

基于 DVS 技术的性能感知反馈调度算法^①

赵冬磊, 杨惠珍, 翟楠楠

(西北工业大学 航海学院, 西安 710072)

摘要: 传统 DVS 算法在能量管理方面没有考虑实际系统性能的需求, 这在一定程度上限制了其节能效果. 针对这一问题, 提出一种基于 DVS 技术的性能感知反馈调度算法. 在反馈调度器中, 分别采用 DVS 技术和模糊控制技术设计 CPU 电压调节模块和控制任务周期调节模块, 实现对系统 CPU 速率和控制任务采样周期的动态调节. 通过与基于固定采样周期的 DVS 反馈调度算法进行对比, 结果表明该算法在保证系统控制性能的同时进一步降低了系统能耗.

关键词: 性能感知; 反馈调度; 动态电压调节; 模糊控制; 嵌入式系统

DVS-Based Performance-Aware Feedback Scheduling Algorithm

ZHAO Dong-Lei, YANG Hui-Zhen, ZHAI Nan-Nan

(School of Marine, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Traditional DVS algorithms have inherent limitation on their capability in energy management because the system performance is not considered. A DVS-based performance-aware feedback scheduling algorithm is proposed to overcome this limitation. In the proposed feedback scheduler, DVS technology and fuzzy control technology are used respectively to design the CPU voltage adjustment module and control tasks periods adjustment module. CPU speed and control tasks sampling periods are dynamically regulated by this two-module scheduler. Simulation results show that the proposed algorithm is able to ensure the system control performance while further reduce the energy consumption of the system by comparing with the DVS feedback scheduling algorithm based on fixed sampling periods.

Key words: performance-aware; feedback scheduling; Dynamic Voltage Scaling(DVS); fuzzy control; embedded systems

1 引言

随着计算机等相关技术的高速发展, 地面移动机器人、无人机、自主水下航行器等以嵌入式计算机为控制器的系统越来越多. 这些系统大多采用电池作为能源. 由于电池能量有限, 在系统设计时, 需要尽可能降低系统能耗以延长系统使用寿命^[1]. 然而, 随着工艺水平的提高, 处理器的集成度越来越高, 人们对控制系统的性能要求也越来越高, 对能量的需求也越来越大. 因此在控制系统和控制器设计中, 如何在有限的电池能量条件下获得较高的系统性能, 即在控制性能(Control Performance)与能量消耗(Energy Consumption)之间寻求一种平衡, 是系统设计者必须

面对的一个难题. 目前一种较为流行的节能技术是动态电压调节(Dynamic Voltage Scaling, DVS)技术. 该技术基于能耗与 CPU 电压的二次型关系(即), 针对不同的系统负载, 在满足任务调度实时性需求的同时, 尽可能降低处理器的电压/频率, 从而降低系统能耗^[2,3]. 许多学者将 DVS 技术与反馈调度结合, 实现了负载可变动态环境下的能量管理. 文献[4]针对负载变化的硬实时系统, 从安全性考虑, 将一个任务分为两个子任务, 对第一个子任务设计 DVS 反馈调度器, 并根据第一个子任务的执行对系统资源的影响决定第二个子任务的执行频率, 保证任务硬实时性的同时降低了系统能耗. 文献[5]通过建立任务概率负载模型, 将能耗问

^① 基金项目:陕西省自然科学基金(2013JM8026);西北工业大学基础研究基金(2011JC0242)

收稿时间:2013-07-14;收到修改稿时间:2013-09-09

题转化为约束优化问题. 利用拉格朗日乘数法解得时钟频率来控制 CPU 的运行. 文献[6]在反馈控制的框架下采用动态松弛管理技术(Dynamic Slack Management Technique), 基于 DVS 算法动态管理任务运行过程中的空闲时间(Idle Time)来降低能耗. 但这些算法在调节 CPU 电压/频率的时候仅从系统资源利用与能耗角度考虑, 没有涉及被控对象的动态性能和应用需求.

