Performance Evaluation 125 (2018) 21–47

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

Message response time analysis for automotive cyber–physical systems with uncertain delay: An *M/PH/1* queue approach

Hongfang Gong [a,b], Renfa Li [a,*], Yang Bai [a], Jiyao An [a], Keqin Li [a,c]

[a] College of Computer Science and Electronic Engineering, and Key Laboratory for Embedded and Network Computing of Hunan Province, Hunan University, Changsha, Hunan, 410082, China
[b] School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha, Hunan, 410114, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Article history:
Received 16 August 2017
Received in revised form 16 May 2018
Accepted 9 July 2018
Available online 26 July 2018

Keywords:
Automotive cyber–physical systems
Bus configuration
Delay prediction
Improved grey model (1,1)
Message response time analysis
M/PH/1 queueing system

ABSTRACT

The analysis problem of message response time is a key and challenging problem for in-vehicle networks that has gained much research attention in recent years. The arbitrated networked control system (ANCS), which is a special cyber–physical system (CPS), was recently designed for scheduling or arbitrating networks in a control system. A dominant feature of this CPS is the arbitration of messages on a shared communication medium. The control applications in an ANCS are sensitive to the end-to-end delay of message responses from sensors to actuators. In this study, a multi-hierarchical flexible TDMA/fixed priority scheduling policy is first adopted to configure the bus in an ANCS, which is based on the event trigger protocol. Then, an *M/PH/1* queue model with random-sized batch arrivals is used to model the control application in the ANCS, and to obtain the stationary probability for each control application. On the basis of the model, a constrained optimization problem is formulated to minimize the nonlinear cost function of the ANCS, and the probability density function of the uncertain response delay for each message is generated. The use of an optimal *minislot* algorithm that is based on the DFP variable-scale method and a parameter estimation algorithm of the *pdfs* of the uncertain response delays is proposed to obtain optimal values of the bus *minislot* and matrix parameters of PH distribution, respectively. Furthermore, the probability transformed GM (1,1) (PTGM (1,1)) with two parameters, which is a novel grey model (1,1) (GM (1,1)), is proposed for predicting the response delay for each message, in which a two-parameters optimization algorithm of PTGM (1,1) that is based on the DFP-method is provided to minimize the mean absolute percentage error and obtain the optimal values of the two parameters for each message. Extensive numerical results are provided to illustrate the usefulness of the proposed algorithms and show the accuracy of the predicted results.

* Corresponding author.
E-mail addresses: gonghf@csust.edu.cn (H. Gong), lirenfa@hnu.edu.cn (R. Li), baiyang@hnu.edu.cn (Y. Bai), jt_anbob@hnu.edu.cn (J. An), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.peva.2018.07.001
0166-5316/

1. Introduction

1.1. Background

Automotive, avionics, and industrial automation systems are often distributed embedded systems (DES) with many processing units (PUs), sensors, and actuators that communicate via shared buses, such as controller area network (CAN), local interconnect network (LIN), and FlexRay, and the recent automotive Ethernet. Such architectures are utilized to run distributed control applications, often with multiple quality-of-control constraints [1]. A modern high-end car consists of up to 100 electronic control units (ECUs) with 100 million lines of code [2], and this number will continue to grow in the near future [3,4]. Typical examples include engine control units, body control subsystems, chassis controls, and safety functions, such as adaptive cruise control. With the increasing complexity of automotive electronic systems, the gap between high-level control models and their actual implementations inevitably widens. The main problem that can be attributed to this gap is the end-to-end delay between the signal sensed at the plant output and the signal sent to the control actuator [5]. The actual delay that a control system experiences on a platform varies with time and can significantly differ from and be smaller than the worst-case delay. Hence, designing platforms under the worst-case assumption often wastes many resources.

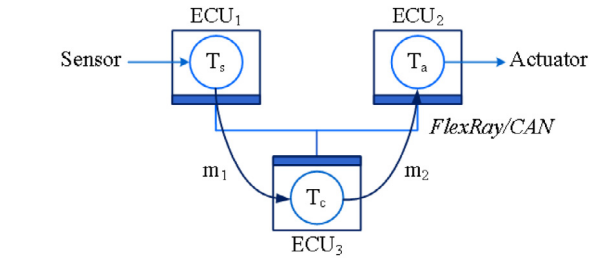
The arbitrated networked control system (ANCS), which is a special cyber–physical system (CPS), was proposed in [1,6] to emphasize that control systems are to be designed for networks or buses that are scheduled or arbitrated. A dominant feature of this CPS is the arbitration of messages on a shared communication medium [7]. In automotive CPSs (ACPSs), various functionalities are realized by many distributed tasks, which communicate by exchanging messages over the shared networks [3]. Hence, the architecture of an ACPS is in accordance with the feature of the ANCS platform, which comprises multiple distributed control applications. The performance of these applications depends on the response time of messages transmitted over the shared communication networks; the response time depends on the arbitration policy implemented [3]. A typical distributed control application and the message response time (MRT), i.e., sensor-to-actuator delay, are shown in Figs. 1(a) and (b), respectively (see Fig. 1 in [1]). An ECU collects sensor data (denoted as task T_s). A communication bus (e.g., FlexRay) then transmits the data as message m_1 to a second ECU (marked 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. Here, the sampling period is h, and the MRT is indicated by t (Fig. 1(b)) [1].

In an ANCS, the control applications are sensitive to the MRT t. For example, in FlexRay, each communication cycle of the bus is divided into a time-triggered (TT) (or static, ST) segment and an event-triggered (ET) (or dynamic, DYN) segment. On the TT segment, the tasks are given access to the bus only at their predefined slots. The tasks are assigned priorities to arbitrate for access to the ET segment. By their segment, the temporal effects of these segments on the control messages vary [6]. As application complexity and communication requirements continue to grow, the bandwidth of the TT segment becomes insufficient, and a purely TT implementation might be overly expensive. Meanwhile, priority-driven ET implementations suffer from the usual temporal non-determinism which results in poor control performance [3]. These delays, which are usually random, can significantly degrade system performance and even cause system instability [8]. The question for the fixed MRTs generated by the TT protocol was investigated in [1,6,7]. However, the approaches in these works are based on the assumption that the MRT uniformly varies between the best- and the worst-case delays, thereby leading to an overly conservative design and failing to identify feasible designs. The problem of varying delay is not considered in the ANCS. In this study, we focus on the issue of uncertain MRT analysis in ANCSs.

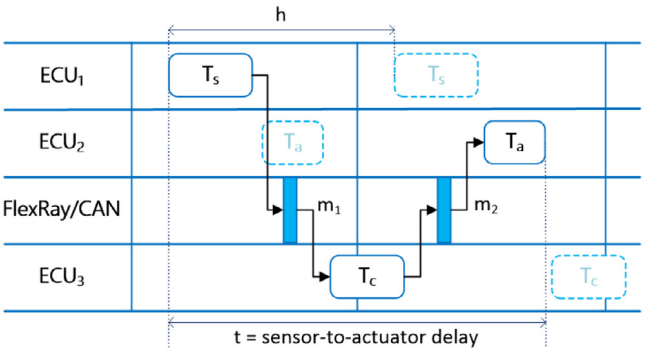
The MRTs of in-vehicle networks have been analyzed and discussed in past research works. Response-time analysis (RTA) [9–11] is a powerful, mature, and well-established method of calculating upper bounds on the response times of tasks or messages in a real-time system or a network, respectively. Tindell et al. [12] developed RTA for CAN with priority queues. This analysis provided a method for calculating the worst-case response times (WCRTs) of all CAN messages and guaranteed that all messages would meet their deadlines. This seminal work led to a large body of research into MRT analysis for in-vehicle networks and has become the basis of the schedulability analysis of in-vehicle communication buses. In these research works, the techniques applied for MRT analysis include: (i) priority- and first in first out-queue analysis [13], (ii) offset-based analysis [14], (iii) abortable and non-abortable transmit requests [15], (iv) integer linear programming [16,17], (v) probability model [18,19], (vi) heuristics based-on directed acyclic graph [20,21], and (vii) real-time calculus [1,22]. However, these studies did not theoretically consider the problem of probability distribution of MRT and the relationship between message generation (arrival) in the sensor and transmission (service) in the network. The accurate predictive problem of MRT for each message is also not studied.

1.2. Motivation

The active safety functions in automotive electronic systems, such as adaptive cruise control, gather 360° views of the environment through radar and cameras and require several processing stages before generating the actuation signals, including sensor fusion, object detection, control, and arbitration layers creation [19]. The system decreases the target speed of the cruise control until it matches the speed of a detected obstacle. In such a system, a hard deadline can be defined. The WCRT allows preventing a collision. Therefore, the WCRT of messages must be analyzed as accurately as possible to obtain a safe upper bound of the response time. A comparison of the WCRT with the corresponding deadline can validate the schedulability of messages [23]. However, in many time-critical and non-time-critical applications, the performance and



(a) Task allocation



(b) Time line

Fig. 1. A typical distributed embedded system (DES) [1].

the quality of controls depend on the average response time of tasks or messages, which must be analyzed and minimized. The probability of a worst-case event can be smaller than that of failure of the hardware components. In this case, designing for worst-case guaranteed performance can be wasteful. Therefore, worst-case evaluation may not be sufficient and must be supplemented by probabilistic analysis [24]. In the ANCS (Fig. 2), all control applications share the forward bus from sensor to controller and backward bus from controller to actuator to transmit messages. For each control application, the message generated by the sensor is successively processed by the forward bus, controller, backward bus, and finally the actuator. In certain instances, multiple messages may arrive at the shared bus simultaneously. Hence, the ANCS can be abstracted as a multi-queue multi-server serial–parallel hybrid queueing system (Fig. 6). The dynamic process for a control application conforms to the dynamic behavior of the *M/PH/1* queue.

An *M/PH/1* queue is a quasi-birth-and-death (QBD) process, where customers arrive according to a Poisson process with intensity λ. The service time has a phase-type (PH) distribution with representation $(\beta,\ S)$ of order n and mean μ^{-1}, where $(\beta,\ \beta_{n+1})$is the initial probability of the process, and $\beta_{n+1} = 0$. The service time distribution is characterized by the infinitesimal generator $\begin{pmatrix} S & S^0 \\ 0 & 0 \end{pmatrix}$, and $Se + S^0 = 0$, where S is a square matrix with order n whose diagonal elements are negative and all other elements are non-negative; S^0 is a non-negative column vector with dimension n; and e is an appropriate dimensional column vector whose entries are all equal to 1 [25]. In this work, we focus on the establishment of probability models in this ANCS. We model each control application of the ANCS as a QBD process using a continuous-time *M/PH/1* queueing system with random-sized batch arrivals to analyze the uncertain response delays of the messages in this control application, and to obtain their probability density functions (*pdf*) and cumulative distribution functions (*cdf*) in early architecture exploration stages.

In automotive electronic systems, communication protocols consist of defining a deadline for messages and adopting a switching control strategy that depends on this deadline. If the message does not exceed the deadline, then a nominal controller is applied. The message is aborted if it exceeds the deadline. We generally assume that the message is dropped and use the previous input, i.e., utilize the zero-order-hold algorithm, because the information available to the controller is significantly delayed [5]. This technique may make the system unstable. Predicting the current delay before designing the control law is a feasible manner of compensating the current delay by the control law [8]. In this work, we predict the delay using grey system theory [26] in the ANCS. Probability transformed GM (1,1) (PTGM (1,1)) with two parameters, which is a novel grey model (1,1) (GM (1, 1)), is proposed to accurately predict the MRT for each message. In the PTGM (1,1) model, the obtained *cdfs* of the MRT are used.

To save system implementation costs, we must configure the shared communication bus in the ANCS. In ET communication protocols, the length of the DYN segment is specified for the number of *minislot* s. Thus, during the DYN segment, if no message is to be sent during a certain slot, then that slot will have a remarkably small length (equal to the length of a

minislot); otherwise, the length of the DYN slot will be equal to the number of *minislots* needed for transmitting the entire message [16]. In this study, we first configure the communication bus in the ANCS using the obtained parameter values of the system and the proposed bus protocol. A constrained optimization model is proposed to minimize the system cost function. An optimal *minislot* algorithm is presented to solve the optimization model and obtain the optimal value of decision variables *minslot*. The other parameter values of the communication bus, such as the length and *minislot* number of each bus cycle, are also yielded. According to the bus configuration, we can obtain the sample values of the MRTs of all messages and use these sample data to estimate the parameter values of the *cdf* of the MRT for each message. To the best of our knowledge, *our technique is the first that attempts to calculate the pdfs of MRTs for ACPSs with uncertain delay using an M/PH/1 queue model under consideration of optimal minislot and then forecasts the delay using grey system theory*.

1.3. Our contributions

The highlights of this study are as follows.

- We propose a constrained optimization model for decision variables *minslot* to minimize the system cost function that is established using the average number of customers in the *M/PH/1* queue. An optimal *minislot* algorithm based on the DFP variable-scale method [27,28] and a parameter estimation algorithm of *pdfs* of the MRTs are proposed to obtain the optimal values of the bus *minislot* and matrix parameters of PH distribution, respectively. We configure the shared communication bus in the ANCS using the obtained parameter values and a multi-hierarchical flexible TDMA/fixed-priority (FTDMA/FP) bus protocol that is proposed in this work. According to the bus configuration, we can obtain the sample values of the response times of all messages.
- We model each control application in the ANCS as a QBD process using a continuous-time *M/PH/1* queueing system with random-sized batch arrivals and analyze the uncertain response delays of the messages in the control application for early architecture exploration. The closed-form *pdf* of the MRTs for each control application is generated using the calculation method of the waiting time distribution in the QBD process. With use of the obtained sample values of the MRTs and the maximum likelihood estimation method, a parameter estimation algorithm based on DFP-method is presented to obtain the optimal value of the matrix parameter of the PH distribution for each message and to generate the corresponding *pdf* of each message.
- The probability transformed GM (1,1) (PTGM (1,1)) model with two parameters, which is a novel GM (1, 1) model, is proposed to forecast the response time for each message. A two-parameter optimization algorithm of PTGM (1,1) based on the DFP-method is presented to minimize the mean absolute percentage error (MAPE) [29] and obtain the optimal values of the two parameters. With the PTGM (1,1) model, the *cdfs* of the MRT can be predicted accurately.

For reader's convenience, we provide Table 1, which gives a summary of notations and their definitions in this paper.

The rest of this paper is organized as follows. Section 2 shows the platform architecture of the system and depicts the system dynamics. In Section 3, we model the control application in the ANCS using a continuous-time *M/PH/1* queueing system with random-sized batch arrivals and formulate certain assumptions. Section 4 configures the bus in the ANCS using an example. A constrained optimization model is established to minimize the expected cost function per unit time. An optimal *minislot* algorithm based on the DFP-method is proposed to derive the optimal value of the *minislot*. In Section 5, the *pdf* and *cdf* of uncertain delay are calculated. The fitting precision is evaluated. In Section 6, a PTGM (1,1) model with two parameters is proposed to forecast the response time for each message. A two-parameters optimal algorithm of PTGM (1,1) is presented to obtain the optimal values of the parameters. In Section 7, we describe the related works, and compare the related works with our work. Finally, Section 8 concludes the paper.

