

文章编号: 1002-0446(2007)02-0117-06

基于 MSP430 的自动巡线机器人姿态调整方法研究

宋波, 肖世德, 华仕容

(西南交通大学机械工程学院, 四川 成都 610031)

摘要: 将机器人与引导线的相对位置关系分成典型的七种状态, 控制系统分别就这七种状态做出相应的调整策略. 论证了在这种机构设计下通过不断调整底盘姿态的方法可以让底盘中心线逐渐与引导线重合的实现原理. 基于 MSP430 平台采用 PWM 来控制左右驱动轮的转速以产生转速差, 来调整机器人与引导线的相对位置, 实现了快速、稳定、准确的巡线.

关键词: 表征量; 姿态; PWM; 速度差**中图分类号:** TP24 **文献标识码:** B

Posture Adjustment of an Autonomous Line-tracking Robot Based on MSP430

SONG Bo, XIAO Shi-de, HUA Shi-rong

(School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: The relative position relationship between the robot and the leading line is divided into seven typical states, according to which the control system adopts an adjustment strategy respectively. An implementation principle using this mechanism design is demonstrated by which the chassis centerline can approach the leading line gradually through continuously adjusting the chassis posture. Based on MSP430, the speed difference between the left and right wheels is produced by the Pulse-Width Modulation to adjust the relative position of the robot and the leading line, and quick, stable and accurate line-tracking is realized.

Keywords: token state; posture; PWM; speed difference

1 引言 (Introduction)

全自动移动机器人在工业自动化中的使用主要表现在针对特定的路面, 找到适合自身机械特点的定位方法, 通过不断调整到达目标, 最终完成设定的任务. 如何采用简洁、快速、稳定的控制方法和检测设备一直是工业机器人的关键. 一般而言, 在编制工业机器人的运动策略时会根据所在场地的具体情况制定特定的行进方法^[1]. 场地如图 1 所示, 它是在深蓝色地板上放置若干交叉白色引导线, 机器人在场地上行进到目标后执行后续动作. 问题的核心还是如何让机器人准确地到达目标. 因此, 根据场地特征采用适合机器人自身机械特性的算法让机器人实现准确行走定位是关键. 本文首先根据机器人的机械特征, 分析了通过直流电机的起或停控制机器人调整的可能性, 然后把传感器检测到的机器人底盘与

引导线的相对位置划分为典型的 7 种状态, 做出相应的调整策略. 继而分析做出这种调整策略会校正机器人姿态的可行性原理, 再具体到 TI 公司的低功耗系列单片机 MSP430F449 的实现方法. 最后, 改进了这种一停一动的校正方法, 提出了一个新方法, 用 PWM 控制左右电机的转速产生速度差, 由于左右驱动轮的速度差值小了, 因此可使机器人在调整的过程中减轻晃动, 并使行走加快.

2 总体分析 (General analysis)

如图 1 所示场地, 黑色交叉线条实际代表线宽 30 mm 的白色引导线, 白色底色代表深蓝色地板. 引导线构成了宽度为 500 mm 的正方形格子.

机器人底盘长 500 mm, 宽 650 mm. 如图 3 所示,

后排左右轮是两个独立的由直流电机驱动的轮子, 前排放置两个万向轮. 直流电机空载力矩大, 可以很好地控制所驱动的轮子起或停, 万向轮在任何方向都可以实现运动, 因此整个机器人的行走方向是依靠后排两个驱动轮的起动或者停止来改变的.

