

## 差速轮式移动机器人的 导航算法研究及仿真实验

Navigation algorithms and the simulation experiment  
of differential wheeled mobile robot

马国庆, 刘 丽, 于正林, 曹国华

MA Guo-qing, LIU Li, YU Zheng-lin, CAO Guo-hua

(长春理工大学 机电工程学院, 长春 130022)

**摘 要:** 针对差速轮式移动机器人的导航算法中的直线和圆弧导航只是针对单一角度变量进行纠偏, 且越接近终点时纠偏效果越剧烈, 当PID参数很大时, 会导致很大超调, 因此对该算法进行了改进, 提出了新的导航算法。首先介绍差速轮式移动机器人的定位算法和导航算法, 然后分析导航存在的问题及产生的原因并提出了新的导航方法, 最后建Simulink模型进行仿真, 仿真结果显示新的导航方法取得了良好的效果。

**关键词:** 差速轮式机器人; 导航算法; Simulink

**中图分类号:** TM383.6

**文献标识码:** B

**文章编号:** 1009-0134(2018)05-0028-04

### 0 引言

轮式移动机器人主要有4种结构形式被广泛的熟知, 即前轮驱动并可转向移动机器人、两轮差动驱动的移动机器人、汽车式移动机器人、带拖车的轮式移动机器人、三轮全向移动机器人<sup>[1]</sup>。移动机器人因具有多种机构形式, 使得它的应用也渗入到各个领域, 包括深海无人潜水器、无人飞行器、服务机器人和工业机器人等。移动机器人的广泛应用使得人们的生活和生产发生了很大的变化, 提高了生产效率和生活质量。因此移动机器人的协调和协作、导航定位、路径规划、控制等问题的研究引起了广大科研工作者的关注。其中自主导航技术是其研究的核心<sup>[2,3]</sup>, 而定位是完成导航首先要解决的问题<sup>[4]</sup>。

本文主要对轮式移动的差速底盘的机器人的定位导航问题展开研究, 着重分析导航算法中直线导航和圆弧导航, 针对这两种导航算法只是基于单一的角度变量进行纠偏的问题进行分析和改进, 并提出新的改进算法, 为移动机器人的更深入的研究奠定坚实的基础。

如图1所示为移动机器人路径规划导航定位结构图。

### 1 轮式机器人差速定位算法

#### 1.1 差速轮式机器人的系统介绍

差速轮式移动机器人, 主要依靠后边的驱动轮驱动

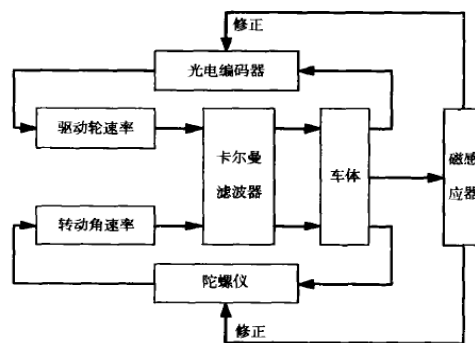


图1 移动机器人导航结构图

前方的轮子进行控制, 使机器人按照预定的轨迹行走。这种结构底盘的机器人灵活转向, 依靠速率和方向的不同, 实现直线行走和转弯的目的<sup>[5]</sup>。

本系统机器人采用光纤陀螺与光电码盘实现机器人的定位, 使用光纤陀螺的信号可以读取相对于起始点的角度数据, 这样就可以通过相对位置、角度信息解算得到机器人相对于起始坐标系的坐标信息, 从而实现机器人的定位。

#### 1.2 轮式机器人差速定位

如图2所示为系统定位的示意图,  $[xy\theta]$ 表示机器人

收稿日期: 2017-12-18

基金项目: 吉林省科技发展计划资助项目(20160204016GX); 吉林省省级产业创新专项资金资助项目(2016C088); 长春理工大学青年科学基金(XQNJJ-2016-04); 吉林省教育厅“十三五”科学技术项目(JJKH20170626KJ); 吉林省省级产业创新专项资金项目(2017C045-2)

