

DOI: 10.7652/xjtuxb201304019

一种采用神经网络 PID 控制的自适应资源分配方法

林军^{1,2,3}, 倪宏^{2,3}, 孙鹏^{2,3}, 张辉^{2,3}

(1. 中国科学院大学, 100049, 北京; 2. 中国科学院声学研究所, 100190, 北京;
3. 中国科学院国家网络新媒体工程技术研究中心, 100190, 北京)

摘要: 针对有限资源开放式系统多任务资源分配问题,提出了采用神经网络 PID 控制的自适应资源分配方法。该方法以求解任务间公平性服务质量水平(QoS)为目标,使用当前 QoS 与系统平均 QoS 的偏差作为控制器输入,通过多输入多输出神经网络 PID 控制,在线调整了资源分配。该方法不需要精确预知任务服务质量函数形式,而是通过自适应微调 PID 参数,快速有效地获得了近似公平的资源分配。实验结果表明,该方法能保证 QoS 快速收敛,与现有方法相比,资源调整次数减少了近 10 次,扩大了稳定范围。

关键词: 资源分配;神经网络;自适应;公平性;服务质量

中图分类号: TP37 **文献标志码:** A **文章编号:** 0253-987X(2013)04-0112-06

Adaptive Resource Allocation Based on Neural Network PID Control

LIN Jun^{1,2,3}, NI Hong^{2,3}, SUN Peng^{2,3}, ZHANG Hui^{2,3}

(1. University of Chinese Academy of Sciences, Beijing 100049, China; 2. Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; 3. National Network New Media Engineering Research Center, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: An adaptive resource allocation based on neural network proportion integration differentiation control is proposed to deal with the multi-tasking resource allocations in limited open resource systems, whose target focuses on searching the fair quality of service levels among the tasks. The controller takes errors between the current task Qos levels and the system average Qos level as inputs, and adjusts the resource allocation by multiple-input multiple-output neural network PID control. It is unnecessary to take the precise Qos functions as a prior, instead, the fair resource allocation can be obtained quickly and efficiently by fine-tuning PID parameters. The result shows that the proposed method enables to decrease about ten resource adjustments and widen stability range.

Keywords: resource allocation; neural network; adaptive; fairness; quality of service

嵌入式系统,资源的分配策略决定了应用可提供的服务质量。传统的资源分配策略基于预知的服务质量水平(QoS),典型的如 Rajkumar 等提出的基于任务的资源分配模型^[1](Q-RAM),在满足应用最

小资源需求约束的基础上,分配多种资源的最大化系统总效用,将其归纳为非线性或整数规划问题,并提出离线的解决方案。文献[2]利用凸优化理论和定价机制来求解多资源分配问题,但在开放式系统

收稿日期: 2012-06-25。 作者简介: 林军(1986—),男,博士生;倪宏(通信作者),男,研究员。 基金项目: 国家“863 计划”重点资助项目(2011AA01A102);国家科技支撑计划资助项目(2011BAH16B03);中国科学院战略性先导科技专项资助项目(XDA06010302)。

网络出版时间: 2013-03-11

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20130311.1749.001.html>

中,应用的加入、退出以及相关的运行参数都是不能预测和获取的,需要合适的在线资源分配策略来保证多应用的服务质量。

文献[3-4]引进弹簧任务模型,通过扩充和压缩任务的执行时间来进行CPU资源自适应控制。文献[5]中基于软实时任务作业完成时刻与作业截止期的偏差,使用前馈控制来改变预留的CPU带宽,以获取更好的服务质量。文献[6]中的控制理论用于数据库的负载均衡管理,同时在分配资源收益及改变资源分配花销中取舍平衡。文献[7]提出了一种基于PID控制的资源分配策略,并通过平衡点附近的线性化处理来证明控制闭环的稳定性。采用现代控制方法在求解非线性控制问题时,需要对控制对象进行系统辨识或对预知具体对象函数进行线性化处理。在开放的动态系统中,无法预知应用的服务质量函数,因此较难求得精确的控制反馈。常规的PID控制对非线性、时变系统不能达到理想的控制效果,其控制参数不能适应应用的特征变化,因此性能较差。神经网络PID控制^[8]在常规的PID控制基础上利用神经网络参数自适应技术,微调控制参数,可满足复杂系统和高指标的控制要求。

