DOI：10.13195/j.cd.2007.05.47.duany.009
第 22 卷 第 5 期 控 制 与 决 策 2007 年 5 月
Vol. 22 No. 5 Control and Decision May 2007

文章编号：1001-0920(2007)05-0525-05

基于模糊神经网络的强化学习及其在机器人导航中的应用

段 勇，徐心和
(东北大学 信息科学与工程学院，沈阳 110004)

摘 要：研究基于行为的移动机器人控制方法. 将模糊神经网络与强化学习理论相结合，构成模糊强化系统. 它既可获取模糊规则的结论部分和模糊隶属度函数参数，也可解决连续状态空间和动作空间的强化学习问题. 将残差算法用于神经网络的学习，保证了函数逼近的快速性和收敛性. 将该系统的学习结果作为反应式自主机器人的行为控制器，有效地解决了复杂环境中的机器人导航问题

关键词：强化学习；模糊神经网络；$Q(\lambda)$学习；机器人导航
中图分类号：TP181 **文献标识码**：A

Reinforcement learning based on FNN and its application in robot navigation

DUAN Yong, XU Xin-he
(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: DUAN Yong, E-mail: duanyong0607@126.com)

Abstract: Behavior-based robot navigation is studied. The fuzzy neural network (FNN) and reinforcement learning (RL) are integrated. RL is utilized for structure identification and parameters tuning of FNN. The problem of continuous, infinite states and actions in RL is solved by using the function approximation of FNN. Furthermore, the residual algorithm is applied to the FNN learning, which guarantees the convergence and rapidity. Then, the learning results are employed to design the controller of the reactive robot system, by which the problem of navigation under complicated environment is solved effectively.

Key words: Reinforcement learning; Fuzzy neural network; $Q(\lambda)$-learning; Robot navigation

1 引 言

基于行为的机器人能直接完成从感知到行为的映射，具有快速执行性和灵活性，已成为机器人学和人工智能领域的研究热点之一. 传统的反应式机器人研究方法通常基于具体的环境模型，存在环境知识获取困难、环境模型难以建立、自适应能力差等问题. 强化学习具有不依赖于环境模型、不需要先验知识以及鲁棒性强等优点，已成为基于行为的机器人研究的一个新的方向.

强化学习(RL)是指 Agent 从环境状态到动作映射的学习，以使动作从环境中获得的累积强化信号(回报)最大. 在强化学习的实际应用中，当状态空间和动作空间连续或数量过多时，强化学习收敛速度过慢甚至难以实现. 解决这一问题的有效方法是利用函数逼近算法来逼近状态空间到动作空间的映射. 神经网络(NN)和模糊推理系统(FIS)具有广泛的逼近特性，可实现从输入到输出的任意非线性映射[1]. 近年来，一些多层前馈神经网络已用于实现强化学习算法[2-4]，基于 FIS 的 Q 学习算法[5]也已提出. 神经网络具有容错能力强、自适应学习等优点，但它不能很好地利用经验知识，使得网络学习时间较长，也较难收敛到全局极值. FIS 则能充分利用先验知识，其推理方式也符合人类的思维模式，但它的自学习能力和自适应能力较差[1].

模糊神经网络(FNN)将 FIS 与 NN 相结合，具有二者的优点，目前已广泛应用于求解具有不确定性和非线性的控制问题. FNN 具有广泛的函数逼近特性，用它实现 RL 能有效解决状态空间过大时算

收稿日期：2006-03-20；修回日期：2006-04-28.
基金项目：国家自然科学基金项目(60475036).
作者简介：段勇(1978—)，男，沈阳人，博士生，从事智能机器人、机器学习的研究；徐心和(1940-)，男，河北山海关人，教授，博士生导师，从事智能机器人、模式识别等研究.

法难以收敛等问题,并可输出连续的动作. RL 可在没有教师信号指导的情况下,构建 FNN 模糊规则的结论部分和调整模糊隶属度函数的参数.

本文将 $Q(\lambda)$强化学习算法与 FNN 相结合,构成 FNN–$Q(\lambda)$学习系统. 采用 Bellman 残差算法[6]实现 FNN 的学习. 该算法综合了直接梯度下降算法和残差梯度算法的优点,从而保证了在输入状态有限的条件下,函数逼近系统的收敛性和稳定性. 将该算法应用于移动机器人导航. 对复杂导航任务进行分解,分别利用 FNN–$Q(\lambda)$系统学习两种行为控制器,通过两种控制器的自适应切换,使机器人有效地摆脱凹型障碍物陷阱并接近目标. 实验研究表明本文算法是有效的,并具有较强的适应性和鲁棒性,可以完成复杂环境的机器人导航任务.

