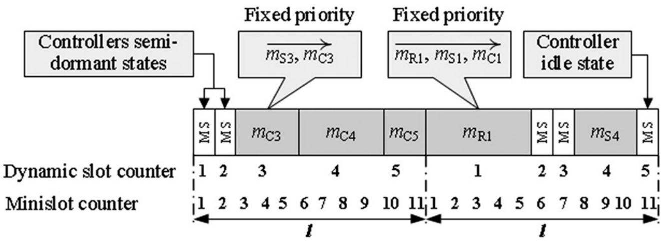

Fixed priority
Fixed priority
Controllers semi-dormant states
mS3, mC3
mR1, mS1, mC1
Controller idle state
MS MS mC3 mC4 mC5 mR1 MS MS mS4 MS
Dynamic slot counter 1 2 3 4 5 1 2 3 4 5
Minislot counter 1 2 3 4 5 6 7 8 9 10 11 1 2 3 4 5 6 7 8 9 10 11
l
l

Fig. 5. Communication channel A: Hierarchical FTDMA/FP scheduler, $l =$ FTDMA cycle length.

It is widely believed that as application complexity and hence communication requirements continue to grow, the bandwidth of the time-triggered segment will not suffice and a purely time-triggered implementation might be overly expensive. On the other hand, priority-driven event-triggered (ET) implementations suffer from the usual temporal nondeterminism which results in poor control performance [9]. Zeng *et al.* [10] used statistical analysis methods to compute the probability distribution of CAN message response times when only partial information is available about the functionality and architecture of a vehicle. However, the tradeoff between system implementation cost and system performance was not studied. In this paper, we employ the queueing theory to model DES based on FlexRay network with ET protocol, and compute stationary probability distribution of system and system performance metrics.

In [11], the dormant multicontroller model is presented for software-defined networking, where network control is decoupled from forwarding [12]. In this paper, we propose a flat semi-dormant multicontrollers (FSDMC) model for a special CPS based on hierarchical modeling theory. The FSDMC is based on a wakeup mechanism on the communication system (e.g., FlexRay, see [13, Fig. 2]). FlexRay supports a standby and dormant modes for bus drivers (BD) of an ECU. Wakeup symbol is used to bring the multinode clusters out of the low-power state by sending a pattern that causes the BDs to wake up the local ECU. The wakeup signals distributed on the network are supported by communication controllers and BD devices [14]. We consider a typical distributed embedded architecture shown in–Fig. 1 (see [1, Fig. 1(a)]).

FlexRay networking supports single-channel configuration, as well as the dual-channel one with some nodes connected to both channels and others connected to just one channel [13]. We extend the above-mentioned ANCS to a hybrid bus-star topology with two channels (see [14, Figs. 1–10]). Here, channel **A** is implemented as a bus topology connection, while channel **B** is implemented as a ring topology connection. All nodes are connected to channel **A**, but only controller nodes are interconnected via channel **B**. These controllers constitute a flat multicontrollers structure using equal mode, that is, the servers are equally likely, in order to maintain load balancing and guarantee global consistency. In this paper, with the purpose of a good tradeoff between system implementation cost and system performance, we focus on the scheduling issues of all controllers. We apply the queueing theory to the FSDMC model for the ANCS, and dynamically configure the system parameters using the optimal stopping strategy [15] and multiobjective particle swarm optimization (MOPSO) [16] algorithm.

The highlights of this paper are summarized as follows.

1) A novel ANCS architecture is proposed. An FSDMC model is presented to allow part of the idle controllers to enter the semi-dormant state under light traffic case by building a queuing system.

2) We consider a finite capacity *M/M/c* queuing system with *N*-policy and synchronous multiple working vacations (SMWV) of partial servers, *N/(d,c)-M/M/c/K/SMWV* for short, where c represents the number of controllers, d denotes the number of semi-dormant controllers, and K describes the system capacity. We use this queuing system to quantify various performance indices of the system and analyze the system cost and performance.

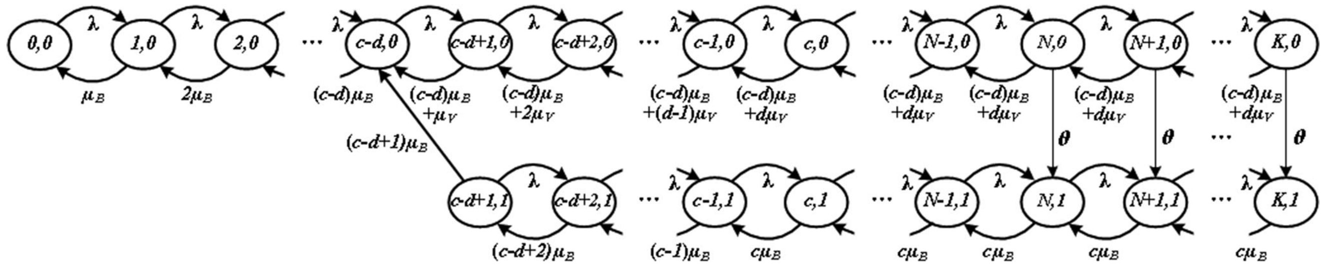
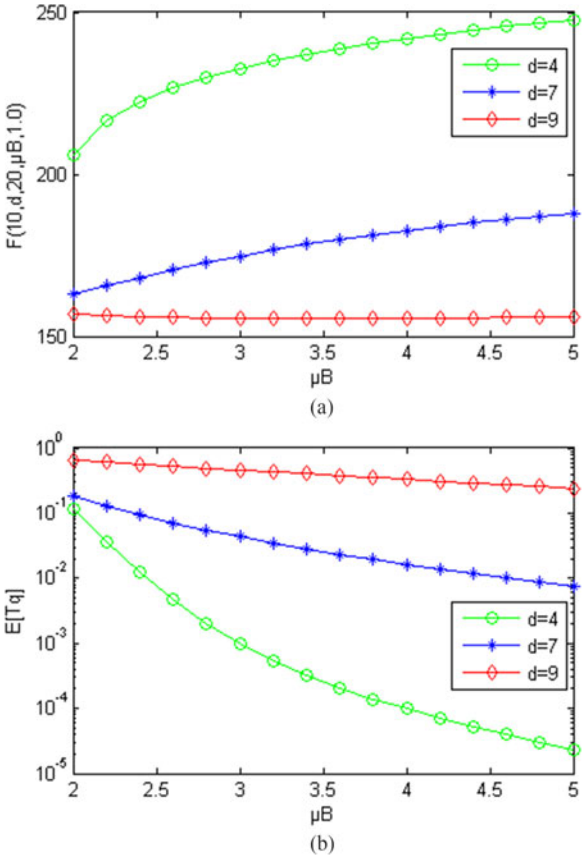


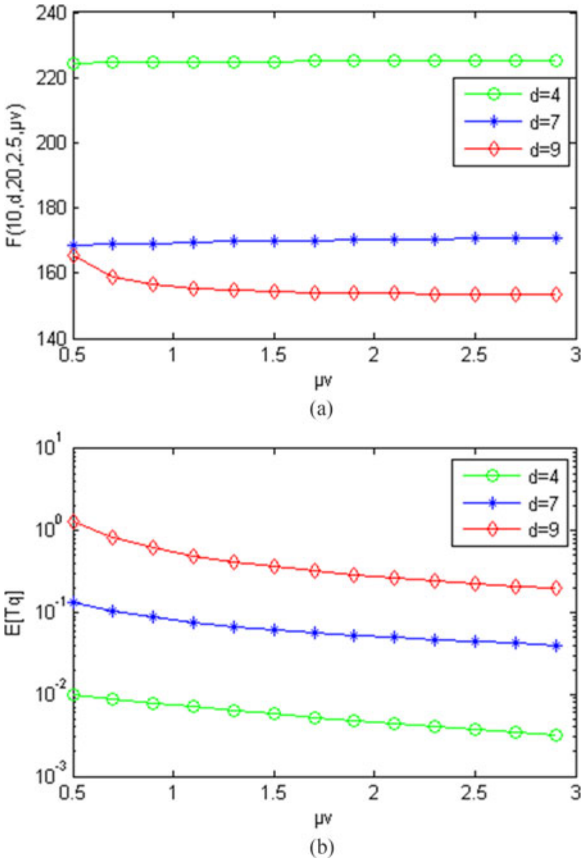
Fig. 6. State-transition-rate diagram for the *N/(d,c)-M/M/c/K/SMWV* queuing system.



(a)

(b)

Fig. 7. Plot (a), (b) of $F(10, d, 20, \mu_B, 1.0)$ and $E[T_q]$ versus μ_B and d, respectively.



(a)

(b)

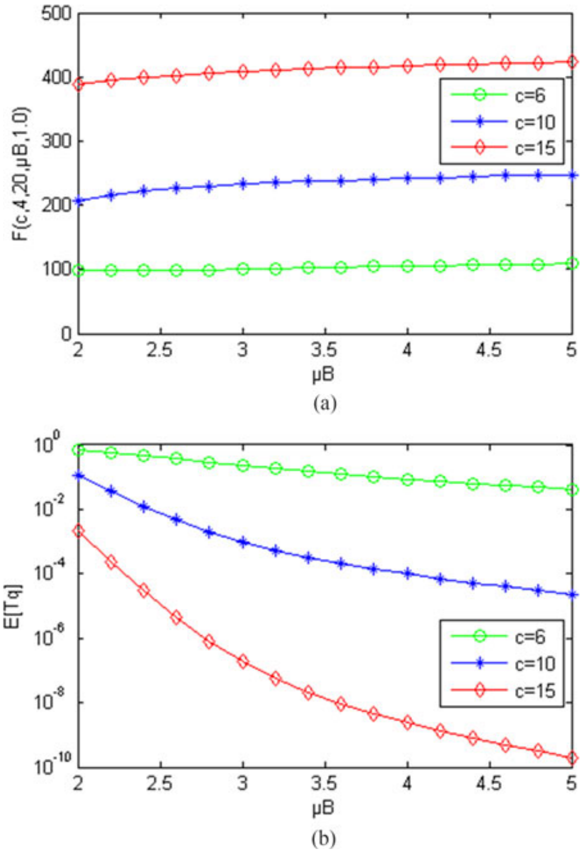
Fig. 8. Plot (a), (b) of $F(10, d, 20, 2.5, \mu_V)$ and $E[T_q]$ versus μ_V and d, respectively.

3) We establish a multiobjective optimization model to minimize the nonlinear energy consumption function and the nominal delay function. The MOPSO algorithm is employed to generate the Pareto front and the corresponding nondominated vector sets. We propose an optimal stopping algorithm and a scheduling algorithm of FSDMC to derive the optimal value of the number of semi-dormant controllers and the optimal values of the decision variables, respectively. These parameters values can be employed to design our proposed ANCS. To the best of our knowledge, there has been no such research in ANCS till now.

The rest of this paper is organized as follows. In Section II, we describe the related work. Section III presents the platform architecture of system and depicts the system dynamics. In Section IV, we model the queuing system as a quasi-birth-and-death (QBD) process and compute the stationary distributions and the system performance measures by using matrix-geometric method based on the Markovian model. Section V establishes the expected cost function per unit time and the performance measure function, and designs an optimal stopping algorithm and a scheduling algorithm of FSDMC. In Section VI, we provide numerical results for sensitivity analysis of system cost and performance, and solve the optimal values of system parameters. Finally, Section VII concludes the paper.

