

一种 CANbus 动态优先级策略的研究与仿真

袁 熹¹, 刘朝晖^{1*}, 吴志强², 韩文兴²

(1. 南华大学 计算机科学与技术学院, 湖南 衡阳 421000;

2. 中国核动力研究设计院核反应堆系统设计技术重点实验室, 四川 成都 610041)

摘 要:在通信确定性要求相对较高的系统中,若采用静态优先级调度策略,优先级低的节点可能长时间竞争不到总线,从而导致信息得不到发送.为了克服静态优先级的这种缺陷,可以提出一种新的动态优先级调度策略:以固定的周期遍历每个节点,并根据每个节点获得总线仲裁的频率来动态调整优先级.与此同时,每个节点调度的预值区间和阈值可根据需要灵活设定.通过 Matlab 对原始 CANbus 的静态优先级调度策略和本动态调度策略进行仿真对比研究,本策略的可行性即可获得验证.通信的确定性亦可得到保证.

关键词:优先级;CAN 总线;调度策略;仿真

中图分类号:TP273

文献标识码:

Study and Simulation of a New Dynamic Priority Strategy of CAN

YUAN Xi¹, LIU Zhao-hui^{1*}, WU Zhi-qiang², HAN Wen-xing²

(1. School of Computer Science and Technology, University of South China, Hengyang,

Hunan 421001, China; 2. Key Laboratory of Reactor System Design Technology of

Nuclear Power Institute of China, Chengdu, Sichuan 610041, China)

Abstract: In the system requiring comparatively higher deterministic of communication, static priority scheduling algorithm may bring about that the nodes of lower priority fail to access to the bus within a long period, which leads to a result that some data couldn't be sent. To overcome this defect, a new dynamic priority scheduling strategy is put forward: Traverse each node periodically and then adjust the priority of each node dynamically as per its bus arbitrary frequency. In the meantime, the pre-value interval and threshold of each node could be flexibly specified as required. Once the emulational comparison between the original CANbus static priority scheduling strategy and this new dynamic strategy is proceeded, the feasibility of the latter approach would be verified. The deterministic of

收稿日期:2014-12-15

作者简介:袁 熹(1988-),男,湖北黄冈人,南华大学计算机科学与技术学院硕士研究生.主要研究方向:现场总线.* 通讯作者.

communication would be guaranteed as well.

key words: priority; CAN; scheduling algorithm; simulation

0 引言

CANbus 总线现在已经成为一种广泛使用的现场总线,已经应用在工业控制、汽车电子以及机械制造等领域^[1].并广泛地应用于分布式监测和控制系统,其主要原因是因为 CAN 总线是一种多主总线,并且具有可靠性高、实时性好,数据传输速率高等优点.

在 CANbus,即控制局域网^[2-3]中,数据链路层协议采用的是 CSMA/CD 机制,总线空闲时,每个需要发送数据的节点都可以参与总线竞争.若有多组报同时竞争总线仲裁时,具有高优先级的报文获得总线仲裁,发送数据.这样就会带来一个问题:优先级低的节点(如 Node_ID = 11111111)可能长时间在竞争中失败,数据无法发送.

按照周期性划分,实时系统任务可分为周期性和非周期性^[4].按实时任务对计算完成时间的约束要求分为硬实时任务和软实时任务.在软实时系统中,任务的短暂延时是可以容忍的,如音视频传送、化工生产等,但是,若相当比例的软实时任务长时间得不到响应,这也是不能容忍的.硬实时任务要在确定的截止期限内完成,否则可能会带来严重的后果,如汽车刹车制动系统、电气跳闸、核反应堆控制系统等等.实时系统可分为:保证响应(Guaranteed-Response)实时系统和最大努力(Best-Effort)实时系统^[5].

大多数系统在现实中都混合了软实时和硬实时任务.显然,对于系统中的硬实时任务,CAN 协议的“碰撞竞争”带来的问题是绝对不允许存在的,硬实时任务必须要在其时限内完成;这包括两方面的含义,一是实时性要得到保证,任务执行时间短;二是确定性也要得到保证,该任务必须在规定的时间内完成任务.