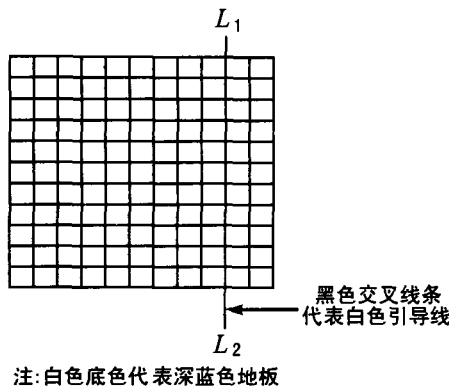


图1 场地外形及尺寸
Fig. 1 Shape and dimension of the ground

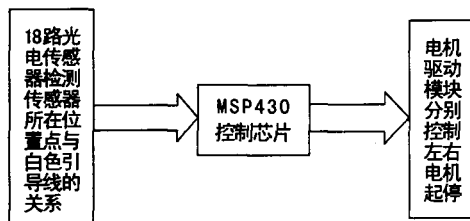


图2 信息采集和控制方法
Fig. 2 The information collection and control method

由于底盘安装精度和后排左右驱动轮子之间的转动速度不可能完全一致, 因此即使驱动板向左右驱动电机输出的功率相同, 也会使机器人的中心线偏离引导线. 如图2所示, 机器人通过后排左右轮起或停从而调整底盘姿态, 目的是保持底盘中心线与白色引导线重合, 从而让机器人沿着引导线行进到达目的地. 所采用的传感器是自制的光电传感器, 当检测到的颜色是路面底色(深蓝色)时向单片机输出高电平; 当检测到的颜色是引导线颜色(白色)时向单片机输出低电平. 这些向单片机输入的高低电平实际是底盘与引导线的相对位置的表征量, 单片机根据这些表征量来做出调整输出, 通过电机驱动板控制直流电机的起或停.

机器人的底盘整体较大, 所以布置在底盘的传感器应尽量多, 这样可以检测到底盘的多个位置, 做出相应位置的快速响应.

如图3所示传感器布置方法, 传感器采用20个, 其中18个用于机器人姿态定位, 2个用于计算所走格子数. 前后两排传感器分别居于前后轮的中心线上. 在这种排布方式下, 某排传感器检测到水平白色引导线时, 对应该排的轮中心刚好在白色引导线上, 可方便转弯和在交叉引导线处及时调整. 每排每个传感器的间距为40 mm. 对其标号 $S_1 \sim S_{20}$.

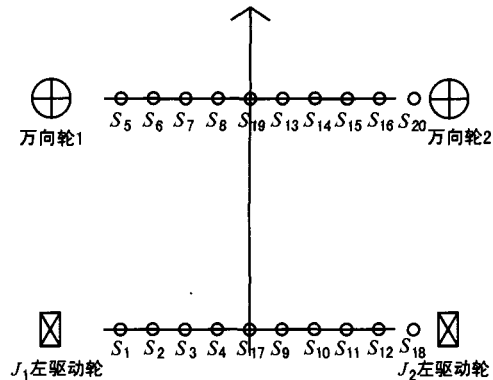


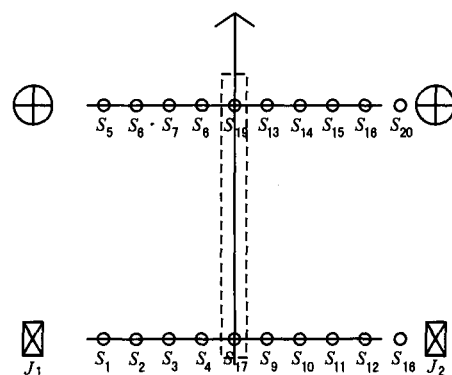
图3 传感器布局图(J_1 、 J_2 是转动轮中心)

Fig. 3 Position of sensors

(J_1, J_2 is the center of the driving wheels)

3 姿态分析 (Posture analysis)

根据18路传感器各自所在位置与引导线的相对关系表征量, 可以确定此时底盘与引导线的偏移情况. 通过不同的优先级处理这些状态量校正底盘的偏移. 核心思想是让底盘中心线上的传感器 S_{17} 、 S_{19} 不断地向引导线靠近, 使得整个车身以引导线为中心线. 问题就转化为让传感器尽量多地包围引导线, 此种做法可以保证在调整的过程中至少可以检测到两个引导线, 不至于前排或后排传感器都向一侧偏离, 使得最终巡线失败; 并且可以保证在调整的过程中, S_{17} 、 S_{19} 是不断向引导线靠拢的.