本文基于任务间的加权QoS公平性指标,利用神经网络PID控制器,实现有效的在线资源分配策略,其中控制器与被控对象一起作为广义的神经网络,不需要辨识任务对象。

1 任务集及资源分配模型

开放系统中包含 n 个实时任务,用实时任务集 $\Gamma=\{\tau_1, \tau_2, \dots, \tau_n\}$ 表示,每个任务 τ_i 赋予权重系数 $w_i (\in R_+)$ 表示任务的相对重要性,任务越重要,权重系数越大。权重系数的大小可以通过多种方式确定,如可以直接使用任务的优先级,或者由用户在系统中根据自己的偏好设置。当存在多个权重因素时,权重向量可以使用层次分析法(AHP)^[9]等方法计算获得。任务间存在资源约束,其共享资源集 $R=\{R_1, R_2, \dots, R_k\}$,每个资源都存在有限的容量上限 $R_{j\max}, j \in [1, k]$ 。以CPU资源利用率为例,对于单调速率调度算法, $R_{\max}=n(2^{1/n}-1)$,对于最早截止时间优先算法, $R_{\max}=1$ 。

假设 $r_{ij} (\in R_+)$ 为分配给任务 τ_i 的资源 R_j 的数量,任务 τ_i 获得的服务质量水平记为 S_i ,分配更多的资源数量将增大任务的 S_i ,资源 $\{r_{i1}, r_{i2}, \dots, r_{ik}\}$ 与 S_i 的关系可以通过单调递增的服务质量函数表述。 $S_i \in [S_{i\min}, S_{i\max}]$, $S_{i\min}$ 表示 τ_i 可接受的最坏

服务质量, $S_{i\max}$ 表示 τ_i 可获得的最佳服务质量,其对应资源 R_j 的需求为 $r_{ij\min}$ 和 $r_{ij\max}$ 。为了体现不同任务间的QoS关系,将每个任务 τ_i 的归一化QoS表示为

$$Q_i = \frac{S_i - S_{i\min}}{S_{i\max} - S_{i\min}}, Q_i \in [0, 1] \quad (1)$$

当 $Q_i=0$ 时, τ_i 获得 $S_{i\min}$,当 $Q_i=1$ 时,表示 τ_i 获得 $S_{i\max}$ 。本文只考虑多任务单资源的分配,单资源的分配在实时系统中具有很大的实际意义,如CPU的资源分配调度策略是保证任务实时性能的基础。假设该单资源为 $R_1, r_{i1}, r_{i1\min}, r_{i1\max}$ 分别简称为 $r_i, r_{i\min}, r_{i\max}$ 。使用服务质量函数描述单个资源与归一化服务质量的关系 $\varphi_i: [r_{i\min}, r_{i\max}] \rightarrow [0, 1]$,其中 $\varphi_i(r_i)$ 表示分配资源为 r_i 时 τ_i 获得的QoS。 φ_i 是一个单调递增的连续函数,当 $r_{i\min}=0$ 时, $\varphi_i(0)=0$ 。将系统任务QoS的公平性定义如下。

定义 假设分配给 $\tau_i (i \in \{1, 2, \dots, n\})$ 的资源为 r_i , Q_i, w_i 分别表示 τ_i 归一化的QoS和权重。如果下列公式成立,可获得公平的资源分配为

$$\frac{Q_1}{w_1} = \frac{Q_2}{w_2} = \dots = \frac{Q_n}{w_n} = \bar{Q} \quad (2)$$

