

优先级周期性互换的实时调度算法

王彬^{1*}, 王聪², 薛洁³, 刘辉¹, 熊新¹

(1. 昆明理工大学 信息工程与自动化学院, 昆明 650500; 2. 深圳市易之威计算机技术有限公司, 广东 深圳 518040;

3. 云南警官学院 信息网络安全学院, 昆明 650223)

(* 通信作者电子邮箱 wangbin1@vip.sina.com)

摘要: 针对实时多任务调度时低优先级任务的延迟问题, 提出了一种优先级周期性互换的静态优先级调度算法。该方法以固定的时间片为周期, 对多任务系统中的某两个不同优先级的独立性任务, 周期性地互换它们的优先级级别, 在保证较高优先级任务的执行时间的前提下, 使得较低优先级的任务有机会尽快执行, 以缩短其执行过程中的延迟时间。所提方法能有效解决低优先级任务的实时性问题, 从而提高实时多任务系统的整体控制性能。

关键词: 优先级互换; 实时多任务控制; 周期性互换; 静态调度算法; 时间片

中图分类号: TP301.6 **文献标志码:** A

Real-time scheduling algorithm for periodic priority exchange

WANG Bin^{1*}, WANG Cong², XUE Jie³, LIU Hui¹, XIONG Xin¹

(1. Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming Yunnan 650500, China;

2. Shenzhen Easyway Computer Technology Company Limited, Shenzhen Guangdong 518040, China;

3. Faculty of Information Network Security, Yunnan Police Officer Academy, Kunming Yunnan 650223, China)

Abstract: A static priority scheduling algorithm for periodic priority exchange was proposed to resolve the low-priority task latency problem in real-time multi-task system. In this method, a fixed period of timeslice was defined, and the two independent tasks of different priorities in the multi-task system exchanged their priority levels periodically. Under the precondition that the execution time of the task with higher priority could be guaranteed, the task with lower priority would have more opportunities to perform as soon as possible to shorten its execution delay time. The proposed method can effectively solve the bad real-time performance of low-priority task and improve the whole control capability of real-time multi-task system.

Key words: priority exchange; real-time multi-task control; periodical exchange; static scheduling algorithm; timeslice

0 引言

实时系统最显著的特点是对“时间”的严格要求, 只有在规定时间内产生正确的逻辑才能保证最终结果的正确性^[1]。整个系统的各个部分都有关于时间的非功能约束, 以确保整个实时系统的可靠性。实时任务是实时系统中确保完成某种功能的可供调度的基本单位, 与一般的任务相比较, 实时任务多了一个时效性限制, 必须在有效截止期内完成。实时多任务系统的关键在于多任务调度算法, 因为不同的调度算法会对系统的整体性能有重要的影响。目前在实时系统中实时多任务的调度算法主要采用优先级驱动调度算法, 根据任务优先级的确定时机, 优先级驱动的调度算法分为静态调度和动态调度两种。在静态调度算法中, 所有任务的优先级在设计时就确定下来了, 且在运行过程中不会发生变化; 在动态调度算法中, 任务的优先级则在运行过程中确定, 并可能不断地发生变化^[2]。静态调度比较简单, 但缺乏灵活性, 不利于系统扩展, 动态调度增加了灵活性, 可以处理一直在变化的系统情

况, 但实现过程比较复杂。

由于多任务调度重要性, 很多学者对其进行了深入研究, 提出了很多相关算法, 已被证明效果较好的静态调度算法是单调速率(Rate Monotonic, RM)调度算法^[2-3], 而效果较好的动态优先级算法包括截止时间优先(Earliest Deadline First, EDF)调度算法和最短空闲时间优先级(Least Laxity First, LLF)^[4-5]调度算法。Brandenburg等又提出了几种能使处理器利用率较高的算法, 典型算法如GEDF(Global Earliest Deadline First)^[6]以及Pfair(Proportionate fairness)^[7]。

上述研究针对独立的周期性任务以及有依赖性的非周期任务调度问题进行了较为深入的研究, 但对同时具有依赖关系与周期性的任务却很少涉及。而在工业实时控制领域中, 这类具有依赖关系的周期性多任务调度问题有着极为广泛的应用。在现有的研究中, 无论是静态优先级驱动任务调度还是动态优先级驱动调度, 由于系统总是先执行优先级比较高的任务, 所以低优先级的任务总是要等到高优先级任务的执行间隙才能得到执行。尽管优先级较高的任务得到了保

