doi:10.6041/j.issn.1000-1298.2014.07.012

移动机器人自主寻路避障启发式动态规划算法*

方　啸[1,2]　郑德忠[1]

(1. 燕山大学电气工程学院，秦皇岛 066004；2. 罗德岛大学电气工程学院，罗德岛州金士顿 02881)

摘要：用启发式动态规划算法解决移动机器人自主寻路、避障问题。提出了传感器检测环境状态的方法，以及对传感器检测信息进行归一化处理的方案。对输入、输出量以及学习系统的强化信号进行定义，设计了机器人自主学习寻路、避障的控制策略。定义了连续型强化信号，使机器人通过学习，对优先寻路还是优先避障做出决策判断。为验证启发式动态规划算法在移动机器人寻路、避障问题上的学习能力，设计了 3 种不同的寻路、避障仿真实验：同目标、不同起始点，同起始点、不同目标，和移动目标仿真实验。仿真结果表明，对于不同的寻路、避障要求，基于启发式动态规划算法的移动机器人具有良好的学习及适应能力。

关键词：移动机器人　寻路避障　启发式动态规划　强化学习

中图分类号：O221.3；TP242　　**文献标识码**：A　　**文章编号**：1000-1298(2014)07-0073-06

引言

移动机器人自主寻路、避障问题是设计一个有自主学习能力的移动机器人，使其在未知环境里能通过学习，无碰撞地从给定起点行走到达指定目标[1-4]。目前，学者们最常讨论的方法是模糊神经网络算法[2-7]。该算法通过神经网络对输入量进行模糊化处理，并利用逻辑语言对输出量进行逻辑规则定义。其控制策略是通过输出量对逻辑规则表寻址，做出相应的寻路、避障动作。然而，该算法众多逻辑规则(如文献[7]定义了 48 条逻辑规则)占据了大量的储存空间，影响了计算速度。且对输出量进行人为规则定义，在环境变化的状态下其适用性不强。

本文提出用启发式动态规划算法[8]解决移动机器人寻路、避障问题的方案。与模糊神经网络算法相比，自适应动态规划算法无需通过逻辑语言对机器人行为进行人为定义，只需给出相应的环境状态信息，机器人便可在线学习寻路、避障策略[9]。在算法设计上，本文通过归一化处理输入、输出信号，对机器人寻路、避障策略进行设计。此外，连续型强化信号的定义使机器人能在学习过程中对寻路和避障的优先选择权做出自主判断。

1　启发式动态规划算法原理

自适应动态规划算法(Adaptive dynamic programming，ADP)是解决动态规划问题较好的算法之一[10-13]。其基本思想是采用贝尔曼最优化原理，通过在线环境交互，自行学习并改善控制策略(函数逼近 Hamilton-Jacobi-Bellman(HJB)方程近似解)，进而使系统趋于最优[8,14-16]。这种在线学习方式属于强化学习(Reinforcement learning，RL)[17-18]过程。它有别于监督学习(Supervised learning，SL)：监督学习是通过比较实际输出值与期望输出值的误差数值来调节系统的控制策略；而在强化学习里，系统并不知期望的输出值，仅通过学习过程中从环境里实时反馈的强化信号(奖励(reward)值或惩罚(punish)值)来判断当前控制策略的“好”、“坏”[10]。其目的是通过自主调节控制策略，使系统趋于“好”(最优)的状态[15,19]。

在自适应动态规划算法的结构里，启发式动态规划(Heuristic dynamic programming，HDP)算法是自适应动态规划算法里一个最基本的扩展结构[8,19-20]。该算法结构由一个动作网络和一个评价网络组成(如图 1 所示)。

其中动作网络为系统提供行为策略，评价网络则对当前行为策略进行评估[8,19]。算法的具体工作原理为：①两个网络里均含有一个多层感知机(Multi-layer perception，MLP)结构的神经网络，且神经网络里均含有一个隐藏层[21-22]。②动作网络根据系统的当前状态量 $X(t)$，提供一个决策动作

收稿日期：2014-02-23　修回日期：2014-03-21

* 国家火炬计划资助项目和国家重点新产品专项基金资助项目(2009GJA20001)

