

## 机器人对多运动障碍物环境中方向可变运动目标的跟踪

樊晓平<sup>1</sup>, 李双艳<sup>2</sup>, 瞿志华<sup>1</sup>

(1. 中南大学 自动化工程研究中心, 湖南 长沙 410075; 2. 中南林业科技大学 工业学院, 湖南 长沙 410004)

**摘要:** 机器人要实现对动态环境中可变方向运动目标的跟踪, 必须采用动态的规划算法. 本文在快速随机搜索树算法的基础上, 采用滚动时间帧的思想, 周期性地采集环境信息与目标运动状况, 来预测未来环境中障碍物的分布情况及运动目标位置. 在每个周期内用 BIAS\_RRT 来引导机器人行走, 以适应障碍物与目标运动方向的变化. 仿真结果表明, 该算法能有效跟踪在多运动障碍物环境中方向可变的运动目标.

**关键词:** 运动目标; 动态避障; 有偏快速随机搜索树; 有限时间帧

**中图分类号:** TP24      **文献标识码:** A

## Changeable moving-goal tracking for robots in the environment of dynamic multi-obstacles

FAN Xiao-ping<sup>1</sup>, LI Shuang-yan<sup>2</sup>, QU Zhihua<sup>1</sup>

(1. Research Center for Automation Engineering, Central South University, Changsha Hunan 410075, China;

2. College of Industrial Engineering, Central South University of Forestry & Technology, Changsha Hunan 410004, China)

**Abstract:** When a robot is required to track a moving object in dynamic environment, a dynamic algorithm must be taken. An algorithm called rolling timeframe biased rapidly-exploring random tree is proposed in this paper. Based on the analysis of the stochastic characteristics of rapidly-exploring random tree, a parameter called bias is introduced to speed up the search. Taking advantages of the rolling timeframe, robots collect the information of dynamic obstacles and object at the beginning of a period, and estimate their distributions in operation space of next timeframe. The robot plans local path using biased rapidly-exploring random tree algorithm. After many times of such local planning, the robot gets its object at last. Simulation results show that the algorithm can obtain good results in tracking the object with changing direction in dynamic environments.

**Key words:** moving object; dynamic obstacle avoidance; biased rapidly-exploring random tree; limit timeframe

### 1 引言 (Introduction)

随着对机器人智能要求的提高, 希望机器人不但可以到达指定的静止目标, 而且可以跟踪运动的目标, 当目标运动发生变化时, 机器人要能适应其变化. 已有对动态未知环境中机器人的避障规划研究<sup>[1, 2]</sup>. 如文献[3]对简单环境下的动态目标的跟踪进行了研究. 但简单环境在实际中是少见的, 实际的运行环境往往很复杂, 且存在各种障碍. 本文应用滚动时间帧的思想<sup>[4]</sup>, 通过不断进行局部规划, 可以适应复杂的动态环境, 到达最终目标. 其局部规划可采用一种随机性的路径规划算法 RRT<sup>[5]</sup>, 它具有概率统一性, 因而搜索能力很强, 但反过来又会降低搜索效率. 论文采用有限深度与有限节点数的 BIAS\_

RRT 来规划机器人在每个周期内的路径, 引导机器人避开动态障碍物, 跟踪运动方向发生变化的目标, 提高搜索效率. 论文详细分析了 BIAS\_RRT 的优点, 并用矢量和方法获取新节点, 使得算法可以满足物理现实的限制. 最后给出了仿真结果. 结果表明, 该算法能有效地跟踪在多运动障碍物环境中方向可变的运动目标.

### 2 快速扩展随机树简介 (Brief introduction to rapidly-exploring random tree)

快速扩展随机树 (rapidly-exploring random tree, RRT), 是一种随机性搜索算法, 具有其它随机性算法 (如概率路标) 一样的优点, 有无启发函数都可运行, 对整个工作空间的搜索是概率均等的, 因而

搜索能力很强<sup>[6]</sup>.

