

Research Article

The Fuzzy Feedback Scheduling of Real-Time Middleware in Cyber-Physical Systems for Robot Control

Feng Tang, Ping Zhang, and Fang Li

South China University of Technology, Guangzhou, China

Correspondence should be addressed to Feng Tang; fengtang@scut.edu.cn

Received 4 March 2016; Revised 20 July 2016; Accepted 21 July 2016

Academic Editor: Antonio Fernández-Caballero

Copyright © 2016 Feng Tang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cyber-physical systems for robot control integrate the computing units and physical devices, which are real-time systems with periodic events. This work focuses on CPS task scheduling in order to solve the problem of slow response and packet loss caused by the interaction between each service. The two-level fuzzy feedback scheduling scheme is designed to adjust the task priority and period according to the combined effects of the response time and packet loss. Empirical results verify the rationality of the cyber-physical system architecture for robot control and illustrate the feasibility of the fuzzy feedback scheduling method.

1. Introduction

Cyber-physical systems (CPSs) are integrations of the physical world, computation, communication, and control designs [1]. Many cyber-physical systems are equipped with distributed computing units that communicate with each other through the network [2]. As [3, 4] propose, application architecture, control technique, and resource allocation are important research topics in CPSs. One representative CPSs application architecture is stated in [5]. Another application of CPSs for neutrally controlled artificial legs is put forward in [6]. Reference [7] outlines the architecture of passive control CPSs. All these systems embody the features of CPS like distributed computing.

Distributed robot control systems consist of many computing nodes, actuators, sensing devices, and network topologies [8–10]. These are typical applications of distributed CPSs that must handle many aperiodic and periodic events. The computing nodes or devices have different interfaces. Applying middleware can realize network interconnection, data integration, and integrated application, which accounts for the increasing significance of middleware technologies with respect to distributed robot control CPSs. Real-time CORBA [11] is a vital real-time middleware frequently used in distributed systems. A reconfigurable real-time middleware is

proposed in [12], which provides a flexible software platform for CPSs with timing constraints.

The scheduling models of traditional real-time CORBA are completely independent of each node [13]. That is to say, the scheduling in each node is independent. The scheduling method does not take nodal performance into account; therefore, it fails to guarantee global optimum. For that, the fuzzy feedback scheduling, a global scheduling framework based on end-to-end real-time CORBA scheduling model, is proposed in this paper. According to the nodal performance index, this method adjusts the task priority and period to improve the overall performance of CPSs for robot control and enhance the performance of the end-to-end real-time system.

2. Systems Architecture of CPSs for Robot Control Based on Real-Time Middleware

As the core mechanism of CORBA, ORB (Object Request Broker) realizes transparent interaction between objects residing at various computing nodes in heterogeneous environments. Using naming service of CORBA, the clients and the CRRBA objects are connected according to the objects' names. This method facilitates programming and establishes seamless connection between objective systems.

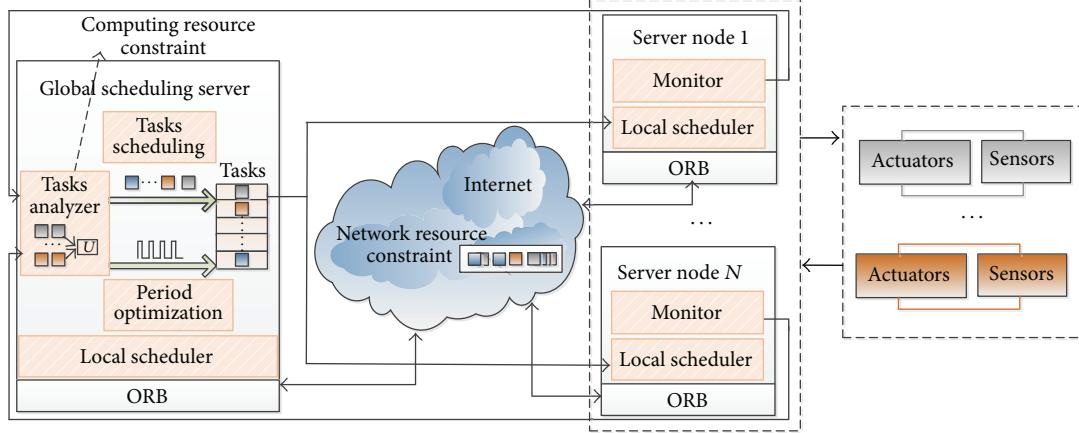


FIGURE 1: Systems architecture of CPSs for robot control.

Figure 1 shows the architecture of CPSs for robot control, which mainly consists of actuators, sensors, server nodes, and global scheduling server. These components communicate with each other via ORB. The tasks that travel along the servers compete for network resources of ORB soft bus. Once the requests are received, global scheduling server schedules the tasks to control multiple actuators, which can be regarded as a task management. Be they aperiodic or periodic events, the control tasks are assigned to server nodes by global scheduling server, resulting in competition for computing resources of global scheduling server. Moreover, implemented in different server nodes, the tasks in one server node compete for computing resources within their own server node while the monitor in charge of this server node monitors performance of this node. Then feedback information is sent to global scheduling server. The global scheduling server decides on the task priority and period according to the feedback information and assigns the tasks to server nodes in the light of priority. Server nodes map task priority to local operating system. This work focuses on resource scheduling and two-level scheduling method is illustrated later in the article.

