

移动机器人全局路径规划的盲人摸路算法

王永兴,丁 睿,姚林泉

(苏州大学数学科学学院,江苏 苏州 215006)

摘要:路径规划是机器人控制技术中的一个重要问题。路径规划方法可分为全局法和局部法两种。首先分析了传统的全局路径规划方法的缺点,提出了移动机器人全局路径规划的一种新方法,即盲人摸路算法。然后给出了算法的详细描述过程并证明了算法的收敛性。与传统的全局路径规划方法相比,在进行路径规划之前,不需要对障碍环境进行特殊的简化处理。算法理论浅显,计算简单,信息存储量小,能够快速搜索出无碰路径。通过大量数值算例说明了算法的正确性和有效性。

关键词:移动机器人;全局路径规划;盲人摸路算法

中图分类号:TP24 **文献标识码:**B

The Blind Groping Algorithm: Global Path Planning for Mobile Robot

WANG Yong-xing, DING Rui, YAO Lin-quan

(School of Mathematical Sciences, Suzhou University, Suzhou Jiangsu 215006, China)

ABSTRACT: Path planning is a critical problem in the area of robot control technique. The types of path planning methods can be classified as global and local. The limitations of classic global path planning methods were analyzed. A new algorithm—the Blind Groping algorithm—for mobile robots collision avoidance planning is provided. A detailed account of the algorithm process is given and a proof has been presented for the convergence. Compared with the classical global path planning methods, the obstacle environment does not need to be simplified before planning. The theory of this algorithm is simple, and the calculation is easy. It only needs a small storage of information. A free path can be found out quickly. At last, some numerical experiment results have justified and validated the algorithm.

KEYWORDS: Mobile robot; Global path planning; The blind groping algorithm

1 引言

路径规划^[1]是指给定环境障碍条件,以及起始和目标的位置,要求选择一条从起始点到目标点的路径,使运动物体(机器人)能安全、无碰撞地通过所有的障碍。这种自主地躲避障碍并完成作业任务的机器人是机器人研究的一个重要内容。移动机器人的路径规划方法可分为基于地图的全局路径规划和基于传感器的局部路径规划^[2]。全局路径规划,作业环境的全部信息已知,又称静态或离线路径规划;局部路径规划,作业环境信息全部未知或部分未知,又称动态或在线路径规划^[3]。全局路径规划和局部路径规划并没有本质区别。局部路径规划只是环境考虑得更复杂一些,即环境是动态的。很多适用于全局路径规划的方法经过改进都可以用于局部路径规划;而适用于局部路径规划的方法都可以适用于全局路径规划。全局路径规划方法主要包括栅格法^[4],可视图法^[4,5],自由空间法^[4],拓朴法^[6],最优控制法,

神经网络法^[7]等。局部路径规划方法主要包括人工势场法^[4,8],模糊逻辑算法^[4,9],神经网络法^[7],遗传算法^[10]等。

以上算法各有其优缺点。栅格法中,栅格大小直接影响环境信息存储量大小和规划时间长短。栅格划分大了,环境信息存储量小,规划时间短,但分辨率下降,在密集环境下发现路径的能力减弱。栅格划分小了,环境分辨率高,在密集环境下发现路径的能力强,但环境信息存储量大,规划时间长^[11];可视图法能够搜索出最短路径,但对障碍复杂的环境,建立可视图计算量比较大;自由空间法的缺点也在于计算复杂度随环境复杂度的增加而急剧增大,且得不到最短路径;拓朴法通常不需要机器人的准确位置,对于位置误差也就有了更好的鲁棒性,缺点是建立拓朴网络的过程相当复杂,特别在增加障碍物时如何有效地修正已经存在的拓朴网是有待解决的问题。

总结以上全局路径规划方法的特点,可以发现,每一种方法在路径规划之前都要对障碍环境进行几何上简化的图形处理,形成所谓的“栅格”、“可视图”、“自由空间”、“拓朴网络”等等,然后再在这些图形中搜索路径。然而这样就降

基金项目:国家自然科学基金资助项目(10201026,10672111)

