



(12) 发明专利申请

(10) 申请公布号 CN 104020674 A

(43) 申请公布日 2014. 09. 03

(21) 申请号 201410160828. 3

(22) 申请日 2014. 04. 21

(71) 申请人 华南农业大学

地址 510642 广东省广州市天河区五山路
483 号

(72) 发明人 赵祚喜 李姣姣 黄培奎 谭志斌
施磊 曹文君 吴志伟 陈嘉琪
陈楚

(74) 专利代理机构 广州市华学知识产权代理有
限公司 44245

代理人 杨晓松

(51) Int. Cl.

G05B 17/02 (2006. 01)

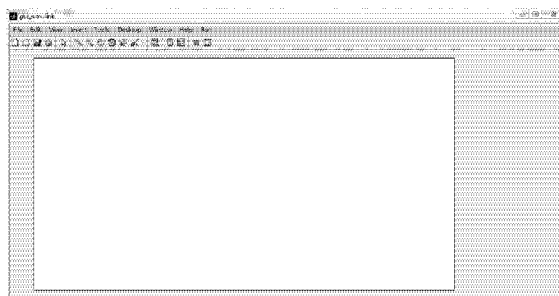
权利要求书3页 说明书11页 附图11页

(54) 发明名称

车辆 Bug 避障算法的 Matlab 仿真可视化平台

(57) 摘要

本发明公开了一种车辆 Bug 避障算法的 Matlab 仿真可视化平台,包括环境地图生成部分、Simulink 算法仿真实现部分、GUI 动画仿真效果界面,每一个部分均可由软件仿真或物理硬件实现,从而可应用到从纯软件仿真开始直至完整的物理机器人系统运行整个开发验证过程中。环境地图的生成可通过鼠标绘制、文件导入和外部传感器实时创建;Simulink 算法仿真实现部分,依赖于 Simulink 循环运行模式,实现算法内部循环和触发条件停止;GUI 动画仿真效果界面,主要实现车辆根据避障算法的运动模拟。通过对折线 Bug 与圆弧 Bug 算法,以及激光测距仪、全景相机等传感器实际验证,该平台操作简单友好,大大提高了车辆自主导航避障算法的研发效率。



1. 车辆 Bug 避障算法的 Matlab 仿真可视化平台,其特征在于,包括:

环境地图生成部分,用于生成仿真用的地图;

Simulink 算法仿真实现部分,算法实现依赖于 Simulink 循环运行模式,利用 Simulink 循环机制实现算法内部循环和触发条件停止,利用 Simulink 的自定义模块 Matlab Function 编写代码,分步实现算法的功能,封装成框图的形式,以供调用, Simulink 每循环一次,代表一次激光传感器对周围环境的判断,更新仿真参数,决定车辆进行下一个动作,这个动作包括转向/转弯或者行走一个单位距离,直到车辆停止,跳出 simulink 循环;

GUI 动画仿真效果界面,主要实现车辆根据避障算法的运动模拟,能够输入各种仿真参数,提供开始、结束按钮,能够进行仿真动画显示,图例标识,提供帮助系统。

2. 根据权利要求 1 所述的车辆 Bug 避障算法的 Matlab 仿真可视化平台,其特征在于,所述环境地图可通过以下三种方式生成:

1) 鼠标绘制,在 GUI 动画仿真效果界面直接通过鼠标绘制获得地图;

2) 文件导入,利用绘图工具制作地图,通过窗口导入到仿真界面;

3) 通过传感器采集环境数据,经过数据处理,实时生成所需的环境地图和障碍物信息。

3. 根据权利要求 2 所述的车辆 Bug 避障算法的 Matlab 仿真可视化平台,其特征在于:所述传感器有激光雷达或激光测距仪,通过对环境扫描可以获得环境数据,扫描时传感器需对多个位置进行扫描,在每个位置上检测到的障碍物数据所对应的是不同坐标,这时若要用一个坐标表示在地图上,则需要对其数据进行 2 维坐标变换,所述 2 维坐标变换公式有:

$$\begin{pmatrix} x_w^P \\ y_w^P \end{pmatrix} = k \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_A^P \\ y_A^P \end{pmatrix} + \begin{pmatrix} x_w^A \\ y_w^A \end{pmatrix}$$

式中, $(x_w^P \ y_w^P)_T$ 表示 P 点在 W 坐标系下的坐标值, $(x_A^P \ y_A^P)_T$ 表示 P 点在 A 坐标系下的坐标值, $(x_w^A \ y_w^A)_T$ 表示 A 坐标系相对于 W 坐标系的偏移, θ 则是 A 坐标系相对于 W 坐标系的转角。

4. 根据权利要求 1 所述的车辆 Bug 避障算法的 Matlab 仿真可视化平台,其特征在于:所述 Simulink 算法仿真实现部分包括有车辆行驶定位模块、传感器探测模块、车身碰撞探测模块、虚拟触角探测模块、转弯路径预测模块、行动决策模块、绘图模块,分别通过 Simulink 中自定义函数模块封装起来,定义输入输出端口,各个模块分工合作,模块内部修改参数,模块之间相互传递参数,利用 Simulink 循环机制实现算法内部循环和触发条件停止;其中,

所述车辆行驶定位模块,用于根据当前位姿和行动决策信息,确定下一个车辆的位姿;输入 input:当前小车位姿,行动决策模块输出的转弯半径和车速数据;输出 output:更新后的小车位姿;

所述传感器探测模块,用于按照真实传感器特性在地图上提取传感区域的障碍物点信息,把障碍物信息提供到其它探测模块;输入 input:当前小车位姿、传感器范围信息、全局地图信息;输出 output:传感器范围障碍物信息;

所述车身碰撞探测模块,用于检测当前小车区域是否与地图环境中的障碍物发生碰

撞,若发生碰撞将停止仿真;输入 input:当前小车位姿、全局地图信息、小车车身范围信息;输出 output:仿真运行状况标志,输出 1 则触发 stop,模块停止仿真;

所述虚拟触角探测模块,用于获取传感器探测模块的障碍物点数据,检测当前各虚拟触角内是否出现障碍物及障碍物到车身的距离;输入 input:当前小车位姿、传感器范围障碍物信息、虚拟触角范围信息;输出 output:两个虚拟触角状态标志,虚拟触角内最近障碍物到车身的距离;

所述转弯路径预测模块,根据 BUG 算法转弯路径选择机制,计算出车辆在各触角/触须,即转弯路线,行驶而脱离障碍物的时间,整合后提供给行动决策模块作分析;输入 input:当前小车位姿、传感器范围障碍物信息、传感器范围障碍物信息、触须离散点信息、虚拟触角范围信息、小车车身范围信息;输出 output:各触须的预测信息;

所述行动决策模块,为避障主算法所在模块,用于获取定位、各种探测和预测模块数据进行综合分析,决定车辆下一步应进入趋向目标还是绕障碍物模式,后者区分为避开转弯、接近转弯和直线行走方式,两种转弯方式都配合转弯路线选择机制确定转弯方向和转弯半径大小,获得的行动决策数据提供到车辆行驶定位模块以更新下一个车辆的位姿;输入 input:当前小车位姿、目标点信息、虚拟触角状态标志、障碍物距离、各触须的预测信息、部分初始化信息;输出 output:转弯半径、车速、仿真运行状况标志及其它状态标志;

所述绘图模块,用于获取车辆行驶状态信息,以绘制车身、虚拟触角、待转弯路线和运动轨迹,每采样周期更新车辆位置信息后,坐标转换得到各顶点或点集,再重新在地图上显示,需要注意的是车辆在转弯过程中选择的转弯路线将实时显示,方便观测小车转弯状态,同时记录和绘制起始点、目标点、碰撞点和脱离点;输入 input:当前小车位姿,起始点、目标点、碰撞点和脱离点信息,需要绘制的图形模块信息,部分初始化信息;输出 output:无。

5. 根据权利要求 1 所述的车辆 Bug 避障算法的 Matlab 仿真可视化平台,其特征在于:所述 GUI 动画仿真效果界面包括有四个部分:菜单栏、工具栏、动态显示窗口、注释部分,所述工具栏保留了原 Matlab 工具栏的功能,能够对进行文件保存、打印、放大缩小、旋转、标记,所述动态显示窗口能动画显示车辆利用 BUG 算法避开障碍物到达目的地的过程,所述注释部分标注了行走轨迹、起始点、脱离点、相遇点和目标点标记;其中,所述菜单栏保留了原 Matlab 图形窗口的基本工具栏和菜单栏,并在主菜单 Run 下新增以下功能:

1) 算法选择,根据需要通过子菜单项来选择需要运行的算法;

2) 参数设置,根据算法需求设置参数,参数包括有:

2.1) 行走距离:行走距离即起始点和目标点之间的直线距离,行走距离默认值是 280,行走距离的设置不能超过地图的范围,不然将导致目标点不可观测;

2.2) 车身长度和宽度:因为当地图导入仿真器中时,比例会有一定的变化,实际车身长度和宽度需根据地图的比例来设计,默认值是 30、24;

2.3) 传感器范围:默认值 150,此处用于判断车辆可直线行走的距离;

2.4) 行走步长:车体每经过一次激光传感器检测行走的距离,默认值 0.02;

2.5) 大、小圆弧虚拟触角半径:虚拟触角位于车载测距传感器探测范围内,其作用是利用测距传感器输入的实时环境信息数据判断是否有障碍物位于这些区域,此处的虚拟触角为半圆形区域,圆弧虚拟触角呈半圆形状,大圆弧虚拟触角半径大于小圆弧虚拟触角半径,小圆弧虚拟触角半径等于车辆半径,即车辆宽度的一半,虚拟触角半径在仿真平台默认值

分别为 26、24；

2.6) 矩形虚拟触角：此处虚拟触角为矩形区域，矩形虚拟触角呈矩形状，大、小圆弧虚拟触角都包含在矩形虚拟触角范围内，因此矩形虚拟触角长度大于大圆弧虚拟触角半径，默认值是长度是 80，宽度为大圆弧虚拟触角直径；

2.7) 参数 ρ ： ρ 是影响因数， ρ 的取值与触须分布状态有关， ρ 越大，触须越往中间靠拢，两旁的触须显得稀疏；反之，触须越往中间两旁扩散，中间的触须显得稀疏甚至不存在；根据不同车速范围取为不同值，以使探测的灵敏区域有所侧重：如车速越快，前方路面较宽畅，行车注意力应主要放在前方中间区域，则应使触须集中在相应区域， ρ 值较大；反之车速越慢，表面前方有障碍物，周围环境复杂，为避开前方障碍物行车注意力不仅在前方中间，更需要在前方两旁区域，则应使触须均匀分布， ρ 值相对较小，默认值是 1.3；

2.8) 最小转弯半径：根据车辆的转弯半径按比例设置，默认值是 60；

2.9) 大、小矩形虚拟触角：大矩形虚拟触角半径大于小矩形虚拟触角半径，小圆矩形拟触角半径等于车辆半径，即车辆宽度的一半，默认值宽度分别为 32、38，长度为 65；

3) 地图导入和地图绘制功能，通过地图导入选项选择现有地图，导入的地图黑色区域表示障碍物或边界，白色区域表示可到达地方，定义 X、Y 坐标轴的方向和航向方向，X 正半轴是 0° ，X 负半轴是 -180° 与 $+180^\circ$ 交界；小车行驶方向与 X 正半轴的夹角为 θ ，即小车航向，将图像转换成灰度图像，将其图片信息数据化保存在 400×640 的矩阵中，菜单栏中的地图绘制功能能够通过鼠标绘制获得地图以供仿真使用；