3. Fuzzy Feedback Scheduling Method

The CPSs for robot control are an evolution of the real-time networked control systems [14, 15]. The tasks scheduling of CPSs can be based on the involved research results. Scheduling strategy can be divided into static scheduling and dynamic scheduling according to the implementation method. Liu and Layland proposed RM (Rate Monotonic) scheduling algorithm and EDF (Earliest Deadline First) real-time scheduling algorithm [16]. They are the most representative of the research results in scheduling algorithm. At present, they have become the research foundation of scheduling algorithm. RM algorithm is a static priority assignment algorithm which assigns priorities to tasks according to their request rates; the more frequent the task is, the higher in priority the task is. Reference [16] proves that the RM algorithm is the best one in the static scheduling algorithm. EDF scheduling algorithm is a dynamic priority

scheduling algorithm. it assigns priorities to tasks according to their absolute deadline. A task will be assigned the highest priority if the deadline of its current request is the nearest and will be assigned the lowest priority if the deadline of its current request is the furthest. In the EDF scheduling algorithm, tasks with the highest priority level are always to be executed first, while the lower-priority tasks are to be preempted.

Because the robot control system environment is complex, RM scheduling algorithm has poor environment suitability [16]. Reference [17] makes it clear that dynamic scheduling algorithm outperforms static scheduling algorithm in changeable environments. Therefore, this work centers on EDF real-time scheduling algorithm. Reference [18] presents an improved dynamic EDF scheduling method for the network control systems, where task priority is changed according to the control error. References [19–21] demonstrate similar scheduling methods to set the task priority and reduce packet loss.

Studies have shown that feedback scheduling can better utilize resources and improve control performance in the distributed systems [22]. So both EDF dynamic scheduling and feedback schedule are taken into consideration. Reference [23] illustrates a fuzzy feedback scheduler that makes better use of resources according to the system performance while [24] shows an online feedback scheduling algorithm for the sake of error reduction by optimizing the object function for a robot manipulator.

3.1. The Fuzzy Feedback Scheduling Framework. Figure 2 presents the fuzzy feedback scheduling framework. Two-level fuzzy feedback scheduling method is designed in this paper where the object function of the method includes the response time and transmission error.

The effect of the first level scheduling is setting priority. An EDF dynamic scheduling algorithm is designed at this level. First of all, requirements of resource utilization U_i for each close-loop by fuzzy feedback controller are estimated. Then, the priority configurator computes the task priority in consideration of deadline, importance, and requirements of resource utilization. Lastly, tasks are transferred in the light of priority order and mapped to the operating system.

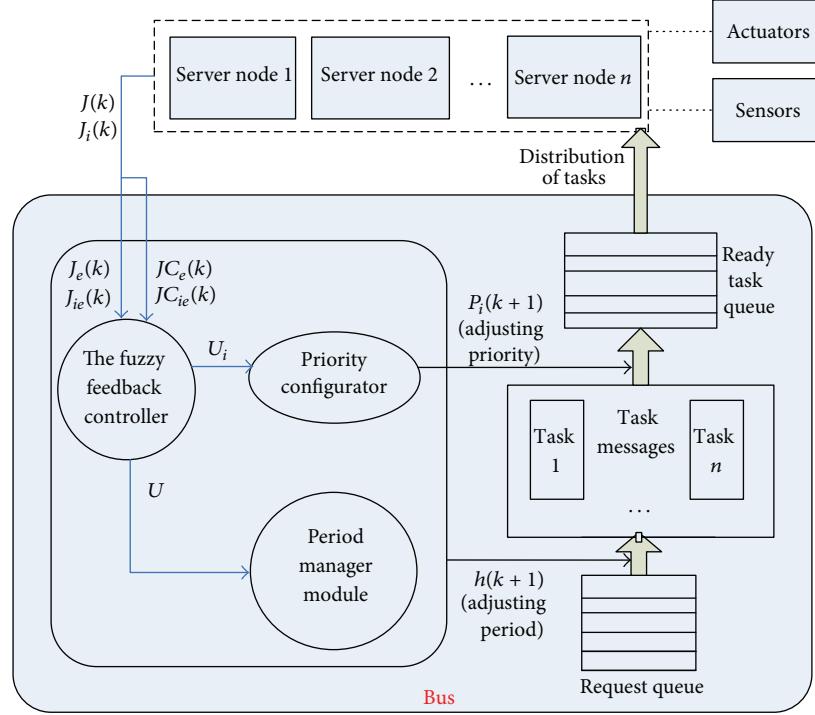


FIGURE 2: The fuzzy feedback scheduling framework.

The effect of the second level scheduling is adjusting period. Firstly, the total resource utilization U is computed by fuzzy feedback controller, and then period manager module adjusts the task period according to the total resource utilization. This method can address network congestion and achieve better resource utilization.

The feedback performance criteria function of the i th control loop is shown as follows: $J_i(k) = \alpha E_i(k) + \beta \tau_i(k)$, $E_i(k) = (X_i(k) - X_{is}(k))/X_i(k)$. $X_i(k)$ representing all packets of the i th control loop at the k th cycle is computed by the monitor of the global scheduling server. $X_{is}(k)$ referring to the packets that have been transmitted successfully is calculated by the monitor of the server nodes and sent to the global scheduling server. $\tau_i(k)$ represents the response time ratio and follows the equation $\tau_i(k) = H_i(k)/H_{imax}$. $H_i(k)$ is the execution time of the i th task. Global scheduling server computes the execution time via the starting time and terminal time that are reported from the actuators. H_{imax} is the maximum execution time, while α and β are the weight of the packet loss ratio and response time ratio following the equation of $\alpha + \beta = 1$.

The deviation of performance indicator of the i th control loop is described as follows: $J_{ie}(k) = J_{ir} - J_i(k) = \alpha(e_{ir} - E_i(k)) + \beta(l_{ir} - \tau_i(k))$. J_{ir} represents the desired performance criteria and follows the equation $J_{ir} = \alpha e_{ir} + \beta l_{ir}$. The differential of performance criteria follows the formula $JC_{ie}(k) = J_{ie}(k) - J_{ie}(k-1)$, which echoes the direction of change for performance criteria.

