文章编号: 1001-0920(2019)05-0927-10 DOI: 10.13195/j.kzyjc.2017.1497

穿越稠密障碍物的自适应动态窗口法

王永雄[1,2†], 田永永[1], 李 璇[1], 李梁华[1]

(1. 上海理工大学 光电信息与计算机工程学院，上海 200093；
2. 上海理工大学 上海康复器械工程技术研究中心，上海 200093)

摘 要: 针对应用广泛的局部避障算法 —— 动态窗口法(DWA)穿越稠密障碍物时存在路径不合理、速度和安全性不能兼顾等问题,提出参数自适应的DWA算法,根据机器人与障碍物距离和障碍物的密集度自动调整目标函数中的权值,以自适应环境的动态变化,从而获得移动机器人的最佳运行速度和合理路径. 该方法可明显改善机器人穿越稠密障碍物区域时的性能;同时,该方法还可避免机器人从密集障碍物区域外绕行以及轨迹不平滑现象. 仿真实验表明:改进的DWA算法在复杂环境中通过逐步优化可使运行轨迹更加合理,能够同时兼顾路径平滑性和安全性;机器人在离稠密障碍物较远处保持高速,通过狭窄通道或者稠密障碍物区域时速度适当降低,安全性更高,实验中总迭代次数和运行时间可缩短20 %以上.

关键词: 机器人；避障；动态窗口法；参数自适应；人性化；局部路径规划

中图分类号: TP242 **文献标志码:** A

Self-adaptive dynamic window approach in dense obstacles

WANG Yong-xiong[1,2†], *TIAN Yong-yong*[1], *LI Xuan*[1], *LI Liang-hua*[1]

(1. School of Optical-Electrical and Computer Engineering，University of Shanghai for Science and Technology，Shanghai 200093，China；2. Shanghai Engineering Research Center of Assistive Devices，University of Shanghai for Science and Technology，Shanghai 200093，China)

Abstract: Dynamic window approaches (DWAs) are widely used in local obstacle avoidance of mobile robots. A self-adaptive DWA algorithm is proposed for the problem that the classical DWA exists unreasonable path in dense obstacles and it cannot ensure both speed and security. In order to adapt to the dynamic environment, the weight of speed in the target function is adjusted automatically based on the distance and orientation between the robot and obstacles. The optimal velocity and reasonable planned path of the mobile robot are obtained using the self-adaptive DWA. Thus the performance of the robot crossing an area of dense obstacle is significantly improved. The problems that the robot might move around the dense obstacles and result in an unsmooth path are solved. Our experiments show that the proposed algorithm can make running track more reasonable through gradual optimization in the complex environment than old one, and it ensures the smoothness and security of robot route simultaneously. When the robot is far from the dense obstacles, the robot keeps high speed. When the robot passes through a narrow passage or a dense obstacle area, the speed appropriately reduces and the safety grows high. The total number of iterations and the run-time can be reduced more than 20 % in our experiments.

Keywords: robot；avoid obstacles；DWA；self-adaptation；humanization；local path planning

0 引 言

自主导航是移动机器人核心技术之一. 在实际环境中,特别是人机共存的复杂环境下,机器人可以获取环境大致地图信息,但由于存在移动物体、人或者其他易变因素,很难获得环境的完整信息[1]. 在局部地图信息已知情况下,局部动态路径规划方法是实现智能机器人自主导航的首选方法[2]. 例如,机场搬运机器人必须对无法预知的行人、推车等变化做出快速反应,能够在环境有较大变化的情况下自主完成避障和新的路径规划.

最简单的思路是沿着起始点与目标点的连线运动,遇到障碍物时,沿着障碍物边缘绕行,但该方法

收稿日期: 2017-11-07；**修回日期:** 2018-04-18.

基金项目: 国家自然科学基金项目(61673276, 61703277, 61473193).

责任编委: 方勇纯.

作者简介: 王永雄(1970−), 男, 副教授, 博士, 从事智能机器人及视觉等研究；田永永(1990−), 男, 硕士生, 从事移动机器人的研究.

†通讯作者. E-mail: wyxiong@usst.edu.cn.

会导致路径不圆滑,总长度增加. 人工势场法将目标点抽象成引力源,障碍物抽象为斥力源,将合力作为机器人的作用力,从而推算出机器人的运动方向和位姿,在局部避障方面有较好的实时性,但是在某些情况下,它容易让机器人陷入局部极值点,例如多个相近障碍物间可能无法找到路径,在狭隘的通道中会产生振荡,生成的路径明显不合理. 1991年,Borenstein[3]提出了基于向量场直方图的VFH(Vector field histogram)算法,在多障碍物环境或者狭窄通道中能够找出局部较优的运动方向,运行轨迹合理,但此方法没有考虑机器人的尺寸、动力学和运动学等特性. 在此基础上,文献[4]充分考虑了机器人的尺寸、轨迹和动力运动学性能,提出了改进的VFH+方法. 此后,文献[5]再加入预测,提出了VFH*算法,使得机器人能够选择一个局部较优的运动方向.

上述算法几乎都不能直接产生机器人避障时的最优速度,且没有考虑机器人自身物理限制. Simmons[6]提出了曲率速度法(CVM),它将避障问题描述为速度空间带约束的优化问题,考虑了机器人的速度和加速度等物理限制和障碍物的环境约束. 在满足所有约束的情况下,建立了包含速度、安全性和路径3个因素的优化目标函数. 在CVM的基础上,Fox等[7]提出了更完善的动态窗口法(DWA),充分考虑了机器人的物理限制、环境约束以及当前速度等因素. DWA算法首先根据机器人外形尺寸和安全距离,将障碍物进行圆形膨化,然后根据当前速度和加速度建立一个预选速度窗口,再通过目标函数优化得到下一时刻的最优速度(包括速率和方向). 目标函数综合考虑了航向角、速率和障碍物距离3个因素,同时考虑了真实机器人自身物理约束(最大线速度、最大角度)、环境约束,该方式能够直接获得机器人的期望线速度和角速度,得到的轨迹比较平滑,适合真实移动机器人的运行,有效解决了围绕障碍物绕行的问题,也解决了VFH和VFH+算法中对运动学及动力学过度简化的问题,且多数情况下能避免陷入局部极值,因此在现有的ROS机器人系统上得到了广泛应用. Seder等[8]在DWA基础上,将Focused D*与DWA结合,有效解决了针对移动障碍物的避障问题. 但是,现有的DWA方法依然存在以下问题: 1) 对于复杂环境,机器人得到的轨迹不够平滑,在障碍物较稠密区域,机器人可能不会从稠密区中选择短路径通过,而是绕开稠密区域从而导致路径过长. 2) 当DWA目标函数中的速度权值较大时,通过两个障碍物中间或狭窄通道时,机器人太靠近某一障碍物,如果是行人或者运动的物体,容易出现碰撞,导致安全性和人性化大幅降低; 当速度权值较小时,路径安全合理,但会导致整个行程速度明显偏低,总体运行时间变长. 其根本原因是: 采用固定的权值,无法适应各种复杂或动态的环境.