收稿日期: 2013-09-25; 修回日期: 2013-11-22。 基金项目: 国家自然科学基金资助项目(61263017); 云南省自然科学基金资助项目(2011FZ060); 昆明理工大学人才培养基金资助项目(KKS201303120); 国家留学基金管理委员会资助项目(留金发[2011]5024号)。

作者简介: 王彬(1977-), 女, 黑龙江哈尔滨人, 副教授, 博士, CCF会员, 主要研究方向: 工业实时控制、模型驱动的软件设计、实时数据分析、图像处理; 王聪(1986-), 男, 陕西千阳人, 助理工程师, 硕士, 主要研究方向: 实时操作系统原理、工业实时控制; 薛洁(1969-), 女, 山东济南人, 副教授, 博士, 主要研究方向: 工业实时控制、模型驱动的软件设计、数字集成电路设计; 刘辉(1984-), 男, 陕西蒲城人, 讲师, 博士, 主要研究方向: 图像处理、模式识别、工业实时控制; 熊新(1977-), 男, 安徽金寨人, 高级工程师, 硕士, 主要研究方向: 电机控制、工业实时控制。

证,但低优先级任务很容易因此而产生延迟,使得该低优先级任务的自身控制效果降低,并因此而影响实时多任务控制系统的整体性能指标。本文重点针对这类实时多任务控制系统中低优先级任务的延迟问题展开实时多任务系统的调度算法研究,在时间片轮转(Round Robin, RR)调度算法的基础上提出了一种优先级周期性互换的多任务静态调度算法,其基本思想为:对实时系统中的某两个不同优先级的独立性任务,以固定的时间片为单位周期性地互换它们的优先级级别,这样可以在较高优先级任务的执行时间得到保证的前提下,使得较低优先级的任务有机会尽快执行,以减少低优先级任务执行的延迟时间,从而保证整个控制系统的实时性要求。

1 优先级周期性互换多任务调度算法

优先级周期性互换任务调度算法的基本思想以时间片轮转法调度任务算法中的周期性任务调度机制为基础。时间片轮转任务调度算法属于时间驱动任务调度算法,这种算法将系统时间平均分成等长时间片段,每个就绪态任务的执行时间就是这一小片,等时间片用完或者任务在这一小片时间内执行完毕或是挂起,操作系统就将CPU等资源分配给下一个任务,直到所有就绪态任务都被执行完一遍后,操作系统再把资源分配给最早未执行完的就绪态任务^[8]。

1.1 优先级周期性互换调度算法说明

在目前的研究中,使用优先级周期性互换任务调度算法的任务的可调度条件是所调度的任务是周期性的独立任务,即实时系统中某个功能只需要一个实时任务就可完成,不需要和其他任务之间进行交互,且任务执行时间为固定周期。在验证多任务调度算法的时候,对独立性任务的调度能更直观地显现出调度算法的优劣。为了方便对周期任务调度的描述,本文采用如下符号表示:

- 1) 实时系统任务集为 $\Gamma = \{\tau_1, \tau_2, \dots, \tau_n\}$ 。
- 2) 周期为 $T_1, T_2, \dots, T_n$ 。
- 3) 相位为 $\Phi_1, \Phi_2, \dots, \Phi_n$ 。
- 4) 执行时间为 $C_1, C_2, \dots, C_n$ 。
- 5) 任务优先级为 $P_1, P_2, \dots, P_n$ (P 值越小,优先级越高)。
- 6) 相对截止时间为 $E_1, E_2, \dots, E_n$ 。
- 7) 任务表示为 $\tau_i = (T_i, \Phi_i, C_i, P_i, E_i)$ 。
- 8) 剩余执行时间为 $L'_1, L'_2, \dots, L'_n$ 。
- 9) CPU 利用率为 $U = \sum_{i=1}^n \frac{C_i}{T_i}$ 。

无论采取何种任务调度方式,所有任务可调度的必要条件是:

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1 \quad (1)$$

由于使用优先级周期性互换调度算法的前提是采用某种静态优先级抢占式调度算法后所有任务均可调度。所以,优先级周期性互换任务调度算法不再分析系统所有任务调度的可行性,而是在可调度基础上来分析使用优先级周期性互换任务调度算法的条件。