作者简介: 马国庆(1988-), 男, 吉林长春人, 讲师, 硕士, 研究方向为机电系统控制和机器人技术等。

的系统定位信息。

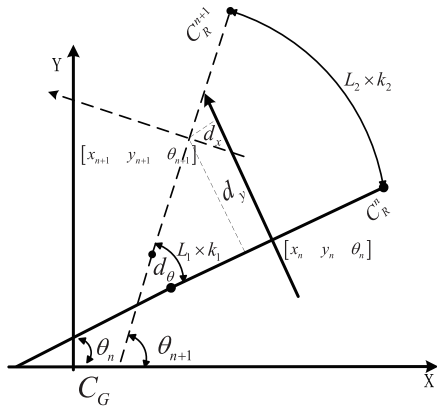


图2 系统定位示意图

图中： $x, y$ 为机器人全局坐标系下的坐标； $\theta$ 为机器人的方位角； $C_G, C_R$ 为全局、局部坐标系。

假设机器人由第 $m$ 个状态到达第 $m+1$ 个状态时，即经实线到达虚线时，产生很小的位移量。从图中可以看出从动轮会行走一段很短的圆弧。假定机器人左侧编码器的变化量为 $L_1$ ，右边的变化量为 $L_2$ 。 $N$ 表示编码器的分辨率， $r$ 表示从动轮半径， $Len$ 为两边编码器从动轮的中心轮距<sup>[6]</sup>。则有以下公式：

$$d_\theta = \frac{L_2 \times k_2 - L_1 \times k_1}{Len} \quad (1)$$

式中： $d_\theta$ 表示机器人旋转的角度， $k_1 = 2\pi r_1 / N$ 和 $k_2 = 2\pi r_2 / N$ 表示两侧编码器的矫正系数，再由 $d_\theta$ 值能够求出在 $C_R$ 中 $x, y$ 方向的变化量：

$$\begin{cases} d_x = -\frac{L_2 \times k_2 + L_1 \times k_1}{2(L_2 \times k_2 - L_1 \times k_1)} \times Len \times [1 - \cos(d_\theta)] \\ d_y = \frac{L_2 \times k_2 + L_1 \times k_1}{2(L_2 \times k_2 - L_1 \times k_1)} \times Len \times \sin(d_\theta) \end{cases} \quad (2)$$

进行以下公式的变换，可以将 $d_y, d_x, d_\theta$ 转换为在全局坐标系中的角度与位置信息：

$$\begin{cases} \theta_{n+1} = \theta_n + d_\theta \\ x_{n+1} = x_n + [d_x \times \cos(\theta) - d_y \times \sin(\theta)] \\ y_{n+1} = y_n + [d_x \times \sin(\theta) + d_y \times \cos(\theta)] \end{cases} \quad (3)$$

其中 $[x_n, y_n, \theta_n]$ 表示前一时刻的定位信息， $[x_{n+1}, y_{n+1}, \theta_{n+1}]$ 表示当前时刻的位置信息。得到的坐标信息和陀螺得到的坐标信息做数据融合后经过累加得到机器人的实时坐标信息。

由以上的运算可知，这个过程存在累积误差，经过长时间的作业之后，累计误差使机器人的定位信息出现偏差。以下使用累计误差修正算法对误差进行消除<sup>[7]</sup>。

圆形的中心与从动轮中心的距离为1，圆的坐标为 $[X_{Island}^n, Y_{Island}^n]$ ，则机器人的当前的位置信息如下公式所示：

$$\begin{cases} x = X_{Island}^n + l \times \sin(\theta) \\ y = Y_{Island}^n + l \times \cos(\theta) \end{cases} \quad (4)$$

实验表明，通过以上的累计误差修正算法的机器人，在没有受到严重的破坏的条件下能够很好的实现系统的定位信息，文献[7]也通过具体的实例验证了这种累计误差修正法的有效性和正确性。

