

基于 Euler 算法的全向轮机器人位姿检测

许伦辉, 朱群强, 胡发焕

(江西理工大学 电气工程及自动化学院, 江西 赣州 341000)

摘要: 现代机器人多借助于参照物定位, 多受制于参照物; 对于此问题, 这里介绍了一种基于码盘和 Euler 算法定位的无参照物的全向轮机器人设计; 利用全向轮的结构优势, 设计了一款能实现在二维平面内无参照物移动定位的机器人; 此机器人通过采集 2 个相对编码器和一个绝对编码器数据来获取机器人位姿信息, 并接收上位机发送的坐标指令, 用 Euler 算法计算机器人运动坐标, 然后将机器人位姿信息与计算出来的坐标相对比, 对机器人进行姿态矫正, 最终达到给定坐标位置。

关键词: 相对编码器; 绝对编码器; 全向轮; Euler 算法

Detecting and Positioning Omni-directional Wheel Robot Pose Based on Euler Algorithm

Xu Lunhui, Zhu Qunqiang, Hu Fahuan

(Jiangxi University of Science and Technology, Ganzhou 341000, China)

Abstract: Modern robots are positioned mostly by means of references, which subject to references. For this problem, This paper introduces the design of the omni-directional wheel robot without reference based on the Euler algorithm and encoder positioning. Taking advantages of the distribution of the omni-directional wheel, we designed the robot that can rmobile and positioning without referencse in two-dimensional plane. The robot acquires data by two relatively encoder and an absolute encoder to get robot pose information. And it receives instructions that the upper machine sends and it uses Euler algorithm calculates robot motion coordinates. Then by comparing the calculated coordinates with robot pose information to correct the robot posture. Ultimately it achieves the goal that arrives at the position of the given coordinates.

Keywords: relative encoder; absolute encoder; omni-directional wheel; Euler algorithm

0 引言

随着中国经济的快速发展, 无论是工业领域, 还是农业领域对产品和生产自动化, 智能化的需求越来越迫切。因为产品和生产自动化, 智能化不仅可以提高生产效率, 而且也可以为企业节约大量人力, 物力。所以各国都在努力推动机器人产业发展, 由此推动了全球机器人产业迅猛发展, 尤其是中国, 更是一度成为全球机器人最大市场^[1-3]。

基于此背景下。本文介绍了一种通过计算自身相对坐标行走的全向轮定位机器人, 此机器人能在平场地按照一定轨迹行走。此机器人是大胆借鉴前人知识并通过实验验证的一种定位机器人, 采用 2 块 STM32F103 芯片作为姿态信息获取和控制驱动芯片, 通过 SPI 串口实现芯片通信获得机器人相对坐标, 以及在主芯片里面运用算法结合上位机发送过来的坐标数据计算出机器人轨迹相对坐标, 并在主芯片里面完成 2 个相对坐标比较, 输出正确的轨迹坐标, 然后将轨迹坐标分散到 3 个全向轮上面, 最后通过运动学公式将速度将按一定规律分配到每个轮子上, 驱动直流电机实现全向轮平稳运行^[4]。这种机器人可以直接用在工厂或其他地方搬运货物上, 相比传统的 AGV 小车搬运货物更不受环境变化约束。

1 机器人系统设计

机器人控制系统包括了 Master STM32F10 芯片控制系统和 SlaveSTM32F103 芯片控制系统, 主要实现这样的功能: Master STM32F103 芯片接收并分析解决上位机发送过来的坐标指令, 并结合 Euler 算法计算出机器人运动过程中的实时坐标 (x, y, θ) 另一方面, 机器人通过 SlaveSTM32F103 芯片采集码盘和绝对编码器数据建立机器人行走相对坐标 $(X, Y, \theta 1)$, 再通过 SPI 将相对坐标传送到 Master STM32F103 芯片, 然后将 2 个坐标相比较调整行走姿态, 最终实现定位。这里的坐标的 x 轴方向坐标需要进行分段处理, 通过上位机延迟不断发送。