如图1,令 $C$ 代表工作空间,机器人在 $C$ 中搜索路径, $C_{\text{free}}$ 表示自由空间. $x_{\text{init}}$ 为给定机器人的初始状态,属于自由空间,是RRT的根节点,算法中 $\rho > 0$ 为给定的RRT生长的最小单位长度,称为步长. $\text{dis}(x, y)$ 表示两节点间的距离, $T_k$ 表示生成了 $k$ 个节点的RRT.于是,RRT的构建过程可以描述如下:

- Step 1** 任意给出一个树的初始节点  $T$ .  
 $\text{init}(x_{\text{init}})$ ;
- Step 2** 在 $C$ 空间中选取一个随机的状态节点  $x_{\text{rand}}$ ,且  $x_{\text{rand}} \in C_{\text{free}}$ ;
- Step 3** 找出  $T_k$  中距离  $x_{\text{rand}}$  最近的节点  $x_{\text{near}}$ .  
 即  $\text{dis}(x_{\text{rand}}, x_{\text{near}}) = \min_{x \in T_k} \text{dis}(x, x_{\text{rand}})$ ;
- Step 4** 在  $x_{\text{rand}}$  与  $x_{\text{near}}$  的连线上求  $x_{\text{new}}$ ,  $x_{\text{new}}$  必须满足  $x_{\text{new}} \in C_{\text{free}}$  且  $\text{dis}(x_{\text{new}}, x_{\text{near}}) = \rho$  的条件. 如果存在  $x_{\text{new}}$ , 转 Step 5, 否则转 Step 2;
- Step 5**  $T_k$  增加一个新节点. 令  $T_{k+1}$  表示新的RRT, 则  $T_{k+1} = T_k + x_{\text{new}}$ ;
- Step 6** 结束.

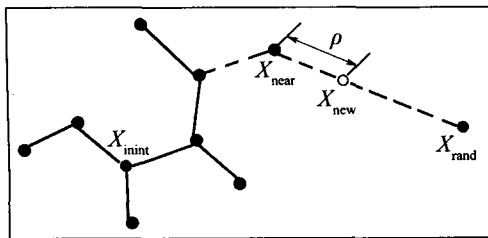


图1 快速扩展随机树构建图

Fig. 1 Graph of rapidly-exploring random tree

可见RRT是从已有的节点中选择一个离随机目标点最近的节点进行扩展.RRT搜索的是整个空间,并倾向于搜索未搜索过的空间.

### 3 有限时间帧的有偏快速扩展随机树 (Limited timeframe biased rapidly-exploring random tree)

显然,用单个RRT来规划动态空间中的避障路径是不可行的,结合滚动时间帧的思想,用多RRT进行规划.即在每个周期内,用RRT生成一棵有限深度的搜索树,来规划各个周期的局部路径.每个周期包含多个时间帧序列,其中每一帧都会生成一个新节点:

在每一周期开始的时间帧,机器人用传感器获取运动目标与障碍物的状态信息(运动速度).假定在这个较短的周期内,运动目标与障碍物的速度方向均不变,则可以预测它们在此周期内未来时间帧

的状态,并用预测时间帧的信息进行规划(因为机器人的目标不是运动物的当前位置,而是预测出的运动物下一帧位置,这样机器人才能跟踪运动目标).规划产生的每个节点包含了在对应时间帧机器人的可行状态(位置、方向与速度).根据产生的节点信息,可推算出控制量的输入(转向角与速度改变的大小).接着,进入下一个周期,作同样的规划,直到跟踪到实际目标为止.

对于每个周期内产生的新路径,机器人并不完全执行,因为预测总是不完全准确的,在规划中,也存在误差.由于一个周期内最后几帧的预测误差最大,因此,在一个周期内最后几帧规划出来的路径不被执行.

机器人模型如图2,其外形为三角形,它的位置为其几何中心坐标 $E(x, y)$ ,机器人只能沿中心轴 $AE$ 直线方向前后运动,且速度大小有一定的约束,必须小于给定的最大速度 $x_{\text{max}}$ .速度方向与 $X$ 轴的正向所成的夹角为 $\alpha$ , $\varepsilon$ 为转角大小 $\varepsilon = \alpha' - \alpha$ ,即机器人从前一个状态到下一个状态要旋转的角度大小.