## 2 直线导航算法

将差速底盘机器人底盘的定位信息通过串口传送到以ARM和FPGA为控制核心的主机上，在主机上采用直线、圆弧导航算法实现机器人的导航。通过分析和对比，发现了这两种导航算法中存在的问题，以下分别对这两种导航算法进行了改进。提高了机器人导航的精度。

### 2.1 直线导航算法

如图3所示为机器人直线导航算法的示意图，根据系统的位置与角度来实现机器人的反馈。利用目标点和编码器中心点的连线和当前机器人行走方向的夹角作为反馈参数，使用机器人与目标点的距离作为判断是否到达的条件<sup>[8]</sup>。

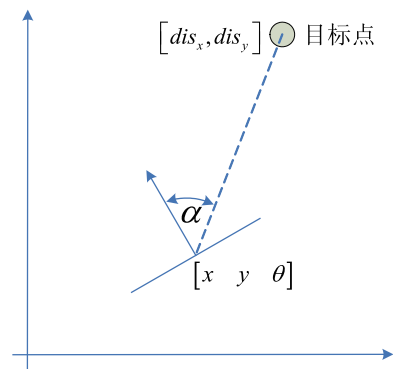


图3 直线导航算法示意图

如图3所示，机器人车头方向与目标点的夹角为 $\alpha$ ，目标点的坐标为 $[dis_x, dis_y]$ ，利用下面的公式可以将目标点的坐标转化为机器人的局部坐标值：

$$X_{local} = (dis_x - x) \times \cos(\theta) + (dis_y - y) \times \sin(\theta) \quad (5)$$

$$Y_{local} = -(dis_x - x) \times \sin(\theta) + (dis_y - y) \times \cos(\theta) \quad (6)$$

利用 $\alpha = \pm \frac{\pi}{2} - \arctg(\frac{Y_{local}}{X_{local}})$ 得到 $\alpha$ 值（当 $X_{local} < 0$ 时取

负号)，由 $\alpha$ 的值使用PID反馈调节，通过下面的公式对车头方向进行调整。

$$PIDresult = \alpha_n P + (\alpha_n - \alpha_{n-1}) D + \alpha_n I \quad (7)$$

式中，PIDresult为调整量， $\alpha_n$ 与 $\alpha_{n-1}$ 为当前时刻和上一时刻的 $\alpha$ 值。

## 2.2 改进的直线导航算法

上述的导航算法是针对角度单一变量进行纠偏，越接近终点时纠偏效果越剧烈，当PID参数很大时，会导致很大超调，因而该算法需要做一定改进，设计直线导航算法：

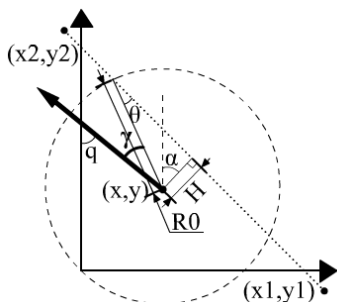


图4 改进的直线导航示意图

在平面坐标系中，机器人位置为 $(x, y)$ 朝向角度为 $q$ 。设定机器人半径为 $R0$ ，目标直线为由点 $(x1, y1)$ 和 $(x2, y2)$ 确定。有以下关系：

$$(y2 - y1)(x - x1) - (x2 - x1)(y - y1) = 0 \quad (8)$$

H为点到直线距离：

$$H = \frac{(y2 - y1)(x - x1) - (x2 - x1)(y - y1)}{\sqrt{(y2 - y1)^2 + (x2 - x1)^2}} \quad (9)$$

又有：

$$\sin \theta = \frac{H}{R0} \quad (10)$$

$$\tan \alpha = \frac{x1 - x2}{y1 - y2} \quad (11)$$

$$\alpha = q - \gamma + \theta \quad (12)$$

可以求出角度 $\gamma$ ，然后对角度 $\alpha_n$ 进行PID控制就行了。

$$\gamma = q + \arcsin \frac{H}{R0} - \arctan \frac{x1 - x2}{y1 - y2} \quad (13)$$

