

# 利用 T-S 模糊自适应 PSO 算法优化 PID 参数

郭 成, 李群湛

GUO Cheng, LI Qun-zhan

西南交通大学 电气工程学院, 成都 610031

School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China

E-mail: gc325@126.com

**GUO Cheng, LI Qun-zhan. PID parameter optimization using T-S fuzzy adaptive particle swarm optimization. Computer Engineering and Applications, 2009, 45(3): 245-248.**

**Abstract:** In order to solve the premature convergence problem of particle swarm optimization, a novel fuzzy adaptive Particle Swarm Optimization based on T-S model (T-SPSO) is presented. The proposed method shapes the T-S rules according to the current best performance evaluation and inertia weight of swarm, which dynamically update the value of inertia weight and significantly speed up the convergence. The improved algorithm significantly improves the performance of parameters applied in parameter setting of PID controller. The simulation results illustrate the effectiveness of this proposed method and superiority of the controller.

**Key words:** Particle Swarm Optimization (PSO); PID control; parameter optimization; Particle Swarm Optimization based on T-S model (T-SPSO); premature

**摘 要:** 针对微粒群优化算法存在的早熟问题, 提出了一种基于 T-S 模型的模糊自适应 PSO 算法 (T-SPSO 算法)。算法依据种群当前最优性能指标和惯性权重值所制定 T-S 规则, 动态自适应惯性权重取值, 改善了 PSO 算法的收敛性。将该算法应用于 PID 控制器的参数整定, 可得到更优的控制参数。仿真结果验证了所提出算法的有效性和所设计控制器的优越性。

**关键词:** 微粒群优化算法; PID 控制; 参数优化; 基于 T-S 模型的模糊自适应 PSO 算法; 早熟

**DOI:** 10.3778/j.issn.1002-8331.2009.03.074 **文章编号:** 1002-8331(2009)03-0245-04 **文献标识码:** A **中图分类号:** TP273.2

PID 控制即比例-积分-微分控制, 它是建立在经典控制理论上的一种控制策略。PID 控制具有结构简单, 易于实现的优点, 能够满足多数工业控制的性能要求, 因而广泛应用于控制领域。PID 控制器的参数整定是 PID 控制器设计的关键所在。目前 PID 参数整定方法主要有两种: 一是基于传统的参数整定方法, 如 ZN 法, 单纯形法等; 二是采用智能化的整定方法, 如基于遗传算法<sup>[1]</sup>、蚁群算法<sup>[2]</sup>、模糊逻辑<sup>[3]</sup>、神经网络<sup>[4]</sup>等智能 PID 优化方法。

微粒群优化算法<sup>[5]</sup>是由 Kennedy 和 Eberhart 博士于 1995 年受鸟类群体行为研究结果的启发, 并利用了生物学家 Frank Heppner 的生物群体模型而提出的一种高效、简单的并行搜索算法, 其优点在于概念简单、实现容易、鲁棒性好, 并且能以较大概率收敛到全局最优。PSO 算法中惯性权重的取值对改善算法的收敛性起着至关重要的作用, 文献[6]提出一种线性递减惯性权重值的 PSO 算法, 文献[7]将这种算法应用于 PID 参数优化, 然而由于控制对象的非线性, 线性递减关系可能只对某些问题有效, 对于大多数问题显然不是最佳的。文献[8]提出一种基于 mamdani 模糊模型的惯性权重调整方法, 仿真结果表明其收敛性优于线性递减惯性权重方法。然而基于 mamdani 模型

的模糊系统由于缺乏系统的稳定性分析和控制器设计方法长期以来受到质疑<sup>[9]</sup>。

提出一种基于 T-S 模型的模糊自适应 PSO 算法 (T-SPSO 算法), 并用于 PID 控制器参数的优化设计。算法依据种群所处收敛阶段, 通过 T-S 规则, 动态自适应更新惯性权重取值, 使其在起始阶段以较大惯性权重值保证算法的全局搜索能力, 而在收敛的后期阶段以较小惯性权重值加快收敛, 从而解决了 PSO 算法的早熟问题, 改善了算法的收敛性, 可以更好地优化 PID 控制器参数。仿真结果验证了算法的有效性和优越性。