4) 帮助系统，为 Run 主菜单下的用户手册，为初次使用者提供指导，对平台搭建使用的工具做详细介绍，使用的相关函数做详细介绍，并为今后的改进完善提出建议；为方便用户阅读该帮助系统，所述帮助系统分为五大块，分别为入门指南、用户手册、实例说明、函数介绍、仿真工具箱功能构想；所述帮助系统是通过 QuickCHM 软件编译集成，利用 word 保存的网页文件，导入到 QuickCHM，经过目录安排，排版编辑，编译后就可生成需要的 chm 文件，再将该文件利用到 matlab GUI 中；在帮助系统中能够分层浏览，并用图文说明了该软件的功能，其中用到的算法，以及操作方法；除了通过点击左边的在主目录和子目录中跳到相应的页面，还提供了超链接，可以在各个网页件切换。

6. 根据权利要求 5 所述的车辆 Bug 避障算法的 Matlab 仿真可视化平台，其特征在于，所述参数设置有两种方法：一种是通过输入对话框进行设置，一种是通过鼠标绘制；输入对话框设置能够量化参数，直接在对话框内输入参数即可；鼠标绘制可以通过鼠标点击来绘制车身大小，触角大小，分别对折线 BUG 算法和圆弧 BUG 算法进行鼠标绘制，可以在选择需要仿真的地图进行。

车辆 Bug 避障算法的 Matlab 仿真可视化平台

技术领域

[0001] 本发明涉及计算机仿真的技术领域,尤其是指一种车辆 Bug 避障算法的 Matlab 仿真可视化平台。

背景技术

[0002] 对于避障方法国内外学者早已进行了大量的研究,并且提出了许多可行的方法,比如 BUG 算法、人工势场法、向量场直方图法、动态窗口法、可视图法、切线图法等等。在算法的研究时必须对其效果进行验证,可以通过两种方法,一种是通过实验验证,一种是通过计算机仿真验证。实验验证需要直面实验过程中带来的干扰,比如获取障碍物信息的精确度,在机器人或者车辆运行过程中转向的误差,定位误差等,这些干扰与算法本身的正确性关系不大,却能直接影响到实验结果,很难排除是干扰导致结果的偏差还是算法本身的缺陷,同时实验验证需要消耗人力物力,比如实验器材,实验场景,实验所花费的时间等。目前大多数研究者采用计算机仿真,或者通过专用的机器人仿真。通过仿真验证算法的有效性后,再进一步在真实车辆中实验验证算法的鲁棒性和抗干扰能力。

[0003] 计算机仿真是指以计算机为主要工具,运行真实系统或预研系统的仿真模型,通过对计算机输出信息的分析与研究,实现对实际系统运行状态和演化规律的综合评估与预测。目前大多数研究者采用计算机仿真,他们结合所用的算法特点,为了验证试验效果开辟出了一套自己的仿真方法。目前应用较广的是机器人仿真软件 Mobilesim, Mobilesim 是用户定制的能够扩展到所有的机器人及其附件的仿真工具平台,该平台能够接入各种传感器,结合 VS 调试平台,能够直接看到机器人行走情况,但其算法必须借助 Aria 库函数编程,实现起来复杂。为了脱离 Aria 平台,现很多算法研究者使用 Matlab 来进行算法仿真实现,利用 matlab 强大的计算和绘图功能、大量稳定可靠的算法库和简洁有效的编程语言,算法实现起来容易。

发明内容

[0004] 本发明的目的在于克服现有技术的不足与缺点,提供一种车辆 Bug 避障算法的 Matlab 仿真可视化平台,能更好地验证避障算法的有效性和可行性。

[0005] 为实现上述目的,本发明所提供的技术方案为:车辆 Bug 避障算法的 Matlab 仿真可视化平台,包括:

[0006] 环境地图生成部分,用于生成仿真用的地图;

[0007] Simulink 算法仿真实现部分,算法实现依赖于 Simulink 循环运行模式,利用 Simulink 循环机制实现算法内部循环和触发条件停止,利用 Simulink 的自定义模块 Matlab Function 编写代码,分步实现算法的功能,封装成框图的形式,以供调用, Simulink 每循环一次,代表一次激光传感器对周围环境的判断,更新仿真参数,决定车辆进行下一个动作,这个动作包括转向/转弯或者行走一个单位距离,直到车辆停止,跳出 simulink 循环;

[0008] GUI 动画仿真效果界面, 主要实现车辆根据避障算法的运动模拟, 能够输入各种仿真参数, 提供开始、结束按钮, 能够进行仿真动画显示, 图例标识, 提供帮助系统。

[0009] 所述环境地图可通过以下三种方式生成:

[0010] 1) 鼠标绘制, 在 GUI 动画仿真效果界面直接通过鼠标绘制获得地图;

[0011] 2) 文件导入, 利用绘图工具制作地图, 通过窗口导入到仿真界面;

[0012] 3) 通过传感器采集环境数据, 经过数据处理, 实时生成所需的环境地图和障碍物信息。

[0013] 所述传感器有激光雷达或激光测距仪, 通过对环境扫描可以获得环境数据, 扫描时传感器需对多个位置进行扫描, 在每个位置上检测到的障碍物数据所对应的是不同坐标, 这时若要用一个坐标表示在地图上, 则需要对其数据进行 2 维坐标变换, 所述 2 维坐标变换公式有:

$$[0014] \quad \begin{pmatrix} x_W^P \\ y_W^P \end{pmatrix} = k \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_A^P \\ y_A^P \end{pmatrix} + \begin{pmatrix} x_W^A \\ y_W^A \end{pmatrix}$$

[0015] 式中, $(x_W^P \ y_W^P)^T$ 表示 P 点在 W 坐标系下的坐标值, $(x_A^P \ y_A^P)^T$ 表示 P 点在 A 坐标系下的坐标值, $(x_W^A \ y_W^A)^T$ 表示 A 坐标系相对于 W 坐标系的偏移, θ 则是 A 坐标系相对于 W 坐标系的转角。

[0016] 所述 Simulink 算法仿真实现部分包括有车辆行驶定位模块、传感器探测模块、车身碰撞探测模块、虚拟触角探测模块、转弯路径预测模块、行动决策模块、绘图模块, 分别通过 Simulink 中自定义函数模块封装起来, 定义输入输出端口, 各个模块分工合作, 模块内部修改参数, 模块之间相互传递参数, 利用 Simulink 循环机制实现算法内部循环和触发条件停止; 其中,

[0017] 所述车辆行驶定位模块, 用于根据当前位姿和行动决策信息, 确定下一个车辆的位姿; 输入 input: 当前小车位姿, 行动决策模块输出的转弯半径和车速数据; 输出 output: 更新后的小车位姿;

[0018] 所述传感器探测模块, 用于按照真实传感器特性在地图上提取传感区域的障碍物点信息, 把障碍物信息提供到其它探测模块; 输入 input: 当前小车位姿、传感器范围信息、全局地图信息; 输出 output: 传感器范围障碍物信息;

[0019] 所述车身碰撞探测模块, 用于检测当前小车区域是否与地图环境中的障碍物发生碰撞, 若发生碰撞将停止仿真; 输入 input: 当前小车位姿、全局地图信息、小车车身范围信息; 输出 output: 仿真运行状况标志, 输出 1 则触发 stop, 模块停止仿真;

[0020] 所述虚拟触角探测模块, 用于获取传感器探测模块的障碍物点数据, 检测当前各虚拟触角内是否出现障碍物及障碍物到车身的距离; 输入 input: 当前小车位姿、传感器范围障碍物信息、虚拟触角范围信息; 输出 output: 两个虚拟触角状态标志, 虚拟触角内最近障碍物到车身的距离;

[0021] 所述转弯路径预测模块, 根据 BUG 算法转弯路径选择机制, 计算出车辆在各触角/触须, 即转弯路线, 行驶而脱离障碍物的时间, 整合后提供给行动决策模块作分析; 输入 input: 当前小车位姿、传感器范围障碍物信息、传感器范围障碍物信息、触须离散点信息、

虚拟触角范围信息、小车车身范围信息 ;输出 output :各触须的预测信息 ;

[0022] 所述行动决策模块,为避障主算法所在模块,用于获取定位、各种探测和预测模块数据进行综合分析,决定车辆下一步应进入趋向目标还是绕障碍物模式,后者区分为避开转弯、接近转弯和直线行走方式,两种转弯方式都配合转弯路线选择机制确定转弯方向和转弯半径大小,获得的行动决策数据提供到车辆行驶定位模块以更新下一个车辆的位姿 ;输入 input :当前小车位姿、目标点信息、虚拟触角状态标志、障碍物距离、各触须的预测信息、部分初始化信息 ;输出 output :转弯半径、车速、仿真运行状况标志及其它状态标志 ;

[0023] 所述绘图模块,用于获取车辆行驶状态信息,以绘制车身、虚拟触角、待转弯路线和运动轨迹,每采样周期更新车辆位置信息后,坐标转换得到各顶点或点集,再重新在地图上显示,需要注意的是车辆在转弯过程中选择的转弯路线将实时显示,方便观测小车转弯状态,同时记录和绘制起始点、目标点、碰撞点和脱离点 ;输入 input :当前小车位姿,起始点、目标点、碰撞点和脱离点信息,需要绘制的图形模块信息,部分初始化信息 ;输出 output :无。

[0024] 所述 GUI 动画仿真效果界面包括有四个部分 :菜单栏、工具栏、动态显示窗口、注释部分,所述工具栏保留了原 Matlab 工具栏的功能,能够对进行文件保存、打印、放大缩小、旋转、标记,所述动态显示窗口能动画显示车辆利用 BUG 算法避开障碍物到达目的地的过程,所述注释部分标注了行走轨迹、起始点、脱离点、相遇点和目标点标记 ;其中,所述菜单栏保留了原 Matlab 图形窗口的基本工具栏和菜单栏,并在主菜单 Run 下新增以下功能 :

[0025] 1) 算法选择,根据需要通过子菜单项来选择需要运行的算法 ;

[0026] 2) 参数设置,根据算法需求设置参数,参数包括有 :

[0027] 2.1) 行走距离 :行走距离即起始点和目标点之间的直线距离,行走距离默认值是 280,行走距离的设置不能超过地图的范围,不然将导致目标点不可观测 ;

[0028] 2.2) 车身长度和宽度 :因为当地图导入仿真器中时,比例会有一些的变化,实际车身长度和宽度需根据地图的比例来设计,默认值是 30、24 ;

[0029] 2.3) 传感器范围 :默认值 150,此处用于判断车辆可直线行走的距离 ;

[0030] 2.4) 行走步长 :车体每经过一次激光传感器检测行走的距离,默认值 0.02 ;

[0031] 2.5) 大、小圆弧虚拟触角半径 :虚拟触角位于车载测距传感器探测范围内,其作用是利用测距传感器输入的实时环境信息数据判断是否有障碍物位于这些区域,此处的虚拟触角为半圆形区域,圆弧虚拟触角呈半圆形状,大圆弧虚拟触角半径大于小圆弧虚拟触角半径,小圆弧虚拟触角半径等于车辆半径,即车辆宽度的一半,虚拟触角半径在仿真平台默认值分别为 26、24 ;

[0032] 2.6) 矩形虚拟触角 :此处虚拟触角为矩形区域,矩形虚拟触角呈矩形状,大、小圆弧虚拟触角都包含在矩形虚拟触角范围内,因此矩形虚拟触角长度大于大圆弧虚拟触角半径,默认值是长度是 80,宽度为大圆弧虚拟触角直径 ;