从实际情况来看, 系统在运行过程中对资源的实际需求是动态变化的, 同时这种变化又会导致系统性能的改变. 例如在过渡过程或由于干扰引起系统超调量增大时, 系统性能波动较大, 此时需要更多的资源及时调整系统偏差, 恢复系统稳定性; 而当系统处于稳态时, 系统性能较好, 对资源的需求也较小. 因此一些学者从系统性能与资源、能耗角度考虑来设计调度控制器. 文献[7]根据任务控制回路性能状况确定各任务的相对截止时间, 进而确定供电电压. 但文中各任务的相对截止时间仅在有限个值之间进行简单的切换, 对于复杂环境下的控制系统, 缺乏一定的灵活性. 文献[8]根据控制回路的绝对误差, 设计了指数型周期调节算法, 实现了对回路采样周期的连续性调节, 但文中假设系统任务的所有时态参数精确已知, 缺乏一定的实际性. 文献[9]根据当前资源的可用状况和机械臂的轨迹误差来给各个任务分配不同的资源, 提高了资源利用率和系统性能, 但没有考虑系统能耗对性能影响.

本文基于 DVS 技术, 在反馈调度框架下设计了一个双模块反馈调度器, 根据系统的资源利用情况和控制性能特性, 同时对引起系统能耗变化的 CPU 速率和控制任务采样周期进行调节, 在保证系统动态性能的同时尽可能降低系统能耗.

2 问题描述及相关技术

2.1 单处理器多任务嵌入式控制系统

单处理器多任务嵌入式控制系统如图 1 所示, 个相互独立的控制任务共享同一资源有限的嵌入式处理器. 由于任务之间的相互竞争使得任务的时态参数(如采样周期、执行时间等)是时变的, 而且这种变化会影响任务对资源的占用进而影响系统控制性能, 因此需要针对资源的利用情况和系统性能需求调整系统时态参数, 实现动态环境下的柔性 QoC(Quality of Control)管理, 从而优化系统整体控制性能.

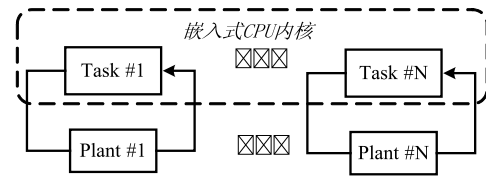


图 1 单处理器多任务嵌入式控制系统

2.2 采样周期对系统性能的影响

一般来说, 采用较大的周期可以降低能耗. 但是, 根据离散控制理论, 采样周期的增大会导致控制性能降低, 甚至造成系统不稳定. 因此, 在传统固定时间约束下直接增大采样周期来降低能耗会损害系统性能. 为解决这个问题, 本文采用弹性周期(Elastic Period)调节策略, 下面以一个具体实例来说明该策略的基本原理和可行性.

考虑一个单任务回路, 被控对象为一个直流电机:

$$G(s) = \frac{1000}{s^2 + s}$$

考察以下三种方案: 1)系统以 6ms 的采样周期运行; 2)系统初始周期为 6ms, 当运行至稳定状态时, 采样周期增大至 24ms; 3)系统以 24ms 的采样周期运行. 系统的阶跃响应性能图和任务执行情况如图 2, 图 3 所示.

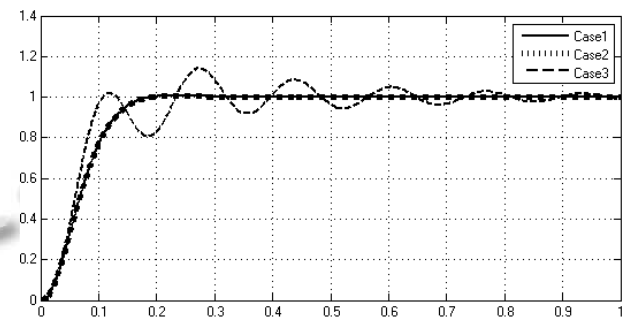


图 2 三种方案下直流电机的阶跃响应

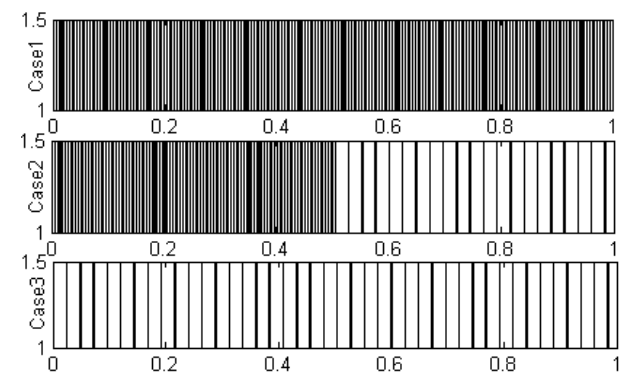


图 3 三种方案下的控制任务执行周期

从图2可以看出,虽然方案2在后期周期变大,但却获得了与方案1相同的控制性能,而且方案2中周期增大能有效降低系统能耗.方案3中,系统在过渡过程即采用较大的采样周期,导致系统出现较大的振荡,且调节时间很长.