## 3 圆弧导航

圆弧的导航可以参照图3。只不过目标点改为了圆心，我们所要PID的角度变成了机器人现有角度与圆切线间的夹角。利用下面的公式可以把目标点的坐标转化

为机器人的局部坐标：

$$X_{local} = (dis_x - x) \times \cos(\theta) + (dis_y - y) \times \sin(\theta) \quad (14)$$

$$Y_{local} = -(dis_x - x) \times \sin(\theta) + (dis_y - y) \times \cos(\theta) \quad (15)$$

之后使用 $\alpha = \arctg(\frac{Y_{local}}{X_{local}})$ 来求出 $\alpha$ 值。

调整公式依然是：

$$PIDresult = \alpha_n P + (\alpha_n - \alpha_{n-1}) D + \alpha_n I \quad (16)$$

## 3.1 改进的圆弧导航算法

由于原有导航算法是针对角度单一量进行纠偏，越接近目标时纠偏效果越剧烈，当PID参数很大时，会导致很大超调，因而该算法需要做一定改进，设计直线导航算法圆弧导航算法，设计算法图示如图5所示。

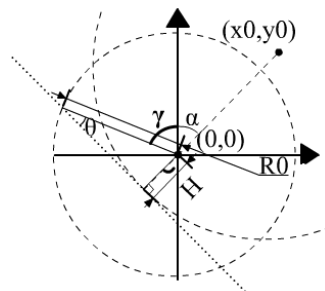


图5 圆弧导航示意图

首先将目标圆心进行坐标系转化，得到其在机器人的坐标系（机器人所在位置为原点，机器人运动方向为x轴方向）内的坐标 $(x0, y0)$ 。以直线 $y = \frac{y0}{x0}x$ 与目标圆交点处切线方向进行改进的直线导航即可。有：

$$H = R - \sqrt{x0^2 + y0^2} \quad (17)$$

$$\sin \theta = \frac{H}{R0} \quad (18)$$

$$\tan \alpha = \frac{x0}{y0} \quad (19)$$

$$\gamma + \alpha = \theta + \frac{\pi}{2} \quad (20)$$

可以求出角度 $\gamma$ ：

$$\gamma = \arcsin \frac{H}{R0} - \arctan \frac{x0}{y0} + \frac{\pi}{2} \quad (21)$$

对角度 $\gamma$ 进行PID之后，对于轨迹是直线和圆弧相切时，导航的目标点也是一个连续变化的点，不会出现导航的突变，机器人的运动会更平滑精确。

## 3.2 驱动轮速分配

基于对多种情况的考虑, 可以将速度分配写成一个通式,  $V, V_1, V_2$  表示机器人实时的平均、右轮、左轮速度。

$$\begin{cases} V_1 = (1 - \frac{L}{2R} - U) \cdot V \\ V_2 = (1 + \frac{L}{2R} + U) \cdot V \end{cases} \quad (22)$$

式中:  $R$  为圆弧半径 (向量), 规定圆心在路径右侧为正, 左侧为负;  $L$  为驱动轮间的距离;  $U$  为由偏差产生的控制量;

将  $U$  定义为向量, 基于不同的情况进行考虑, 总结控制量与偏差量符号变化情况, 将其归纳为一个统一公式。

$$E = K\gamma \times \gamma + KH \times H \quad (23)$$

之后再将偏差  $E$  经过 PID 控制, 然后使用控制量  $U$  分配左右轮的速度从而达到纠偏的用途。

## 4 Simulink 仿真实验

针对以上提出的改进算法, 对圆弧导航算法的改进前后进行仿真验证, 仿真数学模型如式(24)所示:

$$\begin{cases} H = H_0 + \int_0^t \frac{V_1 + V_2}{2} \cdot \sin \gamma \cdot dt \\ \gamma = \gamma_0 - \int_0^t \frac{V_1 - V_2}{L} \cdot dt \\ V_1 = (1 + K\gamma \cdot \gamma + KH \cdot H) \cdot V \\ V_2 = (1 - K\gamma \cdot \gamma - KH \cdot H) \cdot V \end{cases} \quad (24)$$