在人机共融环境中,不仅要求避开障碍物,还要求在穿越人群时,保持与人的“社交距离”,尊重个人空间[9]. 为了提高导航的准确度,将Kinect与声呐测得的障碍物信息进行融合,得到精确的障碍物位置信息,有助于提高动态环境中运动规划的准确性和鲁棒性[10]. 文献[11]设计了一种混合运动规划器(HM),将全局规划与局部规划结合起来,并利用平滑算法减少抖动,使其在具有动态障碍物的狭窄通道中可以安全运行; 文献[12]提出了在动态环境中进行社会适应性路径规划的框架,通过生成模拟人的路径轨迹,使得算法更加智能化、人性化. 这几种方法主要考虑了人的舒适性和人性化,但忽视了机器人的效率. 因此,一个好的避障方法,要同时满足合理性、高效性、最优性和人性化[13].

基于以上分析,本文揭示DWA方法作为一种局部动态避障算法存在的局限性,利用移动机器人自身携带的传感器,快速学习机器人与障碍物距离和障碍物的密集度信息,然后借鉴控制理论中的自适应思想,提出参数自适应的DWA算法,根据环境的变化自动调整权值,得到机器人的最佳运动速度,从而获得更加安全、合理的路径. 该方法可明显改进DWA方法穿越稠密障碍物区域的性能,能够同时兼顾速度、安全性和人性化.

1 自适应动态窗口法

1.1 经典DWA算法简介

根据机器人位置和速度的对应关系,DWA算法将机器人的位置控制转化为速度控制,将避障问题描述为速度空间带约束的优化问题,其中包含两类约束: 1) 机器人速度和加速度限制; 2) 环境和障碍物约束.

根据机器人速度限制,速度组合$V_s(v,\omega)$满足

$$V_s=\{(v,\omega)|0\leqslant v\leqslant v_{\max},-\omega_{\max}\leqslant\omega\leqslant\omega_{\max}\}. \tag{1}$$

其中: v表示线速度,ω表示角速度.

整个机器人轨迹可以认为是由n个时段的n个小圆弧组成的,从安全性考虑,根据简单的运动学推算,在某一时刻,不发生碰撞的允许速度V_a为(机器人和障碍物尺寸做过安全性扩展)

$$V_a=\{(v,\omega)|v\leqslant\sqrt{2\cdot\mathrm{dist}(v,\omega)\cdot\dot{v}},\ \omega\leqslant\sqrt{2\cdot\mathrm{dist}(v,\omega)\cdot\dot{\omega}}\},\tag{2}$$

其中$\mathrm{dist}(v,\omega)$表示下一时刻机器人位置离障碍物的距离. 由于受机器人动力学限制,线加速度v和角加速度ω有上下限. 在给定机器人的当前线速度为v_{curr}、当前角速度为v_{curr}和时间间隔为d_t的条件下,下一时刻的可达速度$V_d(v,\omega)$满足

$$V_d=(\nu,\omega)=\begin{cases}\nu_{\mathrm{curr}}-\dot{\nu}_{\max}d_t\leqslant\nu\leqslant\nu_{\mathrm{curr}}+\dot{\nu}_{\max}d_t,\\ \omega_{\mathrm{curr}}-\dot{\omega}_{\max}d_t\leqslant\omega\leqslant\omega_{\mathrm{curr}}+\dot{\omega}_{\max}d_t.\end{cases}\tag{3}$$

最终的速度范围由上述3个速度交集合成,即

$$V_r=V_s\bigcap V_a\bigcap V_d.\tag{4}$$

根据环境和障碍物约束,以及机器人动力学约束,得到一个带动力学和环境等约束的二维速度空间,其中横坐标为角速度,纵坐标为线速度,如图1所示.

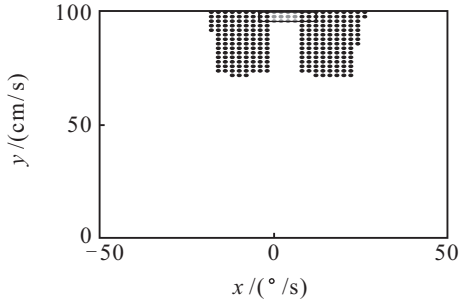


图1　速度空间

图1中: V_s表示整个速度空间,其范围是$[-50,50;0,100]$; V_d是一个动态窗口,表示下一时刻($t+1$时刻)可达速度,如图中黑色方框区域; V_a表示整个速度空间中去除黑色区域的部分(灰色和白色部分); V_r是所有区域相交得到的灰色区域,表示$t+1$时刻可行速度.

机器人通过目标函数对下一时刻的速度做预测,从中选择满足约束的最优速度. 目标函数综合考虑了机器人运动速度、航向和安全性. 定义如下:

$$G(v,\omega)=\alpha\cdot\mathrm{heading}(v,\omega)+\beta\cdot\mathrm{dist}(v,\omega)+\gamma\cdot v(v,\omega).\tag{5}$$

使得$G(v,\omega)$值最大的(v,ω)即为最优速度. 其中: $\mathrm{heading}=\pi-\theta$, θ表示机器人航向与目标线(机器人位置同目标点连线称为目标线)之间的夹角, heading用于衡量机器人对目标的方向性,当机器人运动方向完全指向目标点时,其值最大; dist表示预轨迹(图2中弧线由点组成)中距离障碍物的最小距离; v表示圆弧轨迹中$t+1$时刻的线速度,优化结果将尽量选择动态窗口V_r中线速度大的值; α,β,γ为3个权值参数,为避免其中一项值占比过高,在优化之前,先将目标函数的3项参数归一化处理为[0, 1]之间的参数. 机器人多个预测轨迹示意图如图2所示. 在图2中: A表示$(v_{\mathrm{cuur}}+\dot{v}\cdot d_t,\omega_{\mathrm{curr}}-\dot{\omega}\cdot d_t)$, B表示$(v_{\mathrm{cuur}}-\dot{v}\cdot d_t,\omega_{\mathrm{curr}}+\dot{\omega}\cdot d_t)$.