Similarly, the overall feedback performance criteria function of all control loops are defined as follows: $J(k) = \alpha E(k) +$

$\beta \tau(k)$, $E(k) = (X(k) - X_S(k))/X(k)$. $X(k)$ represents all packets of the whole control loops, while $X_S(k)$ represents the packets that have been transmitted successfully of the whole control loops. In $\tau(k) = (\sum_{i=1}^n H_i(k))/(\sum_{i=1}^n H_{imax})$, $\sum_{i=1}^n H_i(k)$ refers to the total response time of all control loops and $\sum_{i=1}^n H_{imax}$ the total maximum response time of all control loops. Deviation of overall performance indicator is described as $J_e(k) = J_r - J(k) = \alpha(e_r - E(k)) + \beta(l_r - \tau(k))$. Differential of overall performance criteria follows the equation $JC_e(k) = J_e(k) - J_e(k-1)$.

In this paper, the fuzzy feedback scheduling framework adjusts the task priority and period according to the feedback performance. The greater the priority is, the sooner the task should be handled. The worse the performance of the control loops is, the more resources they receive. When the total resource utilization U is great, it implies network congestion and the period should be shortened, and vice versa.

3.2. Fuzzy Feedback Control. The deviation of performance indicators $J_{ie}(k)$, $J_e(k)$ and the differential of performance criteria $JC_{ie}(k)$, $JC_e(k)$ are the inputs into the fuzzy feedback controller. Total resource utilization U and the i th resource utilization U_i are accessible through fuzzy feedback control. The relations between inputs and outputs are illustrated in the following equations: $U = f(J_e(k), JC_e(k))$, $U_i = f(J_{ie}(k), JC_{ie}(k))$.

There are four steps in fuzzy feedback control: (1) fuzzification; (2) establishing fuzzy control rules; (3) fuzzy composition; and (4) fuzzy judgment. The i th resource utilization U_i is taken as an example to illustrate these processes.

TABLE 1: Subordinate degree value for $J_{ie}(k)$.

Variable	$J_{ie}(k)$						
	-0.6	-0.4	-0.2	0	0.2	0.4	0.6
NB	1.0	0.5	0	0	0	0	0
NM	0.5	1	0.5	0	0	0	0
NS	0	0.5	1	0.5	0	0	0
Z	0	0	0.5	1	0.5	0	0
PS	0	0	0	0.5	1	0.5	0
PM	0	0	0	0	0.5	1	0.5
PB	0	0	0	0	0	0.5	1

TABLE 2: Subordinate degree value for $JC_{ie}(k)$.

Variable	$JC_{ie}(k)$						
	-0.3	-0.2	-0.1	0	0.1	0.2	0.3
NB	1.0	0.5	0	0	0	0	0
NM	0.5	1	0.5	0	0	0	0
NS	0	0.5	1	0.5	0	0	0
Z	0	0	0.5	1	0.5	0	0
PS	0	0	0	0.5	1	0.5	0
PM	0	0	0	0	0.5	1	0.5
PB	0	0	0	0	0	0.5	1

TABLE 3: Subordinate degree value for U_i .

Variable	U_i		
	0.6	0.4	0.2
PB	1.0	0.5	0
PM	0.5	1	0.5
PS	0	0.5	1

3.2.1. Fuzzification. The fuzzy sets of $J_{ie}(k)$ and $JC_{ie}(k)$ are expressed as {NB, NM, NS, Z, PS, PM, PB} which, respectively, denote negative big, negative medium, negative small, zero and positive small, positive medium, positive big. The fuzzy sets of $J_{ie}(k)$ and $JC_{ie}(k)$ are quantified as $\{-0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6\}$ and $\{-0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3\}$, respectively, whose subordinate degree values are correspondingly presented in Tables 1 and 2. The fuzzy set of the output variable U_i is expressed as {PS, PM, PB}, which, respectively, denotes small, medium, and big values. The fuzzy set is quantified as $\{0.2, 0.4, 0.6\}$, whose subordinate degree values are shown in Table 3.

3.2.2. Establishing Fuzzy Control Rules. The basic idea of fuzzy control rules is shown in Table 4. When the performance criteria and its differential value both are positives or negatives and their absolute values are relatively big, the system's performance is diverging from the calibrated value with a tendency of continuous deviation. Under such circumstance, the system performs poorly. So a bigger value should be assigned to the requirement of resource utilization U_i . When the performance criteria and its differential values are

TABLE 4: Fuzzy control rules.

U_i	$JC_{ie}(k)$						
	NB	NM	NS	Z	PS	PM	PB
$J_{ie}(k)$	NB	PS	PS	PS	PS	PM	PM
	NM	PS	PS	PS	PM	PM	PM
	NS	PS	PS	PM	PM	PM	PB
	Z	PS	PS	PM	PM	PB	PB
	PS	PS	PM	PM	PB	PB	PB
	PM	PM	PM	PB	PB	PB	PB
	PB	PM	PM	PB	PB	PB	PB

smaller, the system's performance is close to the calibrated value, which means the system performs well. In this case, a smaller value should be assigned to U_i . When the performance criteria and its differential are valued one positive and another negative, the system's performance is approaching to the calibrated value with a tendency of improvement. A medium value should be assigned to U_i . Fuzzy control rules can be expressed in fuzzy conditional statements, for example,

$$\begin{aligned} \text{if } J_{ie}(k) &= \text{NB or NM or NS,} \\ JC_{ie}(k) &= \text{NB or NM or NS} \\ \text{then } U_i &= \text{PS.} \end{aligned} \quad (1)$$