## 1 T-SPSO 算法

### 1.1 PSO 算法描述

在  $N$  维空间中有  $m$  个微粒, 设  $x_i=(x_{i1}, x_{i2}, \dots, x_{in})$ ,  $v_i=(v_{i1}, v_{i2}, \dots, v_{in})$  分别为微粒  $i$  的当前位置、当前速度; 设  $p_i=(p_{i1}, p_{i2}, \dots, p_{in})$  为微粒  $i$  所经历的最好位置,  $p_g$  为群体中所有微粒所经历的最好位置。对每一代微粒的速度和位置, 根据式(1)~(3)进化:

$$v_{id}^{t+1} = \omega \cdot v_{id}^t + c_1 \cdot r_1 \cdot (p_{id}^t - x_{id}^t) + c_2 \cdot r_2 \cdot (p_{gd}^t - x_{id}^t) \quad (1)$$

**基金项目:** 国家科技支撑计划资助项目 (Project Supported by the National Science and Technology Support Program under Grant No.2007BAA12B5)。

**作者简介:** 郭成 (1978-), 男, 博士生, 从事智能控制、智能控制在电力系统中应用研究; 李群湛 (1957-), 男, 博士, 教授, 博导, 从事牵引供电理论、电能质量控制研究。

**收稿日期:** 2008-07-21 **修回日期:** 2008-09-21

$$\begin{cases} v_{id}^{t+1} = v_{\max}, & \text{if } v_{id}^{t+1} > v_{\max} \\ v_{id}^{t+1} = -v_{\max}, & \text{if } v_{id}^{t+1} < -v_{\max} \end{cases} \quad (2)$$

$$x_i^{t+1} = x_i^t + v_{id}^{t+1} \quad (3)$$

式中,  $t$  或  $t+1$  表示迭代次数,  $\omega$  为惯性权重,  $c_1$  和  $c_2$  是加速常数, 通常取  $c_1=c_2=2$ ,  $r_1$  和  $r_2$  为两个在  $[0, 1]$  范围内变化的随机数,  $v_{\max}$  为最大飞行速度。

设  $f(x)$  为最小化的目标函数, 则微粒  $i$  的当前最好位置  $p_i^{t+1}$  由式(4)确定:

$$p_i^{t+1} = \begin{cases} x_i^t & f(x_i^{t+1}) \geq f(x_i^t) \\ x_i^{t+1} & f(x_i^{t+1}) < f(x_i^t) \end{cases} \quad (4)$$

设群体中所有微粒所经历的最好位置为  $p_g$ , 则所有微粒的最好位置由式(5)确定:

$$p_{gn} = \min\{f(p_{1n}^t), f(p_{2n}^t), \dots, f(p_{nn}^t)\} \quad (5)$$

## 1.2 T-S 模糊模型

由 Takagi 和 Sugeno 提出的 T-S 模糊模型是由 IF-THEN 规则描述的, 这些规则代表了非线性系统局部的输入与输出关系<sup>[10]</sup>。全局的模糊系统模型是这些线性系统的“组合”。T-S 模糊模型可以任意精确地描述一个非线性系统。

一阶 T-S 模糊模型可表述为:

$$R^i: \text{If } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \dots \text{and } x_k \text{ is } A_k^i \\ \text{Then } y = p_0 + p_1 x_1 + \dots + p_k x_k \quad (i=1, 2, \dots, l)$$

式中  $A_j^i$  ( $j=1, 2, \dots, n$ ) 为模糊集合, 简称前件参数。  $x_1 \dots x_k$  为隐含前件变量,  $p_0 \dots p_k$  为后件参数,  $l$  为规则的个数。