[0033] 2.7) 参数 ρ : ρ 是影响因数, ρ 的取值与触须分布状态有关, ρ 越大,触须越往中间靠拢,两旁的触须显得稀疏 ;反之,触须越往中间两旁扩散,中间的触须显得稀疏甚至不存在 ;根据不同车速范围取为不同值,以使探测的灵敏区域有所侧重 :如车速越快,前方路面较宽畅,行车注意力应主要放在前方中间区域,则应使触须集中在相应区域, ρ 值较大 ;反之车速越慢,表面前方有障碍物,周围环境复杂,为避开前方障碍物行车注意力不仅在前

方中间,更需要在前方两旁区域,则应使触须均匀分布, ρ 值相对较小,默认值是 1.3;

[0034] 2.8) 最小转弯半径:根据车辆的转弯半径按比例设置,默认值是 60;

[0035] 2.9) 大、小矩形虚拟触角:大矩形虚拟触角半径大于小矩形虚拟触角半径,小圆矩形虚拟触角半径等于车辆半径,即车辆宽度的一半,默认值宽度分别为 32、38,长度为 65;

[0036] 3) 地图导入和地图绘制功能,通过地图导入选项选择现有地图,导入的地图黑色区域表示障碍物或边界,白色区域表示可到达地方,定义 X, Y 坐标轴的方向和航向方向, X 正半轴是 0° , X 负半轴是 -180° 与 $+180^\circ$ 交界;小车行驶方向与 X 正半轴的夹角为 θ ,即小车航向,将图像转换成灰度图像,将其图片信息数据化保存在 400×640 的矩阵中,菜单栏中的地图绘制功能能够通过鼠标绘制获得地图以供仿真使用;

[0037] 4) 帮助系统,为 Run 主菜单下的用户手册,为初次使用者提供指导,对平台搭建使用的工具做详细介绍,使用的相关函数做详细介绍,并为今后的改进完善提出建议;为方便用户阅读该帮助系统,所述帮助系统分为五大块,分别为入门指南、用户手册、实例说明、函数介绍、仿真工具箱功能构想;所述帮助系统是通过 QuickCHM 软件编译集成,利用 word 保存的网页文件,导入到 QuickCHM,经过目录安排,排版编辑,编译后就可生成需要的 chm 文件,再将该文件利用到 matlab GUI 中;在帮助系统中能够分层浏览,并用图文说明了该软件的功能,其中用到的算法,以及操作方法;除了通过点击左边的在主目录和子目录中跳到相应的页面,还提供了超链接,可以在各个网页件切换。

[0038] 所述参数设置有两种方法:一种是通过输入对话框进行设置,一种是通过鼠标绘制;输入对话框设置能够量化参数,直接在对话框内输入参数即可;鼠标绘制可以通过鼠标点击来绘制车身大小,触角大小,分别对折线 BUG 算法和圆弧 BUG 算法进行鼠标绘制,可以在选择需要仿真的地图进行。

[0039] 本发明与现有技术相比,具有如下优点与有益效果:

[0040] 1、利用 MATLAB 软件创建了 Simulink 仿真技术和 GUI 界面设计相结合下的车辆避障仿真可视化平台,能够对避障算法进行动画仿真;充分利用 Simulink 和 GUI 各自强大的功能,将二者结合起来实现车辆避障的动态仿真,具有模块化设计,编程简单,数据输入方便,界面交互性强的特点;打开 GUI 动画仿真效果界面时读取仿真参数默认值并显示在参数输入框里,参数可在本次仿真运行前修改,能够实现车身大小和触角大小的参数修改;运行后首先存入运行内存中完成仿真的初始化,并调用 Simulink 文件运行仿真测试,仿真动画实时显示在绘图窗口上,后台记录和保存相关数据;避障算法运行到结束条件后退出 Simulink 仿真;整个仿真能够准确、直观地模拟出小车的避障过程,检验避障算法性能;

[0041] 2、传感器与仿真效果展示既可纯软件仿真,也可连接实际的导航传感器与车辆,完成硬软件结合的半实物仿真;Simulink 仿真技术和 GUI 界面设计相结合下的车辆避障仿真可视化平台能够进行连接车辆,或者只接传感器,或者传感器和车辆同时连接,虚拟和现实相结合,通过最大限度的模拟现实,能够解决更多在现实状况中遇到的问题,为算法的改进提供指导。

附图说明

[0042] 图 1 为激光测距仪在两个不同方位对同一个障碍物的扫描示意图。

[0043] 图 2 为物体位置示意图。

- [0044] 图 3 为障碍物点云图。
- [0045] 图 4 为车辆避障仿真的 Simulink 框图模型设计图。
- [0046] 图 5a 为折线 BUG 鼠标绘制虚拟触角图。
- [0047] 图 5b 为圆弧 BUG 鼠标绘制虚拟触角图。
- [0048] 图 6 为仿真环境地图的坐标系图。
- [0049] 图 7 为绘图窗口图。
- [0050] 图 8 为绘图窗口绘制图形效果图。
- [0051] 图 9 为帮助系统界面图。
- [0052] 图 10 为仿真平台界面图。
- [0053] 图 11 为动画仿真界面图。
- [0054] 图 12a 为折线 BUG 算法虚拟触角设计图。
- [0055] 图 12b 为折线 BUG 算法程序框图。
- [0056] 图 13a 为车体与触须分布状态示意图。
- [0057] 图 13b 为圆弧 BUG 算法流程图。

具体实施方式

[0058] 下面结合具体实施例对本发明作进一步说明。

[0059] 本实施例所述的车辆 Bug 避障算法的 Matlab 仿真可视化平台,是利用 MATLAB 软件创建了 Simulink 仿真技术和 GUI 界面设计相结合下的车辆避障仿真可视化平台,如图 10 和图 11 所示,能够对避障算法进行动画仿真。它包括有:

[0060] 1) 环境地图生成部分,用于生成仿真用的地图;

[0061] 2) Simulink 算法仿真实现部分,算法实现依赖于 Simulink 循环运行模式,利用 Simulink 循环机制实现算法内部循环和触发条件停止,利用 Simulink 的自定义模块 Matlab Function 编写代码,分步实现算法的功能,封装成框图的形式,以供调用, Simulink 每循环一次,代表一次激光传感器对周围环境的判断,更新仿真参数,决定车辆进行下一个动作,这个动作包括转向 / 转弯或者行走一个单位距离,直到车辆停止,跳出 simulink 循环。

[0062] 3) GUI 动画仿真效果界面,主要实现车辆根据避障算法的运动模拟,能够输入各种仿真参数,提供开始、结束按钮,能够进行仿真动画显示,图例标识,提供帮助系统;该界面遵循软件开发模式设计的,最大程度地做到美观和方便操作,帮助设计者提供一个简洁易操作的界面,仿真结果易于观察,对算法的优劣比较,算法的改进具有很大的指导作用。

[0063] 所述环境地图可通过以下三种方式生成:

[0064] 1) 鼠标绘制,本发明可直接通过菜单栏中的用户绘图项目进行绘制,通过数据画图如图 7 所示,保存为 .png 文件,再通过仿真窗口导入,如图 8 所示。

[0065] 2) 文件导入,利用画图软件获得黑白地图,保存为 .png 文件后能够直接导入到仿真界面。

[0066] 3) 通过传感器采集环境数据,经过数据处理,实时生成所需的环境地图和障碍物信息。

[0067] 所述传感器有激光雷达或激光测距仪,通过对环境扫描可以获得环境数据,扫描

时传感器需对多个位置进行扫描,在每个位置上检测到的障碍物数据所对应的是不同坐标,这时若要用一个坐标表示在地图上,则需要对其数据进行 2 维坐标变换,所述 2 维坐标变换公式有:

$$[0068] \quad \begin{pmatrix} x_W^P \\ y_W^P \end{pmatrix} = k \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_A^P \\ y_A^P \end{pmatrix} + \begin{pmatrix} x_W^A \\ y_W^A \end{pmatrix}$$

[0069] 式中, $(x_W^P \ y_W^P)^T$ 表示 P 点在 W 坐标系下的坐标值, $(x_A^P \ y_A^P)^T$ 表示 P 点在 A 坐标系下的坐标值, $(x_W^A \ y_W^A)^T$ 表示 A 坐标系相对于 W 坐标系的偏移, θ 则是 A 坐标系相对于 W 坐标系的转角。

[0070] 如图 1 所示,位置 A 的坐标系为 -90° ; k 表示缩放系数,这里的 k 值取 1,不进行缩放。

[0071] 为使实验精确,本实施例通过三个位置,三个角度进行采集,采集的场地如图 2 所示,采集成成的效果如图 3 所示。

[0072] 所述 Simulink 算法仿真实现部分包括有车辆行驶定位模块、传感器探测模块、车身碰撞探测模块、虚拟触角探测模块、转弯路径预测模块、行动决策模块、绘图模块,分别通过 Simulink 中自定义函数模块封装起来,定义输入输出端口,各个模块分工合作,模块内部修改参数,模块之间相互传递参数,利用 Simulink 循环机制实现算法内部循环和触发条件停止。各模块之间通过单箭头连线的形式传递参数或者通过工作内存空间获取数据,模块的参数可以通过 Edit Data/Port 进行配置,比如输入参数定义为 input,输出参数定义为 output,如果是工作空间的参数定义为 Parameter。Embedded MATLAB Function 模块位于 Simulink/User-Defined Functions 模块库中。该模块工作于 matlab 语言的一个子集(称为 embedded matlab subset,不在此子集的函数,需要用 eml.extrinsic 声明或采用 feval 才能使用),该模块能够提供高效的代码(编译后高效)。在定义 Matlab Function 不支持的函数时,需要做一个说明,matlab2010 版本注意第 2 行的“%#eml”不可省略,matlab2011b 版本注意第 2 行的“%#codegen”不可省略。其中:

[0073] 所述车辆行驶定位模块(drive),用于根据当前位姿和行动决策信息,确定下一个车辆的位姿;输入 input:当前小车位姿,行动决策模块输出的转弯半径和车速数据;输出 output:更新后的小车位姿。

[0074] 所述传感器探测模块(maxrange),用于按照真实传感器特性(如采用 180° 范围传感器)在地图上提取传感区域的障碍物点信息,把障碍物信息提供到其它探测模块;输入 input:当前小车位姿、传感器范围信息、全局地图信息;输出 output:传感器范围障碍物信息。

[0075] 所述车身碰撞探测模块(collision),用于检测当前小车区域是否与地图环境中的障碍物发生碰撞,若发生碰撞将停止仿真;输入 input:当前小车位姿、全局地图信息、小车车身范围信息;输出 output:仿真运行状况标志,输出“1”则触发“stop”,模块停止仿真。

[0076] 所述虚拟触角探测模块(antenna),用于获取传感器探测模块的障碍物点数据,检测当前各虚拟触角内是否出现障碍物及障碍物到车身的距离;输入 input:当前小车位姿、传感器范围障碍物信息、虚拟触角范围信息;输出 output:两个虚拟触角状态标志,虚拟触

角内最近障碍物到车身的距离。

[0077] 所述转弯路径预测模块(tentacle),根据 BUG 算法转弯路径选择机制相关内容,计算出车辆在各触角/触须,即转弯路线,行驶而脱离障碍物的时间,整合后提供给行动决策模块作分析;输入 input:当前小车位姿、传感器范围障碍物信息、传感器范围障碍物信息、触须离散点信息、虚拟触角范围信息、小车车身范围信息等;输出 output:各触须的预测信息。