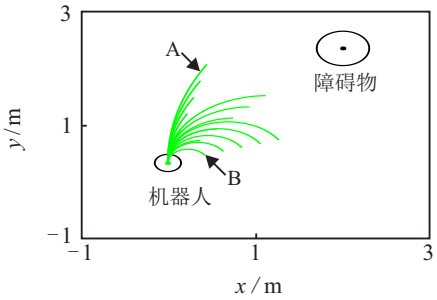
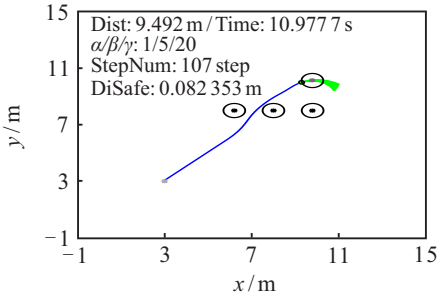
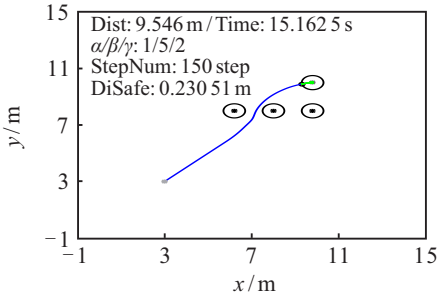


图2　机器人多个预测轨迹

通过式(5)计算图中每一个预轨迹(弧线)对应的函数G,使得G最大速度即为$t+1$时刻的最优速度. 现有DWA算法的速度权值固定,当速度权值较大时,机器人运行步数少、时间短,但是存在安全性低等问题,当速度权值较小时,运行步数偏多、时间偏长,主要问题如图3所示.



(a)　固定高速度权值(权值分别是1/5/20)



(b)　固定低速度权值(权值分别是1/5/2)

图3　机器人运行轨迹

机器人运行轨迹如图3所示. 在图3中: StepNum表示总步数; Dist表示总路径长度; Time表示总运行时间; DiSafe表示安全距离,即所有障碍物与轨迹的最短距离;浅灰色点表示机器人起始点;黑色圆点表示经过膨化后的障碍物;深灰色点表示目标点;设置

机器人到达目标点中心0.5 m的范围内即为到达目标点.

由图3(a)可以看出,当速度权值较高时,虽然步数少(107步)、运行时间短(10.977 7s),但机器人穿过狭窄通道时,过于靠近一侧障碍物,安全性低(安全距离为0.082 353 m). 如果障碍物是人,机器人距离人太近,则对人造成较大的压迫感,缺少人性化. 由图3(b)可以看出,当速度权值低时,机器人从狭窄通道中运行时路径合理,安全性高(安全距离为0.230 51m),但是步数大(150步)、运行时间长(15.162 5 s),效率远低于图3(a)中速度权值较高的情况. 另外,现有的DWA算法还存在以下问题:机器人可能围绕密集障碍物区域外运行,或者轨迹不够平滑,如果是载人机器人,则曲折的路径会降低舒适度.

1.2 自适应动态窗口法

当机器人选择较高速度运行时,在密集障碍区域,基于现有DWA算法的通过性明显变差[14],规划的路径可能太靠近某一障碍物,导致安全性大幅降低;当选择低速运行时,在密集障碍区域通过性变好,路径安全合理,但会导致整个行程速度明显偏低,总体运行时间变长. 另外,当密集度进一步增加时,会导致无法从多个障碍物之间穿过[14],而是从多个障碍物的外围绕行,致使路径过长,这时需要降低速度(通过降低权值γ实现),但降低权值会导致路径不平滑. 上述问题的根本原因是目标函数中的权值一旦确定,无法适应各种复杂环境或动态环境. 因此,借鉴控制理论中的自适应思想,提出自适应DWA算法,根据不同的障碍物信息,动态调整目标函数中速度权值γ. 为简单起见,本文只动态调整速度权值. 另外,本文还提出一种简单的障碍物密集度判断方法.

1.2.1 基于障碍物信息的动态参数调整

考虑到现有移动机器人普遍携带了测距(避障)传感器,能在一定范围内快速检测到障碍物,这里假设机器人可获得中等或较低精度的障碍物距离和方位. 在t时刻,选取机器人运行方向的一定角度扇形区域作为局部密集障碍物的计算依据,如图4所示.

设扇形区域的障碍物个数为M,第i个障碍物与机器人的距离为D_i,方位角为θ_i,定义M大于阈值时(实验中阈值为2),此区域为障碍密集区. 经过简单的几何计算,可以快速计算出第i个障碍物与第j个障碍物之间的间距Int_{ij},即

$$\mathrm{Int}_{ij}=\sqrt{D_j^2+D_i^2-D_jD_j\cos(\theta_i-\theta_j)},\ D_i\leqslant D_j. \tag{6}$$

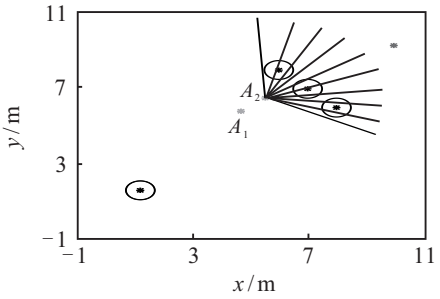


图 4 实验仿真环境

当间距Int_{ij}大于一定阈值(大于2倍的障碍膨化半径R)时,可以推断机器人能安全地从两个障碍物之间穿越. 鉴于安全性是首要保证条件,选择机器人与局部区域内所有障碍物的实际最短距离$D_{\min}$为自适应函数动态权值函数的输入. 当障碍物与机器人距离大于距离阈值时(即$D_{\min}\geqslant D_s$),速度项的权值始终选取最大值. 根据机器人性能定义阈值D_s如下:

$$D_s=l\cdot(v_{\max}/\dot{v}). \tag{7}$$

其中:D_s与机器人最高线速度$v_{\max}$成正比,与线加速度v成反比. 制动能力越强,动态阈值越小,l为参数(实验中为0.9).