基于上述分析,可以看出当控制回路误差较小时,可以适当地增大采样周期,此时不会对系统性能造成明显的负面影响,同时能有效地降低对系统资源(如CPU时间、能量等)的消耗.

2.3 动态电压调节技术

DVS技术的核心思想是针对不同的负载,在满足任务实时性的前提下,尽可能降低电压和时钟频率以减少处理器空闲时间,降低能耗.

考虑图1所示系统,系统支持DVS技术,且该CPU的工作电压/频率可由一个归一化的调节因子 $\alpha \in [\alpha_{\min}, 1]$ 进行连续调节,其中 $\alpha = f/f_{\max}$, f 为CPU当前工作频率(即工作速率),因此 α 也可称为CPU的相对工作速率.设 c_{i_min} 为CPU最高速率时任务*i*执行时间,则任务*i*的实际执行时间为 $c_i = \frac{c_{i_min}}{\alpha}$.

对于一个给定的DVS系统,其系统能耗可以用一个简单的二次模型近似估计^[10]:

$$E(\alpha) = \alpha^2 \tag{1}$$

由公式(1)可知,功耗与相对速率成二次方正比关系,即降低处理器速率能有效减少系统功耗.

由公式(1)可知,功耗与相对速率成二次方正比关系,即降低处理器速率能有效减少系统功耗.

3 DVS反馈调度策略

考察CPU在时刻的利用率:

$$U(k+1) = \sum_{i=1}^n \frac{c_i(k)}{h_i(k)} = \frac{1}{\alpha(k)} \sum_{i=1}^n \frac{c_{i_min}}{h_i(k)} \tag{2}$$

式中 $h_i(k)$ 为任务*i*在*k*时刻的执行周期, n 为任务个数.

设 $\omega(k) = \sum_{i=1}^n \frac{c_{i_min}}{h_i(k)}$ 则(2)式变为

$$U(k+1) = \frac{\omega(k)}{\alpha(k)} \tag{3}$$

很明显,系统输出 $U(k+1)$ 与系统输入 $\alpha(k)$ 是非线性反比关系,为了获得线性模型,设 $\beta(k) = \frac{1}{\alpha(k)}$,则

(3)式化为

$$U(k+1) = \beta(k) \cdot \omega(k) \tag{4}$$

式中 $\omega(k)$ 的值依赖于当前任务运行状况(如任务周期,任务执行个数等),因此上式是一个变参数线性模型.为了得到一个标准线性模型,我们忽略 $\omega(k)$ 项并通过

在线增益 $\frac{1}{\omega(k)}$ 对其实行动态补偿^[11,12].因此

$$U(k+1) = U(k) + \beta(k) - \beta(k-1) = U(k) + \Delta\beta(k) \tag{5}$$

对(5)式进行z变换可得

$$G_p(z) = \frac{U(z)}{\Delta\beta(z)} = \frac{1}{z-1} \tag{6}$$

采用离散PI控制算法设计电压调节控制器,其具体实现形式如下:

$$G_c(z) = \frac{\Delta\beta(z)}{\Delta U(z)} = K_p + K_I \frac{z}{z-1} \tag{7}$$

于是,根据上述方法设计的电压调节模块如图4所示.

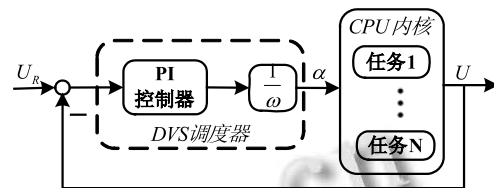


图4 电压调节模块

其工作流程描述如下: DVS反馈调度器采集当前的CPU利用率 U ,与期望值 U_R 进行比较得到控制偏差 $\Delta U(k)$,然后PI控制器根据该偏差计算出 $\Delta\beta(k)$,

再利用计算式 $\alpha = \frac{1}{\beta(k)} = \frac{1}{\Delta\beta(k) + \beta(k-1)}$ 求得 $\alpha(k)$,

经过增益补偿因子 $\frac{1}{\omega(k)}$ 对其进行调节后,对CPU电

压/频率进行调节.