式中: $\bar{Q}$ 为公平分配的QoS。由于 φ_i 是一个单调递增的连续函数,所以式(2)有解,因此本文的资源分配问题可以归结为满足约束 $\sum_{i=1}^n r_i \leq R_{\max}$,以及求解最大 $\bar{Q}$ 值的问题。

2 自适应资源分配的方法

2.1 系统控制架构

如图1所示,QoS监控模块收集和评估任务完成获得QoS,并将数据反馈给资源管理器;调度执行器保证作业对资源的使用,不同的资源涉及到不同的调度算法。以CPU资源实时调度为例,常见的调度算法有EDF算法、RM算法和LSF算法,不同算法涉及的资源上限也会存在差别。资源管理器负责管理资源池,分成自适应反馈控制器和资源分配模块两部分。自适应反馈控制器在离散的控制激活点依据任务QoS监控模块反馈过来的数据,采取合适的机制提供资源划分参考。资源分配模块负责检验参考资源数据的有效性,并划分实际资源给各个任务。

2.2 采用神经网络控制设计

在图1系统控制架构的基础上,设计具体控制规则来保证任务间QoS的公平性。为简单起见,考

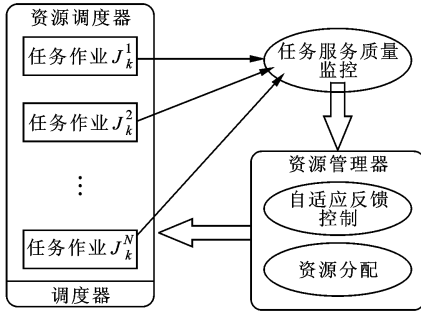


图1 系统控制架构图

考虑 $\omega_i=1$, 且不考虑输出测量干扰和资源分配干扰 (见图2)。 τ_i 获得的 r_i 在控制期间动态变化, $r_i(k)$ 表示在时间段 $[t_k, t_{k+1})$ 的值, t_k 表示第 k 次系统资源的分配时刻, $Q_i(k)$ 为任务 τ_i 作业在此次资源更新后获得的服务质量水平。时刻 t_0 控制器先给每个任务分配资源 $r_i(0)$, 当自适应控制在时刻 t_{k+1} 激活时, 分配的资源从 $r_i(k)$ 更新为 $r_i(k+1)$ 。

2.2.1 神经网络PID控制器的实现 图2中除神经网络控制器(NNC)以外, 其他部分都可以认为是广义的多输入多输出被控对象 A , 因此采用多输入多输出的神经网络PID控制器。假设系统中存在 n 个任务, 认为 A 是具有 $n-1$ 个输入 $\{u_1, u_2, \dots, u_{n-1}\}$ 和 $n-1$ 个输出 $\{e_1, e_2, \dots, e_{n-1}\}$ 的耦合系统, 因此 NNC 由 $n-1$ 个PID子神经网络组成, 子网络输入层至隐层各自独立, 隐层至输出层之间不独立, 互相有连接权。PID子神经网络是3层前向神经网络, 具有输入层节点2个, 隐层节点3个, 输出层节点1个, 因此整体网络记为 $N_{2(n-1), 3(n-1), n-1}$ 。隐层节点的3个神经元分别为比例 P 、积分 I 、微分 D 单元, 是动态前馈网络, 采用网络前向技术实现控制规律, 采用网络的反向算法实现参数自适应调整, 网络输入层的输入、输出相等。第 s 个子网络的输入集合为