Liu 在文献[6]中论述了 RM 算法是最优的静态优先级调度算法,RM 调度算法的本质还是固定优先级调度的算法,只是在原有 CAN 协议优先级调度的基础上做了优化,仍然具有固定优先级调度算法的缺陷.静态调度算法的优点是运行开销小,可预测性强,但是,静态调度算法在运行期间是无法改变执行顺序的,它的灵活性较差.因此,静态调度适合于问题需求比较确定的情况,如工业过程控制,而不适合于不可预测环境下.

相对截止期调度算法 EDF^[7-10]是目前动态优先级策略中最优算法,距离任务截止期最近的任务被分配最高优先级,任务就绪队列中队首具有最高优先级.EDF 算法可以带来更低的调度和任务迁移的开销^[11].当系统负载率较低且所有任务周期是固定时,该算法是非常有效的,并且能保证所有任务可调度,满足时限要求;但是,当系统负载较大或者有部分偶发任务时,系统花费在调度上的开销加剧,系统性能急剧下降,会引起大量任务错过截止期.

为了使 CAN 总线更好适用于高可靠网络和多种任务混合系统的要求,充分发挥其优势,本文提出了一种新的动态优先级调度的方法.该方法在不影响系统实时性的同时,规避了静态优先级固有的缺陷,在一定程度上确保了系统通信的确定性;同时也弥补了 EDF 调度算法应对偶发任务的不足.在混合弱实时任务和偶发任务的过程控制系统中,该方法可以很好的保证系统的实时性和确定性.降低网络负载是提升网络性能最有效的方法^[12],而从仿真结果来看,该策略在网络负载较大时,优势尤为明显.

1 基于总线访问频率的动态优先级调度策略

1.1 策略介绍

动态调度算法指的是系统在运行时对网络中传输消息的优先级进行动态的构建,可调度分析也必须是动态进行,所以大大减少了在系统运行前的工作量.本文提出的基于总线访问频率(AF, Access Frequency)动态优先级调度策略的核心思想是:在系统启动之前,每个节点的优先级固定且不同,称为节点初始优先级.在运行过程中记录每个节点已访问总线的频率(AF),总线仲裁之前根据 AF 值来动态调整每个节点的优先级.每个节点的 AF 值也是在动态变化的,AF 值低的节点,其优先级动态递增(提高);当 AF 值低于阈值 F_{th} (Frequency Threshold),则将其优先级直接设为最高.

当优先级低的任务被高优先级任务长时间阻塞时,其 AF 值就会下降,当 AF 值下降到调度算法中预先设置的值时,就提高任务优先级以增大获得总线仲裁的机会.对于强实时偶发任务,直接动态设定其优先级为最高.

对 Node_{*i*}(第 *i* 个节点)的 *AF* 值 Node_*AF_i* 定义如下:

$$\text{Node_}AF_i = S_i / \sum_i S_i, S \geq 0 \quad (1)$$

其中, *n* 为总线挂载节点总数, *S_i* 为 Node_{*i*} 已获得总线访问的次数, 其意义为 Node_{*i*} 参与总线通信的频率。

定义 1(理想调度): 不同于最优调度, 理想调度重点不是避免“碰撞”, 而是期望所有节点的 *AF* 值相等, 即所有节点能够“均匀”获得总线的仲裁。其定义如公式(2)

$$\begin{aligned} k * n &= \sum_i S_i, (k = 1, 2, 3 \dots) \text{ 时刻有:} \\ \forall i, j, 1 \leq i, j \leq n, i &\neq j \\ \text{Node_}AF_i &= \text{Node_}AF_j \end{aligned} \quad (2)$$

则称该调度为理想调度。

系统刚上电后有些任务还没产生, 若此时开始采集 *AF* 值, 则可能有部分节点的 *AF* 值是“0”, 此时采集的值是没有意义的。因此, 每一次 *AF* 值采集的时间点是在所有节点访问总线次数之和是节点数的整数倍的时候。一是可以保证所有节点有时间完成初始化; 二是该值得参考点是 $1/n$ 。我们期望在整个调度过程中, 调度都是处于理想调度状态下。在该状态下, 所有节点都可以在合理、可预测的时间内获得总线仲裁, 高效利用总线资源, 确保实时数据的时限。但是, 在实际情况中, 由于偶发任务或者非周期任务的存在, 总线不可能总是处于理想调度的状态。为了评价这种系统偏离理想调度状态的程度, 我们提出了一个评价指标: 调度偏离值 (*SD*, Scheduling Deviation)。