3.2.3. Fuzzy Composition. Compositional rule of fuzzy inference expresses a fuzzy relation by means of the Cartesian product. For instance, “if A and B then C ” is expressed as $R = (A \cap B) \rightarrow C$. According to the rule of Mamdani inference, the following expressions are concluded, where A^L refers to the transpose of A [25]:

$$R = (A \cap B) \times C = [A^L \circ B]^L \circ C. \quad (2)$$

There are 49 fuzzy control rules in Table 4, so the fuzzy relation is expressed as (3) shows [26]:

$$R = R_1 \cup R_2 \cup \dots \cup R_{49}. \quad (3)$$

R_1, R_2, \dots, R_{49} can be calculated from the following formula:

$$\begin{aligned} R_1 &= D_1^L \times (\text{PB})_{U_i}, \\ D_1 &= (\text{PB})_{J_{ie}(k)} \times (\text{PB})_{JC_{ie}(k)}, \\ &\vdots \\ R_{49} &= D_{49}^L \times (\text{NB})_{U_i}, \\ D_{49} &= (\text{NB})_{J_{ie}(k)} \times (\text{NB})_{JC_{ie}(k)}. \end{aligned} \quad (4)$$

D_1 can be calculated from the following equation:

$$\begin{aligned}
D_1 &= (\text{PB})_{J_{ie}(k)} \times (\text{PB})_{J_{C_{ie}}(k)} = (\text{PB})_{J_{ie}(k)}^L \circ (\text{PB})_{J_{C_{ie}}(k)} \\
&= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 1 \end{bmatrix} \circ [0 \ 0 \ 0 \ 0 \ 0 \ 0.5 \ 1] \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 1 \end{bmatrix}.
\end{aligned} \tag{5}$$

R_1 can be figured out from (6) by putting (5) into (4):

$$R_1 = D_1^L \times (\text{PB})_{U_i} = \left[(\text{PB})_{J_{ic}(k)}^L \circ (\text{PB})_{J_{C_{ic}(k)}} \right]^L \circ (\text{PB})_{U_i}$$

$$= \begin{bmatrix} 40 \times 1 & 0 \\ 0.5 & \\ 0.5 & \\ 0 & \\ 0 & \\ 0 & \\ 0 & \\ 0 & \\ 0.5 & \\ 1 & \end{bmatrix} \circ [1 \ 0.5 \ 0] = \begin{bmatrix} 40 \times 3 & 0 \\ 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 0 \\ 5 \times 3 & 0 \\ 0.5 & 0.5 & 0 \\ 1 & 0.5 & 0 \end{bmatrix}. \quad (6)$$

It is found from (6) that R_1 is a 49×3 matrix. R can be calculated out of (7) on the base of (3) and (4):

$$R = R_1 \cup R_2 \cup \cdots R_{49} = \begin{bmatrix} 0 & 0.5 & 1 \\ 0 & 0.5 & 1 \\ \vdots & & \\ 1 & 0.5 & 0 \\ 1 & 0.5 & 0 \end{bmatrix}. \quad (7)$$

Then the requirement of resource utilization U_i can be computed through fuzzy synthetic operation:

$$U_i = (J_{i\rho}(k) \times JC_{i\rho}(k)) \circ R. \quad (8)$$

For example, if,

$$J_{ie}(k) = (\text{NB})_{J_{ie}(k)} = [1.0 \ 0.5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1], \quad (9)$$

$$JC_{ie}(k) = (\text{NB})_{JC_{ie}(k)} = [1.0 \ 0.5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1],$$

then

$$U_i = (J_{ie}(k) \times JC_{ie}(k)) \circ R$$

$$= \begin{bmatrix} 1.0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & \vdots & 35 \times 0 \end{bmatrix}$$

$$\circ \begin{bmatrix} 0 & 0.5 & 1 \\ 0 & 0.5 & 1 \\ & \vdots & \\ 1 & 0.5 & 0 \\ 1 & 0.5 & 0 \end{bmatrix} = [0 \ 0.5 \ 1]. \quad (10)$$

3.2.4. Fuzzy Judgment. There are mainly two judgment methods: weighted average method and max membership function. The weighted average method is expressed as follows:

$$u = \frac{\sum_{i=1}^n u_i \mu(u_i)}{\sum_{i=1}^n \mu(u_i)}. \quad (11)$$

In this paper, weighted average method is adopted to determine the requirements of resource utilization U_i , while the max membership function is used to study the total resource utilization U . For example, if $U_i = [0 \ 0.5 \ 1]$, then U_i is calculated by the weighted average method as follows:

$$U_i = \frac{\sum_{i=1}^3 u_i \mu(u_i)}{\sum_{i=1}^3 \mu(u_i)} = \frac{0 \times 0.6 + 0.5 \times 0.4 + 1 \times 0.2}{0 + 0.5 + 1} \approx 0.267. \quad (12)$$

If $U = [0 \ 0.5 \ 1]$, then U is figured out by the max membership function as follows:

$$U = \frac{0}{0.6} + \frac{0.5}{0.4} + \frac{1}{0.2}. \quad (13)$$

Since $\mu(0.2) = 1$ is the largest of all subordinates, $U = 0.2$ is selected.

3.3. Priority Assignment Algorithm. Once the requests are received, the global scheduling server will schedule the tasks. Priority is given to each task according to the following scheduling algorithm: the task message (including the priority attributes such as the task's arrival time and deadline) is arranged to the ready task queue according to the priority and then it is distributed to trigger execution of each task in server node. The priority attributes of each task and its subtasks should be consistent within task cycle h_i .