$$R_s(k) = [d_1(k), d_2(k)] = [e(k), e_i(k)]$$

式中: $d_1(k)$ 、 $d_2(k)$ 为第1、第2个输入子网络; $e(k)$ 为系统的设计误差; $e_i(k)$ 为系统的控制误差。

采用当前系统的平均 QoS 作为参考输出, 即

$$\bar{Q}(k) = \frac{1}{n} \sum_{i=1}^n Q_i(k) \quad (3)$$

针对 τ_i 的反馈设计, 用当前任务获取的 QoS 与系统平均水平的差值作为控制误差, 即

$$e_i(k) = \bar{Q}(k) - Q_i(k) \quad (4)$$

当 $e(k)=0$ 时, 为保证系统稳定时获得公平的资源分配, 系统控制误差应满足

$$e_i(k) = e(k) = 0 \quad (5)$$

PID神经网络的第 s 个子网络的第 i 个隐层节点的输入为

$$x_{si}(k) = \sum_{j=1}^2 w'_{sij} R_{sj}(k), i = 1, 2, 3 \quad (6)$$

式中: w'_{sij} 为第 s 个子网络连接第 j 个输入层节点至第 i 个隐层节点的权值; $R_{sj}(k)$ 为 $R_s(k)$ 的第 j 个变量。第 s 个子网络的隐层比例、积分、微分神经元输出分别为

$$o_{s1}(k) = \begin{cases} x_{s1}(k), & -1 \leq x_{s1}(k) \leq 1 \\ 1, & x_{s1}(k) > 1 \\ -1, & x_{s1}(k) < -1 \end{cases} \quad (7)$$

$$o_{s2}(k) = \begin{cases} o_{s2}(k-1) + x_{s2}(k), & -1 \leq o_{s2}(k) \leq 1 \\ 1, & o_{s2}(k) > 1 \\ -1, & o_{s2}(k) < -1 \end{cases} \quad (8)$$

$$o_{s3}(k) = \begin{cases} x_{s3}(k) - x_{s3}(k-1), & -1 \leq o_{s3}(k) \leq 1 \\ 1, & o_{s3}(k) > 1 \\ -1, & o_{s3}(k) < -1 \end{cases} \quad (9)$$

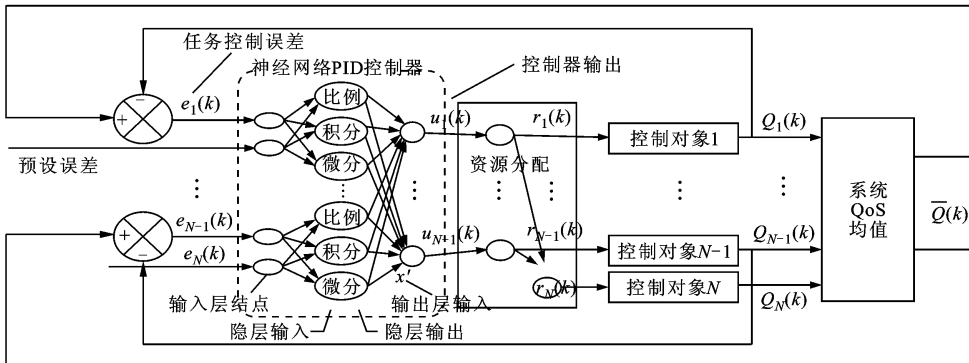


图2 神经网络自适应闭环控制图

第 $h \in [1, n-1]$ 个输出层输入 $x'_h(k)$ 为隐层各节点输出的加权和。输出层神经元使用线性模型,第 h 个输出层节点的输出为

$$u_h(k) = x'_h(k) = \sum_{s=1}^{n-1} \sum_{i=1}^3 w_{shi}^2 o_{si}(k) \quad (10)$$

式中: w_{shi}^2 为第 s 个子网络连接第 i 个隐层节点至第 h 个输出层节点的权值。控制器的学习算法采用反向传播(BP)在线训练,以保证准则函数在要求的限度内。在线训练的准则函数为

$$J = E(k) = \frac{1}{2} \sum_{i=1}^{n-1} [d_1(k) - d_2(k)]^2 = \frac{1}{2} \sum_{i=1}^{n-1} [\bar{Q}(k) - Q_i(k)]^2 \leq \epsilon \quad (11)$$