4 性能感知反馈调度策略

4.1 基本思想

以控制的观点,采样周期越小越有利于控制回路快速恢复到稳态,从而抑制扰动的影响,改善控制性能.但是,当系统处于稳定状态时,过小的采样周期

却意味着计算资源的浪费,因此此时适当增大采样周期一般都不会导致控制性能明显退化^[11].

因此,本文提出的性能感知(Performance Aware)周期调节算法是:当控制误差较大时,尽量减小任务周期,为其分配更多的资源,保证控制性能;当回路误差较小时,适度地增大任务周期,减少资源消耗.这样既有利于节约能耗,又不会对控制性能造成明显的负面影响,提高了系统能效.

4.2 模糊反馈调度器设计

由于控制回路误差变化无规律且与任务周期之间没有现成的数学解析式,因此我们采用模糊控制算法设计周期调度器.

模糊反馈调度器主要由模糊化、知识库(包含数据库和规则库)、推理机制以及反模糊化四个部分组成^[11].调度控制器(即模糊控制器)的输入量为任务 i 控制回路误差 $e_i(k)=r_i - y_i$ 和误差变化 $ec_i(k)=e_i(k)-e_i(k-1)$, 输出为调节因子 $\lambda_i(k)$. 调度控制器的基本工作流程:首先由模糊化模块将系统当前输入量 $e_i(k)$ 和 $ec_i(k)$ 通过量化因子转化为量化值 E 和 EC , 然后由模糊推理机制根据此量化值在模糊控制表中查找出当前时刻的输出量化值 RF , 最后由反模糊化模块通过比例因子将输出量化值 RF 转化为最终的输出 $\lambda_i(k)$.

性能感知模糊反馈调度器基本原理如图 5 所示.反馈调度器采集各任务的动态输出并与输入信号进行比较,得到控制误差和误差变化作为模糊控制器的输入.模糊控制器计算出调节因子 $\lambda_i(k)$, 然后调度器通过下式对每个任务的周期进行调节.

$$h_i(k) = \lambda_i(k)h_i(k-1) \quad (8)$$

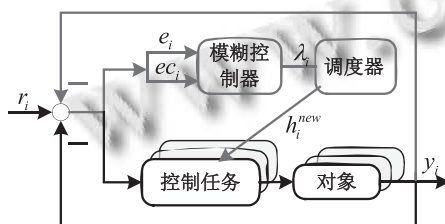


图 5 性能感知反馈调度器

为了防止系统在线调度过程中周期过大或过小导致系统失去可调度性,有必要设定任务采样周期的范围,即 $h_i \in [h_{\min}, h_{\max}]$, 其中 h_{\max} 和 h_{\min} 分别是任务周期的最大值和最小值.

由于各回路采样周期的配置仅与自身当前的控制

性能有关,即各回路采样周期的计算是独立运行的.为方便起见,在下面的叙述中,我们将省去下标 i . 模糊控制器的具体设计步骤如下:

1) 设计模糊反馈调度器结构

选择一个双输入单输出的二维模糊控制器,即两个输入量 $e(k)$, $ec(k)$ 和一个输出量 $\lambda(k)$. 取 $e(k)$, $ec(k)$ 和 $\lambda(k)$ 的基本论域分别为 $[-0.2,0.2]$, $[-0.2,0.2]$ 和 $[0.4,1.6]$.

2) 确定输入量输出量的模糊语言描述

取误差和误差变化的模糊语言变量 $E=EC=\{NB, NM, NS, ZE, PS, PM, PB\}$

取调节因子的模糊语言变量为: $RF=\{NB, NM, NS, ZE, PS, PM, PB\}$

其中, NB, NM, NS, ZE, PS, PM, PB 分别表示“负大”,“负中”,“负小”,“零”,“正小”,“正中”,“正大”.