(a)

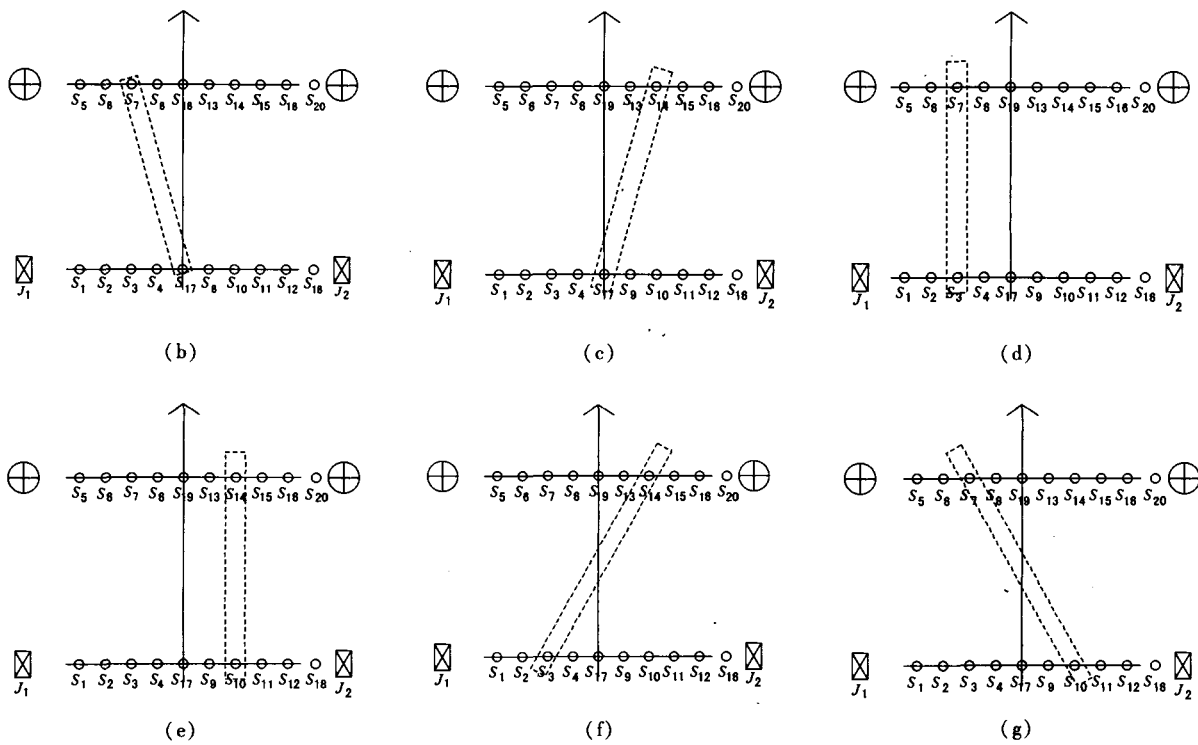


图 4 状态分析
Fig. 4 Posture analysis

(注:在图 4 中长条矩形代表宽 30 mm 的白色引导线)

根据图 4(a) ~ (g) 的缩略图,通过分析底盘与引导线偏移状态研究出相应的调整法则:

①图 4(a),前后中心传感器 S_{17} 、 S_{19} 检测到引导线,底盘中心线与引导线重合,说明机器人在所期待的正确位置,机器人直走,两后轮驱动电机同时转动。

②图 4(b),后排中心传感器 S_{17} 检测到引导线,前排左侧 S_5 、 S_6 、 S_7 、 S_8 中的一个检测到引导线。机器人底盘右侧大部分偏出,应该向左偏移。驱动轮左轮停,右轮转动,向左移动使底盘中心靠近引导线。

③图 4(c),后排中心传感器 S_{17} 检测到引导线,前排右侧 S_{13} 、 S_{14} 、 S_{15} 、 S_{16} 中的一个检测到引导线。机器人底盘左侧大部分偏出,应该向右偏移。驱动轮左轮转动,右轮停,向右移动使底盘中心靠近引导线。