式中: $E(k)$ 为第 k 次训练时网络的准则函数; ϵ 为控制精度。实际上当满足式(11)时, $\tau_i, i \in [1, n-1]$ 获得的 QoS 接近系统均值。设

$$|e_i(k)| = |\bar{Q}(k) - Q_i(k)| \leq \epsilon_i, i \in [1, n-1]$$

式中: ϵ_i 为控制误差精度。此时第 n 个任务的误差

$$|e_N(k)| = |\bar{Q}(k) - Q_N(k)| = \left| \sum_{i=1}^{N-1} e_i(k) \right| \leq \sum_{i=1}^{N-1} |e_i(k)| \leq \epsilon_N \quad (12)$$

式中: ϵ_N 为足够小的正数。 ϵ_N 使得第 n 个任务的误差满足式(12),此时第 n 个 τ_n 的 QoS 也接近系统均值,因此反馈控制设计不考虑输入第 n 个任务的误差。

神经网络按梯度法调节控制器权值,设置调整步长 η 后,第 k 次训练后的权值集合为

$$W(k+1) = W(k) - \eta \frac{\partial J}{\partial W} \quad (13)$$

其中隐层至输出层的梯度

$$\begin{aligned} \frac{\partial J}{\partial w_{shi}^2} &= \sum_{p=1}^{n-1} \frac{\partial J}{\partial e_p(k)} \frac{\partial e_p(k)}{\partial x'_h(k)} \frac{\partial x'_h(k)}{\partial w_{shi}^2} = o_{si}(k) \sum_{p=1}^{n-1} \delta_{ph}(k) \\ \delta_{ph}(k) &= e_p(k) \operatorname{sgn} \left(\frac{e_p(k) - e_p(k-1)}{u_h(k-1) - u_h(k-2)} \right) \end{aligned} \quad (14)$$

输入层至输出层的梯度

$$\begin{aligned} \frac{\partial J}{\partial w_{sj}^1} &= R_{sj}(k) \sum_{h=1}^{n-1} \sum_{p=1}^{n-1} w_{shi}^2 \cdot \\ &\operatorname{sgn} \left(\frac{o_{si}(k) - o_{si}(k-1)}{x_{si}(k-1) - x_{si}(k-2)} \right) \delta_{ph}(k) \end{aligned} \quad (15)$$

文献[8]依据利亚普洛夫稳定性原理,给出了多输入多输出PID神经网络控制的稳定条件如下。

定理 当PID神经网络的学习步长满足 $0 < \eta < \epsilon^{-2}$ ($\epsilon = -\frac{1}{2} J^{-1/2} \frac{\partial J}{\partial w}$) 时,系统渐近稳定。定理证明

参见文献[8],在实际应用中,可以先选择较小的步长,然后根据定理进行步长的调整。

2.2.2 资源分配模块 资源分配模块负责检验从神经网络PID控制器输出的资源数据的有效性,在模型建立时,指出给 τ_i 分配的资源必须满足 $r_i \in [r_{\min}, r_{\max}]$,因此可判断控制返回的输出数据 $\{u_1, u_2, \dots, u_{n-1}\}$ 是否都满足各自的取值区间。如果 $u_i < r_{\min}$,取 $r_i = r_{\min}$,以保证任务的最坏 QoS。如果 $u_i > r_{\max}$,取 $r_i = r_{\max}$,可不浪费资源且不破坏 τ_i, φ_i 的单调性。其他情况则考虑 $r_i = u_i$ 。当 $\{r_1(k), r_2(k), \dots, r_{n-1}(k)\}$ 确定时, τ_n 获得的资源也唯一确定,即

$$r_n(k) = R_{\max} - \sum_{i=1}^{n-1} r_i(k)$$

这是控制器不考虑第 n 个任务的误差作为输入的一个原因。由于 τ_n 获得的资源不是独立的,如果 $r_{\min} > 0$,则

$$\left. \begin{aligned} r'_{\max} &= r_{\max} - r_{\min}; \quad r'_i = r_i - r_{\min} \\ r'_{\min} &= 0; \quad R'_{\max} = R_{\max} - \sum_{i=1}^n r_{\min} \end{aligned} \right\} \quad (16)$$