The traditional EDF algorithm cannot promptly adjust the task priority in an uncertain operation environment. In the meanwhile, the feedback scheduling fails to dispatch in time as a result of an introduced feedback mechanism

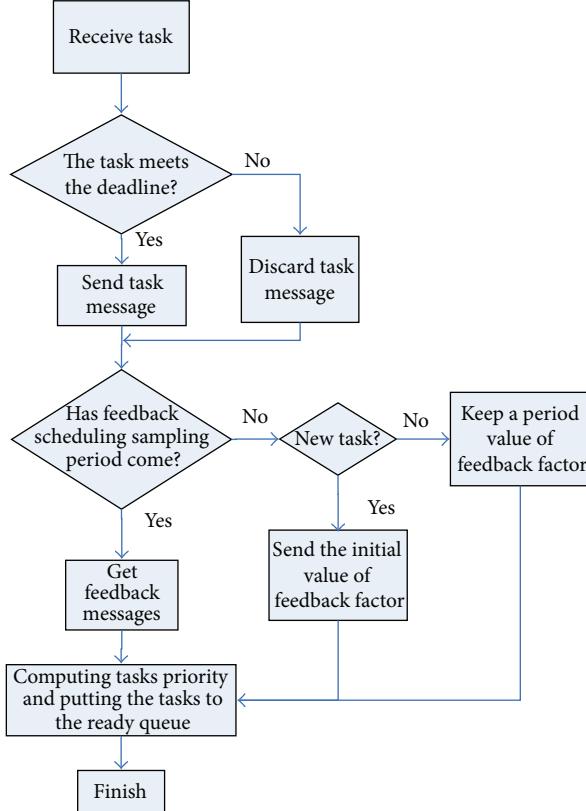


FIGURE 3: Task scheduling process.

(scheduling period should not be too short so as to avoid frequent switching task priority). “The fuzzy feedback improved EDF scheduling policy” proposed in this work is aimed at assigning the priority of each task and sorting the task messages in the ready task queue in progressive priority order. The basic idea is described as follows (Figure 3):

- (1) When a task message arrives, its schedulability is analyzed at first to judge whether it applies to the deadline or not.
- (2) When the scheduling sampling period comes, the fuzzy scheduling method as Section 3.2 shows is adopted to calculate the requirement of resource utilization U_i according to the feedback information $(J_{ie}(k), JC_{ie}(k))$. Then U_i is used to compute the priority as the next step proposes.
- (3) Task T_i , typically, has a residual deadline de_i , resource utilization U_i , and the importance of the task im_i . Its particular priority $P(T_i, de_i, U_i, im_i)$ ($i = 1, \dots, n$) is represented as P_i .

Resource utilization U_i is introduced as one of the metrics to determine the priority of task T_i together with the task's importance im_i and residual deadline de_i . The task priority is expressed in the following equation:

$$P_i = \zeta U_i \times \frac{im_i}{de_i} \quad (14)$$

Now adjusting the list!!

Size = 6

1: service ID = 16202 deadline = 129964974490228663
 2: service ID = 30186 deadline = 129964974493067141
 3: service ID = 7363 deadline = 129964974488790034
 4: service ID = 8679 deadline = 129964974491642807
 5: service ID = 12856 deadline = 129964974491628663
 6: service ID = 9832 deadline = 129964974494505122

Scheduling !!!!!!!

Service 7363 : priority = 0.018733
 Service 16202 : priority = 0.017710
 Service 12856 : priority = 0.016687
 Service 8679 : priority = 0.017632
 Service 30186 : priority = 0.016609
 Service 9832 : priority = 0.015589

Size = 6|

1: service ID = 7363 deadline = 129964974491628663
 2: service ID = 16202 deadline = 129964974493067141
 3: service ID = 12856 deadline = 129964974488790034
 4: service ID = 8679 deadline = 129964974494505122
 5: service ID = 30186 deadline = 129964974490228663
 6: service ID = 9832 deadline = 129964974491642807

FIGURE 4: A scheduling example.

If there comes a periodic task, the initial value of the resource utilization U_i is set at 0.3. As for an aperiodic task, U_i is set at 0.6. The adjustment coefficient ζ needs to be determined by an experimental method. It is possible that multiple tasks share the same priority. In this case, the tasks can be sorted according to the feedback performance or deadline. All in all, the shorter the deadline lasts, the greater resource utilization and task importance it embodies, and therefore the greater the priority is, and vice versa. This method excels itself in the full consideration of multifaceted attributes of the task and the presence of multiple parameters in the priorities. This method bridges the gap of traditional EDF algorithm which only takes deadline into account.

3.4. Period Adjustment Strategy. Too frequent periodic adjustment makes the system complicated and unstable. Therefore, three kinds of period ($P = 4, 8, 16$) are developed. To judge the total resource utilization U , the max membership function is employed to produce the fuzzy feedback controller outputs of $U = 0.2, 0.4, 0.6$. The period manager module computes the period P according to the mapping relations between resource utilization and task period which is presented in Table 5 and then writes the period to the task messages and transfers the message to related services. Finally, the services adjust the period according to the received message.

4. Algorithm Simulation

4.1. Priority Assignment Examples. A scheduling example is shown in Figure 5; the arrangement of tasks on ready task queue before scheduling stands at the upper part of Figure 4. To be specific the tasks ID are 16202, 30186, 7363, 8679, 12856, and 9832.

The lower part of this figure is the arrangement of tasks after scheduling. It shows that task 1 and task 3 embrace

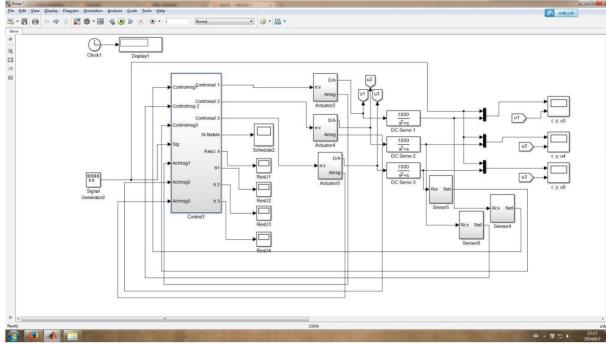


FIGURE 5: Simulation model for the scheduling system.