收稿日期:2009-01-16 修回日期:2009-04-16

低了效率,障碍环境复杂时,也就增加了算法的复杂度。

本文提出了移动机器人避碰规划的一种新方法——盲人摸路算法。该算法理论浅显,计算简单,信息存储量小。该算法的优点是,在进行路径规划之前,不需要对障碍环境进行处理,只要将障碍物的边界用离散点存储起来即可。然后用本文2.2.2节讲的一条简单的规则就可以直接搜索出无碰路径。该算法的缺点是搜索出的路径不一定是最短路径。

2 盲人摸路算法

2.1 算法思路

移动机器人避碰规划的盲人摸路算法,正是联想到盲人走路的方式,而形成的一种启发式算法。总结盲人走路的方式,可得到机器人避碰的一个思路:(1)首先指向目标点走,遇到障碍物则按一定方向沿着障碍物走;(2)按照一定的法则,若能够绕过障碍物,则机器人又指向目标点走,如此反复;(3)按照(1)、(2)的行走方式,便能得到一条连接起始点和目标点的无碰路径,再对得到的无碰路径按一定的方法进行简化。

2.2 算法过程

建立直角坐标系,假设 S_0 为起始点, D 为机器人要到达的目标点,点 S_0 与点 D 之间有若干障碍物。记所有障碍物的集合为 $OBS = \{O_1, O_2, \dots, O_n\}$,机器人“已到过的障碍物”集合记为 $sOBS$,则有 $sOBS \subseteq OBS$ 。所谓“已到过的障碍物”是指机器人已经沿着走过的障碍物,包括当前机器人所在的障碍物。

S_0, S_1, \dots, S_N 为机器人的行走路线, $S_k (k = 0, 1, \dots, N)$ 表示机器人当前所处的位置, $S_k - O_j$ 为点 S_k 到第 j 号障碍物的最小距离($k = 0, 1, \dots, N; j = 1, 2, \dots, n$), \vec{p}_k 为机器人下一步行走的方向, s 为机器人行走的步长, $0-1$ 变量 $state$ 表示机器人当前行走的状态, $state$ 为1表示指向目标点走, $state$ 为0表示沿着障碍物走。

2.2.1 障碍物边界的离散及障碍物的编号

障碍物边界的离散是指在障碍物边界上取若干点来近似的表示障碍物的形状。离散障碍物边界是为了计算机器人当前所处的位置 S_k 到障碍物的距离 $S_k - O_j (k = 0, 1, \dots, N; j = 1, 2, \dots, n)$,当 $S_k - O_j$ 小于事先设定的值 δ 时,则认为机器人即将碰到障碍物,那么下一步机器人应沿着障碍物走。

障碍物的编号是指按照障碍物的几何描述,根据空间连通性,对不连通的障碍物给予不同的编号,例如 O_1, O_2, \dots, O_n 。

2.2.2 机器人行走方向 \vec{p}_k 的确定

(1) 若机器人指向目标点走,则 $\vec{p}_k = \frac{D - S_k}{|D - S_k|}$ 。

(2) 若机器人沿着障碍物走,可以按照下述方法确定 \vec{p}_k 。

设机器人当前所处的位置 $S_k (k = 0, 1, \dots, N)$ 到障碍物

(离散后的障碍物)上 m 个点 M_1, M_2, \dots, M_m 的距离都小于事先设定的值 δ ,则令 $\vec{S}_k \vec{M} = \vec{S}_k \vec{M}_1 + \vec{S}_k \vec{M}_2 + \dots + \vec{S}_k \vec{M}_m$,取与 $\vec{S}_k \vec{M}$ 垂直且与 $\vec{S}_{k-1} \vec{S}_k$ 的夹角成锐角或直角的单位向量作为 \vec{p}_k ,具体有以下三种情况:

令 $\vec{p} = \frac{\vec{S}_k \vec{M}}{|\vec{S}_k \vec{M}|} = [p_1, p_2]^T$, ①若 $p_1 = 0, p_2 \neq 0$,则 $\vec{p}_k = \pm [1, 0]^T$; ②若 $p_1 \neq 0, p_2 = 0$,则 $\vec{p}_k = \pm [0, 1]^T$; ③若 $p_1 \neq 0, p_2 \neq 0$ 则 $\vec{p}_k = \pm [p_2, -p_1]^T$ 。