当分配资源 $r'_i \in [0, r'_{\max}]$ 给 τ_i 时,实际分配的資源为 $r'_i + r_{\min}$,经式(16)的变换后,资源的上限记为 $R'_{\max}$,变换后的值为 $r_i(k), i \in [1, n]$ 。

考虑 $\{r_1(k), r_2(k), \dots, r_{n-1}(k)\}$ 取值的 2 种情况,如果

$$\sum_{i=1}^{n-1} r_i(k) \leq R'_{\max}$$

此时资源充足

$$r_n(k) = R'_{\max} - \sum_{i=1}^{n-1} r_i(k)$$

如果

$$R_{\text{sum}} = \sum_{i=1}^{n-1} r_i(k) > R'_{\max}$$

此时 τ_n 分配的资源 $r_n(k) = 0$,同时压缩 $\{r_1(k), r_2(k), \dots, r_{n-1}(k)\}$,以保证资源分配的可行性。由于 $r_i \in [0, r_{\max}]$,因此采用比例压缩各个任务的资源后

$$r_i^*(k) = r_i(k) R'_{\max} / R_{\text{sum}} \quad (17)$$

3 仿真实验

通过仿真实验,来分析评价采用神经网络PID控制的自适应资源的分配方法。仿真实验主要从任务资源分配控制激活(调整)次数、任务 QoS 资源分配的公平性、控制系统的误差收敛速度,以及稳定性范围等方面来衡量资源的分配方法。

仿真考虑了 10 个周期任务(嵌入式系统中并发应用数小于 10),仿真任务的参数设置如表 1 所示,其中任务作业的相对截止期等于相邻作业的间隔时间。本文采用 EDF 算法进行作业调度,作业完成时获得的 QoS 通过监控作业实际完成时刻与作业截止期的调度偏差进行量化。作业的完成时刻越早,对应的 QoS 就越高。通过记录任务作业获取的资源与其完成时产生的调度偏差,确定典型的服务质量函数如下

$$\varphi_i(r)=\frac{r-r_{imin}}{r_{imax}-r_{imin}}\tag{18}$$

$$\varphi_i(r)=\sin\left(\frac{\pi}{2}\frac{r-r_{imin}}{r_{imax}-r_{imin}}\right)\tag{19}$$

$$\varphi_i(r)=\frac{1}{2}\left(\sin\left(\frac{\pi r}{r_{imax}-r_{imin}}-\frac{\pi}{2}\right)+1\right)\tag{20}$$

$$\varphi_i(r)=1-\sin\left(\frac{\pi}{2}\frac{r_{imax}-r}{r_{imax}-r_{imin}}\right)\tag{21}$$

如表 2 所示,仿真任务从表 1 中选择任务的种类,提供初始资源分配 $\{r(0)\}$,在总资源 $R_{\max}=0.8$ 的条件下,比较神经网络 PID 控制和普通 PID 控制的性能。神经网络 PID 控制选择权值调整步长 $\eta_{sj}=0.1,\eta_{di}=0.2(s=h),\eta_{hi}=0.15(s\neq h)$ (s 为子网标识, h 为输出标识)。为了保证可行性和稳定性,PID 的积分控制系数^[7] $\alpha=0.159$ 。

表 1 任务的仿真参数

任务种类	周期/ms	相位/(°)	r_{imin}	r_{imax}
1	500	0	0	0.400
2	400	40	0	0.250
3	700	30	0	0.875
4	600	50	0	0.500
5	300	10	0	0.500
6	600	0	0	0.333