当机器人与障碍物的实际距离$D_{\min}\leqslant D_s$时,考虑到线速度v的动态权值γ_d与距离$D_{\min}$成单调递增趋势,定义速度的动态权值γ_d为

$$\gamma_d=\begin{cases}\gamma_{\min}+\kappa(\gamma_{\max}-\gamma_{\min})\left(\dfrac{D_{\min}}{D_s}\right)^a, D_{\min}\leqslant D_s;\\ \gamma_{\max},\ D_{\min}>D_s.\end{cases} \tag{8}$$

其中:$D_{\min}$表示机器人离最近障碍物的距离;a表示指数(这里取$a=1.5$);为防止产生的权值过大或者过小,设置的权值范围为$[\gamma_{\min},\gamma_{\max}]$. 这里$\gamma_{\max}$和$\gamma_{\min}$的选取方法为:在原DWA中,$\gamma_{\max}$是以最短时间通过障碍物密集区域的对应值;$\gamma_{\min}$是通过狭窄通道且最安全的对应值. 综合以上分析得到如下改进目标函数:

$$\max G(\nu,\omega)=\alpha\times\mathrm{heading}(\nu,\omega)+\beta\times\mathrm{dist}(\nu,\omega)+\gamma_d\times\nu(\nu,\omega). \tag{9}$$

1.2.2 自适应动态窗口法流程步骤

Step 1: 根据传感器信息得到实际机器人与障碍物距离和方位,判断是否进入密集障碍物区域;计算障碍物的间距Int_{ij},判断能否穿越密集障碍物区域.

Step 2: 获得$D_{\min}$值,并计算动态范围阈值D_s.

Step 3: 根据式(8)计算线速度v的动态权值γ_d,将得到的动态权值γ_d代入目标函数.

Step 4: 通过以下3步搜索备选速度空间:

1) 形成圆弧轨迹: 动态窗口法的圆弧轨迹由二维的速度空间(v,ω)确定,这里v表示平移速度,ω表示旋转速度;

2) 确定动态窗口: 机器人在当前速度和加速度的情况下,下一个时刻机器人能够达到的速度称为可达速度,所有可达速度形成一个动态窗口;

3) 确定速度空间: 包括不可能碰撞轨迹的速度,以及可能碰撞轨迹但机器人能够及时制动的速度,由两者组合成全部的允许速度(v,ω).

Step 5: 对目标函数的3个输入heading、dist和v分别作归一化,通过目标函数(9)得到最优速度组合$(v,\omega)_{t+1}$,令其作为$t+1$时刻机器人运行的速度.

Step 6: 执行速度,判断是否到达目标点: 若是,则结束; 否则返回Step 1,进入下一时刻循环.

不可能碰撞轨迹定义: 在每个预测轨迹终点处,以其速度方向做射线,与障碍物不相交(切)的轨迹称为不可能碰撞轨迹,否则称为可能碰撞轨迹.

2　仿真验证及结果分析

为了验证自适应DWA算法性能,首先进行不同权值下(分别为低、高和动态权值)速度优化对比实验; 然后验证不同复杂环境下算法改进前后的综合性能,在保证安全性的情况下对时间、路程等进行定性和定量比较.

2.1　机器人和算法参数初始化

本文构建的模拟地图如图4所示,在此基础上进行多障碍物的自动避障仿真实验. 图4中的浅灰色米字形A_1表示起始点1,且此时机器人状态为S_1,浅灰色米字形A_2表示另一起始点2,且此时机器人状态为S_2,黑色圆形表示4个经膨化后的障碍物,深灰色米字形C表示目标点,扇形区域为A_2点处的障碍物密集计算区.

DWA算法涉及多个机器人参数和算法参数,根据实际情况和常规硬件配置,将选取的参数列入表1～表4中.

表 1　机器人物理限制

最小线速度 $v_{\min}/(\mathrm{m/s})$	最大线速度 $v_{\max}/(\mathrm{m/s})$	线加速度 $\dot{v}/(\mathrm{m/s^2})$	最小角速度 $\omega_{\min}/(^\circ/\mathrm{s})$	最大角速度 $\omega/(^\circ/\mathrm{s})$	角加速速度 $\dot{\omega}/(^\circ/\mathrm{s^2})$
0	1	0.4	−50	50	80

表 2　经典DWA算法参数

障碍膨化半径 R/m	时间分辨率 d_t/s	线速度分辨率 $d_v/(\mathrm{m/s})$	角速度分辨率 $d_\omega/(^\circ/\mathrm{s})$	轨迹预测时间 evaldt/s	距离阈值 distThers/m
0.5	0.1	0.02	2	2	0.5

表 3　目标函数和速度项参数

角度项α	安全距离项β	速度项最小阈值$\gamma_{\min}$	速度项最大阈值$\gamma_{\max}$	M	参数l	参数k	参数a
1	5	2	20	⩾2	0.9	1	1.5

表 4　起始点A_1和A_2的机器人状态(x,y,θ,v,ω)

起始点	x / m	y / m	θ / rad	v / (cm / s)	ω / (° / s)
A_1	4.683 3	5.796 5	0.680 7	100	4
A_2	5.535 7	6.534 8	0.750 5	48	2

为了避免dist值过大造成评价函数值的权重过大,这里阈值distThers限定为0.5 m(具体与机器人参数、性能参数相关).

2.2　目标函数优化结果

为得到$t+1$时刻的最优速度,首先对3个输入heading、dist、v分别进行归一化,然后通过目标函数$\max\ G(v,\omega)$筛选出$t+1$时刻的最优速度. 因此DWA算法在速度空间中通过对目标函数进行寻优搜索实现局部最优.

为了保证整个目标函数的完整性,假设机器人t时刻的加速度能够从0到任意大,线速度范围为[0, 100]、角速度范围为$[-50,50]$,绘制在此种情况下t时刻的3D目标函数,如图5所示.

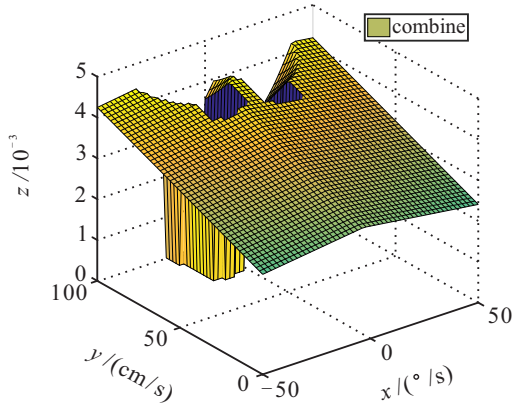


图 5　忽略加速度约束时目标函数的3D图

在图5中:x轴为机器人的角速度,y轴为线速度,z轴为目标评价函数值.

机器人速度为$(v,\omega)_t = (100,4)$,位置为$(x,y,\theta)_t = (4.683\,3,5.796\,5,0.680\,7)$. 对比图1的速度空间可知:图1中黑色区域(碰撞速度)对应图5的目标函数值等于0,表示不可达速度;非黑色区域的目标函数值大于0,表示可达速度,其值越高,对应的速度组合越合理即为最优速度,被选为$t+1$时刻机器人的执行速度.