2. Platform architecture

We consider the typical DES architecture shown in Fig. 1. A platform architecture with multiple distributed control applications is shown in Fig. 2 (see Fig. 2 in [1]). The control applications are partitioned into many tasks that are mapped onto different PUs. All PUs communicate via the shared bus and run different tasks from one or more control applications. In the 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 forms the basis of the techniques we propose. We consider six control applications that are divided into two classes. The first class involves control applications 1 and 2, which are shown in Fig. 3 (see Fig. 3 in [1,30]). These applications are partitioned into four tasks: T_{Ri}, T_{Si}, T_{Ci}, and T_{Pi}, $i = 1, 2$. The second class contains control applications 3, 4, 5, and 6, which are shown in Fig. 4 (see Fig. 4 in [1,30]). These applications are partitioned into three tasks: T_{Sj}, T_{Cj}, and T_{Pj}, $j = 3, 4, 5, 6$.

In the platform architecture, the communication bus protocol is designed as a multi-hierarchical FTDMA/FP bus scheduling policy, as shown in Fig. 5. The upper-layer protocol follows a bus scheduling policy in which the bus in a cycle is occupied sequentially in a fixed priority order $E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_4 \rightarrow E_5 \rightarrow E_6$ for ECUs in all control applications, where E_i ($i = 1, 2, \ldots, 6$), as shown in Fig. 5. The middle-layer protocol adopts the first-come-first-service (FCFS) rule, which is applied to the message instances in the message queue with batch arrivals for each control application. The lower-layer protocol follows a hierarchical FTDMA/FP [16] bus scheduling policy with the ET protocol. The FTDMA technique was

Table 1
Summary of notations and definitions.

Notation	Definition	Notation	Definition
l	The length of bus cycle	ρ	The system utilization
λ_i	The average arrival rate of the ith subqueue	μ_i	The average service rate of the ith subqueue
(i, j)	A state in the *M/PH/1* queue	R	The rate matrix
$\Re_{ij}$	The jth service phase of the ith control application	$MsgID_i$	The set of messages of the ith control application
B_i	The maximal batch number of messages of the ith subqueue	ε_i	The transmission time of message m_i
$m_{i,j}$	The jth instance of message m_i	P_i	The fixed priority of message m_i
N_i	The size of a batch	λ_{ia}	The batch arrival rate
H	The hyperperiod	T_i	The period of message m_i
α_{ij}	The probability that an arbitrary batch consists of j messages in the ith subqueue	$E[N_i]$	The expectation of the random variable N_i.
ζ_i	The number of messages in the ith subqueue	ξ_i	The number of message instances in the ith subqueue
φ_{ij}	The instances number of the jth message in the ith subqueue	ξ	The number of bus cycles in one hyperperiod
MSNo	The number of the *minislot* of each bus cycle	$\hat{t}_{i,k,j}$	The sampling value of the jth instance for the kth message in the ith control application
$t^q_{h,i,j,s}$	The waiting time of the message instance $m_{i,j}$ transmitted in the sth bus cycle in the hth control application	$t_{h,i,j,s}$	The total end-to-end delay of the message instance $m_{i,j}$ transmitted in the sth bus cycle in the hth control application
κ_i	The WCRT of the ith message	$Z^{(1)}$	The background value array
$X^{(1)}$	The first-order accumulated generating operation sequence	α, γ	The two parameters in the background value $Z^{(1)}$
MRT	Message response time	WCRT	Worst-case response time
RTA	Response time analysis	DMR	Deadline miss ratio
PU	Processing unit	PH	Phase-type
TT	The time-triggered	ET	The event-triggered
ST	The static segment	DYN	The dynamic segment
pdf	Probability density function	cdf	Cumulative distribution function
LST	Laplace–Stieltjes transform	GM(1, 1)	Grey model (1,1)
ITGM(1,1)	Improved transformed GM(1,1)	PTGM(1,1)	Probability transformed GM(1,1)
MAPE	Mean absolute percentage error	RMSPE	Root mean square percentage error

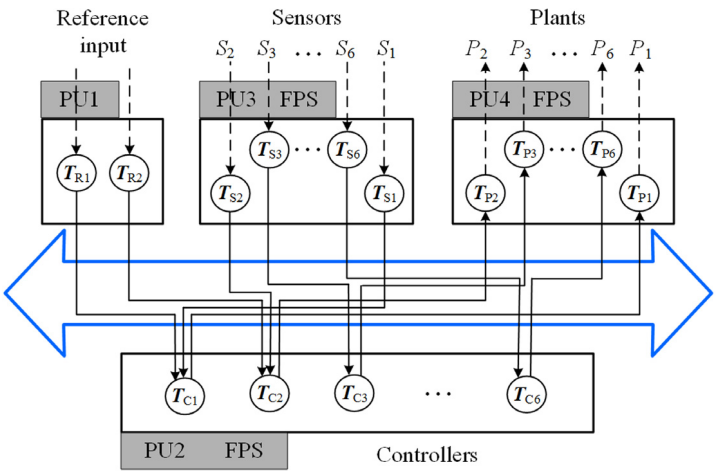


Fig. 2. System architecture with six distributed control applications [1].

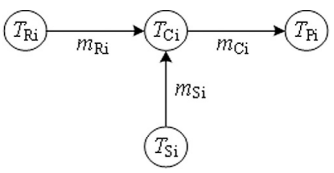


Fig. 3. Task graph for control application i, $i = 1, 2$ [1,30].

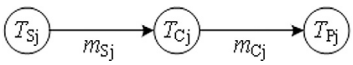


Fig. 4. Task graph for control application j, $j = 3, 4, 5, 6$ [1,30].

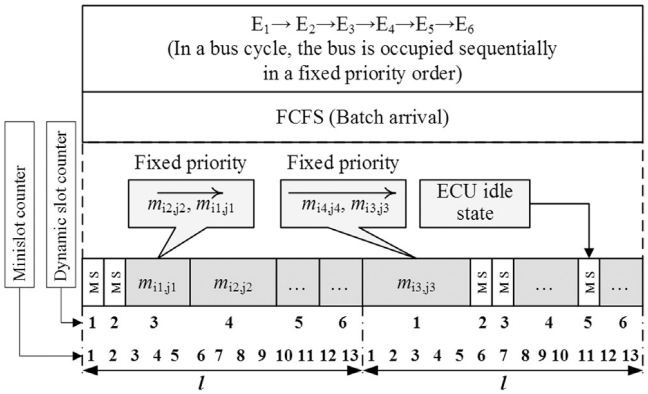


Fig. 5. Communication bus: multi-hierarchical FTDMA/FP scheduler, l = FTDMA cycle length.

conceived primarily to overcome the limitations of CAN and support high-performance real-time communications [31]. An FTDMA scheduler runs on the top-level of the scheduler, i.e., the communication bandwidth is divided into equal cycles with length l. All the messages coming into/from the control application, which are transmitted via the communication bus in DYN segment slot i, follow a fixed-priority scheduler (FPS). If no message is to be sent during a certain slot k, then slot k will have a remarkably small length equal to that of a *minislot*; otherwise, the length of slot k will be equal to the number of *minislot* s needed for transmitting the entire message, as shown in Fig. 5. Slot 5 has 2 *minislots* (10 and 11) in the first bus cycle when message m_{C5} is transmitted. However, in the second bus cycle, Slot 5 has only one *minislot* (marked "MS") when no message is sent. When a controller is in an idle state, the corresponding slot is denoted as MS, e.g., Controllers 2 and 3 are in an idle state in the second bus cycle [30].

With the purpose of configuring the bus and obtaining the parameter values of the *cdf* of the response time of each message, we use an *M/PH/1* queueing system to model the control application in the ANCS and to calculate the cost function of the system and the *cdf* of the MRTs, respectively.

3. Queueing model

3.1. Model description

In this section, the control application in the ANCS is modeled using a continuous-time *M/PH/1* queueing system with random-sized batch arrivals. In an *M/PH/1* queue, we assume $\beta_{n+1} = 0$, which means $\beta e = 1$. Hereinafter, e is assumed to be an appropriate dimensional column vector whose entries are equal to 1; I is an appropriate order identity matrix. Furthermore, the service rate and Laplace–Stieltjes transform (LST) of the service time are as follows:

$$\mu^{-1} = -\beta S^{-1} e, \quad f^*(s) = \beta(sI - S)^{-1} S^0. \tag{1}$$

In the ANCS platform architecture shown in Fig. 2, the c control applications sequentially share the forward bus from sensor to controller and the backward bus from controller to actuator on the basis of the communication protocol shown in Fig. 5. We abstract the forward bus and backward bus as the forward node T_{fwd} and backward node T_{bwd}, respectively. In the platform architecture, for each control application, a corresponding subqueue is generated by the sensor's sampling data and the reference command. The c subqueues are derived by c control applications, where $c = 6$. Hence, this platform architecture is abstracted as a multi-queue multi-server serial–parallel hybrid queueing system shown in Fig. 6. For the ith corresponding subqueue of the ith control application, we assume that the messages from a sensor arrive at the corresponding ECU_i in accordance with a Poisson process having rate λ_i ($\lambda_i > 0$); i.e., the time between successive arrivals are independent exponential random variables having the mean $\frac{1}{\lambda_i}$.

In the ith subqueue ($i = 1, 2, \ldots, 6$), each message goes directly into service upon arrival if server $\Re_{i1}$ (i.e., Sensor ECU, expressed as T_{Si} or E_i node) is idle; otherwise, the message joins the queue. When $\Re_{i1}$ finishes processing a message, the message is transmitted to server $\Re_{i2}$ (i.e., FlexRay/CAN bus, as T_{fwd} node) and then to $\Re_{i3}$ (i.e., Controller ECU, as T_{Ci} node) and $\Re_{i4}$ (i.e., FlexRay/CAN bus, as T_{bwd} node), and finally entered into the corresponding actuator (i.e., Actuator ECU, as T_{Pi}

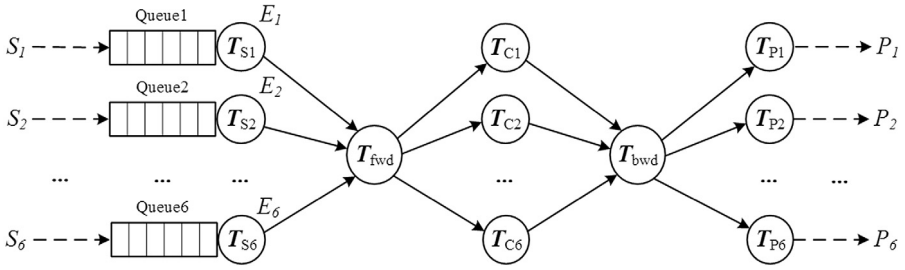


Fig. 6. Multi-queue multi-server serial–parallel hybrid queueing system for platform architecture.

node). The next message in line, if any, enters $\Re_{i1}$. Therefore, in the ith subqueue, each arriving message is successively served by the servers $\Re_{i1}$, $\Re_{i2}$, $\Re_{i3}$, and $\Re_{i4}$, and finally absorbed by the actuator ECU. The service time has a PH distribution. For convenience, we refer to a message as a *customer* and an ECU or bus as a *server* (e.g., $\Re_{ij}$, $i = 1, 2, \ldots, 6, j = 1, 2, 3, 4$).

We adopt the following assumptions.

A1: For every subqueue, messages arrive at server $\Re_{i1}$ (i.e., E_i) in accordance with a Poisson process. The arrival process of the messages from the user is assumed to be a Poisson process as well (e.g., in Control Applications 1 and 2). The sum of several independent Poisson processes is also a Poisson process. Hence, for the ith control application, the messages arrive according to a Poisson process with rate λ_i ($\lambda_i > 0$). The total arrival rate is $\lambda = \sum_{i=1}^{c} \lambda_i$. In Fig. 6, we omit the reference commands T_{R1} and T_{R2} for simplicity.

A2: For every subqueue, the service time in server $\Re_{ij}$ is an exponential random variable with mean service rate x_i ($x_i > 0$). After a customer is served in server $\Re_{ij}$, it immediately enters the next server $\Re_{i,j+1}$ until it is absorbed. A server can serve only one customer at a time.

A3: The actuators of control applications adopt high-performance processors. Once an actuator receives the controller's signals, it immediately drives the plant to perform the appropriate operation. The computing time of the actuator is ignored. Hence, in the queue model, the service time distribution is of phase-type with order four, i.e., $n = 4$. The state of an actuator is an absorbing state, and $\beta_{n+1} = 0$.

A4: The arriving customers in different batches are served on the basis of the FCFS policy. Customers in the same batch are served following a fixed priority.

A5: For each control application, all inter-arrival and service times are assumed to be mutually independent for the *M/PH/1* queue [30].

3.2. QBD process model

In [25], the *M/PH/1* queue was studied as a QBD process with the state space $\Omega = \{0\} \cup \{(i, j) : i \geq 1, 1 \leq j \leq n\}$, where the state $\mathbf{0}$ corresponds to an empty queue, the state (i, j) corresponds to having i ($i \geq 1$) customers in the system and the service process is in the phase j ($1 \leq j \leq n$).

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

$$R^2 S^0 \beta + R(-\lambda I + S) + \lambda I = 0,$$

and this solution R is called the rate matrix.

If $\rho = \lambda \mu^{-1} < 1$, then the matrix equation above has the minimal non-negative solution $R = \lambda(\lambda I - \lambda e\beta - S)^{-1}$ (see Theorem 3.2.1 in [25]). Obtaining the average number of customers in each control application is easy; i.e.,

$$E[L] = (1 - \rho)\beta R(I - R)^{-2} e. \tag{2}$$

According to Little's formula, the average time of customers in the control application is

$$E[\omega] = \frac{E[L]}{\lambda}.$$

3.3. Message queues on a reference bus

We adopt the message set on one of the CAN buses of an experimental vehicle in [24] as our test data. The test set consists of 6 ECUs and 69 messages, as shown in Table 2. All messages m_i in the system are indexed in fixed-priority order and shown with their periods and transmission times in milliseconds (ms). At each ECU node, the bus adapter enqueues the messages in fixed-priority (ID) order, and the message with the highest priority is always considered for arbitration. For each message m_i, T_i is the period, ε_i the transmission time, and P_i the fixed priority. For each periodic activation, a *message instance* arrives at the corresponding ECU node, and is *queued* by a periodic middleware task. We denote the jth instance of message m_i as

Table 2
An example automotive CAN system [24].