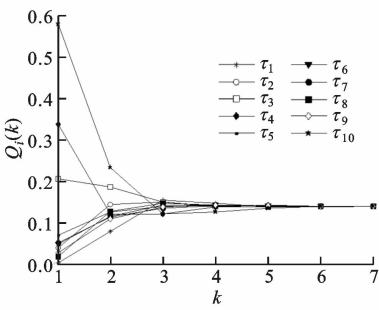
表 2 任务初始资源表

任务	τ_1	τ_2	τ_3	τ_4	τ_5
种类	2	6	3	2	4
$r(0)$	0.02	0.06	0.16	0.19	0.05

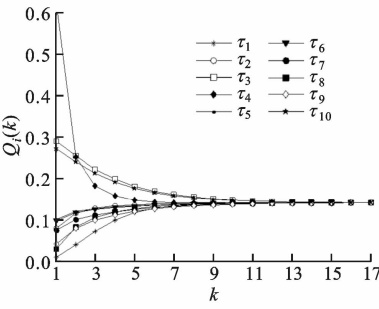
任务	τ_6	τ_7	τ_8	τ_9	τ_{10}
种类	5	3	1	4	3
$r(0)$	0.10	0.04	0.01	0.02	0.15

图 3 显示了 2 种控制方法任务 QoS 的变化曲线,图中每条曲线代表一个任务 QoS 的收敛过程,所有的 QoS 都收敛到同样的稳态值。图 4 显示了 2 种控制方法任务资源分配的变化曲线,任务资源最

终收敛到使得任务获得系统公平 QoS 均值 $\bar{Q}$ 时的资源值。

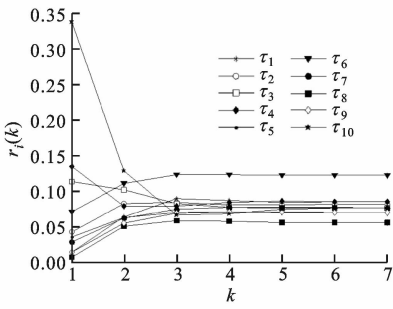


(a)神经网络 PID 控制 QoS

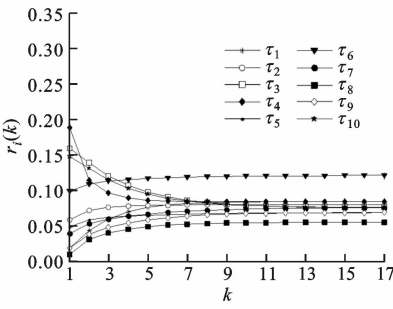


(b)PID 控制 QoS

图 3 2 种控制的 QoS 比较



(a)神经网络 PID 控制资源分配



(b)PID 控制资源分配

图 4 2 种资源分配过程的对比

在资源分配过程中,控制激活次数 k 表示资源的调整次数,资源调整带来的如磁盘读写等操作,在

嵌入设备中将带来较大的费用开支,因此控制激活次数越少,控制性能就越好。图3表明,神经网络PID控制比普通PID控制激活的次数少,表明神经网络PID的性能更好。

为了更好地与文献[7]中的积分控制进行比较,将神经网络PID控制简化成神经网络积分控制。取输入层至隐层的权值 $\omega_{ij} = \alpha = 0.159$, 连接隐层积分节点至输出层的权值为 $1 (s = h)$, 而其他权值为0, 并且权重调整算法只对积分节点相关权值进行处理。