2.2.1 速度权值γ低时经典DWA优化实验

为验证速度权值γ低时($\gamma=2$),机器人与障碍物的距离对速度优化的影响,分别通过图6和图7演示距离障碍物较远和较近时,原DWA算法的优化结果.

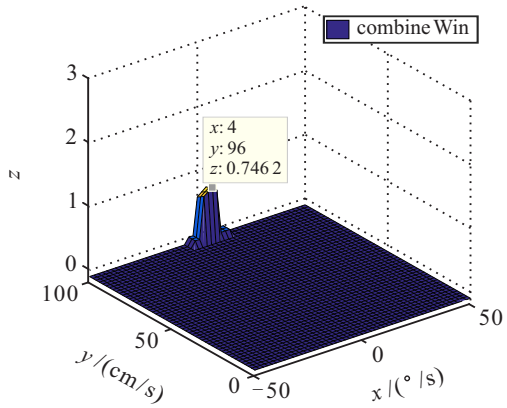


图6 机器人离障碍物较远时目标函数3D图(γ较低)

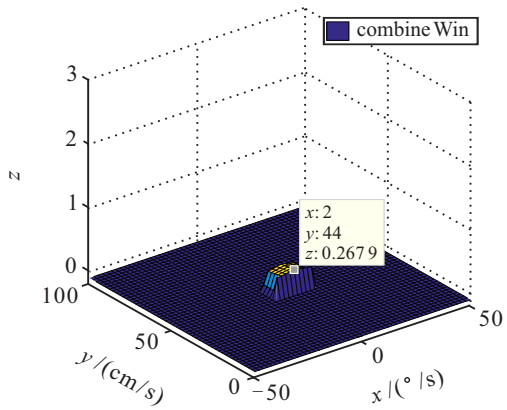


图7 机器人离障碍物较近时目标函数3D图(γ较低)

当机器人离障碍物较远时(如图4中A_1所示),$\gamma=2$,机器人位姿$(x,y,\theta)_t=(4.683\,3,5.796\,5,0.680\,7)$,速度$(v,\omega)_t = (100,4)$,同时考虑机器人加速度约束情况下,DWA算法得到的目标函数如图6所示. 图6中凸起区域是动态窗口区域(在图5中依据当前速度和加速度截取一个窗口,因为图5中使用整个速度空间中所有速度进行归一化,而图6中只使用速度窗口中的速度进行归一化,所以两幅图函数值大小不同),表示$t+1$时刻的可达速度,其中包含备选速度(函数值大于0)和碰撞速度(函数值等于0),当动态窗口不接触x、y轴时,域中心点就是t时刻的机器人实际速度;非凸起区域是动态窗口之外区域,表示$t+1$时刻不可达速度(其函数值小于0). 设t时刻的线速度为100,动态速度窗口中的线速度、角速度范围分别是[96, 100]和[−4, 12],备选速度中最大函数值对应的(v,ω)即为最优速度(图6中黑顶点所示),函数值为0.746 2,对应速度为(96, 4),即$v_{t+1}<v_t$,并将其选为$t+1$时刻的执行速度. $v_{t+1}=96$,从图6中可以看出,最优的线速度是动态窗口中最低线速度. 由此可知,速度权值γ低且机器人离障碍物远时,$t+1$时刻的最优结果获得了动态速度窗口中的较低线速度.

当t时刻机器人离障碍物较近(如图4中A_2所示)时,$\gamma=2$,并且机器人位姿$(x,y,\theta)_t = (5.535\,7,6.534\,8,0.750\,5)$,速度$(v,\omega)_t=(48,2)$,得到一个动态窗口目标函数如图7所示. 优化后得到的$t+1$时刻最优速度是(44, 2),即$v_{t+1}<v_t$. 由此可得,当速度权值γ低且机器人离障碍物较近时,$t+1$时刻的最优线速度是当前窗口中较低线速度.

由图6和图7可知,当目标函数中速度项的权值γ较低时,不管机器人距离障碍物多远,$t+1$时刻最优速度的线速度都是备选速度中较低的线速度.

2.2.2 速度权值γ较高时经典DWA优化结果

同样地,为了验证速度项为高权值时($\gamma=20$),机器人距离障碍物的远近对速度优化的影响,这里分别演示距离障碍物较远和较近时经典DWA优化的结果. 当机器人离障碍物较远时,原DWA算法优化得到的最优速度如图8所示. 同理,当机器人离障碍物较近时,优化得到的最优速度如图9所示.

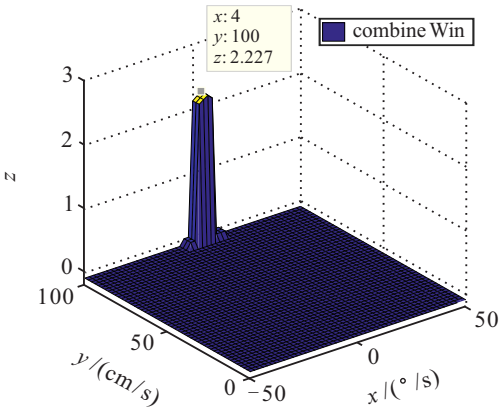


图8 机器人离障碍物较远时目标函数3D图(γ较高)

当t时刻机器人离障碍物较远时(如图4中A_1所示),$\gamma=20$,机器人位姿$(x,y,\theta)_t = (4.683\,3,5.796\,5,0.680\,7)$,速度$(v,\omega)_t = (100,\ 4)$,DWA得到的目标函数如图8所示. 优化后的$t+1$时刻最优速度是(100,4),即$v_{t+1}=v_t$,也是机器人最高线速度. 由此可知,当速度权值$\gamma$高且机器人离障碍物较近时,$t+1$时刻的最

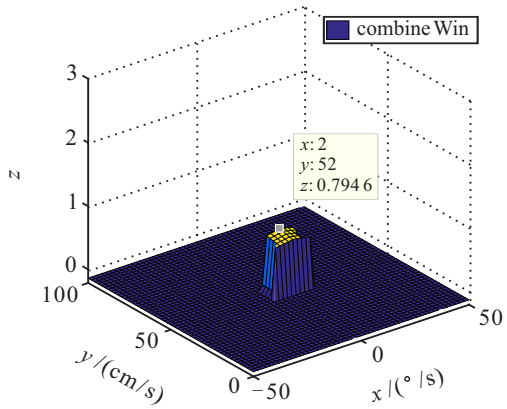


图 9　机器人离障碍物较近时目标函数3D图(γ较高)