[0078] 所述行动决策模块(action),为避障主算法所在模块,用于获取定位、各种探测和预测模块数据进行综合分析,决定车辆下一步应进入“趋向目标”还是“绕障碍物”模式,后者区分为避开转弯、接近转弯和直线行走方式,两种转弯方式都配合“转弯路线选择机制”确定转弯方向和转弯半径大小,获得的行动决策数据提供到车辆行驶定位模块以更新下一个车辆的位姿;输入 input:当前小车位姿、目标点信息、虚拟触角状态标志、障碍物距离、各触须的预测信息、部分初始化信息;输出 output:转弯半径、车速、仿真运行状况标志及其它状态标志。

[0079] 所述绘图模块(drawbot),用于获取车辆行驶状态信息,以绘制车身、虚拟触角、待转弯路线和运动轨迹,每采样周期更新车辆位置信息后,坐标转换得到各顶点或点集,再重新在地图上显示,需要注意的是车辆在转弯过程中选择的转弯路线将实时显示,方便观测小车转弯状态,同时记录和绘制起始点、目标点、碰撞点和脱离点;输入 input:当前小车位姿,起始点、目标点、碰撞点和脱离点信息,需要绘制的图形模块信息,部分初始化信息;输出 output:无。

[0080] 本发明所述平台基于 BUG 算法来搭建,原则上能够适用于其他算法,BUG 算法是最简单的避障方法,其思想是车辆路途中,跟踪各障碍物的轮廓,从而绕开它,到达目的地。算法实现依赖于 Simulink 循环运行模式,利用 Simulink 的自定义模块 MatlabFunction(Matlab2012a 版本前名为 Embedded Matlab Function)编写代码,分步实现算法的功能,封装成框图的形式,如图 4 所示,以供调用。

[0081] 所述 GUI 动画仿真效果界面包括有四个部分:菜单栏、工具栏、动态显示窗口、注释部分,所述工具栏保留了原 Matlab 工具栏的功能,能够对进行文件保存、打印、放大缩小、旋转、标记,所述动态显示窗口能动画显示车辆利用 BUG 算法避开障碍物到达目的地的过程,所述注释部分标注了行走轨迹、起始点、脱离点、相遇点和目标点标记;其中,所述菜单栏保留了原 Matlab 图形窗口的基本工具栏和菜单栏,并在主菜单 Run 下新增以下功能:

[0082] 1) 算法选择,根据需要通过子菜单项来选择需要运行的算法。

[0083] 2) 参数设置,根据算法需求设置参数,参数包括有:

[0084] 2.1) 行走距离:行走距离即起始点和目标点之间的直线距离,行走距离默认值是 280,行走距离的设置不能超过地图的范围,不然将导致目标点不可观测;

[0085] 2.2) 车身长度和宽度:因为当地图导入仿真器中时,比例会有一定的变化,实际车身长度和宽度需根据地图的比例来设计,默认值是 30、24;

[0086] 2.3) 传感器范围:默认值 150,此处用于判断车辆可直线行走的距离;

[0087] 2.4) 行走步长:车体每经过一次激光传感器检测行走的距离,默认值 0.02;

[0088] 2.5) 大、小圆弧虚拟触角半径:虚拟触角位于车载测距传感器探测范围内,其作用是利用测距传感器输入的实时环境信息数据判断是否有障碍物位于这些区域,此处的虚拟

触角为半圆形区域,圆弧虚拟触角呈半圆形状,大圆弧虚拟触角半径大于小圆弧虚拟触角半径,小圆弧虚拟触角半径等于车辆半径,即车辆宽度的一半,虚拟触角半径在仿真平台默认值分别为 26、24;

[0089] 2.6) 矩形虚拟触角:此处虚拟触角为矩形区域,矩形虚拟触角呈矩形状,大、小圆弧虚拟触角都包含在矩形虚拟触角范围内,因此矩形虚拟触角长度大于大圆弧虚拟触角半径,默认值是长度是 80,宽度为大圆弧虚拟触角直径;

[0090] 2.7) 参数 ρ : ρ 是影响因数, ρ 的取值与触须分布状态有关, ρ 越大,触须越往中间靠拢,两旁的触须显得稀疏;反之,触须越往中间两旁扩散,中间的触须显得稀疏甚至不存在;根据不同车速范围取为不同值,以使探测的灵敏区域有所侧重:如车速越快,前方路面较宽敞,行车注意力应主要放在前方中间区域,则应使触须集中在相应区域, ρ 值较大;反之车速越慢,表面前方有障碍物,周围环境复杂,为避开前方障碍物行车注意力不仅在前方中间,更需要在前方两旁区域,则应使触须均匀分布, ρ 值相对较小,默认值是 1.3;

[0091] 2.8) 最小转弯半径:根据车辆的转弯半径按比例设置,默认值是 60;

[0092] 2.9) 大、小矩形虚拟触角:大矩形虚拟触角半径大于小矩形虚拟触角半径,小圆矩形虚拟触角半径等于车辆半径,即车辆宽度的一半,默认值宽度分别为 32、38,长度为 65;

[0093] 参数设置有两种方法:一种是通过输入对话框进行设置,另一种是通过鼠标绘制。输入对话框设置能够量化参数,直接在对话框内输入参数即可。鼠标绘制可以通过鼠标点击来绘制车身大小,触角大小,分别对折线 BUG 算法和圆弧 BUG 算法进行鼠标绘制,如图 5a 和 5b 所示。绘制矩形车体大小,通过起始点与鼠标点击点来获得,通过如下公式:

[0094] $vichleyuanhulong=2*|xv-posn(1)|$

[0095] $vichleyuanhuwidth=2*|yv-posn(2)|$

[0096] 能够获得车体长度和宽度, (posn(1), posn(2)) 为起始点, (xv, yv) 为绘制车体时鼠标点击点。

[0097] 通过如下公式:

[0098] $W1=((xDs-posn(1))^2+(yDs-posn(2))^2)^{0.5};$

[0099] $W2=((xDb-posn(1))^2+(yDb-posn(2))^2)^{0.5};$

[0100] 能够获得大、小圆弧触角的半径分别为 W2、W1, (xDs, yDs) 设计小触角时鼠标点击点, (xDb, yDb) 设计大触角时鼠标点击点。

[0101] 通过如下公式:

[0102] $vichleyuanhuW1=2*|yDs-posn(2)|;$

[0103] $vichleyuanhuW2=2*|yDb-posn(2)|;$

[0104] 分别获得两个大小矩形触角的宽度, yDs 和 yDb 分别为绘制大小矩形触角时鼠标点击点的 y 坐标值。

[0105] 通过如下公式:

[0106] $vichleyuanhuL=|xDs-xv|$

[0107] 获得矩形触角的长度, xDs 为绘制大小触角是鼠标点击点的 x 坐标值, 由于两个矩形触角的长度相同, 所以 xDs 等于 xDb。

[0108] 通过鼠标绘制来设置参数,可以在选择需要仿真的地图进行,满足了仿真的可控可观性需求。

[0109] 3) 地图导入和地图绘制功能, 本发明能够通过地图导入选项选择现有地图。导入的地图黑色区域表示障碍物或边界, 白色区域表示可到达地方。设定如图 6 所示的坐标系, 定义了 X, Y 坐标轴的方向和航向方向, X 正半轴是 0° , X 负半轴是 -180° 与 $+180^{\circ}$ 交界; 图中 θ 表示小车行驶方向与 X 正半轴的夹角, 即小车航向, 将图像转换成灰度图像, 将其图片信息数据化保存在 400×640 的矩阵中。菜单栏中的地图绘制功能能够通过鼠标绘制获得地图以供仿真使用。

[0110] 4) 帮助系统, 为 Run 主菜单下的用户手册, 为初次使用者提供指导, 对平台搭建使用的工具做详细介绍, 使用的相关函数做详细介绍, 并为今后的改进完善提出建议, 如图 9 所示。为方便用户阅读该帮助系统, 所述帮助系统分为五大块, 分别为入门指南、用户手册、实例说明、函数介绍、仿真工具箱功能构想。

[0111] 入门指南中包含软件的概述、操作注意事项。操作注意事项说明操作过程中会出现的一些情况, 包括操作规范等, 比如初始化需要等待一段时间才能进行仿真, 一些快捷键的使用。软件概述主要介绍开发过程中需要使用的工具等, 如 Matlab, GUI, Simulink, QuickCHM 等。

[0112] 用户手册中介绍了试验中用到的两种算法, 分别为圆弧 BUG 算法和折线 BUG 算法; simulink 模块介绍, 各个模块的作用; simulink 与 GUI 连接方法和坐标转换。

[0113] 实例说明中举了两个实例分别为折线 BUG 仿真和圆弧 BUG 仿真, 其中在 simulink 中和 GUI 用到的函数介绍, 仿真工具箱功能构想对需要实现的功能做了一个总结, 方便今后的完善。

[0114] 帮助系统是通过 QuickCHM 软件编译集成, 利用 word 保存的网页文件, 导入到 QuickCHM, 经过目录安排, 排版编辑, 编译后就可生成需要的 chm 文件, 再将该文件利用到 matlab GUI 中。

[0115] 在帮助系统中能够分层浏览, 并用图文说明了该软件的功能, 其中用到的算法, 以及操作方法; 除了通过点击左边的在主目录和子目录中跳到相应的页面, 还提供了超链接, 可以在各个网页件切换。

[0116] 该平台能够实现半实物仿真。Simulink 仿真技术和 GUI 界面设计相结合下的车辆避障仿真可视化平台能够进行连接车辆, 或者只接传感器, 或者传感器和车辆同时连接。在半实物仿真时, 传感器采集的地图, 通过图像处理之后保存到 course 数组 (使用 `course=rgb2gray(imread('X.png'))`) 语句将真彩色 RGB 图像转换成灰度图像, 将其图片信息数据化保存在 course 数组中, course 数组能够按照图像的大小来决定数组大小), 通过 course 数组时更新地图显示, 同时 course 数组数据传入 simulink。该处模拟传感器检测功能。通过传感器实时采集数据, 在 matlab 中实时绘制, 导航车辆在这种实时地图中行走, 在避障算法的帮助下避开障碍物, 到达目的地。该处模拟实际车辆在行走过程中传感器检测情况, 并且做出相应的动作。在虚拟仿真中的信号参数能够输出到车辆控制平台上, 转换成车辆控制信号对车辆的运动做出指导, 实现真实车辆运行轨迹跟仿真车辆轨迹的对比。

[0117] 备注: 本发明所用的两种 BUG 算法分别为一种适用于非 360° 探测机器人的避障导航算法和非 360° 探测范围四轮导航车辆的平滑路径避障算法, 分别简称为“折线 BUG 算法”和“圆弧 BUG 算法”, 下面对两种算法进行介绍:

[0118] 1) 折线 BUG 算法

[0119] BUG 类算法认为机器人是平面上的一点,具有 360° 障碍检测范围,具有贴着障碍物边缘绕行的能力。对通常的非 360° 障碍检测范围且具有物理尺寸的机器人来说,Bug 类算法就不能直接应用。折线 BUG 算法包括结合 DistBug 算法、VisBug 算法得到较短路径的“相遇点”与“脱离点”的确定方法,基于按分段直线与速度空间法结合得到圆滑自然路径的绕行障碍物边缘算法。还提出“虚拟触角”的概念来分析利用传感器数据,机器人依靠多种“虚拟触角”实现基于传感器的改进 Bug 算法。