取误差和误差变化的量化等级均为 $\{-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7\}$

调节因子的量化等级为 $\{-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7\}$

3) 建立模糊规则表

建立模糊控制规则的基本思想:当某个任务的控制回路误差较大时,应选择较小的调节因子,使采样周期尽快减小从而改善系统性能;当控制回路误差较小时应逐步增大采样周期,降低系统能耗.

例如,当误差 $e(k)$ 为负大时,说明此时控制误差较大,系统性能较差,应当快速减小采样周期.因此在 $ec(k)$ 为负(表明控制误差还有增大的趋势)时, $\lambda(k)$ 应取最小值(负大);当 $ec(k)$ 为正大时,说明控制误差已有明显下降的趋势,此时 $\lambda(k)$ 取负小避免系统出现严重振荡.分别对其他情况进行逐个分析,归纳出 49 条控制规则,如表 1 所示.

表 1 模糊规则表

RF	EC						
	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NM	NM	NM	NS	NS
NM	NB	NM	NM	ZE	ZE	PS	PM
NS	NB	NM	ZE	PM	PB	PB	PB
ZE	NB	NS	ZE	PM	ZE	NM	NM
PS	PM	PS	PS	ZE	NS	NM	NB
PM	PS	PS	ZE	ZE	NM	NB	NB
PB	ZE	NS	NM	NM	NM	NB	NB

4) 模糊推理和反模糊化

在推理机制中, 采用 Mamdani 极大-极小推理法; 在反模糊化模块, 采用重心法求取输出论域中的精确值。

不难看出, 采用这种模糊反馈调度算法, 反馈调度器在运行过程中只需通过一些简单的运算以及查询一个 14*14 维数据表即可求出采样周期调节因子, 计算速度相当快, 因而反馈调度开销非常小, 适于实时在线应用。

5 基于DVS的性能感知反馈调度策略

本文提出的基于 DVS 的性能感知反馈调度 (Performance-Aware Feedback Scheduling, PAFS) 系统如图 6 所示. 该系统的反馈调度器包含两个模块: 电压调节器和周期调节器. 电压调节器根据当前 CPU 利用率, 基于 DVS 技术调节 CPU 工作速率, 进而调节任务执行时间; 周期调节器根据任务控制回路的性能状态, 采用模糊控制技术调节任务的执行周期. 一方面, 任务执行时间越大, 意味着 CPU 速率越低, 能耗越低; 另一方面, 任务执行周期越小, 系统性能越好, 但能耗越高. 在一定资源约束下, 增大任务执行时间和减小周期是相互冲突的, 因为它们都导致 CPU 利用率增大. 从这个意义上讲, 本文提出的性能感知反馈调度策略是通过动态调节任务周期和执行时间在控制性能和能耗之间实现折中(Trade Off).

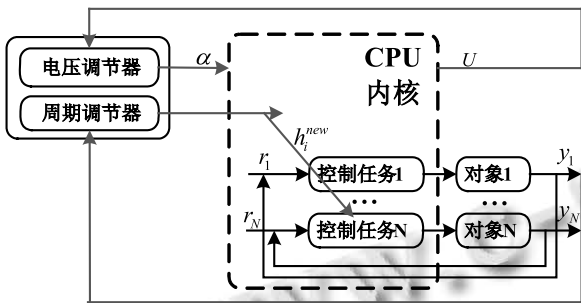


图 6 基于 DVS 的性能感知反馈调度系统

6 仿真实验

6.1 实验方案

考虑一个处理器同时控制三个倒立摆的系统, 其中倒立摆的状态空间模型采用如下形式^[13]

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

定义 $x(t) = [\theta(t), \dot{\theta}(t)]^T$, 其中 $\theta(t)$ 为倒立摆与竖直方向的夹角, $\dot{\theta}(t)$ 为角速度。

$$A = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -\frac{r}{J} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ l \end{bmatrix} \quad C = [1 \quad 0]$$

