

Research on Obstacle Avoidance System and Path Planning of Unmanned Ground Vehicle Based on PX4

Qiongge Jiang, Lifeng Wang

Institute of Electrical and Control Engineering, North China University of Technology, Beijing

Email: 1686308254@qq.com, lifeng_wang@ncut.edu.cn

Received: Apr. 1st, 2019; accepted: Apr. 12th, 2019; published: Apr. 25th, 2019

Abstract

The obstacle avoidance and path planning of the unmanned ground vehicle mean that the unmanned vehicle can automatically avoid obstacles during the automatic cruising process, re-plan the current travel route and re-enter the preset route. This article uses HPI's Savage Flux 2350 differential drive off-road vehicle as an experimental platform and open source hardware flight control Pixhawk as a motion controller, to conduct secondary developments of PX4 Rover software system. The data acquisition module is used to obtain the ultrasonic data and communicates with the flight control board through the serial port mode to complete the data exchange, thereby realizing the real-time detection of the surrounding environment of the unmanned ground vehicle and completing the obstacle avoidance function in the cruise mode.

Keywords

Unmanned Ground Vehicle, Obstacle Avoidance, PX4

基于PX4的地面无人车 避障系统及路径 规划研究

姜琼阁, 王立峰

北方工业大学电气与控制工程学院, 北京

Email: 1686308254@qq.com, lifeng_wang@ncut.edu.cn

收稿日期: 2019年4月1日; 录用日期: 2019年4月12日; 发布日期: 2019年4月25日

摘要

地面无人车避障及路径规划是指, 无人车在自动巡航过程中, 遇到障碍物能够自动避开, 并对当前行进路线进行重新规划, 重新进入预设路线行驶。本文使用HPI公司的Savage Flux 2350差速驱动越野车作为实验平台, 选用开源硬件Pixhawk飞控作为运动控制器, 对PX4 Rover软件系统进行二次开发。使用数据采集模块获得超声波数据, 并通过串口方式与飞控板进行通信, 完成数据交换, 从而实现对地面无人车周围环境的实时检测, 完成巡航模式下的避障功能。

关键词

地面无人车, 避障, PX4

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

地面无人车辆(Unmanned ground vehicles, UGV)集环境感知、规划决策与多等级辅助驾驶功能于一体, 是涉及传感器、信息融合、计算机、通讯、人工智能及自动控制技术等多学科、多领域的高新技术集合体[1]。地面无人车又常被称为无人车、智能车、智能机器人等。近些年来, 它凭借自身优势, 体积小、机动灵活、易操作等特点被广泛应用于各个行业[2]。

地面无人车的研究在国外开始较早。随着近些年计算机视觉、人工智能、电子电路等学科快速发展, 这些学科技术也被广泛应用于无人车研究中[3]。近年来, 关于无人车、无人船的研究和使用开始火热起来, 涉及航空、航海、农业等各个领域, 国内外许多研究机构也开始关注地面无人车自动控制领域[4][5]。目前研究人员重点关注避障技术的研究, 主要包括障碍物识别、信息位置的获取、车速和转向控制等方面[6]。地面无人车自主避障系统可以利用传感器来获取周围环境信息, 具备感知周围环境和自适应的能力, 在完成给定任务的同时, 能够在过程中实现自动避障及路径规划[7]。

本文以开源飞控 Pixhawk 作为运动控制器, 通过对无人车进行建模, 结合串级 PID 控制算法, 实现无人车的自主巡航功能。针对数据采集过程中出现错误数据的问题, 提出一种新的数据处理算法。并对障碍物检测区域进行划分, 使无人车在不同障碍物方位下采取不同的避障策略。

2. 地面无人车系统架构

2.1. 无人车基本组成

地面无人车的基本组成可以分为三部分, 分别是机械系统、动力系统和控制系统[8][9]。本文使用 HPI 公司的 Savage Flux 2350 四驱差速越野车作为地面无人车的实验平台。动力系统中主要是由舵机、电调、电机和电池四部分构成, 提供无人车的动力来源。控制系统主要由飞控板、一系列配套外设和地面站组成。本系统中使用的是 Pixhawk 控制器, 外部设备包括数传模块、GPS、电源模块、蜂鸣器、安全开关、遥控器接收机等。搭建完成的无人车基本结构如图 1 所示。