II. Related Works

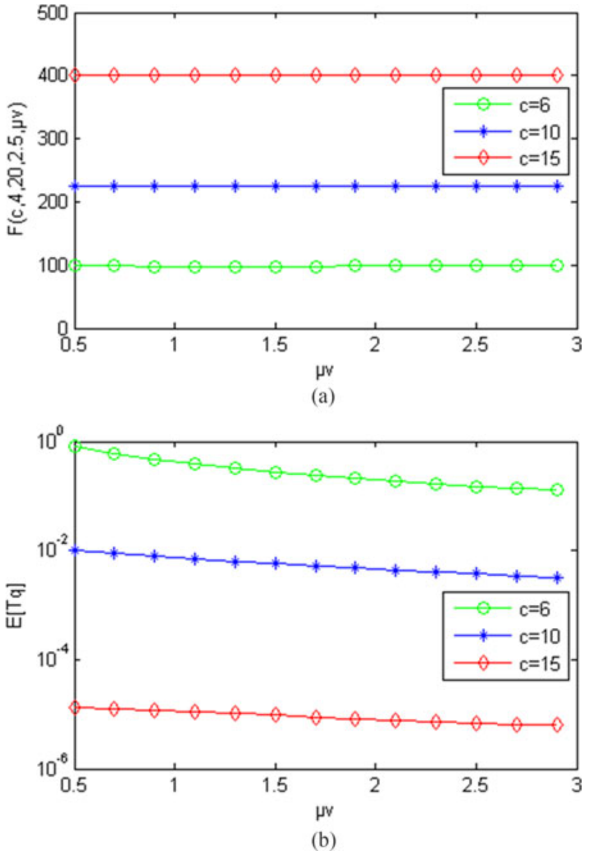
How to enable the CPS nodes to efficiently collaborate so as to accomplish more computing tasks is a very challenging problem [17]. The codesign of control and architecture can



(a)

(b)

Fig. 9. Plot (a), (b) of $F(c, 4, 20, \mu_B, 1.0)$ and $E[T_q]$ versus μ_B and c, respectively.



(a)

(b)

Fig. 10. Plot (a), (b) of $F(c, 4, 20, 2.5, \mu_V)$ and $E[T_q]$ versus μ_V and c, respectively.

allow the system to adapt to the changes in the environment and workload, or even to the changes in system architecture caused by reconfigurations or failures. The result could be used to improve system performance and higher resource utilization [9], [18]. Connections between control performance and architecture design have been explored in the past two decades [1], [3], [4], [19]– [29]. The idea in most of those papers is that better control performance is achieved by redesigning the architecture in a suitable manner. In [19] and [20], a closed-loop state-based strategy is used to determine a aperiodic sampling sequence and a corresponding scheduling strategy is used to ensure the desired control performance. In [21] and [22], a closed-loop output-based feedback strategy is proposed to guarantee stability and desired performance for ET control systems using tradeoffs between closed-loop performance and communication load. Other event-based strategies are used in [23] and [24] so as to determine the sampling sequence to guarantee system stability. In [25] and [26], an event-based control and scheduling codesign strategy for networked embedded control systems is presented to improve the control performance provided that the limited resources are used efficiently. In [27], an automotive distributed architecture based on a high speed power line communication protocol is designed using a transaction level modeling approach for networked embedded systems. Kauer *et al.* [28] deals with synthesis of distributed embedded control systems closed over a faulty or severely constrained communication network. A distributed embedded architecture is modeled as a network of Event Count Automata to verify the bound on deadline misses. A robust stabilizing controllers design is also given by introducing a novel fault-tolerant control strategy for improving QoC. In [29], a framework for the ET stabilization of nonlinear systems is proposed using hybrid system tools, and it is especially well suited for embedded systems and networked control systems since it is able to reduce the amount of communication traffic and computation resources needed for control. However, in these papers, the arbitration of messages on a shared communication medium is not involved.

Voit [3] presented several approaches to deal with the integrated co-design of control and communication for a particular kind of ANCS, which employs hierarchical and hybrid communication networks. Annaswamy *et al.* [1] considered whether or not a constant delay value is smaller or larger than the sampling period for ANCSs and proposed a control design that explicitly makes use of the delay value, thereby ensuring a stable closed-loop system. Cao *et al.* [30] proposed a joint optimization framework that online optimizes the control intervals with codesign of control and communication using simulated annealing algorithm for networked CPSs. In [31], another special class of CPS named power-junction-network (PJN) was first proposed to allow a single controller to accurately control the position of multiple plants even if the dynamics of the plants are different for wireless control networks. In [32] and [33], both averaging

PJN and consensus PJN were studied to make it possible to allow ***m*** strictly output passive digital-controllers to control up to ***n-m*** strictly output passive continuous- or discrete-time-plants so that L_2^m- or l_2^m-stability was guaranteed. Kottenstette *et al.* [34] presented a resilient PJN for addressing the question how complex dynamic plants are to be controlled safely and reliably when a control system is under a cyber-attack. However, all of the work above does not consider the problems of unbalanced load.

In queueing system, the *N*-policy, which is used to control the service performance, is a threshold policy introduced by Yadin and Naor [35]. In [36], the *N*-policy *M/G/1* queueing system with a reliable server was first studied, and the *N*-policy was proved to be the optimal policy for deriving various cost functions for finite or infinite horizon problems in the system. Chang and Pearn [37] investigated an infinite capacity *N*-policy *M/G/1* queueing system with a single removable server station. Some research work also focused on the multiple servers queuing system with single or multiple vacations policy. Zhang and Tian [38] and Xu and Zhang [39] analyzed the *M/M/c* vacation systems with a *partial server multiple vacation policy* in which some servers take single or multiple vacations. Ke *et al.* [40] considered a finite buffer *M/M/c* queueing system in which servers are unreliable and follow a (*d*, *c*) vacation policy. However, queueing models with single or multiple working vacations (WVs) are more realistic in real-world situations.

Servi and Finn [41] considered a server which works at different rates rather than completely stops during the vacation period. Such a vacation is called a WV. Zhang and Xu [42] investigated an *M/M/1* queue with multiple WVs and the *N*-policy using a QBD process and a matrix-geometric solution method. With a similar approach, Lin and Ke [43] considered an *M/M/R* queue with a single WV and exhaustive service. Yang and Wu [44] investigated the *N*-policy *M/M/1* queueing system with multiple WV and server breakdowns. Jain and Upadhyaya [45] discussed modeling and analysis problem of unreliable Markovian multiserver finite-buffer queue with discouragement and synchronous WV policy. However, the existing research work, including those mentioned above, have not addressed the optimization issue in the finite buffer multiple-server queuing systems with *N*-policy and (*d*, *c*) WV mechanism.

III. Platform Architecture

In order to illustrate an FSDMC model for the ANCS in this paper, we still consider the typical distributed embedded architecture proposed in [1] (see–Fig. 1), where an ECU collects sensor data (denoted as a task T_s). A communication bus (e.g., FlexRay) then transmits the data as message m_1 to a second ECU (marked as ECU$_3$), where the resident control algorithm is implemented (denoted as task T_c). The output of the controller is then sent as a message m_2 to the actuator in ECU$_2$, which activates the actuator task T_a.

We consider a platform architecture with multiple distributed control applications (see Fig. 2). The control applications are partitioned into a number of tasks that are mapped onto different PUs. All PUs communicate via a shared communication channel **A** and run different tasks from one or more control applications. However, only all the controller nodes of the PUs are interconnected via channel **B** to maintain load balancing and guarantees global consistency. In our architecture, PU1 hosts the tasks responsible for reading the reference command from the user, PU2 hosts all tasks that compute control commands, PU3 hosts the tasks responsible for reading sensors, and the tasks responsible for providing plant commands are mapped onto PU4. This setup will form the basis of the techniques we propose. In this paper, we view such controller clusters as an FSDMC model, and focus on the stochastic scheduling problem of multicontrollers in this model.

In the platform architecture mentioned above, each controller is connected to an actuator and a sensor, and handles tasks that are sent by the corresponding sensor (or user) via the shared communication channel **A**. When a controller is overloaded, it will forward partial tasks to other semi-dormant controllers for processing via the channel **B** using token ring strategy. After a task is handled by a semi-dormant controller, it will be transmitted to the corresponding overloaded controller via the channel **B**. Then, the overloaded controller will send this handled task to the corresponding actuator via the channel **A**. We consider five control applications that are divided into two classes: one involves *Controller applications 1* and *2* shown in Fig. 3 (see [1, Fig. 3]), which are partitioned into four tasks: T_{Ri}, T_{Si}, T_{Ci}, and T_{Pi}, $i = 1, 2$; the other contains *Controller applications 3, 4,* and *5* shown in–Fig. 4 (see [1, Fig. 4]), which are partitioned into three tasks: T_{Sj}, T_{Cj}, and T_{Pj}, $j =$ 3, 4, 5. Task T_{Si} reads data from sensor S_i and sends sensor signal m_{Si} to the task T_{Ci} via channel **A**. Similarly, task T_{Ri} reads the reference command m_{Ri} from the user and also sends it to task T_{Ci} via channel **A**. The control input is computed in T_{Ci} using the messages m_{Ri} and m_{Si}. The processed output of task T_{Ci} is sent via channel **A** to task T_{Pi}. The plant P_i receives the control input from the task T_{Pi}, and so forth.

The shared communication channel **A** follows a hierarchical flexible TDMA/fixed priority (FTDMA/FP) [46] bus scheduling policy with the ET protocol shown in–Fig. 5. The FTDMA technique was conceived primarily to overcome the limitations of CAN and, in particular, for supporting high-performance real-time communications [47]. An FTDMA scheduler runs on the top-level of the scheduler, that is, the communication bandwidth is divided into equal cycles with length l. All the messages coming into/from the control application i ($i =$ 1, 2, 3, 4, 5) which are transmitted via channel **A** in dynamic segment slot i follow a fixed priority scheduler. If no message is to be sent during a certain slot k, slot k will have a very small length called *minislot*; otherwise the slot k will have a length equal to the number of minislots needed for transmitting the whole message. This can be seen in Fig. 5. The slot 5 has two minislots (10 and 11) in the first bus cycle when message m_{C5} is transmitted. However, in the second bus cycle, the slot 5 has only one minislot (denoted with "MS") when no message is sent in slot 5. When a controller is in a state of semi-dormant or idle, the corresponding slot is represented as MS, for example, controllers 2 and 3 are in the state of semi-dormant in second bus cycle.

With the purpose of a good tradeoff between system implementation cost and system performance, we establish the FSDMC model as a queue to compute various performance indices of the system in the following sections. According to the performance indices, a multiobjective optimization model is proposed to minimize the nonlinear cost function and the nominal delay function, and to obtain the optimal values of system parameters.

IV. Queueing Model

A. Model Description

In this section, we propose a finite capacity *M/M/c* queuing system with *N*-policy and SMWV of partial servers, *N/(d,c)-M/M/c/K/SMWV* for short. The queuing system is used to analyze the FSDMC model. For convenience, we refer to a packet-in message as a *customer*, a controller as a *server*, and a semi-dormant state as a *WC state*.

In the FSDMC model, for each control application, a corresponding subqueue is generated by the sensor's sampling data and the reference command. The c subqueues are respectively derived by c control applications. The arbitration mechanism employed by CAN means that messages are sent as if all the nodes in the network share a single global priority-based queue [48]. Therefore, the c subqueues constitute a single global queue for the sharing bus. We assume that the packets from a sensor arrive at the corresponding controller in accordance with a Poisson process. It is assumed that the arrival process of the packets from the user is a Poisson process as well (e.g., in control application 1 and 2). The sum of independent Poisson processes is also a Poisson process. Hence, we assume that for each control application, the packets arrive at the corresponding controller with a Poisson process having rate λ_i ($\lambda_i > 0$). Each packet, upon arrival, goes directly into service if the controller is free; otherwise, the packet is added to the queue. When the controller finishes processing a packet, the packet is transmitted to the corresponding actuator, and the next packet in line, if there is any, enters the controller. It is obvious that the interarrival time in the single global queue is an exponential random variable with parameter λ, and $\lambda = \sum_{i=1}^{c} \lambda_i$.