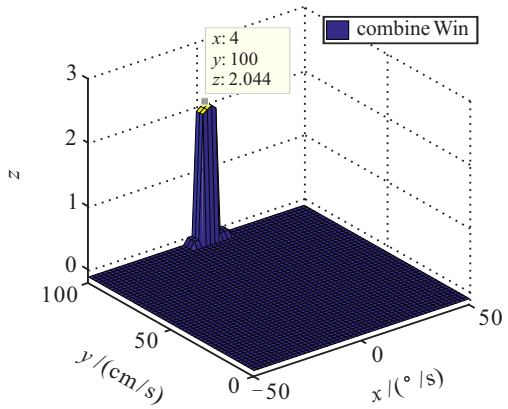


图 10　距离障碍物较远时基于动态速度权值的目标函数3D图

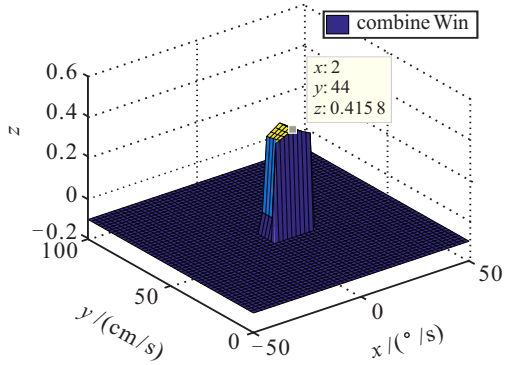


图 11　距离障碍物较近时基于动态速度权值的目标函数3D图

优线速度是当前窗口中较高速度.

当t时刻机器人离障碍物较近时(如图4中A_2所示),$\gamma = 20$,机器人位姿$(x,y,\theta)_t = (5.535\,7, 6.534\,8, 0.750\,5)$,速度$(v,\omega)_t = (48,2)$,得到的动态窗口目标函数如图9所示. 优化后得到的$t+1$时刻最优速度是$(52,2)$,即$v_{t+1} > v_t$. 由此可见,速度权值高且机器人离障碍物较近时,得到的$t+1$时刻最优线速度是当前窗口中较高的线速度.

由图8和图9可以看出,当目标函数中速度项的权值γ较高时,不管机器人距离障碍物多远,$t+1$时刻最优线速度都是备选速度中较高的线速度.

2.2.3　基于速度权值动态变化的自适应DWA优化实验

为了验证速度权值动态变化时($\gamma_d \in [2,20]$),机器人距离障碍物的远近对速度优化的影响,图10和图11分别演示距离障碍物较远和较近时自适应动态窗口法的优化结果.

当t时刻机器人距离障碍物较远时(如图4中A_1所示),γ为动态权值,机器人位姿$(x,y,\theta)_t = (4.683\,3, 5.796\,5, 0.680\,7)$,速度$(v,\omega)_t = (100,4)$,同时考虑机器人加速度限制的情况下,自适应动态窗口法得到的目标函数如图10所示. 优化后的$t+1$时刻最优速度是(100,4),即$v_{t+1} = v_t$为窗口中最高线速度,也是机器人最高线速度.

当采用动态权值γ_d且机器人离障碍物远时,$t+1$时刻最优速度的线速度是当前窗口中较高的线速度. 当t时刻机器人离障碍物较近时(如图4中A_2所示),$\gamma = \gamma_d$,机器人位姿$(x,y,\theta)_t = (5.535\,7, 6.534\,8, 0.750\,5)$,速度$(v,\omega)_t = (48,2)$,得到的动态窗口目标函数如图11所示. 优化后得到的$t+1$时刻最优速度是$(44,2)$,即$v_{t+1} < v_t$. 由此可见,当采用动态权值γ_d且机器人离障碍物近时,$t+1$时刻的最优线速度是当前窗口中较低的线速度.

通过图10和图11可以看出:当速度权值为动态权值且机器人离障碍物远时,得到的$t+1$时刻最优线速度是备选速度中较高的线速度;而且离障碍物近时,$t+1$时刻最优线速度是备选速度中较低的线速度. 为更加清晰地显示算法改进前后的对比效果,将γ分别为低、高和动态3种情况的目标函数优化结果列入表5.

从表5中可以清楚地看出:算法修改后,离障碍物较远时机器人选择窗口中的较高速,以提高速度性;离障碍物较近时选择窗口中的较低速,以保证安全性. 上述选择完全符合人的思维逻辑,根据不同环境自适应地调整机器人运行中的速度和轨迹,同时保

表 5　不同速度权值对比效果

离障碍物距离	低权值$\gamma=2$		高权值$\gamma=20$		动态权值$\gamma_d=[2,20]$	
	远	近	远	近	远	近
t时刻速度	(100, 4)	(48, 2)	(100, 4)	(48, 2)	(100, 4)	(48, 2)
$t+1$时刻速度	(96, 4)	(44, 2)	(100, 4)	(52, 2)	(100, 4)	(44, 2)
最优线速在窗口中高低	较低	较低	较高	较高	较高	较低

证高速性和安全性.

2.3 自适应DWA与原算法的避障、动态路径规划效果对比

2.3.1 动态权值γ_d函数参数选择

为了选取式(8)中合适的k和a值,设$m = k \cdot (D_{\min}/D_{\text{scale}})^a, p = D_{\min}/D_{\text{scale}}$,则$m = k \cdot p^a$. 其中:$a$为自变量,$m$为因变量. 令$m_1 = f(a_1), m_2 = f(a_2)$,当$0 < a_1 < a_2$且$p \in (0, 1)$时,$m_1 > m_2, m$是关于$a$的单调递减函数. 当机器人接近障碍物时,速度应较低,因此速度权值需减小,对应的m值应该较小,则a应该较大,但是当a大于一定值时,会造成动态权值γ_d过小,导致在接近障碍物时速度过低,整个运行时间过长,从而影响机器人效率,因此a需要在一个合理的范围内. 为简单起见,设$k = 1, a > 0$,a分别取不同值进行对比实验,为得到定量的对比,设定$k = 1, 1 \leqslant a \leqslant 1.6$,进行7组实验,结果如表6所示.

表 6 不同a时的实验对比($k = 1$)

	$a = 1$	$a = 1.1$	$a = 1.2$	$a = 1.3$	$a = 1.4$	$a = 1.5$	$a = 1.6$
步数	155	155	162	165	170	173	176
距离/m	13.616	13.632	13.672	13.684	13.696	13.686	13.672
时间/s	15.782 3	15.781 9	16.486 7	16.789 9	17.296 7	17.602 3	17.909 8
安全距离/m	0.122 55	0.130 14	0.162 08	0.168 73	0.176 23	0.181 313	0.187 13
安全性	差	不好	较好	好	优秀	优秀	优秀

从表6可以看出:当$a < 1.1$时,速度性虽然好,但安全性很差;随着a逐渐变大,速度性下降,但安全性得到提高;当$a \geqslant 1.5$时,安全性已经不能提高,但速度性却降低. 因此a的较合理的取值范围为[1.3, 1.5].