[0120] 折线 BUG 算法提出虚拟触角概念,即利用测距传感器输入的实时环境信息数据判断是否有障碍物位于这些区域,如图 12a 所示,设计了三类五个虚拟触角感知障碍物信息。用纯直线行走与纯转向方式实现“对准目标前进”和“绕障碍物边缘”两种行走模式。纯转向行走分为三种:接近转向,避开转向,对准转向。算法最大特点是避障路径呈现分段折线形式,虽能达到避障效果,收敛性好。

[0121] 结合图 12b 避障算法程序框图对避障算法进行解释,机器人起始点为 S,目标点为 T。机器人当前位置为 X,H 为机器人与障碍物的相遇点,L 为机器人与障碍物的脱离点, L_{n-1} 为机器人与障碍物上一次的相遇点, L_{n-1} 为机器人与障碍物的上一次脱离点。

[0122] 机器人从起始点 S 开始运动,若未对准目标,作纯转动直到对准目标,进入“对准目标前进”模式开始直线前进。 $D(X, T)$ 表示当前位置 S 到目标点 T 的距离,F 为机器人在激光测距仪探测范围内可自由行走的距离, $D(X, T) < F$ 表示机器人与目标点的距离小于等于机器人可自由行走的距离,即前方无障碍物,机器人可对着目标点直线行走。如果 $D(X, T) > F$,则前方有障碍物,机器人进入绕行障碍物边缘模式。当内触角 D_{scs} 碰到障碍物,机器人停止直线行走,进入“绕障碍物边缘”模式开始避开转向,先判断障碍物的位置在机器人正前方的左还是右,若在左则往右转,若在右则左转。当外触角 D_{scb} 恰好检测不到障碍物时,机器人停止避开转向,开始直线行走,远离障碍物。直到行走距离达到设定的阈值,开始接近转向。转向方向是一个全局变量,在本次绕行开始点 H 时决定直到本次绕行结束点 L 止保持不变,确定方法是本次避开转向的相反方向。若在接近转向过程中,机器人满足脱离条件,即机器人已脱离障碍物,可对准目标点直线前进;若在接近转向过程中不满足脱离条件,但本次已转动角度达到设定的阈值即可停止接近转向,再次开始直线行走。若在直线行走过程中,内触角 D_{scs} 碰到障碍物,则机器人继续进入避开转向的循环;直线行走直到设定的阈值仍未碰到障碍物,则再次开始接近转向的循环。算法结束标志为机器人到达目标点 T。

[0123] 2) 圆弧 BUG 算法

[0124] 圆弧 BUG 在折线 BUG 的基础上,仍然沿用圆弧 BUG 的思维模式,分为直线行走模式和绕障碍物行走模式,在进行绕障碍物行走模式时加入触须概念,能够通过路径选择从较窄的狭缝中穿过,实现路径优化,避免不必要的绕行转弯,该算法根据四轮车辆进行设计,取消了原折线 BUG 算法原地转弯,采用平滑转弯方式,触角的设计改为宽度不一的矩形。

[0125] 触须又预测和行动指导作用,在进入绕障碍物行走模式时,预测方面,各触须代表当前车辆候选的规划路线,触须的设计符合车辆转弯特性的要求,保证车辆能跟踪此规划路线前进,在候选规划路线的各位置上根据车体的位姿模拟出车体及虚拟触角所在的区域,判断是否与障碍物区域重合;车体区域与障碍物区域重合表示按照该路线行走走到该位置,车辆将发生碰撞,虚拟触角区域与障碍物区域重合表示按照该路线行走走到该位置,车辆

正前方仍有障碍物,最后综合以上两种方式的计算预测候选规划路线的通行性。行动指导方面,是指根据各触须的预测情况,以一定原则处理预测反馈信息,综合计算出合理的转弯路线,包括转弯方向和转弯半径。计算机生成触须时,触须是以离散点表示,为简化触须的设计,本文假设车辆转弯时即进入等速圆周行驶,因此设计的触须是圆弧曲线。为预测车辆在各种转向盘角阶跃输入下的行驶情况,设计了多条触须,以车辆中心为起点延伸至车辆正前方为界的左右两个区域,各分布数条,且左右对称,如图 13a 所示。

[0126] 结合图 13b 避障算法流程图对避障算法进行解释,记出发点为 S,目标点为 T,当前位置值为 X,障碍物上最接近车体的点为 O,用 $d(X, T)$ 表示 X 至 T 的距离,用 $d(X, O)$ 表示 X 到 O 的纵向距离。用相遇点 H 表示朝目标点直线行走过程中碰到障碍物的点,用脱离点 L 表示绕行障碍物边缘的结束点, L' 表示上一次脱离点。g 是车体与障碍物距离相关的函数。与标准 BUG 算法相同,本文将车辆行为分为两种模式:“趋向目标前进”模式和“绕障碍物行走”模式。第一种模式即“趋向目标前进”,车辆对准 T 时直线行走。第二种模式即“绕障碍物行走”模式是结合转弯路线选择方法,采用分段绕行方式行走,具体有三种方式:避开转弯,接近转弯,直线行走。“绕障碍物行走”模式的核心理念是使车辆以交替“远离”和“接近”障碍物的方式绕行障碍物边缘。

[0127] 机器人从起始点 S 开始运动,若未对准目标,作纯转动直到对准目标,进入“对准目标前进”模式开始直线前进。 $D(X, T) < F$ 表示机器人与目标点的距离小于等于机器人可自由行走的距离,即前方无障碍物,机器人可对着目标点直线行走。如果 $D(X, T) > F$,则前方有障碍物,机器人进入绕行障碍物边缘模式。

[0128] 当内触角 $Drecs$ 碰到障碍物,机器人停止直线行走,进入“绕障碍物边缘”模式开始避开转弯,转弯路径选择最小转弯半径的触须,先判断障碍物的位置在机器人正前方的左还是右,若在左则往右转弯,若在右则左转弯。当外触角 $Drecb$ 恰好检测不到障碍物时,机器人停止避开转弯,开始直线行走,远离障碍物;或者当避开转弯方向不能继续通行时停止。

[0129] 机器人直线行走直到距离达到 $g(m)$ 或者直线行走过程中满足脱离条件,如果为第二种情况,对准备目标的直线行走到达 T,如果第一种情况开始接近转弯,仍然选择转弯半径最小的触角,转弯方向与避开转弯方向相反。接近转弯直到 $Drecs$ 触角遇到障碍物或者对准 T 且该点满足脱离条件,如果为前一种情况,继续重复避开转弯与接近转弯的过程,如果为第二种情况,直接对准 T 到达目标。直到到达脱离点 L。到达脱离点之后直线行走,算法结束标志为机器人到达目标点 T。