周期性优先级任务互换调度算法是在两个不同优先级任务之间,按预定的时间片周期性地对这两个任务进行优先级的相互交换。在实时系统多任务调度中,使用优先级周期性互

换调度算法条件如下:

设 $U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1$, 高优先级任务 $\tau_m = (T_m, \Phi_m, C_m, P_m, E_m)$, 低优先级任务 $\tau_n = (T_n, \Phi_n, C_n, P_n, E_n)$; 则高低优先级任务能够互换优先级的条件为:

$$\begin{cases} P_m < P_n \\ L'_m > L'_n \end{cases} \quad (2)$$

且对可能存在两种情况作如下分析。

1) 当较高优先级任务与较低优先级任务之间无其他任务时:对于比任务 τ_m 优先级更高的任务,调换任务 τ_m 与任务 τ_n 之间的优先级对其不产生任何影响;对于比任务 τ_n 优先级更低的任务,任务 τ_m 和任务的 τ_n 执行时间和 $E_m + E_n$ 没有发生变化,因而预留给其他任务的执行时间也没有发生变化,所以也不产生影响。

2) 当较高优先级任务与较低优先级任务之间有其他任务时:对于比任务 τ_m 优先级更高的任务,调换任务 τ_m 与任务 τ_n 之间的优先级对其不产生任何影响;对于介于任务 τ_m 和任务 τ_n 之间的任务,由于 $E_m - L'_m + L'_n < E_m$, 因而对于介于任务 τ_m 和任务 τ_n 之间的任务的总延迟时间减小了 $L'_m - L'_n$;对于比任务 τ_n 优先级更低的任务,任务 τ_m 和任务 τ_n 的执行时间和 $E_m + E_n$ 没有发生变化,因而预留给其他任务的执行时间也没有发生变化,所以也不产生影响。

优先级周期性互换任务调度算法本质上属于静态优先级调度算法,预先确定好多个任务的优先级以及轮换的时间片长度,周期性地互换较高优先级任务和较低优先级任务之间的优先级,使得原较低优先级任务在固定的时间片内轮换为较高的优先级,而原较高优先级任务则在该段时间内轮换为较低的优先级。这样能够让较低优先级任务获得机会尽快执行,在保证较高优先级任务完成时限的要求下减少低优先级任务执行的延迟时间。

1.2 优先级周期性互换调度算法仿真及结果分析

为了验证周期性优先级任务互换调度算法的有效性,选用 Ptolemy II 作为平台进行仿真。Ptolemy II 仿真软件是由美国加州伯克利大学电气工程与计算机科学系研发的复杂嵌入式软件系统软件,主要用于研究实时嵌入式系统中的并行算法仿真及实时嵌入式系统的建模。Ptolemy II 采用面向角色的设计方法很好地解决了不同结构之间的交互,从而实现了异构系统的模拟仿真^[9-10]。Ptolemy II 仿真软件在仿真单处理器多任务环境中使用方便,十分有利于在单处理器多任务环境中仿真各种调度算法的可行性。便于修改调度算法,减小开发成本和时间。

1.1 节中已经对多任务系统中该算法对其他未参与优先级轮换的任务的影响进行了分析,在此研究基础上,设计仿真实验如下。

设在调度系统中有如下3个任务:任务 $\tau_1 = (T_1, \Phi_1, C_1, P_1, E_1)$, $\tau_2 = (T_2, \Phi_2, C_2, P_2, E_2)$, $\tau_3 = (T_3, \Phi_3, C_3, P_3, E_3)$ 。其中: $\tau_1 = (4, 0, 1, 3, 1, 2)$, $\tau_2 = (10, 0, 4, 4, 9)$, $\tau_3 = (20, 0, 5, 5, 14)$ 。使用 Ptolemy II 中的时间多任务域所建立的单处理器多任务系统模型如下所示: TM Director 是多任务调度器; priority1、priority2 和 priority3 分别是任务 1、任务 2 和任务 3 的优先级; DiscreteClock 用来模拟周期性任务的触发,并在其

中设置触发相位参数; Task1、Task2 和 Task3 分别用来模拟 3 个任务,并在其中添加执行时间 ExecutionTime 参数和 priority 参数,用来控制任务的执行时间和优先级。