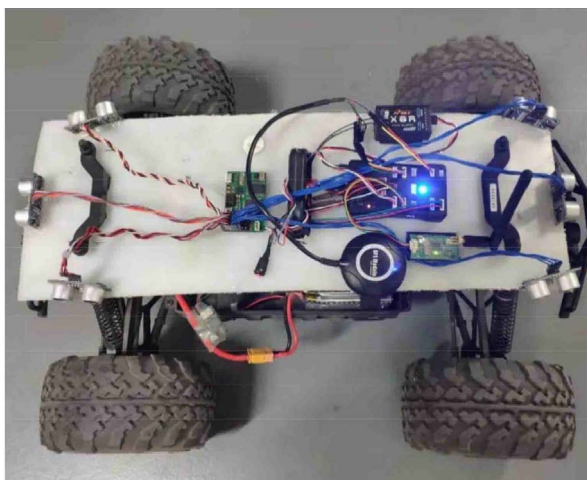


Figure 1. Unmanned vehicle basic structure

图 1. 无人车基本结构

2.2. 数据采集模块

2.2.1. 数据采集模块硬件连接方式

本文通过在车头和车尾各安装三个 HC-SR04 超声波, 来实现对无人车周围障碍物的实时监测。由于 Pixhawk 硬件接口限制, 不能支持外接多个超声波模块, 因此这里使用数据采集模块对超声波数据进行预处理, 将处理后的数据通过串口发送给 Pixhawk 控制器, 数据采集模块与 Pixhawk 连接方式如图 2 所示。

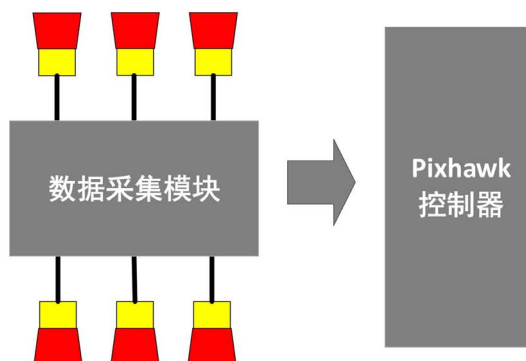


Figure 2. Data acquisition module connection diagram

图 2. 数据采集模块连接图

2.2.2. 数据采集模块软件设计

在对数据采集程序进行设计的时候, 需要考虑多个超声波传感器共同工作时带来的信号交叉问题。交叉问题是指使用多个超声波传感器时, 由传感器 a 发出的声波, 在经过反射后可能被传感器 b 和 c 获得, 这时候 b 和 c 就会根据这个声波信息来计算距离, 造成测量数据的混乱。针对这个问题, 我们对超声波传感器进行编码, 车头左侧的超声波定义为 U1, 沿顺时针方向分别定义为 U1、U2、U3、U4、U5、U6, 在测距程序中设置每个超声波触发信号的时间间隔为 40 ms。在保证超声波数据实时性的同时有效地解决了超声波数据交叉影响的问题。

对超声波数据处理过程中, 最初使用常用的均值滤波算法, 采集 5 次超声波数据取平均值作为最终

输出,但是后来发现在实际的应用中传感器的数值并不是很稳定,每10分钟可能会出现一到两次的数据错误,在多个超声波上都出现了类似的错误,经过观察总结发现错误数据有两个特点:

- 1) 错误数据一般连续出现,个数通常不超过三个。
- 2) 错误数据离真实值偏离较大,并且连续的错误数据都比较相近。

无人车在运行过程中,需要获得稳定的超声波数据。根据错误数据的特点,本文对传统的均值滤波算法进行改进,引入新的数据处理算法对超声波数据进行处理。算法的主要思想是,利用少数遵循多数的原则,把采集到的一系列数据分成两部分,将相邻两个数据值之差最大的数作为数据的分界点。然后取两部分数据中个数最多的部分作为正确的测量值,另一部分为无效测量值。算法流程如下图3所示,经过处理后的数据相比之前更加稳定,错误数据数量明显减少。

