第 28 卷第 2 期
2011 年 2 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 28 No. 2
Feb. 2011

文章编号: 1000−8152(2011)02−0256−05

一种状态自动划分的模糊小脑模型关节控制器值函数拟合方法

闵华清[1], 曾嘉安[2], 罗荣华[2], 朱金辉[2]

(1. 华南理工大学 软件学院, 广东 广州 510006; 2. 华南理工大学 计算机科学与技术学院, 广东 广州 510006)

摘要: 在庞大离散状态空间或连续状态空间中, 强化学习(RL)需要进行值函数拟合以寻找最优策略. 但函数拟合器的结构往往由设计者预先设定, 在学习过程中不能动态调整缺乏自适应性. 为了自动构建函数拟合器的结构, 提出一种可以进行状态自动划分的模糊小脑模型关节控制(FCMAC)值函数拟合方法. 该方法利用Bellman误差的变化趋势实现状态自动划分, 并且探讨了两种选择划分区域的机制. 汽车爬坡问题和机器人足球仿真平台中的实验结果表明新算法能有效拟合值函数, 而且利用所提出的函数拟合器智能体可以进行有效的强化学习.

关键词: 强化学习; 值函数; 状态自动划分; 模糊小脑关节模型控制器

中图分类号: TP181　　**文献标识码:** A

Fuzzy cerebellar model arithmetic controller with automatic state partition for value function approximation

MIN Hua-qing[1], ZENG Jia-an[2], LUO Rong-hua[2], ZHU Jin-hui[2]

(1. School of Software Engineering, South China University of Technology, Guangzhou Guangdong 510006, China;
2. School of Computer Science and Technology, South China University of Technology, Guangzhou Guangdong 510006, China)

Abstract: In continuous-state space or large discrete-state space, the reinforcement learning (RL) uses function approximation approaches to represent the value function in seeking the optimal policy. However the structures of function approximators which will greatly influence the learning performance are often designed in advance. To generate the structure of function approximator automatically, a novel function approximator called the automatic state-partition-based fuzzy cerebellar model arithmetic controller (ASP-FCMAC) is proposed. In ASP-FCMAC, the variation tendency of Bellman error is used to determine the best time to perform state partition and two mechanisms are also discussed for determining which state should be partitioned at each time step. Experimental results in solving mountain car problem and RoboCup Keepaway problem demonstrate that ASP-FCMAC can automatically generate the structure of FCMAC for effective reinforcement learning.

Key words: reinforcement learning; value function; automatic state partition; fuzzy CMAC

1 引言(Introduction)

强化学习(reinforcement learning, RL)中智能体通常采用值函数迭代的方法来寻找最优策略[1], 以使长期回报最大化, 而这需要在庞大的状态空间中进行函数拟合. 小脑模型关节控制器(cerebellar model arithmetic controller, CMAC)也被称为瓦片编码(tile coding), 是一种主要的函数拟合方法. 但利用CMAC进行函数拟合需要预先设定结构与参数, 而CMAC对参数极其敏感, 为复杂问题设计性能良好的CMAC非常困难[2]. 最近很多研究致力于自动生成函数拟合器结构, Whiteson等人提出状态解析度(tile number)自动划分[3], 并给出值函数标准和策略标准2种划分方式; Sherstov等人提出泛化参数的自适应控制[4]; Lanzi等人利用进化方法调整CMAC结构[5]. 此外, Mahadevan等人通过分析状态空间的拓扑结构以生成函数拟合器的结构[6]; Keller等人利用邻近成分分析的方法创建基函数, 以自动生成函数拟合器的结构[2].

在上述研究基础上, 本文提出一种状态自动划分(automatic state partition, ASP)的模糊CMAC值函数拟合方法(ASP-FCMAC). ASP-FCMAC通过引入模糊隶属度函数, 减少CMAC所存储的训练权值数目, 不仅可以减轻计算负担, 而且无需指定泛化参数[7], 同时利用贝尔曼(Bellman)误差的变化趋势选择划分时机, 自动划分状态空间, 从而自动生成FCMAC(fuzzy cerebellar model arithmetic controller) 结构.

收稿日期: 2008−11−10; 收修改稿日期: 2010−5−25.