图 1 为任务仿真建模工程;图 2 是采用基于单调速率调度算法 RM 进行任务调度时的多任务调度仿真。由图 2 可以看出:由于采用了单调速率调度算法任务调度算法,使得任务 Task3 大幅度延迟。此时 Task2 的实际最大执行时间是 6,满足要求,但 Task3 的实际执行时间为 18,超出了相对截止时间 12,达不到要求。

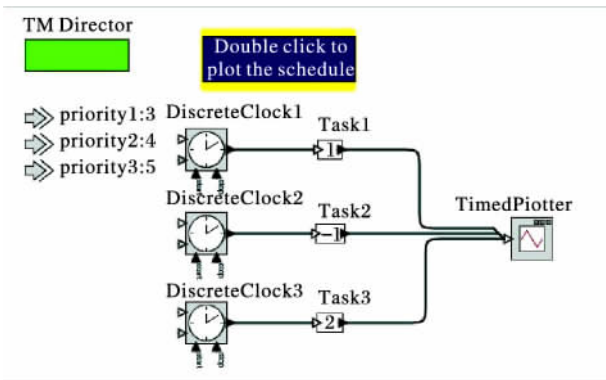


图1 任务仿真建模工程

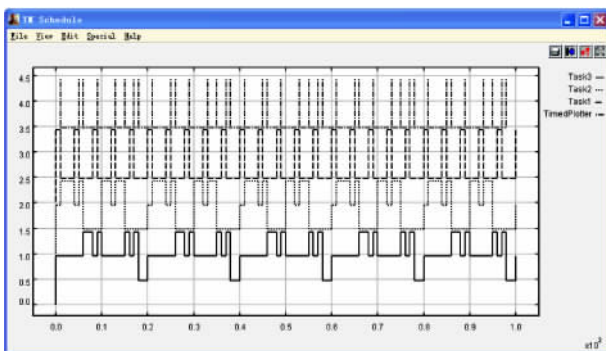


图2 单调速率调度算法多任务仿真结果

由图 2 可以看出,任务 Task3 在 $20i + 10$ ($i = 0, 1, \dots$) 时刻处由于 Task2 的受到了较大的延迟,此时若能提高任务 Task3 的优先级,连续执行任务 Task3,将大大缩短任务 Task3 的执行时间,从而保证在该任务在相对截止期内完成。由此建立优先级周期性互换调度算法工程,针对 Task2 和 Task3 进行优先级周期性互换任务调度,以缩短 Task3 的执行时间。在此用来产生周期信号以实现优先级的周期性互换;图 3 是周期性优先级转换的时序图。

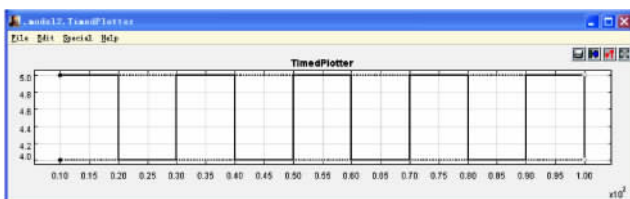


图3 周期性优先级转换时序图

采用了优先级周期性互换调度算法的仿真结果如图 4 所示,任务 Task3 在时刻 $20i + 10$ ($i = 0, 1, \dots$) 继续执行,直到剩余执行任务完成。任务 Task1 的优先级仍旧保持最高优先级不变;任务 Task2 的实际最大执行时间仍小于最大相对截止时间 8 的要求;而任务 Task3 的最大相对截止时间则缩短到 12。由此可见,采用优先级周期性互换调度算法后,较好地改

善了任务 Task3 的延迟时间,而给任务 Task2 带来的延迟仍然保持在相对截止时间内,能够满足系统控制要求。

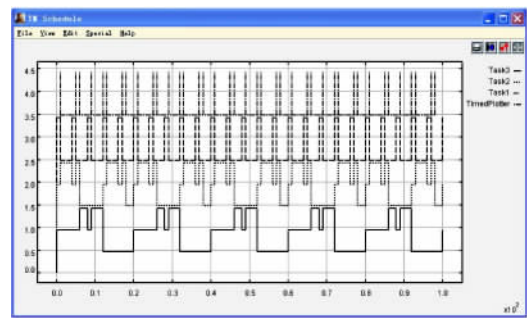


图4 对任务级别采用优先级周期性互换调度算法的仿真结果

本文同时也对 1.1 节中其他情况下使用该算法的多任务调度进行了仿真实验研究。综合仿真实验结果可知,只要保证所选取的轮换周期合理,采用优先级周期性互换任务调度算法均能在保证较高优先级任务运行时间的前提下,明显改善较低优先级任务的延迟时间。目前,关于该算法的研究和实验主要针对的是多任务中的某两个任务之间的优先级进行周期性轮换而其他任务优先级保持不变而展开的,仿真实验达到了预期效果。以此为基础,将进一步展开优先级周期性互换算法在多个任务中的研究。