It is assumed that the service time during a normal busy period is an exponential random variable with the mean service rate μ_B. The service time during a WC follows another exponential distribution with the mean service rate μ_V ($\mu_V < \mu_B$). We assume that the threshold of the ith ($i = 1, 2, \cdots, c$) subqueue is N_i (e.g., $N_i = \lfloor \frac{N}{c-d} \rfloor$), the capacity is K_i (e.g., $K_i = \lceil \frac{K}{c-d} \rceil$), but, in general, $N \neq \sum_{i=1}^{c} N_i$ and $K \neq \sum_{i=1}^{c} K_i$. It is assumed that the number of customers in the ith subqueue system is η_i. Let $\psi = \sum_{i=1}^{c} \eta_i$. If $\psi \leq c - d$, the d idle servers start to take WCs synchronously, where vacation time follows exponential distribution with rate θ. When vacation has been completed, if $\psi < N$, then d vacation servers all take another WCs; otherwise, the d vacation servers return from the WCs, and proceed to serve the waiting customers with service rate μ_B until $\psi \leq c - d$. If $\psi = K$, no customer is allowed to enter any of the subqueues. In the ith subqueue system, if $\eta_i \geq N_i$, then the ith server is in an overloaded state. It will transmit packet-in messages by channel **B** to d vacation servers for processing with rate μ_V.

In general periodic real-time systems, the periodic tasks are considered as a set of mutually independent tasks. Each task gives rise to an infinite sequence of jobs. Each job is assumed to be independent of other jobs of the same task and those of other tasks [49]. Lehoczky [50], [51] investigated real-time system scheduling problems using real-time queueing theory with Poisson arrivals and exponential service times. Hence, in our work, all the interarrival time, service time, and vacation time are assumed to be mutually independent. We further assume that arriving customers are served based on the first-come, first-served discipline. Suppose that a server can serve only one customer at a time.

B. QBD Process Model

Let $L_v(t)$ be the number of messages in the system at time t and let

$$J(t) = \begin{cases} 0, & c-d\,\text{controllers on the working period and} \\ & d\,\text{controllers on the semi-dormant period,} \\ 1, & c\text{controllers on the working period.} \end{cases}$$

Then ,$\{L_v(t), J(t)\}$ is a QBD process with the state space

$$\Omega = \{(k,0) : 0 \leq k \leq c-d\} \cup \{(k,j) : c-d < k \leq K, \\ j = 0,1\}.$$

$$Q = \begin{pmatrix}
A_0 & C_0 & & & & & & & & & \\
B_1 & A_1 & C_1 & & & & & & & & \\
 & \ddots & \ddots & \ddots & & & & & & & \\
 & & B_c & A_c & C_c & & & & & & \\
 & & & B_{c+1} & A_{c+1} & C_{c+1} & & & & & \\
 & & & & \ddots & \ddots & \ddots & & & & \\
 & & & & & B_{N-1} & A_{N-1} & C_{N-1} & & & \\
 & & & & & & B & A & C & & \\
 & & & & & & & \ddots & \ddots & \ddots & \\
 & & & & & & & & B & A & C \\
 & & & & & & & & & B_K & A_K
\end{pmatrix}.$$

The state-transition-rate diagram for the queuing system is shown in Fig. 6 .

Using the lexicographical sequence for the states, the infinitesimal generator of the process can be expressed as

We let $\sigma_j = (c-d)\mu_B + j\mu_V$, $\delta_j = (c-d+j)\mu_B$, $0 \le j \le d$, then the submatrices in Q are written as follows:

$$A_k = \begin{cases} -(\lambda + \delta_{k-c+d}), 0 \le k \le c-d, \\ \begin{pmatrix} -(\lambda+\sigma_{k-c+d}) & 0 \\ 0 & -(\lambda+\delta_{k-c+d}) \end{pmatrix}, \\ c-d < k \le c-1, \\ \begin{pmatrix} -(\lambda+\sigma_d) & 0 \\ 0 & -(\lambda+\delta_d) \end{pmatrix}, c \le k \le N-1, \\ \begin{pmatrix} -(\sigma_d+\theta) & \theta \\ 0 & -\delta_d \end{pmatrix}, k = K, \end{cases}$$

$$B_k = \begin{cases} \delta_{k-c+d}, 1 \le k \le c-d, \\ \begin{pmatrix} \sigma_1 \\ \delta_1 \end{pmatrix}, k = c-d+1, \\ \begin{pmatrix} \sigma_{k-c+d} & 0 \\ 0 & \delta_{k-c+d} \end{pmatrix}, c-d+2 \le k \le c-1, \\ \begin{pmatrix} \sigma_d & 0 \\ 0 & \delta_d \end{pmatrix}, c \le k \le N-1, \end{cases}$$

$$C_k = \begin{cases} \lambda, 0 \le k < c-d, \\ \begin{pmatrix} \lambda & 0 \end{pmatrix}, k = c-d, \\ \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}, c-d < k \le N-1 \end{cases}$$

and

$$A = \begin{pmatrix} -(\lambda+\sigma_d+\theta) & \theta \\ 0 & -(\lambda+\delta_d) \end{pmatrix}, N \le k < K,$$

$$B = \begin{pmatrix} \sigma_d & 0 \\ 0 & \delta_d \end{pmatrix}, \ N \le k \le K,$$

$$C = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}, \ N \le k < K.$$

To analyze this QBD process, it is necessary to find the minimal nonnegative solution of the matrix quadratic equation

$$R^2 B + RA + C = 0 \tag{1}$$

and this solution R is called the rate matrix [52].

Theorem 1: If $\rho = \lambda \delta_d^{-1} < 1$, the matrix (1) has the minimal nonnegative solution

$$R = \begin{pmatrix} r & \dfrac{\theta r}{\delta_d(1-r)} \\ 0 & \rho \end{pmatrix} \tag{2}$$

where $r = \frac{1}{2\sigma_d}(\lambda + \sigma_d + \theta - \sqrt{\Delta})$, and r satisfies the quadratic (3) and the relationship (4):

$$\sigma_d z^2 - (\lambda + \sigma_d + \theta) z + \lambda = 0, \tag{3}$$

$$\lambda + \theta + \sigma_d(1-r) = \sigma_d + \frac{\theta}{1 \, - \, r} = \frac{\lambda}{r} \tag{4}$$

where $\Delta = (\lambda + \sigma_d + \theta)^2 - 4\lambda\sigma_d$, and $0 < r < 1$.

Proof: Assume that another real root of the quadratic (3) is r^*. We have

$$r, r^* = \frac{1}{2\sigma_d}(\lambda + \sigma_d + \theta \pm \sqrt{\Delta}).$$

Obviously, the following inequalities hold:

$$(\lambda - \sigma_d + \theta)^2 < \ \Delta < (\lambda + \sigma_d + \theta)^2, \lambda > \sigma_d$$
$$(\sigma_d - \lambda + \theta)^2 < \ \Delta < (\lambda + \sigma_d + \theta)^2, \lambda < \sigma_d.$$

We can obtain $0 < r < 1$, $r^* > 1$, and $r = \frac{1}{2\sigma_d}(\lambda + \sigma_d + \theta - \sqrt{\Delta})$.

Substituting r into the (3), we can obtain (4).

Each of matrix A, B, and C is an upper triangular matrix. Let $R = \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}$.

Substituting R into the matrix (1), we have

$$\begin{cases} \sigma_d r_{11}^2 - (\lambda + \sigma_d + \theta) r_{11} + \lambda = 0, \\ \delta_d r_{22}^2 - (\lambda + \delta_d) r_{22} + \lambda = 0, \\ \delta_d r_{12}(r_{11} + r_{22}) + \theta r_{11} - (\lambda + \delta_d) r_{12} = 0. \end{cases} \tag{5}$$

To obtain the minimal nonnegative solution R of (1), we take $r_{11} = r$ to be the root of (3) over the interval (0, 1). In (5), let $r_{22} = \rho$. Substituting r_{11}, r_{22} into the third equation in (5), it is easy to obtain r_{12} and (2). Obviously, $SP(R) = \max(r, \rho) < 1$ if and only if $\rho < 1$. ■

C. Stationary Probability Distribution of System

If $\rho < 1$, let (L_v, J) be the stationary limit of the QBD process $\{L_v(t), J(t)\}$. Let

$$\pi_{k,j} = \lim_{t \to \infty} P\{L_v(t) = k, J(t) = j\}, \ \ (k,j) \in \Omega,$$
$$\pi_k = \pi_{k,0}, \ \ 0 \le k \le c-d,$$
$$\pi_k = (\pi_{k,0}, \pi_{k,1}), \ \ c-d < k \le K.$$

Theorem 2: If $\rho < 1$, the stationary probability distribution of (L_v, J) is

$$\pi_{k,0} = \begin{cases} G\frac{1}{k!}\left(\frac{\lambda}{\mu_B}\right)^{k}, 0 \le k \le c-d, \\ G\frac{1}{(c-d)!}\prod_{i=1}^{k-c+d}\frac{\lambda}{\sigma_i} - \frac{\theta r \pi_{c,0}}{\lambda(1-r)}\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c} \\ \quad \times \sum_{i=1}^{k-c+d}\left(\prod_{j=i}^{k-c+d}\frac{\lambda}{\sigma_j}\right), c-d+1 \le k \le c, \\ \pi_{c,0}\left(\frac{\lambda}{\sigma_d}\right)^{k-c}, c < k < N, \\ r\pi_{c,0}\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}, k = N. \end{cases} \tag{6}$$

$$\pi_{k,1} = \begin{cases} \pi_{c,0}\frac{\theta r}{\lambda(1-r)}\sum_{i=1}^{k-c+d}\left(\prod_{j=i}^{k-c+d}\frac{\lambda}{\delta_j}\right)\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}, \\ c-d+1 \le k \le c, \\ \pi_{c,1}\left(\frac{\lambda}{\delta_d}\right)^{k-c}, c < k < N, \\ \pi_{c,1}\left(\frac{\lambda}{\delta_d}\right)^{N-c} + \pi_{c,0}\frac{\theta r}{\delta_d(1-r)}\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}, \\ k = N \end{cases} \tag{7}$$

where

$$\pi_{c,0} = G\frac{\frac{1}{(c-d)!}\left(\frac{\lambda}{\mu_B}\right)^{c-d}\prod_{i=1}^{d}\frac{\lambda}{\sigma_i}}{1+\frac{\theta r}{\lambda(1-r)}\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}\sum_{i=1}^{d}\left(\prod_{j=i}^{d}\frac{\lambda}{\sigma_j}\right)}, \tag{8}$$

$$\pi_{c,1} = G\frac{\frac{1}{\lambda(c-d)!}\left(\frac{\lambda}{\mu_B}\right)^{c-d}\prod_{i=1}^{d}\frac{\lambda}{\sigma_i}\sum_{i=1}^{d}\left(\prod_{j=i}^{d}\frac{\lambda}{\delta_j}\right)}{\frac{1-r}{\theta r}\left(\frac{\lambda}{\sigma_d}\right)^{c+1-N}+\frac{1}{\lambda}\sum_{i=1}^{d}\left(\prod_{j=i}^{d}\frac{\lambda}{\sigma_j}\right)} \tag{9}$$

and, the constant factor G can be determined by the normalization condition.