2 机器人运动坐标系建立

SlaveSTM32F103 模块主要负责采集 2 个相对编码器和绝对编码器传过来的脉冲数目, 并通过圆弧算法实现机器人行走过程中测量机器人转过的角度和计算机器人行走距离, 坐标。在采集相对编码器脉冲数目的同时采集绝对编码器的各项的脉冲数目, 这样就能够保证机器人前后部分保持测量准确, 能够保证数据的同步更新。我们这里采用欧姆龙 E6B2-C 增量式编码器 (1200P/R) 作为相对编码器和 E6CP-AG5C (10 线制) 绝对编码器。10 线制的绝对编码器可以很精确地采集到轴角度的变化, 而且角度值固定只要选取其中一点作为初始角度值, 就可以实现正负角度值的测量。

欧姆龙 E6B2-C 相对编码器采集脉冲数一部分转换为机器人行走距离。具体转换如下: 1200P/R 相对编码器每一周

收稿日期: 2015-01-28; 修回日期: 2015-03-24。

基金项目: 国家自然科学基金 (61263024)。

作者简介: 许伦辉 (1965-), 男, 江西赣州人, 博士生导师, 主要从事智能交通系统、交通信息工程及控制方向的研究。

1200 个脉冲，测出相对编码器上的码盘直径为 D ，则码盘直径为 πD 。SlaveSTM32F103 定时采集脉冲数目累加值为 K ，则码盘随机器人转动行驶的距离为 $\frac{K\pi D}{1200}$ 。假设左右两边码盘行走的距离分别为 S_1 和 S_2 ，则有机器人行走的距离 $S = \frac{S_1 + S_2}{2}$ 。

另外，一部分转化为码盘行驶过程中转过的相对角度和行驶的相对坐标。具体算法如图 1 所示： L_1 和 L_2 分别为左右码盘在很短时间 t 内走过的路程，则在很短时间里码盘转过的角度 $\angle b \approx \angle a = \frac{L_1 - L_2}{L}$ 。这里的 $\angle a, \angle b$ 表示为弧度制， L 表示 2 个码盘中心之间的轴距离。在具体芯片里面需要将弧度值转换为角度，方便角度计算与程序调试修改参数用。则有： $\frac{180^\circ}{\pi} \approx 57.3^\circ, \angle b \approx \angle a = 57.3^\circ * (\frac{L_1 - L_2}{L})$ 。

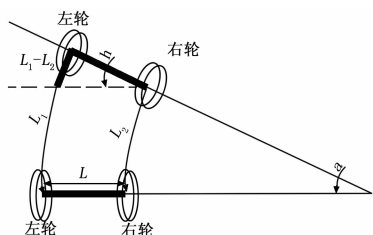


图 1 编码器组成的码盘分析图

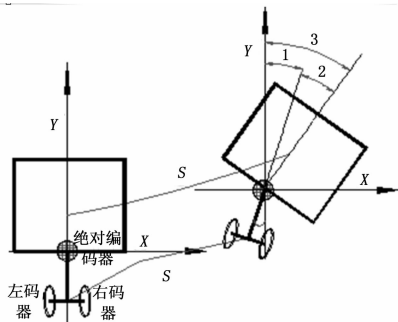


图 2 机器人小车行走路径图

由以上关系得出 2 个码盘中心相对坐标：

$$\nabla s = L_1 - L_2 \tag{1}$$

$$k = 57.3 \tag{2}$$

$$x_1 = x_0 + \nabla s * \cos \angle ka \tag{3}$$

$$y_1 = y_0 + \nabla s * \sin \angle kb \tag{4}$$

x_0, y_0 表示左右码盘上的编码器上次采集和计算的相对数字坐标值， (x_1, y_1) 表示后一次采集和计算得到的坐标，整个相对数字坐标值 (x, y) 实质上是通过 STM32F103 芯片中断程序采集和累加迭代更新计算得到的，而这里采集到的相对数字坐标是仅仅是 2 个码盘轴中心点的相对坐标。我们需要的是整个机器人车身的数字坐标，所以这里加入了绝对编码器 E6CP-AG5C 来实现。因此绝对编码器起器 E6CP-AG5C，起到了一个测量机器人车身相对 2 个码盘轴中心转过角度的作用。通过码盘转过角度和车身相对码盘轴中心转过角度的矢量和，实现整个机器人车身转过的角度的测量，从而进一步计算出机器人行走的相对坐标。如图 2 所示， $\angle 1 = \angle a$ 表示码盘行驶过程中相对转过的角度， $\angle 3 = \angle 1 \pm \angle 2$ 表示机器人整车相对转过的角度， $\angle 2$ 角表示机器人车身相对 2 个码盘轴中心