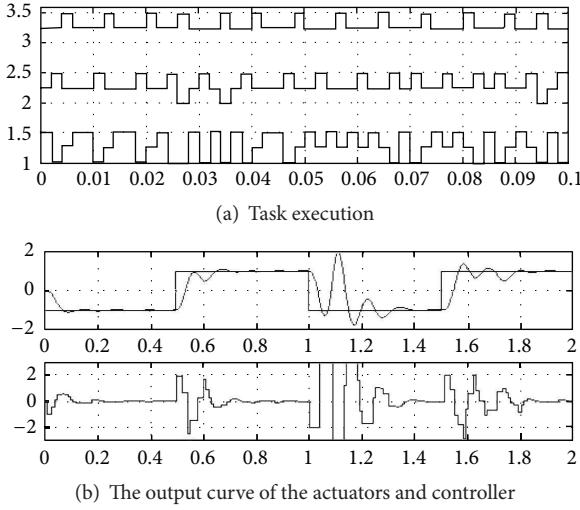


FIGURE 6: Simulation results without using fuzzy feedback scheduling.

TABLE 5: Mapping relations between resource utilization and tasks period.

U	Task period
0.2	4
0.4	8
0.6	16

greater importance. In this regard, the task with ID “16202” and that with ID “7363” stand at the first place, even though they have longer deadline. But the deadline of the task with ID “12856” is right behind the aforementioned two, thus making this task prior to that with ID “8679.” The task with ID “8679” performs worst. So it is prior to the tasks with ID “16202” and “7363.” With the smallest deadline, the task with ID “16202” is prior to that with ID “7363.”

4.2. Period Adjustment Examples. A robot control system simulation platform based on CPS architecture is developed by means of toolbox of Matlab/TrueTime, which is shown in Figure 5. The module consists of three major parts: global scheduling server, server nodes. and network. The toolbox

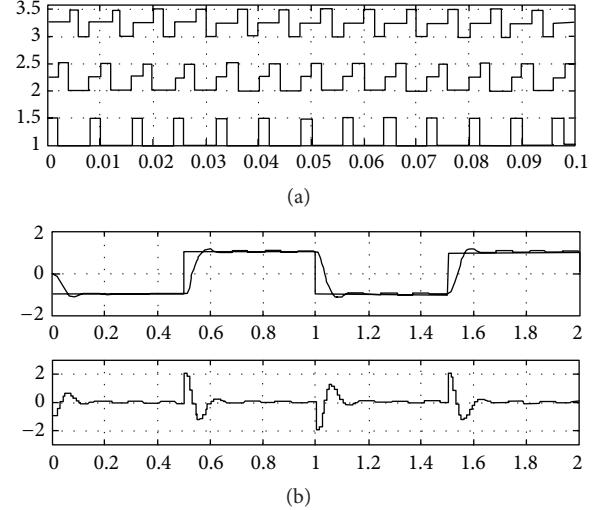


FIGURE 7: Simulation results using fuzzy feedback scheduling.

of TrueTime Kernel contributes to this integrated model. The global scheduling server separately controls the three actuators; each close-loop control system represents a task as T_i . Sensors send the outputs of actuators to the global scheduling server which then computes the priority and adjusts the period according to the feedback information.

A comparison study of using and not using fuzzy feedback scheduling is conducted. Suppose the interpolation period of three control loops is 4 ms. The three tasks cannot be scheduled without using fuzzy feedback scheduling. The task execution is shown in Figure 6(a) and the output curve of three actuators and controller in Figure 6(b). By using fuzzy feedback scheduling, the interpolation period of three control loops is adjusted to 8 ms. The three tasks can be scheduled. The task execution is presented in Figure 7(a) and the output curve of three actuators and controller in Figure 7(b). Obviously, the fuzzy feedback scheduling method improves task schedulability and reduces the error of robot control system.

5. Experimental Results

There is one global scheduling server and two server nodes. They control two 6-DOF mechanical arms under such operating environment: YANXIANG EC3-1711CLDNA IPC; CPU: Genuine Intel(R) processor 600 Mhz; Memory: 599 Mhz, 480 MB; Hard Disk: 60 G; Operating System: Windows XP, Ardence RTX 8.1. The experimental operation object (Figure 8) is six-degree-of-freedom robot arm of Googol Tech (GRB3016).

5.1. Real-Time Test and Analysis. After testing a dozen sets of data, the response time of the control tasks is recorded in Figure 9. As Figure 9 shows, the average response time of control task is 8.323 ms with the presence of fuzzy feedback scheduling method and 10.2387 ms with the absence of the scheduling method. The average response time is much shorter when the scheduler is in place. It implies that the scheduling module can, to some extent, mitigate delay



FIGURE 8: 6-DOF mechanical arm.

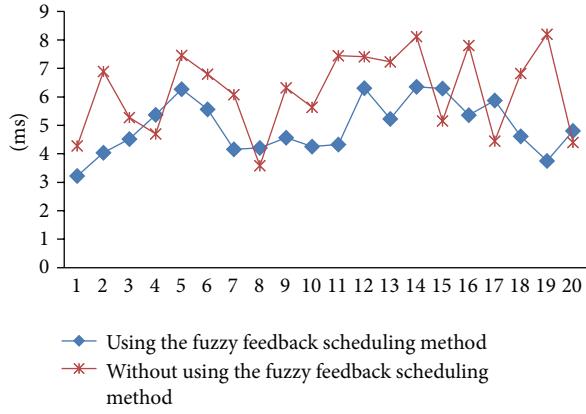


FIGURE 9: The response time of the control task.

