

基于多片 FPGA 的双优先级动态调度算法

杜双枝¹, 王 勇^{2*}, 陶晓玲³

(1. 桂林电子科技大学 电子工程与自动化学院, 广西 桂林 541004;

2. 桂林电子科技大学 计算机科学与工程学院, 广西 桂林 541004; 3. 桂林电子科技大学 信息与通信学院, 广西 桂林 541004)

(* 通信作者电子邮箱 wang@guet.edu.cn)

摘 要: 针对单片现场可编程门阵列(FPGA)在处理高速网络中海量数据时存在效率低下的问题,结合多处理器的双优先级调度算法,在所构建的多片FPGA并行处理的高速数据采集和处理模型上,提出一种基于多片FPGA的双优先级动态调度算法,并对处于低优先级段的强实时周期任务提出一种最早截止期临界松弛调度(EDCL)算法。根据任务的松弛度确定任务的优先级,若提升时间到达时仍未完成,则将其提升到高优先级段;对软实时周期任务,设置在中优先级段,通过延长当前任务截止期至动态模糊阈值进行调度。实验结果表明,该算法能很好地调度强实时周期任务,保证重要任务的优先执行,并能降低由于抢占造成的软实时周期任务错失率。

关键词: 并行处理; 任务调度; 多片现场可编程门阵列; 双优先级调度算法; 松弛度

中图分类号: TP393.08 **文献标志码:** A

Dual priority dynamic scheduling algorithm based on multi-FPGA

DU Shuangzhi¹, WANG Yong^{2*}, TAO Xiaoling³

(1. College of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin Guangxi 541004, China;

2. College of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin Guangxi 541004, China;

3. College of Information and Communication, Guilin University of Electronic Technology, Guilin Guangxi 541004, China)

Abstract: When single Field-Programmable Gate Array (FPGA) deals with the huge amounts of data in high-speed network, low efficiency problem occurs. According to dual priority schedule algorithm for multi-processor and high-speed data acquisition and processing model based on multi-FPGA, a dual priority dynamic scheduling algorithm was proposed based on multi-FPGA. For strong real-time periodic tasks set in low priority queue, the Earliest Deadline Critical Laxity (EDCL) scheduling algorithm was given to determine the priority of task according to the degree of relaxation of the tasks. If the task was not finished when the promotion time was up, it would be promoted to high priority queue. For soft real-time periodic tasks, an algorithm was put forward to assign the tasks to middle priority queue and schedule them by delaying the deadline of tasks to dynamic blur threshold. The experimental results show that the proposed algorithms can well schedule strong real-time periodic tasks and guarantee the priority execution of important tasks, and it can also reduce miss rate of soft real-time periodic tasks caused by preemption.

Key words: parallel processing; tasks scheduling; multiple Field-Programmable Gate Array (FPGA); dual priority scheduling algorithm; relaxation degree

0 引言

随着计算机网络技术的发展和互联网规模的扩大,保障网络的可靠运行变得越来越重要。为了实现对网络安全的有效监控,必须及时处理网络中的海量流量信息。而现场可编程门阵列(Field-Programmable Gate Array, FPGA)由于其高度的并行性,可使数据分析、处理、替换等并行进行,因此被广泛地应用于高速数据采集系统中^[1-5]。但单片FPGA由于芯片资源的限制往往不足以胜任全部系统功能,即使可以胜任,在开发一块高密度FPGA时,很可能面临模块分割、综合布线、系统维护等方面的困难,从而导致开发效率下降。根据一定的耦合关系使用多片FPGA构建一个大规模处理器是解决这个问题有效方法。

在高速数据采集系统中,为了快速处理海量的数据包,可以采用多片FPGA级联的结构并行处理流量数据。一个合理的任务调度算法,应使多片FPGA处理效率最高,任务执行时

间最短。虽然多片FPGA的任务调度问题报道不多,但是多处理器的任务调度算法一直是个重要的研究课题,调度算法层出不穷^[6-11]。

多处理器调度中的双优先级算法与EDF(Earliest Deadline First)、ESF(Earliest Start time First)、MPF(Minimum Processor time First)、MLF(Minimum Laxity First)等复杂算法相比,它是将全局调度和局部调度相结合,具有开销小、性能高、实现简单的优点,比较适合处理实时调度问题。其最早在1993年由Burns和Wellings提出并应用于实时系统的任务调度,以充分利用处理器的容量;1995年Davis和Wellings利用双优先级算法调度软实时和强实时任务,在保证强实时任务满足时限的情况下,调度软实时任务。Banus等^[12]又将双优先级算法应用在多核处理器系统中,提出基于多核处理器系统的强实时周期任务和非周期任务的混合调度算法;刘怀等^[13]将双优先级算法进行改进,使之可以在单核上调度强实时周期性任务、软实时周期性任务和非周期性任务;朱俊