## 1.3 T-SPSO 算法

惯性权重  $\omega$  对 PSO 算法的收敛性有着重要的作用。  $\omega$  越大, 微粒的飞行速度越大, 微粒将以较大步长进行搜索。  $\omega$  越小, 其飞行速度越小, 微粒更易于收敛。在搜索过程中通过对  $\omega$  进行动态调整可以获得最佳的收敛性。即: 在算法的初期阶段赋予  $\omega$  一个较大的正值, 以获得较好的全局搜索能力, 而在算法的后期阶段赋予  $\omega$  一个较小的值, 可使算法更易于收敛。

由此, 提出一种基于 T-S 模型的 T-SPSO 算法, 用以动态解决在不同收敛阶段的惯性权重取值问题。 T-SPSO 算法需要两个输入参数: 当前种群最优性能指标 (NCBPE) 和当前的惯性权重  $\omega$  值。算法的输出参数为惯性权重的变化值。以最小化的目标函数为例, NCBPE 定义为:

$$NCBPE = \frac{CBPE - CBPE_{\min}}{CBPE_{\max} - CBPE_{\min}} \quad (7)$$

其中, NCBPE 为当前种群最优性能指标; CBPE 为当前种群最优适应值函数值;  $CBPE_{\max}$  为最大适应值函数估计值;  $CBPE_{\min}$  为最小适应值函数估计值。

两个输入参数均采用 trimf 隶属度函数, 在对输入变量模糊化的过程中, 对输入变量 NCBPE 和  $\omega$  采用相同的模糊词集: 低(L)、中(M)、高(H), 用集合表示为: (L, M, H)。对两个输入变量的隶属度函数边界设定如表 1 所示。

表 1 输入变量隶属度函数边界设定

输入变量	L	M	H
NCBPE	(0, 0.12)	(0.1, 0.4)	(0.35, 1)
$\omega$	(0.2, 0.5)	(0.45, 0.95)	(0.9, 1.3)

$CBPE_{\max}$  是待优化控制器的最大性能指标允许值,  $CBPE_{\min}$  代表的是期望达到的最小性能指标。 NCBPE 是当前所有微粒已搜索到的最优适应值性能指标, 它体现的是当前种群隶属于哪一个收敛阶段。事实上, NCBPE 的取值与  $CBPE_{\max}$  和  $CBPE_{\min}$  的值密切相关, 因此它们的值直接决定着算法应在何时加强全局搜索, 何时进入收敛阶段。对于 NCBPE 隶属的不同模糊集应制定不同的 T-S 规则。考虑以下几种情况:

(1) 若当前 CBPE 值大于  $CBPE_{\max}$  值, 此时 NCBPE 大于 1 (即算法还未收敛到规定的最大性能指标范围之内), 则  $\omega$  应取最大值, 以获得最大的搜索速度。

(2) 若当前 CBPE 值小于  $CBPE_{\max}$  值, 但 NCBPE 隶属于大, 则  $\omega$  也应取大, 使微粒获得较大搜索步长, 避免陷入局部最优, 继续探索更优适应值。

(3) 若当前 CBPE 值小于  $CBPE_{\max}$  值, 但 NCBPE 隶属于中, 则  $\omega$  不能过大, 以免微粒“飞跃”最优适应值;  $\omega$  也不能过小, 以免算法收敛速度太慢或微粒陷入局部最优。

(4) 若当前 CBPE 值已接近于  $CBPE_{\min}$ , 即此阶段算法已十分接近所期望的优化值, 此时 NCBPE 隶属于小, 则  $\omega$  也一定要取小, 使微粒易于收敛。

在搜索过程中, 当前的 NCBPE 和  $\omega$  值通过 T-S 规则, 作用于  $\omega$  的变化, 自适应更新  $\omega$  值, 以便  $\omega$  能够动态追踪 NCBPE 的变化, 使算法在不同的收敛阶段, 可以取到较为适合的  $\omega$  值, 从而达到改善算法收敛性的目的。