(3) 机器人“指向目标点走”与“沿着障碍物走”按照下述方法进行转化:

当机器人指向目标点走时,若下一步即将撞到障碍物,即机器人当前位置 S_k 距障碍物的距离 $S_k - O_j$ 小于事先设定的值 δ ,则下一步沿着障碍物走,此时按照(2)的方法确定 \vec{p}_k ;否则,继续指向目标点走,即按照(1)的方法确定 \vec{p}_k 。

当机器人沿着障碍物走时,若机器人当前所处的位置 S_k 和目标点 D 的连线 $S_k D$ 与机器人“已到过的障碍物”不相交,即对 $\forall O_j \in sOBS, S_k D \cap O_j = \emptyset$,则下一步指向目标点走,此时按照(1)的方法确定 \vec{p}_k ;否则,继续沿着障碍物走,即按照(2)的方法确定 \vec{p}_k 。

2.2.3 简化路径 $S_0, S_1, \dots, S_k, \dots, S_N$

简化路径是指尽可能用较短的路径来代替初次搜索出的路径。初次简化从起始点开始考虑,再次简化从目标开始考虑,具体按以下方法来完成:

初次简化,从 S_0 开始,依次判断线段 $S_0 S_N, S_0 S_{N-1}, \dots, S_0 S_1$ 是否与障碍物相交,若 $S_0 S_N, S_0 S_{N-1}, \dots, S_0 S_{k+1}$ 均与障碍物相交,而 $S_0 S_k$ 与障碍物不相交,则路径可简化为 S_0, S_k, \dots, S_N ,再从 S_k 开始,做同样的操作;若 $S_0 S_N, S_0 S_{N-1}, \dots, S_0 S_k$ 均与障碍物相交,则从 S_1 开始,做同样的操作。记简化以后的路径为 $\vec{S}_0, \vec{S}_1, \dots, \vec{S}_N$,在 $\vec{S}_i, \vec{S}_j (0 \leq i < j \leq N)$ 之间按照步长 S 插入新的节点,由此得到的路径记为 $P_1, P_2, \dots, P_k, \dots, P_M$ 。

再次简化,从 P_M 开始,依次判断线段 $P_M P_1, P_M P_2, \dots, P_M P_k, \dots, P_M P_{M-1}$ 是否与障碍物相交,若 $P_M P_1, P_M P_2, \dots, P_M P_{k-1}$ 都与障碍物相交,而 $P_M P_k$ 与障碍物不相交,则路径可简化为 $P_1, P_2, \dots, P_k, P_M$,再从 P_k 开始,做同样的操作;若 $P_M P_1, P_M P_2, \dots, P_M P_k, \dots, P_M P_{M-1}$ 均与障碍物相交,则从 P_{M-1} 开始,做同样的操作。

2.3 程序步骤

2.3.1 搜索初始路径

Step0: (初始步) $k = 0, sOBS = \emptyset$,给定 ε 。若存在 $O_j \in OBS, (j = 1, 2, \dots, n)$ 使得 $S_k - O_j < \delta$,则 $state = 0$,否则 $state = 1$;

Step1: (停止准则)若 $\|S_k D\| < \varepsilon$ 则停止;Step2: (选取 \vec{p}_k)

如果 $state = 1$:若存在 $O_j \in OBS$,使得 $S_k - O_j < \delta$,则按照2.2.2节(2)取 \vec{p}_k ,同时置 $sOBS = sOBS + \{O_j\}$,且 $state = 0$,否则按照2.2.2节(1)取 \vec{p}_k ;

如果 $state = 0$: 若存在 $O_j \in sOBS$, 使得 $S_k D \cap O_j \neq \Phi$, 则按照 2.2.2 节(2) 取 \dot{p}_k , 否则按照 2.2.2 节(1) 取 \dot{p}_k , 同时置 $state = 1$;

Step3:(循环) 令 $S_{k+1} := S_k + s \cdot \dot{p}_k, k := k + 1$, 转到 Step 1;

2.3.2 初次简化路径

Step0:(初始步) 赋值 $i := 0, k := N$;