作者简介：方啸，博士生，主要从事虚拟现实技术、机器学习、动态规划研究，E-mail：fangxiao220@gmail.com

通讯作者：郑德忠，教授，博士生导师，主要从事测试技术、仪器仪表研究，E-mail：1076694895@qq.com

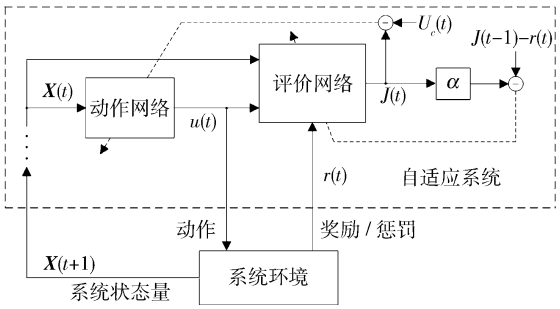

图 1 启发式动态规划算法结构图

Fig. 1 Structure of heuristic dynamic programming

$u(t)$。③该决策动作 $u(t)$ 与 $X(t)$ 一同输入到评价网络。④评价网络根据动作网络输入量以及系统环境提供的强化信号 $r(t)$，计算出代价函数 $J(t)$，用以对当前决策动作进行评估。⑤系统利用梯度下降法则依次对动作网络和评价网络里的神经网络权值进行反馈调节，最小化代价函数 $J(t)$，进而最优化控制策略 $u(t)$ [10-11,19,23]。

任意时刻 t，代价函数 J 的计算公式为

$$J(X(t),t)=\sum_{t=i}^{\infty}\alpha^{t-i}U(X(t),u(t),t) \tag{1}$$

式中 $X(t)$——系统状态量

$u(t)$——系统输出信号 U——效用函数

α——折扣因子($0<\alpha<1$)，本文取 $\alpha=0.95$

动态规划的最优原理是根据当前系统状态量 $X(t)$，寻找一个最优输出量 $u(t)$，使系统的效用(利益)最大化[10,19]。效用最大化即代价最小化，因此动作网络反馈调节原理是通过比较效用函数期望值 U_c(本文取 $U_c=0$)和代价函数 $J(t)$ 的大小，从而最小化代价函数。代价函数最小化公式为

$$J^*(X(t))=\min_{u(t)}\{U(X(t),u(t))+\alpha J^*(X(t+1))\} \tag{2}$$

动作网络的动作误差为

$$e_a(t)=J(t)-U_c \tag{3}$$

$$E_a(t)=\frac{1}{2}e_a^2(t) \tag{4}$$

为最小化动作误差 $E_a(t)$(使 $J(t)$ 趋于 U_c)，动作网络里的神经网络权值更新法则为

$$w_a(t+1)=w_a(t)+\Delta w_a(t) \tag{5}$$

$$w_a(t)=-\frac{\partial E_a(t)}{\partial w_a(t)}l_a(t) \tag{6}$$

$$\frac{\partial E_a(t)}{\partial w_a(t)}=\frac{\partial E_a(t)}{\partial J(t)}\frac{\partial J(t)}{\partial u(t)}\frac{\partial u(t)}{\partial w_a(t)} \tag{7}$$

式中 $w_a(t)$——动作网络权值矩阵

$l_a(t)$——动作网络学习速率，$l_a(t)>0$

评价网络的作用是通过强化信号 $r(t)$ 对系统当前动作 $u(t)$ 做出实时评估。根据马尔可夫决策理论[24]，任意时刻 t 的折扣奖励值无穷累加和 $R(t)$ 计算公式为

$$R(t)=\sum_{i=1}^{n}\alpha^{i-1}r(t+i) \tag{8}$$

式中 $r(t+1)$——时刻 $t+1$ 的强化信号

评价网络反馈调节策略是利用代价函数 $J(t)$ 去近似折扣奖励值无穷累加和 $R(t)$。因此，评价网络的评价误差为

$$e_c(t)=\alpha J(t)-(J(t-1)-r(t)) \tag{9}$$

$$E_c(t)=\frac{1}{2}e_c^2(t) \tag{10}$$

为最小化评价误差 $E_c(t)$，评价网络里的神经网络权值更新法则为

$$w_c(t+1)=w_c(t)+\Delta w_c(t) \tag{11}$$

$$w_c(t)=-\frac{\partial E_c(t)}{\partial w_c(t)}l_c(t) \tag{12}$$

$$\frac{\partial E_c(t)}{\partial w_c(t)}=\frac{\partial E_c(t)}{\partial J(t)}\frac{\partial J(t)}{\partial w_c(t)} \tag{13}$$