其中 g 为重力加速度, l 为倒立摆杆长, m 为摆的质量, $J=ml^2$ 为转动惯量, γ 为摩擦常数. $g=9.8m/s^2$, $[l_1 \ l_2 \ l_3] = [0.2 \ 0.35 \ 0.5]$, $m=0.5kg$, $\gamma = 0.6$. 三个任务的周期初始值 $h_0 = [6 \ 5 \ 4]ms$, 任务执行时间 $c_{i_min} = 1ms (i=1,2,3)$, $h_{min} = h_0$, $h_{max} = [28 \ 26 \ 24]ms$, $\alpha_{min} = 0.1$. 反馈调度器周期 $h_{FBS}=100ms$, 执行时间 $C_{FBS}=1ms$. 任务在一个优先级抢占的实时内核中运行, 调度器的优先级最高, 三个控制任务的优先级遵循 RM(Rate Monotonic)调度. $U_{sp}=0.85$ 以使系统获得较好的控制性能同时又不至于错过太多的截止期限. 三个倒立摆控制器采用离散 PID 算法, 利用 Matlab/True Time 工具箱进行仿真^[14]. 实验中, 任务 1 和任务 2 从 0 时刻开始, 任务 3 在 2s 时开始执行。

仿真结果与基于 DVS 技术的能量感知反馈调度 (Energy-Aware Feedback Scheduling, EAFS)^[15]进行对比. EAFS 不考虑系统性能需求, 基于固定采样周期对 CPU 速率进行调节. 设 EAFS 算法中三个任务的执行周期为 $h=[6 \ 5 \ 4]ms$, 其他参数与 PAFS 中的参数设置相同。

6.1 实验结果分析

两种调度方法的系统能耗对比如图 7 所示. 可以看出, 基于 DVS 技术的能量感知调度和性能感知调度都能根据当前系统负载需求有效地降低 CPU 能耗. 在 0~2s 内, 只有任务 1 和任务 2 运行, 系统负载较低, 此时 CPU 速率也随之降低, 其中 EAFS 的平均能耗为 $E(\alpha) = 20%$, PAFS 的平均能耗为 $E(\alpha) = 13%$; 在第 2s 时, 随着任务 3 的加入, 系统负载增加, 调度器及时增大 CPU 速率, 保证 CPU 利用率不超过设定值, 同时也使系统性能得到改善. 在 2~5s 内, EAFS 的平均能耗为 $E(\alpha) = 50%$, 而 PAFS 的平均能耗为 $E(\alpha) = 32%$. 相比于 EAFS, 能耗减少了 18%. 显然, PAFS 具有更好的节能性, 并且随着系统负载增加, 节能优势更加明显。

通过图 8 来分析产生上述结果的原因. 相比于 EAFS 采用固定任务周期, PAFS 策略下每个任务周期都有不同程度的增大. 由于任务周期增大, 所以 PAFS 能够比 EAFS 节约更多的能量. 例如在 2.5s 附近, 三个任务周期达到最大, 此时系统能耗仅为 23%, 相比于 EAFS 的 50% 降低了 27%。

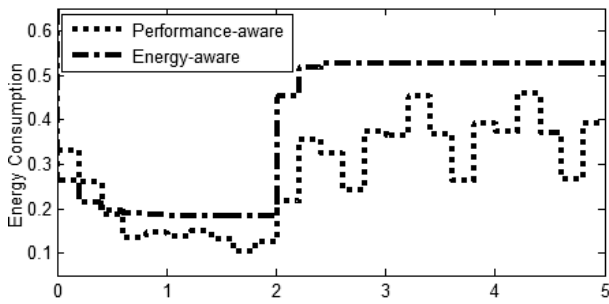


图 7 系统能耗图

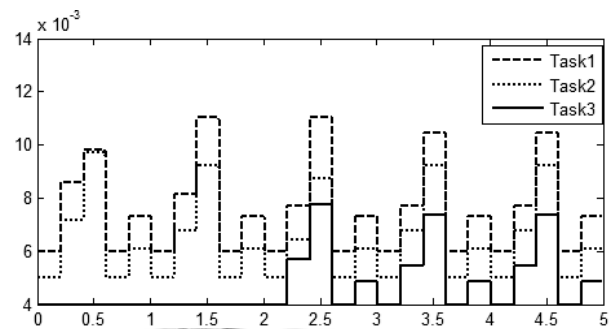


图 8 PAFS 策略下任务周期变化

另外观察 CPU 的调度情况也可以非常直观地看出 EAFS 与 PAFS 在任务周期和执行时间上的不同. 如图 9, 10 所示, 在 EAFS 调度下, 无论回路是否有误差, 各任务的采样周期均保持不变; 而在 PAFS 调度下, 各回路的采样周期都将随着实际控制误差的变化而变化. 如图 10 所示, 在 2s 时, 回路误差较大, 此时控制任务的执行很密集, 说明为了补偿回路误差, 三个任务被分配较小的采样周期和执行时间; 当系统处于稳定状态时(如 2.3~2.5s), 三个任务的周期和执行时间逐渐增大, 意味着 CPU 速率减缓, 系统能耗也随之降低.