Msg	ECU	T_i	ε_i	P_i	Msg	ECU	T_i	ε_i	P_i	Msg	ECU	T_i	ε_i	P_i
m_1	E_2	10	0.27	1	m_{24}	E_3	25	0.23	24	m_{47}	E_2	50	0.19	47
m_2	E_2	10	0.27	2	m_{25}	E_3	25	0.25	25	m_{48}	E_4	100	0.27	48
m_3	E_3	5	0.19	3	m_{26}	E_2	20	0.27	26	m_{49}	E_3	100	0.27	49
m_4	E_3	10	0.25	4	m_{27}	E_5	25	0.27	27	m_{50}	E_1	100	0.27	50
m_5	E_1	10	0.19	5	m_{28}	E_2	20	0.27	28	m_{51}	E_3	100	0.13	51
m_6	E_6	10	0.27	6	m_{29}	E_1	25	0.17	29	m_{52}	E_3	100	0.27	52
m_7	E_1	100	0.27	7	m_{30}	E_2	10	0.21	30	m_{53}	E_1	100	0.19	53
m_8	E_1	100	0.15	8	m_{31}	E_2	20	0.27	31	m_{54}	E_3	100	0.13	54
m_9	E_1	100	0.17	9	m_{32}	E_4	10	0.27	32	m_{55}	E_3	100	0.13	55
m_{10}	E_5	25	0.27	10	m_{33}	E_3	10	0.27	33	m_{56}	E_5	100	0.27	56
m_{11}	E_1	100	0.15	11	m_{34}	E_3	10	0.27	34	m_{57}	E_3	100	0.25	57
m_{12}	E_1	20	0.19	12	m_{35}	E_1	25	0.25	35	m_{58}	E_3	100	0.13	58
m_{13}	E_2	100	0.19	13	m_{36}	E_6	25	0.23	36	m_{59}	E_3	100	0.13	59
m_{14}	E_1	100	0.19	14	m_{37}	E_4	50	0.27	37	m_{60}	E_4	100	0.17	60
m_{15}	E_5	100	0.27	15	m_{38}	E_4	50	0.27	38	m_{61}	E_1	100	0.27	61
m_{16}	E_1	10	0.23	16	m_{39}	E_5	50	0.27	39	m_{62}	E_1	100	0.13	62
m_{17}	E_2	100	0.25	17	m_{40}	E_3	50	0.21	40	m_{63}	E_3	100	0.19	63
m_{18}	E_2	100	0.27	18	m_{41}	E_4	50	0.27	41	m_{64}	E_1	100	0.27	64
m_{19}	E_2	50	0.25	19	m_{42}	E_6	50	0.27	42	m_{65}	E_1	100	0.13	65
m_{20}	E_3	10	0.27	20	m_{43}	E_2	50	0.27	43	m_{66}	E_1	100	0.13	66
m_{21}	E_6	10	0.27	21	m_{44}	E_5	100	0.27	44	m_{67}	E_2	50	0.27	67
m_{22}	E_6	25	0.27	22	m_{45}	E_2	25	0.15	45	m_{68}	E_1	100	0.13	68
m_{23}	E_3	25	0.23	23	m_{46}	E_2	50	0.19	46	m_{69}	E_4	100	0.27	69

Table 3
Average arrival rates, number of messages, and their instances for every subqueue in one hyperperiod.

Ctrl. App. ID	1	2	3	4	5	6
Arrival rates	0.46	0.62	0.83	0.19	0.13	0.3
MsgNo (ζ_i)	18	15	18	7	6	5
InstNo (ξ_i)	46	62	83	19	13	30

$m_{i,j}$. Each message m_i is associated with a unique ID P_i, which also represents its priority. According to the fixed-priority protocol, the smaller the ID number of a message, the higher its priority. The CAN bus rate is 500 kb/s and its utilization is 60.25%.

In Fig. 6, for the ith control application, the message instance $m_{k,j}$, which is periodically sampled by sensor, arrives at the corresponding node E_i with rate $\lambda_k = \frac{1}{T_k}$. Several message instances that have different sampling periods may simultaneously arrive at the node E_i in the sampling instant of common multiple of the message periods, and are queued in the ith subqueue with the fixed-priority order. Therefore, this queue model is an *M/PH/1* queue model with random-sized batch arrivals. The least common multiple of all message periods in the system is called a hyperperiod. Let H represent the hyperperiod. From Table 2, $H = 100$ ms.

For the ith control application, let α_{ij} $(1 \leq i \leq c, j \geq 1)$ denote the probability that an arbitrary batch consists of j messages. Let N_i denote a random variable representing the size of a batch, and $P\{N_i = j\} = \alpha_{ij}$. We let $E[N_i]$ represent the expectation of the random variable N_i. Then, $E[N_i] = \sum_{j=1}^{\infty} j\alpha_{ij}$. Suppose that batch arrivals occur in accordance with a Poisson process having rate λ_{ia}, and the average arrival rate of the ith subqueue is λ_i. Then, $\lambda_i = \lambda_{ia}E[N_i]$. In the ith subqueue, $MsgID_i$ is assumed to denote the set of messages, and B_i denotes the maximum arrival batch number of messages. Then, $B_i = max\left\{\frac{H}{T_k}, k \in MsgID_i\right\}$, and $\lambda_{ia} = \frac{B_i}{H}$. Hence, the average arrival rate of the ith subqueue can be calculated by

$$\lambda_i = \frac{1}{H} max\left\{\frac{H}{T_k}, k \in MsgID_i\right\} \sum_{j=1}^{\infty} j\alpha_{ij}.$$

For example, in the 3rd subqueue, $\lambda_{3a} = 0.2, j = 1, 4, 5, 9, 18$, and the corresponding probability $\alpha_{3j} = 0.4, 0.1, 0.4, 0.05, 0.05$. Then, $E[N_3] = 4.15$, $\lambda_3 = 0.83$. In the ith subqueue, we assume that ζ_i and ξ_i denote the number of messages and their instances, respectively; and φ_{ij} represents the number of instances of the jth message. Then, $\zeta_i = \|MsgID_i\|$, $\xi_i = \sum_{j \in MsgID_i} \varphi_{ij}$, and $\varphi_{ij} = \frac{H}{T_j}$. It is assumed that the number of bus cycles in one hyperperiod is denoted by ξ, then, $\xi = \max\{\xi_i\}$, $1 \leq i \leq c$. The average arrival rates, the number of messages, and their instances for different subqueues in one hyperperiod are shown in Table 3. The message queue for of the 3rd control application in one hyperperiod is shown in Fig. 7. The message queues for other control applications can also be drawn.

In the queue model, the service disciplines of customers follow Assumption A4 (Section 3.1). In the ith subqueue, the instance $m_{k,j}$ of message m_k is sequentially served by the servers $\Re_{i1}$, $\Re_{i2}$, $\Re_{i3}$, and $\Re_{i4}$. Without loss of generality, we assume that the total service time on the four servers for the message instance $m_{k,j}$ is the transmission time ε_k of message m_k.

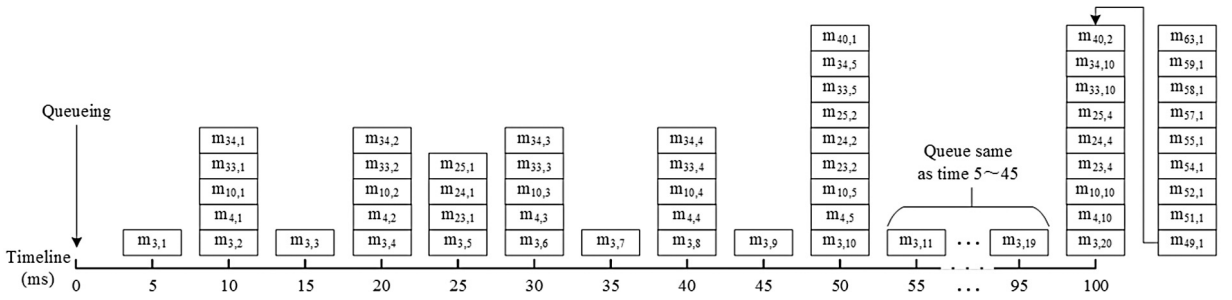


Fig. 7. Message queue for 3rd control application in one hyperperiod.

4. Bus configuration

4.1. Optimization model

In this section, we can numerically evaluate system cost using the average number of customers for the *M/PH/1* queue to yield the optimal *minislot*. We first develop the total expected cost function per customer per unit time for the system. For each control application, the corresponding *M/PH/1* queue is a single-server multi-service phase queueing system. The factors that affect system cost are primarily the number and the sojourn times of customers in the system. The cost parameters are as follows:

$C_h \equiv$ holding cost per unit time when a customer presents in the system;

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

Utilizing the above cost parameters and the concept of crew-service equipment by White et al. [32] and considering Little's formula, we represent the expected cost function per unit time under a complex and nonlinear cost structure by $F(\tau)$, where τ denotes the *minislot*. A constrained optimization model is established as follows:

Minimize

$$\begin{aligned} F(\tau) &= \sum_{i=1}^{c} \frac{\lambda_i}{\lambda} \left(C_h E[L_i] + C_d E[\omega_i]\right) \\ &= \sum_{i=1}^{c} \frac{\lambda_i}{\lambda} \left(C_h + \frac{1}{\lambda_i} C_d\right) E[L_i], \end{aligned} \tag{3}$$

subject to

$$\begin{cases} \gcd\{\varepsilon_i, 1 \le i \le \sigma\} < \tau \le \dfrac{H}{\xi c}, \\ \displaystyle\sum_{i=1}^{\delta} \varepsilon_i + \sum_{j=1}^{c} \tau \left(\left\lfloor \frac{H}{\lfloor H/(\xi\tau) \rfloor \tau} \right\rfloor - \xi_j \right) < H, \\ \xi = \max\{\xi_i, 1 \le i \le c\}, \\ \sigma = \displaystyle\sum_{i=1}^{c} \zeta_i, \delta = \sum_{i=1}^{c} \xi_i, \end{cases}$$

where the average number of customers $E[L_i]$ in the ith control application is obtained from Eq. (2), and $\gcd\{\varepsilon_i, 1 \le i \le \sigma\}$ represents the greatest common divisor of the message service times.

We aim to determine the optimal solution of decision variable τ for the optimization model. We discuss the parameter estimation issue for every control application and then solve the optimization problem described in (3). We assume that a customer is equally likely served at each service phase. Hence, the initial probabilities for any control application are represented as a row vector $\beta = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$. For any control application, the *pdf* of the PH distribution of the service time t $(t \ge 0)$ is given by

$$f(t) = \beta \exp(St) S^0. \tag{4}$$

From Assumption A2 (Section 3.1), let

$$S = \begin{pmatrix} -x_1 & x_1 & & \\ & -x_2 & x_2 & \\ & & -x_3 & x_3 \\ & & & -x_4 \end{pmatrix}.$$

According to the structure of the matrix S, the *pdf* represented by Eq. (4) is called a density function of bidiagonal PH distribution [33]. Then, in Eq. (4), the values of parameters with respect to column vector $\vec{x} = (x_1, x_2, x_3, x_4)^T$ need to be estimated using the maximum likelihood estimation method, where T denotes the transpose. The sampling data of service time come from column ε_i in Table 2. The logarithmic likelihood function for Eq. (4) is written as

$$\psi(\vec{x}) = \sum_{k \in MsgID_i} \sum_{j=1}^{\varphi_{ik}} \log f(\hat{t}_{i,k,j}; \vec{x}), \tag{5}$$

where in the ith control application, $MsgID_i$ denotes the set of message IDs; $\hat{t}_{i,k,j}$ represents the sampling value of the jth instance for the kth message, and $\hat{t}_{i,k,j} = \varepsilon_k$, for any $j \in [1, \varphi_{ik}]$. The gradient of the above equation with respect to $\vec{x}$ is given by

$$\nabla\psi(\vec{x}) = \frac{d\psi(\vec{x})}{d\vec{x}} = \sum_{k \in MsgID_i} \sum_{j=1}^{\varphi_{ik}} \frac{1}{f(\hat{t}_{i,k,j}; \vec{x})} \frac{df(\hat{t}_{i,k,j}; \vec{x})}{d\vec{x}}. \tag{6}$$

4.2. Bus configuration algorithms

The bus configuration algorithms comprise the parameter estimation algorithm that is based on the DFP-method [27,28] and the optimal *minislot* algorithm.

Algorithm 1 Parameter estimation algorithm based on DFP-method

1: Set initial value of vector $\vec{x}_1$, and tolerance $\varepsilon = 10^{-6}$;
2: $H_1 = I$; $k = 1$; Calculate the gradient $\vec{g}_1 = \nabla\psi(\vec{x}_1)$ by Eq. (6);
3: **while** $\left\|\nabla\psi(\vec{x}_k)\right\| > \varepsilon$ **do**
4: Set $\vec{d}_k = -H_k\vec{g}_k$;
5: Search along the direction of $\vec{d}_k$ line search starting from $\vec{x}_k$, and obtain step θ_k such that $\psi(\vec{x}_k+\theta_k\vec{d}_k) = \min_{\theta\geq 0} \psi(\vec{x}_k+\theta\vec{d}_k)$;
6: Set $\vec{x}_{k+1} = \vec{x}_k + \theta_k\vec{d}_k$;
7: **if** $k = n$ **then**
8: $\vec{x}_1 = \vec{x}_{k+1}$;
9: $H_1 = I$; $k = 1$; Calculate $\vec{g}_1 = \nabla\psi(\vec{x}_1)$ by Eq. (6);
10: **else**
11: $\vec{g}_{k+1} = \nabla\psi(\vec{x}_{k+1})$;
12: $\vec{p}_k = \vec{x}_{k+1} - \vec{x}_k$;
13: $\vec{q}_k = \vec{g}_{k+1} - \vec{g}_k$;
14: $H_{k+1} = H_k + \frac{\vec{p}_k\vec{p}_k^T}{\vec{p}_k^T\vec{q}_k} - \frac{H_k\vec{q}_k\vec{q}_k^T H_k}{\vec{q}_k^T H_k\vec{q}_k}$;
15: $k = k + 1$;
16: **end if**
17: **end while**
18: $\vec{x} = \vec{x}_k$;
19: **return** $\vec{x}$.

In Algorithm 1, let $\vec{x}_1 = (1, 1, 1, 1)^T$, $n = 4$, and I represent an appropriate order identity matrix. The main process of this algorithm comes from the DFP algorithm. Algorithm 1 mainly calculates the optimal value of column vector $\vec{x} = (x_1, x_2, x_3, x_4)^T$ to maximize the logarithmic likelihood function. To solve the optimal value of *minislot*, we first obtain the ranges of *minislot* from the constraint conditions in the Optimization Model (3). In the ranges, we search the optimal solution for Model (3). The optimal *minislot* algorithm is depicted by Algorithm 2.