上述模型能够表示出机器人在行进时轮速与偏差量随时间变化的关系,  $H, \gamma$  为距离与角度偏差初值;  $H, \gamma, V_1, V_2, V, KH, K\gamma$  如上述介绍; 为了让仿真变得简单, 设定  $KH$  的值, 只让  $K\gamma$  的值发生改变, 如图6所示的就是仿真得到的改进前后的圆弧导航算法的路径。

图中实线为改进后的圆弧导航算法的路径, 虚线改进前的圆弧导航算法的路径, 横纵坐标轴的都是以 mm 计量的。由图6知, 改进前与改进后在初始时基本没有差异, 但是在最后只有大约 15mm 的误差。从导航数据可以分析出, 此偏差主要是由陀螺引起的。由陀螺传送的与真实角度存在一个误差  $\Delta\alpha$ , 机器人就会纠偏,  $\Delta\alpha$  主要有以下两点产生:

1) 由于本文使用的是单轴陀螺, 即使旋转一圈是  $2\pi$ ,  $\Delta\alpha$  会出现从 0 逐步增大, 再逐不减小到 0 的过程。

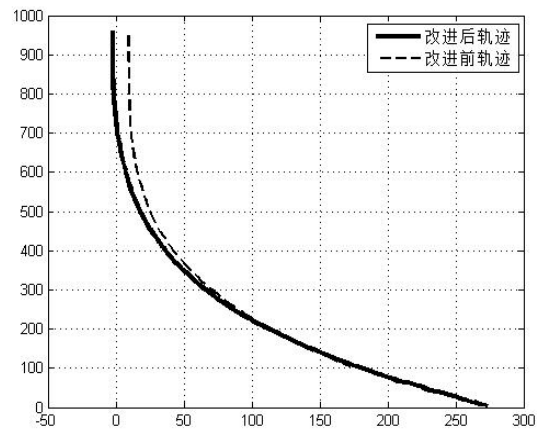


图6 改进前与改进后结果对比图

2) 理想中的陀螺的精度可以满足实际的情况, 但是现实这一定会有误差。

## 5 结论

轮式的差速底盘移动机器人的导航算法中的直线和圆弧导航只是针对单一角度变量进行纠偏, 且越接近终点时纠偏效果越剧烈, 当 PID 参数很大时, 会导致超调很大, 因此对该算法进行了改进, 提出了新的导航算法。最后建立 Simulink 模型进行仿真。仿真结果表明新的导航方法取得了良好的效果。

## 参考文献:

- [1] 沈猛. 轮式移动机器人导航控制与路径规划研究[D]. 陕西: 西北工业大学, 2006: 3-4.
- [2] Bashiri M, Vatankhah H, Ghidary S. Hybrid adaptive Differential Evolution for Mobile Robot Localization[J]. Intelligent Service Robotics. 2012, 5(2): 99-107.
- [3] Cheng P Y, Chen P J. Navigation of mobile robot by using D+algorithm[J]. Intelligent Service Robotics. 2012, 5(2): 229-243.
- [4] 贝旭颖, 平雪良, 高文研. 轮式移动机器人里程计系统误差校核[J]. 计算机应用与研究, 2017, 35(9).
- [5] 亢海龙, 徐娅萍, 王涛, 等. 基于卡尔曼滤波器的轮式机器人定位方法[J]. 机械与电子, 2009, 72-74.
- [6] 王鸣. 机器人实时定位与导航系统仿真设计研究[J]. 计算机仿真, 2012, 12(03): 216-218.
- [7] 李毅, 吴焱岷, 宏英. 机器人实时定位与导航系统的设计研究[J]. 制造业自动化, 2012, 34(4): 132-134.
- [8] 朱磊磊, 陈军. 轮式移动机器人研究综述[J]. 机床与液压, 2009, 37(8): 242-247.