④图 4(d),后排左侧传感器 S_1 、 S_2 、 S_3 、 S_4 中的一个与前排左侧 S_5 、 S_6 、 S_7 、 S_8 中的一个同在引导线上。机器人底盘右侧大部分偏出,应该左转。驱动轮左轮停,右轮转动,车身向左移动。

⑤图 4(e),后排右侧传感器 S_9 、 S_{10} 、 S_{11} 、 S_{12} 中的一个与前排右侧 S_5 、 S_6 、 S_7 、 S_8 中的一个同在引导线上。机器人底盘左侧大部分偏出,应该右转。驱动轮左轮转动,右轮停,车身向右移动。

⑥图 4(f),后排左侧传感器 S_1 、 S_2 、 S_3 、 S_4 中的一个与前排右侧传感器 S_{13} 、 S_{14} 、 S_{15} 、 S_{16} 中的一个同在引导线上或者后排左侧传感器 S_1 、 S_2 、 S_3 、 S_4 中的一个在引导线上且前排中心传感器 S_{19} 在引导线上,则右转。让传感器多数进入引导线附近,以免传感器都检测不到引导线而出现盲区。

⑦图 4(g),后排右侧传感器 S_9 、 S_{10} 、 S_{11} 、 S_{12} 中的一个与前排左侧传感器 S_5 、 S_6 、 S_7 、 S_8 中的一个同在引导线上或者后排右侧传感器 S_9 、 S_{10} 、 S_{11} 、 S_{12} 中的一个在引导线上且前排中心传感器 S_{19} 在引导线上,则左转。让传感器多数进入引导线附近,以免传感器都检测不到引导线而出现盲区。

①~⑤的思想是使机器人底盘的正中传感器向引导线靠近,使得机器人沿着引导线向前移动。

⑥、⑦的思想是使传感器尽量多地居于引导线附近,以免传感器全都偏离出引导线而检测不到自身的位置,就无法进行判断了。

4 可行性原理分析(Feasibility analysis)

假设车身首先在状态 7,如图 5 所示,底盘中心线与白色引导线产生了一定夹角,引导线被 S_{11} 和 S_7 检测到。这种情况应该满足让传感器尽量多地靠近引导线的原则,以免动作过大后前后排传感器没有

任意两个检测到引导线而无法获得底盘与引导线的相对位置,最终使后面巡线失败.所以,首先左转,转动中心是左轮中心 J_1 ,前后两排传感器均以 J_1 为中心旋转.当 S_{13} 进入引导线(位置为 S'_{13})后状态7结束,进入状态5.在这个转动过程中, S_{13} 的转动角度为 α ,根据以 J_1 为转动中心的所有质点的角速度相同原理, S_{11} 、 S_{10} 、 S_9 均转动 α 分别到达 S'_{11} 、 S'_{10} 、 S'_9 ,与 L_1L_2 的距离由原来的 0 、 d_{10} 、 d_9 分别变化为 d'_{11} 、 d'_{10} 、 d'_9 .

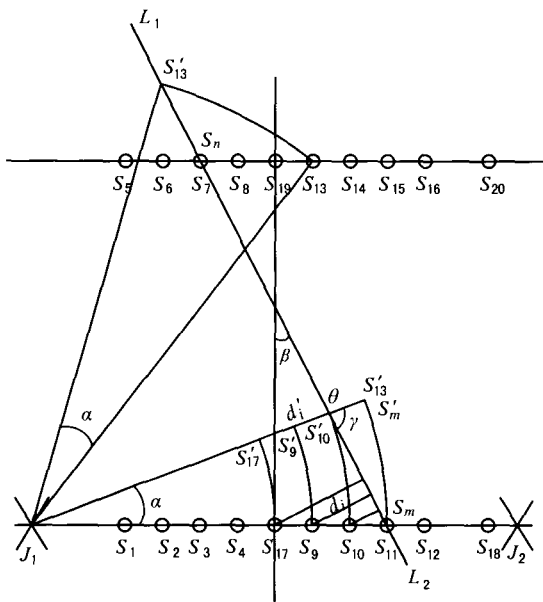


图5 状态7分析
Fig. 5 Analysis on posture 7

不失一般性,这里的 S_{11} 用 S_m , S_7 用 S_n 来代替.设后排 S_m 左侧靠近底盘中心线的传感器为 S_i . S_i 转动前后与引导线的距离分别为 d_i 和 d'_i ($m > i$).