In Algorithm 2, to guarantee that all message instances can be transmitted, we select the maximum number of message instances for c control applications as the number of bus cycles in one hyperperiod (Steps 3 and 4). We compute the lower and upper bounds of decision variable τ from the constraint conditions in Optimization Model (3) to obtain the ranges of *minislot*. In these ranges, for each control application, we calculate the *minislot* number in each bus cycle and the total

Algorithm 2 Optimal *minislot* algorithm

For each control application i ($i = 1, 2, \cdots, c$), count the message IDs to get the set $MsgID_i$, and let $\zeta_i = \|MsgID_i\|$;
Compute gcd $\{\varepsilon_i, 1 \le i \le \sigma\}$, and the hyperperiod H= lcm $\{T_i, 1 \le i \le \sigma\}$, where $\sigma = \sum_{i=1}^{c} \zeta_i$, and *lcm* represents least common multiple;
Calculate $\xi_i = \sum_{j \in MsgID_i} \varphi_{ij}$, $\varphi_{ij} = \frac{H}{T_j}$, and $\xi = \max\{\xi_i\}$, $1 \le i \le c$;
Calculate the length of bus cycle $\ell = \frac{H}{\xi}$;
for each control application i **do**
  compute the arrival rate λ_i, $i = 1, 2, \cdots c$;
end for
Compute the lower bound a and upper bound b of decision variable τ from the constraint conditions in the optimization model (3), and set $C_h = C_d = 1$;
for $\tau = a$ to b step h **do**
  for each control application i **do**
    Calculate the *minislot* number (*MSNo*) in each bus cycle, i.e., $MSNo = \lfloor \frac{\ell}{\tau} \rfloor$;
    Calculate the total number of idle *minislot* in one hyperperiod, i.e., $\nu_i = \xi - \xi_i$;
    Update $MsgID_i$ using $MsgID_i \cup \{0\}$, and set $\varphi_0 = \nu_i$, $\varepsilon_0 = \tau$;
    For each $k \in MsgID_i$ and $j \in [1, \varphi_{ik}]$, if mod$(\varepsilon_k, \tau) = 0$, then $\hat{t}_{i,k,j} = \varepsilon_k$; else $\hat{t}_{i,k,j} = \left(\lfloor \frac{\varepsilon_k}{\tau} \rfloor + 1\right) \tau$;
    Call Algorithm 1 using parameter $\hat{t}_{i,k,j}$ to obtain $\vec{x}_i$;
    Calculate $E[L_i]$ by Eq. (2);
  end for
  Calculate $F(\tau)$ by Eq. (3);
end for
Select $\tau_{\min}$ such that $F(\tau_{\min})$ is minimum;
return $\tau_{\min}$, $F(\tau_{\min})$.

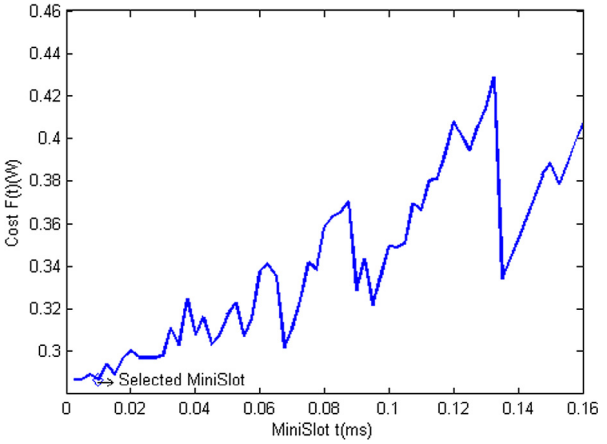


Fig. 8. Relationship of *minislot* and system cost.

number of idle *minislots* in one hyperperiod. We use Algorithm 1 to obtain $\vec{x}_i$. In Step 13, we assume that in the ith control application, a *virtual* message with message ID zero, service time τ, and instance number ν_i is transmitted in an idle *minislot*. Hence, in the calculation of system costs, the cost yielded by the idle *minislot* is also considered.

4.3. Experiment results

In our experiment, we obtain $\tau \in [0.01, 0.151)$ from Model (3), set $a = 0.0025$ and $b = 0.16$, and let step $h = 0.0025$. Then, we implement Algorithm 2 to obtain the optimal solution of Optimization Model (3), i.e., $\tau_{\min} = 0.01$ ms, $F(\tau_{\min}) = 0.2868w$. The length l and the *MSNo* of each bus cycle are also obtained, which are 1.2 ms and 120, respectively. The relationship of the *minislot* and system cost is shown in Fig. 8.

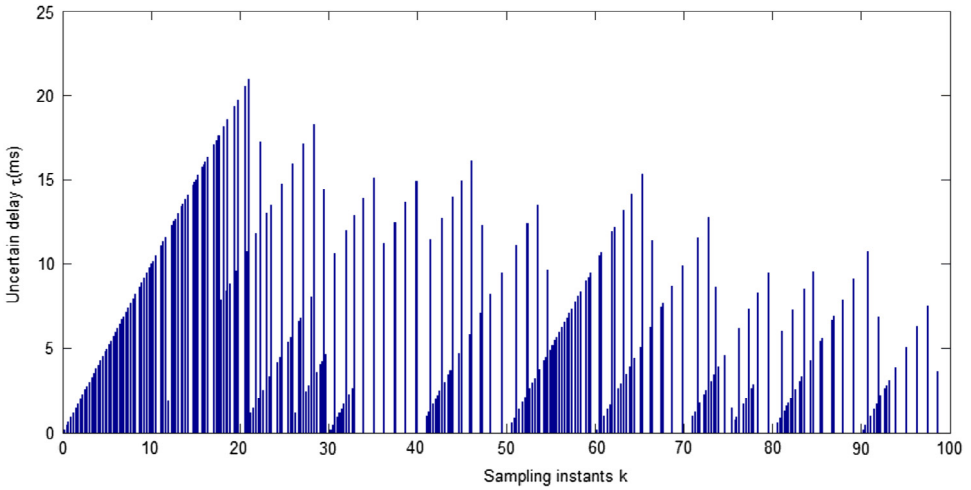


Fig. 9. Uncertain delay over reference system.

According to the experiment results, $\psi(\vec{x})$ in Eq. (5) is maximum when $x_1 = x_2 = x_3 = x_4$. Without loss of generality, we let $S = xS_0$, where

$$S_0 = \begin{pmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \\ & & & -1 \end{pmatrix}.$$

Then, $\vec{x} = (x, x, x, x)$. For the six control applications, the values of x can all be generated, i.e., $x = 19.6872$, 13.8126, 12.5334, 26.0310, 33.4069, and 19.3936. Hence, the probability density represented by Eq. (4) is a density function of canonical PH distribution [33,34].

In one hyperperiod, suppose that message instance $m_{i,j}$ in the hth ($h \leq c$) control application is transmitted in the sth($1 \leq s \leq \xi$) bus cycle. The waiting time of $m_{i,j}$ in the hth subqueue is given by

$$t^q_{h,i,j,s} = (s-1)\ell + \sum_{v=1}^{h-1} \left\lceil \frac{\varepsilon_{v,s}}{\tau} \right\rceil \tau - jT_i,$$

where $\varepsilon_{v,s}$ represents the end-to-end service time of the corresponding message instance in the sth bus cycle in the vth control application. If the sth bus cycle has an idle *minislot*, then $\varepsilon_{v,s} = \tau$. The total end-to-end delay is written by

$$t_{h,i,j,s} = t^q_{h,i,j,s} + \varepsilon_{h,s}. \tag{7}$$

We may assume that the sampling instant consists of the total end-to-end delay and the arrival instant for message instance $m_{i,j}$, i.e., $k = t_{i,j,s} + jT_i$. We can obtain the uncertain end-to-end delay of all message instances over the reference system, which is shown in Fig. 9.

5. Probability distribution of uncertain delay

5.1. Probability density function (pdf)

We consider the length of time a customer waits before entering service under the FCFS discipline and the length of uncertain MRT, which consists of the preceding waiting time and the service time of the customer. We denote the stationary distribution of the two kinds of waiting times as $W_q(t)$ and $W(t)$, respectively, and their LSTs as $W^*_q(s)$ and $W^*(s)$, respectively. According to the *Pollaczek–Khintchine transform equation* [35], the LST of the *cdf* of the MRT for each control application is initially obtained. Then, the integral form of the *cdf* is calculated using the calculation method of the waiting time distribution in the QBD process. By integral transformation, the closed expression of the *cdf* is obtained.

Theorem 1. *The LSTs of the cdf of the MRT for each control application are written as*

$$W^*(s) = (1-\rho)\beta \sum_{k=0}^{\infty} \lambda^k [(sI - S)^{-1} e\beta]^k (sI - S)^{-1} S^0 \tag{8}$$

for Re $s \geq 0$.

Proof. According to the *Pollaczek–Khintchine transform equation*, we obtain

$$W_q^*(s) = \frac{(1-\rho)s}{s-\lambda+\lambda f^*(s)}. \tag{9}$$

Eq. (9) can be written as

$$W_q^*(s) = (1-\rho)\sum_{k=0}^{\infty}\left(\frac{\lambda(1-f^*(s))}{s}\right)^k. \tag{10}$$

The following equation holds (see p. 86 of [25]):

$$f^*(s) = 1 - s\beta(sI-S)^{-1}e. \tag{11}$$

By substituting (11) into (10), we can obtain

$$W_q^*(s) = (1-\rho)\sum_{k=0}^{\infty}[\lambda\beta(sI-S)^{-1}e]^k. \tag{12}$$

The sojourn time of the customer in the system is equal to the sum of the preceding waiting time and the service time, according to Theorem 2.9.3(d) in [35]; hence, we must obtain

$$W^*(s) = W_q^*(s)f^*(s). \tag{13}$$

With substitution of (1) and (12) into (13), Eq. (8) can be generated. □

Theorem 2. *The cdf of the MRT for each control application is given by*

$$W(t) = (1-\rho)\beta\int_0^t \Phi(u)\exp(-uS)du\cdot\exp(tS)S^0, \tag{14}$$

where $t \geq 0$,

$$\Phi(t) = -V^0S - V^0\exp[t(-V^0)^{-1}]\lambda e\beta, \tag{15}$$

and

$$V^0 = -(\lambda e\beta + S)^{-1}. \tag{16}$$

Proof. From (8), we let $X = \lambda I$ and consider the matrix

$$\Phi^*(s) = \sum_{k=0}^{\infty}X^k[(sI-S)^{-1}e\beta]^k. \tag{17}$$

Eq. (17) can be written as

$$\Phi^*(s) = I + X\Phi^*(s)(sI-S)^{-1}e\beta. \tag{18}$$

We define $\Upsilon[C]$ to be the $(m_1 \times m_2)$-dimensional vector obtained by forming the *direct sum* of the rows of matrix C with dimensions $m_1 \times m_2$ [25]. If we let $\phi^*(s) = \Upsilon[\Phi^*(s)]$ and $\upsilon = \Upsilon[I]$, then Eq. (18) can be denoted as follows:

$$\phi^*(s) = \upsilon + \phi^*(s)[X^T \otimes (sI-S)^{-1}e\beta], \tag{19}$$

where X^T is the transposed matrix of X and $\otimes$ is the Kronecker product operator. With use of the classical properties of Kronecker products, Eq. (19) may be transformed as

$$\phi^*(s) = \upsilon + \upsilon(sI\otimes I - I\otimes S - X^T\otimes e\beta)^{-1}(X^T\otimes e\beta). \tag{20}$$

We let $\Phi(t)$ be the matrix of mass-functions with LST $\Phi^*(s)$ and set $\phi(t) = \Upsilon[\Phi(t)]$ for $t \geq 0$. Upon inversion of the transforms, Eq. (20) leads to

$$\begin{aligned}\phi(t) &= \upsilon + \upsilon\int_0^t \exp[(I\otimes S + X^T\otimes e\beta)u]du(X^T\otimes e\beta)\\ &= \upsilon + \upsilon(I\otimes S + X^T\otimes e\beta)^{-1}[\exp(I\otimes S\\ &\quad + X^T\otimes e\beta)t - I\otimes I](X^T\otimes e\beta).\end{aligned} \tag{21}$$

Let V^0 be the square matrix of order m for which

$$\Upsilon[V^0] = -\upsilon(I\otimes S + X^T\otimes e\beta)^{-1} = \upsilon^0.$$

Then, V^0 is the unique solution to

$$V^0 S + X V^0 e\beta = -I. \tag{22}$$

With substitution of $X = \lambda I$ into (22), Eq. (16) is obtained.
Let the v^2-dimensional vector $\theta(t)$ be defined by

$$\theta(t) = v^0 \exp[(I \otimes S + X^T \otimes e\beta)t],\ t \geq 0, \tag{23}$$

and let $\Upsilon[\Theta(t)] = \theta(t)$, then (23) shows that $\theta(t)$ satisfies the differential equation

$$\dot{\theta}(t) = \theta(t)(I \otimes S + X^T \otimes e\beta),\ t \geq 0$$

with $\theta(0) = v^0$, which is equivalent to the matrix differential equation

$$\dot{\Theta}(t) = \Theta(t)S + X\Theta(t)e\beta,\ t \geq 0, \tag{24}$$

which has initial condition $\Theta(0) = V^0$.
By substituting $X = \lambda I$ into (24), we can obtain

$$\Theta(t) = V^0 \exp[t(-V^0)^{-1}]. \tag{25}$$

From (21), the matrix $\Phi(t)$ is given by

$$\Phi(t) = I + XV^0 e\beta - X\Theta(t)e\beta. \tag{26}$$

From (22), (25) and (26), Eq. (15) is obtained.
Consider the matrix

$$\Psi^*(s) = \Phi^*(s)(sI - S)^{-1}. \tag{27}$$

The corresponding matrix $\Psi(t)$ satisfies the matrix differential equation

$$\dot{\Psi}(t) = \Psi(t)S + \Phi(t),\ t \geq 0 \tag{28}$$

with $\Psi(0) = 0$. From (28), we can easily attain

$$\Psi(t) = \int_0^t \Phi(u)\exp(-uS)du \cdot \exp(tS). \tag{29}$$

From (8), (27), and (28), the *cdf* $W(t)$ is written as

$$W(t) = (1-\rho)\beta\Psi(t)S^0. \tag{30}$$

With substitution of (29) into (30), Eq. (14) is generated. □

Lemma 1. *Suppose that S is a nonsingular matrix and t an arbitrary real number. The matrix equations below hold.*
(i) $S\exp(tS) = \exp(tS)S$, *and* $S^{-1}\exp(tS) = \exp(tS)S^{-1}$.
(ii) $\exp(t\lambda e\beta) = I - e\beta + \exp(t\lambda)e\beta$.

Proof. According to matrix series, by direct calculation, the above equations hold. □

Theorem 3. *The cdf and pdf of the MRT for each control application are represented as*

$$\begin{aligned} W(t) &= 1 + \beta\left(S^{-1} - tI - \mu^{-1}tS\right)\exp(tS)S^0 \\ &\quad + \frac{1}{\lambda}(\beta - \rho\beta + \mu^{-1}\beta S)[\exp(t\lambda) - 1]\exp(tS)S^0 \end{aligned} \tag{31}$$

and

$$\begin{aligned} p(t) &= (1-\rho)\beta\exp\left[t\left(\lambda I + S\right)\right]S^0 \\ &\quad + \left[\frac{1}{\lambda}\exp(t\lambda) - \frac{1}{\lambda} - t\right](\beta + \mu^{-1}\beta S)\exp(tS)SS^0, \end{aligned} \tag{32}$$

respectively, where $t \geq 0$.

Proof. By substituting (16) into (15), we obtain

$$\begin{aligned} \Phi(t) &= (\lambda e\beta + S)^{-1}S + \exp[t(\lambda e\beta + S)] \\ &\quad - (\lambda e\beta + S)^{-1}\exp[t(\lambda e\beta + S)]S. \end{aligned} \tag{33}$$

Table 4
Estimated value of corresponding parameter x_i for message ID i and corresponding WCRT.