2 算法在实时多任务控制系统中应用仿真

为了更好地研究该算法在实时多任务控制中的效果,本文对优先级周期性互换任务调度算法在控制中的应用展开了研究。目前,实时控制中最常用的控制策略是 PID 控制。PID 控制器具有控制方式简单、参数少且易于整定的优点。为了验证优先级周期性互换任务调度算法的实用性,对一个包含 3 个独立 PID 控制任务的多任务实时系统进行仿真。其中被控对象的数学模型来自于实际生产过程中被控对象典型的经验模型, PID 参数采用经验值。

在该实时系统中有 3 个独立任务,分别描述如下。任务分别为: $\tau_1 = (T_1, \Phi_1, C_1, P_1, E_1)$, $\tau_2 = (T_2, \Phi_2, C_2, P_2, E_2)$, $\tau_3 = (T_3, \Phi_3, C_3, P_3, E_3)$ 。其中:

任务 1 $\tau_1 = (0.04, 0, 0.01, 2, 0.02)$, PID 参数分别设定为: $K_p = 100$, $K_i = 25$, $K_d = 10$, 被控对象传递函数为:

$$\frac{1}{5s^2 + 2s + 10} \quad (3)$$

任务 2 $\tau_2 = (0.06, 0, 0.02, 3, 0.04)$, PID 参数分别设定为: $K_p = 40$, $K_i = 5$, $K_d = 6.5$, 被控对象传递函数为:

$$\frac{1}{2s^2 + 0.1s + 1} \quad (4)$$

任务 3 $\tau_3 = (0.12, 0, 0.03, 5, 0.07)$, PID 参数分别设定为: $K_p = 20$, $K_i = 0.8$, $K_d = 1$, 被控对象传递函数为:

$$\frac{1}{3s^2 + 2s + 1} \quad (5)$$

针对上述模型在 Ptolemy II 中建立工程如图 5 所示。其中: Const 是系统的给定值; AddSubtract 是进行输出与输入比较; Sampler 是采样器; ZeroOrderHold 是零阶保持器; Plant 是被控对象所对应的传递函数。TM_CPU_PID 组件是模拟仿真 CPU 的运行环境。其内部构造如图 6 所示。