定义 2(调度偏离值): 调度偏离值是用来评价调度偏离理想调度的程度。其值越大, 表示偏离程度越大, 反之, 则说明调度状态越靠近理想调度。如公式(3)

$$SD = \sum_i (S_i / \sum_i S_i - 1/n)^2 \quad (3)$$

在理想调度的状态下, $SD = 0$; 在最糟糕的情形下, 只有一个节点占用总线, $SD = n + 1/n - 2$ 。 *SD* 值越小, 说明调度越稳定。如果系统中所有节点都是产生的周期性数据, 通信速率一定, 系统稳定后, 总线总的访问次数和时间成正比, 即 $kt = S_1 + S_2 + \dots + S_n$ 其中 *t* 为时间, *k* 为一个常数, *S_i* 为 Node_{*i*} 已获得总线仲裁次数。

定理 1 若系统通信速率一定且每个节点的周期固定, 调度偏离值的范围为 $[0, (1 - 1/n)]$ 。

证明 令关于 *S_i* 函数

$$F(S_1, S_2 \dots S_n) = \sum_i S_i^2 + \lambda (\sum_i S_i - \kappa t),$$

$$\frac{\partial F}{\partial S_1} = 2S_1 + \lambda; \dots \frac{\partial F}{\partial S_n} = 2S_n + \lambda$$

令上述偏导数等于 0, 取极值点, $S_1 = S_2 = \dots S_n = -\lambda/2$, 则可求得 $F(S_1, S_2 \dots S_n)$ 的极小值为 $[(\kappa t)]^2/n$, 则 $SD_{\min} = 0$;

又 $\sum_i S_i^2 \leq (\sum_i S_i)^2 = (\kappa t)^2$, 所以, $SD_{\max} = 1 - 1/n$ 。

故原命题得证。

证毕。

定理 1 说明了, 根据各节点任务的周期, 每个节点选取合适的 F_{t1} , F_{t2} 和 F_{th} 值, 系统是可以达到理想调度的状态的; 当系统 $SD = 1 - 1/n$ 时, 则表明有节点长时间占据总线, 导致其他节点阻塞。当 $n \rightarrow \infty$, *SD* 收敛于 1, 换就话说, 系统 *SD* 值实际上是不可能到达 1, 反之, 则说明系统可能出现了故障。

1.2 实现原理

该策略是在总线由“空闲”(idle)状态转为“忙碌”(busy)中有效, 并且其执行时间在总线仲裁之前, 其间根据各节点当前 *AF* 值来动态设定各节点优先级。在这个过程中, 节点优先级可能会改变, 也可能不改变, 甚至可能设为最高。在下一轮收发周期之前, 等待收发的节点的优先级保持不变, 当前收发周期结束后, 当前占用总线节点优先级回归到初始设定值。如图 1, 策略模型, 其中 *P* 表示节点的优先级, *P* 值越大, 优先级越高, $P_{\max}$ 为最高优先级。

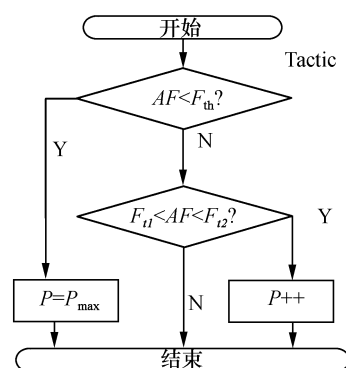


图 1 策略流程图

Fig. 1 Strategy flowchat

如图 1, 其中 F_{th} , F_{t1} 和 F_{t2} 的值可以根据具体的系统及其应用场景, 通过一定的计算, 由领域专

家来设定,设定的依据可以是该节点数据在系统中的重要性和周期等等.对于系统中的重要节点,如在核电的反应堆控制系统中,对于反应堆堆内温度、压力采集节点等,由于它们是系统的关键安全参数,我们偏向于将其优先级预设较高, F_{th} 、 F_{rl} 、 F_{r2} 值设低.而对于系统中关键节点或可能产生偶发硬实时任务的节点,如保护动作节点,可以直接将其 P_{max} 单独设定一个较大值,这样可以保证关键节点的数据能优先抢占总线.

2 仿真

Simulink/Stateflow 是非常有名的工具链,在控制系统、自动化和飞机制造业等众多领域的建模仿真中,已成为事实上的标准.本方案采用 Simulink/Stateflow 对原始的 CANbus 协议和加上动态优先级晋升策略之后的改进 CANbus 协议进行仿真研究.分别从网络性能和 SD 值来分析该策略的可行性及优越性.