ID	λ_i	μ_i	x_i	κ_i	ID	λ_i	μ_i	x_i	κ_i	ID	λ_i	μ_i	x_i	κ_i
1	0.10	0.61	1.5316	11.6	24	0.04	0.17	0.4155	39.9	47	0.02	0.11	0.2832	54.0
2	0.10	0.44	1.1087	16.2	25	0.04	0.15	0.3866	43.1	48	0.01	0.21	0.5315	26.8
3	0.20	0.32	0.7895	27.5	26	0.05	0.30	0.7464	22.3	49	0.01	0.11	0.2727	51.6
4	0.10	0.24	0.6070	31.1	27	0.04	0.33	0.8175	19.9	50	0.01	0.10	0.2588	54.4
5	0.10	0.49	1.2126	14.7	28	0.05	0.25	0.6337	26.5	51	0.01	0.10	0.2514	56.0
6	0.10	0.36	0.9062	20.0	29	0.04	0.29	0.7295	22.3	52	0.01	0.09	0.2326	60.5
7	0.01	0.81	2.0187	7.5	30	0.10	0.30	0.7596	24.3	53	0.01	0.10	0.2391	58.8
8	0.01	0.47	1.1706	12.6	31	0.05	0.20	0.5042	33.7	54	0.01	0.09	0.2163	65.1
9	0.01	0.32	0.7973	18.2	32	0.10	0.97	2.4194	7.3	55	0.01	0.08	0.2037	69.1
10	0.04	0.51	1.2708	12.8	33	0.10	0.21	0.5199	37.2	56	0.01	0.13	0.3295	42.8
11	0.01	0.24	0.6114	23.4	34	0.10	0.20	0.4905	39.8	57	0.01	0.08	0.1882	74.9
12	0.05	0.37	0.9363	17.7	35	0.04	0.24	0.6087	26.8	58	0.01	0.07	0.1801	78.3
13	0.01	0.44	1.0906	13.5	36	0.04	0.18	0.4470	37.0	59	0.01	0.07	0.1709	82.5
14	0.01	0.17	0.4157	34.1	37	0.02	0.49	1.2331	12.5	60	0.01	0.18	0.4463	31.8
15	0.01	0.34	0.8456	17.2	38	0.02	0.35	0.8700	17.5	61	0.01	0.09	0.2204	63.8
16	0.10	0.33	0.8173	22.4	39	0.02	0.20	0.4881	31.0	62	0.01	0.08	0.2073	67.9
17	0.01	0.30	0.7488	19.3	40	0.02	0.11	0.2790	54.8	63	0.01	0.07	0.1629	86.7
18	0.01	0.23	0.5807	24.6	41	0.02	0.26	0.6594	23.0	64	0.01	0.08	0.1931	72.9
19	0.02	0.25	0.6333	23.9	42	0.02	0.10	0.2569	59.7	65	0.01	0.07	0.1835	76.8
20	0.10	0.23	0.5658	33.7	43	0.02	0.15	0.3645	41.7	66	0.01	0.07	0.1739	81.1
21	0.10	0.31	0.7719	23.8	44	0.01	0.17	0.4301	33.0	67	0.02	0.10	0.2612	58.7
22	0.04	0.20	0.4899	33.6	45	0.04	0.21	0.5162	31.8	68	0.01	0.07	0.1654	85.3
23	0.04	0.18	0.4558	36.2	46	0.02	0.12	0.3045	50.1	69	0.01	0.15	0.3766	37.5

By substituting (23) into (33) and from Lemma 1, we attain

$$\begin{aligned}&(1-\rho)\beta\Phi(t)\exp(-tS)\\ =&\beta\exp(-tS)+(1-\rho)\beta\exp(t\lambda)+\mu^{-1}\beta S\exp(t\lambda)-\beta-\mu^{-1}\beta S.\end{aligned}\tag{34}$$

From (14) and (34), we can obtain Eq. (31).
From (31), the derivation for t, Eq. (32) is generated. □

5.2. WCRT of each message

In each control application, the *cdf* and *pdf* of the response times of each message can be calculated using Eqs. (31) and (32), respectively. We first consider the parameter estimation issue of the *pdf* of the response times of each message. In the hth control application, for the ith message, let $\beta = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$ and $S_i = x_i S_0$. The sampling data $t_{h,i,j,s}$ of the uncertain end-to-end time for each message instance come from Eq. (7) and are shown in Fig. 9. We assume that these sampling data are drawn from independent and identically distributed random variables. In Eq. (32), the value of parameter x_i with respect to $S_i = x_i S_0$ must be estimated using the maximum likelihood estimation method. The logarithmic likelihood function for Eq. (32) is written as

$$\psi(x_i) = \sum_{j=1}^{\varphi_{hi}} \log p(t_{h,i,j,s}; x_i),\tag{35}$$

where $1 \le h \le c$, and $1 \le s \le \xi$.

The derivative of the above equation with respect to x_i is given by

$$\nabla\psi(x_i) = \frac{d\psi(x_i)}{dx_i} = \sum_{j=1}^{\varphi_{hi}} \frac{1}{p(t_{h,i,j,s}; x_i)} \frac{dp(t_{h,i,j,s}; x_i)}{dx_i}.\tag{36}$$

Deriving the closed form of Eq. (36) directly is difficult because Eq. (35) is a complicated nonlinear function. We use the symbolic computation method in MATLAB to discuss the parameter estimation problem. The function *Jacobian()* in MATLAB is adopted to derive the symbolic expression of Eq. (36). For each message, the value of parameter x_i is estimated in Eq. (35) through the implementation of Algorithm 1; i.e., for the ith message, the estimated value of the corresponding parameter x_i is generated through the application of Algorithm 1. The estimated value of parameter x_i for the corresponding probability density of the ith message is shown in Table 4.

For the ith message in Table 4, the arrival rate $\lambda_i = \frac{1}{T_i}$ and the service rate $\mu_i^{-1} = -\beta S_i^{-1} e$. x_i is the element of the matrix S_i that is the matrix parameter of the *pdf* of the response times of the ith message, and $S_i = x_i S_0$. According to the estimated value of parameter x_i, the closed-form expression of the *pdf* of the response times of the ith message is obtained. By experiments, we find a non-negative real root in the equation $p_i(t) = 0$, where $p_i(t)$ is the *pdf* of the response times of

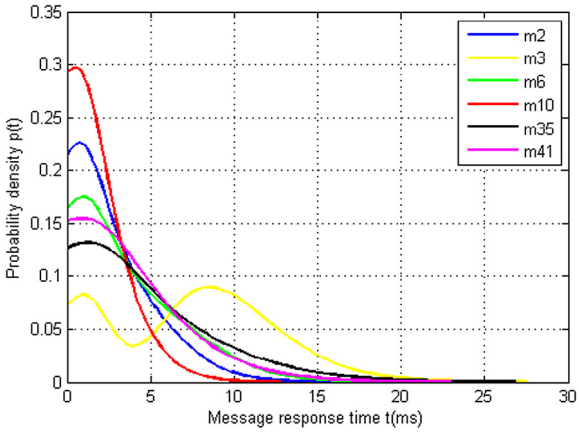


Fig. 10. Response time *pdfs* of partial messages that come from different control applications.

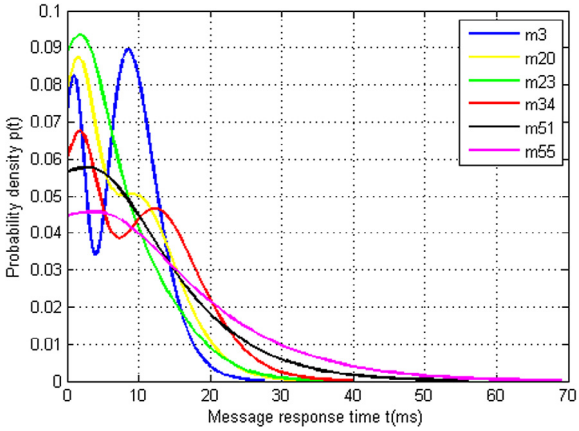


Fig. 11. Response time *pdfs* of partial messages that come from 3rd control application.

the ith message. The root is the upper bound of the response times, i.e., WCRT κ_i. Therefore, the WCRT κ_i of the ith message is accurately obtained. From the same table, we find that the larger the parameter x_i, the smaller the WCRT κ_i. In general, as the parameter x_i increases, the service rate μ_i presents volatility but shows an increasing trend. The system utilization ρ_i of each message can be obtained using $\rho_i = \frac{\lambda_i}{\mu_i}$, and $\rho_i < 1$. The *pdfs* of the response times of partial messages that come from different and the same (3rd) control applications are shown in Figs. 10 and 11, respectively.

5.3. Evaluating fitting accuracy

We use the quantile–quantile (Q–Q) plot to graphically compare the probability distributions from the simulation data and the fitted model [24]. If the compared distributions are similar, then the points in the Q–Q plot will lie approximately on the line. One thousand random samples are obtained from each distribution, and the results for messages m_5 and m_{69} (high- and low-priority cases) are shown in Figs. 12 and 13, respectively. The Q–Q plots for these messages are approximately linear, with a high relative accuracy in the case of the low-priority message m_{69}. The fitting accuracy in our results is also higher than that in [24].

We employ several metrics to assess the accuracy of the fitted distribution and thus quantitatively evaluate the fitting accuracy that the *cdf* of the MRT can be approximated by Eq. (31). The Monte Carlo simulation method is used to obtain the simulation data of the *cdf* of the MRT after 10,000 random samplings. From the parameters values in Table 4, we compute the statistic analysis indices, such as the root mean squared error (RMSE), the coefficient of determination (R^2), and the Kolmogorov–Smirnov statistic (K–S statistic), as shown in Table 5. The maximum RMSE is 0.0068, the minimum R^2 is 0.99939, and the maximum K–S statistic is 0.017. Compared with the results in Table 2 in [24], ours are more accurate for early estimation.

In our work, the analysis of MRT based on the queueing model is a stochastic method. In [24], the authors compared the run-time information for the example system shown in Table 1 and indicated that stochastic analysis was more than 10^7

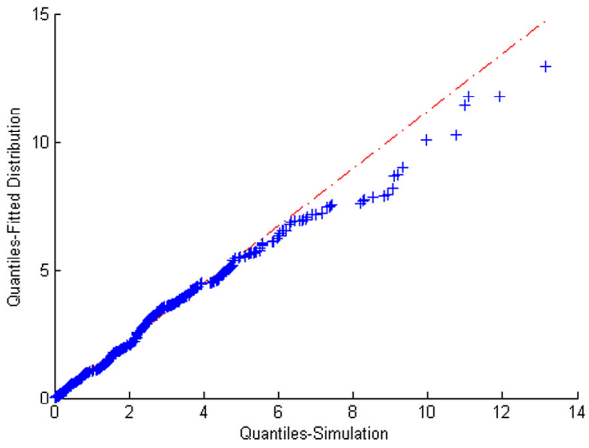


Fig. 12. Quantile–quantile plot of samples from simulation and fitted distribution for m_5.

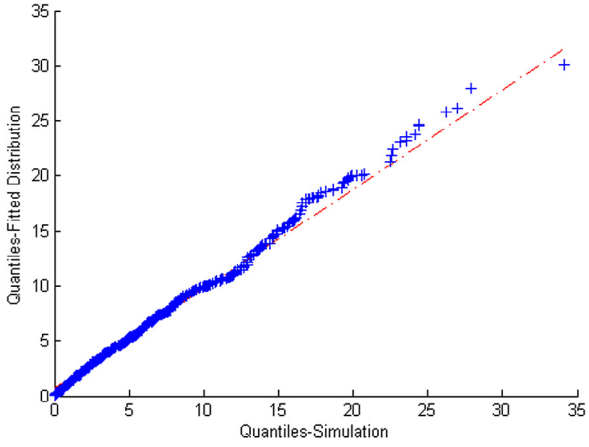


Fig. 13. Quantile–quantile plot of samples from simulation and fitted distribution for m_{69}.

times slower than statistical analysis. Therefore, stochastic analysis is significantly slower. In our experiments, the run-time information for the example system in Table 2 is also obtained. For example, the run-time values of the parameter values of the *pdfs* for the messages m_3, m_5, m_{12}, and m_{69} are 65.17, 39.71, 27.53, and 11.75s, whose periods are 5, 10, 20, and 100 ms, respectively. Moreover, the run-time values of the parameter values of the *pdfs* for all messages in Table 2 are within the interval (10, 70). Hence, from our experiment results and Table 8 in [24], we find that stochastic analysis is approximately five orders of magnitude slower than statistical analysis. However, stochastic analysis is more accurate than statistical analysis when the complete message set is available. We can arrive at this conclusion by comparing Table 5 in the current work with Table 2 in [24].

6. Predicting MRTs

6.1. Background values with two parameters in GM (1,1)

Grey system theory deals with uncertainty issues that feature limited data and imperfect information [26]. GM (1,1), which is a grey forecasting model, has been implemented in many forecasting problems (see [36] and references therein), such as the prediction of electric load, tourism flow, airline industry, automobile production, wave, wind speed and power, per capita annual net income forecast of rural households, fuel production, carbon emissions, energy demand, and economic growth, and so on. In this study, the ANCS has the characteristics of a grey system for uncertain delay. However, the traditional GM (1,1) cannot be directly employed in the ANCS due to delay uncertainty.

In the traditional GM (1,1) model [36,37], $X^{(0)}$ is assumed to be the initial sequence, which can be illustrated as

$$X^{(0)} = \left\{x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n)\right\}.$$

Table 5
Statistics of fitted distributions for messages on ANCS.

ID	RMSE	R^2	K–S	ID	RMSE	R^2	K–S	ID	RMSE	R^2	K–S
1	0.0013	0.99998	0.005	24	0.0032	0.99986	0.008	47	0.0016	0.99996	0.006
2	0.0024	0.99992	0.006	25	0.0025	0.99992	0.006	48	0.0041	0.99977	0.010
3	0.0024	0.99995	0.007	26	0.0014	0.99997	0.004	49	0.0036	0.99982	0.010
4	0.0029	0.99990	0.008	27	0.0032	0.99986	0.009	50	0.0029	0.99989	0.008
5	0.0022	0.99993	0.008	28	0.0016	0.99997	0.005	51	0.0022	0.99994	0.007
6	0.0036	0.99982	0.009	29	0.0034	0.99984	0.009	52	0.0025	0.99992	0.009
7	0.0019	0.99995	0.006	30	0.0068	0.99939	0.017	53	0.0028	0.99989	0.009
8	0.0030	0.99988	0.007	31	0.0030	0.99988	0.010	54	0.0025	0.99992	0.007
9	0.0025	0.99991	0.008	32	0.0027	0.99989	0.009	55	0.0015	0.99997	0.005
10	0.0016	0.99997	0.005	33	0.0028	0.99991	0.008	56	0.0025	0.99991	0.009
11	0.0019	0.99995	0.007	34	0.0040	0.99983	0.013	57	0.0026	0.99991	0.010
12	0.0027	0.99989	0.008	35	0.0040	0.99977	0.012	58	0.0038	0.99981	0.012
13	0.0024	0.99992	0.006	36	0.0020	0.99994	0.006	59	0.0031	0.99988	0.009
14	0.0034	0.99984	0.010	37	0.0011	0.99998	0.004	60	0.0036	0.99983	0.009
15	0.0021	0.99994	0.007	38	0.0028	0.99989	0.008	61	0.0015	0.99997	0.006
16	0.0031	0.99987	0.008	39	0.0019	0.99995	0.005	62	0.0019	0.99995	0.004
17	0.0026	0.99991	0.008	40	0.0041	0.99978	0.009	63	0.0026	0.99991	0.007
18	0.0011	0.99998	0.004	41	0.0024	0.99992	0.006	64	0.0019	0.99995	0.006
19	0.0022	0.99993	0.008	42	0.0038	0.99981	0.014	65	0.0020	0.99994	0.006
20	0.0030	0.99989	0.010	43	0.0026	0.99991	0.010	66	0.0016	0.99996	0.006
21	0.0023	0.99993	0.009	44	0.0029	0.99988	0.009	67	0.0021	0.99994	0.007
22	0.0019	0.99995	0.006	45	0.0027	0.99990	0.007	68	0.0032	0.99987	0.009
23	0.0017	0.99996	0.005	46	0.0028	0.99990	0.007	69	0.0017	0.99996	0.006

The first-order accumulated generating operation (AGO) sequence is

$$X^{(1)} = \left\{x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(n)\right\}, \tag{37}$$

where $x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i)$, $k = 1, 2, \ldots, n$.