式中 $w_c(t)$——评价网络权值矩阵

$l_c(t)$——评价网络学习速率，$l_c(t)>0$

2 移动机器人寻路避障设计

2.1 移动机器传感器设置

本文移动机器人利用多个传感器检测环境状态，其传感器设置如图 2 所示。

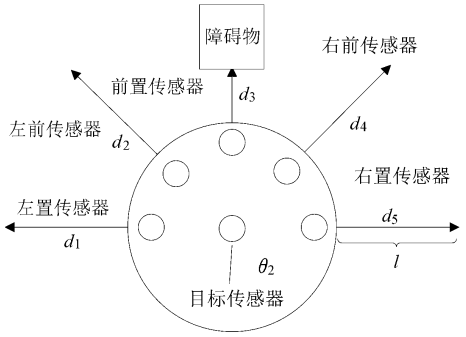

图 2 移动机器人传感器设置原理图

Fig. 2 Schematic of sensors setting for mobile robot

该移动机器人安置有 6 个传感器，其中前置、左置、右置、左前、右前传感器均为距离传感器，用于检测机器人前进途中的周边环境状态[7]。其测量范围均为 l(本文取 $l=10$ m)。d_1、d_2、d_3、d_4、d_5 为 5 个传感器测量值，即机器人与障碍物或目标的距离。

为使距离传感器在检测过程中能区分障碍物和目标，本文将障碍物和目标设置为不同形状、不同颜色，如图 3 所示障碍物：蓝色圆形；目标：红色星形，并定义了一个区分系数 γ：当检测到的物体为目标时，$\gamma=1$；当检测到的物体为障碍物时，$\gamma=-1$；当传感器未检测到任何物体时 $\gamma=0$。

除了距离传感器外,机器人还安置有一个目标传感器,该传感器用于检测目标所在方向与机器人前进方向的夹角 θ_2。在强化学习里,目标的具体位置是未知的,目标传感器的作用只是为机器人提供一个目标所在的大致方位,以便于其寻路。

2.2　移动机器人自适应学习系统设计

6 个传感器测量值将被归一化处理,作为学习系统的输入信号。归一化处理方式为:

距离传感器测量值

$$x_i = \gamma \frac{l - d_i}{l} \quad (i = 1,2,3,4,5) \tag{14}$$

5 个距离传感器的测量值 d_i 被归一化到值域 $[-1,1]$,其中 $x_i = -1$ 表示机器人撞到障碍物,$x_i = 1$ 表示机器人到达终点,$x_i = 0$ 表示未检测到障碍物或目标。

目标传感器测量值

$$\Delta\theta = \theta_1 - \theta_2 \tag{15}$$

$$\theta = \begin{cases} 1 & (\Delta\theta \geqslant 90^\circ) \\ \dfrac{\Delta\theta}{\pi} & (-90^\circ < \Delta\theta < 90^\circ) \\ -1 & (\Delta\theta \leqslant -90^\circ) \end{cases} \tag{16}$$

式中　θ_1——移动机器人前进方向与水平方向夹角

θ_2——目标所在方向与机器人前进方向的夹角

$\Delta\theta$——机器人前进方向与目标所在方向的角度偏差

θ——归一化处理后的测量值,当输入量 $\theta = 1$ 或 $\theta = -1$ 时,表示机器人已背向目标行驶

归一化处理 6 个传感器测量值后,系统状态输入量 $X(t)$ 为

$$X(t) = (x_1, x_2, x_3, x_4, x_5, \theta) \tag{17}$$

对于系统的决策动作,本文定义机器人以恒定的速度($v = 1$ m/s)在环境中行驶,且在任意时刻 t,机器人可对其前进方向进行 -10°(向左)、10°(向右)或 0°(直行)的调整。系统的决策动作利用 sgn 函数定义为

$$\mathrm{sgn}(u(t)) = \begin{cases} 10^\circ & (u(t) > 0) \\ 0^\circ & (u(t) = 0) \\ -10^\circ & (u(t) < 0) \end{cases} \tag{18}$$

当系统输出量 $u(t) > 0$ 时,机器人向左调整其前进方向 10°;$u(t) < 0$ 时,机器人向右调整前进方向 10°;$u(t) = 0$ 时,机器人将保持其前进方向直行。