收稿日期: 2012-08-28; 修回日期: 2012-10-23。 基金项目: 国家自然科学基金资助项目(61163058); 广西自然科学基金资助项目(2011GXNSFB018076); 广西科学研究与技术开发计划项目(桂科攻11107006-21)。

作者简介: 杜双枝(1986-),女,湖北随州人,硕士研究生,主要研究方向: 计算机网络流量分类; 王勇(1964-),男,四川阆中人,教授,博士,主要研究方向: 计算机网络、信息安全、计算智能; 陶晓玲(1977-),女,江苏金华人,副研究员,硕士,主要研究方向: 网络安全、计算智能。

超^[14]在文献[13]的基础上改进双优先级算法,使之可以在多处理器系统中调度强实时周期性任务和软实时周期性任务。

由于多片FPGA与多处理器的任务调度问题类似,因此本文借鉴多处理器双优先级调度算法,针对采用多片FPGA级联并行处理海量数据包的任务调度问题,提出一种基于多片FPGA的双优先级动态调度算法,并对强实时周期任务的优先级判定方法加以改进。

1 基于多片FPGA的双优先级任务调度模型

1.1 多片FPGA并行高速数据处理模型

现有的高速数据采集和处理系统大多是以单片FPGA为控制核心实现的,本文实现的系统是以单片FPGA为核心,多片FPGA级联并行处理数据包。首先,硬件电路从千兆以太网口捕获的网络数据包经过RJ45接口通过A/D转换电路进入到网络接口模块;然后网络接口模块与FPGA处理模块中的千兆以太网IP核通过标准的SGMII接口连接,实现网络数据包的采集;最后,采集到的网络数据包由FPGA处理模块根据CAM内容寻址存储器的用户规则进行相应的分析处理后,根据各个FPGA上的负载情况,采取适当的调度策略和负载均衡策略,将任务分配到各片FPGA上,以使系统完成时间最短,资源利用率最高。其模型如图1所示。

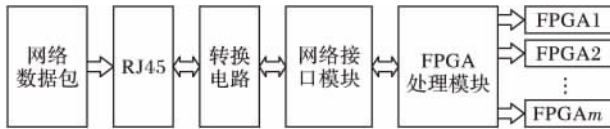


图1 多片FPGA并行高速数据处理模型

多片FPGA任务调度是将 n 个任务分配到 m 片FPGA上,每个任务可以分配到任意一片FPGA上执行。其可调度性判断依据为:

对于单片FPGA:

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

对于 m 片FPGA:

$$U' = \sum_{j=1}^m \sum_{i=1}^n \frac{C_i}{P_i} \leq m \quad (2)$$

其中: U 和 U' 表示系统总利用率, C_i 为周期任务的最坏执行时间, P_i 为任务的周期。只有满足式(1)和式(2)的条件,才可能处理所有负载。

1.2 双优先级任务调度模型

定义1 强实时周期任务。必须在规定的时间内完成,不允许错失截止期限的任务。如果出现超出错失截止期限的强实时任务,则会对系统造成不可估量的损失。

定义2 软实时周期任务。通常允许任务错失截止期限,错失截止期限后的计算结果仍然有一定的意义,但其意义随着超时时间的增加而下降。

定义3 最坏执行时间。任务在最坏情况下无中断地执行所需的处理器时间。

定义4 任务周期。周期任务两次执行的时间间隔。

定义5 截止期。指一个时刻 t ,在此时刻 t 之前要求实时任务完成它的执行。

定义6 强实时周期任务优先级提升时间。指强实时周期任务实例的优先级从低优先级提升到高优先级的时间。

据此,本文对强实时周期任务和软实时周期任务作如下形式化描述:

强实时周期任务集表示为 $S_h = \{\tau_{h1}, \tau_{h2}, \dots, \tau_{hn}\} (n \geq 1)$,每个强实时周期任务 τ_{hi} 描述为 $\tau_{hi} = \langle C_{hi}, T_{hi}, D_{hi}, P_{hi} \rangle (i = 1, 2, \dots, n)$,其中 $C_{hi}, T_{hi}, D_{hi}, P_{hi}$ 分别表示任务 τ_{hi} 的最坏执行时间、任务周期、截止期和优先级提升时间。