2 基于 FNN 的 $Q(\lambda)$学习

2.1 基本 $Q(\lambda)$学习算法

在马尔可夫决策过程(MDP)中,Agent 所在的环境描述为状态集合 $S = \{s_i \mid s_i \in S\}$,它可执行的动作集合表示为 $A = \{a^i \mid a^i \in A\}$. Agent 在状态 s_t 下,选择动作 a_t 并执行. 此时状态转移到 s_{t+1},并从环境得到强化信号 r_t. 强化学习的任务是得到一个控制策略 $\pi: S \to A$,使状态–动作序列的累积回报最大.

Q 学习是一种重要的强化学习算法[2,7],它利用函数 $Q(x,a)$ 来表达与状态相对应的各个动作的评估. Q 学习算法的基本形式为[8]

$$Q(s_t,a_t) = Q(s_t,a_t) + \eta_t[r_t + \gamma \max_a Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)]. \quad (1)$$

其中:η_t 为学习率,γ 为折扣因子. 式(1)使用下一状态的估计来更新 Q 函数,称为一步 Q 学习[8,9].

将 TD(λ)的思想引入 Q 学习过程,形成一种增量式多步 Q 学习,称为 $Q(\lambda)$ 学习[9]. 这种方法首先根据标准的一步 Q 学习来更新当前的 Q 函数,然后使用贪婪策略的瞬时差分再次更新 Q 值. 基于以上特点,$Q(\lambda)$ 算法的收敛速度比单步 Q 学习更快,在很多情况下比单步 Q 学习更有效[8,9]. 在每一时刻,所有参数根据多步瞬时差分误差来更新. 多步瞬时差分由相应的资格迹 $e_t(s,a)$ 来实现[2,8]. 令

$$\xi_t = r_t + \gamma \max Q_{t+1} - \max Q_t, \quad (2)$$

$$\zeta_t = r_t + \gamma \max Q_{t+1} - Q_t. \quad (3)$$

$Q(\lambda)$ 学习算法如下:

如果 $s = s_t, a = a_t$,则

$$Q(s_t,a_t) = Q(s_t,a_t) + \eta_t[\zeta_t + \xi e_t(s_t,a_t)]; \quad (4)$$

否则

$$Q(s_t,a_t) = Q(s_t,a_t) + \eta_t \xi e_t(s_t,a_t). \quad (5)$$

2.2 FNN–$Q(\lambda)$ 学习系统结构

FNN–$Q(\lambda)$ 系统结构如图 1 所示,强化学习的目的是进行 FNN 的结构辨识和参数整定. 即通过 $Q(\lambda)$ 学习在给定规则前件的条件下,确定模糊规则的结论部分,并对模糊隶属度函数的相关参数进行调整,以提高系统的性能.

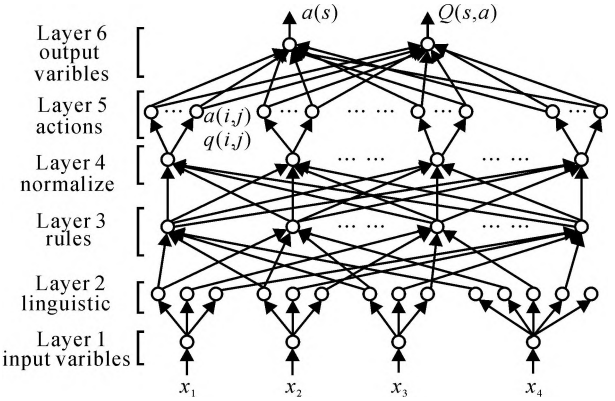

图 1 FNN–$Q(\lambda)$ 系统结构

使用 FNN 实现 $Q(\lambda)$ 学习的方式是将强化学习的状态矢量作为 FNN 的输入变量,模糊规则的输出部分为强化学习的动作空间,利用 FNN 广泛的函数逼近特性来实现状态到动作的映射. 由 $Q(\lambda)$ 学习从环境中获得的强化信号构成 FNN 输出的误差代价函数,通过误差的反向传播来确定模糊规则和调整模糊隶属度函数参数.