The background value array $Z^{(1)}$ can be calculated as

$$Z^{(1)} = \left\{z^{(1)}(2), z^{(1)}(3), \ldots, z^{(1)}(n)\right\}, \tag{38}$$

where $z^{(1)}(k) = \alpha x^{(1)}(k) + (1-\alpha)x^{(1)}(k-1)$, $k = 2, 3, \ldots, n$, and the generating coefficient α is typically set as 0.5.

The generating coefficient is not always the same for the entire system [38]. Several studies [39,40] have shown that an optimal α should be identified to reduce the modeling error of grey prediction. Hsu [37] proposed an improved transformed GM (1,1) model (ITGM (1,1)) to find the optimal value of parameter α by using a genetic algorithm. The ITGM (1,1) model forecasting equation was given by Eq. (11) in [37]. However, the computing method of the background value $z^{(1)}(k)$ with single parameter α is not suitable for the predictive issues in DESs due to the resource constraints in DESs,. In this work, the background value $z^{(1)}(k)$ in Eq. (38) is assumed to be calculated with two parameters α and γ, where their sum is not always equal to 1; i.e.,

$$z^{(1)}(k) = \alpha x^{(1)}(k) + \gamma x^{(1)}(k-1), \tag{39}$$

where $\alpha, \gamma \in (-\infty, \infty)$ and $k = 2, 3, \ldots, n$.

Theorem 4. *The first-order differential equation of GM (1,1) is $x^{(0)}(k) + az^{(1)}(k) = b$, which can be transformed as*

$$x^{(0)}(k) = \left[\frac{b - (\alpha+\gamma)ax^{(0)}(1)}{1+\alpha a}\right]\left(\frac{1-\gamma a}{1+\alpha a}\right)^{k-2}, \tag{40}$$

where a and b are the developing coefficient and grey input, respectively, and $k = 2, 3, \ldots, n$.

Proof. From Eq. (37), we obtain

$$x^{(1)}(k) = x^{(1)}(k-1) + x^{(0)}(k), \tag{41}$$

where $k = 2, 3, \ldots, n$.

From (39) and (41), the first-order differential equation is transformed as

$$(1+\alpha a)x^{(0)}(k) + (\alpha+\gamma)ax^{(1)}(k-1) = b.$$

When $k = 2$, we obtain

$$x^{(0)}(2) = \frac{b - (\alpha+\gamma)ax^{(0)}(1)}{1+\alpha a}.$$

When $k = 3$, we generate

$$(1 + \alpha a)x^{(0)}(3) + (\alpha + \gamma)ax^{(1)}(2) = b.$$

Then, we can yield

$$\begin{aligned} x^{(0)}(3) &= \frac{b - (\alpha + \gamma)ax^{(0)}(1) - (\alpha + \gamma)ax^{(0)}(2)}{1 + \alpha a} \\ &= x^{(0)}(2)\frac{1 - \gamma a}{1 + \alpha a} \\ &= \left[\frac{b - (\alpha + \gamma)ax^{(0)}(1)}{1 + \alpha a}\right]\frac{1 - \gamma a}{1 + \alpha a}. \end{aligned}$$

We can generally obtain

$$x^{(0)}(k) = x^{(0)}(k - 1)\frac{1 - \gamma a}{1 + \alpha a}. \tag{42}$$

By iterative calculation of Eq. (42), Eq. (40) is yielded. □

6.2. Probability transformed GM (1,1) model (PTGM (1,1))

We propose a novel GM (1,1) model, i.e., probability transformed GM (1,1) model (PTGM (1,1)) with two parameters, to predict the MRT in the ANCS. In the PTGM (1,1), the probability values are computed by the *cdf* of MRT, and the background value $z^{(1)}(k)$ is calculated using Eq. (39). The predictive value of sequence $\hat{X}^{(0)}$ is defined by

$$\hat{x}^{(0)}(k) = \left[\frac{b - \omega(\alpha + \gamma)ax^{(0)}(1)}{1 + \alpha a}\right]\left(\frac{1 - \omega \min\{\alpha, \gamma\} a}{1 + \alpha a}\right)^{k-2}, \tag{43}$$

where $k = 2, 3, \ldots, n$.

In Eq. (43), we let

$$\omega = \begin{cases} 1 - W(x^{(0)}(k - 1)), & P\left\{W(x^{(0)}(j)) < 0.5\right\} > 0.5, \\ W(x^{(0)}(k - 1)), & \text{Otherwise}, \end{cases}$$

where $W(x^0(i))$, which denotes the probability of $x^{(0)}(i)$, is computed using Eq. (31); $P\left\{W(x^0(j)) < 0.5\right\}, 1 \le j \le k - 1$ represents the ratio of the number of MRT whose probability is less than 0.5.

We optimally determine the two parameters α and γ of PTGM (1,1) to improve the accuracy of predicting the response times of the messages. A fitness function is initially defined to evaluate the appropriateness value of each iteration in the model. We use the MAPE to build the fitness function as

$$MAPE(\alpha, \gamma) = \frac{1}{n - 1}\sum_{k=2}^{n}\left|\frac{x^{(0)}(k) - \hat{x}^{(0)}(k)}{x^0(k)}\right| \times 100\%, \tag{44}$$

where $x^{(0)}(k)$ denotes the actual value and $\hat{x}^{(0)}(k)$ represents the corresponding predicted value. We derive the optimal solutions of decision variables α and γ using the DFP-method to minimize MAPE(α, γ).

Martin and Witt [41] stated that forecasting performance can be compared not only among different forecasting techniques but also across varying units; MAPE and root mean square percentage error (RMSPE) are the most commonly used accuracy measures (see Table 1 in [37]). The RMSPE is given by Eq. (45). The lower the MAPE and RMSPE values, the more accurate the forecast.

$$RMSPE(\alpha, \gamma) = \sqrt{\frac{1}{n - 1}\sum_{k=2}^{n}\left[\frac{x^{(0)}(k) - \hat{x}^{(0)}(k)}{x^0(k)}\right]^2} \times 100\%. \tag{45}$$

The two-parameter optimization algorithm of PTGM (1,1) is depicted as Algorithm 3.

In Algorithm 3, we use the symbolic operation method in MATLAB to generate the optimal values of parameters α and γ. We yield the symbolic expression of background value $Z^{(1)}$ and the symbolic formula of $MAPE(V)$. The gradient $\nabla MAPE(V)$ of the symbolic formula $MAPE(V)$ is obtained using *jacobian()* in MATLAB. With the implementation of Algorithm 1, the optimum values of decision variable V are generated.

Algorithm 3 Two-parameter optimization algorithm of PTGM (1, 1)

1: Calculate the AGO sequence by Eq. (37);
2: Get the symbolic expression of background value $Z^{(1)}$ by Eq. (39) with respect to α and γ;
3: Compute parameters a and b using the least square method, i.e.,

$$\hat{U} = \begin{pmatrix} a \\ b \end{pmatrix} = \left(B^T B\right)^{-1} B^T Y,$$

where $Y = \begin{pmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{pmatrix}, B = \begin{pmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(n) & 1 \end{pmatrix}$;

4: Calculate predictive values of sequence $\hat{X}^{(0)}$ by Eq. (43);
5: Generate the symbolic formula of MAPE(V) using Eq. (44), where column vector $V = (\alpha, \gamma)^T$;
6: Derive the gradient ∇MAPE(V) of Eq. (44) for V;
7: Set initial value of vector V, and tolerance $\varepsilon = 10^{-5}$;
8: Call Algorithm 1 to obtain the optimum solution $\hat{V}$ of decision variable V;
9: Compute the minimum value of MAPE($\hat{V}$) and RMSPE($\hat{V}$) using Eq. (44) and (45), respectively;
10: Calculate $\hat{U}$ and predictive value of sequence $\hat{X}^{(0)}$;
11: **return** $\hat{V}$, MAPE($\hat{V}$), and RMSPE($\hat{V}$).

Table 6
Optimal parameter values and performance measures of different GMs (1,1)

ID	PTGM (1,1)				ITGM (1,1)			GM (1,1)	
	α	γ	MAPE	RMSPE	α	MAPE	RMSPE	MAPE	RMSPE
1	0.9110	0.0579	10.7793	11.6546	0.5557	11.2592	12.0446	10.9111	16.6843
2	0.6005	−0.2031	18.7455	21.9080	1.3279	19.2495	22.5744	20.5430	23.8058
3	0.4605	0.2319	14.7597	18.3841	0.5845	18.1305	21.8840	18.8721	25.9465
4	0.7146	−0.1339	10.6770	13.5263	0.5323	11.2842	14.4131	12.5637	15.5185
5	0.3588	0.3763	15.5991	24.4904	0.5058	16.4415	20.7637	16.2007	27.2693
6	0.2061	0.8660	40.0277	45.6045	1.5866	65.4939	83.9672	87.0142	127.279
10	0.5806	0.3986	3.7043	4.7236	0.4912	24.5622	29.2825	25.0412	28.2737
12	−0.2580	1.2511	25.8821	27.1803	0.5079	25.2914	30.0797	26.1496	29.7466
16	−2.9678	3.9124	27.3957	36.6965	0.5261	41.6109	47.5201	40.7243	47.7868
20	0.7768	−0.1137	9.2270	11.3267	0.5300	9.7522	11.7460	10.2999	12.3121
21	0.1671	0.6845	34.4147	46.1168	2.0491	42.8135	53.2240	59.4914	82.3315
22	0.5492	0.4508	38.7443	48.7817	0.9450	51.3058	63.9753	108.632	142.062
23	0.5362	0.4014	0.4337	0.5274	0.4959	1.3591	1.5318	1.5590	1.5920
24	0.0384	0.9382	0.3169	0.4530	0.4987	0.4010	0.4486	0.5234	0.5479
25	0.5179	0.4818	0.0186	0.0293	0.5001	0.0209	0.0231	0.2742	0.2866
26	−1.0914	3.2596	20.1295	24.1307	0.4798	23.6006	27.9809	23.7720	28.4443
27	0.0699	0.9066	12.1753	15.1631	0.7937	34.3377	38.4766	34.5011	42.1164
28	−0.8172	2.7321	18.0720	21.2076	0.4860	19.9415	24.1121	20.0489	24.3637
29	0.4922	−0.0591	18.8756	22.1001	0.7958	21.8774	27.7722	41.5503	52.2167
30	0.5489	0.0943	25.7931	45.6623	0.6217	27.0471	47.0546	31.3708	50.4620
31	−0.3446	1.8946	12.9282	15.2651	0.4947	13.3163	16.5628	13.3755	16.5916
32	0.3004	0.6815	20.9190	22.7608	0.0287	21.1864	23.3555	21.4853	24.9075
33	0.4942	0.1605	7.3898	8.3553	0.5410	9.1657	10.6462	9.5682	11.5760
34	0.5168	0.1702	6.8103	7.8237	0.5381	8.5217	9.6392	8.8311	10.2358
35	1.0923	−0.2910	14.3668	16.9134	0.6387	14.4420	16.7770	18.5735	20.2554
36	0.4770	0.2860	37.3835	44.1359	0.6605	78.0861	107.221	79.3627	101.474
45	0.4836	0.0186	29.9192	34.3351	0.6603	34.5990	40.6272	38.6430	46.0276

6.3. Measures of forecasting performance

When evaluating the forecasting performance, we do not consider the delay prediction problem of the messages that have only one or two message instances in a hyperperiod because the experiment results show that GM (1,1) and its improved models have high accuracy when forecasting the delay of such special sequences that contain only one or two elements. In Algorithm 3, let initial value of vector V be $(0.5, 0.5)^T$. For messages that contain more than three message instances in one hyperperiod, we implement Algorithm 3 to derive the optimal parameter values of the PTGM (1,1) model, which are shown in Table 6.

For forecasting performance evaluation and comparison, we derive the optimal parameter value α of ITGM (1,1) using the DFP-method and obtain the MAPE and RMSPE values of ITGM (1,1) and GM (1,1), respectively, as shown in Table 6. For

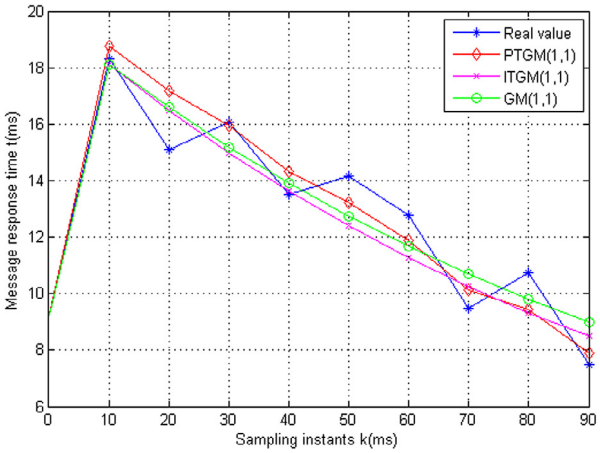


Fig. 14. Comparison of real and forecasting values for m_{34}.