记两排传感器的垂直距离为 L ,即 $S_{17}S_{19} = L$;每个传感器的间距为 s_0 , $S_{17}S_m = ms_0$, $S_{19}S_n = ns_0$.

底盘中心线与引导线 L_1L_2 的夹角 $\beta = \arctan \frac{S_{17}S_m + S_{19}S_n}{S_{17}S_{19}} = \arctan \frac{(m+n)s_0}{L}$.

在转动前,传感器 S_i 到 L_1L_2 的距离为: $d_i = S_m S_i \cos \beta = (m-i)s_0 \cos \beta$.

记 J_1 、 J_2 到底盘中心线的距离为 J , $J_1S'_m$ 与 L_1L_2 的交点为 q , 夹角为 γ :

$$\gamma = \alpha + (\pi/2 - \beta) = \pi/2 + \alpha - \beta$$

在 $\Delta J_1 S_m q$ 中,有: $\frac{J_1 q}{\sin(\angle J_1 S_m q)} = \frac{J_1 S_m}{\sin \gamma}$; 得到:

$$J_1 q = \frac{\cos \beta}{\cos(\beta - \alpha)} J_1 S_m; \text{ 则:}$$

$$\begin{aligned} d'_i &= (J_1 q - J_1 S_i) \sin \gamma \\ &= \left(\frac{\cos \beta}{\cos(\beta - \alpha)} J_1 S_m - J_1 S_i \right) \cos(\beta - \alpha) \\ &= J_1 S_m \cos \beta - J_1 S_i \cos(\beta - \alpha) \\ \Delta d &= d_i - d'_i \\ &= S_m S_i \cos \beta - [J_1 S_m \cos \beta - J_1 S_i \cos(\beta - \alpha)] \\ &= J_1 S_i [\cos(\beta - \alpha) - \cos \beta] \end{aligned}$$

以上采用工程软件 MATLAB 6.0 计算^[2].

根据底盘的机械参数 $\beta - \alpha < \beta$, 可以判断调整前后底盘中心部分传感器与引导线距离的差值是大于0的.对于这种情况, S_i 与引导线的距离是不断缩小的.

所以可以推知,在绕左轮直到前排右侧的传感器检测到引导线的转动过程中,会逐渐让 S_m 左侧靠近中心线的传感器进入引导线区域.然后进入状态5.

状态5如图6所示,这种情况应该满足中心传感器尽量靠近引导线的原则,此时应该右转,前后两排传感器以右轮为中心旋转.整个过程是在前排中心传感器 S_{19} 检测到引导线后停止的(此种情况进入状态7).

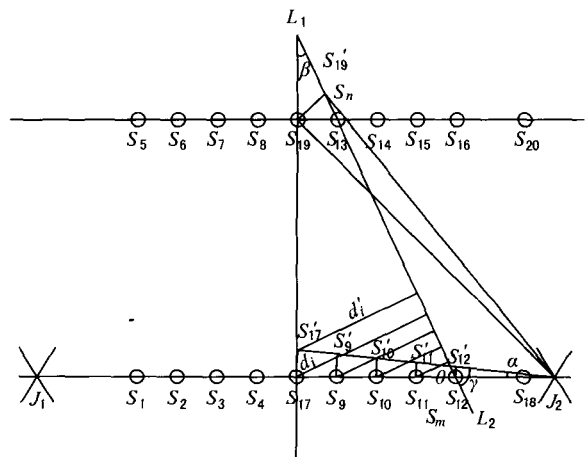


图6 状态5分析
Fig. 6 Analysis on posture 5

设 S_{19} 转 S'_{19} 到的旋转度数为 α , 则 S_{12} 及其左侧靠近轴心的传感器 S_{11} 、 S_{10} 、 S_9 和 S_{17} 转动的度数也为 α , 记转动后的位置分别为 S'_{12} 、 S'_{11} 、 S'_{10} 、 S'_9 、 S'_{17} .