基金项目: 国家自然科学基金资助项目(61005061); 广东省科技计划资助项目(2009A040300008); 广州市科技计划资助项目(2009KP008); 广东省科技计划资助项目(2010B010600016).

2 CMAC和模糊CMAC(CMAC and fuzzy CMAC)

2.1 背景(Background)

强化学习可以表示为马尔科夫决策过程(Markov decision process, MDP)或者半马尔科夫决策过程(semi-Markov decision process, SMDP). 这里仅给出MDP的数学表达, 但本文的方法同样可以应用于SMDP. MDP由4元组$< S, A, P, R >$构成, 其中: S表示状态空间; A表示动作集合; $P : S \times A \to S$是转移函数, 定义了在某一状态下执行某一动作后转移到另一状态的概率; $R : S \times A \to R$定义了智能体执行某动作后的立即回报. RL的目标是最大化以下值函数:

$$V^{\pi}(s) = E_{\pi}\{\sum_{t=0}^{\infty}\gamma^{t}R(s_t, a_t)|s_0 = s\}, \tag{1}$$

其中: s_t和a_t分别表示t时刻的状态和动作, $\gamma \in (0, 1)$是折扣因子, $R(s_t, a_t)$是值函数, 描述了累积回报. RL通常采用迭代方式计算最优值函数, 在迭代过程中根据当前动作的累积回报与估计值的差(Bellman误差)更新值函数. Bellman误差表示如下:

$$\Delta V(s) = \max_{a}\{R(s, a) + \gamma V(P(s, a))\} - V(s). \tag{2}$$

在高维离散状态空间或连续状态空间中, 上述值函数不能被精确表示, 需要进行函数拟合.

2.2 CMAC函数拟合和模糊CMAC(Function approximation with CMAC and fuzzy CMAC)

CMAC是一种广泛使用的线性函数拟合方法[1], 其状态空间被一系列“盖瓦”(tiling)所覆盖; 每个tiling具有相同的结构, 相邻tiling之间存在微小的位移. tiling由若干个带权值的“瓦片”(tile)组成. 泛化过程中, 当前状态会激活每个tiling中某个tile, 对所有tile的权值进行求和, 就可得到当前的函数拟合值. 拟合过程可以表示为:

$$\tilde{V} = \sum_{i=1}^{n_1}\phi_i(s)w_i,$$
$$\phi_i = \begin{cases} 1, \text{ if第}i\text{个tile是激活的}, \\ 0, \text{ otherwise}, \end{cases} \tag{3}$$

其中: w_i表示第i个tile所对应的权值, ϕ_i表示第i个tile所对应的特征值, n_1是CMAC的存储容量, 即tile的总数. 在CMAC中, 值函数的更新实际上是tile权值的更新, 更新公式如下:

$$w_i \leftarrow w_i + \alpha\phi_i\Delta V(s), \tag{4}$$

其中α表示学习速率.

CMAC的特征值是一个二值函数, 表示tile是否激活. 为了达到较好的泛化水平, CMAC需要预先设定tiling数目[4]. J.Nie等人提出把模糊集引入CMAC, 把二值函数改为模糊隶属度函数[8], 隶属度表示tile的激活程度, 这样可以把tiling数目降为1. 引入模糊集后, 函数拟合过程可以表示为

$$\tilde{V} = \sum_{i=1}^{n_2}\eta_i(s)w_i, \tag{5}$$

其中: η_i表示模糊隶属度函数, n_2表示FCMAC的存储容量. 权值更新公式如下:

$$w_i \leftarrow w_i + \alpha\eta_i\Delta V(s). \tag{6}$$

3 基于状态自动划分的FCMAC(Automatic state partition based FCMAC)

3.1 划分时机(State partition time)

划分时机的选择将影响函数拟合的性能, 过快的状态划分, 会导致拟合结果大幅波动; 而过慢的状态划分, 则会导致学习速度缓慢. 由于Bellman误差$|\Delta V(s)|$是衡量学习算法是否收敛的标准, 其变化反映了学习算法逼近最优解的速度. 因此, 可以根据Bellman误差的变化趋势选择状态划分的时机. 假设样本容量为T, 样本范围内的平均误差可表示为:

$$|\overline{\Delta V(s)}| = \sum_{i=1}^{T}|\Delta V_i(s)|/T, \tag{7}$$

其中$|\Delta_i V(s)|$表示样本范围内的单次误差. 新划分的判断条件定义如下:

$$|\overline{\Delta V(s)}| > |\overline{\Delta V(s)}_{\text{last}}|, \tag{8}$$

其中$|\overline{\Delta V(s)}_{\text{last}}|$表示上一次的平均值. 如果式(8)成立, 则说明学习过程没有继续向最优逼近, 需要对状态进行新一轮划分. 由于参数T将影响状态划分的频率, 这里T的取值将根据当前划分区域数自适应调整:

$$T = f(p), \tag{9}$$

其中: p是当前状态变量所划分的区域数, 函数f为增函数

$$f = kp^{a}, \tag{10}$$

其中k, a为非负常数. a的取值影响着函数f的变化速率. 当$a > 1$时, 函数f递增速度较快; 当$a < 1$时, 函数f对p的变化并不敏感; 当$a = 1$时, 函数f为线性函数, 函数拟合速度快, 更适用于一些实时性要求高的系统.

3.2 划分区域(State partition region)

当智能体决定对状态进行重新划分时, 需要选择合适的划分区域. 一般而言, 状态空间中不同状态变量需要有不同细致程度的分区方案. 但在实际测试中, 某一状态变量过细的划分对函数拟合的整体效果没有太大影响. 因此, 本文不区别各状态变量, 即同时划分所有状态变量, 并采用对等的方式划分区域. 为了选择合适的划分区域, 本文将提出2种寻找划分区域的方法:

I) 划分经常访问的区域.

采用下式选择划分区域:

$$p=\max_{i=1}\{v(p_i)\},\tag{11}$$

其中$v(p_i)$表示状态变量中区域p_i的访问次数. 区域p_i访问频繁表明区域p_i对智能体的策略选择有重要影响, 提高解析度能使智能体更清晰区分不同状态的差异. 采用式(11)进行划分, 智能体将关注值函数变化较大的区域, 这类似于Whiteson等人提出的值函数标准[3]. 图1(a)给出一个划分经常访问区域的例子.

II) 划分具有最大权值和的区域.

采用下式选择划分区域:

$$p=\max_{i-1}(w_i+w_{i+1}),\tag{12}$$

其中w_i和w_j分别表示区域p_i两端所对应的权值. 图1(b)给出一个划分最大权值区域的例子.

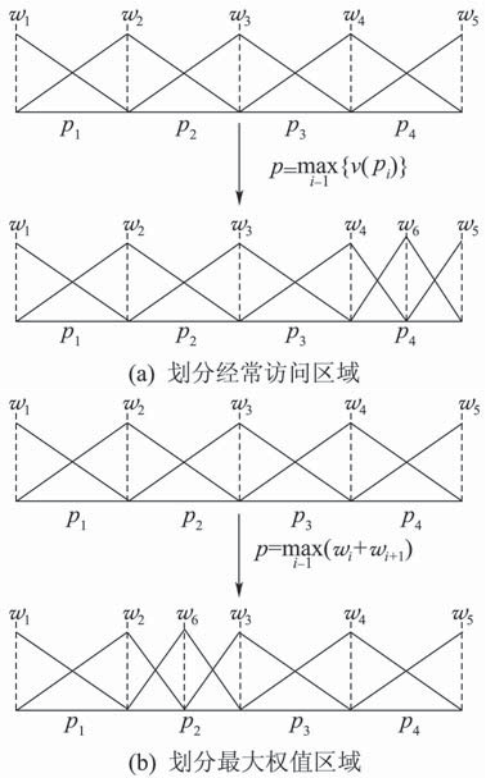


图 1 2种不同的状态划分方法

Fig. 1 Two methods for state partition

进行新划分时, 首先计算区域p_i的权值之和w_i+w_j, 并划分w_i+w_j取值最大的区域. 在采用FCMAC后$|\Delta V(s)|$表示为

$$|\Delta V(s)|=R(s,a)+\sum_{i=1}^{n_2}\eta_i w_i-V(s).\tag{13}$$

由上式可知, w_i+w_j将影响迭代误差.

每次新划分将产生一个新权值w_k和2个新区域. 如果w_k的初值过大, 将使w_i+w_k和w_k+w_j过大, 导致算法总是在w_k附近区域划分. 为了避免上述情况, 新权值初值应接近或等于零值.