Proof: The $(2N-c+d+1)$-dimensional row vector $(\pi_0, \pi_1, \pi_2, \cdots, \pi_N)$ satisfies the set of equations

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

where $B[R]$ is a $(2N-c+d+1) \times (2N-c+d+1)$ square matrix, and

$$B[R] = \begin{pmatrix} A_0 & C_0 & & & & & & \\ B_1 & A_1 & C_1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & B_c & A_c & C_c & & & \\ & & & B_{c+1} & A_{c+1} & C_{c+1} & & \\ & & & & \ddots & \ddots & \ddots & \\ & & & & & B_{N-1} & A_{N-1} & C_{N-1} \\ & & & & & & B & RB+A \end{pmatrix}.$$

Substituting $B[R]$ into the above equation, we obtain $2N-c+d+1$ linear equations:

$$-\lambda\pi_{0,0} + \mu_B\pi_{1,0} = 0 \tag{10}$$

$$\lambda\pi_{k-1,0} - (\lambda+\delta_{k-c+d})\pi_{k,0} + \delta_{k+1-c+d}\pi_{k+1,0} = 0, \quad 1 \le k < c-d \tag{11}$$

$$\lambda\pi_{c-d-1,0} - (\lambda+\sigma_0)\pi_{c-d,0} + \sigma_1\pi_{c-d+1,0} + \delta_1\pi_{c-d+1,1} = 0 \tag{12}$$

$$\lambda\pi_{k-1,0} - (\lambda+\sigma_{k-c+d})\pi_{k,0} + \sigma_{k+1-c+d}\pi_{k+1,0} = 0, \quad c-d+1 \le k < c \tag{13}$$

$$-(\lambda+\delta_1)\pi_{c-d+1,1} + \delta_2\pi_{c-d+2,1} = 0 \tag{14}$$

$$\lambda\pi_{k-1,1} - (\lambda+\delta_{k-c+d})\pi_{k,1} + \delta_{k+1-c+d}\pi_{k+1,1} = 0, \quad c-d+2 \le k < c \tag{15}$$

$$\lambda\pi_{k-1,0} - (\lambda+\sigma_d)\pi_{k,0} + \sigma_d\pi_{k+1,0} = 0, c \le k < N \tag{16}$$

$$\lambda\pi_{k-1,1} - (\lambda+\delta_d)\pi_{k,1} + \delta_d\pi_{k+1,1} = 0, c \le k < N \tag{17}$$

$$\lambda\pi_{N-1,0} - [\lambda+\theta+\sigma_d(1-r)]\pi_{N,0} = 0 \tag{18}$$

$$\lambda\pi_{N-1,1} + \frac{\theta}{1-r}\pi_{N,0} - \delta_d\pi_{N,1} = 0. \tag{19}$$

From (10) and (11), we have

$$\pi_{k,0} = \pi_{0,0}\frac{1}{k!}\left(\frac{\lambda}{\mu_B}\right)^{k}, 0 \le k \le c-d. \tag{20}$$

It is easy to prove that (16) and (17) can be satisfied by the following two expressions, respectively. They are expressed as

$$\pi_{k,0} = \pi_{c,0}\left(\frac{\lambda}{\sigma_d}\right)^{k-c}, \pi_{k,1} = \pi_{c,1}\left(\frac{\lambda}{\delta_d}\right)^{k-c}, c \le k < N. \tag{21}$$

From (4) and (18), we obtain

$$\pi_{N,0} = r\pi_{N-1,0} = r\pi_{c,0}\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}. \tag{22}$$

From (19), we have

$$\pi_{N,1} = \frac{\theta r}{\delta_d(1-r)}\pi_{N-1,0} + \frac{\lambda}{\delta_d}\pi_{N-1,1}. \tag{23}$$

Substituting (21) into the above (23), the third expression in (7) is yielded.

From (4), (16), and (22), we can obtain

$$\sigma_d\pi_{c,0} - \lambda\pi_{c-1,0} = \sigma_d\pi_{N,0} - \lambda\pi_{N-1,0} = -\frac{\theta}{1-r}\pi_{N,0}. \tag{24}$$

From (13) and (24), we have

$$\sigma_{k-c+d}\pi_{k,0} - \lambda\pi_{k-1,0} = -\frac{\theta}{1-r}\pi_{N,0}, c-d+1 \le k \le c. \tag{25}$$

Iterating over the above (25), the second expression in (6) is derived. Let $k = c$, we obtain (8).

In a similar way, from (14), (15), (17), and (19), we obtain

$$\delta_{k-c+d}\pi_{k,1} - \lambda\pi_{k-1,1} = \frac{\theta}{1-r}\pi_{N,0}, c-d+2 \le k \le c.$$

Hence, we have

$$\pi_{c-d+1,1} = \frac{\theta}{\delta_1(1-r)}\pi_{N,0}. \tag{26}$$

Iterating over the above equations, the first expression in (7) is obtained. Substituting (8) into it, (9) is yielded.

Substituting (25) and (26) into (12), and considering (11), that (12) always holds.

Finally, G can be determined by the normalization condition as $\sum_{i=0}^{K}\pi_{i,0} + \sum_{i=c-d+1}^{K}\pi_{i,1} = 1$. ∎

Theorem 3: If $k \ge N$, the stationary probability distribution of (L_v, J) is

$$\pi_{k,0} = \begin{cases} \pi_{c,0} r^{k+1-N}\left(\dfrac{\lambda}{\sigma_d}\right)^{N-1-c}, N \le k < K, \\ \pi_{c,0}\dfrac{\lambda r^{K-N}}{\sigma_d+\theta}\left(\dfrac{\lambda}{\sigma_d}\right)^{N-1-c}, k = K. \end{cases}$$

$$\pi_{k,1} =$$

$$\begin{cases} \pi_{c,0}\dfrac{\theta r}{\delta_d(1-r)}\left(\dfrac{\lambda}{\sigma_d}\right)^{N-1-c}\left(\dfrac{\lambda}{\delta_d}\right)^{k-N}\left[1+\sum_{j=1}^{k-N}\left(\dfrac{r\delta_d}{\lambda}\right)^j\right] \\ \quad + \pi_{c,1}\left(\dfrac{\lambda}{\delta_d}\right)^{k-c}, N \le k < K, \\ \pi_{c,0}\dfrac{r}{\delta_d}\left(\dfrac{\lambda}{\sigma_d}\right)^{N-1-c}\left[\dfrac{\lambda r^{K-1-N}}{\sigma_d+\theta} + \dfrac{\theta}{1-r}\left(\dfrac{\lambda}{\delta_d}\right)^{K-N}\right. \\ \quad \left. \times\left(1+\sum_{j=1}^{K-1-N}\left(\dfrac{r\delta_d}{\lambda}\right)^j\right)\right] + \pi_{c,1}\left(\dfrac{\lambda}{\delta_d}\right)^{K-c}, k = K. \end{cases}$$

Proof: When $N \le k < K$, with the matrix-geometric solution [52], we have

$$\pi_k = \pi_N R^{k-N}, \text{ i.e., } (\pi_{k,0}, \pi_{k,1}) = (\pi_{N,0}, \pi_{N,1})R^{k-N}. \tag{27}$$

From (2), we have

$$R^k = \begin{pmatrix} r^k & \dfrac{\theta r}{\delta_d(1-r)}\sum_{j=1}^{k} r^{j-1}\rho^{k-j} \\ 0 & \rho^k \end{pmatrix}.$$

Substituting R^k into (27), we obtain

$$\pi_{k,0} = \pi_{N,0} r^{k-N} = \pi_{c,0} r^{k+1-N}\left(\frac{\lambda}{\sigma_d}\right)^{N-1-c}, \tag{28}$$

$$\pi_{k,1} = \pi_{N,0}\frac{\theta r}{\delta_d(1-r)}\sum_{j=1}^{k-N} r^{j-1}\rho^{k-N-j} + \pi_{N,1}\rho^{k-N}. \tag{29}$$

When $k = K$, based on the equilibrium equations, we have

$$\lambda\pi_{K-1,0} - (\sigma_d+\theta)\pi_{K,0} = 0, \tag{30}$$

$$\lambda\pi_{K-1,1} + \theta\pi_{K,0} - \delta_d\pi_{K,1} = 0. \tag{31}$$

From (30), we obtain

$$\pi_{K,0} = \frac{\lambda}{\sigma_d+\theta}\pi_{K-1,0}. \tag{32}$$

From (31) and (32), we obtain

$$\pi_{K,1} = \frac{\lambda\theta}{\delta_d(\sigma_d+\theta)}\pi_{K-1,0} + \frac{\lambda}{\delta_d}\pi_{K-1,1} \tag{33}$$

Substituting (28) and (29) into (32) and (33), both $\pi_{k,0}$ and $\pi_{k,1}$ can be derived. ∎

D. System Performance Measures

Using the obtained stationary distribution, we can numerically evaluate the system performance for the *N/(d, c)-M/M/c/K/SMWV* queue. The expressions for various performance measures of the system are as follows.

1) The expected number of customers in the system is

$$E[L_s] = \sum_{i=0}^{K} i\pi_{i,0} + \sum_{i=c-d+1}^{K} i\pi_{i,1}. \tag{34}$$

2) The expected number of customers in the queue is

$$E[L_q] = \sum_{i=c-d+1}^{K}(i-(c-d))\pi_{i,0} + \sum_{i=c+1}^{K}(i-c)\pi_{i,1}. \tag{35}$$

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

$$E[NB] = c - E[WV] - E[I]. \tag{36}$$

4) The average number of servers during WC periods is

$$E[WV] = d\sum_{i=0}^{K}\pi_{i,0}. \tag{37}$$

5) The average number of servers during idle periods is

$$E[I] = \sum_{i=0}^{c-d}(c-d-i)\pi_{i,0} + \sum_{i=c-d+1}^{c}(c-i)\pi_{i,1}. \tag{38}$$

6) The expected delay time that the customer sojourns in the system is

$$E[T] = \frac{E[L_s]}{\lambda(1-P_{\text{loss}})}. \tag{39}$$

7) The probability that the customers are lost is

$$P_{\text{loss}} = \pi_{K,0} + \pi_{K,1}. \tag{40}$$

V. Optimization Analysis

A. Cost and Performance Analysis

We first develop the total expected energy consumption (or cost) function per customer per unit time for the system. We define the following cost parameters:

$C_h \equiv$ holding cost per unit time for each customer which presents in the system;

$C_b \equiv$ cost incurred per unit time when the server provides service during a normal busy period;

$C_v \equiv$ cost incurred per unit time when the server provides service during a WC period;

$C_i \equiv$ cost incurred per unit time when the server is idle in the system;

$C_d \equiv$ penalty cost per unit time when the customer is delayed in the system;

$C_l \equiv$ cost per unit time when the customer is lost.

Utilizing the above cost parameters and the concept of crew-service equipment by White *et al.* [53], the expected cost function per unit time under a complex and nonlinear cost structure is given by

$$F(d, N, K, \mu_B, \mu_V) = C_h E[L_s] + C_b E[NB] + C_v E[WV] + C_i E[I] + C_d E[T] + C_l P_{\text{loss}} \tag{41}$$

where $E[L_s]$, $E[NB]$, $E[WV]$, $E[I]$, $E[T]$, P_{loss} are given in (34), (36)–(40), respectively. In (41), the first term and the last term are the cost incurred by the customers, the others are incurred by the servers.

In [1], the nominal time delay τ_{nom} is chosen as the average value of the best-case and the worst-case delay. However, the best-case delay is assumed as the lower bound nh of every sampling period, where h is the sampling period, and n is the sampling number. This will cause the implementation bias of system which will lead to system instability. In principle, an appropriate τ_{nom} can guarantee the closed-loop system stability. We choose the mean delay $E[T_q]$ of customers in the queue as the nominal time delay, that is,

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

In theory, more controllers in dormant states mean less cost of system and greater average delay $E[T_q]$ of customers in the queue. We consider the multiobjective optimal model as follows:

Minimize

$$f = (F(d, N, K, \mu_B, \mu_V), \tau_{\text{nom}}), \tag{42}$$

subject to

$$\begin{cases} 0 < d < c < N < K < K', \mu_V < \mu_B, \\ \mu'_B \le \mu_B \le \mu''_B, \mu'_V \le \mu_V \le \mu''_V, \\ E[NB] \le c - d \end{cases}$$

where K' denotes the upper bound of the capacity K, μ'_B (or μ''_B)) and μ'_V (or μ''_V)) represent the lower (or upper) bounds of the μ_B and μ_V, respectively.