Step1:(停止准则) 若 $k - i = 1$, 则停止;

Step2:(判断步) 若存在 $j \in \{1, 2, \dots, n\}$, 使得 $S_i S_k \cap O_j \neq \Phi$, 则令 $k := k - 1$, 继续执行 Step2, 否则执行 Step 3;

Step3:(循环) 路径简化为 $S_0, \dots, S_i, S_k, \dots, S_N, i := k, k := N$, 转到 Step1。

2.3.3 再次简化路径

Step0:(初始步) 赋值 $i := M, k := 1$;

Step1:(停止准则) 若 $i - k = 1$, 则停止;

Step2:(判断步) 若存在 $j \in \{1, 2, \dots, n\}$, 使得 $P_i P_k \cap O_j \neq \Phi$, 则令 $k := k + 1$, 继续执行 Step2, 否则执行 Step 3;

Step3:(循环) 路径简化为 $P_1, \dots, P_k, P_i, \dots, P_N, i := k, k := 1$, 转到 Step1。

3 收敛性证明

下面证明: 若起始点和目标点存在无碰路径, 则本文提出的盲人摸路算法一定能搜索到目标点。现采用反证法。

根据第 2 节对算法描述可知: 该算法不会使机器人停下来不动, 机器人要么沿着障碍物走, 要么指向目标点走。那么假如机器人未能到达目标点而以某种方式停滞不前。那么只能有下面三种情形:

1) 机器人一直指向目标点走;

2) 机器人一会儿指向目标点走, 一会儿沿着障碍物走, 如此反复。即: 指向目标点走 \rightarrow 沿着某个障碍物走 \rightarrow 指向目标点走 \rightarrow 沿着另一不同障碍物走 $\rightarrow \dots$, 如此无限循环下去;

3) 机器人一直绕着某一障碍物转。

总之机器人不会停下来。

对于情形 1) 和 2), 由于起始点与目标点的距离是有限的, 且障碍物的数量也是有限的, 那么这两种情形都是不会出现的;

对于情形 3), 假如机器人绕着如图 1 所示的障碍物 A 转。由第 2 节对算法描述可知, 它之所以没有从 A 表面的某一点逃逸障碍物而指向目标点走, 只能是因为该障碍物表面上任何一点与目标点的连线之间都有“已到过的障碍物”。

那么机器人又是怎样从这些“已到过的障碍物”而逃逸到障碍物 A 上的呢? 因为按照本文算法的描述, 机器人只能从障碍物 A 左边的锥形区域 B 才能到达障碍物 A, 所以它一定得绕到区域 B, 再往 A 上逃逸。

然而区域 B 中任意一点与目标点的连线之间也是有“已到过的障碍物”的, 按照本文算法的要求, 此时是不能逃逸

的。这便产生了矛盾, 所以情形 3) 也不会出现。证毕。

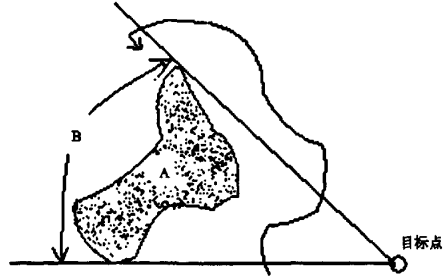


图 1 障碍物与目标点

4 数值算例

下面给出了五个算例, 第一个算例采用文献[12]实验的障碍物, 为了与原文献的试验结果进行比较, 我们的方法中没有画出简化后的路径; 第二、三个算例中的障碍物是作者自行设计的, 主要是为了检验对于凹型的复杂障碍物本文算法的有效性; 第四个算例的障碍物是一个迷宫图案; 上四个算例中, 均用一质点代表机器人, 最后一个算例中, 用一个半径为单位 1 的圆圈代表机器人。五个算例均采用 Matlab 编程实现。

算例一: 如图 2 所示。下方的小三角形表示起始点, 上方的五角星形表示目标点。图中左侧的折线路径是文献[12]中用蚁群算法结合人工势场方法的实验结果, 路径长度为 181.62。右侧的曲线路径是本文算法初次搜索出的路径, 计算时取 $\delta = 10, s = 1$, 路径长度为 180。