由以上分析, 制定 T-S 规则如下:

规则 1 If NCBPE is L and  $\omega$  is L, then  $w\_change = 0.002 \times NCBPE + 0.002 \times w$ ;

规则 2 If NCBPE is L and  $\omega$  is M, then  $w\_change = 0.01 \times NCBPE + 0.01 \times w + 0.005$ ;

规则 3 If NCBPE is L and  $\omega$  is H, then  $w\_change = 0.02 \times NCBPE + 0.02 \times w + 0.015$ ;

规则 4 If NCBPE is M and  $\omega$  is L, then  $w\_change = -0.01 \times NCBPE - 0.01 \times w - 0.005$ ;

规则 5 If NCBPE is M and  $\omega$  is M, then  $w\_change = 0.002 \times NCBPE + 0.002 \times w$ ;

规则 6 If NCBPE is M and  $\omega$  is H, then  $w\_change = 0.01 \times NCBPE + 0.01 \times w + 0.005$ ;

规则 7 If NCBPE is H and  $\omega$  is L, then  $w\_change = -0.02 \times NCBPE - 0.02 \times w + 0.015$ ;

规则 8 If NCBPE is H and  $\omega$  is M, then  $w\_change = -0.01 \times NCBPE - 0.01 \times w - 0.005$ ;

规则 9 If NCBPE is H and  $\omega$  is H, then  $w\_change = 0.002 \times NCBPE + 0.002 \times w$ ;

在得出  $w\_change$  值后, 按式(8)可得到第  $t+1$  代的惯性权重  $\omega(t)$  的值:

$$\omega(t+1) = \omega(t) - w\_change \quad (8)$$

## 2 T-SPSO 算法优化 PID 参数

### 2.1 PID 控制原理

PID 控制器是一种线性控制器, 单位负反馈 PID 控制器原理图如图 1 所示。图 1 所示 PID 系统的偏差信号为:

$$e(t) = r(t) - y(t) \quad (9)$$

PID 控制器是将偏差信号  $e(t)$  进行比例(P)、积分(I)、微分(D)运算后, 再将它们进行线性组合, 构成控制量  $u(t)$ , 进而对被控对象进行控制。

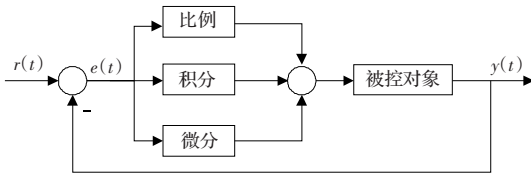


图1 单位负反馈PID控制系统原理图

PID控制器的传递函数形式可被表述为:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (10)$$

## 2.2 目标函数选取

选取目标函数目的在于使优化后控制系统的偏差趋于零,并且具有较快的响应速度和较小的超调量。目标函数采用常用的ITAE指标,即:

$$J_{ITAE} = \int_0^{\infty} t \cdot |e(t)| dt \quad (11)$$

## 2.3 微粒初始化方法

微粒群算法的初始范围对改善算法的收敛速度及收敛性能有着重要的作用。本文采用的初始化范围是以采用ZN法获取的参数为中心,向左右两侧拓展而形成的范围。

## 2.4 $CBPE_{max}$ 及 $CBPE_{min}$ 值的确定方法

$CBPE_{max}$  及  $CBPE_{min}$  的值体现的是对优化后控制系统的最佳性能期望及最大性能期望,决定着算法的收敛阶段划分,因此其取值对 T-SPSO 优化算法是非常重要的。以采用 ZN 法得到 PID 参数所对应的 ITAE 指标作为  $CBPE_{max}$  值,而  $CBPE_{min}$  体现的是期望的最优值,可依据工程的实际需求得出,也可按照式(12)整定。其中  $\alpha$  是一个大于零而小于 1 的比例系数。