2.1 仿真方案及配置

整个方案的建模分为三部分:系统 Simulink 建模、CANbus 协议 Stateflow 建模和现场节点建模.系统总线挂载 9 个节点,都是产生周期任务;数据大小固定在 8 Byte 内;无 CANbus 错误处理机制;通信速率采用 1Mbit/s. Matlab7.0,步长为适合离散系统的变步长,ode45 解法,仿真时间设为 inf. 仿真在 CPU 为 Intel Celeron 主频 2.5 GHz,内存 2.0 GB 的 PC 机上进行.

2.2 总线通道模型

总线通道模型有两种状态:空闲(idle)和忙碌(busy),缺省进入空闲状态,每一次从空闲到忙碌状态的跳转,就有一帧数据被处理,处理完了之后经过延时函数(模拟帧间空间),跳回空闲状态,等待下一个任务.在空闲转向忙碌之前调用 Tactic 和 scheduling 函数,以实现策略和仲裁函数(原始 CANbus 协议仿真中无 Tactic 函数),只有 scheduling 函数返回值为“真”时,总线才进入忙碌状态,如图 2.

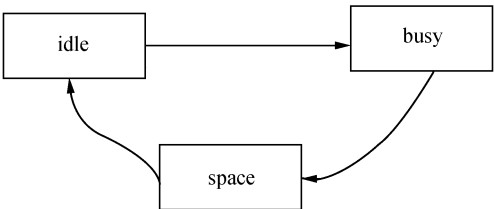


图 2 总线通道模型
Fig. 2 Bus channel model

2.3 策略模型

本文提出调度策略和竞争仲裁函数是由图形函数和 m 函数结合组成的,如图 3. 图形函数 compete 和 M 函数 m_compete 结合完成优先级仲裁功能,其作用是选取本轮通信中已置位竞争的最高优先级. M 函数 scheduling 完成本文提出的策略算法.从初始优先级最低的节点,第 18 号节点开始,依次倒序判定每个节点 AF 值是否在合理区间,并对各节点优先级动态调整.在竞争仲裁函数中,也采用倒序判定.若因为 scheduling 调度函数对节点优先级的动态调整而导致出现相同优先级,倒序判定也可以保证初始低优先级节点获得仲裁,因为在此种状况下,预设低优先级任务更接近其时限.

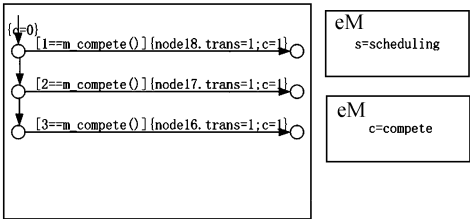


图 3 策略调度模型
Fig. 3 Strategy scheduling model

3 仿真结果及分析

如图 4 为对比仿真中 SD 值随时间的变化;图 5 为低优先级节点组(第 7~9 号节点)参与总线通信的频率;图 6 为时延对比图;图 7 丢包率对比图.粗线为原始 CANbus 协议结果,细线为加上策略的 CANbus 协议结果.