FNN–$Q(\lambda)$ 系统的第 1 层为输入层,它将状态矢量直接传送到下一层.

第 2 层为语言变量层,其中每个节点代表一个语言变量. 该层的作用是计算输入状态分量的隶属度. 这里模糊隶属度函数采用高斯函数,则与输入变量 x_i 相关的第 j 个节点的输出为

$$\mu_i^j = \exp[f_{ij}^{(2)}] = \exp\left[\frac{-(x_i - c_{ij})^2}{\delta_j^2}\right]. \quad (6)$$

其中:c_{ij} 和 δ_{ij} 分别为高斯函数的均值和方差,μ 为模糊隶属度.

第 3 层为规则层,该层的每个节点代表一条模糊规则,用来计算每条模糊规则前件的适应度 α_j. 即

$$f_j^{(3)} = \alpha_j = \min\{\mu_1^{j_1}, \mu_2^{j_2}, \cdots\}. \quad (7)$$

第 4 层为归一化层,用于实现第 3 层输出的归一化计算. 该层节点的输出为

$$f_j^{(4)} = \bar{\alpha}_j = \alpha_j / \sum_{i=1}^{m} \alpha_i. \quad (8)$$

第 5 层为动作选择层. 与每条模糊规则前件相匹配的可能动作作为该规则的结论部分,模糊规则表示为如下形式[5]:

R_j: If s is F^j Then y is $a(j,1)$ with $q(j,1)$,

$$\text{Or } y \text{ is } a(j,i) \text{ with } q(j,i),$$
$$\vdots$$
$$\text{Or } y \text{ is } a(j,l) \text{ wity } q(j,l). \quad (9)$$

其中 $a(j,i)$ 和 $q(j,i)$ 分别表示状态 $\boldsymbol{s}$ 的可能动作及相应的评估值. 第 4 层的每个规则节点对应于第 5 层多个动作节点, 相应节点的连接权表示相应动作的激活程度. 在强化学习过程中, 采用 EEP 搜索策略激活后件动作 $a(j,i^*)$ 作为第 j 条规则的结论.

第 6 层为输出层. 该层节点与第 5 层所有动作节点相连, 但每次学习只有被激活的局部动作节点才有效. 其输出值为作用于环境的连续动作 $a(\boldsymbol{s})$ 及相应的评价值 $Q(\boldsymbol{s},a)$, 动作由所有模糊规则的局部激活动作解模糊得到, 输出动作的评价值 $Q(\boldsymbol{s},a)$ 由局部激活动作对应的评估值 $q(j,i^*)$ 解模糊得到. 采用零阶 T-S 模糊推理系统模型, 计算方法如下:

$$a(\boldsymbol{s}) = \sum_{j=1}^{N} \varphi_j(\boldsymbol{s}) \times a(j,i^*), \quad (10)$$

$$Q(\boldsymbol{s},a) = \sum_{j=1}^{N} \varphi_j(\boldsymbol{s}) \times q(j,i^*). \quad (11)$$

2.3　基于残差的学习算法

FNN-$Q(\lambda)$ 采用两阶段混合学习算法: 第 1 阶段学习最优策略, 更新 FNN 第 5 层每条模糊规则候选动作对应的评估值 $q(j,i)$, 目的是确定模糊规则的后件; 第 2 阶段调整模糊隶属度函数的参数.

在神经网络的学习中, 计算值函数逼近误差的梯度是算法的关键. 通常直接梯度算法具有较快的学习速度, 但不能保证收敛性; 残差梯度算法可以保证收敛性, 但收敛速度过慢. 对比以上两种算法, 残差算法具有更好的收敛性和泛化性能, 并能使学习过程快速稳定地进行[6]. 以瞬时差分的 Bellman 均方残差作为误差代价函数, 即

$$E_t = \frac{1}{2}\sum_{s}[r_t + \gamma \max_{a\in A} Q(\boldsymbol{s}_{t+1}, a_{t+1}) - Q(\boldsymbol{s}_t, a_t)]^2. \quad (12)$$

残差算法的权值增量迭代公式为

$$\Delta w = -\beta \partial E/\partial w = -\beta \delta_t [\phi\gamma \frac{\partial Q(\boldsymbol{s}_{t+1}, a_{t+1})}{\partial w} - \frac{\partial Q(\boldsymbol{s}_t, a_t)}{\partial w}]. \quad (13)$$