Our purpose is to determine the optimal value of the number of the semi-dormant controllers d, say d^*, the optimal value of the threshold N, say N^*, the optimal value of the capacity K, say K^*, the optimal service rate during the normal busy period μ_B, say μ_B^*, and the optimal service rate during the WC period μ_V, say μ_V^*, in order to minimize the expected cost function and the mean delay of customers in the queue. To this end, a PSO algorithm is implemented to deal with the multiobjective optimization problem numerically. We adopt the MOPSO approach proposed in [16] to generate the Pareto optimal solutions of our multiobjective optimal model.

Algorithm 1: Optimal Stopping Algorithm.

```
Input: $\omega_1(t), \omega_2(t), \cdots, \omega_c(t)$, and tolerance $\varepsilon = 10^{-7}$.
Output: $d^*$.
1: Let $S_1$
   and $S_2$ be the sets of states where one continues and
   stops, respectively, and let $S_1 = S_2 = \emptyset$, where $\emptyset$ is
   denoted as an empty set.
2: Let $u_1(t) = [u_{11}(t), u_{12}(t), \cdots, u_{1c}(t)]^T$ be the initial
   expected payoff, and let
      $u_{1j}(t) = \begin{cases} 0, h_j(t) = 0, \\ \max\{h_i(t), 1 \le i \le c\}, h_j(t) \ne 0, \end{cases}$
   where $1 \le j \le c$.
3: Let $u_i(t) = \max\{Pu_{i-1}(t), H(t)\}$, where the
   expected payoff $u_i(t) = [u_{i1}(t), u_{i2}(t), \cdots, u_{ic}(t)]^T$
   and $i = 2, 3, \cdots$
4: For every $i = 2, 3, \cdots$, repeat Step 3 until
   $|u_i(t) - u_{i-1}(t)| < \varepsilon e$, where $e$ is a $c$-dimension row
   vector with all elements equal to one. The maximum
   expected payoff function $u_k(t) = [u_{k1}(t),$
   $u_{k2}(t), \cdots, u_{kc}(t)]^T$ is obtained.
5: For every $j \in [1, c]$, if $u_{kj}(t) = f_j(t)$ then
   $S_2 = S_2 + \{j\}$, else $S_1 = S_1 + \{j\}$.
6: Let $d^* = |S_2|$, where $|S_2|$ is the number of elements in
   stopping set $S_2$. Output the elements $j_1, j_2, \cdots, j_{d^*}$ in
   $S_2$ to show all semi-dormant controllers.
7: return $d^*$.
```

B. Optimal Stopping Algorithm

Because the corresponding nondominated vectors of the Pareto front are not unique, one needs to determine the unique optimal value of the discrete decision variable d, and take the optimal values of the discrete decision variables N, K, and continuous decision variables μ_B and μ_V simultaneously. We use the optimal stopping strategy of Markov chains presented in [15] to determine the optimal semi-dormant number d^* for c controllers.

We suppose that $P = (p_{ij})_{c \times c}$ is the transition matrix for a discrete-time Markov chain X_n with state space $S = \{1, 2, \ldots, c\}$, and let $p_{ij} = \frac{1}{c}$. Suppose that the packet-in message number of the ith controller application is $\omega_i(t)$ in $[0, t]$. Let the payoff function be given by a column vector $H(t) = [h_1(t), h_2(t), \ldots, h_c(t)]^T$ in $[0, t]$, and we have

$$h_j(t) = \begin{cases} \omega_k(t) - \min\{\omega_i(t), 1 \le i \le c\} = 0, \\ k = \arg\min_i \{\omega_i(t)\}, \\ \frac{1}{c}\sum_{i=1}^{c} \omega_i(t) - (\omega_j(t) - \min\{\omega_i(t), 1 \le i \le c\}), \\ \text{otherwise} \end{cases}$$

where $k \in S$ is an absorbing state, and $1 \le j \le c$. The optimal stopping algorithm is described as Algorithm 1.

Algorithm 2: Scheduling Algorithm of FSDMC.

```
Input: $\omega_1(t), \omega_2(t), \cdots, \omega_c(t)$, and $\theta, c$.
Output: The optimal values $\tilde{F}(\tilde{d}^*, \tilde{N}^*, \tilde{K}^*, \tilde{\mu}_B^*, \tilde{\mu}_V^*)$, $\tilde{\tau}_{\text{nom}}$
    and $\tilde{\Phi} = \left[\tilde{d}^*, \tilde{N}^*, \tilde{K}^*, \tilde{\mu}_B^*, \tilde{\mu}_V^*\right]$.
1: Initialize the upper and lower bounds of decision
   variables, i.e., $N(c < N < K)$, $K(N < K < K')$,
   $\mu_B(\mu'_B \le \mu_B \le \mu''_B)$ and $\mu_V(\mu'_V \le \mu_V \le \mu''_V)$.
2: Sample the arrival packet-in message number for the $i$th
   controller application $\omega_{ij}(\Delta t)$ of the $j^{th}$ times in time
   interval $t$, where $i = 1, 2, \cdots, c$. Suppose that times of
   sampling is $M$, where $M$ represents a larger positive
   integer, e.g., $M = 10000$. The total average arrival rate
   is $\lambda = \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{c}\frac{\omega_{ij}(\Delta t)}{\Delta t}$.
3: For each $j \in [1, M]$, call Algorithm 1, and count the
   frequency $q(d^*)$ of the number of semi-dormant
   controllers $d^*(d^* = 1, 2, \cdots, c)$, respectively.
4: Let $\tilde{d}^* = \arg\max_{d^*}\{q(d^*), 1 \le d^* \le c\}$. Determine
   the optimal values $\tilde{d}^*$ of the number of semi-dormant
   controllers.
5: For each $k \in [1, M']$, call MOPSO algorithm (see [16]),
   and compute the stationary probability $\pi_{i,0}$ $(0 \le i \le K)$
   and $\pi_{j,1}$ $(c - d + 1 \le j \le K)$ using Theorem 2 and 3
   so that the Pareto front for decision variables
   $d, N, K, \mu_B$ and $\mu_V$ is generated, where $M'$ denotes the
   times of calling the MOPSO algorithm. Then, select the
   unique Pareto optimal solution and the corresponding
   nondominated vector $\Phi_k^* = \left[d_k^*, N_k^*, K_k^*, \mu_{B\,k}^*, \mu_{V\,k}^*\right]$ of
   model (42) according to the optimal value $\tilde{d}^*$.
6: Compute the average value $\tilde{\Phi}$ of all nondominated
   vectors $\Phi_k^*(1 \le k \le M')$, i.e., $\tilde{\Phi} = \frac{1}{M'}\sum_{k=1}^{M'}\Phi_k^*$. The
   vector $\tilde{\Phi} = \left[\tilde{d}^*, \tilde{N}^*, \tilde{K}^*, \tilde{\mu}_B^*, \tilde{\mu}_V^*\right]$ of the optimal values
   of all decision variables is obtained, where
   $\tilde{N}^* = \text{INT}\left(\frac{1}{M'}\sum_{k=1}^{M'} N_k^*\right)$ and $\tilde{K}^* =$
   $\text{INT}\left(\frac{1}{M'}\sum_{k=1}^{M'} K_k^*\right)$, and INT is a rounding function.
7: Substitute $\lambda$, $\theta$, $c$, $\tilde{d}^*$ and $\tilde{\Phi}$ into $F(d, N, K, \mu_B, \mu_V)$
   and $\tau_{\text{nom}}$, respectively, then generate the optimal values
   $\tilde{F}(\tilde{d}^*, \tilde{N}^*, \tilde{K}^*, \tilde{\mu}_B^*, \tilde{\mu}_V^*)$ and $\tilde{\tau}_{\text{nom}}$.
8: return $\tilde{\Phi}, \tilde{F}, \tilde{\tau}_{\text{nom}}$.
```

The purpose of optimal stopping algorithm is to find an optimal stopping strategy for maximizing the expected payoffs (see [15]). We define the payoff function to be given by a column vector $H(t) = [h_1(t), h_2(t), \cdots, h_c(t)]^T$ in $[0, t]$. The corresponding state k for the minimum packet-in message number in all control applications is defined as an absorbing state in $[0, t]$, and its payoff is zero. The payoffs in other states are defined as the difference between the average packet-in message number in all control applications and the difference of the packet-in message number of the current state and the state k.

$u_i(t)$ is denoted as the expected payoff function that is obtained after the $(i-1)$th iteration. We first initialize the expected payoff function $u_1(t) = [u_{11}(t), u_{12}(t), \ldots, u_{1c}(t)]^T$ in Step 2. The maximum expected payoff function $u_k(t) = [u_{k1}(t), u_{k2}(t), \ldots, u_{kc}(t)]^T$ is obtained by multiple iterations in Steps 3 and 4. The state space of Markov chain is divided into two sets S_1 and S_2. If the state of the chain is in S_1, one continues; if it is in S_2, it stops. The number of elements in stopping set S_2 is the optimal value of the number of semi-dormant controllers.

TABLE I
SYSTEM PERFORMANCE MEASURES FOR DIFFERENT VALUES OF (μ_B, μ_v) WITH $\lambda = 5.0$, $\theta = 0.05$, $d = 4$, $c = 10$, $N = 20$

(μ_B, μ_v)	(2.5, 1.0)	(3.5, 1.0)	(4.5, 1.0)	(2.5, 0.5)	(2.5, 1.5)
$E[L_s]$	2.0617	1.4298	1.1113	2.0867	2.0450
$E[L_q]$	0.0372	0.0013	0.0002	0.0502	0.0287
$E[NB]$	2.0245	1.4285	1.1111	2.0366	2.0163
$E[WV]$	4.0000	4.0000	4.0000	4.0000	4.0000
$E[I]$	3.9755	4.5715	4.8889	3.9634	3.9837

C. Scheduling Algorithm of FSDMC

We need to determine the optimal values $\tilde{d}^*$, $\tilde{N}^*$, $\tilde{K}^*$, $\tilde{\mu}_B^*$, and $\tilde{\mu}_V^*$ in the *N/(d,c)-M/M/c/K/SMWV* queuing system. The scheduling algorithm of FSDMC is shown as Algorithm 2.

Algorithm 2 first calls Algorithm 1 M times to count the frequency of the number of semi-dormant controllers. The peak value of the frequency is selected as the optimal value $\tilde{d}^*$ of the number of semi-dormant controllers. In Step 5, the MOPSO algorithm is called M' times to generate M' Pareto optimal solution sets of the multiobjective optimal model (42). In each Pareto optimal set, the unique nondominated vector Φ_k^* is selected to assign the values of system parameters by $d_k^* = \tilde{d}^*$. The average value $\tilde{\Phi}$ of all nondominated vectors Φ_k^* $(1 \le k \le M')$ is defined as the vector of the optimal values of system parameters.

VI. NUMERICAL RESULTS

A. Sensitivity Analysis of the System Performance Measure

Under the stability condition, we get some results of numerical experiment to show that the system performance is influenced by the change of system parameters. We fix the maximum capacity $K = 28$ and consider different values of system parameters in the following three cases.

Case 1: $\lambda = 5.0$, $\theta = 0.05$, $d = 4$, $c = 10$, $N = 20$, and varying the values of (μ_B, μ_V).