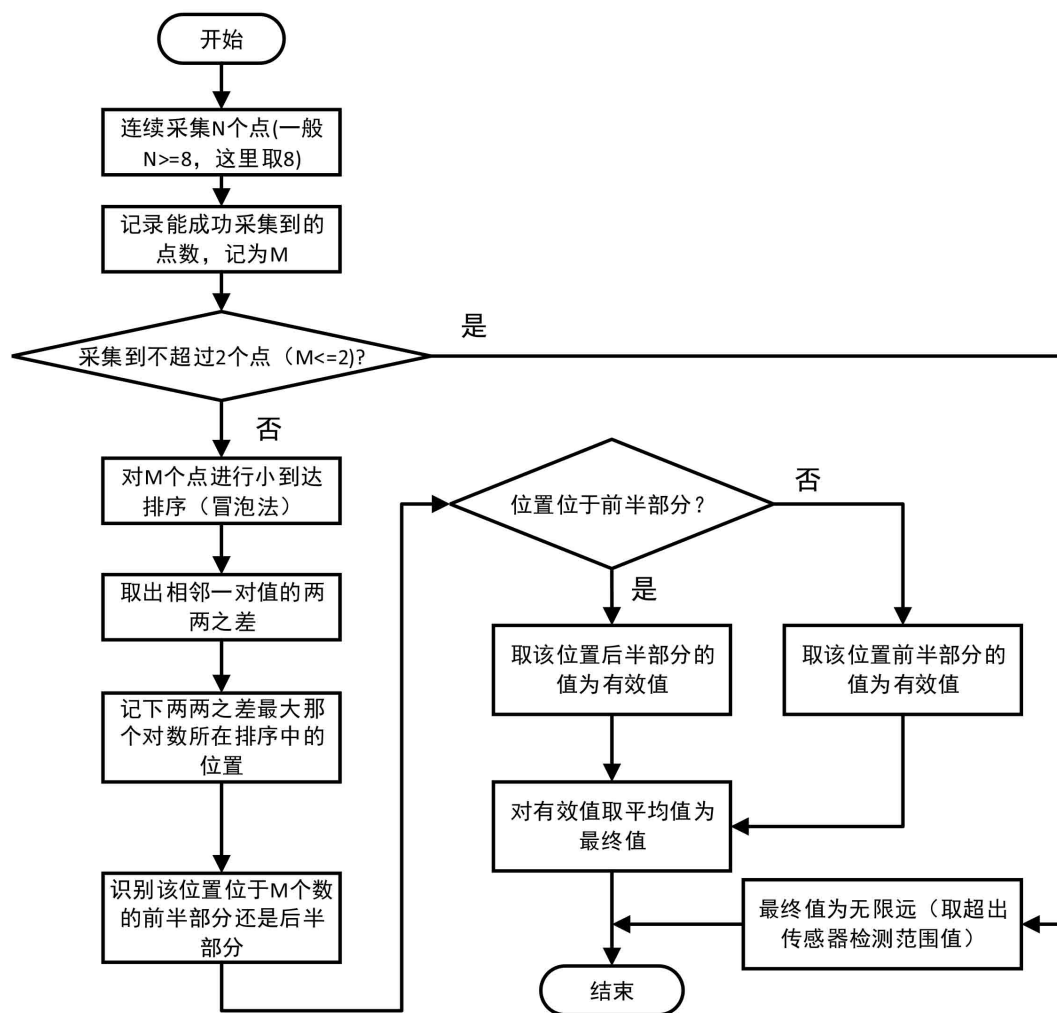


Figure 3. Algorithm flowchart

图3. 算法流程图

最后经过处理的超声波数据,通过数据采集模块的串口发送到 Pixhawk 的 TELEM2 接口。首先对 ATmega2560 串口 1 进行初始化,设置串口通信波特率为 9600;对每个超声波设置启动时间,获得超声波检测数据,并将数据进行格式化,精确到个位数;在对超声波数据进行处理后,将得到的新的数据列通过串口 1 进行发送。

3. 无人车避障及路径规划设计

3.1. 无人车建模

车辆运行路径的动态调整是实现自动巡航功能的核心, 在获得车辆精准的位置信息后, 如何使车辆能够精确地沿着预设的路线进行巡航, 是下一步需要研究的问题。要实现无人车的控制分析, 首先需要对控制对象进行建模, 下面对本文中使用的无人车进行建模分析。

1) 无人车建模

这里我们参考自行车模型[10]对无人车的运动进行简化, 主要有以下两个假设:

- ① 忽略车辆在垂直(z 轴)方向的运动, 可以理解为在高处以俯视的角度对车辆进行观察。
- ② 假设车辆的前面两个轮胎具有相同的角度和转速, 同样后面两个轮胎也是如此, 这样就可以把车辆的结构看作自行车, 前后的轮胎就可以各用一个轮胎来描述。与自行车原理相同, 假设车辆的转向完全由前轮控制。这样无人车模型就可以简化为如图 4 所示的自行车模型。

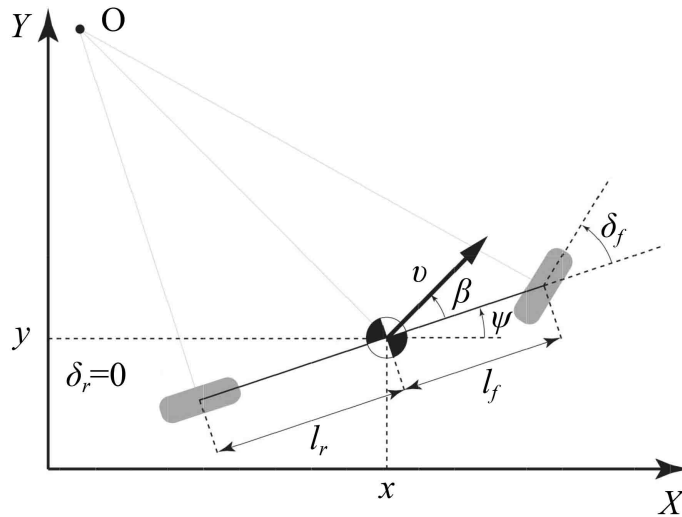


Figure 4. Bicycle model
图 4. 自行车模型示意图

在该模型中, 因为我们使用的车辆后轮不能旋转, 这里设置后轮的转向角控制输入 $\delta_r = 0$, 也就是说方向盘的控制输入都反映到前轮的转角上, 因此整个系统的控制量可以简化为 (a, δ_f) , 其中 a 代表车辆的加速度。加速度的正负分别代表无人车处于加速或减速阶段, δ_f 代表当前轮胎的转角, 这样就使用两个变量描述车辆的控制输入。使用四个状态变量描述车辆当前的状态, (x, y) 表示车辆当前位置; ψ 表示当前车辆的偏航角, 逆时针方向为正; v 表示当前的车速; β 表示车辆当前的质心侧偏角, 即小车质心的速度相对于无人车纵轴的角度。使用 l_r 和 l_f 表示前轮和后轮到车辆重心的距离, 根据动力学原理, 无人车的各个状态变量的更新公式如下所示:

$$x_{t+1} = x_t + v_t \cos(\psi_t + \beta) * dt \quad (1)$$

$$y_{t+1} = y_t + v_t \sin(\psi_t + \beta) * dt \quad (2)$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{l_r} \sin(\beta) * dt \quad (3)$$

$$v_{t+1} = v_t + a * dt \quad (4)$$

$$\beta = \tan^{-1} \left(\frac{l_r}{l_f + l_r} \tan(\delta_f) \right) \quad (5)$$

基于该运动学自行车模型, 在给定一个时刻的控制输入后, 可以通过计算求得在 dt 时间以后车辆的状态信息 (x, y, ψ, v) , 这样就可以使用该模型结合串级 PID 控制算法实现对控制量 (a, δ_f) 的动态调整, 从而实现对车辆状态的实时调整, 进而实现自动巡航功能。

3.2. 避障及路径规划设计

因为无人车不具备垂直方向的运动能力, 这里我们仅考虑在二维空间中的避障策略。由前文超声波的布局方式可知, 超声波的探测区域可以被简化为一个扇形区域。超声波的最大测量距离为 4 m, 但是实验中发现 3 m 到 4 m 范围内的检测数据容易出现波动, 因此这里将超声波的探测区域简化为半径 $r = 3$ m、角度 $\theta = 45^\circ$ 的扇形区域, 如下图 5 所示。

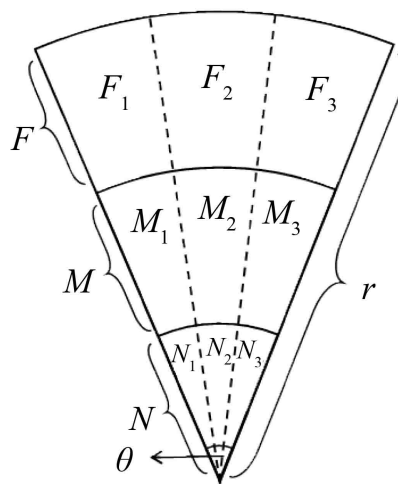


Figure 5. Detection area division
图 5. 检测区域划分示意

下面首先对检测区域进行划分, 根据检测到的距离将检测区域划分为 F、M、N 为三个扇形区域, 分别表示距离无人车 2 米到 3 米之间的区域、1 米到 2 米之间的区域、0 到 1 米之间的区域。根据超声波传感器的检测角度, 可以将检测区域划分为九个检测区域, 分别表示为 F1、F2、F3、M1、M2、M3 和 N1、N2、N3 九个检测区域, 角标代表传感器编号, 1、2、3 号传感器代表在车身上安装的 U1、U2、U3 超声波传感器, 分别用来检测无人车的左侧、前方、右侧区域。这里将 F、M、N 分别定义为减速区、避障区和停止区, 在不同的距离范围内无人车采取不同的避障操作。无人车的避障实现流程如图 6 所示。当检测到障碍物的时候, 首先对障碍物所处区域进行判断, 然后判断障碍物的方位, 根据障碍物的方位执行相应的避障操作指令。

下面对障碍物位于 M 区域的情况进行简要分析, 当障碍物由 1 号超声波检测到时, 表明障碍物在 M1 区域位于无人车的左侧, 此使应让无人车执行小角度右转操作; 当障碍物由 2 号超声波检测到的时候, 此时障碍物位于 M2 区域, 障碍物为无人车的正前方, 考虑到操场的实际情况, 这里执行大角度左转操作; 同理, 当障碍物被 3 号超声波检测到时, 障碍物位于无人车右侧, 此时执行小角度左转。

真实的场景下可能会遇到体积比较大的障碍物, 同时会被多个超声波检测到。这里针对该种情况进行讨论, 当障碍物由 1 号和 2 号超声波同时检测到, 此时障碍物跨越了 M1 和 M2 区域位于无人车的左

前方, 这时候仅执行小角度的右转已经不能满足避障的要求, 需要使无人车执行大角度右转来躲避障碍物。同样, 当障碍物被 2 号和 3 号传感器同时检测到时, 执行大角度左转来躲避障碍物。更极端的一种情况, 障碍物同时被三个传感器检测到, 这时仅执行转向操作已经不能实现较好的避障, 此时无人车执行和在刹车区域时相同的操作, 首先停车, 然后执行倒车指令, 进入避障区域重新执行避障操作。