软实时周期任务集表示为 $S_s = \{\tau_{s1}, \tau_{s2}, \dots, \tau_{sn}\} (n \geq 1)$,每个软实时周期任务 τ_{si} 描述为 $\tau_{si} = \langle C_{si}, T_{si}, D_{si} \rangle (i = 1, 2, \dots, n)$,其中 C_{si}, T_{si}, D_{si} 分别表示任务 τ_{si} 最坏执行时间、任务周期和截止期。

根据多核多任务调度模型,把双优先级调度算法应用到多片FPGA调度中,提出基于多片FPGA的双优先级算法的任务调度模型如图2所示。

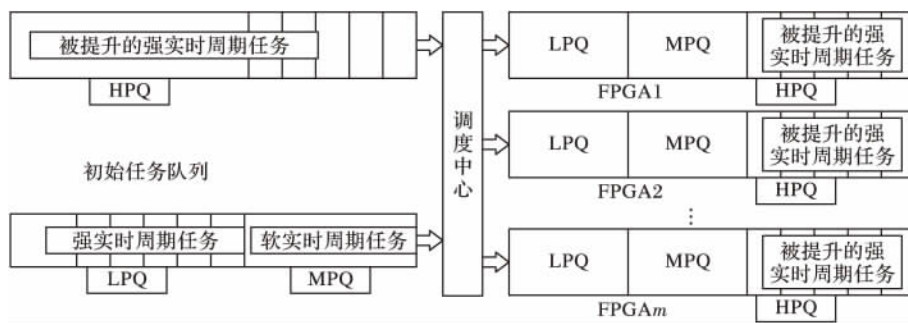


图2 基于多片FPGA的双优先级任务调度模型

如图2所示,本文采用集中式调度,初始时,把强实时周期任务放到低优先级段(Low Priority Queue, LPQ),经过各个任务优先级提升后,占有高优先级段(High Priority Queue, HPQ);软实时周期任务放入到中优先级段(Middle Priority Queue, MPQ)中。各片FPGA上也有各自的优先级队列。

1.3 强实时周期任务优先级判定及调度算法改进

定义7 剩余执行时间。任务完成时刻与当前时刻之间的差值。

定义8 松弛度。从当前时刻直到其截止期的时间与其剩余执行时间之间的差。

基于多处理器的双优先级算法中,直接将单处理器的单速率调度(Rate-Monotonic Scheduling, RMS)算法应用到多

处理器环境中,此时的RMS算法已不是最优算法,因此本文提出一种最早截止期限临界松弛调度(Earliest Deadline Critical Laxity, EDCL)算法调度处于低优先级段中的强实时周期任务。

EDCL算法是最早截止期零松弛度(Earliest Deadline Zero Laxity, EDZL)算法的改进,基本思想是只要任务的松弛度达到一定的值,就根据全局EDF调度算法确定任务的优先级。一旦任务的松弛度低于这个值时,任务会被提升到高优先级段中,以在截止期内完成。

多片FPGA处理系统中,初始队列LPQ中的强实时周期任务的优先级按照最早截止期限临界松弛调度EDCL算法来确定。算法伪代码如下所示:

```

when any tasks are released or complete do
  if  $n < m$  then
    execute all tasks in  $S_h$ ;
  else
     $C_{\min} = \min\{C_{hi}(P_{hi}) \mid \tau_{hi} \in S_h\}$ ;
     $\zeta = \{\tau_{hi} \mid lax_i(P_{hi}) < e_{\min}\}$ ;
    if  $|\zeta| \leq m$  then
      execute  $m$  tasks in  $\zeta$ ;
    else
      execute all tasks in  $\zeta$ ;
      execute  $m - |\zeta|$  tasks with the earliest deadlines in  $S_h$ ;
    end if
  end if
end when

```

步骤如下:

1) 由于各片 FPGA 是同构的,其处理速度一样,如果任务数 $n < m$ (m 为处理器数目),从中任意选择 n 片 FPGA 分配任务即可。

2) 如果任务数 $n > m$,EDCL 算法按照截止期,选出其中前 m 个最早截止期的任务分配到 FPGA 上;计算这 m 个任务的剩余执行时间,并选出其中的最小剩余执行时间 $C_{\min}$ 。