其中

$$\delta_t = -\partial E_t/\partial Q(\boldsymbol{s}_t) = r_t + \gamma \max_{a\in A} Q(\boldsymbol{s}_{t+1}, a_{t+1}) - Q(\boldsymbol{s}_t, a_t),$$

w 为待学习的参数, β 为学习率, ϕ 为 0 ~ 1 之间的实数, 表示残差算法接近直接梯度和残差梯度的程度. 当 $\phi = 0$ 时, 残差算法退化为直接梯度算法; 当 $\phi = 1$ 时, 算法变为残差梯度算法.

下面讨论如何混合 $Q(\lambda)$ 算法来计算 $\partial Q/\partial w$. 通过反向传播误差信号得到中间变量如下:

$$\delta_j^{(4)} = -\frac{\partial Q(\boldsymbol{s})}{\partial f_j^{(4)}} = \sum_{i=1}^{M} q(j,i),$$

$$\delta_j^{(3)} = -\partial Q(\boldsymbol{s})/\partial f_j^{(3)} = \left(\frac{1}{\sum_{k=1}^{N}\alpha_k}\right)^2 \left(\delta_j^{(4)} \sum_{l=1,\, l\neq j}^{N} \alpha_i - \sum_{m=1,\, m\neq j}^{N} \delta_m^{(4)} \alpha_i\right),$$

$$\delta_j^{(2)} = -\frac{\partial Q(\boldsymbol{s})}{\partial f_{ij}^{(2)}} = \sum_{k=1}^{N} \delta_k^{(3)} \mu_i^j.$$

因此有

$$\frac{\partial Q(\boldsymbol{s})}{\partial w} = \frac{\partial Q(\boldsymbol{s})}{\partial f_{ij}^{(2)}} \frac{f_{ij}^{(2)}}{\partial w} = -\delta_{ij}^{(2)} \frac{f_{ij}^{(2)}}{\partial w}. \quad (14)$$

利用 $Q(\lambda)$ 学习算法更新 q 值, 定义资格迹更新规则为

$$e_{t+1}(j,i) = \begin{cases} \gamma\lambda e_t(j,i) + \nabla_q Q_t, & i = i^*; \\ \gamma\lambda e_t(j,i), & \text{其他}. \end{cases} \quad (15)$$

其中: $\nabla_q Q(\boldsymbol{s},a) = \partial Q(\boldsymbol{s},a)/\partial q = \varphi_j$, λ 为资格迹学习率. 更新候选动作的 q 值为

$$q_{t+1}(j,i) = \begin{cases} q_t(j,i) + \eta_t(\zeta_t \nabla_q Q_t + \xi_t e_t), & i = i^*; \\ q_t(j,i) + \eta_t \xi_t e_t, & \text{其他}. \end{cases} \quad (16)$$

其中 ξ_t 和 ζ_t 的定义同式(2)和(3).

根据式(14)可计算出

$$\frac{\partial Q(\boldsymbol{s}_t, a_t)}{\partial c_t(i,j)} = -\delta_{ij}^{(2)} \frac{2[s_t(i) - c_t(i,j)]}{\sigma_t^2(i,j)}, \quad (17)$$

$$\frac{\partial Q(\boldsymbol{s}_t, a_t)}{\partial \sigma_t(i,j)} = -\delta_{ij}^{(2)} \frac{2[s_t(i) - c_t(i,j)]^2}{\sigma_t^3(i,j)}. \quad (18)$$

执行网络的输出动作 $a(\boldsymbol{s}_t)$ 使状态转移到 $\boldsymbol{s}_{t+1}$. 同理可计算出 $\frac{\partial Q(\boldsymbol{s}_t+1, a_t+1)}{\partial c_t(i,j)}$ 和 $\frac{\partial Q(\boldsymbol{s}_t+1, a_t+1)}{\partial \sigma_t(i,j)}$. 从而由式(13)得到高斯型模糊隶属度函数参数调整算法如下:

$$c_{t+1}(i,j) = c_t(i,j) + \beta \frac{\partial E_t}{\partial c_t(i,j)}, \quad (19)$$

$$\sigma_{t+1}(i,j) = \sigma_t(i,j) + \beta \frac{\partial E_t}{\partial \sigma_t(i,j)}. \quad (20)$$

由于第 4 层的每个规则节点与第 5 层的所有候选动作节点相连, 在学习结束后, 选择第 5 层具有最大 q 值的动作节点作为规则的后件节点, 并断开其他相连接的动作节点, 从而确定每条模糊规则的后件.