强化信号的设计,首先机器人根据区分系数 γ 的正负值对检测到的是目标还是障碍物进行判断。如果检测到的是目标,则分析函数 Sen 将取 x_i 中的最大值;相反,如果检测到是障碍物,分析函数 Sen 将取 x_i 中的最小值。即

$$\mathrm{Sen} = \begin{cases} \max(x_i) & (\gamma = 1) \\ \min(x_i) & (\gamma = -1) \end{cases} \tag{19}$$

其次,就系统优先寻路还是优先避障的决策问题,本文设计一个系数 β,并定义

$$\beta = \begin{cases} 0 & (d_1 = d_2 = d_3 = d_4 = d_5 = 0) \\ 0.75 & (d_i \neq 0) \end{cases} \tag{20}$$

这样,强化信号可以设定为

$$r(t) = \begin{cases} -(1-\beta)|\theta| + \beta\mathrm{Sen} & (\text{正常行驶}) \\ -1 & (\text{发生碰撞}) \end{cases} \tag{21}$$

此连续型强化信号设定具有以下意义:①传感器未检测到障碍物或目标($\beta = 0$),系统主要任务是寻路(100%);传感器检测到障碍物($\beta = 0.75$),系统的首要任务是避障(75%),其次是寻路(25%)。②当机器人接近障碍物,由式(19)可得 Sen 将为负值,从而使 $r(t)$ 逐渐减小(得到惩罚);当机器人接近目标,Sen 将为正值,从而使 $r(t)$ 逐渐增大(得到奖励)。直至到达目标,机器人将获得最大的奖励值。当机器人到达目标,该行为决策将被看作为一次正确的行为决策,通过奖励值被机器人记住,在以后的寻路、避障时,相同的情况会被优先考虑。③当机器人发生碰撞,将得到最大的惩罚值 -1。此次行为决策将被作为一次失败的行为决策,机器人将被重置到起始状态重新开始寻路。失败的行为决策将以惩罚值的形式被机器人记住,在以后的寻路、避障时,机器人将尽量避免此错误决策再次发生。

3　仿真实验

为验证启发式动态规划算法在移动机器人寻路、避障问题上的学习能力,本文利用 Matlab 平台进行了仿真实验。其中 Matlab 仿真步长为 0.02 s,启发式动态规划算法的动作网络和评价网络随机初始化权值取值范围为 $[-1,1]$,算法其它参数设计如表 1 所示。

表 1　启发式动态规划算法参数设计

Tab. 1　Parameters of HDP design

l_a	l_c	N_a	N_c	T_a	T_c	N_h
10^{-5}	10^{-4}	100	80	0.02	0.05	6

表中　l_a——动作网络学习速率

l_c——评价网络学习速率

N_a——动作网络内循环次数

N_c——评价网络内循环次数

T_a——动作网络训练误差阈值

T_c——评价网络训练误差阈值

N_h——所有网络的隐藏层节点

仿真实验的主要目的为：随机初始化动作/评价网络的神经网络权值，机器人能否通过自主学习，避开障碍物到达目标；机器人通过学习到达目标之后，如果目标位置改变或机器人出发位置发生变化，机器人是否能继续行走到达目标；对于复杂的环境，如移动目标，机器人是否仍然能通过学习到达目标。

基于以上检验目的，本文设计了 3 种不同的仿真实验。

实验 1：随机选取动作/评价网络的神经网络初始化权值和机器人初始前进方向 $\theta_1(0)$，验证机器人是否能从起始点(10,10)无碰撞到达目标所在地(90,90)。

实验 1 的仿真结果如图 3 所示。其中蓝色边框为机器人所在环境的墙壁；红色星形为目标；蓝色圆形为障碍物；红色方块为机器人初始位置；绿色轨迹为机器人失败的碰撞轨迹；红色轨迹为机器人寻路、避障成功的轨迹。从图中可以看出，机器人在环境中多次碰撞墙壁和障碍物之后，通过调节动作/评价网络的神经网络权值，学会了如何无碰撞行驶到达目标地。