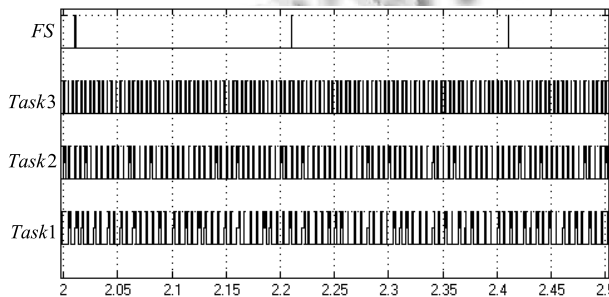


图 9 电压感知方法的 CPU 调度图

为了更直观地比较两种方案下系统整体控制性能, 定义系统总控制代价函数

$$J(t) = \sum_{i=1}^3 \int_0^t (y_i(\tau) - r_i(\tau))^2 dt \quad (9)$$

其中 r_i, y_i 分别为任务 i 的输入量和输出量. 图 11 给出了系统总控制代价变化曲线. 显然, 在 2s 之前系统负载较低, 各回路性能较好, 系统之间总控制代价相差不大; 在 2s 以后, 系统总控制代价逐渐加大, 主要原因是 PAFS 控制下, 任务周期和执行时间变大, 导致回路延迟有所增大, 因此控制性能有所下降(如图 12, 13 所示), 但整体性能依然较好.

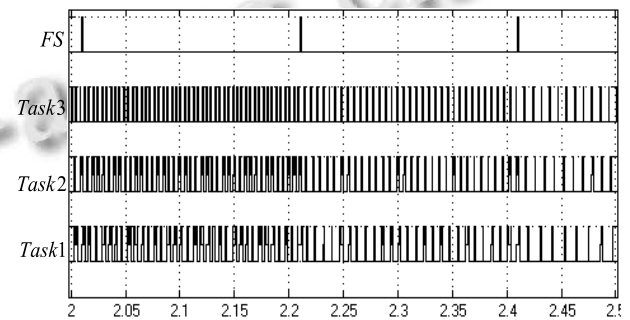


图 10 性能感知方法的控制性能效果图

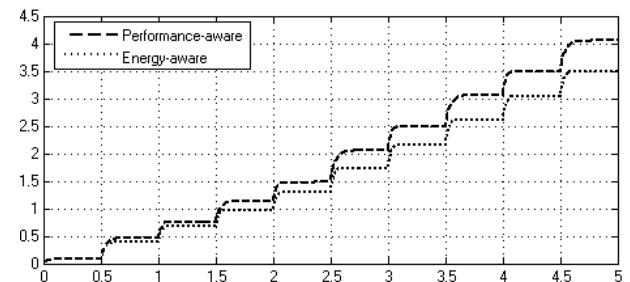


图 11 系统总控制代价

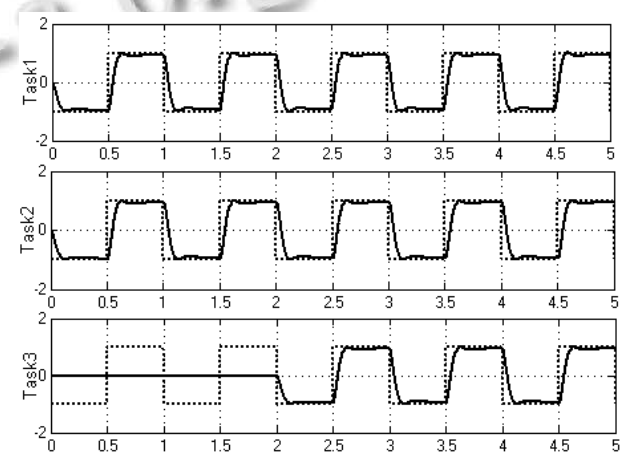


图 12 电压感知方法的控制性能效果图

7 结语

本文分析了系统性能与采样周期之间的关系, 结

合动态电压调节技术提出了一种性能感知反馈调度算法。在利用DVS技术对CPU速率进行调节的同时,根据各控制回路的实际控制性能对其采样周期进行动态调节,实现了系统负载、能耗与性能之间的权衡。最后,通过与EAFS算法比较验证了该算法在节能方面的高效性。