本文采用均方误差来计算多任务的输出误差,由图5可以看出,神经网络PID控制的误差收敛下降最快且性能最好。

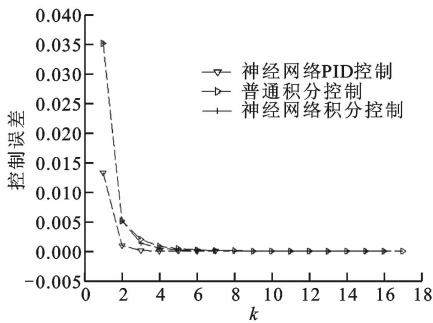
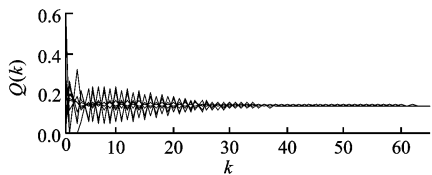
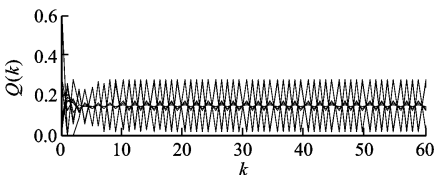


图5 控制误差的收敛曲线

当普通积分控制系数 $\alpha = 0.159$ 时, QoS 的收敛过程能保持稳定。图6为积分系数 $\alpha = 0.7$ 时, QoS 的收敛过程, 普通积分控制显然已经不能稳定收敛。实际上当 $\alpha > 0.159$ 时, 已经破坏了文献[7]提供的可行性和稳定性范围, 而本文采用的自适应公平分配方法的可行性和稳定性范围都比文献[7]中提供的普通积分控制方法的范围大得多。



(a)神经网络积分控制



(b)普通积分控制

图6 2种积分控制的QoS比较

4 结束语

本文针对有限资源开放式系统下多任务资源的分配问题,定义了任务的QoS公平性模型,并提出了采用神经网络PID自适应控制的资源分配方法。神经网络控制是通过微调PID参数,来解决复杂的非线性、不确定、不确定系统的控制收敛问题,从而能够在未知多任务特征参数且变化的情况下,保证任务的QoS快速收敛至系统QoS均值。神经网络控制资源调整算法的复杂度取决于权值调整算法的实现,而嵌入式系统的任务数相当有限,因此算法能瞬间完成分配决策。仿真实验表明,本文采用的神经网络PID控制的稳定性及收敛速度都强于普通的PID控制。

参考文献:

- [1] RAJKUMAR R, LEE C, LEHOCZKY J, et al. A resource allocation model for QoS management [C] // IEEE Real-Time Systems Symposium. Piscataway, NJ, USA: IEEE, 1997: 298-307.
- [2] 陈俊杰, 倪宏, 孙鹏. 采用定价机制的多媒体系统多资源分配算法 [J]. 西安交通大学学报, 2012, 46(6): 98-103.
CHEN Junjie, NI Hong, SUN Peng. Pricing mechanism based multi-resource allocation for multimedia system [J]. Journal of Xi'an Jiaotong University, 2012, 46(6): 98-103.
- [3] BUTTAZZO G, LIPARI G, ABENI L. Elastic task model for adaptive rate control [C] // IEEE Real-Time Systems Symposium. Piscataway, NJ, USA: IEEE, 1998: 286-295.
- [4] BUTTAZZO G, LIPARI G, CACCAMO M, et al. Elastic scheduling for flexible workload management [J]. IEEE Transactions on Computers, 2002, 51(3): 289-302.
- [5] ABENI L, BUTTAZZO G. Adaptive bandwidth reservation for multimedia computing [C] // IEEE Real-Time Computing Systems and Applications. Piscataway, NJ, USA: IEEE, 1999: 70-77.
- [6] DIAO Y, HELLERSTEIN J L, STROM A J, et al. Incorporating cost of control into the design of a load balancing controller [C] // IEEE Real-Time and Embedded Technology and Applications Symposium. Piscataway, NJ, USA: IEEE, 2004: 376-385.
- [7] HARADA F, USHIO T, NAKAMOTO Y. Adaptive resource allocation control for fair QoS management

(下转第136页)