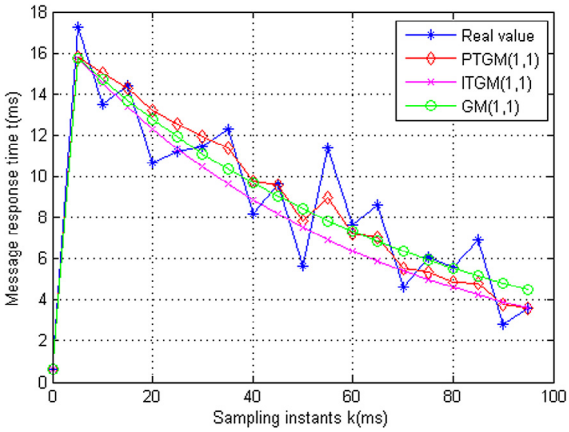


Fig. 15. Comparison of real and forecasting values for m_3.

all messages in Table 6, the MAPE values of PTGM (1,1) are less than those of ITGM (1,1) and GM (1,1). Low MAPE values indicate a small deviation between the predicted and real values. With consideration of the results in Table 6 and according to the criteria of MAPE in [37], the MAPE values of all messages for the PTGM (1,1) model are less than 50%, 20%, and 10%; these findings indicate reasonable, good, and high-accuracy forecasting capabilities, respectively. By contrast, in the ITGM (1,1) model, the MAPE values of several messages, such as m_6, m_{22}, and m_{36}, are greater than 50%, which indicates inaccurate forecasting results. The forecasting effect of the GM (1,1) model is more inaccurate. According to Table 6, the parameters α and γ can be negative, and the equation $\alpha + \gamma = 1$ does not necessarily hold in the PTGM (1,1) model.

According to the model forecast results shown in Table 6, the proposed PTGM (1,1) model obtains smaller MAPE and RMSPE values than do the ITGM (1,1) and GM (1,1) models. Therefore, the PTGM (1,1) model is an appropriate model for forecasting the MRT of the ANCS. We select three representative messages, m_{34}, m_3, and m_6, which have highly accurate, good, and reasonable forecasting values, respectively, using the PTGM (1,1) model. Their forecasting values are shown in Figs. 14, 15, and 16. The forecasting values of the PTGM (1,1) model are remarkably close to the real values of the MRTs.

The *cdfs* of the forecasting response times of the three messages above are shown in Figs. 17, 18, and 19, respectively. The more accurate the forecasting values, the closer the corresponding *cdf* to the real values of the MRTs. As shown in the figures, the quality of approximation does not necessarily depend on the priority of the message but rather on the forecasting model. A comparison with the predictive results in [24] shows that the *cdfs* of the forecasting values of m_3 and m_{34} yielded by the PTGM (1,1) model are more accurate (see Figs. 13 and 14 in [24]). In Table 6, the MAPE value of message m_6 is largest in the values of all reference messages. Therefore, the forecasting value of m_6 calculated by the PTGM (1,1) model is the most inaccurate in all reference messages. However, the *cdf* of its forecasting value is closer to the actual *cdf* than that in [24] (see Fig. 12 in [24]).

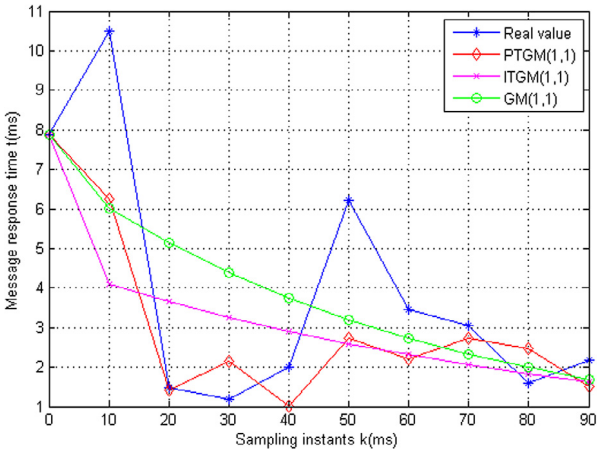


Fig. 16. Comparison of real and forecasting values for m_6.

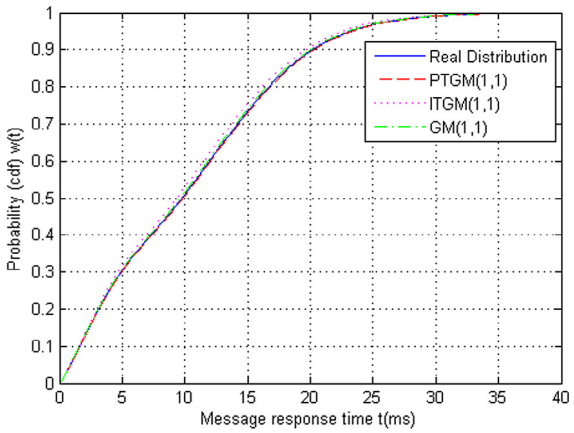


Fig. 17. Prediction of response time *cdf* for m_{34}.

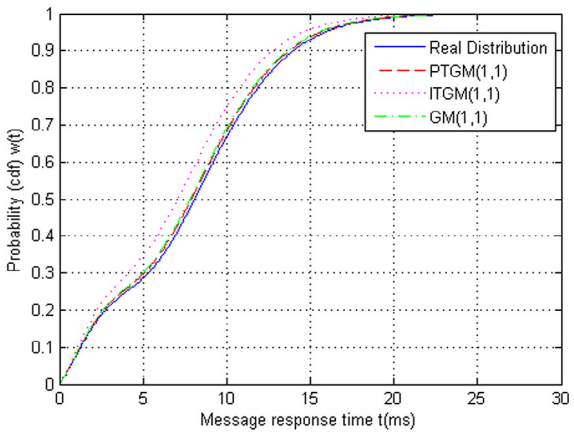


Fig. 18. Prediction of response time *cdf* for m_3.

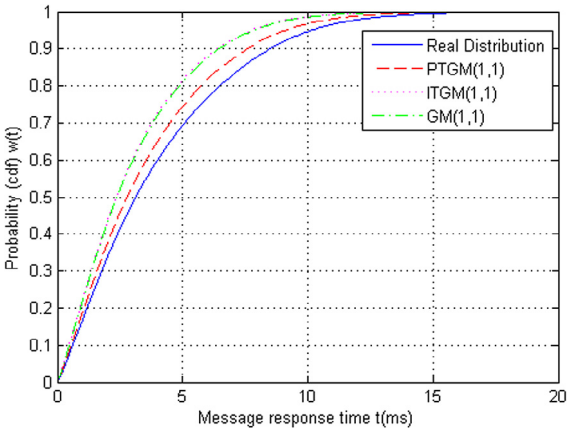


Fig. 19. Prediction of response time *cdf* for m_6.

7. Related works

MRT analysis is an important work related to in-vehicle networks. Davis et al. [42] found that the seminal analysis of [12] was flawed in certain cases. Accordingly, the authors refuted and revisited the original analysis and provided a new version that can be used to calculate correct WCRTs for CAN. However, this work did not involve the analysis of the uncertain MRT under optimal bus configuration. Thereafter, Davis et al. and Mubeen et al. extended the MRT analysis techniques with priority- and FIFO-queues (see Fig. 1 in [13]), offsets-based analysis (see Fig. 1 in [14]), and abortable and non-abortable transmit requests (see Fig. 1 in [15]). Mubeen et al. [13] extended the seminal analysis [12,42] to support the WCRT calculations for mixed messages that are simultaneously time- and event-triggered in CAN using priority- and FIFO-queued models. Davis et al. [43,44] extended the analysis for CAN where some nodes implemented priority queues, whereas others implemented FIFO queues and supported the analysis for CAN messages with arbitrary deadlines. Schmidt et al. [45] proposed an improved algorithm for speeding up the WCRT computation with priority- and FIFO-queues in CAN and addressed a practical case where an existing CAN message set is extended by new messages. Xie et al. [46] proposed an explorative WCRT computation method for gateway-integrated CAN messages. These works focused on the issues of priority- and FIFO-queues and provided RTA and appropriate priority assignment policies for CANs. WCRT computation methods were also developed. However, these works neither studied the problem of uncertainty MRT analysis from the perspective of CPS nor modeled the problem of priority or FIFO queues from the standpoint of queueing theory, thereby resulting in an incapability to quantify the dynamic behavior of messages. In the present work, for each control application in the ANCS, a corresponding subqueue is generated by the sensor's sampling data and the reference command. Each message goes directly into service upon arrival if the corresponding server is free; otherwise, the message is added to the queue. Therefore, we model each control application in the ANCS as a QBD process using an *M/PH/1* queue and focus on the uncertainty MRT analysis of the system while considering the optimal bus configuration under the optimal minslot and the multi-hierarchical FTDMA/FP bus protocol. We accurately obtain the WCRT of each message.

To avoid deadline violations caused by high transient loads, automotive embedded systems are often scheduled with offsets [47]. WCRT analysis for CAN messages with offsets was proposed in [48]. Mubeen et al. [49] extended the existing analysis for CAN [50] to support mixed messages that are scheduled with offsets due to limitations regarding message jitter and deadlines. Mubeen et al. [51] removed these limitations and extended the analysis for mixed messages by building it upon the analysis for CAN messages with offsets. The existing analysis for mixed messages with offsets does not support transmission abort requests in CAN controllers. Mubeen et al. presented their work in progress for the extension of RTA for mixed messages in CAN with abortable transmit buffers [52] and extended this work to support RTA with offsets for mixed messages in CAN [14]. The extended analysis is applicable to any high-layer CAN protocol that uses periodic, sporadic, and mixed transmission of messages, where the periodic and mixed message can be scheduled with offsets. Analysis of WCRT with offsets provides the architect with a set of values that can be used to base evaluation on what-if analyses and integrated processes. However, the worst-case evaluation is not sufficient and must be supplemented by probabilistic analysis. In our work, we focus on the establishment of probability models and analysis of uncertain MRT in the ANCS.

In the real-time system community, much work has been devoted toward the probabilistic analysis of MRT. An original approach to the analysis of task response times is real-time queueing theory [53]. The task set is modeled as a Markovian process with Poisson arrivals and exponential service times. Given a scheduling algorithm and the distribution of the deadlines, the distributions of the remaining time until the deadline of jobs are computed. However, the approach is merely an approximation because it assumes heavy traffic conditions and a remarkably large set of tasks with high utilization ($\rho \rightarrow 1$) [19]. The traffic in a DES is small, and the system utilization is low. In this study, Table 4 shows that the system

utilization of all messages except the 3rd and 34th is less than 0.5, where the system utilization of each message can be obtained using $\rho_i = \frac{\lambda_i}{\mu_i}$. In all messages, the system utilization of 88.4% messages is less than 0.3. Kim and Shin [54] used a continuous-time Markov chain to model the application as a queueing network, considering exponentially distributed task execution times scheduled by FIFO. However, these works neither explicitly address the uncertainty of the processor load nor are easily extendable to network scheduling or arbitration. In this work, we consider not only the service time of the sensors and controllers but also the message arbitration mechanism under the optimal configuration of the shared communication bus.

For messages scheduled on a CAN bus, Navet et al. [55] introduced the concept of worst-case deadline failure probability because of transmission errors. In [56], Nolte et al. extended the analysis of WCRT using uncertain message transmission times when considering the probability of a given number of stuff bits; they created a trade-off between reliability and timeliness, but the analysis was still performed in the worst-case scenario. Broster et al. [57] presented a probabilistic analysis of the effect of random network faults on CAN message latencies in the worst-case scenario. Manolache et al. [58] considered tasks with varying execution times with given generalized probability distributions and then presented an approach to obtain the expected deadline miss ratio (DMR) of the tasks. However, this work assumed that the task execution times followed the generalized probability distribution; moreover, the task execution deadlines were given and arbitrary, thereby leading to inaccurate analysis results. Tanasa et al. [18] computed the DMR of each message. This work relied on a mixed-integer linear program formulation to build a transition graph and compute the DMR using a GPU-based engine. However, these research works on message scheduling analysis of CAN buses are based on the worst-case scenario. The probabilistic evaluation of CAN MRTs was also not adequately addressed in these techniques, especially in the presence of uncertainty. In this research, we model the control application in the ANCS using a continuous-time *M/PH/1* queueing system and analyze the uncertain response delays of the messages in this control application. The closed-form expressions of the *pdf* and *cdf* of the MRTs for each control application are accurately calculated. We provide the theory for the probabilistic analysis of the MRTs in the end-to-end propagation of information among periodically activated messages.

In [59] and [60], Diaz et al. presented a stochastic analysis framework to compute the probability distribution of task response times for fixed-priority systems where tasks are periodic, preemptable, and activated with known offsets. The probability mass functions (*pmfs*) of the response times of a set of independent periodic tasks executed on a single processor by a preemptive priority-based scheduler are computed. The activation times of all the task instances are known, and the task execution times are defined by *pmfs*. However, these analyses assumed knowledge of the (relative) activation time of the task and pessimistically estimates the delay in blocking due to a lack of preemption and release of jitter. Zeng et al. [19] extended previous work on stochastic analysis of response times for software tasks to controller area network messages and compared them with sampling delays to compute the probability distributions of end-to-end latencies. The authors used one characteristic interference message for each remote node to model the interference caused by messages from the said node. This work changed the queueing model of remote characteristic interference messages from deterministic periodicity with random initial offsets to deterministic offsets, periodic activation times, and random queueing jitter. The introduction of characteristic interference messages and random jitter led to inaccurate system modeling and analysis. Zeng et al. [24] used statistical analysis methods to compute the probability distribution of the MRTs in CAN when only partial information is available about the functionality and architecture of a vehicle. The authors used the probability mix model of the degenerate distribution and the gamma distribution to fit the MRTs for the CAN bus and used regression analysis to obtain the expression of the *cdf* of the response times of the messages. This method of probability distribution fitting based on regression analysis led to inaccurate results. A good review of the MRT analysis in CAN was conducted by [61]. In the current work, we use a continuous-time *M/PH/1* queueing system to model the control application in the ANCS as a QBD process and utilize the calculation method of the waiting time distribution in the QBD process to generate the closed-form expressions of the *pdf* and the *cdf* of the response times of each message. Experiments reveal that our results are more accurate than those in [24] for early estimation.

8. Conclusion

In this study, we calculated the probability distribution of uncertain message response delay and predicted the delay in an ANCS. By analyzing the characteristics of the system, an *M/PH/1* queue model with random-sized batch arrivals was established to construct the cost function of the ANCS and obtain the *pdf* of the MRTs for early estimation. On the basis of the model and the DFP variable-scale method, an optimal *minislot* algorithm and a parameter estimation algorithm of *pdfs* were proposed for deriving the optimal values of the bus *minislot* and matrix parameter of PH distribution, respectively. We utilized the message set on one of the CAN buses of an experimental vehicle in [24] as our test data. Present experiment results showed that the fitting accuracy was higher than that in [24]. We presented a PTGM (1,1) model with two parameters to predict the response delay for each message. MAPE and RMSPE were used to evaluate forecasting accuracy. We designed a two-parameter optimization algorithm of PTGM (1,1) that is based on the DFP-method to minimize the MAPE and to obtain the optimal values of the two parameters for each message. Our numerical results showed that the forecasting results of PTGM (1,1) were more accurate than those of the existing ITGM (1,1) and GM (1,1) models. The prediction MRTs can be used to design a controller to compensate the system delay for closed-loop systems in future research. A comparison with the predictive results in [24] showed that the *cdfs* of the forecasting values obtained using the PTGM (1,1) model were closer to the actual distributions than those in [24]. We believe these results are important in the early architecture exploration phase of ACPSs. Using our approach, developers can explore different priorities of new messages and evaluate different probability distributions of MRTs for assignments with varying priorities.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments and suggestions on improving the quality of the paper. The work was partially supported by the National Key Research and Development Plan of China under Grants 2016YFB0200405 and 2012AA01A301-01, by the National Natural Science Foundation of China under Grants 61672217, 61370097, 61370095, and 11771060, by the SRF of Hunan Provincial Education Department under Grant 17A003.