[0130] 以上所述之实施例子只为本发明之较佳实施例,并非以此限制本发明的实施范围,故凡依本发明之形状、原理所作的变化,均应涵盖在本发明的保护范围内。

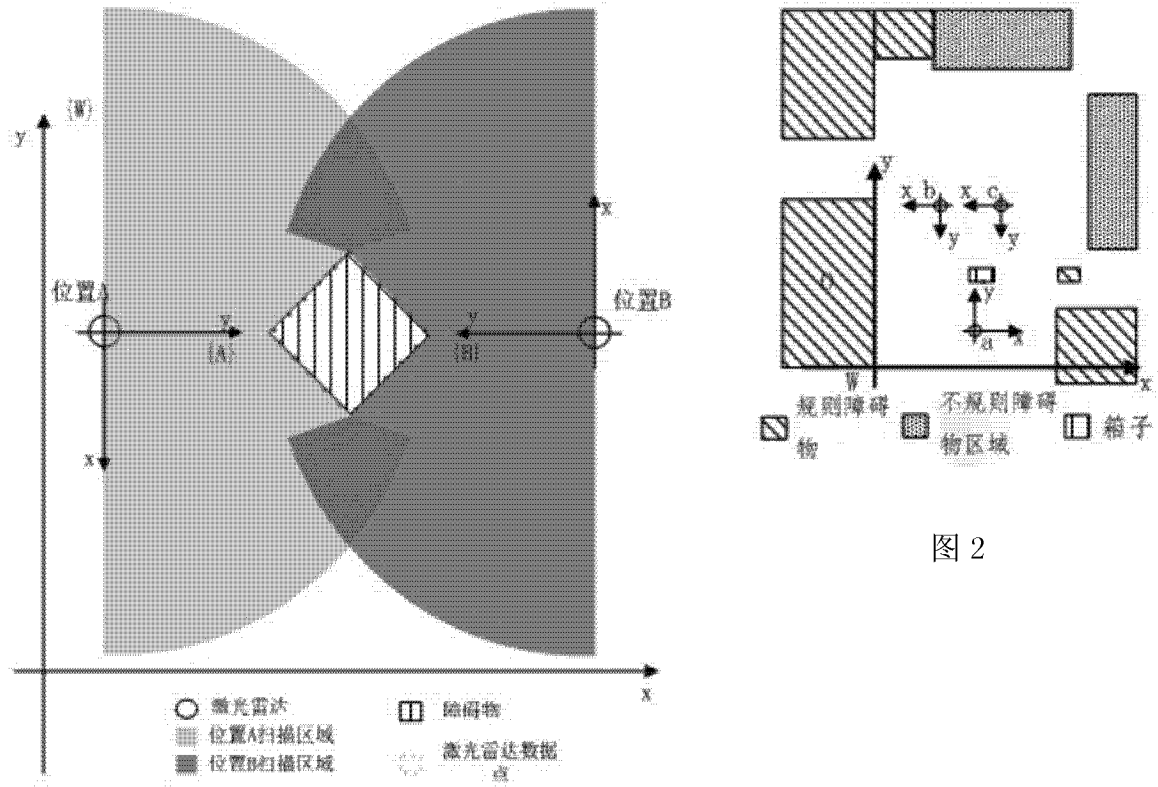


图 1

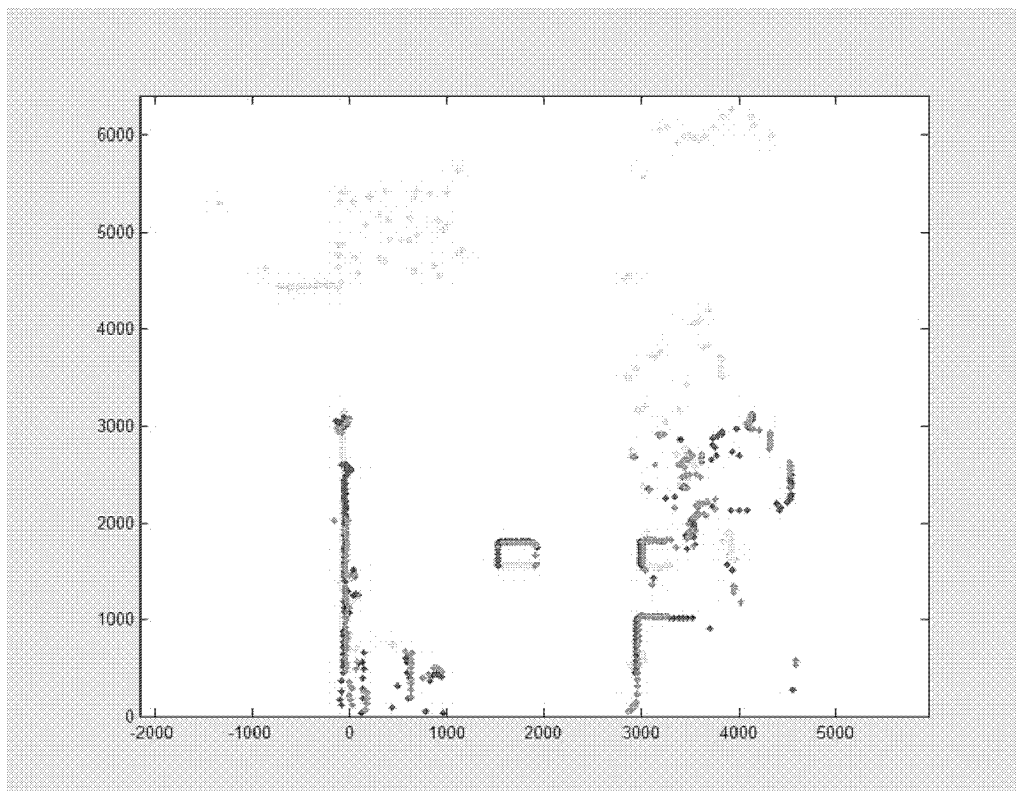


图 3

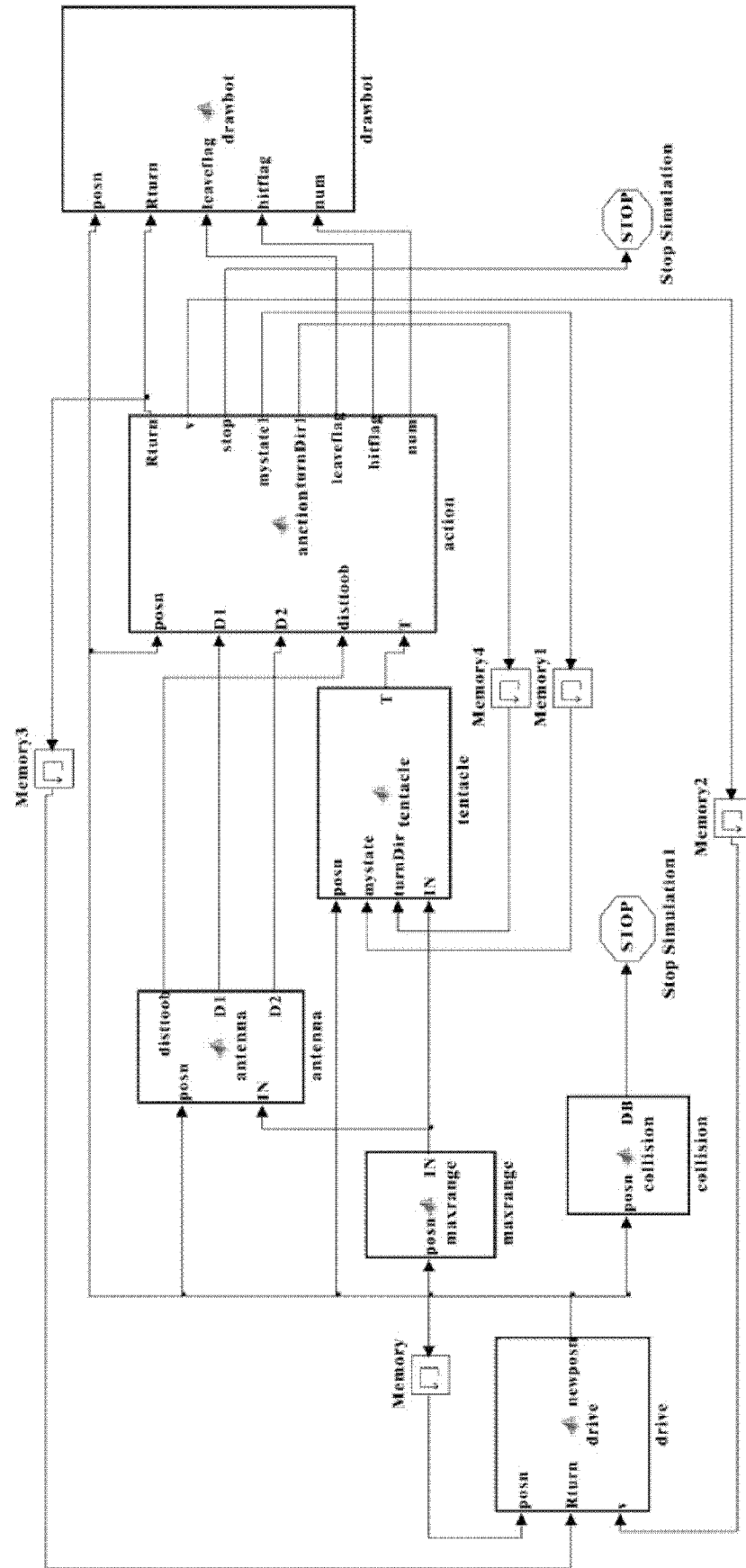


图 4

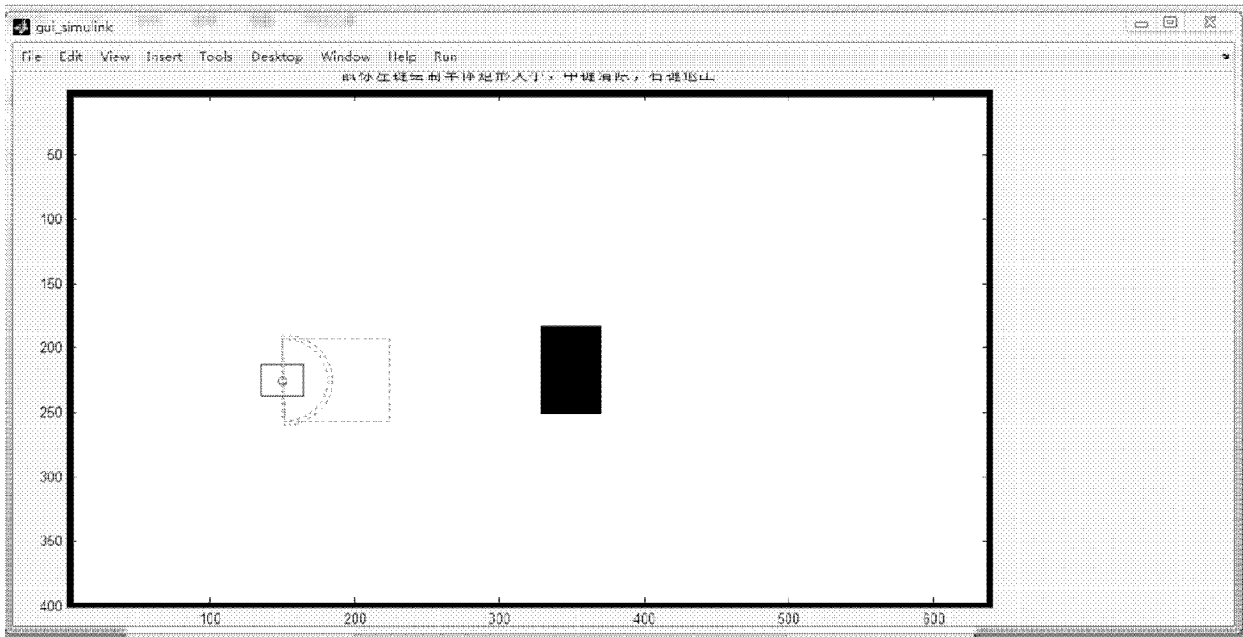