不失一般性,记检测到引导线的前排传感器为 S_n , 后排传感器为 S_m , 设后排 S_m 左侧靠近底盘中心线的传感器为 S_i . S_i 转动前后与引导线的距离为 d_i 和 d'_i ($m > i$).

引导线 L_1L_2 与底盘中心线 $S_{17}S_{19}$ 的夹角为:

$$\beta = \arctan \frac{(m-n)s_0}{L}$$

转动前 S_i 到 L_1L_2 的距离为:

$$d_i = S_m S_i \cos\beta = (m - i) s_0 \cos\beta$$

而

$$\angle S_n S_m J_2 = \pi/2 + \beta$$

$J_2 S_{17}$ 与 L_1L_2 的夹角:

$$\gamma = \pi - \angle S_n S_m J_2 - \alpha = \pi/2 - \beta - \alpha$$

在 $\Delta q S_m J_2$ 中, 由

$$\frac{S_m J_2}{\sin\gamma} = \frac{q J_2}{\sin\angle S_n S_m J_2}$$

得到:

$$q J_2 = S_m J_2 \frac{\sin(\pi/2 + \beta)}{\sin(\pi/2 - \beta - \alpha)} = S_m J_2 \frac{\cos\beta}{\cos(\beta + \alpha)}$$

$$\begin{aligned} d'_i &= (J_2 S_i - q J_2) \sin\gamma \\ &= [J_2 S_i - S_m J_2 \frac{\cos\beta}{\cos(\beta + \alpha)}] \sin(\pi/2 - \beta - \alpha) \\ &= J_2 S_i \cos(\beta + \alpha) - J_2 S_m \cos\beta \end{aligned}$$

转动前后的差值为:

$$\begin{aligned} \Delta d &= d_i - d'_i \\ &= S_m S_i \cos\beta - [J_2 S_i \cos(\beta + \alpha) - J_2 S_m \cos\beta] \\ &= J_2 S_i \cos\beta - J_2 S_i \cos(\alpha + \beta) \\ &= J_2 S_i [\cos\beta - \cos(\alpha + \beta)] \end{aligned}$$

根据机器人的几何尺寸: $0 < \alpha < \pi/2$, $0 < \beta < \pi/2$, 所以 $\cos\beta - \cos(\alpha + \beta) > 0$.

可见调整前后, 底盘中心部分的传感器距离引导线的差值是大于 0 的. 对于这种情况, S_i 与引导线的距离不断缩小. 这样就会逐渐让 S_m 左侧的底盘中心部分传感器进入引导线区域, 底盘姿态进入图 5. 底盘的位姿会不断地调整, 并且靠近中心的传感器与引导线的距离不断缩小的过程会使得靠底盘中心线的传感器 S_{17} 逐渐靠近引导线, 最终结果会使底盘中心线和引导线重合.

根据以上分析, 底盘来回调整于各个状态之间, 最终会使驱动轮轴线上的中心传感器 S_{17} 和 S_{19} 逐渐靠近引导线.

5 MSP430 实现算法 (Algorithm using MSP430)

根据以上分析, 通过 TI 公司的 MSP430F449^[3] 单片机来实现, 具体实现算法如下.