3　FNN-$Q(\lambda)$ 在机器人导航中的应用

3.1　复杂导航学习任务分解

机器人在包含凹型障碍物的复杂环境工作时, 很难依靠一种行为顺利完成导航任务. 面对凹型障碍物单纯的避障行为, 机器人会出现振荡或死循环现象, 结果使其陷入凹型陷阱而不能顺利到达目的地. 这主要是由于机器人只能通过获得的局部环境信息进行路径规划, 而不能记忆曾经走过的路径.

解决问题的关键是如何判断进入陷阱和如何从陷阱中逃逸[10, 11]. 要从凹型障碍物逃逸并成功到达目的地, 机器人所要执行的动作不仅与当前状态有关, 而且与以前的某些状态有关. 通过强化学习使机器人学习包含凹型障碍物环境的导航行为, 很难取得良好的控制效果. 为此, 本文将复杂导航任务分解成两个子任务, 分别通过 FNN-$Q(\lambda)$ 方法学习每个分解行为.

将导航任务分解成两种行为: 一种是避障且接近目标行为; 另一种是沿墙走行为. 通过 FNN-$Q(\lambda)$ 分别学习两种行为. 在导航过程中通过两种行为控制器的切换, 使机器人从凹型陷阱逃逸并到达目的地.

3.2 状态空间和动作空间

合理选择状态矢量和动作空间是实现强化学习的基础. 这里将机器人通过传感器感知的环境距离信息作为强化学习的状态矢量, 以机器人运动的线速度 v_c 和角速度 ω 作为动作空间. 将机器人配置的测距传感器覆盖范围分为 3 组, 分别用于获取机器人前方、左侧和右侧障碍物的距离信息. 每组中分别取各传感器测得距离的最小值作为该方向障碍物的距离值, 即 $D_{\min} = \min(d_i)$. 定义目标点相对于机器人的方向角(即机器人中心和目标点的连线与机器人运动正方向的夹角) 为 θ.

3.3 机器人避障并接近目标行为的 FNN-$Q(\lambda)$ 实现

把左、前、右 3 个方向的传感器测量值 D_l, D_c, D_r 以及目标相对于机器人的方向角 T_D 作为强化学习的输入状态矢量, 其中 T_D 的范围为 $(-180^\circ, +180^\circ]$. 输出动作变量角速度 ω 和线速度 v_c 采用离散模糊集, 其参数不需调整.

当机器人通过传感器获取环境信息后, 将状态矢量输入 FNN-$Q(\lambda)$ 系统, 通过强化学习来获取模糊规则的结论部分, 并整定模糊隶属度函数参数. 把机器人控制变量角速度 ω 和线速度 v_c 的不同组合作为强化系统的动作集合, 由此建立起 FNN 的各层节点, 通过式(10) ~ (20) 更新每条模糊规则可能动作的 q 值和模糊隶属度函数参数. 学习之后, 选择规则中具有最大 q 值的动作作为规则节点的结论部分, 从而得到最终的机器人动作控制器.

机器人所要学习的动作包括避障和接近目标两种行为: 对于避障行为, 希望机器人与障碍物的距离越大越好, 机器人越接近障碍物, 越应受到惩罚(负强化信号); 机器人越远离障碍物, 越应获得奖励(正强化信号). 对于接近目标行为, 机器人接近目标应获得奖励, 远离目标应受到惩罚. 综合两种行为, 确定强化信号函数如下:

$$r_t = \begin{cases} -2, & d_t \leqslant D_S; \\ -\tau(D_A - d_t), & D_S < d_t \leqslant D_A; \\ +0.5, & d_t > D_A,\ d_{T_1} \leqslant d_{T_0}; \\ 0, & \text{其他.} \end{cases} \tag{21}$$

其中: r_t 为机器人 t 时刻获得的回报; d_t 为机器人 3 个方向测距值的最小值, 即 $d_t = \min\{D_l, D_c, D_r\}$; τ 为根据实际情况确定的比例系数; D_s 为安全阈值, 当机器人与障碍物的距离小于该值时, 认为发生碰撞; D_A 为机器人避障阈值, 在该范围内机器人可以进行避障; d_{T_0} 为前一时刻机器人与目标的距离值; d_{T_1} 为当前时刻机器人与目标的距离值.