Case 2: $\mu_B = 2.5$, $\mu_V = 1.0$, $d = 4$, $c = 10$, $N = 20$, and varying the values of (λ, θ).

Case 3: $\lambda = 5.0$, $\theta = 0.05$, $\mu_B = 2.5$, $\mu_V = 1.0$, and varying the values of (d, c, N).

The numerical results of the system performances for the above three cases are reported in Tables I–III, respectively. From Table I, we can see that with increasing values of μ_B or μ_V, $E[L_s]$, $E[L_q]$, and $E[NB]$ all decrease, but $E[I]$ increases. It is found from Table II that

TABLE II
SYSTEM PERFORMANCE MEASURES FOR DIFFERENT VALUES OF (λ, θ) WITH $\mu_B = 2.5$, $\mu_V = 1.0$, $d = 4$, $c = 10$, $N = 20$

(λ, θ)	(4,0.1)	(5, 0.1)	(6,0.1)	(5, 0.05)	(5, 0.15)
$E[L_s]$	1.6034	2.06171701	3.0974	2.06171703	2.06171700
$E[L_q]$	0.0031	0.03719511	0.3986	0.03719514	0.03719509
$E[NB]$	1.6003	2.02452190	2.6988	2.02452189	2.02452191
$E[WV]$	4.0000	3.99999993	4.0000	3.99999995	3.99999991
$E[I]$	4.3997	3.97547817	3.3012	3.97547815	3.97547818

TABLE III
SYSTEM PERFORMANCE MEASURES FOR DIFFERENT VALUES OF (d, c, N) WITH $\lambda = 5.0$, $\theta = 0.05$, $\mu_B = 2.5$, $\mu_V = 1.0$

(d, c, N)	(4, 10, 20)	(7, 10, 20)	(4, 15, 20)	(4, 10, 15)
$E[L_s]$	2.06171707	2.2657	2.0002	2.06171667
$E[L_q]$	0.03719520	0.4031	0.0001	0.03719458
$E[NB]$	2.02452187	1.8626	2.0001	2.02452209
$E[WV]$	4.00000000	7.0000	4.0000	3.99999945
$E[I]$	3.97547813	1.1374	8.9999	3.97547846

TABLE IV
EXPECTED COST FUNCTION AND THE NOMINAL TIME DELAY FOR DIFFERENT VALUES OF (N, μ_B) WITH $\lambda = 5.0$, $\theta = 0.05$, $\mu_V = 1.0$, $c = 10$, $d = 4$

(N, μ_B)	(15, 2.0)	(15, 2.5)	(15, 3.0)
$F(N, \mu_B)$	206.062959	224.747422	232.647279
$E[T_q]$	0.11410504	0.00743892	0.00097276
(N, μ_B)	(20, 2.0)	(20, 2.5)	(20, 3.0)
$F(N, \mu_B)$	206.062781	224.747418	232.647279
$E[T_q]$	0.11411172	0.00743904	0.00097277

TABLE V
EXPECTED COST FUNCTION AND THE NOMINAL TIME DELAY FOR DIFFERENT VALUES OF (N, μ_V) WITH $\lambda = 5.0$, $\theta = 0.05$, $\mu_B = 2.5$, $c = 10$, $d = 4$

(N, μ_V)	(15, 0.5)	(15, 1.0)	(15, 1.5)
$F(N, \mu_V)$	224.510822	224.747422	224.909647
$E[T_q]$	0.0100299	0.00743892	0.00573672
(N, μ_V)	(20, 0.5)	(20, 1.0)	(20, 1.5)
$F(N, \mu_V)$	224.510807	224.747418	224.909646
$E[T_q]$	0.01003027	0.00743904	0.00573677

1) both $E[L_s]$ and $E[L_q]$ drastically increase as λ increases, but slightly decrease as θ increases;
2) $E[NB]$ increases as λ increases, but $E[I]$ decreases as λ increases;
3) both $E[NB]$ and $E[I]$ slightly increase as θ increases.

Table III shows that

1) both $E[L_s]$ and $E[L_q]$ increase as d increases, but $E[NB]$ and $E[I]$ decrease as d increases;
2) $E[L_s]$, $E[L_q]$ and $E[NB]$ all decrease as c increases, but $E[I]$ drastically increases as c increases;
3) similar to the previous 1), both $E[L_s]$ and $E[L_q]$ slightly increase, but $E[NB]$ and $E[I]$ slightly decrease as N increases.

From Tables I–III, we find that $E[WV]$ is almost equal to d no matter how the other parameters change. The above results are consistent with the actual situation for the *N/(d,c)-M/M/c/K/SMWV* queue.

B. Sensitivity Analysis of the Expected Cost Function and the Nominal Time Delay

As a typical example of ANCS, a high-end automotive electronic system often employs FlexRay buses to connect or organize a large number of ECUs. For example, in a vehicle cruise subsystem, it is assumed that multiple tasks are mapped to ten ECUs connected with dual-channel FlexRay buses at 5 kB/s. We consider a good tradeoff between system implementation cost and system performance by generating appropriate system parameters. The system parameters used in this work are in accordance to [50] and [51]. We first analyze the sensitivity of the expected cost function and the nominal time delay with respect to decision variables. Subsequently, we compute the energy consumption of the system and the mean waiting delay in the queue by solving the multiobjective optimal model (42). We fix the values of the cost parameters: $C_h = 1$ mW, $C_b = 35$ mW, $C_v = 15$ mW, $C_i = 10$ mW, $C_d = 8$ mW, $C_l = 10\,000$ mW and the values of the decision variables: $\lambda = 5.0, \theta = 0.05, K = 28$. We focus on the following six cases:

Case 4: $c = 10$, $N = 20$, $\mu_V = 1.0$, and varying the values of μ_B from 2.0 to 5.0 and $d = 4, 7, 9$.

Case 5: $c = 10$, $N = 20$, $\mu_B = 2.5$, and varying the values of μ_V from 0.5 to 3.0 and $d = 4, 7, 9$.

Case 6: $d = 4$, $N = 20$, $\mu_V = 1.0$, and varying the values of μ_B from 2.0 to 5.0 and $c = 6, 10, 15$.

Case 7: $d = 4$, $N = 20$, $\mu_B = 2.5$, and varying the values of μ_V from 0.5 to 3.0 and $c = 6, 10, 15$.

Case 8: $d = 4$, $c = 10$, $\mu_V = 1.0$, and varying the values of μ_B from 2.0 to 5.0 and $N = 15, 20$.

Case 9: $d = 4$, $c = 10$, $\mu_B = 2.5$, and varying the values of μ_V from 0.5 to 3.0 and $N = 15, 20$.

The numerical results of the expected cost function F and the nominal time delay $E[T_q]$ are depicted in Figs. 7–10 for Cases 4– 7, and in Tables IV and V for Cases 8 and 9, respectively. It can be seen from Fig. 7 that F increases as μ_B increases, but decreases as d increases. It is revealed from Fig. 8 that F slightly increases as μ_V increases, but also decreases as d increases. However, the change of $E[T_q]$ is just the opposite to F as μ_B, μ_V, and d increase, respectively. The situation of Figs. 9 and 10 and Tables IV and V is similar to Figs. 7 and 8, respectively.

Next, we examine the behavior of the expected cost function under different values of the cost parameters. We first fix the parameters of the system as follows: $\lambda = 5.0, \theta = 0.05, \mu_B = 2.5, \mu_V = 1.0, d = 4, c = 10, N = 20$, and $K = 28$. Tables VI–VIII illustrate the effects of (C_h, C_i), (C_v, C_b), and (C_d, C_l) on the expected cost function, respectively. It is found that C_h affects $F(4, 20, 2.5, 1.0)$ significantly when the other cost parameters are fixed. Apparently, $F(4, 20, 2.5, 1.0)$ is increasing in C_h. This implies that when the holding cost increases, the expected cost

TABLE VI
EFFECTS OF (C_h, C_i) ON THE EXPECTED COST FUNCTION $F(4, 20, 2.5, 1.0)$ WITH $C_v = 15$, $C_b = 35$, $C_d = 8$, $C_l = 10\,000$

(C_h, C_i)	(1, 10)	(3, 10)	(5, 10)	(1, 12)	(1, 14)
F	224.7475	228.871	232.9944	228.7966	232.8456

TABLE VII
EFFECTS OF (C_v, C_b) ON THE EXPECTED COST FUNCTION $F(4, 20, 2.5, 1.0)$ WITH $C_h = 1$, $C_i = 10$, $C_d = 8$, $C_l = 10\,000$

(C_v, C_b)	(15, 35)	(17, 35)	(19, 35)	(15, 37)	(15, 39)
F	224.7475	232.7475	240.7475	232.6985	240.6495

TABLE VIII
EFFECTS OF (C_d, C_l) ON THE EXPECTED COST FUNCTION $F(4, 20, 2.5, 1.0)$ WITH $C_h = 1$, $C_b = 35$, $C_v = 15$, $C_i = 10$

(C_d, C_l)	(8,10000)	(10,10000)	(12,10000)	(8,20000)	(8,30000)
F	224.7475	225.5722	226.3969	224.7477	224.7478

TABLE IX
SAMPLING DATA OF THE ARRIVAL PACKET-IN MASSAGES NUMBER

Ctrl. App. ID	1	2	3	4	5
Sample Data(B/s)	628	[361]	841	665	473
Ctrl. App. ID	6	7	8	9	10
Sample Data(B/s)	667	[266]	[256]	446	[397]

will go up. The effects of the other cost parameters on the system performance can be quantified too. However, it is emphasized that a numerical result would be inconsistent with the above observation for different values of the system parameters.

C. Optimal Values of System Parameters

In this section, we present the numerical results to solve the optimal values $\tilde{d}^*$, $\tilde{\Phi}$, $\tilde{F}$, and $\tilde{\tau}_{\text{nom}}$ using Algorithm 2. We still choose various parameters in the system as follows: $C_h = 1$ mW, $C_b = 35$ mW, $C_v = 15$ mW, $C_i = 10$ mW, $C_d = 8$ mW, $C_l = 10000$ mW, $\theta = 0.05$, $c = 10$. We first determine the optimal value $\tilde{d}^*$ of the number of semi-dormant controllers. Table IX shows a sampling data generated randomly of the arrival packet-in massages for all control applications in unit time.

In Table IX, the data in square brackets is selected to represent that the corresponding controller will enter the dormant state in light traffic using Algorithm 1. Therefore, the number of the semi-dormant controllers is obtained, that is, $d = 4$. The total arrival rate is also generated by summing the second row, that is, $\lambda = \sum_{i=1}^{c} \omega_i(t) = 5000\text{B} = 5\text{kB}$.

When Algorithm 1 is called ten thousand times, that is, $M = 10\,000$, the statistical frequencies of the number of semi-dormant controllers is derived, and is shown in Fig. 11. We find that the number of semi-dormant controllers generally follows a normal distribution. Hence, we choose the corresponding semi-dormant controller number of maximal frequency as the optimal value $\tilde{d}^*$, that is, $\tilde{d}^* = 4$. It is noteworthy that the sampling data in Fig. 11 is not unique, but generally follows a normal distribution.