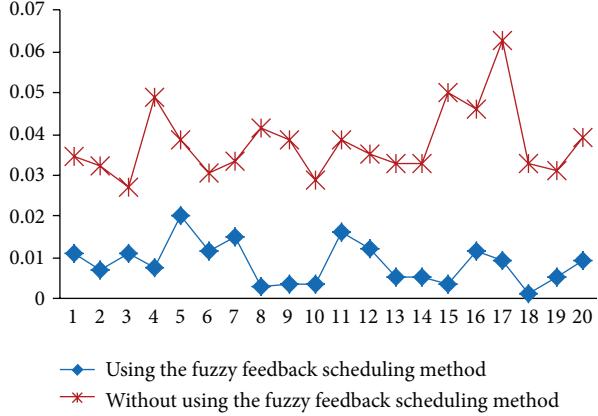
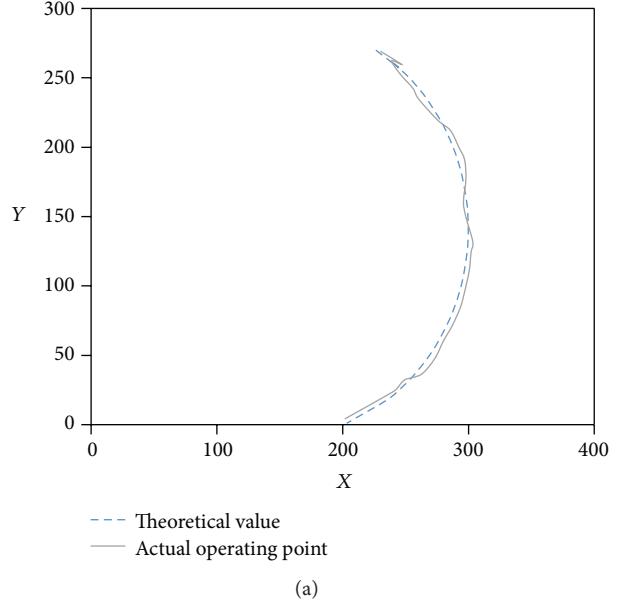


FIGURE 10: Packet loss rate test without using the scheduling module.

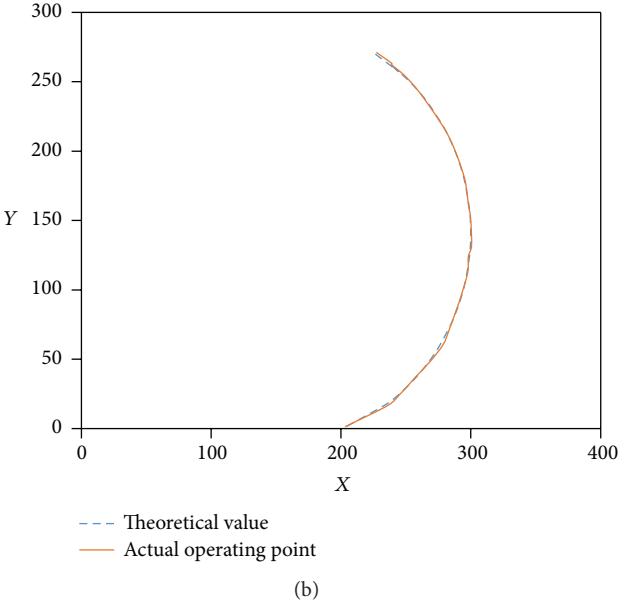
between tasks caused by transmission congestion and ensures the real-time performance of the system.

5.2. Packet Loss Ratio Test. The deadline of the tasks is set at 20 ms and a test program for interference is available. Figure 10 shows a total packet loss ratio of 3.69% without using the scheduling method and 0.76% with the use of the scheduling method. The latter is significantly lower than the former. The experiment proves that the scheduling method not only ensures lower packet loss ratio but also effectively improves the utilization.

5.3. Comparison of Operating Error. Three teaching points are selected: (200.482, 282.073), (299.878, 141.112), and (200, 0). There are 39 interpolation points. The experimental



(a)



(b)

FIGURE 11: Experimental trajectory of circular interpolation.

trajectory of circular interpolation is described in Figure 11. Figure 11(a) depicts the experimental results without using fuzzy feedback scheduling method and Figure 11(b) with the use of scheduling method. The dashed line refers to the theoretical value while the full line represents the actual operating point.

From the error graph (Figure 12), it is learned that the maximum error is 1.86 mm with the use of the fuzzy feedback scheduler. The manipulator can accord to the predetermined path in a smaller error range. On the contrary, not using the fuzzy feedback scheduler results in the maximum error of 5.879 mm. Evidently, the fuzzy feedback scheduling method can reduce the error of robot control system and improve the system's performance.

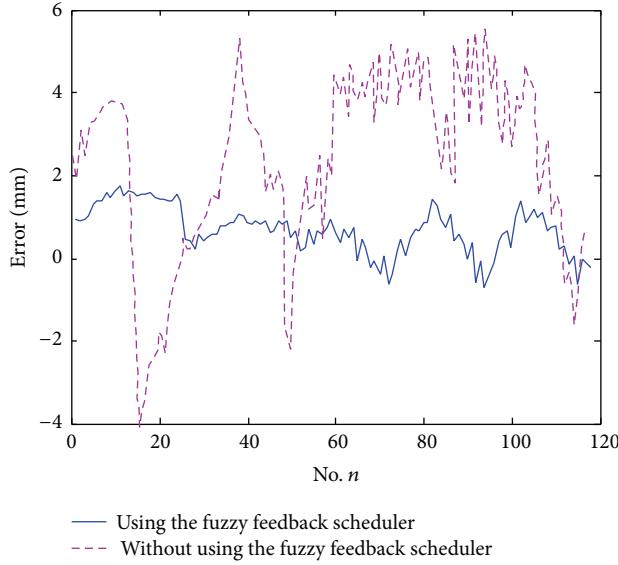


FIGURE 12: Error curve of circular interpolation.