3) 执行过程中,如果等待队列中有任务的松弛度小于 $C_{\min}$,那么该任务就是临界任务。

①如果临界任务数目 $|\zeta| \geq m$,则按照各自的松弛度从小到大排序,根据各片 FPGA 的负载执行情况,从 ζ 中选择前 m 个任务,将它们提升到负载最小的 FPGA 的最高优先级立即执行,然后再去接着执行非临界截止期任务;

②如果临界任务数 $|\zeta| < m$,则从 m 片 FPGA 上选择负载较小的 $|\zeta|$ 片 FPGA 进行负载分配。

每片 FPGA 上也有 3 个优先级队列: HPQ、MPQ 和 LPQ。分配给各片 FPGA 的任务,由于各片 FPGA 上会存在未完成的的任务,因此需要等待一段时间,此时任务的优先级仍然通过最早截止期临界松弛度算法进行确定。

1.4 基于多片 FPGA 的双优先级动态调度算法

基于多片 FPGA 的双优先级动态调度算法描述如下:

1) 强实时周期任务拥有 HPQ 和 LPQ,软实时周期任务拥有 MPQ,HPQ 中的任务总是比 MPQ 和 LPQ 中任务的优先级高,而 MPQ 中任务的优先级总是高于 LPQ 中的优先级。

2) 对于强实时周期任务,其调度方法按照下列步骤进行:

①将强实时周期任务按照 1.3 节描述的全局调度算法分配到指定的 FPGA 上的 LPQ 中,并由调度中心离线计算各自的提升时间。采用固定优先级提升时间,并且在系统运行时保持不变。提升时间的计算公式如下:

$$W_i^{n+1} = \sum_{j \in hp(i)} \lceil W_i^n / T_{hj} \rceil C_{hj} + C_{hi} \quad (3)$$

其中: $hp(i)$ 表示优先级高于 τ_{hi} 的所有强实时周期任务集合; C_{hi} 和 T_{hj} 分别是强实时周期任务的最差执行时间和周期;递归起始条件 $W_i^0 = 0$,结束条件 $W_i^{n+1} = W_i^n$; τ_{hi} 的提升时间 $P_{hi} = D_{hi} - W_i^0$ 。

②在到达提升时间之前,强实时周期任务和软实时周期任务一起按照各自的优先级进行全局的动态调度。

③到达提升时间时,则立即将强实时周期任务从预先分配好的 FPGA 上的 LPQ 提升到 HPQ,使其优先执行,保证其能够在其有效时限内完成。

④对于新到达的强实时周期任务,首先判断是否有空闲的 FPGA。如果有空闲 FPGA 则调度任务在空闲 FPGA 上执

行;如果此时没有空闲的 FPGA,就会对预先设定的 FPGA 进行判断,如果预先设定的 FPGA 上正在运行的是软实时周期任务,那么将进行无条件的抢占;如果 FPGA 上运行的是强实时周期任务,那么也将发生抢占,而正在运行的强实时周期任务将会按照同样的方式进行抢占。

3) 对于软实时周期任务,根据各片 FPGA 上的负载情况,分配到相应的 FPGA 的 MPQ 中,该类任务可以抢占 LPQ 中的任务执行。当被 HPQ 中的强实时周期任务抢占时,通过延长当前任务截止期至动态模糊阈值的方法进行调度^[16],先让强实时周期任务执行,若强实时周期任务执行完毕,则接着执行软实时周期未完成的部分,若此时已经超过软实时周期任务的截止期,则丢弃该任务。

2 实验结果及分析

实验采用 Matlab 软件进行仿真,每次随机产生 50 个任务,任务集分为强实时周期任务、软实时周期任务。记 $M(X, Y)$ 表示 X, Y 之间随机分布函数,设每个强实时周期任务和软实时周期任务的周期范围 $M(10, 1000)$,每个 FPGA 预先分配 5 个强实时周期任务,并使每个 FPGA 预置的负载利用率相同,软实时周期任务数固定为 10。本实验中, FPGA 的片数 m 取值范围为 2~6。

限定每次仿真运行的持续时间为 1000 个时间单元,以下所有的仿真结果都是由 100 次独立仿真结果的平均值得到的。

衡量调度算法性能的标准:

1) 系统总利用率: $U' = \sum_{j=1}^m \sum_{i=1}^n \frac{C_i}{P_i}$ 。其中: C_i 表示任务的最坏执行时间, P_i 表示任务的周期, n 为单片 FPGA 上任务数目, m 为 FPGA 总数目。