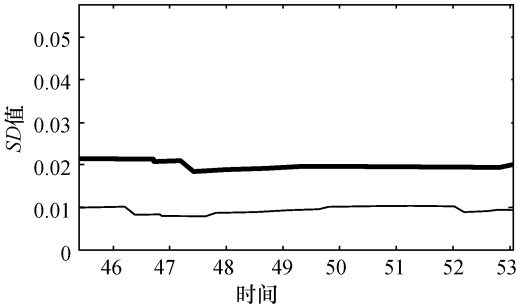


图 4 SD 值对比
Fig. 4 SD value contradistinction

受计算机内存等因素的限制,该仿真不能持续一个较长的时间,而且系统稳定之后,各个值也

趋于稳定. 上述两组图是选取了系统稳定之后的结果. Matlab 中采用的时间是仿真时间, 不等于实时时间, 因此, 仿真结果只能用于对现实系统做定性的分析, 得到的数值是不能作为真实系统的参考, 更不能应用. 如图 4, SD 值都维持在一个较低水平, 说明该仿真系统是处于正常运行中. 但是, 对于特定的系统, 受节点数目、系统波特率和处理器数目、效率等等因素的限制, 反应系统正常运行的 SD 值是不一样的, 不能一概而论. 在同样的条件下, 该优先级策略能合理分配总线资源, SD 值较加入策略之前提高了大约 10%. 当网络负载较大时, 这一特性更为明显, 甚至可以将 SD 值提高一个数量级. 这说明该策略可能更适用于负载率高的网络. 如图 5, 为低优先级节点组参与总线通信的频率, 值越大表明参与越频繁. 如图所示, 加上策略之后低优先级节点组的平均 AF 值增大, 说明低优先级节点参与通信的频率增加. 换句话说, 低优先级节点被阻塞的可能性降低.

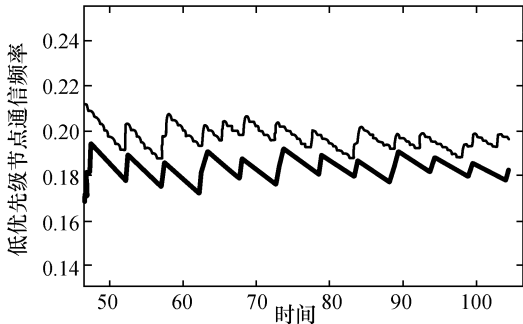


图 5 低优先级节点参与频率

Fig. 5 Low priority nodes access frequency

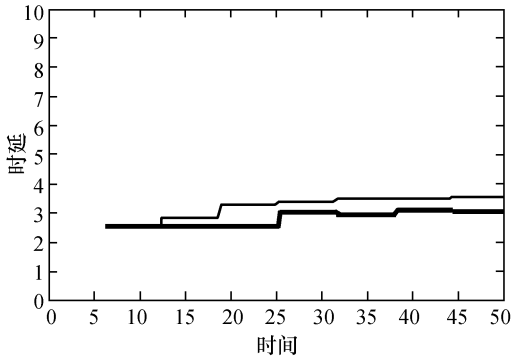


图 6 时延对比图

Fig. 6 Time delay contradistinction

仿真选取了反应网络性能的四个主要指标: 时延, 丢包率, 吞吐量和信道利用率, 由于版面有限, 选取了时延和丢包率的对比图. 如图 6, 提出

算法的平均时延较原始算法大, 但是这个差距不是非常明显, 因为在保证确定性的时候, 不可避免的需要系统部分开销, 在需要高确定性通信的场合, 损失实施性能是不可避免的. 如图 7, 两种算法的丢包率持平, 经过较长时间的仿真观测, 最后两者基本都维持在 20%, 提出算法没有明显改善通信的丢包率.

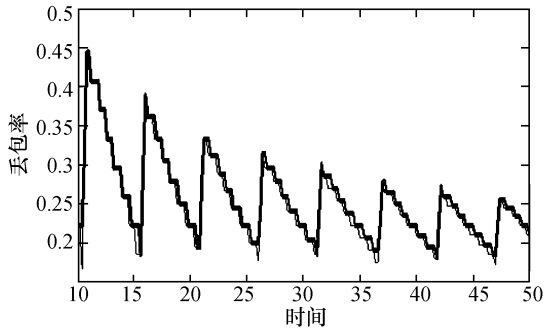


图 7 丢包率

Fig. 7 Package loss rate

在确定性方面, 该策略的算法可以从根本上保证系统通信的确定性. 这是因为每个节点的预值区间和阈值不必一致, 可以根据该节点在系统中的功能、重要性等因素来设定. 设定的值越小, 节点优先抢占资源的几率就越大, 其确定性也就得以保证.

4 不足及下一步工作

本策略的仿真是纯软件仿真, 由于 Matlab 仿真的效果受到计算机本身的限制, 所以在可接受的范围内, 与实际情况有偏差. 下一步的工作是将该策略放到半实物仿真和真实场景中, 进一步验证其可行性和上述结论, 并着手该算法和 EDF 调度算法对比的研究工作.

参考文献:

[1] 关学峰. CAN 总线网络的实时性能分析_改进和测试 [D]. 成都: 西华大学, 2008.

[2] 刘亚南. 基于 CAN 现场总线测控网络技术研究 [D]. 青岛: 青岛大学, 2009.

[3] 李美玲. 基于 CAN 总线的主从式数控系统通信的研究与实现 [D]. 太原: 太原理工大学, 2008.