$$CBPE_{min} = \alpha \times CBPE_{max} \quad (0 < \alpha < 1) \quad (12)$$

## 2.5 T-SPSO 算法优化 PID 参数流程

T-SPSO 算法优化 PID 控制器参数流程如图 2 所示,其步骤如下。

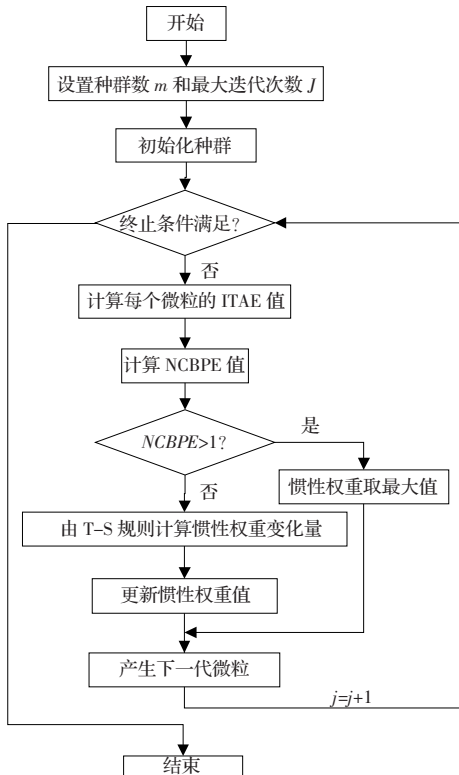


图2 T-SPSO 算法流程图

**步骤 1** 初始化算法的相关参数:以 ZN 法获取的参数为中心初始化微粒的位置;初始化最大搜索速度  $v_{max}$ ;在最大速度范围内初始化微粒的速度;初始化每个微粒的  $p_{best}$  为其初始位置, $p_{best}$  中最好的值初始化为  $g_{best}$  的初始值;初始化惯性权重取值;初始化  $CBPE_{max}$ 、 $CBPE_{min}$  值;

**步骤 2** 根据 PID 控制系统及式(11)评价每个微粒的适应值,并根据式(7)、T-S 模糊规则及式(8)计算微粒的  $NCBPE$ 、 $\omega_{change}$  和  $\omega$  值;

**步骤 3** 根据方程(1)、(2)、(3)更新当前微粒的速度、位置;

**步骤 4** 根据方程(4)、(5)将当前每个微粒的适应值与其所经历过的最好位置  $p_{best}$  的适应值进行比较,如果优于  $p_{best}$ ,则将此微粒位置更新为当前的最好位置  $p_{best}$ ;

**步骤 5** 对每个微粒,将其适应值与群体所经历过的最好位置  $g_{best}$  的适应值进行比较,如果优于  $g_{best}$ ,则将其作为群体最优位置,并重新设置  $g_{best}$  的索引号;

**步骤 6** 检查终止条件(满足最大迭代次数或所有微粒均已收敛),若条件满足,终止迭代,否则返回步骤 2。

## 3 仿真实例

本文控制对象传递函数如下:

$$\text{对象 1 } G(s) = \frac{1}{(s+2)(0.2s+1)^2}$$

$$\text{对象 2 } G(s) = \frac{e^{-0.5}}{(s+1)^2}$$

PID 控制器中需要优化的参数有  $K_p$ 、 $K_i$ 、 $K_d$  三个,故取  $N=3$ ;选择种群数  $m=100$ ;最大迭代次数 200 代。依据 ZN 法得到对象 1 及对象 2 的  $K_p$ 、 $K_i$ 、 $K_d$  参数如表 3 所示。对象 1 与对象 2 的  $CBPE_{max}$ 、 $CBPE_{min}$  取值见表 2,其中  $J_{ZN}$  为 ZN 法所对应的 ITAE 指标。经过大量仿真实验,证实当  $\alpha$  在 0.15~0.3 内取值时算法收敛性较好。