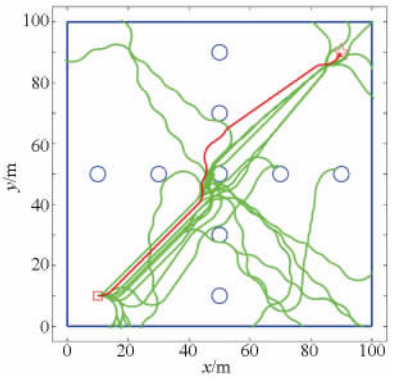

图 3 随机初始化权值路径仿真结果

Fig. 3 Simulation results of tracking trajectories using random initial weights

实验 2：环境状态变化：①同目标、不同起始位置：目标位置仍为(90,90)，机器人分别从 5 个不同点((15,15)，(90,10)，(70,30)，(30,70)，(10,90))出发。②同起始位置、不同目标：机器人起始点位置仍为(10,10)，目标分别设立于 5 个不同点((12,90)，(40,75)，(70,10)，(90,30)，(90,90))。利用实验 1 学习过的(pre-trained)神经网络权值，机器人随机选取初始前进方向 $\theta_1(0)$，验证是否仍能到达相应目标。

同目标、不同起始位置的仿真结果如图 4、5 所示。其中，图 4 为机器人行驶路径仿真结果，图 5 为与图 4 相对应的路径规划过程中代价函数的变化值。从图 4 可以看出，通过实验 1 的学习，无论初始位置如何变化，机器人均能绕过障碍物到达目标点，且碰撞次数较实验 1 大幅度减少。此外，从图 5 可以看出，在 5 个不同起点的路径规划过程中，代价函数的值均能从振荡状态(远离目标)逐渐收敛到最小值(接近目标)。

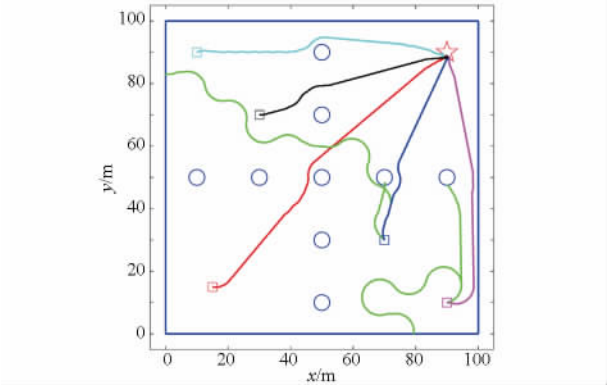

图 4 同目标、不同起始位置路径仿真结果

Fig. 4 Simulation results of tracking trajectories with same goal and different initial points using pre-trained weights

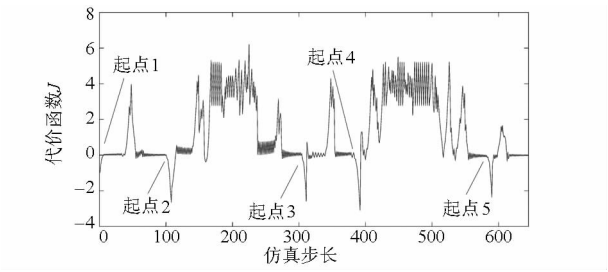

图 5 同目标、不同起始位置代价函数仿真结果

Fig. 5 Simulation results of cost function J with same goal and different initial points using pre-trained weights

同起始位置、不同目标的仿真结果如图 6、7 所示。其中，图 6 为机器人行驶路径仿真结果，图 7 为与图 6 相对应的路径规划过程中强化信号的变化值。从图 6 可以看出，通过先前实验的学习，无论目标位置如何变化，机器人均能无碰撞地行驶到目标

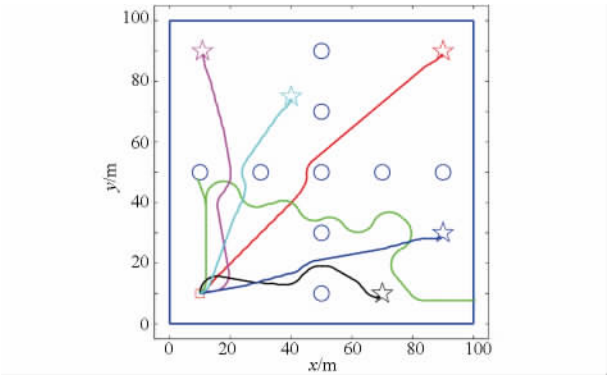

图 6 同起始位置、不同目标路径仿真结果

Fig. 6 Simulation results of tracking trajectories with same initial point and different goals using pre-trained weights