图 5a

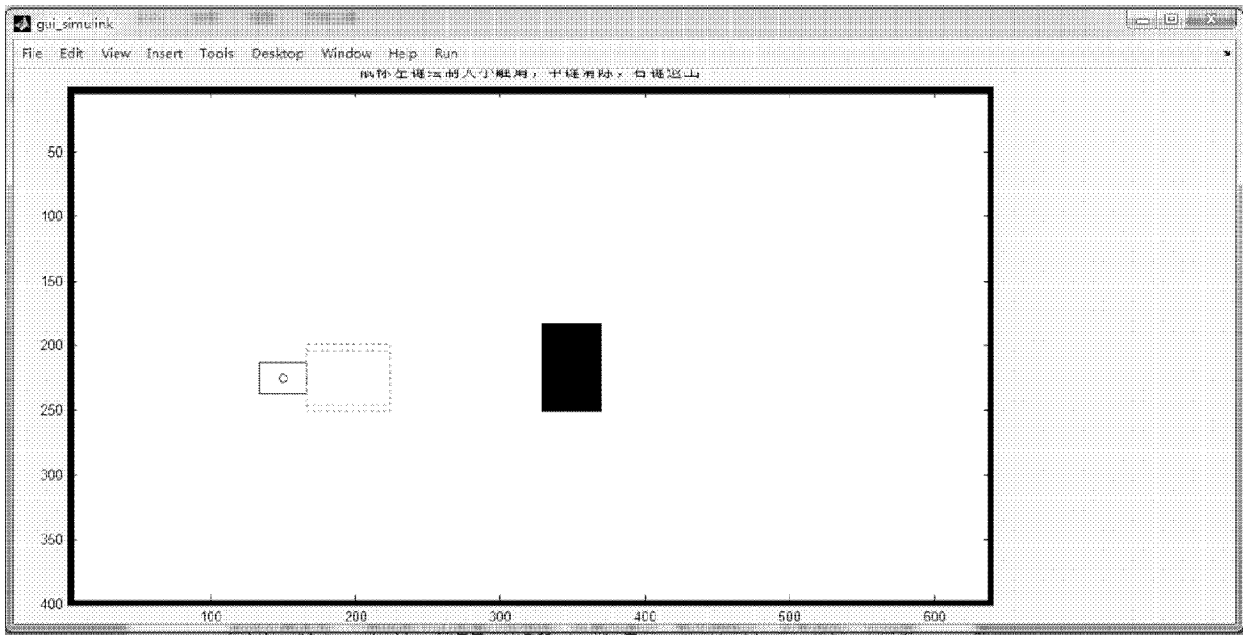


图 5b

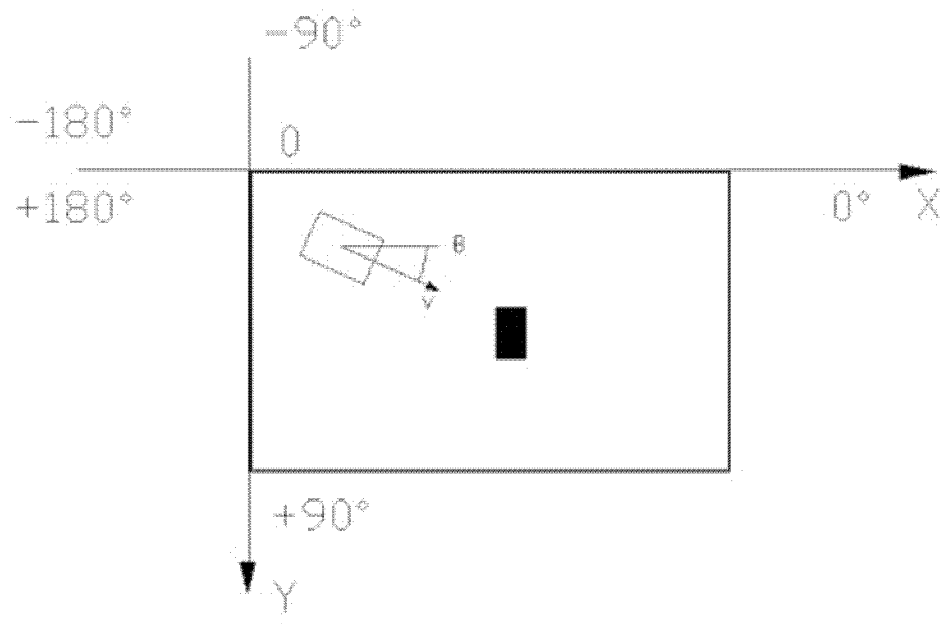


图 6

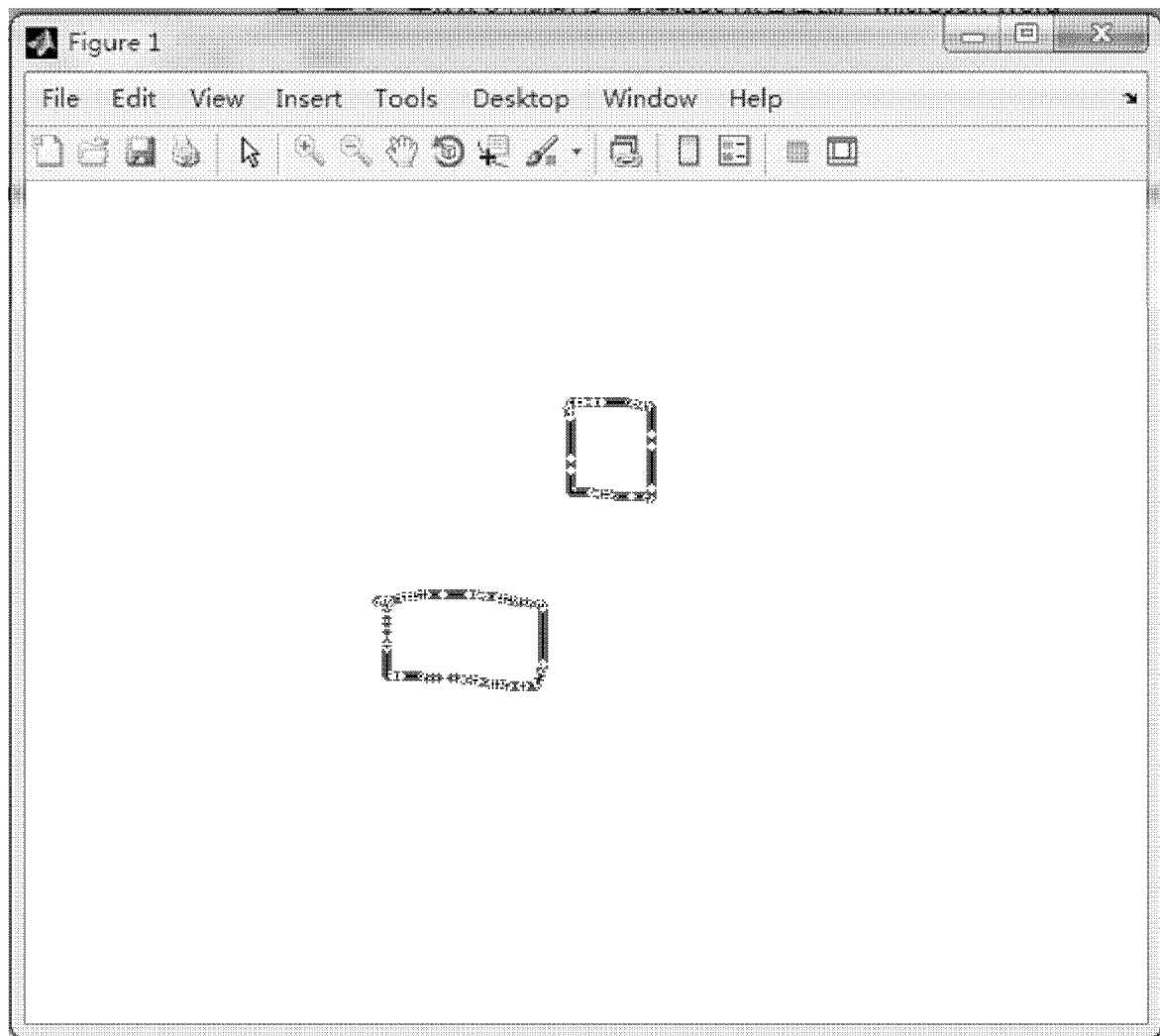


图 7

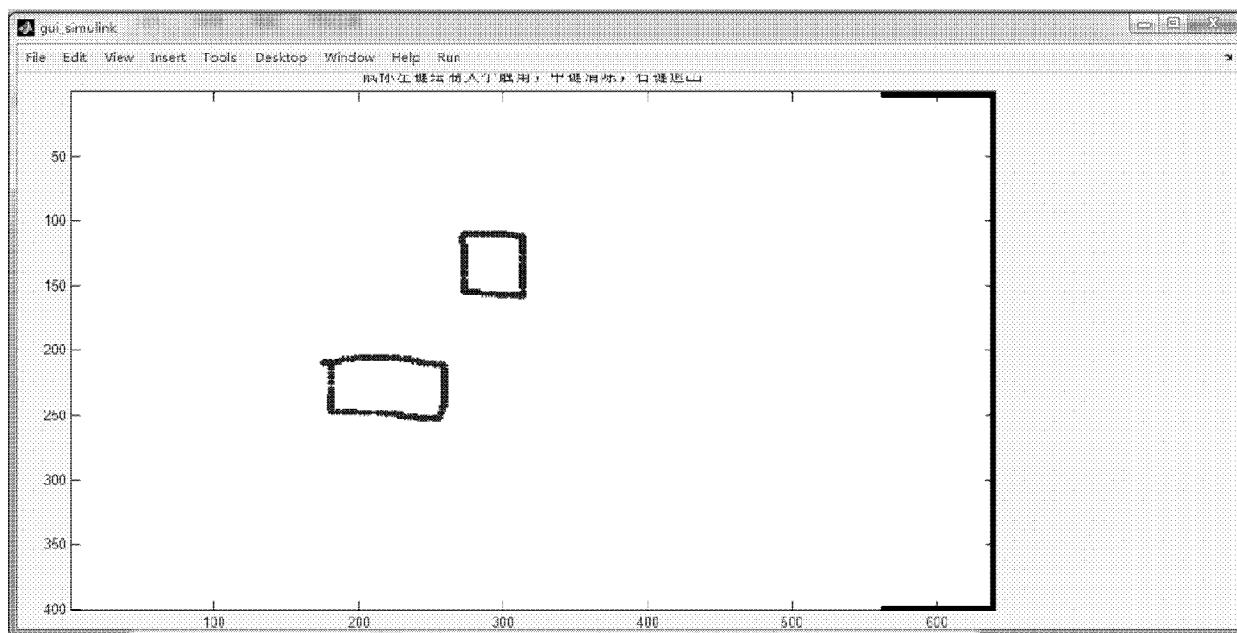


图 8



图 9

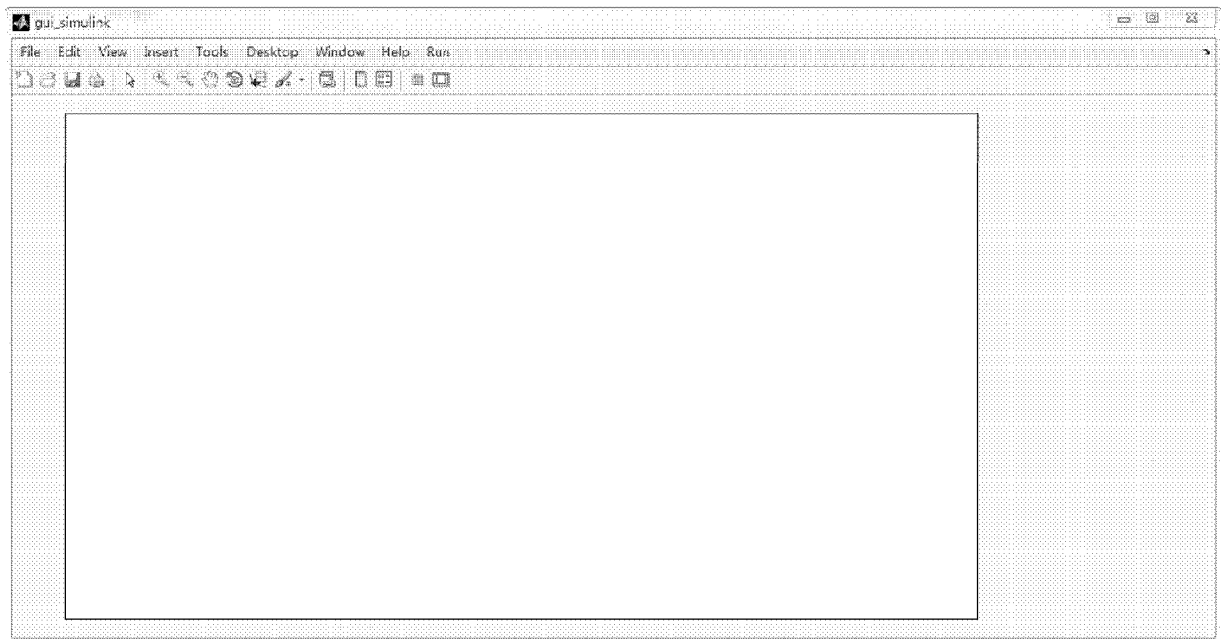