3.3 ASP-FCMAC算法(ASP-FCMAC algorithm)

算法1给出ASP-FCMAC的清晰描述. η为特征值向量, GetFeatureValue函数返回当前状态所对应特征值的模糊值. 函数ShouldPartition以3.1节的方法判断是否做新一次划分, 函数Partition采用3.2节的方法I)或方法II)可实现状态空间的自动划分.

算法 1 ASP-FCMAC 算法.

初始化;

for t =1,2,$\cdots$, do

输入当前状态s_t;

$\Delta V(s)=\max_a[R(s,a)+\gamma V(P(s,a))]-V(s)$;

$\eta\leftarrow\text{GetFeatureValue}(s_t)$;

for i=1 to n_2 do

$w_i\leftarrow w_i+\alpha\eta_i\Delta V(s)$;

end for

if ShouldPartition($|\Delta V(s)|$) then

Partition(S);

end if

end for

4 实验结果及分析(Experimental results and analysis)

4.1 汽车爬坡问题实验结果(Experiments in mountain car problem)

汽车爬坡问题(mountain car, MCar)是一个二维RL问题[1]. 一辆汽车在U字形山谷底, 目标是爬上其中一面山坡, 由于引擎动力不足, 汽车无法通过向前加速爬上山坡顶. 相反, 汽车需要通过向后加速, 爬上另一面山坡, 借助山坡高度所产生的动能往下冲, 向目标坡顶前进. 实验主要指标是汽车成功爬坡所需要的时间步, 时间步越少所采取的策略越好.

MCar问题的RL参数为: $\alpha=0.25$, $\varepsilon=0.01$, $\gamma=1$, $\lambda=0.9$, ASP-FCMAC参数为: $\beta=0.1$, $T=10p$. 本节对4种学习算法进行比较, 分别是CMAC, FCMAC, 划分经常访问区域的ASP-FCMAC(ASP-FCMAC-Visit)以及划分最大和值区域的ASP-FCMAC(ASP-FCMAC-Max). 值函数计算方式采用Sarsa(λ). FCMAC对每个状态变量均分为8个区, FCMAC, ASP-FCMAC-Visit和ASP-FCMAC-Max的模糊隶属度函数为三角形函数. CMAC的tiling数目设置为10, tile数目设置为8. 每种算法学习5000个训练步, 实验结果是8次运行的平均值.

训练次数与时间步和训练次数与误差的关系分别如图2(a)和2(b)所示. 从图2(b)可看出, FCMAC与ASP-FCMAC-Max的Bellman误差的下降速度相近, 但是ASP-FCMAC-Max的变化比较平稳, 而ASP-FCMAC-Visit的下降速度最慢. 表1给出了4种方法的时间步和标准差. 从表1可看出, 与

FCMAC相比, ASP-FCMAC-Max可以大幅度降低汽车成功爬坡的时间步, 改善了FCMAC的学习效果; 而且ASP-FCMAC-Max的学习标准差最低, 表明状态自动划分能改善FCMAC在学习过程中出现的波动.

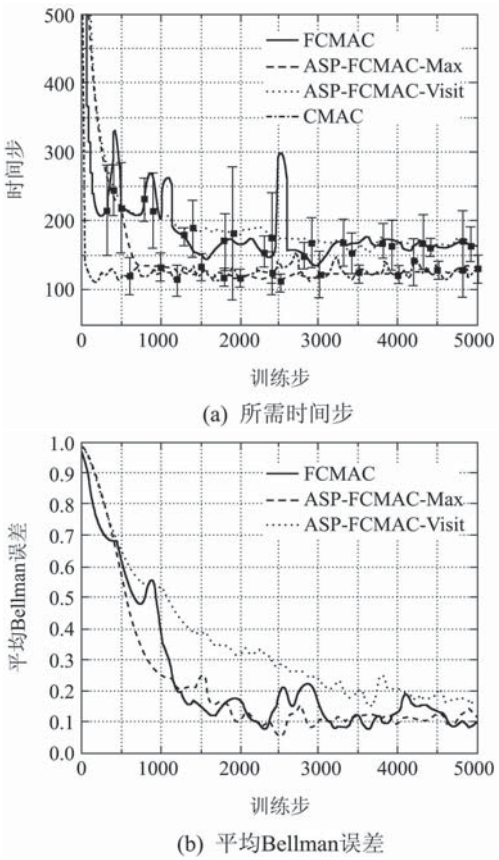


图 2 汽车爬坡问题实验结果

Fig. 2 Experimental result in mountain car problem

表 1 汽车爬坡问题实验的时间步

Table 1 Time steps in mountain car problem

函数拟合算法	汽车爬坡
CMAC	114 ± 10
FCMAC	155 ± 30
ASP-FCMAC-Visit	156 ± 23
ASP-FCMAC-Max	118 ± 5