点转过的角度（通过绝对编码器测得）。

由以上关系得出机器人车身相对坐标：

$$k = 57.3 \tag{4}$$

$$X = X_0 + \nabla s * \cos(k * \angle 3) \tag{5}$$

$$Y = Y_0 + \nabla s * \sin(k * \angle 3) \tag{6}$$

X_0, Y_0 大小分别等于 x_0, y_0 ，具体由机器人车身起始绝对坐标决定。

整个机器人运动坐标系建立是以机器人运动起点为原点开始建立。通过相对编码器和绝对编码器测出机器人相对初始位置运动的距离 s 和机器人转过的角度 θ ，从而求出此刻机器人的实时坐标 (X, Y) 。进而可以知道机器人的行走的位姿 (X, Y, θ) 。机器人运动就相当于在以机器人运动起点为坐标原点建立的坐标系里面运动。

整个运动坐标的建立其实是运用机器人比赛里面常用的圆弧算法为基础建立起来的，在整车行走过程中都是采用同步采集数据，然后整车计算。调试过程采用 10 m 距离先直线反复调试取平均值，修改其中的矫正参数。然后，弯道测试，一般先不加驱动采用画好的弯道线推动小车速度从慢到快测量。弯道当中，坐标一般显示出来不是很稳定，而且与小车转过的角度有很大关系^[5]。所以需要加入另外算法才能实现小车坐标平稳变化^[6]。故此借助上位机，运用一定算法实现了坐标姿态检测与矫正。

3 基于 Euler 方法的坐标计算

Euler 方法（数值方法）常用于导弹追踪目标问题的解决。本文采用 Euler 方法进行追踪给定坐标，并在其中实时测量机器人小车坐标，实现机器人位姿坐标检测。Euler 方法原理比较简单，就是利用差商代替微商即：

$$\frac{dy}{dt} \approx \frac{\Delta y}{\Delta t} = \frac{y_{k+1} - y_k}{t_{k+1} - t_k}, \frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x_{k+1} - x_k}{t_{k+1} - t_k} \tag{8}$$

通常取 Δt 为常数 τ ，就得到由第 K 步的值到第 $K+1$ 步的关系式^[6]。

图 3 中，建立坐标系，小圆圈代表机器人，并以小圆圈运动来描述机器人的运动轨迹。 $N_4(x, y)$ 位置代表上位机发给机器人的坐标。在 N_4 坐标的同一直线上随机取点 N_0, N_1, N_2, N_3 这样就表示了给定坐标相当于从 N_0 点以一定速度 v 移动到 N_4 位置，这里上位机可以实现这样的功能：可以将定点坐标横坐标分成 N 份，并将每一份的坐标都求取出来。例如 $N_0(0, Y_0), N_1(X_1, Y_0), N_2(X_2, Y_0), N_3(X_3, Y_0) \dots$ 同时通过设定延时定时发送（这里由后面的 τ 决定）。这样就模拟了机器人追踪给定坐标的实际过程，因此可以建立起数学模型分析。设定 N_0 到 N_4 的时间为 t ，从 N_0 到 N_4 位置为速度某一范围值 V （这里取值 35 cm/s，结合机器人运动速度设定）。追踪机器人小车的速度为 V_t 。

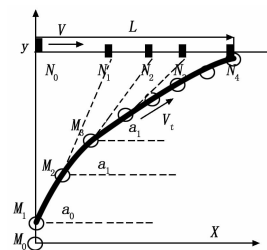


图 3 机器人定点坐标追踪图

当 $t=0$ 时机器人在 M_0 位置, 方向指向 M_1 方向。机器人运动到 M_1 位置 (x_1, y_1) , $x_1=0, y_1=Vt * \tau$, 这时给定的坐标位于 $N_1(0, h)$ 。此时机器人沿着 M_1N_1 方向运动, M_1N_1 的倾角 $\alpha_0 = \arctan \frac{h - V * \tau}{Vt * \tau}$ 。