记: 后排中心传感器 S_{17} 为 BACK, 前排中心传感器 S_{19} 为 FRONT; 后排最右侧传感器 S_{18} 为 BACK-COUNTER, 前排右侧传感器 S_{20} 为 FRONT-COUNTER; 这 4 个传感器接入 P1 口的 P14 ~ P17 中. 左侧传感器 $S_1, S_2, S_3, S_4, S_5, S_6, S_7$ 和 S_8 接 P3

口; 右侧传感器 $S_9, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{15}$ 和 S_{16} 接 P4 口. 对于放置于底部的颜色光电传感器, 若检测到深蓝色便向单片机输出高电平, 若检测到白色便向单片机输出低电平.

P3_HIGH = (P3IN&0xf0) >> 4; //将 P3 口的高四位值 (即 S_5, S_6, S_7, S_8) 取出存于一个寄存器的低四位中

P3_LOW = P3IN&0x0f; //将 P3 口的低四位值 (即 S_1, S_2, S_3, S_4) 取出存于一个寄存器的低四位中 //同理设置 P4 口

P4_HIGH = (P4IN&0xf0) >> 4;

P4_LOW = P4IN&0x0f;

由于传感器是检测到白色引导线为低, 如果 P3_HIGH! = 0x0f, 则说明 S_5, S_6, S_7, S_8 中必有一个检测到白色引导线. P3_LOW、P4_HIGH、P4_LOW 同理.

函数实现如下:

```
void XunXian_BackDrive(void) //后轮驱动巡迹
{
    if(((P1IN&FRONT)==0x00)&&((P1IN&BACK)==0x00))
        //传感器 BACK, FRONT 同时在白线内则直进
        Run_BackDrive();
    else if(((P1IN&BACK)==0x00)&&(P3_HIGH!=0x0f))
        //前排左侧的传感器有一个在白线上 &BACK 在白线上则左转
        TurnLeft_BackDrive();
    else if(((P1IN&BACK)==0x00)&&(P4_HIGH!=0x0f))
        //前排右侧中的传感器有一个在白线上 &BACK 在白线上则右转
        TurnRight_BackDrive();
    else if((P3_LOW!=0x0f)&&(P3_HIGH!=0x0f))
        //左侧的传感器 1,2,3,4 与 5,6,7,8 同在白线上, 则左转
        TurnLeft_BackDrive();
    else if((P4_LOW!=0x0f)&&(P4_HIGH!=0x0f))
        //右侧的传感器 9,10,11,12 与 13,14,15,16 同在白线上, 则右转
        TurnRight_BackDrive();
    else if
        (((P3_LOW!=0x0f)&&(P4_HIGH!=0x0f)) || ((P3_LOW!=0x0f)&&(P1IN&FRONT)==0x00))
        //1,2,3,4 与 13,14,15,16 同在白线上或者 1,2,3,4 在白线且 FRONT 在白线上, 则右转
        TurnRight_BackDrive();
    else if
        (((P4_LOW!=0x0f)&&(P3_HIGH!=0x0f)) || ((P4_LOW!=0x0f)&&(P1IN&FRONT)==0x00))
        //9,10,11,12 与 5,6,7,8 同在白线上或者 9,10,11,12 在白线且 FRONT 在白线上, 则左转
        TurnLeft_BackDrive();
    else
        Run_BackDrive();
}
```

其中,Run_BackDrive()是让后排两个驱动轮同时转动;TurnRight_BackDrive()是让后排左轮转动,右轮停止,使机器人向右移动;TurnLeft_BackDrive()是让后排右轮转动,左轮停止,使机器人向左移动。

假设驱动后排左右轮的电机转速均为 V ,当向右移动时(以 J_2 为旋转中心),如图 5、6 所示,底盘左侧沿引导线 L_1L_2 的速度为 $V\cos\beta$ 。在调整过程中,底盘有向前移动的速度分量,所以可以使机器人不断向前移动。

循环调用巡线函数就会得到很好的调整效果,由于是以 4 个传感器为一组获取底盘的姿态,因此即使在个别传感器失灵的情况下也可以很好地巡线。

6 改进方法(Improvement method)