2) 软实时周期任务平均丢失率: 指在软实时周期任务的执行过程中,由于错过任务时限而被丢弃的软实时周期任务实例个数与到达总个数之比。

3) 任务丢失率: 算法调度失败的任务个数与参与调度的总任务个数之比。

4) 截止期错过率: 单位时间内实时任务错过任务截止期的数目与这段时间内的任务总数的比值。

图 3 给出了系统实际利用率对软实时周期任务平均丢失率的影响,其中软实时周期任务平均丢失率是指在软实时周期任务的执行过程中,由于错过任务时限而被丢弃的软实时周期任务实例个数与到达总个数之比。

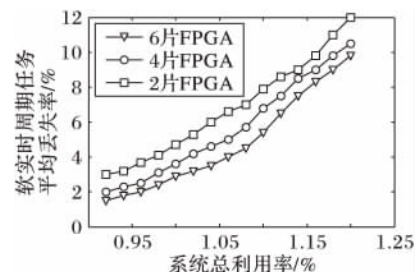


图3 系统实际利用率对软实时周期任务丢失率的影响

由图 3 可知,在系统利用率相同的情况下, FPGA 的片数越多其软实时周期任务的丢失率越小。但不能无限地扩大 FPGA 的数量,这样会增加成本,因此以后的工作就是进一步考虑 FPGA 使用数量问题,使得任务丢失率小且成本最小。

在 4 片 FPGA 时分别对 EDF、双优先级算法 (dual-priorities)、PFair 这三种调度算法进行调度性能分析,在相同情况下得到各自的任务丢失率如图 4 所示。当负载小于 4

时,双优先级算法任务错失率介于 EDF 和 PFair 算法之间,但负载一旦超过 4,双优先级算法任务错失率明显要低于另外两种算法。

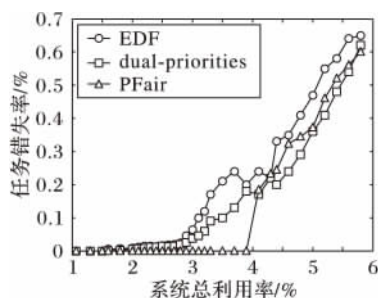


图4 三种调度算法任务错失率对比

双优先级算法中用 EDCL 算法调度强实时周期任务,用延长截止期的方法调度软实时周期任务,使得任务截止期错失率明显下降。三种调度算法性能比较如图 5 所示。

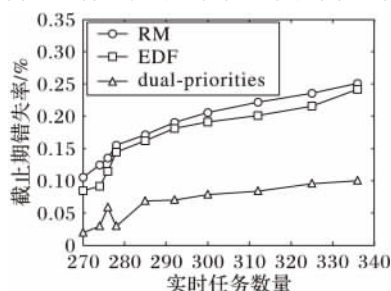


图5 截止期错失率测试结果

实验结果表明,本文提出的调度算法可以很好地调度强实时任务,保证重要任务的优先执行,并降低了强实时任务抢占软实时任务造成的软实时任务错失率。

3 结语

本文针对高速网络中的海量数据处理效率低下问题,提出了一种多片 FPGA 并行处理的模型,并基于此模型,对强实时周期任务提出了一种基于最早截止期临界零松弛度的优先级判定方法,从而保证了强实时周期任务的优先执行。虽然 FPGA 使用的数目越多,但成本会越高,因此需要进一步实现模型中 FPGA 的数量最优问题。

参考文献:

[1] 刘航,戴冠中,李晖晖,等.基于FPGA的高速网络入侵检测系统

[J]. 计算机应用,2004,24(5):33-35.