地。同时,从图 7 可以看出,对于不同的目标,只要距离传感器检测到目标,强化信号便变为正值并迅速增大,表明机器人在环境状态的“奖励”下逐渐向目标靠近。

实验 3: 目标以 0.8 m/s 的恒定速度从(70,90)到(30,90)做水平方向往复平移运动,利用实验 1 学习过的神经网络权值,验证机器人是否能从初始位置(10,10)追随上目标。

实验 3 的仿真结果如图 8 所示。从该图可以看出,通过先前实验的学习,机器人不断地调整方向追随移动目标。在经历了几次失败之后,最终机器人调转方向追上了目标。

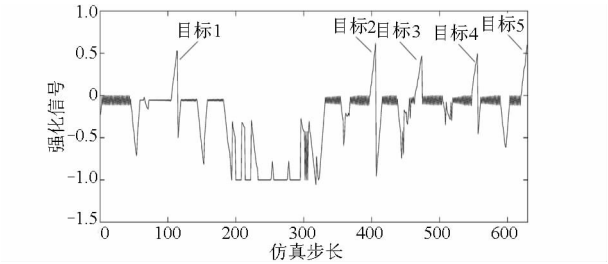

图 7　同起始位置、不同目标强化信号仿真结果

Fig. 7　Simulation results of reinforcement signal with different goal and same initial points using pre-trained weights

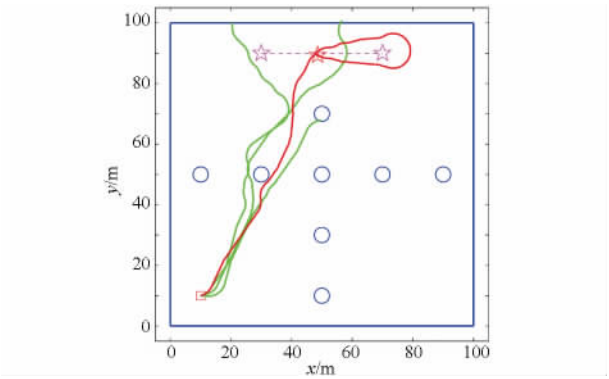

图 8　移动目标路径仿真结果

Fig. 8　Simulation results of tracking trajectories with moving goal using pre-trained weights

4　结论

(1) 设计了基于启发式动态规划算法的移动机器人寻路、避障系统。

(2) 提出了机器人寻路、避障的方法。

(3) 提出了机器人是优先寻路还是优先避障的决策策略。

(4) 仿真结果表明机器人在环境状态变化的情况下仍具有良好的学习和适应能力。

参　考　文　献

1　刘培艳. 移动机器人的控制系统研制 [D]. 西安: 西安科技大学,2008.
Liu Peiyan. Research on the control system of mobile robot [D]. Xi'an: Xi'an University of Science and Technology, 2008. (in Chinese)

2　钱夔,宋爱国,章华涛,等. 基于自适应模糊神经网络的机器人路径规划方法 [J]. 东南大学学报: 自然科学版,2012,42(4):638 – 641.
Qian Kui, Song Aiguo, Zhang Huatao, et al. Path planning for mobile robot based on adaptive fuzzy neural network [J]. Journal of Southeast University: Natural Science Edition, 2012, 42(4): 638 – 641. (in Chinese)

3　Mitic M, Miljkovic Z, Babic B, et al. Q-Leaning Framework as a solution for an obstacle avoidance problem in unknown environment [J]. Total Quality Management and Excellence, 2011, 39(2): 21 – 25.

4　姚佳. 智能小车的避障及路径规划 [D]. 南京: 东南大学,2005.

5　程志江,李剑波. 基于模糊控制的智能小车控制系统开发 [J]. 计算机应用,2008,28(12):350 – 353.
Cheng Zhijiang, Li Jianbo. Development of smart car's control system based on fuzzy control [J]. Computer Applications, 2008, 28(12): 350 – 353. (in Chinese)

6　崔超,曲伟建,吕丹,等. 未知环境下基于模糊神经网络的机器人路径规划 [J]. 北京理工大学学报,2009,29(8):686 – 689,707.
Cui Chao, Qu Weijian, Lü Dan, et al. Unknown environment based on fuzzy neural network of robot path planning [J]. Transactions of Beijing Insstitute of Technology, 2009, 29(8): 686 – 689,707. (in Chinese)

7　Zhu A, Yang S X. Neuro fuzzy-based approach to mobile robot navigation in unknown environments [J]. IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and Reviews, 2007, 37(4): 610 – 621.