采用单调速率调度算法对该实时系统进行仿真的任务运

行结果如图7所示。

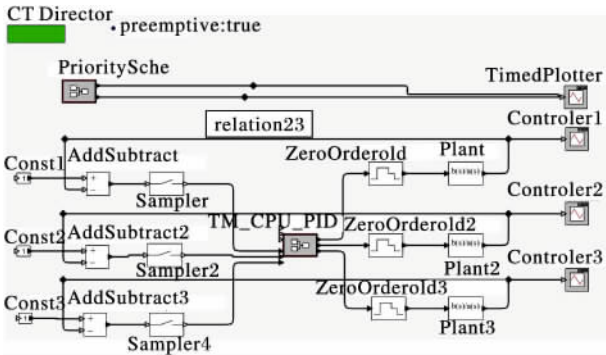


图5 某实时控制系统仿真

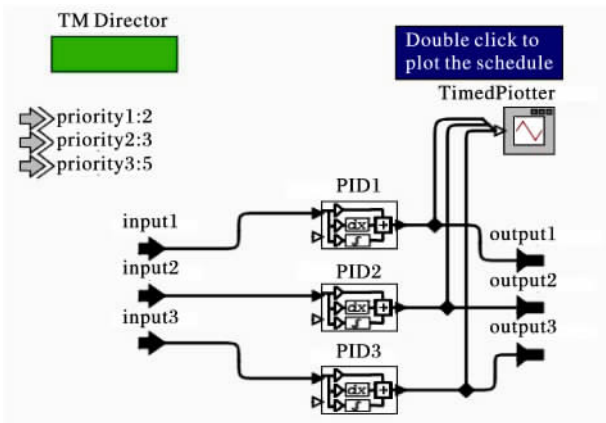


图6 TM_CPU_PID 组件

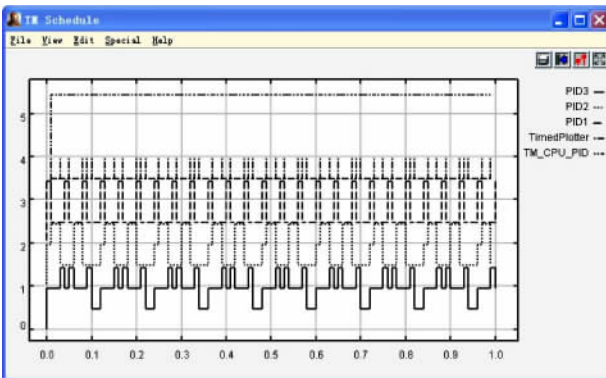


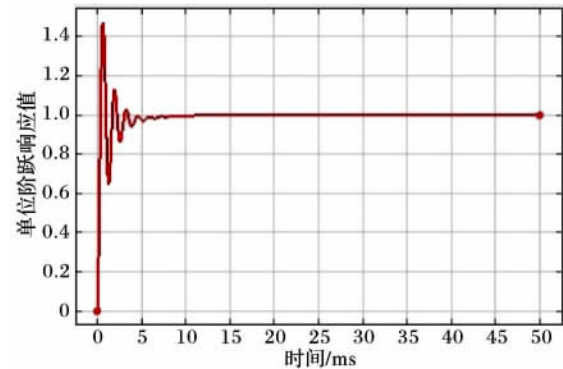
图7 多任务控制系统单调速率调度算法仿真结果

在任务调度仿真图中,分别用不同的线型表示不同PID的Controller的调度仿真。实际运行后的控制相应输出曲线如图8所示。可以看出,PID3由于优先级低,在使用单调速率调度算法的过程中,该任务的执行延迟时间长,导致任务3得不到有效控制,尽管任务1和任务2达到了较好的控制效果,但由于任务3不能提供有效控制,所以系统的整体控制效果达不到预期要求。

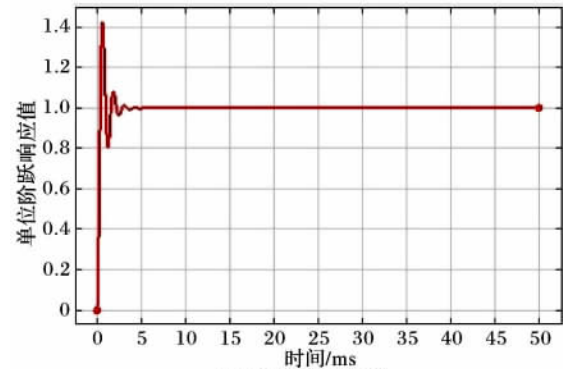
采用优先级周期性互换任务调度算法的调度仿真如图9所示。从图9中可以看到,与图7相比较,在一个执行周期内,PID3任务被延迟的次数由2次减为1次,而任务执行的延迟时间减少为前者的1/3。而由图10所示的Controller3波形图看到该系统已经趋于稳定、可控,并且同时任务1和任务2的控制效果仍然保持理想状态。

综上所述,与采用单调速率调度算法任务调度算法相比较,采用优先级周期性互换调度算法可以在保证较高级别任务的执行时间要求的前提下使得低优先级任务更及时地得到

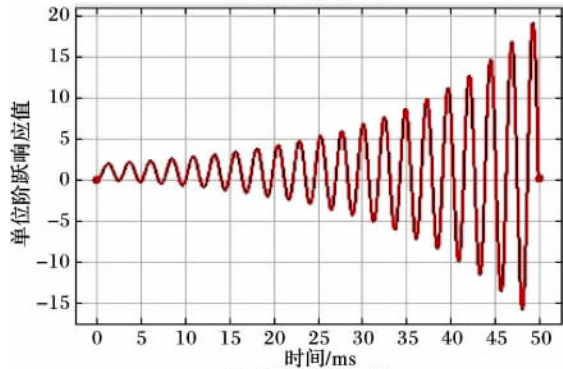
处理,有效缩短其执行延迟时间,从而实现低优先级任务的可控性,明显改善多任务实时系统的整体控制性能。