表 2 T-SPO 算法  $CBPE_{max}$ 、 $CBPE_{min}$  取值

控制对象	$J_{ZN}$	$\alpha$	$CBPE_{max}$	$CBPE_{min}$
对象 1	0.365	0.15	0.365	0.055
对象 2	1.726	0.20	1.726	0.345

图 3 和图 4 分别为 T-SPSO 算法优化对象 1 和对象 2 的优化过程曲线,其中 LPSO 算法为文献[6]提出的线性递减惯性权重 PSO 算法。由于 LPSO 算法并没有很好解决 PSO 算法可能存在的“早熟”问题,因而算法并没有收敛到更优值。T-SPSO 算法中  $CBPE_{max}$ 、 $CBPE_{min}$  值决定着算法收敛阶段,只有当 ITAE 指标接近  $CBPE_{min}$  值时,算法才会使惯性权重取值小,此时算法才会进入收敛阶段,避免了算法的过早收敛。

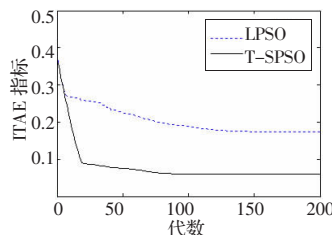


图3 对象 1 的优化过程

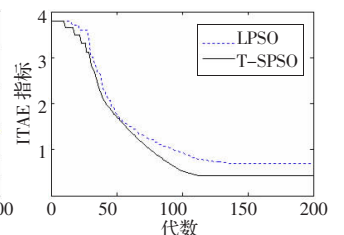


图4 对象 2 的优化过程

图 5 和图 6 分别为对象 1 和对象 2 的单位阶跃响应曲线。基于 LPSO 算法及 T-SPO 算法设计的 PID 控制器参数如表 3

所示。表4是ZN法、LPSO算法、T-SPSO算法的时域控制性能指标对比,其中调节时间 $t_s$ 是基于2%偏差得出的。

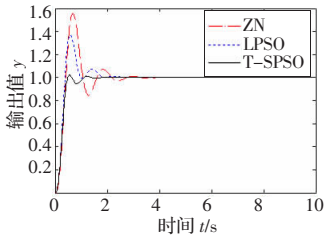


图5 对象1的单位阶跃响应

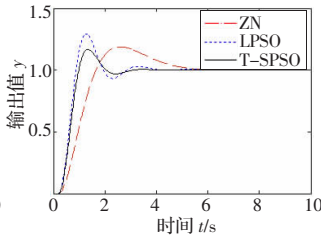


图6 对象2的单位阶跃响应

表3 PID控制器的整定参数

PID 参数	对象1			对象2		
	$K_p$	$K_i$	$K_d$	$K_p$	$K_i$	$K_d$
ZN	11.74	25.06	1.38	2.81	1.72	1.15
LPSO	16.40	23.98	2.25	7.95	2.45	1.89
T-SPSO	17.07	7.16	3.15	7.05	1.83	2.01

表4 PID控制器的性能指标

性能 指标	对象1			对象2		
	$\sigma\%$	$t_r$	$t_s$	$\sigma\%$	$t_r$	$t_s$
ZN	55.59	0.41	2.56	18.89	1.66	5.08
LPSO	37.34	0.38	1.72	29.18	0.88	3.65
T-SPSO	2.23	0.49	1.02	16.75	0.98	2.72

分析以上图表中可见,采用ZN法设计PID控制器其性能指标并不理想。采用基于线性递减惯性权重策略设计的PID控制器,其性能较之ZN法有了很大提高,但尚未达到最优。T-SPSO算法较好解决了PSO算法的早熟问题,因而可以更好地优化控制器的参数。