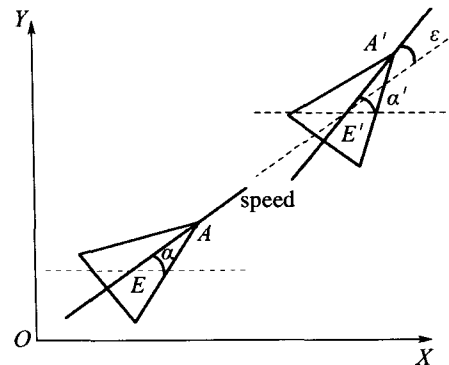


图2 机器人的模型表示

Fig. 2 Robot model

整个路径规划过程如下:

- Step 1** 给出初始环境配置  $C_{\text{init}}$ , 周期  $T$ ;
- Step 2** 在周期  $T$  内, 传感器采集目标与障碍物的运动情况, 根据目标与障碍物的速度信息预测在该周期内下一帧目标与障碍的分布情况, 获取新的环境信息;
- Step 3** 根据预测的新环境信息, 建立有限时间帧的有偏的快速扩展随机树 LIMITBIAS-RRT, 其深度小于给定的最大深度  $\text{MAX\_PATH\_LENGTH}$ ;
- Step 4** 根据选择的最优节点, 推算出控制输入, 即速度改变大小与转角大小;
- Step 5** 输入控制量, 让机器人前进, 机器人只执行部分规划出的路径, 余下最后几帧不予考虑;
- Step 6** 判断机器人是否到达目标, 到达则

规划完成,否则,转到 Step 2.

每个节点包含的元素不仅仅是机器人的位置、指向父节点的指针,还包括机器人的速度大小、方向、节点的深度.这里的机器人不是点状的,而是三角形的.所以每个节点的形状也是三角形的.其中的一些算法细节如下.

LIMITBIAS-RRT 树建立的算法流程为:

**Step 1**  $T.\text{init}(x_{\text{init}})$ ; //任意给出一个树的初始节点;

**Step 2** for  $K = 1$  to  $K < \text{MAX\_TREESIZE}$  do; //做 MAX\_TREESIZE 循环;

**Step 3**  $x_{\text{target}} \leftarrow \text{BIAS\_Configuration}(x_{\text{goal}}, \text{BLAS})$ ; //产生有偏的局部节点  $x_{\text{target}}$ ;

**Step 4**  $x_{\text{near}} \leftarrow \text{Nearest\_Neighbor}(x_{\text{target}}, T)$ ; //找到离局部节点最近的  $x_{\text{near}}$ ;

**Step 5**  $x_{\text{new}} \leftarrow \text{New\_Configuration}(x_{\text{near}}, x_{\text{target}}, \Delta t)$ ; //产生新节点  $x_{\text{new}}$ ;

**Step 6**  $u \leftarrow \text{Output\_Control}(x_{\text{new}}, x_{\text{near}})$ ; //计算控制量  $u$ ;

**Step 7**  $T.\text{add\_vertex}(x_{\text{new}})$ ; //将新节点加入树中;

**Step 8**  $T.\text{add\_edge}(x_{\text{near}}, x_{\text{new}}, u)$ ; //将新的边加入树中;

**Step 9** return  $T$ . //返回.

这里假设第 5 步得到的  $x_{\text{new}}$  是属于自由空间的,否则,回到第 3 步,产生新的局部目标点  $x_{\text{target}}$ .直到得到的  $x_{\text{new}}$  属于自由空间的为止.

新节点也并不是像标准的 RRT 那样通过简单地由  $x_{\text{near}}$  加减一个步长获取的,而是通过矢量运算得到的.如图 3,  $v$  是机器人的速度,  $s$  是它原来的位置,  $t$  是当前帧的目标,  $a$  是机器人下一时刻将到达的位置,  $\vec{sa}$  为机器人当前速度的方向,与  $X$  轴正向成的角为  $\alpha$ ,  $\vec{ap}$  与  $X$  轴正向的角为  $\theta$ ,  $\vec{aq}$  与  $X$  轴正向的角为  $\gamma$ ,  $\vec{st}$  与轴的角为  $\beta$ , 而  $MA$  为给定的加速度的大小.  $i$  是从  $a$  点向  $st$  作垂线的相交点.以  $i$  为基点,新节点是沿  $\vec{st}$  加减  $m$  ( $m = \sqrt{MA^2 - ai^2}$ ) 得到的  $p$  或  $q$ , 在该处,比较  $q$  与  $p$  到目标位置  $t$  距离的大小,取小者为新节点位置,图中新节点的位置为  $p$ , 新节点的速度可用  $\vec{sa}$  与  $\vec{ap}$  之和推出.得到的控制量转角 (rotation) 是  $\alpha - \beta$ , 速度改变大小为  $MA$ .