- [2] 唐杰敏. 基于FPGA的以太网流量发生器的设计与实现[D]. 上海: 同济大学,2008.
- [3] 林洪周. 兆网络数据包捕获系统的研究与开发[D]. 武汉: 华中科技大学,2008.
- [4] 何毅华,易清明,石敏. 低成本数据传输存储系统的FPGA实现[J]. 计算机应用,2009,29(12):3238-3240,3243.
- [5] 朱晴,吴宁,顾薛平. 基于FPGA的千兆网络数据采集系统设计与实现[J]. 微型机与应用,2011,30(21):53-55,59.
- [6] SUGANYA K, NAGARAJAN V. Efficient run-time task allocation in reconfigurable multiprocessor system-on-chip with network-on-chip [C]// 2011 International Conference on Computer, Communication and Electrical Technology. Piscataway, NJ: IEEE Press, 2011: 12-17.
- [7] QI X, ZHU D K, AYDIN H. Global scheduling based reliability-aware power management for multiprocessor real-time systems [J]. Journal of Real-Time Systems, 2011, 47(2): 109-142.
- [8] DAVIS R I, BURNS A. Improved priority assignment for global fixed priority preemptive scheduling in multiprocessor real-time systems [J]. Real-Time Systems Journal, 2010, 47(1): 1-40.
- [9] LI K. Energy efficient scheduling of parallel tasks on multiprocessor computers [J]. The Journal of Supercomputing, 2012, 60(2): 223-247.
- [10] ELLIOTT G A, ANDERSON J H. Globally scheduled real-time multiprocessor systems with GPUs [J]. Real-Time Systems, 2012, 48(1): 34-74.
- [11] SALAMY H, RAMANUJAM J. An effective solution to task scheduling and memory partitioning for multiprocessor system-on-chip [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(5): 717-725.
- [12] BANUS J M, ARENAS A. Dual priority algorithm to schedule real-time tasks in a shared memory multiprocessor [C]// IPDPS 03: Proceedings of the 17th International Parallel and Distributed Processing Symposium. Washington, DC: IEEE Computer Society, 2003: 112.
- [13] 刘怀,费树岷. 基于双优先级的实时多任务动态调度[J]. 计算机工程, 2005, 31(18): 16-18.
- [14] 朱俊超. 基于多处理器的双优先级调度算法改进与实现[D]. 大连: 大连理工大学,2010.
- [15] GOOSSENS J, FUNK S, BARUAH S. Priority-driven scheduling of periodic task systems on multiprocessors [J]. Real-Time Systems, 2003, 25(2/3): 187-205.
- [16] 李琦,巴巍. 两种改进的EDF软实时动态调度算法[J]. 计算机学报, 2011, 34(5): 943-949.

(上接第861页)

- [4] WASLANDER S L, HOFFMANN G M, JANG J S, et al. Multi-Agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning [C]// International Conference on Intelligent Robots and Systems. Piscataway, NJ: IEEE Press, 2005: 468-473.
- [5] BOUABDALLAH S, SIEGWART R. Backstepping and sliding mode techniques applied to an indoor micro quadrotor [C]// International Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2005: 2247-2252.
- [6] 薛文涛,吴晓蓓,翟玉强,等. 三自由度飞行器模型的神经网络PID控制[J]. 控制工程, 2009, 16(2): 214-219.
- [7] 李光春,王璐,王兆龙,等. 基于四元数的四旋翼无人飞行器轨迹跟踪控制[J]. 应用科学学报, 2012, 30(4): 415-422.
- [8] 王璐,李光春,王兆龙,等. 欠驱动四旋翼无人飞行器的滑模控制[J]. 哈尔滨工程大学学报, 2012, 33(10): 1248-1253.
- [9] 卓开育,陆秋君. 基于改进RBF神经网络的车辆类型识别[J]. 计算机应用, 2011, 31(S2): 85-89.
- [10] 贺可鑫,何小海,陶青川,等. 基于RBF神经网络的COSM图像复原算法[J]. 计算机应用, 2009, 29(S1): 79-85.

- [11] 刁琛. 四旋翼无人飞行器的非线性自适应控制器设计[D]. 天津: 天津大学,2010: 13-18.
- [12] 周东华. 非线性系统的自适应控制导论[M]. 北京: 清华大学出版社,2002: 88-103.
- [13] SLOTINE J J, LI W P. 应用非线性控制[M]. 程代展,译. 北京: 机械工业出版社,2004: 211-264.
- [14] 张利军,贾鹤鸣,边信黔,等. 基于L2干扰抑制的水下机器人三维航迹跟踪控制[J]. 控制理论与应用, 2011, 28(5): 645-651.
- [15] FUNAHASHI K-I. On the approximate realization of continuous mappings by neural networks [J]. Neural Networks, 1989, 2(3): 183-192.
- [16] 贾鹤鸣,宋文龙,郭婧. 基于神经网络滑模的采摘机械臂控制设计[J]. 吉林大学学报: 工学版, 2012, 42(3): 709-713.
- [17] POUNDS P, MAHONY R, CORKE P. Modelling and control of a large quadrotor robot [J]. Control Engineering Practice, 2010, 18(7): 691-699.