4.2 RoboCup Keepaway实验结果(Experiments in RoboCup keepaway problem)

RoboCup机器人仿真足球平台(RoboCup soccer simulation, RCSS)是一个复杂多机器人系统(multi-agent system). 在该仿真平台中, 2支队伍在矩形区域内进行抢球对抗, 其中一方称为keeper, 有x个球员; 另一方称为taker, 有y个球员, 通常$x > y$. Keepaway是机器人足球的一个子任务, 其中keeper的任务是尽量使球处于本方的控制之下, 而taker的任务是从keeper中抢球, 3vs.2 Keepaway的场景如图3所示. 在Keepaway中keeper学习一种最优策略, 使控球时间尽可能长. 评价keeper学习效果的主要指标是平均情节持续时间(episode duration), 该时间越长keeper所采取的策略越好.

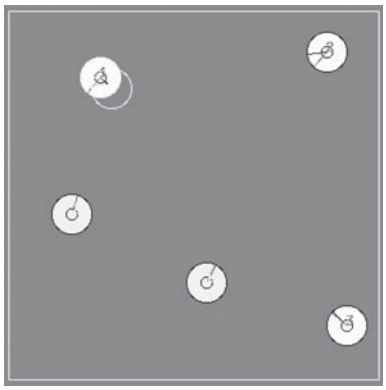
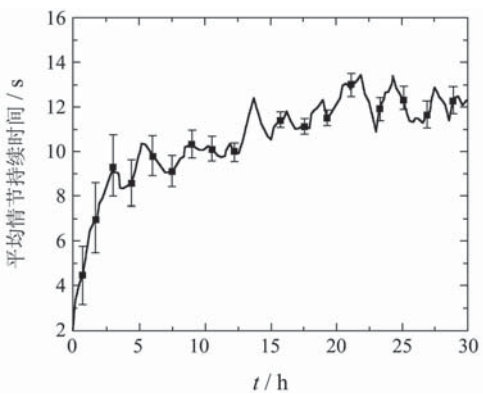


图 3 3vs.2 Keepaway 实验环境

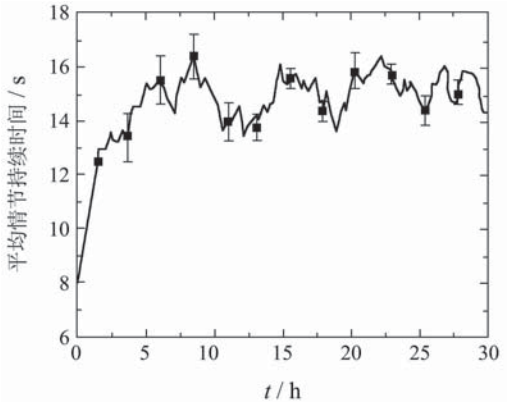
Fig. 3 Experimental environment of 3vs.2 Keepaway

本实验对CMAC, FCMAC, ASP-FCMAC-Max等3种方法进行了对比. RL的参数设置为: $\alpha = 0.03$, $\varepsilon = 0.01$, $\gamma = 1$, $\lambda = 0.65$. FCMAC的参数与CMAC的参数一样: 距离变量的区域宽度为10.0, 角度变量的区域宽度为3.0. ASP-FCMAC-Max的参数为: $\beta = 0.1$, $T = 10p$. Keeper采用Sarsa(λ)计算值函数. 与4.1节一样, FCMAC和ASP-FCMAC-Max的模糊隶属度函数为三角形函数.