(a) 任务1输出曲线



(b) 任务2输出曲线



(c) 任务3输出曲线

图8 采用单调速率调度算法的PID响应曲线

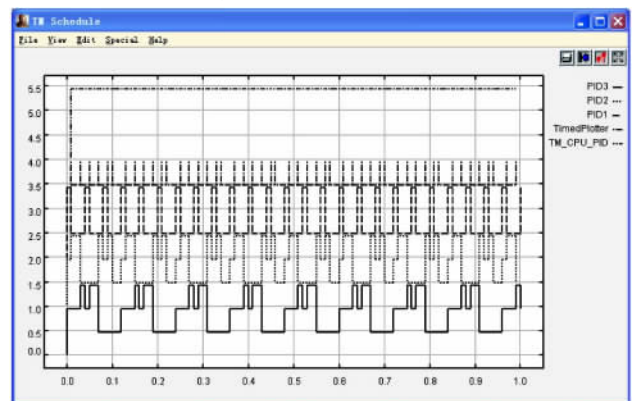


图9 在多任务PID控制系统内的优先级周期性互换任务调度仿真

3 结语

本文提出了一种优先级周期性互换的多任务调度算法。该算法本质上属于静态优先级调度算法,它将系统时间平均

分成等长时间片,预先定义好时间片长度及多个任务的优先级别,以该时间片长度为周期轮换其中两个较高优先级任务的优先级和较低优先级任务的优先级,从而使较低优先级的任务获得更多的执行机会。系统仿真实验结果表明,与单速率调度算法任务调度算法相比较,该算法能够在保证较高优先级任务完成时限的要求下有效地减少低优先级任务执行的延迟时间。该方法既具备静态优先级任务调度算法的系统开销小的优势,又兼顾了动态优先级任务调度的灵活性,能够在保证高优先级任务截止时间的条件下提高较低优先级任务的执行效率,从而保证了整个系统控制的有效性,该方法为提高实时多任务控制系统的整体控制性能提供了一种较好的解决方案。进一步的研究包括两方面:满足条件的任务多组同时进行周期性互换的效果;优先级周期性互换算法在实际工业实时控制过程中的应用。

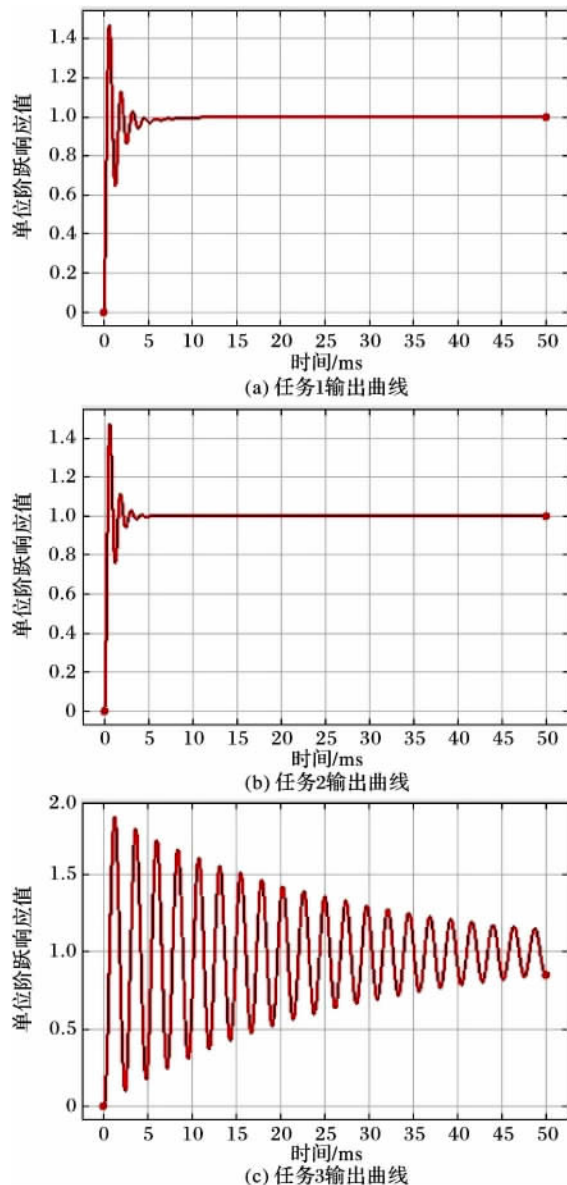


图10 采用优先级周期性互换调度算法的PID响应曲线

参考文献:

- [1] KRISHNA C M, SHIN K G. Real-time systems [M]. New York: McGraw-Hill, 2001
- [2] LIU C L, LAYLAND J W. Scheduling algorithms for multiprogram-

ming in a hard-real-time environment [J]. Journal of the ACM, 1973, 20(1): 46-61.