图 2 为在仿真环境中, 通过强化学习设计的避障且接近目标行为控制器, 控制机器人从不同初始点到达目标点的行走轨迹. 在强化信号的约束下, 可保证机器人按较为优化的路径无碰撞地到达目标.

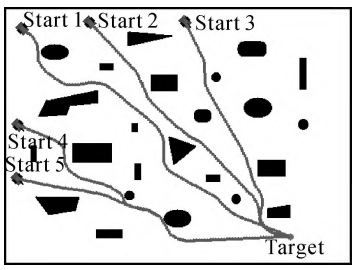

图 2 机器人避障且接近目标

3.4 机器人沿墙走行为的 FNN-$Q(\lambda)$ 实现

本文设计的行为控制器可在复杂的环境中完成导航任务, 然而凹型障碍物会对机器人形成陷阱. 图 3 中的机器人行走轨迹是在典型凹型障碍物中的循环路径. 机器人从起始点出发首先接近目标点, 然后为了躲避前方障碍物, 机器人将向右转, 此时目标相对于机器人的方向角逐渐增大. 由于限定目标方向角的范围为 $(-180^\circ, +180^\circ]$, 当机器人运动到 A 点时, 目标方向角接近 -180°. 为躲避左侧障碍物, 机器人向右转向. 根据模糊规则, 此时目标方向角将由 -180° 转变到 $+180^\circ$, 然后机器人将向左运动到 B 点. 通过类似的分析, 机器人在 B 点又会重新运动到 A 点, 重复该过程, 机器人将陷入死循环.

通过以上分析可知, A 和 B 是两个关键点, 只要

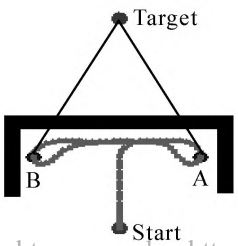

图 3 凹型陷阱分析

在机器人运动过程中检测到类似 A 和 B 的特征点, 机器人就能感知自身已处于凹型陷阱. 此时若采用相应的克服策略, 便可使机器人顺利地从陷阱中逃逸[11]. 检测特征点的方法为: $|T_{D_{t-1}} - T_{D_t}| \leqslant \zeta_{th}$. 其中 T_{D_t} 和 $T_{D_{t-1}}$ 分别表示相邻两个时刻的目标方向角, ζ_{th} 为角度阈值(这里取 $\zeta_{th} = 330°$). 也就是说, 在机器人运动过程中, 当相邻角度发生突变时, 认为机器人当前所处的位置为特征点.

为了解决以上问题, 本文设计了另外一个控制器来控制机器人沿墙走行为. 当机器人检测出特征点时, 切换到沿墙走行为控制器[10], 机器人在其控制下, 可沿着凹型障碍物的内壁走到凹型障碍物外测, 从而从陷阱中成功逃逸. 为了继续接近目标, 机器人控制器需要再次切换到接近目标行为控制器. 切换的条件是: 机器人在沿墙走控制器的作用下, 正方向转角大于 180°.

机器人的沿墙走行为是指机器人能以一定的距离沿着墙壁前进, 在运动过程中既不发生碰壁, 也不远离墙壁. 因此确定强化函数如下:

$$r_t = \begin{cases} -2.0, & d_t \leqslant D_{\min}; \\ -1.0, & d_t \geqslant D_{\max}; \\ 0.0, & D_{\min} < d_t < D_{\max}. \end{cases} \quad (22)$$

其中 $D_{\min}$ 和 $D_{\max}$ 分别为机器人与墙壁之间的最近距离和最远距离, 当机器人超出该范围时, 将受到惩罚. 图 4 为机器人沿墙走行为的仿真实验轨迹. 可见机器人可在距离墙一定宽度的范围内, 以较为光滑的轨迹沿着墙壁前进.