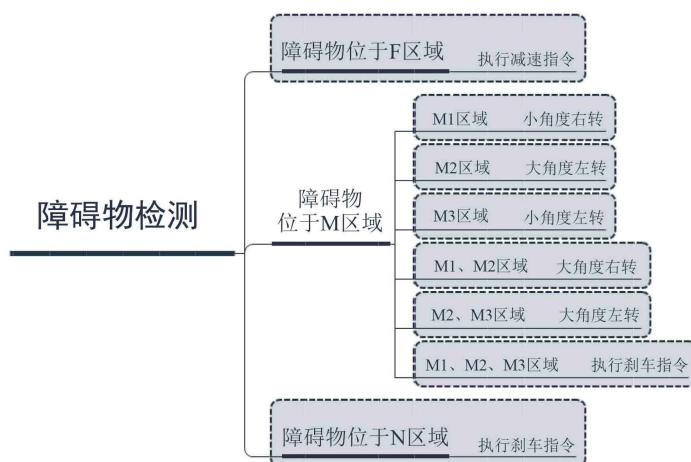


Figure 6. Obstacle avoidance process
图 6. 避障实现流程

在避障策略的研究中, 主要通过减速及转向来实现对障碍物的躲避。这就会产生另外一个问题, 即躲避障碍物后, 如何快速回到最初设定的航线上继续执行巡航任务。

这里使用串级 PID 算法来对无人车的运动状态进行动态调整。在串级 PID 中将角度控制 PID 算法和角速度控制 PID 算法进行串联, 经过串联后的 PID 控制器可以控制多个变量, 系统的抗干扰性得到增强, 无人车的适应能力得到进一步的提升。在执行避障操作后, 无人车的行进路线已经偏离了最初的航线, 根据 PID 原理, 此时的误差值会突然变大, 同时 PID 针对控制量 (a, δ_f) 不断进行优化, 优化过程中无人车逐渐回到最初航线上, 无人车中使用的串级 PID 的原理框图如图 7 所示。

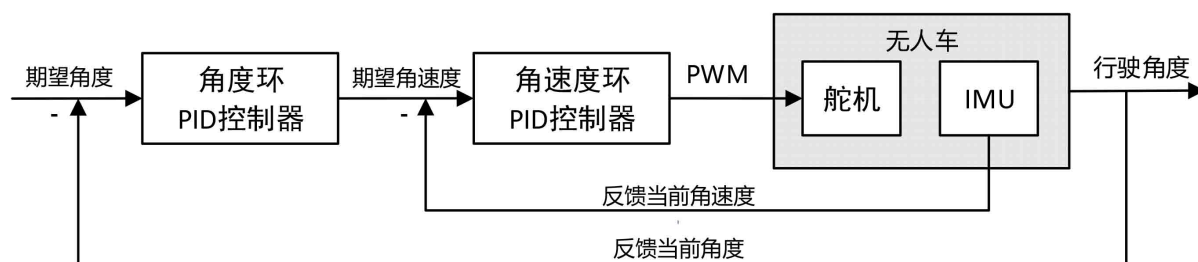


Figure 7. Cascade PID block diagram
图 7. 串级 PID 原理框图

3.3. 自定义主题添加

在进行避障及路径规划实验前, 需要对数据采集模块发送过来的超声波数据进行处理。由于 PX4 采用 uORB (微对象请求代理器)通信机制, 因此通过串口获得的数据并不能直接利用, 需要以发布主题、订阅主题的形式进行消息的传递。uORB 是飞控程序中进行消息传递的异步消息机制。它在 Pixhawk 系统占有重要地位, 负责整个系统的数据传输任务, 飞控中所有的传感器数据、GPS、PPM 信号等数据都

要通过 uORB 传输到系统中各个模块进行计算处理。

Pixhawk 的核心为 NuttX 实时 ARM 操作系统, uORB 允许多个进程打开同一个文件, 进程之间通过该文件节点进行数据的交互与共享。每个进程都可以订阅或者发布主题, 系统中可以存在多个发布者, 一个进程也可以同时订阅多个主题, 但是在一条总线上始终只能有一条消息。飞控中所有的数据都通过这种主题的发布和订阅方式来完成数据交换, 如图 8 所示。下面我们需要通过自定义主题的方式, 将数据采集模块发送过来的超声波信息进行发布, 供其他功能模块订阅来实现信息的交换。

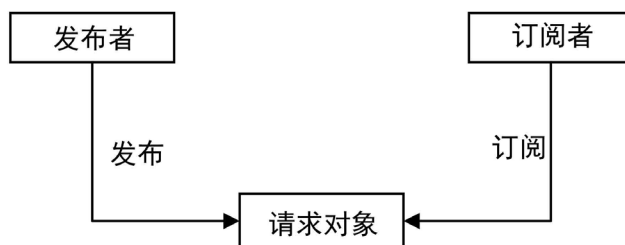


Figure 8. Data exchange process
图 8. 数据交换流程

1) 首先需要在 Firmware/msg 目录下新建一个名为 read_uart_sensor.msg 文件, 该文件定义 Pixhawk 串口接收到所有超声波数据的结构, 然后在 msg/CMakeLists.txt 下添加该文件名。添加完成后重新编译, 会在 Firmware/src/modules/uORB/topics 目录下生成 read_uart_sensor.h 头文件。