- [3] TINDELL K W, BURNS A. Fixed priority scheduling of hard real-time systems [J]. The Computer Journal, 1994, 37(8): 691-697.
- [4] SPURI M, BUTTAZZO G. Scheduling aperiodic tasks in dynamic priority systems [J]. Real-Time Systems, 1996, 10(2): 179-210.
- [5] KREUZINGER J, SCHULZ A, PFEFFER M, et al. Real-time scheduling on multithreaded processors [C]// Proceedings of the 7th International Conference on Real-Time Computing Systems and Applications. Washington, DC: IEEE Computer Society, 2000: 155-159.
- [6] BRANDENBURG B B, ANDERSON J H. On the implementation of global real-time schedulers [C]// RTSS09: Proceedings of the 2009 30th IEEE Real-Time Systems Symposium. Washington, DC: IEEE Computer Society, 2009: 214-224.
- [7] BARUAH S K, COHEN N K, PLAXTON C G, et al. Proportionate progress: a notion of fairness in resource allocation [J]. Algorithmica, 1996, 15(6): 600-625.
- [8] LABROSSE J J. $\mu\text{C}/\text{OS-II}$: the real-time kernel [M]. SHAO B, translation. Beijing: China Electric Power Press, 2001. (LABROSSE J J. $\mu\text{C}/\text{OS-II}$ ——源码公开的实时嵌入式操作系统[M]. 邵贝贝,译. 北京: 中国电力出版社, 2001.)
- [9] DAVIS J, II, HYLANDS C, JANNECK J, et al. Overview of the Ptolemy project [EB/OL]. [2013-06-25]. http://users.ece.utexas.edu/~gerstl/ee382v-ics_f09/soc/articles/overview_ptolemy.pdf.
- [10] ZHAO Y, LIU J, LEE E A. A programming model for time synchronized distributed real-time systems [C]// RTAS07: Proceedings of the 13th IEEE Real-Time and Embedded Technology and Applications Symposium. Washington, DC: IEEE Computer Society, 2007: 259-268.
- [11] LEE E A, ZHENG H. Leveraging synchronous language principles for heterogeneous modeling and design of embedded systems [C]// EMSOFT07: Proceedings of the 7th ACM and IEEE International Conference on Embedded Software. New York: ACM Press, 2007: 114-123.
- [12] ZOU J, MATIC S, LEE E A, et al. Execution strategies for PTIDES, a programming model for distributed embedded systems [C]// Proceedings of 15th IEEE Real-Time and Embedded Technology and Applications Symposium. Washington, DC: IEEE Computer Society, 2009: 77-86.
- [13] ZHAO M. Research on key problem of real-time scheduling with constraint [D]. Shenyang: Northeastern University, 2010. (赵明. 具备约束的实时调度关键问题的研究[D]. 沈阳: 东北大学, 2010.)
- [14] HE Y, QIN X, CHEN J, et al. Code-generation for autonomous vehicles based on Ptolemy II [J]. Journal of System Simulation, 2012, 24(2): 332-338. (何岳, 秦兴国, 陈俊伟, 等. 基于Ptolemy II的自主飞行器控制系统代码生成及应用[J]. 系统仿真学报, 2012, 24(2): 332-338.)
- [15] BA W. Research on the dynamic-priority scheduling algorithms of tasks in real-time systems [D]. Dalian: Dalian University of Technology, 2010. (巴巍. 实时系统动态优先级任务调度算法的研究[D]. 大连: 大连理工大学, 2010.)
- [16] ZHANG Y. Principle and application for real-time control system software design [M]. Beijing: National Defense Industry Press, 1998. (张云生. 自动控制实时控制系统软件设计原理及应用[J]. 北京: 国防工业出版社, 1998.)