由于在巡线过程中采用的是一个驱动轮动(引导线方向的速度为 $V\cos\beta$),另一个驱动轮停(除非底盘中心线和引导线重合时后排两轮才同时转动),使得机器人行进速度不快,并且由于两侧的速度差($V\cos\beta - 0$)比较大,在来回调整时整个车身都可能产生摆动。改进的方法是采用 PWM 信号使两个驱动轮的速度不同,从而产生速度差,速度慢的轮相对速度快的轮停止,这和一停一动的原理是一样的,并且机器人可以边调整边向前运动,通过这个速度差来调整机器人的位姿。

假设进入图 4(b)的状态,底盘应该向左移动以调整偏出量。置左轮速度为 V_1 ,右轮速度为 V_2 ($V_1 < V_2$)。底盘左侧向前移动的速度为 V_1 ,右侧速度为 $V_1 + (V_2 - V_1)\cos\beta$ 。这样可以让机器人在调整的过程中车身整体沿引导线的速度加快。左右轮的速度差设置比较小的值,避免速度差过大造成晃动,在该状态下的调整时间相对长一些。

7 交叉引导线处调整分析(Analysis of adjustment on the cross line)

机器人沿着引导线巡线的同时,其底盘中心线并不是与引导线重合或者平行的,为了使机器人在巡线中大致与引导线相平行,就需要在交叉引导线处调整底盘位置。基于后排传感器的调整方法如下:一直记录该排左右关于中心线对称的两个传感器状态,若左边的先突变(即检测到水平引导线),需让右轮动,左轮停止,使其右边部分也进入水平引导线便可停止在交叉线处的调整;若右边的先突变,需让左轮动,右轮停止,使其左边部分也进入水平引导线。基于前排传感器的调整方法也是如此。机器人在行

进过程中其实不必每次遇到交叉线就调整一次,这样可能会使整个车体产生抖动,因此这种调整方法只是在精确定位(如转弯)时使用,可使机器人在转弯的时候能够尽可能地转过 90° 弯。

8 转弯分析(Analysis on turning)

当计数满足所设定值时,机器人转弯进入另外一条交叉的引导线,此过程为机器人转弯 90° ,让其轴线尽量与引导线对齐,便于整个车身在这条引导线巡线。这就是前面的交叉引导线处的调整方法。

9 巡线失效分析(Analysis on line tracking failure)

两点确定一条直线,在巡线过程中前后排至少各有一个传感器检测到引导线,这样两个传感器便确定了机器人的位置。当前(或后)排的传感器全部偏离到白色引导线一侧时,最多只有一个传感器检测到引导线,这样就无法识别底盘是偏右还是偏左了,这样就会使巡线失败。某排传感器全部偏向一侧的原因,是在调整过程中由于左右轮的速度差过大,使得底盘整体偏离出了引导线。因此在调整时,应尽量避免后排驱动轮的速度差过大。

10 结论(Conclusion)

机器人根据周围的环境找到适合自身特点的参考点行进是个很复杂的问题。本文根据放置在机器人底盘适当位置的辨色传感器来检测各个放置点与路面引导线的相对位置关系,分优先级处理获得的位姿信息,并且论证了采用这种调整方法的可行性。考虑到速度差会产生相对运动,采用 PWM 来调整机器人姿态,在调整的同时可以快速地进行。该巡线机器人根据实际需要可改变自己的机构特征,由于定位非常准确可靠,很适合于工业控制中的物料搬运和定点工作的场合。

参考文献 (References)

- [1] 宗光华,等. 机器人的创意设计与实践[M]. 北京:北京航空航天大学出版社,2004.
- [2] Magrab E B,等. 马会生,等. MATLAB 原理与工程应用[M]. 北京:电子工业出版社,2002.
- [3] 沈建华,杨艳琴,翟晓曙. MSP430 系列超低功耗 16 位单片机原理与应用[M]. 北京:清华大学出版社,2004.

作者简介:

宋 波(1981-),男,硕士生. 研究领域:智能机电技术.

肖世德(1967-),男,教授,博士生导师. 研究领域:计算机集成制造系统.

华仕春(1982-),男,硕士生. 研究领域:智能机电技术.