新节点获取 New\_Configuration 的算法流程如下:

New\_Configuration( $x_{\text{near}}, x_{\text{target}}, \Delta t$ ):

**Step 1**  $s.x \leftarrow x_{\text{near}.x}$ ,  $s.y \leftarrow x_{\text{near}.y}$ ; //找到的最近节点作为当前点;

**Step 2**  $a.x \leftarrow x_{\text{near}.x + x_{\text{near}.xspeed * \Delta t}$ ,  $a.y \leftarrow x_{\text{near}.y + x_{\text{near}.yspeed * \Delta t}$ ; //根据速度计算出机器人下一时刻的位置;

**Step 3**  $i = \text{perpendicular}(a, st)$ ,  $\beta = \text{angle}(st)$ ,  $\theta = \text{angle}(ap)$ ,  $\alpha = \text{angle}(sa)$ ; //  $i$  是  $a, st$  的交点,  $\beta, \theta, \alpha$  分别是  $st, ap, sa$  与水平线的夹角;

**Step 4**  $x_p.x = i.x + m * \cos \beta$ ,  $x_q.y = i.y + m * \sin \beta$ ; //往两个不同方向扩展得到的新节点  $x_p, x_q$ ;

**Step 5**  $x_q.x = i.x - m * \cos \beta$ ,  $x_q.y = i.y - m * \sin \beta$ ;

**Step 6** if  $\text{distance}(x_{\text{target}}, x_p) < \text{distance}(x_{\text{target}}, x_q)$ ; //比较两者,取距离小者为新节点

$x_{\text{new}.x = x_p.x$ ,

$x_{\text{new}.y = x_p.y$ ,

$x_{\text{new}.xspeed = x_{\text{near}.xspeed + MA * \cos \theta}$ ,

$x_{\text{new}.yspeed = x_{\text{near}.yspeed + MA * \sin \theta}$ ;

**Step 7** else,

$x_{\text{new}.x = x_q.x$ ,

$x_{\text{new}.y = x_q.y$ ,

$x_{\text{new}.xspeed = x_{\text{near}.xspeed + MA * \cos \gamma}$ ,

$x_{\text{new}.yspeed = x_{\text{near}.yspeed + MA * \sin \gamma}$ .

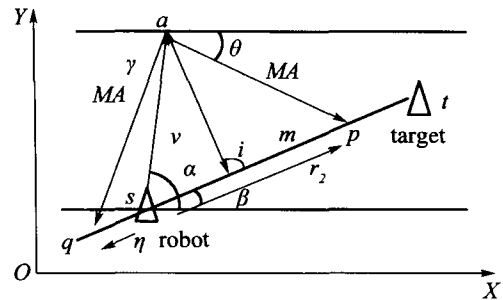


图 3 新节点的产生

Fig. 3 Generation of new nodes

#### 4 仿真结果 (Simulation results)

用 C++ 语言实现上述算法,仿真结果如图 4~6 所示.

1) 图 4 表示在复杂静态障碍物环境中,目标仅作匀速直线运动的情况,这是较简单的情况,机器人从障碍物之间的狭道中向运动目标前进,在第 8 步机器人作了较大的方向变动.

2) 图 5 表示有多个静态障碍物,仅有一个作直线运动障碍物的环境中,机器人跟踪作匀速直线运动目标的情况.

3) 图 6 表示最复杂的一种情况,即有多个运动障碍物,并有两个障碍物在运行中改变,同时运动目

标的运动方向也发生变化的情况. 机器人依然能绕过障碍物追上运动目标.

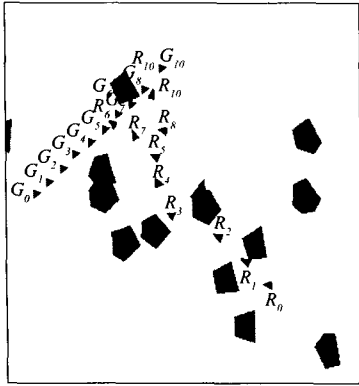


图4 在复杂静态障碍物环境中跟踪运动方向不变的目标  
Fig. 4 Tracking fixed-direction goal with complex static obstacles