6. Conclusion

The fuzzy feedback scheduling method in this paper can be applied to the cyber-physical systems for robot control. The proposed two-level scheduling framework dynamically adjusts the priority and period. Priority is adjusted according to the task's deadline, its importance, and performance feedback, while the period is adjusted according to the total resource utilization. The method controls the traffic of the tasks to ensure schedulability, reduce packet loss and delay, and realize sounder performance compared with the system with a fixed period. The experimental results indicate that the method can effectively handle slow response and high packet loss caused by the introduction of bus into the network.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61262013), Science and Technology Program of Guangzhou (2013Y2-00100), and the Fundamental Research Funds for the Central Universities (2014ZM0048).

References

- [1] J. Wan, H. Yan, H. Suo, and F. Li, "Advances in cyber-physical systems research," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 11, pp. 1891–1908, 2011.
- [2] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems—A Cyber-Physical Systems Approach*, LeeSeshia.org, 2011.
- [3] J. F. Wan, D. Li, Y. Q. Tu, P. Zhang, and F. Li, "A survey of cyber physical systems," in *Proceedings of the IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, Kunming, China, March 2011, <http://www.jiafuwan.net/publications.html>.
- [4] J. Z. Li, H. Gao, and B. Yu, "Concepts, features, challenges, and research progresses of CPSs," *Development Report of China Computer Science*, pp. 1–17, 2009.
- [5] Y. Tan, M. C. Vuran, and S. Goddard, "Spatio-temporal event model for cyber-physical systems," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS '09)*, pp. 44–50, Montreal, Canada, June 2009.
- [6] H. Huang, Y. Sun, Q. Yang et al., "Integrating neuromuscular and cyber systems for neural control of artificial legs," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs '10)*, pp. 129–138, ACM, April 2010.
- [7] X. Koutsoukos, N. Kottenstette, J. Hall et al., "Passivity-based control design for cyber-physical systems," in *Proceedings of the International Workshop on Cyber-Physical Systems-Challenges and Applications (CPS-CA '08)*, Santorini Island, Greece, 2008.
- [8] M. Chen, J. Wan, and F. Li, "Machine-to-machine communications: architectures, standards and applications," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 2, pp. 480–497, 2012.
- [9] C. Zou, J. Wan, M. Chen, and D. Li, "Simulation modeling of cyber-physical systems exemplified by unmanned vehicles with WSNs navigation," in *Proceedings of the 7th International Conference on Embedded and Multimedia Computing Technology and Service*, pp. 269–275, Gwangju, Republic of Korea, 2012.
- [10] J. Wan, M. Chen, F. Xia, D. Li, and K. Zhou, "From machine-to-machine communications towards cyber-physical systems," *Computer Science and Information Systems*, vol. 10, no. 3, pp. 1105–1128, 2013.
- [11] Object Management Group, *Real-Time CORBA Specification*, Version 1.1, 2002.
- [12] Y. Zhang, C. Gill, and C. Lu, "Reconfigurable real-time middleware for distributed cyber-physical systems with aperiodic events," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 581–588, IEEE, Beijing, China, July 2008.
- [13] OMG, *Real-Time CORBA Specification Version 2.0*, 2003.
- [14] J. F. Wan, D. Li, H. H. Yan, and P. Zhang, "Fuzzy feedback scheduling algorithm based on central processing unit utilization for a software-based computer numerical control system," *Journal of Engineering Manufacture*, vol. 224, no. 7, pp. 1133–1143, 2010.
- [15] J. F. Wan and D. Li, "Fuzzy feedback scheduling algorithm based on output jitter in resource-Constrained embedded systems," in *Proceedings of the International Conference on Challenges in Environmental Science and Computer Engineering (CESCE '10)*, pp. 457–460, Wuhan, China, March 2010.
- [16] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 46–61, 1973.
- [17] K. M. Zuberi and K. G. Shin, "Scheduling messages on controller area network for real-time CIM applications," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 310–316, 1997.
- [18] Z.-P. Chen and H.-Q. Xu, "Research on network control system using improved EDF dynamic scheduling algorithm," *Advanced Materials Research*, vol. 403–408, pp. 2420–2423, 2012.

- [19] J. Wang, "A scheduling algorithm based on communication delay for wireless network control system," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 20, pp. 3891–3895, 2012.
- [20] O. Esquivel-Flores, H. Benítez-Pérez, P. Méndez-Monroy, and J. Ortega-Arjona, "Bounded communication between nodes of a networked control system as a strategy of scheduling," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 27, no. 6, pp. 481–502, 2012.
- [21] X. Zhu, P.-C. Huang, S. Han, A. K. Mok, D. Chen, and M. Nixon, "MinMax: A sampling interval control algorithm for process control systems," in *Proceedings of the 18th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '12)*, pp. 68–77, Seoul, South Korea, August 2012.
- [22] F. Xia, X. Dai, Z. Wang, and Y. Sun, "Feedback based network scheduling of networked control systems," in *Proceedings of the 5th International Conference on Control and Automation (ICCA '05)*, pp. 1231–1236, Budapest, Hungary, June 2005.
- [23] Z.-X. Li, W.-L. Wang, B.-C. Lei, and H.-Y. Chen, "Message scheduling based on fuzzy feedback in networked control systems," *Acta Automatica Sinica*, vol. 33, no. 11, pp. 1229–1232, 2007.
- [24] D. Simon, D. Robert, and O. Senane, "Robust control/scheduling co-design: application to robot control," in *Proceedings of the 11th IEEE Real Time and Embedded Technology and Applications Symposium*, pp. 118–127, San Francisco, Calif, USA, March 2005.
- [25] S.-G. Liu, J.-M. Wei, and Z.-C. Zhu, *Fuzzy Control Technology*, China Textile Press, Beijing, China, 2001.
- [26] Q.-G. Han and X.-Q. Wu, *Computer Fuzzy Control Technology and Instrumentation*, China Measurement Press, Beijing, China, 1999.