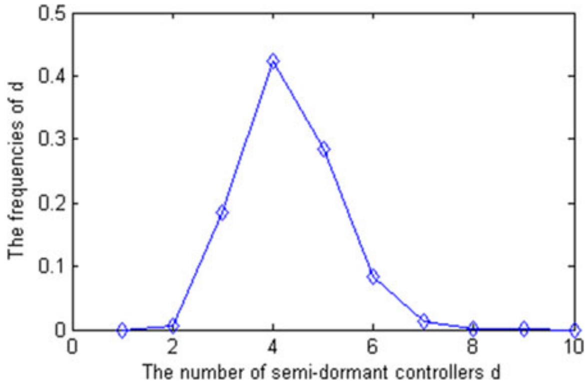


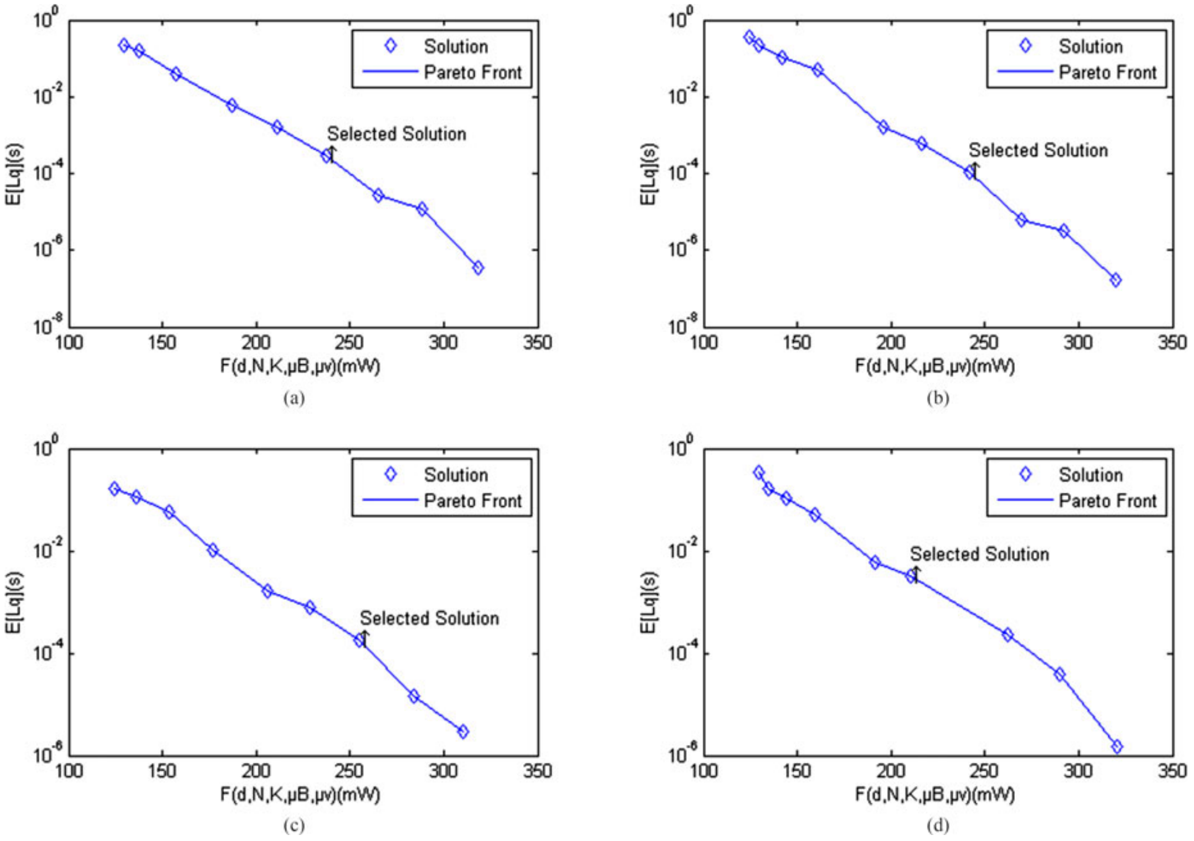
Fig. 11. Frequency diagram of the number of semi-dormant controllers.

Next, we derive the optimal values $\tilde{\Phi}$, $\tilde{F}$, and $\tilde{\tau}_{\text{nom}}$ by Algorithm 2. While calling MOPSO, we use a population of 100 particles, a repository size of 40 particles, a mutation rate of 0.5, an iteration times of 3000, and 30 divisions for the adaptive grid. Furthermore, let $\lambda = 5.0$. The decision variables are chosen as $d \in [1, c]$, $N \in [c+1, K-1]$, $K \in [N+1, K+10]$, $\mu_B \in [2.0, 5.0]$, and $\mu_v \in [0.5, 3.0]$, and satisfy the constraint conditions in model (42). Fig. 12 respectively shows four Pareto fronts that are produced by calling the MOPSO four times. In Fig. 12(a)-(d), the selected Pareto optimal solution is determined while $\tilde{d}^* = 4, 4, 3, 5$, respectively.

The Pareto optimal solutions and the corresponding nondominated vectors in Fig. 12(a) are shown in Table X.

Suppose that the times of calling MOPSO algorithm is 10, that is, $M' = 10$. We obtain ten selected Pareto optimal solutions and the corresponding nondominated vectors (see Table XI).

From Fig. 11 and Table XI, we obtain the design parameter values of the FSDMC model above. While $c = 10, \theta = 0.05$, we have $\tilde{d}^* = 4$, $\tilde{N}^* = 20$, $\tilde{K}^* = 28$, $\tilde{\mu}_B^* = 3.4$, $\tilde{\mu}_V^* = 1.5$. However, the value of λ is determined by $\lambda = \sum_{i=1}^{c} \omega_i(t)$. Here, $\lambda = 5$ kB. Substitute the parameter values into the objective function, then we get the values of the expected cost function F and nominal time delay τ_{nom}, which are $\tilde{F} = 237.07$ mW and $\tilde{\tau}_{\text{nom}} = 2.9259\text{e} - 004$ s, respectively. The result is basically in agreement with those of Tables X and XI. In the vehicle cruise controller system example, four idle controllers among the ten controllers are allowed to enter the semi-dormant state under light traffic case. When the number of the packet-in massages in the global queue comes up to 20, the four semi-dormant controllers return from the WCs. The capacity of the system is assigned as 28. The total Poisson arrival rate of the global queueing model is 5 kB/s. The mean service rates of each controller during a normal busy period and a WC period are 3.4 and 1.5 kB/s respectively. Hence, we can obtain the optimal values of the corresponding expected cost function and nominal time delay. It should be noted that if the bandwidth of FlexRay buses is 10 Mb/s [54], or CAN buses 500 kb/s [55], the optimal values of system parameters can be obtained accordingly. If the



(a) (b) (c) (d)

Fig. 12. Pareto fronts produced by four times calling the MOPSO.

TABLE X
PARETO OPTIMAL SOLUTIONS AND THE CORRESPONDING NONDOMINATED VECTORS IN FIG. 12(A)

d	N	K	μ_B	μ_V	F	τ_{nom}
1	16	22	4.23	0.57	318.52	3.5453E− 07
2	21	25	3.58	1.96	288.71	1.1431E− 05
3	19	28	3.72	2.73	264.89	2.6012E− 05
4	**21**	**28**	**3.41**	**1.43**	**237.16**	**2.8473E− 04**
5	25	28	3.29	1.55	213.94	1.6183E− 03
6	18	28	3.35	1.92	187.01	5.9476E− 03
7	17	23	2.64	2.49	157.11	4.0771E− 02
8	14	28	2.36	2.02	137.78	1.6100E− 01
9	23	28	3.31	1.02	129.43	2.2617E− 01

TABLE XI
SELECTED OPTIMAL SOLUTION AND THE CORRESPONDING NONDOMINATED VECTORS

No.	d	N	K	μ_B	μ_V	F	τ_{nom}
1	3	21	28	3.95	1.17	261.65	1.8834E− 05
2	3	20	28	3.85	1.11	260.91	2.4267E− 05
3	4	21	28	3.41	1.43	237.16	2.8473E− 04
4	4	20	28	3.40	1.91	237.07	2.7046E− 04
5	4	23	28	3.45	2.07	237.55	2.4130E− 04
6	4	21	28	3.36	1.32	236.67	3.3089E− 04
7	5	20	28	3.20	1.43	215.10	1.8553E− 03
8	5	20	28	3.15	2.31	214.59	1.7335E− 03
9	5	17	28	3.15	0.72	214.50	2.4159E− 03
10	6	22	28	3.12	1.95	188.72	7.7024E− 03
Average	4.3	20.5	28	3.404	1.542	–	–

number of controllers c varies in the FSDMC, the number of semi-dormant controllers d maybe change.

D. Comparison and Analysis

In our work, we consider the tradeoff between system implementation cost and system performance. According to the features of the ANCS, we established an *M/M/c* queueing model by adopting multiple strategies, such as *N*-policy, (d, c) vacation mechanism, capacity K, and SMWV. Our model made a more comprehensive consideration about the features of the ANCS than the proposed models in [11], [40], [43], [44] and [45]. We have obtained the closed forms of the stationary probability distribution of the queueing model to compute the system performance measures.

We have established the multibjective optimization model. The single objective optimization problem aiming at minimizing the system implementation cost was only considered in [11], [40], [43]–[45], [54]. The system time delay problem was investigated in [1], [3], [4] and [10], but the optimal problem about the time delay was not considered. Lehoczky [50], [51] modeled the real-time system using the queueing theory, but did not refer to the delay optimal problem as well. Diaz *et al.* [49] and Zeng *et al.* [56] studied the time delay problem of real-time system using stochastic analysis approach, but did not discuss the multiobjective optimization problem for considering the system implementation cost and system performance. Hence, our proposed queueing model is more suitable for modeling the ANCS than those in the above references. We have proposed the optimal stopping algorithm to yield the unique optimal value of the number of semi-dormant controllers. This algorithm is the basis of obtaining the optimal values of other parameters in our work. Our numerical results show the feasibility and the usefulness of the proposed algorithms.

VII. Conclusion

In this paper, we have considered a novel ANCS with dual communication channels, and solved the optimal values of various parameters that are needed for designing such a system. We established an FSDMC model that allows part of idle controllers to enter the semi-dormant states under light traffic condition. We focused on the stochastic scheduling problem of the controller cluster to reduce cost and improve performance. By analyzing the characteristic of system, an *N/(d,c)-M/M/c/K/SMWV* queuing model is presented to construct the cost and performance functions, respectively. For generating the optimal values of system parameters, a multiobjective optimization model was developed to minimize the two functions. We designed a scheduling algorithm of FSDMC and an optimal stopping algorithm to solve the multiobjective optimization problem. The Pareto front and the corresponding nondominated vector set are generated to serve the system designs. Finally, numerical results verify the efficiency of the FSDMC model.

References

[1] A. Annaswamy, D. Soudbakhsh, R. Schneider, D. Goswami, and S. Chakraborty, “Arbitrated network control systems: A co-design of control and platform for cyber-physical systems,” in *Proc. Workshop Control Cyber Phys. Syst.*, Mar. 2013, pp. 339–356.

[2] S. Gao *et al.* “A cross-domain recommendation model for cyber-physical systems,” *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 2, pp. 384–393, Dec. 2013.

[3] H. Voit, “An arbitrated networked control systems approach to cyber-physical systems,” Ph.D. dissertation, Inst. Autom. Control Eng., Tech. Univ. Munich, Munich, Germany, Jun. 2013.

[4] A. Annaswamy, S. Chakraborty, D. Soudbakhsh, D. Goswami, and H. Voit, “The arbitrated networked control systems approach to designing cyber-physical systems,” in *Proc. 3rd IFAC Workshop Distrib. Estimation Control Netw. Syst.*, Sep. 2012, pp. 174–179.

[5] H. Zeng, M. D. Natale, A. Ghosal, and A. Sangiovanni-Vincentelli, “Schedule optimization of time-triggered systems communicating over the FlexRay static Segment,” *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 1–17, Feb. 2011.

[6] J. Dvorak and Z. Hanzalek, “Using two independent channels with gateway for FlexRay static segment scheduling,” *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1887–1895, Oct. 2016.