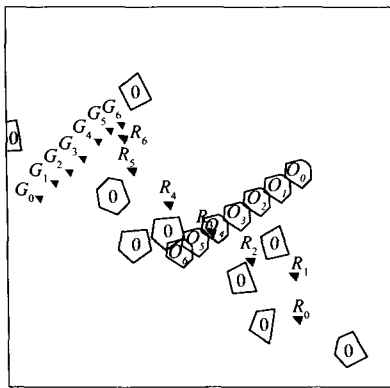


图5 一个障碍物作直线运动时跟踪运动方向不变的目标  
Fig. 5 Tracking fixed-direction goal with one beeline moving obstacle

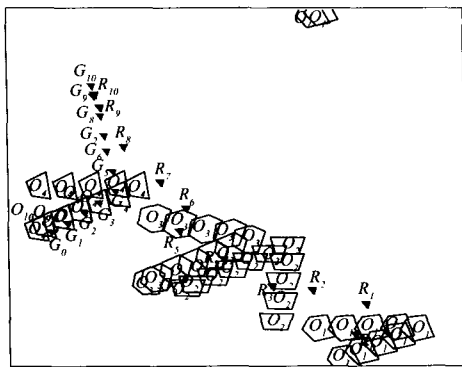


图6 两个障碍物与目标运动方向发生变化的环境  
Fig. 6 Tracking changing-direction goal with two changing-direction obstacles

为了清楚地显示运动的整个过程,在图4至图6中用步数  $n$  代表大致的运动环节. 目标与机器人外形均为三角形,机器人为绿色,用  $R$  表示. 运动目标为红色,用  $G$  表示. 动态障碍物运动的过程也用步数标出,用  $O$  表示. 在仿真中假设各个障碍物可以重

叠,运动目标可以与其它物体相交. 当障碍物运动到一边界时,会出现在另一边,以相同的规律在工作空间中继续运动.

因为该算法含有随机元素,即使在相同的环境中,每次的实验结果也不完全相同,有时,甚至会有很大的差别,为了说明试验效果,对最复杂情况作了多次仿真试验,表1为快速扩展随机树运行纪录.

从表1的结果可以看出,机器人在该算法引导下,能成功地避开动态障碍,跟踪运动变化的目标.

表1 BIAS\_RRT 在环境(3)下的仿真记录

Table 1 Record of BIAS\_RRT in environment(3)

试验次数	扩展节点数	有效节点数	运行时间/秒
1	3448	2746	17
2	3343	2707	16
3	3455	2515	15
4	5441	4798	29
5	3736	2608	21
平均值	3864	3074	20

### 5 结论(Conclusion)

在较短周期内,机器人采集环境信息,预测未来状况,规划局部路径,能够适应运动目标的变化与运动障碍物的变化,跟踪到方向变化的目标. 仿真结果表明上述算法是实时有效的,可运用到实时机器人的动态避障上去. 但预测方法还值得研究,精确的预测才能保证精确的避障.

### 参考文献(References):

- [1] 樊长虹,陈卫东,席裕庚. 动态未知环境下一种 Hopfield 神经网络路径规划方法[J]. 控制理论与应用, 2004, 21(3): 345-350.  
(FAN Changhong, CHEN Weidong, XI Yugeng. Hopfield neural networks for path planning in dynamic and unknown environments [J]. *Control Theory & Applications*, 2004, 21(3): 345-350.)
- [2] 席裕庚. 动态不确定环境下广义控制问题的预测控制[J]. 控制理论与应用, 2000, 17(5): 665-670.  
(XI Yugeng. Predictive control of general control problems under dynamic uncertain environments [J]. *Control Theory & Applications*, 2000, 17(5): 665-670.)
- [3] LAVALLE S M, GONZALEZ-BANOS H H, BECKER C, et al. Motion strategies for maintaining visibility of a moving target. [C]// *Proceedings IEEE Int Conf on Robotics and Automation*. Albuquerque, NM, USA: IEEE Press, 1997: 731-736.