[4] Sprunt B, Sha L, Lehoczky J. Aperiodic task scheduling for Hard-Real-Time systems [J]. Real-Time Systems, 1989, 1(1): 27-60.

取,大大的提高了数据的抓取能力.随着舆情监测系统的不断深入,对于网站数据的采集也提出更高的要求,如何抓取图片、声音、影视文件等非结构化数据,是本系统的一个重要缺陷,如果利用当前强大的云技术来进一步提高系统的相应能力,提高系统的工作效率和系统的吞吐量是本系统接下来要进行进一步研究的主要方向.

参考文献:

- [1] 徐强,万亚平,雷龙眼,等.石鼓区网络舆情监控系统的设计与实现[J].南华大学学报(自然科学版),2013,27(3):2-5.
- [2] Brin S,Page L. The anatomy of a large-scale hypertextual web search engine[J]. Computer Networks and ISDN Systems,2009,30(1-7):107-117.
- [3] Yang Y,Carbonell J G,Brown R D,et al. Learning approaches for detecting and tracking new events[J]. Intelligent Systems and their Applications,IEEE,1999,14(4):32-43.
- [4] 吴丽辉,王斌,余智华.一种通用Web信息采集系统的设计与实现[J].计算机工程,2005,31(3):2-3.
- [5] 冯如晓,刘志明,雷龙艳.基于搜索引擎的关键词舆情过滤算法研究[J].电脑知识与技术,2014,10(6):1327-1332.
- [6] 蒋辉,阳小华,刘志明,等.基于一种文档表示模型的站内搜索引擎设计与实现[J].南华大学学报(自然科学版),2013,27(4):77-81.
- [7] 罗刚,王振东.自己动手写网络爬虫[M].北京:清华大学出版社,2010:11-23.
- [8] 杜阿宁.互联网舆情系统挖掘方法研究[D].黑龙江:哈尔滨工业大学,2007:24-25.
- [9] 陈丽萍.校园BBS舆情数据手机与提取系统的设计与实现[D].武汉:华中科技大学,2009.
- [10] 李玉华.面向主题的舆情采集搜索爬虫的设计与实现[D].济南:山东大学,2009:27-30.
- [11] 褚宏爽.主题搜索引擎网络爬虫的设计与实现[D].北京:北京邮电大学,2012:30-32.
- [12] Menczer F,Pant G,Srinivasan P. Topic web crawlers: Evaluation adaptive algorithms[J]. ACM Trans. on Internet Technologies,2003,26(1):89-113.
- [13] 初建崇,刘培卫. web 文档中词语权重计算方法的改进[J]. 计算机工程与应用,2007,43(19):192-194.
- [14] 代宽,赵辉,韩冬,等.基于向量空间模型的中文网页主题特征项抽取[J]. 吉林大学学报(信息科学版),2014,32(1):88-93.
- [5] Bums A,McDermid J A. Real-safety system: analysis and synthesis[J]. Software Engineering Journal,1994,9(6):267-281.
- [6] Liu C L,Layland J W. Scheduling algorithms for multiprogramming in a hard-real-time environment[J]. Journal of the ACM (JACM),1973,20(1):46-61.
- [7] Livani M,Kaiser J,Jia W J. Scheduling Hard and Soft Real-time Communication in the Controller Area Network[C]. Workshop on Real-time Programming,1998.
- [8] Natale M D. Scheduling the CAN Bus with Earliest Deadline Techniques[C]. 21st IEEE Real-Time Systems,2000.
- [9] Zuberi K M,Shin K G. Non-preemptive Scheduling of Messages on Controller Area Network for Real-time Control Applications[C]. 1st IEEE Real-Time Technology and Applications,1995.
- [10] Santiago F,Francisco R,Alberto B. Software Based EDF Message Scheduling on CAN Networks[C]. Proceedings of the Second International Conference on Embedded Software and Systems,2005.
- [11] Anderson J H,Bud V,Devi U C. An EDF-based Scheduling Algorithm for Multiprocessor Soft Real-Time Systems[C].//Euro-micro Conference on Real-time Systems. IEEE Computer Society,2005:199-208.
- [12] 姚荣斌,邓睿,孙成祥.网络化控制系统的RM调度优化研究[J].淮阴师范学院学报(自然科学版),2013,12(4):327-330.

(上接第101页)