实验分别在20 × 20 和30 × 30 区域中进行. 在4次实验中, ASP-FCMAC-Max的平均学习效果与训练次数的关系如图4所示. CMAC, FCMAC以及ASP-FCMAC-Max 3种方法的平均情节持续时间如表2所示. 从表2可看出, 在20 × 20的3vs.2 Keepaway和30 × 30的4vs.3 Keepaway中, ASP-FCMAC-Max的平均情节持续时间和标准差都与FCMAC的相近. 也就是说, 在Keepaway这样的高维复杂问题中, ASP-FCMAC-Max依然可以通过状态自动划分有效地拟合值函数, 获得与FCMAC相近的结果. 但ASP-FCMAC-Max不需要手工设置状态划分参数, 其自适应性更好.



(a) 3vs.2 20×20 Keepaway



(b) 4vs.3 30×30 Keepaway

图 4 RoboCup Keepaway情节持续时间

Fig. 4 Episode duration in RoboCup Keepaway

表 2 Keepaway中的情节持续时间

Table 2 Episode duration in RoboCup keepaway

函数拟合算法	3vs.2 Keepaway/s	4vs.3 Keepaway/s
CMAC	12.5 ± 0.1	14.3 ± 0.2
FCMAC	12.2 ± 0.2	14.3 ± 0.3
ASP-FCMAC-Max	12.4 ± 0.2	14.1 ± 0.3

5 结论(Conclusion)

本文提出一种状态自动划分的模糊CMAC(ASP-FCMAC), 该方法通过分析Bellman误差的变化趋势和训练权值自动划分状态, 从而自动生成FCMAC的结构. 汽车爬坡问题实验证明ASP-FCMAC能有效地拟合值函数, 状态自动划分能提高FCMAC的拟合能力, 并且能减少FCMAC在学习过程中的波动. Keepaway实验结果表明ASP-FCMAC在高维连续状态空间, 有噪声环境中, 能有效地进行函数拟合.

参考文献(References):

[1] RICHARD S S, ANDREW G B. *Reinforcement Learning: An Introduction*[M]. Cambridge, MA: Massachusettes Institute of Technology, 1998.

[2] KELLER P W, MANNOR S, PRECUP D. Automatic basis function construction for approximate dynamic programming and reinforcement learning[C] //*Proceedings of the 23rd International Conference on Machine Learning*. Pittsburgh: Association for Computing Machinery, 2006: 449 – 456.

[3] WHITESON S, TAYLOR E M, STONE P, et al. *Adaptive tile coding for value function approximation*[R]. Austin: University of Texas. 2007.

[4] SHERSTOV A A, STONE P. Function approximation via tile coding: automating parameter choice[C] //*Proceedings of Symposium on Abstraction, Reformulation, and Approximation (SARA-05)*. Berlin, Germany: Springer-Verlag, 2005: 194 – 205.

[5] LANZI P L, LOIACONO D, WILSON S W, et al. Classifier prediction based on tile coding[C] //*Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. Washington: Association for Computing Machinery, 2006: 1497 – 1504.

[6] MAHADEVAN S. Samuel meets Amarel: automating value function approximation using global state space analysis[C] //*Proceedings of American Association for Artificial Intelligence*. Pittsburgh: American Association for Artificial Intelligence, 2005: 1000 – 1005.

[7] 孙炜, 王耀南. 模糊CMAC及其在机器人轨迹跟踪控制中的应用[J]. 控制理论与应用, 2006, 23(1): 38 – 42.
(SUN Wei, WANG Yaonan. Fuzzy cerebellar model articulation controller and its application on robotic tracking control[J]. *Control Theroy & Applictions*, 2006, 23(1): 38 – 42.)

[8] NIE J, LINKENS D A. FCMAC: a fuzzified cerebellar model articulation controller with self-organizing capacity[J]. *Automatica*, 1994, 30(4): 655 – 664.

作者简介:

闵华清 (1956—), 男, 教授, 博士生导师, 目前研究方向为智能机器人、智能软件系统, E-mail: hqmin@scut.edu.cn;

曾嘉安 (1983—), 男, 硕士研究生, 目前研究方向为智能机器人、机器学习, E-mail: zengjiaan@sina.com.cn;

罗荣华 (1975—), 男, 博士, 副教授, 目前研究方向为智能机器人、机器人视觉和机器人自主导航, E-mail: rhluo@scut.edu.cn;

朱金辉 (1977—), 男, 博士, 讲师, 目前研究方向为智能机器人软件系统, E-mail: csjhzhu@scut.edu.cn.