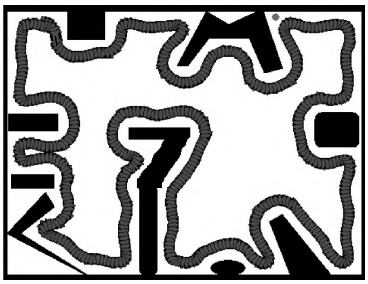

图 4　机器人沿墙走轨迹

4　实验结果及分析

使用以上方法进行含凹型障碍物复杂环境的仿真实验. 学习算法中参数选择如下: 学习率 $0.3 \leqslant \eta \leqslant 0.8$, 折扣算子 $0.9 \leqslant \gamma \leqslant 0.99$, 资格迹学习率 $0.9 \leqslant \lambda \leqslant 0.95$, q 值更新率 $0.01 \leqslant \delta \leqslant 0.3$, Epsilon 贪婪策略最优动作选择概率 $0.2 \leqslant \varepsilon \leqslant 0.5$, 隶属度函数参数学习率 $0.01 \leqslant \beta \leqslant 0.1$.

FNN 有 4 个输入节点与强化学习的状态矢量相对应. 输入变量的隶属度函数采用高斯函数, 参数的初始值通过经验选择, 也可通过聚类方法获得. FNN 动作选择层的节点由候选动作组合, 共 25 个节点, 每个候选动作的评估值 q 初始值为零. 在两种行为控制器作用下, 机器人根据环境信息自动进行控制切换, 从而顺利完成导航任务.

图 5 显示了机器人在两种行为控制器控制下, 从复杂的多凹型陷阱中逃逸并到达目的地. 机器人在到达目标点的过程中进行了多次行为切换, 切换点为 N_i 和 M_i. 从机器人的行走轨迹可以看出, 通过 FNN-$Q(\lambda)$ 系统能较好地实现机器人行为控制器的学习. 机器人能根据传感器信息感知自身所处的环境状态, 即判断何时进入凹型陷阱以及何时从陷阱中逃逸, 从而采用不同的控制策略以完成导航任务. FNN-$Q(\lambda)$ 学习过程是离线进行的, 学习过程结束后得到模糊逻辑控制器, 然后将该控制器用于反应式机器人的导航控制. 模糊逻辑控制器具有快速性的特点, 可以满足移动机器人控制对实时性的要求.

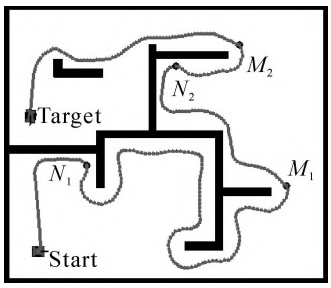

图 5　复杂环境路径规划

5　结　　论

本文将强化学习与模糊神经网络相结合, 通过 FNN 实现 $Q(\lambda)$ 算法. 利用 FNN 的函数逼近特性, 有效地解决了强化学习状态空间过大的问题, 并能输出连续的动作, 实现了连续状态空间和动作空间的强化学习任务. 本文方法可通过 $Q(\lambda)$ 算法学习 FNN 的结构和参数, 因此也可用于设计优化的模糊逻辑控制系统. 在 FNN 的学习中, 利用 Bellman 残差梯度算法来整定 FNN 的参数, 从而有效地保证了学习的收敛性和稳定性. 将 FNN-$Q(\lambda)$ 系统应用于基于行为的移动机器人导航, 分别设计了避障且接近目标行为控制器和沿墙走行为控制器. 仿真实验结果表明, 在机器人导航过程中, 通过两种行为控制器的相互切换, 能使机器人成功地完成复杂环境的导航任务.

参考文献(References)

[1] 孙增圻. 智能控制理论与技术[M]. 北京: 清华大学出版社, 2000.
(Sun Z Q. Intelligent control theory and technology [M]. Beijing: Tsinghua University Press, 2000.)

(下转第 534 页)

上的点既不会发生捕获也不会发生逃逸, 是双方中立的结局. 因此这些轨迹是划分捕获区与逃逸区的分界, 亦即寻找的界栅. 界栅存在的结果说明, 将定量与定性微分对策方法综合集成是可行的.

仿真实例给出了追踪器偏心率和两飞行器偏心率差随倒向时间变化的界栅曲线, 它们是由目标集的可用部分边界与偏心率随倒向时间变化的最大上限和最小下限轨迹组成的封闭曲线. 仿真结果表明, 在对策结束时刻, 逃逸器处于追踪器的正上方或正下方; 在对策过程中, 偏心率的控制由大到小进行, 这对轨道控制较为有利. 同时表明, 推力大小对轨道偏心率的影响较大, 这说明该方法仅适用于小推力的情况.