[7] M. Hu, J. Luo, Y. Wang, M. Lukasiewycz, and Z. Zeng, “Holistic scheduling of real-time applications in time-triggered in-vehicle networks,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1817–1828, Aug. 2014.

[8] M. Kang, K. Park, and M.-K. Jeong, “Frame packing for minimizing the bandwidth consumption of the FlexRay static segment,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 4001–4008, Sep. 2013.

[9] D. Goswami *et al.* “Challenges in automotive cyber-physical systems design,” in *Proc. IEEE Int. Conf. Embedded Comput. Syst., Architectures, Model. Simul.*, Jul. 2012, pp. 346–354.

[10] H. Zeng, M. D. Natale, P. Giusto, and A. Sangiovanni-Vincentelli, “Using statistical methods to compute the probability distribution of message response time in controller area network,” *IEEE Trans. Ind. Informat.*, vol. 6, no. 4, pp. 678–691, Nov. 2010.

[11] Y. Fu, J. Bi, J. Wu, Z. Chen, K. Wang, and M. Luo, “A dormant multi-controller model for software defined networking,” *China Commun.*, vol. 11, no. 3, pp. 45–55, Jun. 2014.

[12] *Software-Defined Networking: The New Norm for Networks*, ONF White Paper, 2012. [Online]. Available: https://www.opennetworking.org/component/content/article/105-module-content/related-documents/840-white-paper

[13] F. Baronti *et al.* “Design and verification of hardware building blocks for high-speed and fault-tolerant in-vehicle networks,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 792–801, Mar. 2011.

[14] *FlexRay Protocol Specification (Version 3.0.1)*, FlexRay Consortium, Garden City, NY, USA, 2010. [Online]. Available: http://www.flexray.com

[15] G. F. Lawler, *Introduction to Stochastic Processes*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2006.

[16] A. C. C. Carlos, T. P. Gregorio, and S. L. Maximino, “Handling multiple objectives with particle swarm optimization,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[17] M. Li and P. Li, “Crowdsourcing in cyber-physical systems: Stochastic optimization with strong stability,” *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 2, pp. 218–231, Dec. 2013.

[18] F. Xia and Y. Sun, “Control-scheduling codesign: A perspective on integrating control and computing,” *Dyn. Continuous, Discrete Impulsive Syst. B*, vol. 13, no. S1, pp. 1352–1358, 2006.

[19] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Trans. Automat. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

[20] H. Xu, A. Sahoo, and S. Jagannathan, “Stochastic adaptive event-triggered control and network scheduling protocol co-design for distributed networked systems,” *IET Control Theory Appl.*, vol. 8, no. 18, pp. 2253–2265, Dec. 2014.

[21] C. Peng and Q.-L. Han, “Output-based event-triggered H control for sampled-data control systems with nonuniform sampling,” in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 1727–1732.

[22] M. C. F. Donkers and W. P. M. H. Heemels, “Output-based event-triggered control with guaranteed L-gain and improved and decentralized event-triggering,” *IEEE Trans. Automat. Control*, vol. 57, no. 6, pp. 1362–1376, Jun. 2012.

[23] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, “Periodic event-triggered control based on state feedback,” in *Proc. 50th IEEE Conf. Decision Control Eur. Control Conf.*, Dec. 2011, pp. 2571–2576.

[24] Y. Guan, Q.-L. Han, and C. Peng, “Decentralized event-triggered control for sampled-data systems with asynchronous sampling,” in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 6565–6570.

[25] S. Al-Areqi, D. Gorges, and S. Liu, “Event-based networked control and scheduling codesign with guaranteed performance,” *Automatica*, vol. 57, no. 4, pp. 128–134, Jul. 2015.

[26] S. Al-Areqi, D. Gorges, and S. Liu, “Event-based control and scheduling codesign: Stochastic and robust approaches,” *IEEE Trans. Automat. Control*, vol. 60, no. 5, pp. 1291–1303, May 2015.

[27] T. Majdoub, S. L. Nours, O. Pasquier, and F. Nouvel, “Performance evaluation of an automotive distributed architecture based on a high speed power line communication protocol using a transaction level modeling approach,” *J. Real-Time Image Process.*, vol. 9, no. 1, pp. 281–295, Mar. 2014.

[28] M. Kauer, D. Soudbakhsh, D. Goswami, S. Chakraborty, and A. Annaswamy, “Fault-tolerant control synthesis and verification of distributed embedded systems,” in *Proc. 17th Design, Autom. Test Europe*, Mar. 2014, pp. 1–6.

[29] R. Postoyan, P. Tabuada, D. Nesi, and A. Anta, “A framework for the event-triggered stabilization of nonlinear systems,”*IEEE Trans. Automat. Control*, vol. 60, no. 4, pp. 982–996, Apr. 2015.

[30] X. Cao, P. Cheng, J. Chen, and Y. Sun, “An online optimization approach for control and communication codesign in networked cyber-physical systems,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 439–450, Feb. 2013.

[31] N. Kottenstette and P. J. Antsaklis, “Control of multiple networked passive plants with delays and data dropouts,” in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 3126–3132.

[32] N. Kottenstette and N. Chopra, “L_2^m-stable digital-control networks for multiple continuous passive plants,” in *Proc. 1st IFAC Workshop Estimation Control Netw. Syst.*, Sep. 2009, pp. 120–125.

[33] N. Kottenstette, J. F. Hall III, X. Koutsoukos, P. Antsaklis, and J. Sztipanovits, “Digital control of multiple discrete passive plants over networks,” *Int. J. Syst., Control Commun.*, vol. 3, no. 2, pp. 194–228, Apr. 2011.

[34] N. Kottenstette, G. Karsai, J. Sztipanovits, “A passivity-based framework for resilient cyber physical systems,” in *Proc. 2nd Int. Symp. Resilient Control Syst.*, Aug. 2009, pp. 43–50.

[35] M. Yadin and P. Naor, “Queueing system with a removable service station,” *J. Oper. Res. Soc.*, vol. 14, no. 4, pp. 393–405, Dec. 1963.

[36] D. P. Heyman, “Optimal operating policies for M/G/1 queueing system,” *Oper. Res.*, vol. 16, no. 2, pp. 362–382, Apr. 1968.

[37] Y. C. Chang and W. L. Pearn, “Optimal management for infinite capacity N-policy M/G/1 queue with a removable service station,” *Int. J. Syst. Sci.*, vol. 42, no. 7, pp. 1075–1083, Apr. 2011.

[38] Z. G. Zhang and N. Tian, “Analysis on queueing systems with synchronous vacations of partial servers,” *Perform. Eval.*, vol. 52, no. 4, pp. 269–282, May 2003.

[39] X. Xu and Z. G. Zhang, “Analysis of multiple-server queue with a single vacation (e, d)-policy,” *Perform. Eval.*, vol. 63, no. 8, pp. 825–838, Aug. 2006.

[40] J.-C. Ke, C.-H. Lin, J.-Y. Yang, and Z. G. Zhang, “Optimal (d, c) vacation policy for a finite buffer M/M/c queue with unreliable servers and repairs,” *Appl. Math. Model.*, vol. 33, no. 10, pp. 3949–3962, Oct. 2009.

[41] L. D. Servi and S. G. Finn, “M/M/1 queues with working vacations (M/M/1/WV),” *Perform. Eval*, vol. 50, no. 1, pp. 41–52, Oct. 2002.

[42] Z. -J. Zhang and X. Xu, “Analysis for the M/M/1 queue with multiple working vacations and N-policy,” *Int. J. Inf. Manage. Sci.*, vol. 19, no. 3, pp. 495–506, 2008.

[43] C.-H. Lin and J.-C. Ke, “Multi-server system with single working vacation,” *Appl. Math. Model.*, vol. 33, no. 7, pp. 2967–2977, Jul. 2009.

[44] D.-Y. Yang and C.-H. Wu, “Cost-minimization analysis of a working vacation queue with N-policy and server breakdowns,” *Comput. Ind. Eng.*, vol. 82, pp. 151–158, Apr. 2015.

[45] M. Jain and S. Upadhyaya, “Synchronous working vacation policy for finite-buffer multiserver queueing system,” *Appl. Math. Comput.*, vol. 217, no. 24, pp. 9916–9932, Aug. 2011.

[46] T. Pop, P. Pop, P. Eles, Z. Peng, and A. Andrei, “Timing analysis of the FlexRay communication protocol,” *Real-Time Syst.*, vol. 39, no. 1, pp. 205–235, Aug. 2008.

[47] G. Cena and A. Valenzano, “On the properties of the flexible time division multiple access technique,” *IEEE Trans. Ind. Informat.*, vol. 2, no. 2, pp. 86–94, May 2006.

[48] G. I. Mary, Z. C. Alex, and L. Jenkins, “Response time analysis of messages in controller area network: A review ”, *J. Comput. Netw. Commun.*, vol. 2013, pp. 1–11, 2013.

[49] J. L. Diaz *et al.*“Stochastic analysis of periodic real-time systems,” in *Proc. 23rd IEEE Real-Time Syst. Symp.*, Dec. 2002, pp. 289–300.

[50] J. P. Lehoczky, “Real-time queueing theory,” in *Proc. 17th IEEE Real-Time Syst. Symp.*, Dec. 1996, pp. 186–195.

[51] J. P. Lehoczky, “Real-time queueing network theory,” in *Proc. 18th IEEE Real-Time Syst. Symp.*, Dec. 1997, pp. 58–67.

[52] M. F. Neuts, *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1981.

[53] J. A. White, J. W. Schmidt, and G. K. Bennett, *Analysis of Queueing Systems*. New York, NY, USA: Academic, 1975.

[54] I. Park and M. Sunwoo, “FlexRay network parameter optimization method for automotive applications,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1449–1459, Apr. 2011.

[55] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, “Controller area network (CAN) schedulability analysis: Refuted, revisited and revised,” *Real-Time Syst.*, vol. 35, no. 3, pp. 239–272, Apr. 2007.

[56] H. Zeng, M. D. Natale, P. Giusto, and A. Sangiovanni-Vincentelli, “Stochastic analysis of CAN-based real-time automotive systems,” *IEEE Trans. Ind. Informat.*, vol. 5, no. 4, pp. 388–401, Nov. 2009.

Hongfang Gong received the B.S. degree in mathematics from Changsha University of Science and Technology, Changsha, China, and the M.E. degree in computer application from Hunan University, Changsha, in 1991 and 2004, respectively. He is currently working toward the Ph.D. degree with the Key Laboratory for Embedded and Network Computing of Hunan Province, Hunan University, Changsha, China.

He is currently an Associate Professor of information science with Changsha University of Science and Technology, Changsha. His research interests include cyber-physical systems, embedded computing system, and distributed control systems.

Renfa Li (M’05–SM’10) received the Ph.D. degree in electronic engineering from Huazhong University of Science and Technology, Wuhan, China, in 2003.

He is a Full Professor and Dean of the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He is the Director of the Key Laboratory for Embedded and Network Computing of Hunan Province, Changsha. He is also an expert committee member of the National Supercomputing Center in Changsha, China. His research interests include computer architecture, embedded computing system, cyber-physical systems, Internet of things.

Prof. Li is a senior member of the ACM, and a member of the Council of China Computer Federation.

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

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

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

Weiwei Chen received the B.S. degree in electronic engineering from Beijing University of Posts and Telecommunications University, Beijing, China, in 2007, and the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2013.

She is currently with the Department of Communication Engineering, Hunan University, Changsha, China. Her research interests include cross-layer optimizations in wireless networks, fifth-generation cellular networks, and future Internet architecture.

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

He is a SUNY Distinguished Professor of computer science. 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. He has published more than 480 journal articles, book chapters, and refereed conference papers, and has received several best paper awards.

Prof. Li has served on the editorial boards of 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.