2) 在 Firmware/src/modules 目录下创建文件夹并命名为 read_uart_sensor 作为超声波主题模块的存放位置。在该文件夹下创建 read_uart_sensor.c 文件和 CMakeLists.txt 文件。

在 read_uart_sensor.c 文件中对发布主题进行定义, 主要包括定义主题、公告主题和发布主题三个过程。程序中, orb_advertise()函数实现公告操作, 在使用 orb_publish()发布新数据前需要先执行公告操作, 公告成功后函数返回的处理值提供给 orb_publish()函数, 在主题发布时进行调用。在对 read_uart_sensor.c 文件进行定义后, 需要对 CMakeLists.txt 文件进行配置, 主要是对编译中的参数进行定义。

3) 完成前面的操作后需要在 Firmware/cmake/configs/nuttx/nuttx_px4fmu-v2_default.cmake 文件中注册自定义的 read_uart_sensor 模块。

4) 将超声波主题添加到启动脚本中。要实现上电后就可以在 Pixhawk 控制器中获得超声波数据, 需要将该模块添加到系统的启动脚本文件中。添加指令 read_uart_sensor start/dev/ttyS2。这样就可以实现当数据采集模块与 TELEM2 端口(/dev/ttyS2)相连的时候, 自动启动超声波模块。

至此已经完成了超声波自定义主题的发布过程, 整体的发布流程如图 9 所示。

在完成主题的发布后, 下一步需要订阅该主题, 获得超声波数据数据。orb_subscribe()函数用来订阅数据, 获得的数据保存在 int sensor_sub_fd 结构体中; orb_set_interval()函数用来设置获取数据的时间间隔; orb_check()函数用来检查上次读取主题后, 主题是否更新过。最后对修改后的源码重新编译, 并上传编译后的固件。经过以上操作, 在 nsh 命令行中就可以通过输入指令, 查看当前数据的订阅结果。

4. 地面无人车避障及路径规划实验

4.1. 自动巡航实验

避障及路径规划是在自动巡航的基础上进行的, 在实验中首先对无人车的自动巡航功能进行验证。在该模式下, 无人车根据地面站上设定好的航线, 自动巡航, GPS 信号与惯性测量单元的精度, 直接决定了巡航路线的精确度。

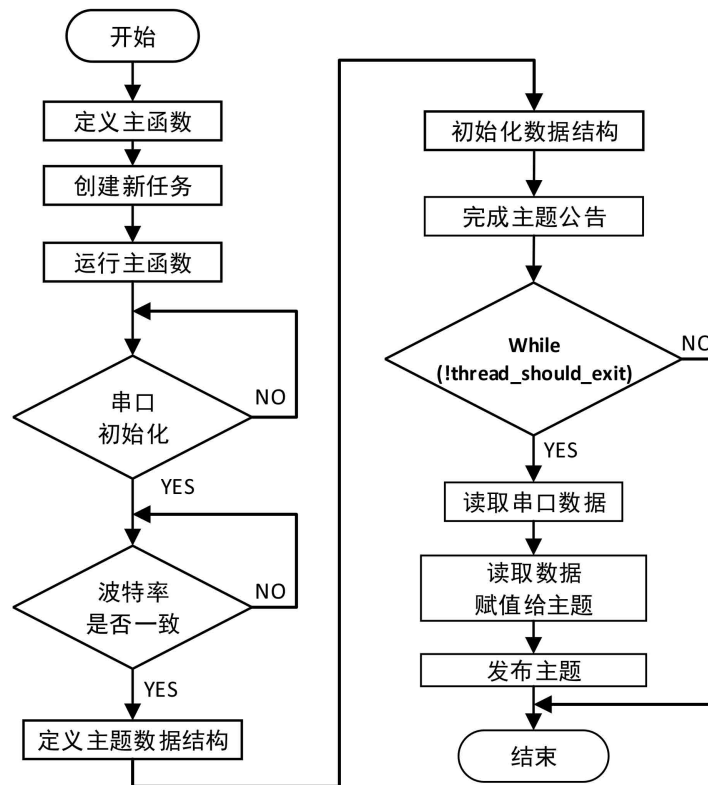


Figure 9. Custom theme publishing process

图 9. 自定义主题发布流程

实验中, 首先确保手动模式下的操作正常, 打开地面站地图界面, 如图 10 所示。在界面的右侧可以实现, 读取航点、写入航点, 起始位置等操作, 主界面下侧设置各航点的具体参数(速度、停留时间、预定航点拍照等), 也可以删除航点。除了设置路线之外, 还可以设置 H 点, H 代表家的位置, 设置该位置可以实现无人车在巡航结束后能够一键返航, 自动返回 H 点。我们在操场中规划简单的路径对自动巡航功能进行验证, 设置完航线之后点击右侧写入航点即可将任务下载到飞行控制器中。