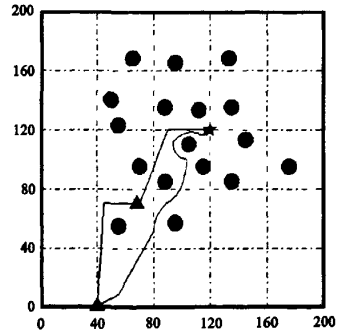


图 2 蚁群算法与盲人摸路法比较

算例二: 如图 3 所示, 机器人分别从起始点 $S_0(1, 4)$, $S_0(1, 7)$ 到达目标点 $D(8, 5)$, 计算时取 $\delta = 0.2, s = 0.1$, 左边的小圆圈表示起始点, 右边的小菱形表示目标点。虚线为初次搜索出的路径, 实线为简化以后的路径。两条路径中, 机器人都经过两个凹型障碍物, 并且均安全的逃逸出了凹型障碍物。

值得注意的是,在4号障碍物的出口处,机器人并未直接指向目标点而返回去行走,是因为这样机器人便会撞到已到过的4号障碍物。下面我们将图3中的障碍物稍微变一下,如图4所示,此时图3中的3、4号障碍物与四面的墙型障碍物视为同一障碍物,即给予同意编号3。



图3 算例二凹型障碍

算例三:如图4所示,机器人从起始点 $S_0(1,7)$ 到达目标点 $D(8,5)$,计算时取 $\delta = 0.2, s = 0.1$ 。注意到当机器人沿左边的墙型障碍物行走时,它并未冲着1号障碍物逃逸,这是因为机器人之前已经到达过1号障碍物。

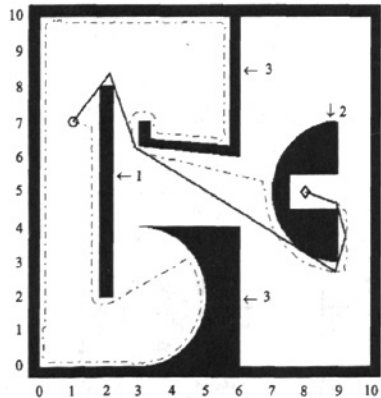


图4 算例三凹型障碍

算例四:如图5所示,机器人从起始点 $S_0(6.2, -1)$,分别到达目标点 $A(0.7, 21.5)$ 和 $B(11.5, 21.5)$,图中的星型线为初次搜索出的路径。第一条路径从 S 到 A ,计算时取 $\delta = 0.1, s = 0.1$,第二条路径从 S 到 B ,计算时取 $\delta = 0.17, s = 0.1$ 。为了便于观察,这里没有画出简化后的路径。

算例五:由于用一个直径为单位1的圆圈代表机器人,所以搜索路径时要考虑到机器人的大小。对于本文提出的算法,可以简单的取 δ 大于0.5。该算例中取 $\delta = 0.8, s = 0.4$ 如图6所示,机器人从起始点 $S_0(3,2)$ 到达目标点 $D(13,9)$ 。

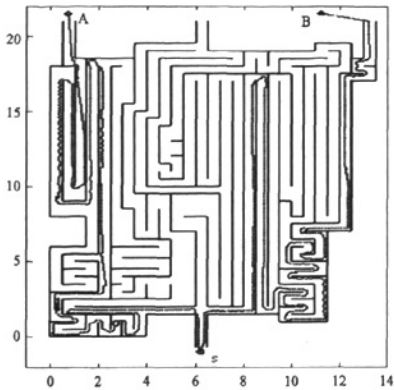


图5 迷宫型障碍

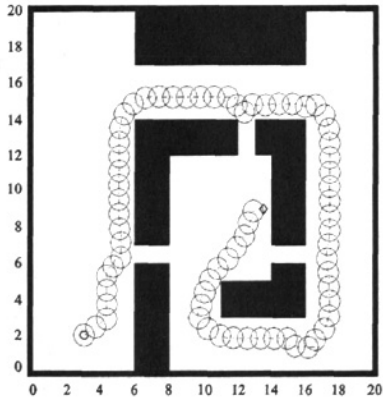


图6 非质点形机器人

5 算法改进及结论

1) 按照本文算法得到的路径,一般不是最短路径,但可以按照下述办法搜索出较短的路径:当机器人靠近障碍物时,可以取两个不同的方向 \vec{p}_k 同时进行搜索,得到不同的路径,进而从中选取较短的路径。如果能够进行并行计算,那么先搜索出的路径便是较短的路径。例如在图4中,当机器人到达圆弧形障碍物时,如果它能选择另外一个方向沿着障碍物走,不但能够很快逃逸障碍物,还可以搜索出较短的路径;