1993.  
(LU Qiang, SUN Yuanzhang. *Nonlinear Control of Power Systems* [M]. Beijing: Science Press, 1993.)
- [9] KRSTIC M, KANELAKOPOULOS I, KOKOTOVIC P. *Nonlinear and Adaptive Control Design* [M]. New York: John Wiley & Sons, INC, 1995.
- [10] HU W, MEI S, et al. Nonlinear adaptive decentralized stabilizing control of multi-machine systems [J]. *Applied Mathematics and Computation*, 2002, 133(2-3): 519-532.
- [11] WENLEI LI, YUANWEI JING, GEORGI M, et al. Robust nonlinear control of TCSC for power system via adaptive backstepping design [C]// *IEEE Conference on Control Applications 03 Conference*. Turkey: IEEE Press, 2003.
- [12] KARAGIANNIS D, ASTOLFI A. Nonlinear adaptive control of systems in feedback form: An alternative to adaptive Backstepping [C]// *IFAC Symposium on Large Scale Systems: Theory and Applications*. Osaka, Japan: [s. n.], 2004.
- [13] ASTOLFI A, ORTEGA R. Immersion and invariance: a new tool for stabilization and adaptive control of nonlinear systems [J]. *IEEE Trans on Automatic Control*, 2003, 48(4): 590-606.
- [14] KHALIL H. *Nonlinear Systems (3rd edition)* [M]. Englewood Cliffs, NJ: Prentice Hall, 2002.

#### 作者简介:

付俊 (1979—),男,东北大学信息科学与工程学院博士研究生,研究方向为电力系统鲁棒非线性控制、电力系统混杂全局控制, E-mail: fuziling@126.com;

赵军 (1957—),男,东北大学信息科学与工程学院教授,博士生导师,1991年于东北大学信息科学与工程学院获博士学位,1998—1999年作为高级访问学者赴美国 Illinois (Urbana-Champaign) 研修,从2003年10月至2005年5月作为高级访问学者在香港城市大学研修. 现为中国自动化学会控制理论委员会委员,《控制理论与应用》编委,研究方向:复杂非线性系统的结构研究,混杂系统、切换系统稳定性研究.

#### (上接第350页)

- [4] 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.  
(XI Yugeng, ZHANG Chungang. Rolling path planning of mobile robot in a kind of dynamic uncertain environment [J]. *Acta Automatica Sinica*, 2002, 28(2): 161-175.)
- [5] CHENG P, SHEN Z, LAVALLE S M. RRT-based trajectory design for autonomous automobiles and spacecraft [J]. *Archives of Control Sciences*, 2001, 11(3-4): 167-194.
- [6] LAVALLE S M, KUFFNER J J. Rapidly-exploring random tree: Progress and prospects. [C]//DONALD B R, LYNCH K M, RUS D, editors, *Algorithmic and Computational Robotics; New Directions*, Wellesley, MA: A K Peters, Ltd, 2001: 293-308.

#### 作者简介:

樊晓平 (1961—),博士,教授,博士生导师,主要研究方向为机器人控制,虚拟现实技术,智能交通系统等,已发表有关论文100余篇, E-mail: xpfan@mail.csu.edu.cn;

李双艳 (1979—),女,硕士,主要研究方向为机器人路径规划,智能控制等;

冀志华 (1963—),博士,教授,博士生导师,长江学者讲座教授,美国中佛罗里达大学 (University of Central Florida) 教授, IEEE 高级会员,主要研究方法为机器人控制,非线性系统理论,鲁棒控制等,已发表有关论文300余篇.

#### (上接第354页)

- [7] WANG G X, LIU Y W, HE Z, et al. The lifting technique for sampled-data systems: useful or useless [J]. *Acta Automatica Sinica*, 2005, 31(3): 491-494.
- [8] CHEN T W, FRANCIS B A.  $H_\infty$ -optimal sampled-data control: computation and design [J]. *Automatica*, 1996, 32(2): 223-228.
- [9] 王广雄,刘彦文,何朕,等.  $H_\infty$  离散化的 MATLAB 程序 [J]. *电机与控制学报*, 2004, 8(4): 316-318.  
(WANG Guangxiong, LIU Yanwen, HE Zhen, et al.  $H_\infty$  discretization using MATLAB [J]. *Electric Machines and Control*, 2004, 8(4): 316-318.)
- [10] SAFONOV M G, LIMEBEER D J N, CHIANG R Y. Simplify-

ing the  $H_\infty$  theory via loop-shifting, matrix-pencil, and descriptor concepts [J]. *Int J of Control*, 1989, 50(6): 2467-2488.

#### 作者简介:

王广雄 (1933—),男,教授,博士生导师,主要研究方向为  $H_\infty$  控制理论及应用,控制系统的鲁棒设计, E-mail: gxwang@hope.hit.edu.cn;

刘彦文 (1976—),女,博士研究生,目前研究方向为采样系统和鲁棒控制系统设计;

何朕 (1972—),女,副教授,硕士生导师,研究方向为  $H_\infty$  控制、广义系统.