Figure 10. Ground station flight plan operation interface

图 10. 地面站飞行计划操作界面

下载完成后进入飞行数据界面, 检查 GPS 信号强度, GPS 卫星数量在十颗以上可以进行自动巡航实验。使用遥控器将无人车从手动模式切换到自动模式, 无人车将沿着预先设定的航线运行, 使用数传模块可以将无人车的运动状态在地面站上实时显示, 主要包括无人车的实时轨迹、位置及速度等信息。图

11 为地面站中实时显示的无人车运动轨迹, 其中黄色曲线为初始设定航线, 紫色曲线为无人车实际运行路线。从图中可以看到实际航线和预定轨迹基本吻合, 因此无人车的自动巡航功能满足要求。

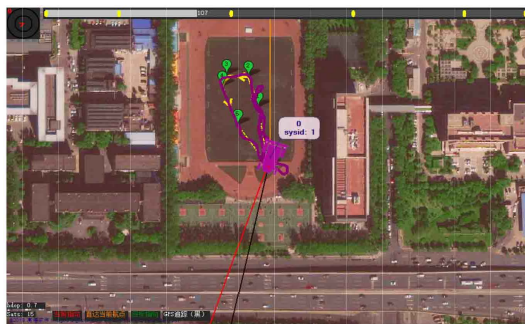


Figure 11. Automatic cruise route map

图 11. 自动巡航路线图

4.2. 避障及路径规划实验

本文的平台搭建、自动巡航功能实现和自定义主题的添加都是为了实现最终的避障及路径规划实验。在前文研究的基础上, 最终构建避障及路径规划的整体流程如下图 12 所示。

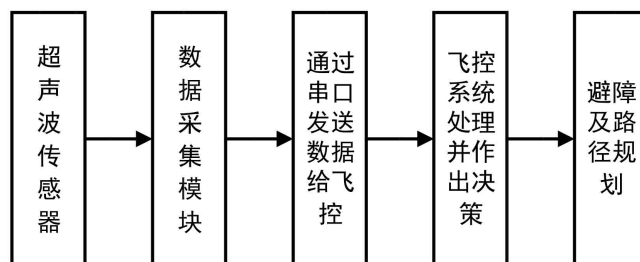


Figure 12. Obstacle avoidance system frame diagram

图 12. 避障系统框架图

首先通过数据采集模块实现超声波传感器的数据采集功能, 然后将采集到的数据通过串口发送给飞控板, 接下来按照 uORB 消息机制添加自定义超声波模块, 对超声波数据主题进行发布和订阅, 最后在手模式模式和自动模式下订阅超声波数据, 并对获得的数据进行处理实现无人车的避障及路径规划。

为了验证文中提出方法的有效性, 下面分别在手动模式和自动模式下对无人车进行避障测试。实验中使用大小为 20 cm * 30 cm * 50 cm 的纸箱做障碍物, 实验前首先确保超声波传感器及地面无人车均可正常工作, 分别在正前方障碍物和两侧障碍物的情况下对避障效果进行检测。

由于目前业界并没有统一的测试标准和评价指标来对避障效果进行量化考核, 因此在测试的时候通常会从障碍物的个数、障碍物的运动状态、运动轨迹是否平滑这几个方面进行考虑。结合实际的场景, 本文以安全避障和耗时最少作为评价指标, 对无人车的避障及路径规划过程进行测试。下面对实验现象和结果进行简要分析。

4.2.1. 正前方障碍物实验与分析

本小节对障碍物在正前方的情况进行实验分析。首先在地面站设置飞行任务, 进行航线规划, 并设置速度参数, 上传到飞行控制系统。地面无人车以 0.8 m/s 的速度巡航行驶, 当距离障碍物 3 m 的时候, 无人车检测到障碍物, 并开始降低巡航速度, 进入避障准备阶段, 障碍物位置如图 13 所示。当行驶至距

离障碍物 2 m 的时候, 无人车进入避障区域, 执行避障操作向左转弯 45° , 绕障碍物行驶如图 14 所示。无人车在绕过障碍物之后返回规划航线中, 并沿规划好的航线继续行驶, 如图 15 所示。整体的避障及路径规划行驶路径如图 16 所示, 图中黄色航线为设定航线, 紫色为无人车实际运行路线, 其中障碍物位于航点 1 和航点 2 之间, 从图中紫色路径可以看到, 无人车在遇到障碍物时, 通过向左绕行实现避障, 并在绕过障碍物后快速回到预设航线中。对多次实验的避障过程时间进行统计发现, 从检测到障碍物开始到回到预设航线中, 耗时在 9~11 s 之间, 符合实际应用中的要求。

实验中设置自动巡航行驶过程中速度为 0.8 m/s。当检测到障碍物距离为 3 m 时, 开始减速行驶, 速度逐渐降低至避障速度 0.5 m/s。经过 1.5 s 左右无人车执行变向避障操作, 转向过程中 U2 超声波逐渐检测不到障碍物。在此过程中以 0.5 m/s 左右的速度行驶, 当绕开障碍物后速度回到预设航线中逐渐回到设定值 0.8 m/s。实验过程中速度的最大误差为 0.2 m/s, 符合误差要求。