2) δ 与 s 的选取问题:理论上只要选取 δ 大于 s ,就能保证机器人不会撞到障碍物,但由于实际计算时存在误差,只有 δ 足够大于 s 时,才能做到这一点。然而, δ 太大又会减少机器人活动的自由空间。至于具体怎么选取 δ 与 s 才最佳,有待进一步讨论。

3) 路径规划是机器人控制技术研究中的一个重要问题。本文提出的盲人摸路算法,理论浅显,计算简单,信息存储量小,能够快速搜索出无碰路径。该算法只是一个初步的尝试,希望这一思路能够在大家的讨论和应用中得到完善和改进。

参考文献:

- [1] 黄献龙,梁斌,吴宏鑫. 机器人避碰规划综述[J]. 航天控制, 2002,20(1):36-40.
- [2] 席裕庚,张纯刚. 一类动态不确定环境下机器人的滚动路径规划[J]. 自动化学报, 2002,28(2):161-175.
- [3] 霍迎辉,张连明. 一种移动机器人的路径规划算法[J]. 自动化技术与应用, 2003,22(5):8-10.
- [4] 李磊,等. 移动机器人技术研究现状与未来[J]. 机器人, 2002,24(5):475-480.
- [5] 王志文,郭戈. 移动机器人导航技术现状与展望[J]. 机器人, 2003.
- [6] 艾海舟,张斌. 基于拓扑的路径规划问题的图形解法[J]. 机器人, 1990,12(5):20-24.
- [7] 孙增圻. 智能控制理论与技术[M]. 北京:清华大学出版社, 1997.
- [8] Khatib. Real-time obstacle for manipulators and mobile robot [J]. The International Journal of Robotic Research, 1986,(1): 90-98.
- [9] H C Nelson, Yung Cang Ye. An Intelligent Mobile Vehicle Navigator Based on Fuzzy Logic and Reinforcement Learning[J]. IEEE

Trans on Sys man and Cybern. Part B: Cybernetic, 1999,29(2): 314-320.

- [10] 李士勇. 模糊控制神经控制与智能控制论[M]. 哈尔滨:哈尔滨工业大学出版社, 1999.
- [11] 马兆青. 基于栅格方法的移动机器人实时导航避障[J]. 机器人, 1996,18(6):344-348.
- [12] 芮延年,蒋晓梅,刘鑫培,朱犖,刘开强. 基于蚁群算法 AS_R 移动式机器人路径规划的研究[J]. 机电工程技术, 2006,35(9):73-75.

[作者简介]



王永兴(1981-),男(汉族),陕西延安人,硕士研究生,助教,研究领域为变分不等式数值解,计算数学;

丁睿(1969-),男(汉族),浙江定海人,博士,教授,研究领域为变分不等式数值解,计算数学,计算力学;

姚林泉(1961-),男(汉族),江苏吴县人,博士,教授,研究领域为计算力学,计算数学。

(上接第148页)

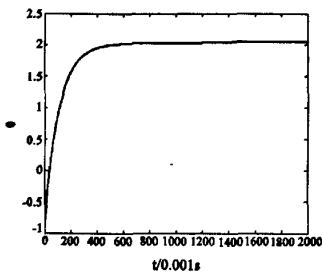


图5 跟踪常设定值的参数估计 $\hat{\theta}$ 变化曲线

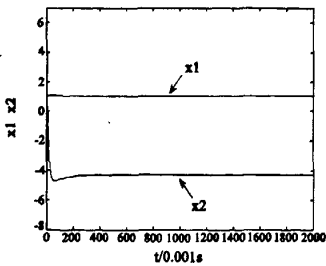


图6 跟踪常设定值的系统状态仿真曲线

参考文献:

- [1] M Krstic, et al. Nonlinear and adaptive control design[M]. New York: Wiley, 1995.
- [2] R A Freeman, M Krstic and P Kokotovic. Robustness of adaptive nonlinear control to Bounded uncertainties [J]. Automatica, 1998,34:1227-1230.