2.3.2 算法改进前后避障总体性能对比

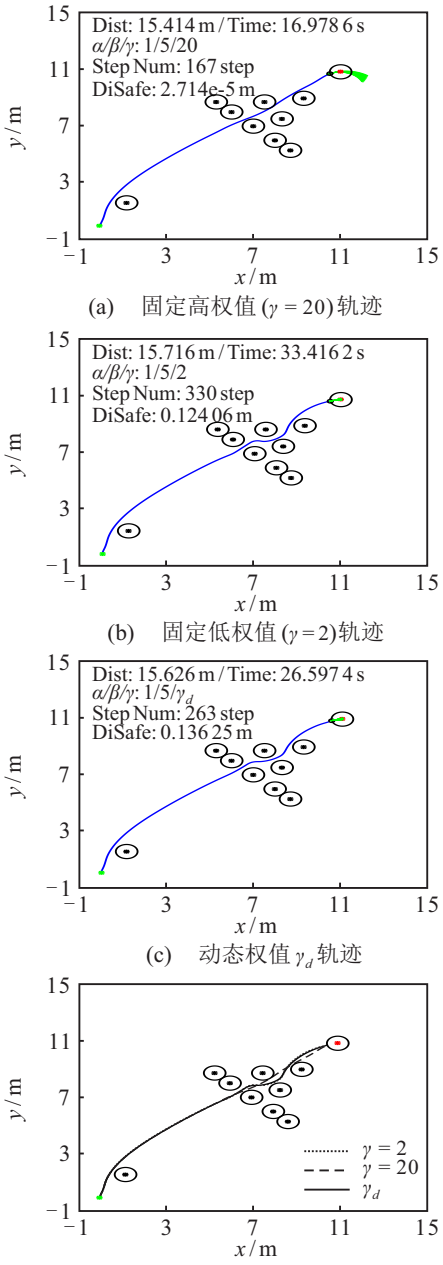
为了验证自适应DWA算法能够同时兼顾安全性和速度性,在同样的场景下进行避障效果对比实验,如图12所示.

图12(a)是目标函数速度项的权值$\gamma = 20$时的机器人运行轨迹图,整个轨迹长度为15.414 m,运行步数(从t时刻到$t+1$时刻的优化过程为一步)为167步,时间为16.978 6 s. 在通过密集障碍物区域时,机器人过于接近障碍物,安全性较低(安全距离为2.714 e-5 m).

图12(b)是目标函数速度权值$\gamma = 2$时的机器人运行轨迹图,整个轨迹长度为15.716 m,运行步数为330步,时间为33.416 2 s,步数和时间远高于高权值时(图12(a))的运行时间. 但是在密集障碍物区域,机器人从障碍物中间通过,安全性较高(安全距离为0.124 06 m).

当采用动态的速度权值时,机器人运行轨迹如图12(c)所示,整个轨迹长度为15.626 m,运行步数为263步,时间为26.597 4 s,安全性较高(安全距离为0.136 25 m). 与图12(a)对比,图12(c)保证了较高的速度,同时在密集障碍物中通过,路径安全性高. 与图12(b)对比,图12(c)兼顾了安全性的同时,运行步数、时间都得到降低.

图12(d)是3种不同权值的运行轨迹图,其中$\gamma = 2$与$\gamma = \gamma_d$时机器人路径几乎重合,路径合理. 实验结果如表7所示.



(a) 固定高权值($\gamma = 20$)轨迹

(b) 固定低权值($\gamma = 2$)轨迹

(c) 动态权值γ_d轨迹

(d) 不同权值轨迹对比

图 12 不同速度权值的复杂环境下运行效果和轨迹对比

表 7　图12算法改进前后数据对比

方法	安全性/(安全距离/m)	迭代次数/步	路程/m	总时间/s
固定$\gamma=20$	低（2.714 e-5）	167	15.414	16.978 6
固定$\gamma=2$	高（0.124 06）	330	15.716	33.416 2
动态γ_d	高（0.136 25）	263	15.626	26.597 4
同安全性$\gamma=2$与γ_d对比	几乎相同	降低20.30 %	降低0.57 %	降低20.41 %

由表7可以看出,算法改进后,在安全为首要的条件下,机器人从起始点到目标点运行的总体效率得到了明显提升,迭代次数降低了20.30 %,总运行时间下降了20.41 %.

图12和表7定性定量地对比了复杂环境下算法改进前后的综合性能. 为了验证算法在复杂应用场景中的普适性,下面对多种不同的复杂环境进行多次试验,由于篇幅限制,只展示不同权值轨迹的对比结果,如图13～图15所示,其他实验结果如表8～表10所示.

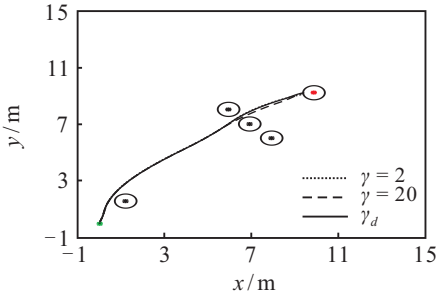


图 13　3种速度权值3个障碍物下运行效果和轨迹对比

为了定量对比算法改进前后在多种环境下的总体性能,将实验平均数据列入表11.