References

[1] A. Annaswamy, D. Soudbakhsh, R. Schneider, D. Goswami, S. Chakraborty, Arbitrated network control systems: A co-design of control and platform for cyber-physical systems, in: Proc. Workshop Control Cyber Phys. Syst., 2013, pp. 339–356.
[2] R.N. Charette, This car runs on code, IEEE Spectr. 46 (46) (2009) 1–6.
[3] D. Goswami, R. Schneider, A. Masrur, M. Lukasiewycz, S. Chakraborty, H. Voit, A. Annaswamy, Challenges in automotive cyber-physical systems design, in: Proc. IEEE Int. Conf. Embedded Comput. Syst.: Architectures, Modeling and Simulation (IC-SAMOS), 2012, pp. 346–354.
[4] S. Chakraborty, M.A.A. Faruque, W. Chang, D. Goswami, M. Wolf, Q. Zhu, Automotive cyber-physical systems: A tutorial introduction, IEEE Design and Test 33 (4) (2016) 92–108.
[5] D. Soudbakhsh, L.T. Phan, O. Sokolsky, I. Lee, A. Annaswamy, Co-design of control and platform with dropped signals, in: Proc. 4th ACM/IEEE Int. Conf. Cyber Phys. Syst. (ICCPS), 2013, pp. 129–140.
[6] A. Annaswamy, S. Chakraborty, D. Soudbakhsh, D. Goswami, H. Voit, The arbitrated networked control systems approach to designing cyber-physical systems, in: Proc. 3rd IFAC Workshop Distrib. Estimat. and Control Netw. Syst.(NecSys), 2012, pp. 174–179.
[7] H. Voit, An Arbitrated Networked Control Systems Approach to Cyber-Physical Systems (Ph.D. thesis), Institute of Auto. Control Eng., Tech. Univ. Munich, Munich, 2013.
[8] Y. Ge, Q. Chen, M. Jiang, Y. Huang, Schmm-based modeling and prediction of random delays in networked control systems, J. Franklin Inst. B 351 (5) (2014) 2430–2453.
[9] M. Joseph, P. Pandya, Finding response times in a real-time system, Comput. J. 29 (5) (1986) 390–395.
[10] N.C. Audsley, A. Burns, M. Richardson, K. Tindell, A.J. Wellings, Applying new scheduling theory to static priority pre-emptive scheduling, Softw. Eng. J. 8 (5) (1993) 284–292.
[11] N.C. Audsley, A. Burns, R.I. Davis, K.W. Tindell, A.J. Wellings, Fixed priority preemptive scheduling: an historic perspective, Real-Time Syst. 8 (2–3) (1995) 173–198.
[12] K. Tindell, H. Hansson, A. Wellings, Analysing real-time communications: controller area network (can), in: Proc. 15th IEEE Real-Time Systems Symposium (RTSS), 1994, pp. 259–263.
[13] S. Mubeen, J. Maki-Turja, M. Sjodin, Extending worst-case response-time analysis for mixed messages in controller area network with priority and fifo queues, IEEE Access 2 (1) (2014) 365–380.
[14] S. Mubeen, J. Maki-Turja, M. Sjodin, Response time analysis with offsets for mixed messages in can supporting transmission abort requests, in: Proc. 19th IEEE Int. Conf. Emerging Technology and Factory Automation (ETFA), 2014, pp. 291–294.
[15] S. Mubeen, J. Maki-Turja, M. Sjodin, Integrating mixed transmission and practical limitations with the worst-case response-time analysis for controller area network, J. Syst. Softw. 99 (2015) 66–84.
[16] T. Pop, P. Pop, P. Eles, Z. Peng, A. Andrei, Timing analysis of the flexray communication protocol, Real-Time Syst. 39 (1) (2008) 205–235.
[17] L. Ouedraogo, R. Kumar, Computation of the precise worst-case response time of flexray dynamic messages, IEEE Trans. Autom. Sci. Eng. 11 (2) (2014) 537–548.
[18] B. Tanasa, U.D. Bordoloi, P. Eles, Z. Peng, Probabilistic timing analysis for the dynamic segment of flexray, in: Proc. 25th Euromicro Conf. Real-Time Syst. (ECRTS), 2013, pp. 135–144.
[19] H. Zeng, M.D. Natale, P. Giusto, A. Sangiovanni-Vincentelli, Stochastic analysis of can-based real-time automotive systems, IEEE Trans. Ind. Inf. 5 (4) (2009) 388–401.
[20] G. Xie, G. Zeng, R. Kurachi, H. Takada, R. Li, Gateway modeling and response time analysis on can clusters of automobiles, in: Proc. 17th IEEE Int. Conf. High Performance Computing and Communications (HPCC), 2015, pp. 1147–1153.
[21] G. Xie, G. Zeng, Z. Li, R. Li, K. Li, Adaptive dynamic scheduling on multi-functional mixed-criticality automotive cyber-physical systems, IEEE Trans. Veh. Technol. 66 (8) (2017) 6676–6692.
[22] D. Soudbakhsh, L.T.X. Phan, A.M. Annaswamy, O. Sokolsky, Co-design of arbitrated network control systems with overrun strategies, IEEE Trans. Control Netw. Syst. 5 (1) (2018) 128–141.
[23] Y. Xie, G. Zeng, Y. Chen, R. Kurachi, H. Takada, R. Li, Worst case response time analysis for messages in controller area network with gateway, IEICE Trans. Inf. Syst. 96-D (7) (2013) 1467–1477.
[24] H. Zeng, M.D. Natale, P. Giusto, A. Sangiovanni-Vincentelli, Using statistical methods to compute the probability distribution of message response time in controller area network, IEEE Trans. Ind. Inf. 6 (4) (2010) 678–691.
[25] M.F. Neuts, Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach, Johns Hopkins University Press, Baltimore, MD, USA, 1981.
[26] J.L. Deng, Introduction to grey system theory, J. Grey Syst. 1 (1) (1989) 1–24.
[27] W.C. Davidon, Variable metric method for minimization, SIAM J. Optim. 1 (1) (1991) 1–17.
[28] R. Fletcher, M.J.D. Powell, A rapidly convergent descent method for minimization, Comput. J. 6 (2) (1963) 163–168.
[29] C.D. Lewis, Industrial and Business Forecasting Methods: A Practical Guide to Exponential Smoothing and Curve Fitting, Butterworths Scientific, London, UK, 1982.
[30] H. Gong, R. Li, J. An, W. Chen, K. Li, Scheduling algorithms of flat semi-dormant multi-controllers for a cyber-physical system, IEEE Trans. Ind. Inf. 13 (4) (2017) 1665–1680.
[31] G. Cena, A. Valenzano, On the properties of the flexible time division multiple access technique, IEEE Trans. Ind. Inf. 2 (2) (2006) 86–94.
[32] J.A. White, J.W. Schmidt, G.K. Bennett, Analysis of queueing systems, Academic Press, Inc., Harcourt Brace Jovanovich Publishers, New York, 1975.
[33] H. Okamura, T. Dohi, K.S. Trivedi, A refined em algorithm for ph distributions, Perform. Eval. 68 (10) (2011) 938–954.
[34] H. Okamura, T. Dohi, Ph fitting algorithm and its application to reliability engineering, J. Oper. Res. Soc. Japan 59 (1) (2016) 72–109.
[35] A.O. Allen, Probability, Statistics, and Queueing Theory With Computer Science Applications (Second Edition), Academic Press, INC, San Diego, CA, USA, 1990.

[36] H. Zhao, H. Zhao, S. Guo, Using gm(1,1) optimized by mfo with rolling mechanism to forecast the electricity consumption of inner mongolia, Appl. Sci. 6 (1) (2016) 1–18.
[37] L.C. Hsu, Using improved grey forecasting models to forecast the output of opto-electronics industry, Expert Syst. Appl. 38 (11) (2011) 13879–13885.
[38] J.C. Wen, K.H. Huang, K.L. Wen, The study of alpha in gm(1,1) model, J. Chinese Inst. Engrs. 23 (5) (2000) 583–589.
[39] K.C. Hung, C.Y. Chien, K.J. Wu, F.Y. Hsu, Optimal alpha level setting in gm(1,1) model based on genetic algorithm, J. Grey Syst. 12 (1) (2009) 23–31.
[40] C.H. Wang, L.C. Hsu, Using genetic algorithms grey theory to forecast high technology industrial output, Appl. Math. Comput. 195 (1) (2008) 256–263.
[41] C.A. Martin, S.F. Witt, Forecasting tourism demand: A comparison of the accuracy of several quantitative methods, Int. J. Forecast. 5 (1) (1989) 7–19.
[42] R.I. Davis, A. Burns, R.J. Bril, J.J. Lukkien, Controller area network (can) schedulability analysis: Refuted, revisited and revised, Real-Time Syst. 35 (3) (2007) 239–272.
[43] R. Davis, N. Navet, Controller area network (can) schedulability analysis for messages with arbitrary deadlines in fifo and work-conserving queues, in: Proc. 9th IEEE Int. Workshop on Factory Comm. Syst. (WFCS), 2012, pp. 33–42.
[44] R. Davis, S. Kollmann, V. Pollex, F. Slomka, Schedulability analysis for controller area network (can) with fifo queues priority queues and gateways, Real-Time Syst. 49 (1) (2013) 73–116.
[45] K.W. Schmidt, B. Alkan, E.G. Schmidt, D.C. Karani, U. Karakaya, Controller area network with priority queues and fifo queues: improved schedulability analysis and message set extension, Int. J. Veh. Des. 71 (1–4) (2016) 335–357.
[46] G. Xie, G. Zeng, R. Kurachi, H. Takada, Z. Li, R. Li, K. Li, Wcrt analysis of can messages in gateway-integrated in-vehicle networks, IEEE Trans. Veh. Technol. 66 (11) (2017) 9623–9637.
[47] P.M. Yomsi, D. Bertrand, N. Navet, R.I. Davis, Controller area network (can): Response time analysis with offsets, in: Proc. 9th IEEE Int. Workshop on Factory Comm. Syst. (WFCS), 2012, pp. 43–52.
[48] A. Szakaly, Response Time Analysis with Offsets for Can (Master's thesis), Department of Computer Eng., Chalmers Univ. Tech., Goteborg, Sweden, 2003.
[49] S. Mubeen, J. Maki-Turja, M. Sjodin, Worst-case response-time analysis for mixed messages with offsets in controller area network, in: Proc. 17th IEEE Int. Conf. Emerging Technology and Factory Automation (ETFA), 2012, pp. 1–10.
[50] Y. Chen, R. Kurachi, H. Takada, G. Zeng, Schedulability comparison for can message with offset: Priority queue versus fifo queue, in: Proc. 19th Int. Conf. on Real-Time and Network Syst. (RTNS), 2011, pp. 181–192.
[51] S. Mubeen, J. Maki-Turja, M. Sjodin, Extending offset-based response-time analysis for mixed messages in controller area network, in: Proc. 18th IEEE Int. Conf. Emerging Technology and Factory Automation (ETFA), 2013, pp. 1–10.
[52] S. Mubeen, J. Maki-Turja, M. Sjodin, Response time analysis for mixed messages in can supporting transmission abort requests, in: Proc. 7th IEEE Int. Symp. on Industrial Embedded Syst. (SIES), 2012, pp. 291–294.
[53] J.P. Lehoczky, Real-Time queueing theory, in: Proc. 17th IEEE Real-Time Systems Symposium (RTSS), 1996, pp. 186–195.
[54] J. Kim, K.G. Shin, Execution time analysis of communicating tasks in distributed systems, IEEE Trans. Comput. 45 (5) (1996) 572–579.
[55] N. Navet, Y.Q. Song, F. Simonot, Worst-case deadline failure probability in real-time applications distributed over can (controller area network), J. Syst. Archit. 46 (7) (2000) 607–617.
[56] T. Nolte, H. Hansson, C. Norstrom, Probabilistic worst-case response-time analysis for the controller area network, in: Proc. 9th IEEE Real-Time and Embedded Tech. Appl. Symp. (RTAS), 2003, pp. 200–207.
[57] I. Broster, A. Burns, G. Rodrguez-Navas, Probabilistic analysis of can with faults, in: Proc. 23rd IEEE Real-Time Systems Symposium (RTSS), 2002, pp. 269–278.
[58] S. Manolache, P. Eles, Z. Peng, Schedulability analysis of applications with stochastic task execution times, ACM Trans. Embed. Comput. Syst. 3 (4) (2004) 706–735.
[59] J.L. Diaz, D.F. Garcia, K. Kim, C.-G. Lee, L.L. Bello, J.M. Lopez, S.L. Min, O. Mirabella, Stochastic analysis of periodic real-time systems, in: Proc. 23rd IEEE Real-Time Systems Symposium (RTSS), 2002, pp. 289–300.
[60] J.M. Lopez, J.L. Diaz, J. Entrialgo, D. Garcia, Stochastic analysis of real-time systems under preemptive priority-driven scheduling, Real-Time Syst. 40 (2) (2008) 180–207.
[61] G.I. Mary, Z.C. Alex, L. Jenkins, Response time analysis of messages in controller area network: A Review, J. Comput. Netw. Comm. 2013 (2013) 1–11.

Hongfang Gong received the B.S. degree in mathematics from Changsha University of Science and Technology, China, the M.E. degree in computer application from Hunan University, China, in 1991 and 2004, respectively. He is currently pursuing 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, China. His research interests include cyber–physical systems (CPS), embedded computing system, and distributed control systems.

Renfa Li is a full professor of College of Computer Science and Electronic Engineering, and the Dean of College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He is the Director of Key Laboratory for Embedded and Network Computing of Hunan Province, Changsha, China. He is also an expert committee member of National Supercomputing Center in Changsha, China. His research interests include computer architecture, embedded computing system, cyber–physical systems (CPS), Internet of things. He is a senior member of IEEE, senior member of ACM, and a member of the Council of China Computer Federation.

Yang Bai received her B.S. and M.S. degrees from Hunan University in 2013 and 2016, respectively. She is currently pursuing the Ph.D. degree with the Key Laboratory for Embedded and Network Computing of Hunan Province, Hunan University, Changsha, China. Her research interests include service computing, embedded systems, cyber–physical systems.

Jiyao An received the M.Sc. degree in Mathematics from Xiangtan University, China, the Ph.D. degree in Mechanical Engineering from Hunan University, China, in 1998, and 2012, respectively. He was a Visiting Scholar with the Department of Applied Mathematics, University of Waterloo, Ontario, Canada, from 2013 to 2014. Since 2000, he joined the College of Computer Science and Electronic Engineering in Hunan University, Changsha, China, where he is currently a full Professor. His research interests include cyber–physical systems (CPS), 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. He is a member of the IEEE and ACM, and a senior member of CCF. He is an active reviewer of international journals.

Keqin Li 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 over 570 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or 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, IEEE Transactions on Sustainable Computing. He is an IEEE Fellow.