8　Wang F, Zhang H, Liu D. Adaptive dynamic programming: an introduction [J]. Computational Intelligence Magazine, 2009, 4(2): 39 – 47.

9　He H, Ni Z, Zhao D. Reinforcement learning and approximate dynamic programming for feedback control [M]. Hoboken, NJ: Wiley-IEEE, 2012.

10　Si J, Barto A G, Powell W B, et al. Handbook of learning and approximate dynamic programming [M]. Paddyfield: John Wiley & Sons, 2004.

11　He H, Ni Z, Fu J. A three-network architecture for on-line learning and optimization based on adaptive dynamic programming [J]. Neurocomputing, 2012, 78(1): 3 – 13.

12　Fu J, He H, Zhou X. Adaptive learning and control for mimo system based on adaptive dynamic programming [J]. IEEE

Transactions on Neural Networks, 2011, 22(7): 1133 - 1148.
13 谢新民,丁峰. 自适应控制系统[M]. 北京: 清华大学出版社,2002.
14 Ni Z, He H. Heuristic dynamic programming with internal goal representation [J]. Soft Computing, 2013, 17(11): 2101 - 2108.
15 Ni Z, He H, Wen J. Adaptive learning in tracking control based on the dual critic network design [J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(6): 913 - 928.
16 Tang Y, He H, Ni Z, et al. Reactive power control of grid-connected wind farm based on adaptive dynamic programming [J]. Neurocomputing, 2013, 125(2): 125 - 133.
17 Sutton R S, Barto A G. Reinforcement learning: an introduction [M]. Cambridge, MA: MIT Press, 1998.
18 Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: a survey [J]. Journal of Artificial Intelligence Research, 1996, 8(5): 997 - 1007.
19 Si J, Wang Y. On-line learning control by association and reinforcement [J]. IEEE Transactions on Neural Networks, 2001, 12(2): 264 - 271.
20 Werbos P J. Approximate dynamic programming for real-time control and neural modeling [M] // White D A, Sofge D A. Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches. NY: Van Nostrand, 1992: 493 - 525.
21 Ni Z, Fang X, He H, et al. Real-time tracking on adaptive critic design with uniformly ultimately bounded condition [C] // IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL'13), IEEE Symposium Series on Computational Intelligence (SSCI), 2013: 39 - 46.
22 Fang X, He H, Ni Z, et al. Learning and control in virtual reality for machine intelligence [C] // 2012 3rd International Conference on Intelligent Control and Information Processing, 2012: 63 - 67.
23 Fang X, He H. A virtual reality learning approach for intelligent systems control and optimization [C] // 16th Intenational Conference on Cognitive and Neural Systems (ICCNS), 2012.
24 胡德文,王正志,王耀南,等. 神经网络自适应控制[M]. 长沙: 国防科技大学出版社,2006: 301 - 311.

Goal Seeking of Autonomous Mobile Robot with Obstacle Avoidance Using Heuristic Dynamic Programming

Fang Xiao[1,2] Zheng Dezhong[1]

(1. *Institute of Electrical Engineering, Yanshan University, Qinhuangdao* 066004, *China*
2. *Department of Electrical Computer and Biomedical Engineering, University of Rhode Island, Kingston, Rhode Island* 02881, *USA*)

Abstract: Heuristic dynamic programming (HDP) design for autonomous mobile robot was put forward to solve the goal seeking with obstacle avoidance problem. A method of sensor detecting was proposed, and the method for system normalizing the sensors' inputs information was discussed. The input/output signal and reinforcement signal were defined, and a self-learning strategy for robot seeking the goal with obstacle avoidance was proposed. A continuous reinforcement signal to improve the system's preferential decision between goal seeking and obstacle avoidance was designed. To verify the learning ability of our algorithm, three different simulation experiments were designed: same goal with different initial points and directions, same initial states with different goals, moving goal. The simulation results show that the HDP approach presents an effective learning ability for autonomous mobile robot on goal seeking with obstacle avoidance problem.

Key words: Mobile robot Goal seeking with obstacle avoidance Heuristic dynamic programming Reinforcement learning