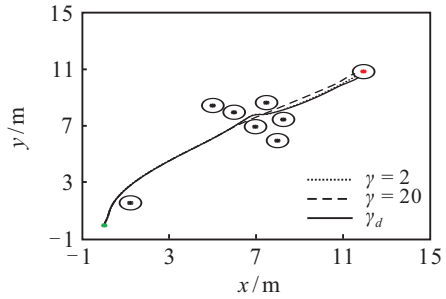


图 14　3种速度权值多个障碍物下运行效果和轨迹对比

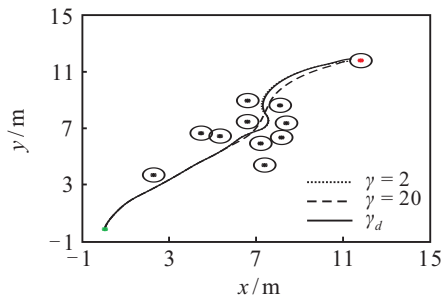


图 15　3种速度权值复杂环境下运行效果和轨迹对比

表 8　图13算法改进前后数据对比

方法	安全性(安全距离/m)	迭代次数/步	路程/m	总时间/s
固定$\gamma=20$	低（0.080 165）	148	13.592	15.077 8
固定$\gamma=2$	高（0.199 05）	223	13.646	22.587 5
动态γ_d	高（0.181 31）	173	13.686	17.602 3
同安全性$\gamma=2$与γ_d对比	几乎相同	降低22.42 %	增加0.29 %	降低22.07 %

表 9　图14算法改进前后数据对比

方法	安全性（安全距离/m)	迭代次数/步	路程/m	总时间/s
固定$\gamma=20$	低（0.054 89）	175	16.254	17.778 3
固定$\gamma=2$	高（0.137 42）	310	16.31	31.427 7
动态γ_d	高（0.136 01）	238	16.354	24.132 2
同安全性$\gamma=2$与γ_d对比	几乎相同	降低23.23 %	增加0.27 %	降低23.21 %

表 10　图15算法改进前后数据对比

方法	安全性(安全距离/m)	迭代次数/步	路程/m	总时间/s
固定$\gamma=20$	低（3.321 5 e-4）	183	17.06	18.578 3
固定$\gamma=2$	高（0.178 98）	303	17.746	30.532 9
动态γ_d	高（0.218 61）	258	17.756	14.47
同安全性$\gamma=2$与γ_d对比	几乎相同	降低23.23 %	增加0.27 %	降低23.21 %

表 11　优化效果平均总体性能对比

方法	安全性(安全距离/m)	平均迭代次数/步	平均路程/m	平均总时间/s
同安全性$\gamma=2$与γ_d对比	几乎相同	降低20.195 %	增加0.001 1 %	降低20.04 %

由表8～表11可以看出,自适应DWA算法在多种复杂环境下具有与表7同样的性能,与高权值相比可以大幅提高安全性,与低权值相比,运行步数、时间都有较大幅度的降低,其中平均总运行时间和迭代次数都下降了20%. 因此,在复杂环境下能够兼顾安全性和速度性.

2.3.3 算法改进前后轨迹平滑性对比

当权值γ固定时,在部分环境中可能存在绕行及轨迹平滑性较差的现象,如图16所示,当速度取值高时,从障碍物区域外围绕行;当权值低时,狭窄障碍物通道处轨迹平滑度较差,如图中A处所示;当使用动态权值时,可以从障碍物区域中间穿行且轨迹平滑度比较好.

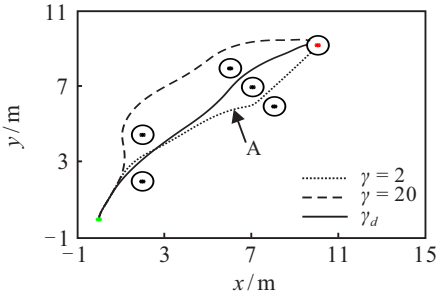


图16 3种速度权值复杂环境下平滑性效果对比

3 结 论

针对经典DWA存在速度性和安全性不能兼顾等问题,本文提出了参数自适应的DWA算法. 改进的算法通过自动调整速度权值自适应环境的变化,获得机器人运动的最佳速度及更加合理的路径,可明显改善原DWA方法穿越稠密障碍物时的性能. 通过实验验证了改进DWA算法的有效性和合理性,使机器人避障更加安全、快捷和可靠. 未来的研究方向是利用传感器信息估算运动障碍物的移动方向和速度,研究移动障碍物的DWA算法.

参考文献(References)

[1] 丛岩峰. 基于滚动优化原理的路径规划方法研究[D]. 吉林: 吉林大学通信工程学院, 2007.
(Cong Y F. The path planning method research based on rolling optimization theory[D]. Changchun: College of Communication Engineering, Jilin University, 2007.)

[2] 马仁利, 关正西. 路径规划技术的现状与发展综述[J]. 现代机械, 2008(3): 22-25.
(Ma R L, Guan Z X. Summarization for present situation and future development of path planning technology[J]. Modern Machinery, 2008(3): 22-25.)

[3] Borenstein J. The vector field histogram-fast obstacle avoidance for mobile robots[J]. IEEE Trans on Robotics & Automation, 2002, 7(3): 278-288.

[4] Ulrich I, Borenstein J. VFH+: Reliable obstacle avoidance for fast mobile robots[M]. Tamil Nadu: Jayalakshmi Institute of Technology, 1998: 1572-1577.

[5] Ulrich I, Borenstein J. VFH*: Local obstacle avoidance with look-ahead verification[C]. IEEE Int Conf on Robotics and Automation. San Francisco: IEEE, 2000: 2505-2511.

[6] Simmons R. The curvature-velocity method for local obstacle avoidance[C]. IEEE Int Conf on Robotics and Automation. Minneapolis: IEEE Xplore, 1996: 3375-3382.

[7] Fox D, Burgard W, Thrun S, et al. The dynamic window approach to collision avoidance[J]. Robotics & Automation Magazine, 1997, 4(1): 23-33.

[8] Seder M, Petrovic I. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles[C]. IEEE Int Conf on Robotics and Automation. Roma: IEEE, 2007: 1986-1991.

[9] Masahiro Shiomi. Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model[J]. Int J of Social Robotics, 2014, 6(3): 443-455.

[10] Doopalam Tuvshinjargal. Hybrid motion planning method for autonomous robots using kinect based sensor fusion and virtual plane approach in dynamic environments[J]. J of Sensors, 2015(5): 1-13.

[11] Ana Lopes A, Rodrigues J, Perdigao J, et al. A new hybrid motion planner: Applied in a brain-actuated robotic wheelchair[J]. IEEE Robotics & Automation Magazine, 2016, 23(4): 82-93.

[12] Beomjoon Kim. Socially adaptive path planning in human environments using inverse reinforcement learning[J]. Int J of Social Robotics, 2016, 8(1): 51-66.

[13] 曲道奎, 杜振军, 徐殿国, 等. 移动机器人路径规划方法研究[J]. 机器人, 2008, 30(2): 97-101.
(Qu D K, Du Z J, Xu D G, et al. Research on Path Planning for a mobile robot[J]. Robot, 2008, 30(2): 97-101.)

[14] Trautman P, Ma J, Murray R M, et al. Robot navigation in dense human crowds: The case for cooperation[C]. IEEE Int Conf on Robotics and Automation. Karlsruhe: IEEE, 2013: 2153-2160.

(责任编辑: 闫 妍)