Figure 13. Obstacle detected

图 13. 检测到障碍物



Figure 14. Bypass obstacles

图 14. 绕开障碍物



Figure 15. Back to the route

图 15. 回到航线



Figure 16. Obstacle avoidance path display
图 16. 避障路径显示

4.2.2. 两侧障碍物实验与分析

本小节对障碍物在左前方和右前方两种情况分别进行实验和分析。

1) 障碍物在左前方

当地面无人车按照航线前进过程中检测到左前方有障碍物时, 超声波 U1 和 U2 均有数据输出, 障碍物位置如图 17 所示。图 18 为地面无人车在自动巡航模式下左前方遇到障碍物时通过地面站显示的避障路径图, 紫色曲线为无人车的实际运行路径, 黄色曲线为规划路径, 障碍物位于航点 1 和航点 2 中间。在实验中, 地面无人车在检测到障碍物之后开始减速, 当行驶至距离障碍物 2 m 的时候, 开始往右侧 45° 转弯进行绕障碍物行驶运动。在绕过障碍物之后返回规划航线, 并沿规划好的航线继续行驶。

2) 障碍物在右前方

在同样的路径规划任务下, 只改变障碍物的位置, 模拟在无人车行进过程中右前方出现障碍物的情况。此时超声波 U2 和 U3 均有数据输出, 障碍物位置如图 19 所示。图 20 为地面无人车在自动巡航模式下, 右前方遇到障碍物时地面站显示的避障路径图, 可以看出障碍物位于航点 1 和航点 2 中间。与左前方出现障碍物时的避障操作相似, 执行一系列的减速动作后, 通过向左侧转向的方式来实现避障。

上述实验表明, 地面无人车在自主巡航时两侧遇到障碍物, 能够顺利地避开障碍物, 并继续回到预设航线中。在左右障碍物情况下, 经多次实验测试, 从开始避障到回到预设航线上的时间在 9~11 s 之间。



Figure 17. U1, U2 ultrasonic detection of obstacles
图 17. U1、U2 号超声波检测到障碍物



Figure 18. Left front obstacle avoidance path display

图 18. 左前方避障路径显示



Figure 19. U2, U3 ultrasonic detection of obstacles

图 19. U2、U3 号超声波检测到障碍物



Figure 20. Right front obstacle avoidance path display

图 20. 右前方避障路径显示

5. 总结

本文采用 Pixhawk 飞控作为主控器, 在对无人车进行建模的基础上, 结合串级 PID 控制算法实现无人车的避障及路径规划功能。本文提出的避障及路径规划方法, 除了在无人车上应用, 还可以扩展到多旋翼无人机的避障中, 实现无人机飞行过程中的避障。

实验表明, 本文提出的避障及路径规划策略在室外场景下获得较好的检测效果。相比传统的单片机

实现方式, 飞控具有强大的数据处理能力, 能够实现更稳定的避障及路径规划效果。与常用的激光雷达实现避障及路径规划的方案相比, 本文提出的基于超声波的避障方案成本更低、实现更加简单, 可以应用到校园巡逻、厂区安全检查等精度要求不是很高的场景中。后续的研究, 可以通过加入其他传感器, 与超声波数据进行融合, 来实现更复杂环境下的避障及路径规划。

参考文献

- [1] 朱炳先, 张朋飞. 地面无人车辆及关键技术研究进展[J]. 山西大同大学学报(自然科学版), 2017, 33(1): 66-70.
- [2] 赵涓涓, 杜麒麟, 屈明月, 等. 智能小车的无支路循迹算法设计[J]. 计算机应用与软件, 2013, 30(6): 147-149.
- [3] 乔维高, 徐学进. 无人驾驶汽车的发展现状及方向[J]. 上海汽车, 2007(7): 40-43.
- [4] 冯华勇. 基于智能小车的模糊寻迹控制[J]. 制造业自动化, 2010, 32(10): 49-51.
- [5] 章志祥. 四旋翼机器人飞行测试及自适应动态逆方法研究[D]: [硕士学位论文]. 北京: 北方工业大学, 2014.
- [6] 耿特, 严军鹏. 基于卷积神经网络(CNN)的无人车避障方法[J]. 工业控制计算机, 2019, 32(2): 61-64.
- [7] 曾迪, 余瑶. 基于模糊神经网络系统的智能小车避障[J]. 数字通信, 2014, 41(3): 83-85.
- [8] 赵航, 王立峰. 基于 pixahwk 的多旋翼无人机避障飞行系统研发[J]. 动力系统与控制, 2017, 6(3): 98-108.
- [9] 赵伦, 王立峰, 董晓婉. 六旋翼飞行器动态逆控制器设计[J]. 建模与仿真, 2016, 5(2): 16-25.
- [10] Kong, J., Pfeiffer, M., Schildbach, G., *et al.* (2015) Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. 2015 *IEEE Intelligent Vehicles Symposium (IV)*, Seoul, 28 June-1 July 2015, 1094-1099.
<https://doi.org/10.1109/ivs.2015.7225830>

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-677X, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: dsc@hanspub.org