## 4 结论

将T-S模糊模型与PSO算法相结合,提出了一种T-SPSO算法,并将该算法应用于PID参数整定。算法基于设定的 $CBPE_{max}$ 、 $CBPE_{min}$ 值和 $CBPE$ 值确定种群当前的收敛阶段,通过T-S模糊规则动态自适应更新惯性权重取值,若当前种群的

$CBPE$ 值远小于所期望(设定)的性能指标,则惯性权重值取大,加大搜索步长,避免陷入局部最优;若当前种群的 $CBPE$ 值已接近设定的 $CBPE_{min}$ 值,则惯性权重取小,便于种群的收敛。这样,通过对 $CBPE_{max}$ 、 $CBPE_{min}$ 值的设定,使算法的收敛行为得到掌控,解决了PSO算法存在的早熟问题,改善了算法的收敛性。此外,在实际工程应用时, $CBPE_{max}$ 、 $CBPE_{min}$ 值可分别体现为对控制性能指标的最小需求目标和最大需求目标,实现了算法收敛与工程需求的结合,避免了不可期望的盲目收敛行为。仿真分析证实了本文所提出的T-SPSO算法的有效性和所设计PID控制器的优越性。

## 参考文献:

- [1] 陈星,李东海.基于遗传算法的分布参数对象PID控制器设计[J].清华大学学报:自然科学版,2007,47(8):1356-1360.
- [2] 尹宏鹏,柴毅.基于蚁群算法的PID控制参数优化[J].计算机工程与应用,2007,43(17):4-7.
- [3] 汪书苹,赵争鸣,带修正因子模糊PID控制的PMSM交流伺服系统[J].清华大学学报:自然科学版,2007,47(1):9-12.
- [4] 付华,冯爱伟,徐耀松.基于单神经元控制器的异步电动机矢量控制[J].中国电机工程学报,2006,26(1):127-131.
- [5] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proc IEEE Int Conf on Neural Networks, Perth, 1995:1942-1948.
- [6] Shi Y, Eberhart R C. Empirical study of particle swarm optimization[J]. IEEE Service Center, 1999:1945-1950.
- [7] 王介生,王金城,王伟.基于粒子群算法的PID控制器参数自整定[J].控制与决策,2005,20(1):73-76.
- [8] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization[J]. IEEE Service Center, 2001.
- [9] Elkan C. The paradoxical success of fuzzy logic[J]. IEEE Expert, 1994:3-8.
- [10] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control[J]. IEEE Trans on System Man, and Cybernetics, 1985, 15(1):116-132.

(上接204页)

- [2] Wake A S, Miller P R, Moxon P, et al. Modular avionics operating system-software concept[J]. Microprocessors and Microsystems, 1997(21):63-48.
- [3] Cook A, Hunt K J R. ARINC 653-Achieving software re-use[J]. Microprocessors and Microsystems, 1997(20):479-483.
- [4] Wika K J. Safety kernel enforcement of software safety policies[D]. Charlottesville, VA: Department of Computer Science, University of Virginia, 1995.
- [5] Kim D, Lee Y H, Younis M. Software architecture supporting integrated real-time systems[J]. The Journal of Systems and Software, 2003(65):71-86.
- [6] Glamastra. The design of operating systems supporting temporal iso-

lation[D]. Pisa, Italy: ReTis Laboratory, Scuola Superiore S Anna, 2000.

- [7] Liu C L, Layland J W. Scheduling algorithms for multiprogramming in a hard real-time environment[J]. Journal of ACM, 1973, 20(1):46-61.
- [8] Audsley N, Burns A, Richardson M, et al. Hard real-time scheduling: The deadline monotonic approach[C]//The 8th IEEE Workshop on Real-Time Operating Systems and Software, Atlanta, CA, USA, 1991.
- [9] Lehoczky J, Liu S, Ding Y. The rate-monotonic scheduling algorithm: Exact characteristics and average case behavior[C]//IEEE Real-Time Systems Symposium, Los Alamitos, CA, 1989.