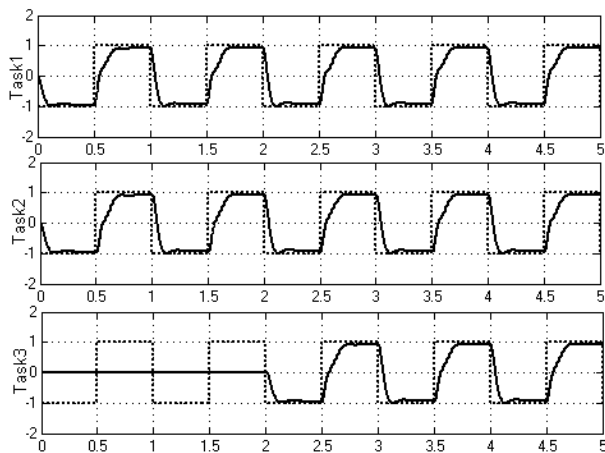


图13 性能感知方法的控制性能效果图

参考文献

- 1 Santinelli L, Marinoni M, Prosperi F, Esposito F, Franchino, Buttazzo G. Energy-aware packet and task co-scheduling for embedded systems. Proc. of the International Conference on Embedded Software. 2010. 279–288.
- 2 Li D, Wu J. Energy-aware scheduling for frame-based tasks on heterogeneous multiprocessor platforms. 2012 41st International Conference on Parallel Processing. 2012. 430–439.
- 3 Kwon K, Chae S, Woo KG. An application-level energy-efficient scheduling for dynamic voltage and frequency scaling. 2013 IEEE International Conference on Consumer Electronics. 2013. 3–6.
- 4 Ahmadian AS, Hosseingholi M, Ejlali A. Discrete feedback-based dynamic voltage scaling for safety critical real-time systems. Scientia Iranica, 2013, 20(3): 647–656.
- 5 Yang M, Wen Y, Cai J, Heng CF. Energy minimization via dynamic voltage scaling for real-time video encoding on mobile devices. 2012 IEEE International Conference on Communications (ICC). 2012. 2026–2031.
- 6 Baskaran S, Thambidurai P. Power aware scheduling for resource constrained distributed real-time systems. International Journal on Computer Science and Engineering, 2010, 2(5): 1746–1753.
- 7 Piao XF, Huang JH. The cost-aware dynamic voltage scaling approach for networked control systems. Journal of Computational Information Systems, 2012, 8(7): 3059–3069.
- 8 Xia F, Liu LP, Ma LH, Sun YX, Dong JX. Performance aware power management in embedded controllers with multiple-voltage processors. Information Technology Journal, 2008, 7(6): 1–9.
- 9 Sename O, Simon D, Gaid MB. A LPV approach to control and real-time scheduling codesign: application to a robot-arm control. Proc. of the 47th IEEE Conference on Decision and Control. 2008. 4891–4897.
- 10 Sinha A, Chandrakasan AP. Energy efficient real-time scheduling. Proc. of the International Conference on Computer-Aided Design, 2001, 458–463.
- 11 夏锋.资源受限实时控制系统反馈调度[博士学位论文].杭州:浙江大学,2006.
- 12 Simon D, Robert D, Sename O. Robust control/scheduling co-design: application to robot control. Proc. of the 11th IEEE Real Time and Embedded Technology and Applications Symposium. 2005. 1–10.
- 13 Zhang FM, Szwaykowski K, Wolf W, Mooney V. Task scheduling for control oriented requirements for cyberphysical systems. Proc. of the 29th IEEE Real-Time Systems Symposium. 2008. 47–56.
- 14 Eker J, Cervin A. A Matlab toolbox for real-time and control systems co-design. Proc. of the 6th International Conference on Real-Time Computing Systems and Applications. 1999. 320–327.
- 15 Xia F, Tian YC, Sun YX, Dong JX. Control-theoretic dynamic voltage scaling for embedded controllers. Computers & Digital Techniques, 2008, 2(5): 377–385.