- [3] E Sontag. Smooth stabilization implies coprime factorization[J]. IEEE Trans on Automatic Control, 1989,34:435-443.
- [4] E Sontag and Y Wang. New characterizations of input-to-state stability[J]. IEEE Trans on Automatic Control, 1996,41:1283-1291.
- [5] P Florchinger. A universal formula for the stabilization of control stochastic differential equations[J]. Stochastic Analysis and Appl., 1993,11:155-162.
- [6] H Deng, et al. Stabilization of stochastic nonlinear systems driven by noise of unknown covariance[J]. IEEE Trans on Automatic Control, 2001,46(8):1237-1253.
- [7] M Krstic, et al. Stabilization of Nonlinear Uncertain Systems[M]. London: Springer Verlag, 1998.
- [8] H Deng and M Krstic. Stochastic nonlinear stabilization - Part I: Backstepping design[J]. Syst. Control Lett., 1997,32:143-150.
- [9] 季海波, 奚宏生, 陈志福, 王俊. 具有不确定噪声的随机非线性系统的鲁棒自适应跟踪[J]. 控制理论与应用, 2003,20(6): 843-848.
- [10] 王俊, 张梅, 奚宏生, 成荣. 随机非线性系统鲁棒自适应控制的模块化设计[J]. 控制与决策, 2005,20(11):1269-1273.

[作者简介]



蔡涛(1984-),男(汉族),安徽合肥人,硕士研究生,专业为控制理论与控制工程,主要研究方向为非线性系统控制;

王俊(1963-),男(汉族),安徽合肥人,博士,教授,研究方向为自适应控制、鲁棒控制优化和设施农业环境调控等。

作者: [王永兴](#), [丁睿](#), [姚林泉](#), [WANG Yong-xing](#), [DING Rui](#), [YAO Lin-quan](#)
作者单位: [苏州大学数学科学学院](#), [江苏](#), [苏州](#), [215006](#)
刊名: [计算机仿真](#) 
英文刊名: [COMPUTER SIMULATION](#)
年, 卷(期): [2010](#), [27](#) (5)

参考文献(12条)

1. [H C Nelson;Yung Cang Ye](#) [An Intelligent Mobile Vehicle Navigator Based on Fuzzy Logic and Reinforcement Learning](#) 1999(02)
2. [Khatib](#) [Real-time obstacle for manipulators and mobile robot](#) 1986(01)
3. [孙增圻](#) [智能控制理论与技术](#) 1997
4. [艾海舟;张跋](#) [基于拓扑的路径规划问题的图形解法](#) 1990(05)
5. [王志文;郭戈](#) [移动机器人导航技术现状与展望](#)[期刊论文]-[机器人](#) 2003(5)
6. [李磊](#) [移动机器人技术研究现状与未来](#)[期刊论文]-[机器人](#) 2002(05)
7. [霍迎辉;张连明](#) [一种移动机器人的路径规划算法](#)[期刊论文]-[自动化技术与应用](#) 2003(05)
8. [芮延年;蒋晓梅;刘鑫培;朱翬](#) [刘开强 基于蚁群算法AS_R 移动式机器人路径规划的研究](#)[期刊论文]-[机电工程技术](#) 2006(09)
9. [马兆青](#) [基于栅格方法的移动机器人实时导航避障](#) 1996(06)
10. [李士勇](#) [模糊控制神经控制与智能控制论](#) 1999
11. [席裕庚;张纯刚](#) [一类动态不确定环境下机器人的滚动路径规划](#)[期刊论文]-[自动化学报](#) 2002(02)
12. [黄献龙;梁斌;吴宏鑫](#) [机器人避碰规划综述](#)[期刊论文]-[航天控制](#) 2002(01)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjzfz201005038.aspx