图 10

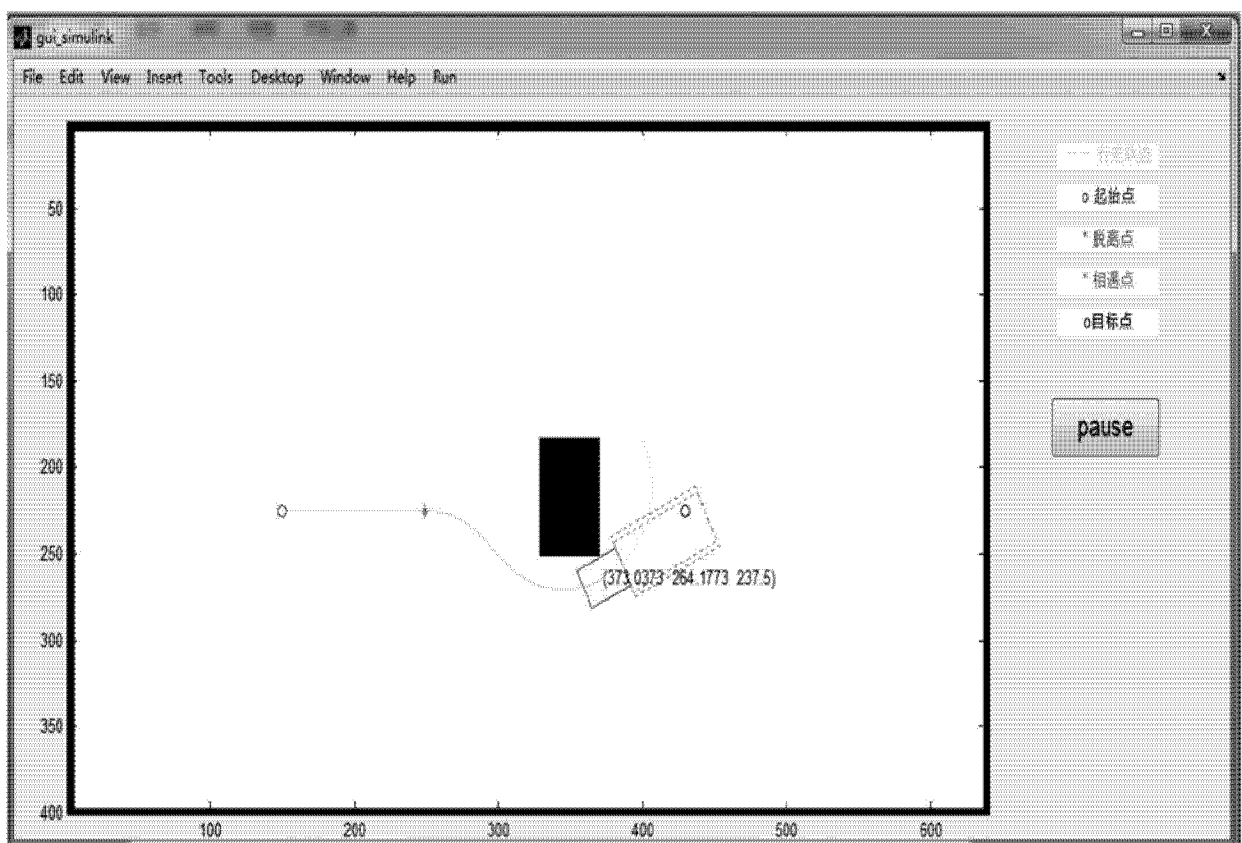


图 11

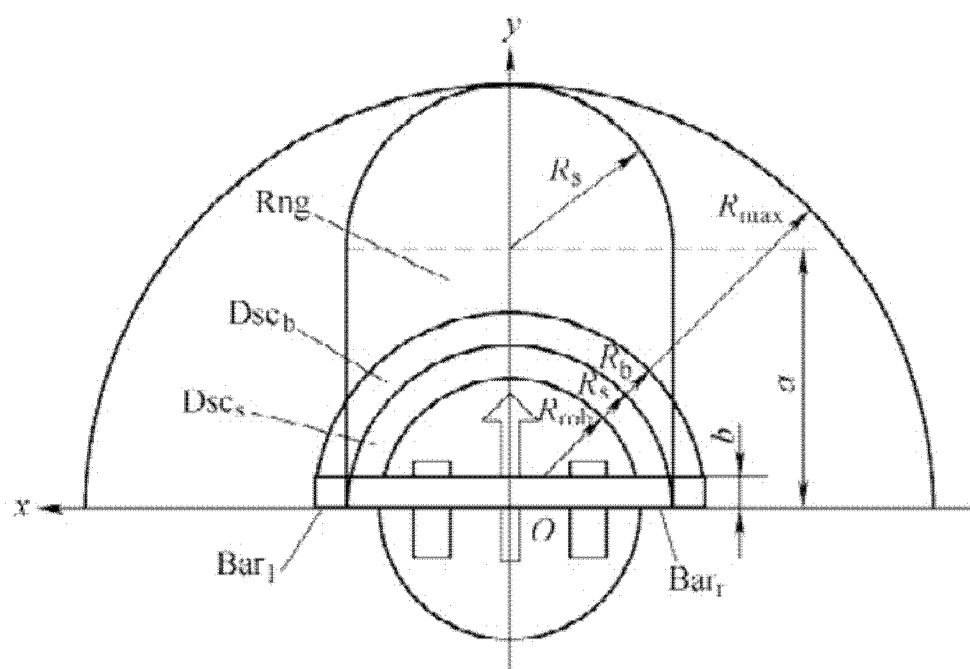


图 12a

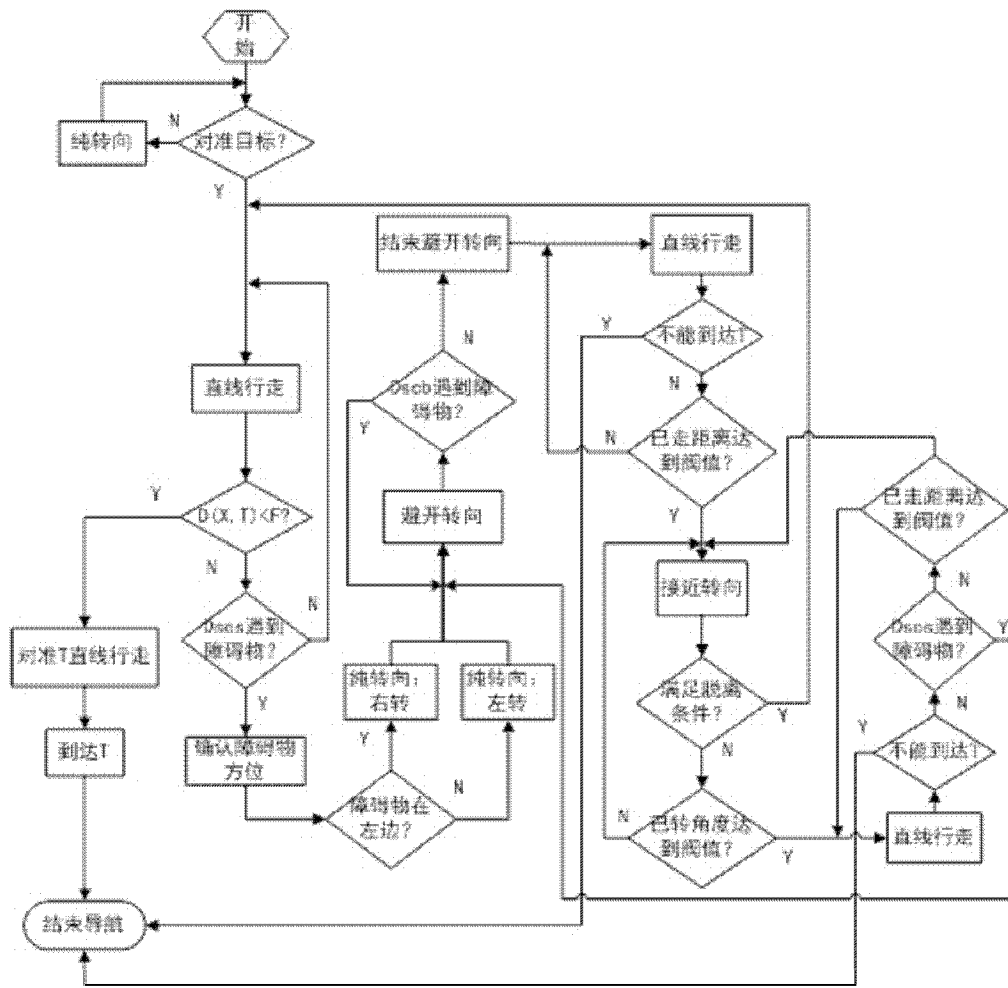


图 12b

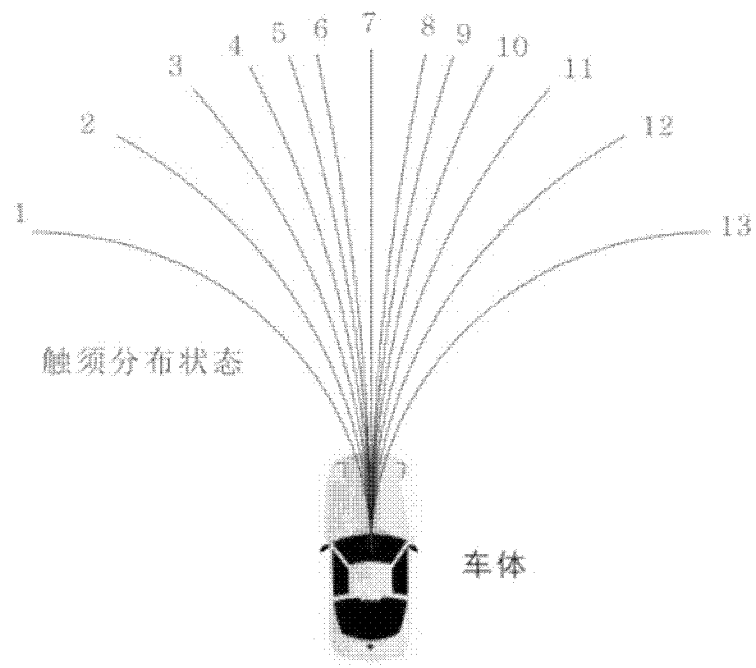


图 13a

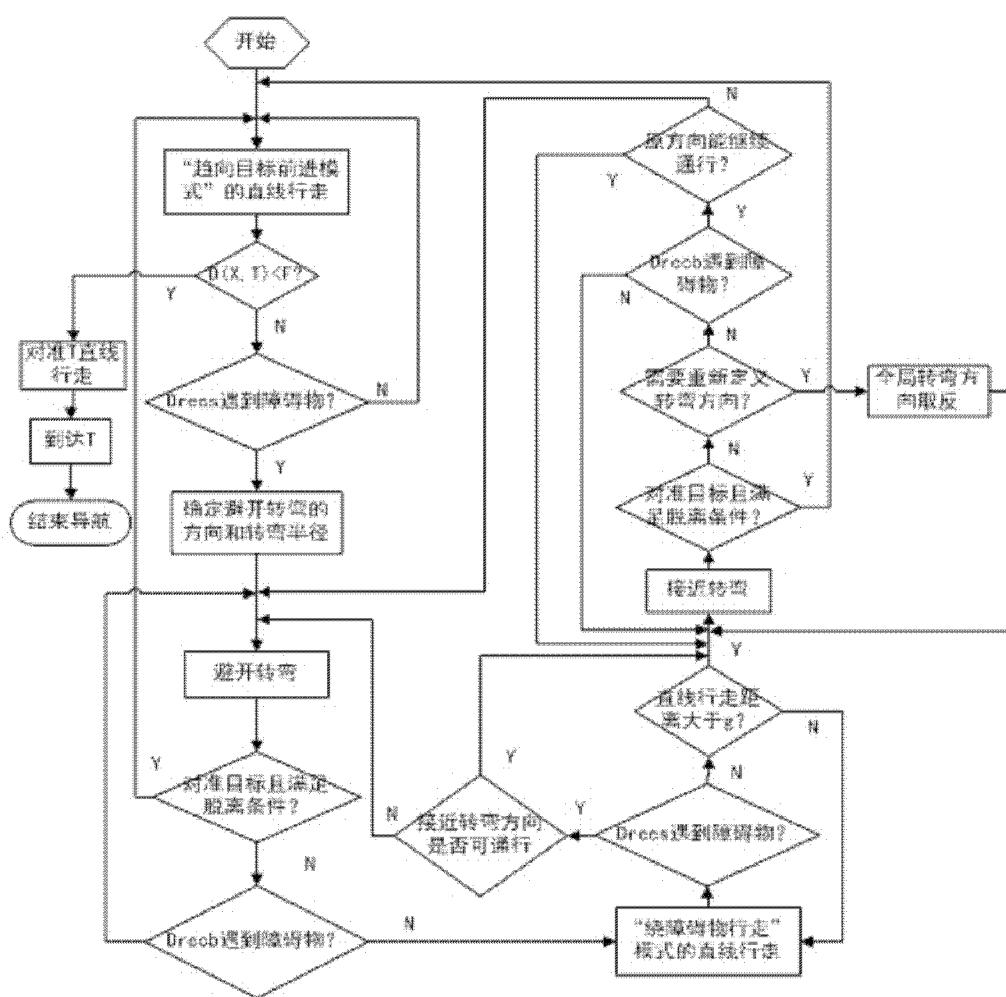


图 13b