参考文献(References)

[1] Glizer V Y. Homicidal chauffeur game with target set in the shape of a circular angular sector: Conditions for existence of a closed barrier [J]. J of Optimization Theory and Applications, 1999, 101(3): 581–598.

[2] Isaacs R. Differential games [M]. New York: John Wiley, 1965.

[3] Turetsky V, Shinar J. Missile guidance laws based on pursuit–evasion game formulations [J]. Automatica, 2003, 39(4): 607–618.

[4] Guelman M, Shinar J, Green A. Qualitative study of a planar pursuit evasion game in the atmosphere[J]. J of Guidance, Control and Dynamics, 1990, 13(6): 1136–1142.

[5] 李登峰. 微分对策及其应用[M]. 北京: 国防工业出版社, 2004.
(Li D F. Differential games and applications [M]. Beijing: Defence Press, 2004.)

[6] Kelley H J, Cliff E M, Lutze F H. Pursuit/evasion in orbit[J]. J of the Astronautical Sciences, 1981, 29(3): 277–288.

[7] 周卿吉, 许诚, 周文松, 等. 微分对策制导律研究的现状及展望[J]. 系统工程与电子技术, 1997, 19(11): 40–45.
(Zhou Q J, Xu C, Zhou W S, et al. Status and prospects of the development of DGGL [J]. Systems Engineering and Electronics, 1997, 19(11): 40–45.)

[8] 刘敦, 赵钧. 空间飞行器动力学[M]. 哈尔滨: 哈尔滨工业大学出版社, 2003.
(Liu D, Zhao J. Dynamics of spacecraft [M]. Harbin: Harbin Institute of Technology Press, 2003.)

[9] 刘林. 航天器轨道理论[M]. 北京: 国防工业出版社, 2000.
(Liu L. Orbit theory of spacecraft [M]. Beijing: Defence Press, 2000.)

[10] 朱仁章, 林彦, 李颐黎. 论空间交会中的径向连续推力机动与 N 次推力机动[J]. 中国空间科学技术, 2004, 24(3): 21–28.
(Zhu R Z, Lin Y, Li Y L. Analysis of constant continuous radial thrust and N–thrusts in space rendezvous[J]. Chinese Space Science and Technology, 2004, 24(3): 21–28.)

(上接第 529 页)

[2] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT Press, 1998.

[3] 蒋国飞, 吴沧浦. 基于 Q 学习算法和 BP 神经网络的倒立摆控制[J]. 自动化学报, 1998, 24(5): 662–666.
(Jiang G F, Wu C P. Learning to control an inverted pendulum using Q–learning and neural networks [J]. Acta Automatica Sinica, 1998, 24(5): 662–666.)

[4] Claude F T. Neural reinforcement learning for behaviour synthesis [J]. Robotics and Autonomous Systems, 1997, 22(3/4): 251–281.

[5] Jouffe L. Fuzzy inference system learning by reinforcement methods [J]. IEEE Trans on Systems, Man and Cybernetics, 1998, 28(3): 338–355.

[6] Baird L C. Residual algorithms: Reinforcement learning with function approximation [C]. Proc of the 12nd Int Conf on Machine Learning. San Francisco, 1995: 9–12.

[7] 张汝波. 强化学习理论及应用[M]. 哈尔滨: 哈尔滨工程大学出版社, 2000.
(Zhang R B. Reinforement learning theory and applications [M]. Harbin: Harbin Engineering University Press, 2000.)

[8] Watkins C J, Dayan P. Q–learning [J]. Machine Learning, 1992, 8(3): 279–292.

[9] Peng J, Williams R J. Incremental multi–step Q–learning [C]. Proc of the 11th Int Conf on Machine Learning. New Brunswick: Morgan Kaufmann, 1995: 226–232.

[10] Lin C H, Wang L L. Intelligent collision avoidance by fuzzy logic control [J]. Robotics and Autonomous Systems, 1997, 20(1): 61–83.

[11] Xu W L, Tso K K. Sensor–based fuzzy reactive navigation of a mobile robot through local target switching [J]. IEEE Trans on Systems, Man and Cybernetics — Part C: Applications and Reviews, 1999, 29(3): 451–459.