当 $t = \tau$ 时, 机器人的位置为 $M2(x_2, y_2)$ 。其中:

$$y_2 = y_1 + Vt * \tau \sin(\alpha_0) \tag{9}$$

$$x_2 = x_1 + Vt * \tau \cos(\alpha_0) \tag{10}$$

$$\sin(\alpha_0) = \frac{h - V * \tau}{\sqrt{(Vt * \tau)^2 + (h - Vt * \tau)^2}} \tag{11}$$

$$\cos(\alpha_0) = \frac{Vt * \tau}{\sqrt{(Vt * \tau)^2 + (h - Vt * \tau)^2}} \tag{12}$$

当 $t = 2\tau$ 时, 此时给定坐标位于 $N_2(2V\tau, h)$, 机器人沿着 $M2N2$ 方向运动。以此类推, 当 $t = K\tau$ 时, 机器人位置为 $Mk(x_k, y_k)$ 机器人沿着 $MkNk$ 方向行走, 则 $MkNk$ 的倾角为 $\alpha_k = \arctan \frac{h - y_k}{kV * \tau - x_k}$, 从而进一步推到 $t = (k+1)\tau$, 机器人位置为 $M_{k+1}(x_{k+1}, y_{k+1})$, 其中:

$$y_{k+1} = y_k + Vt * \tau \sin(\alpha_k) \tag{13}$$

$$x_{k+1} = x_k + Vt * \tau \cos(\alpha_k) \tag{14}$$

$$\sin(\alpha_k) = \frac{h - y_k}{\sqrt{(kV\tau - x_k)^2 + (h - y_k)^2}} \tag{15}$$

$$\cos(\alpha_k) = \frac{kV\tau - x_k}{\sqrt{(kV\tau - x_k)^2 + (h - y_k)^2}} \tag{16}$$

给定坐标位置为 $NK((K+1)V * \tau, h)$ 。

在 MATLAB 里面建立模型分析, 迭代 $k+1$ 次, 使得 $y_k < h, y_{k+1} = h$ 。仿真停止。

在仿真过程中设给定坐标 $y=300$ cm 确定, x 坐标以速度 $V=35$ cm/s 沿着直线依次增大行走到达给定坐标 (250, 300)。假设机器人以 60 cm/s 的速度追踪给定坐标。追踪的过程中因为 $k\tau=t$, 所以追踪给定坐标由 k 和 τ 共同决定。然而在实际上追踪中取决于 τ 值, 因为等分值 τ 的大小决定了算法的迭代次数, 以及追踪的效果。在 $Vt=60$ cm/s 条件下, τ 值的不同追踪计算出的机器人坐标也不同。为了使实验仿真更加明显, 分别取下面 4 个不同的 τ 进行追踪仿真, 导出追踪的最后一组数据, 可以更加明显的看到机器人轨迹随参数 τ 的变化, 仿真结果如图 4~7 所示。

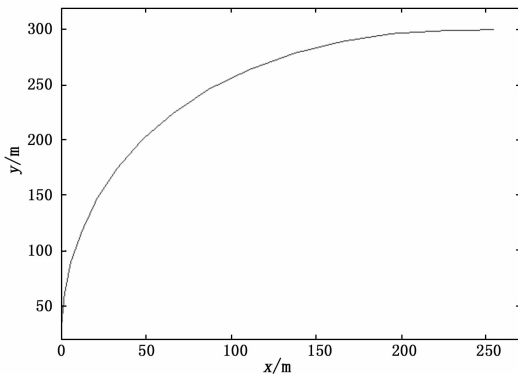


图 4 $\tau = 0.4$

从上面仿真的结果可以看出, 在 $y=300, x=250$ 包含的坐标系中, 在 $0.4 < \tau < 0.8$ 这段取值范围里面输出波形平稳, 坐标测量平稳, 没有出现较大的变化, 机器人沿着一定的轨迹

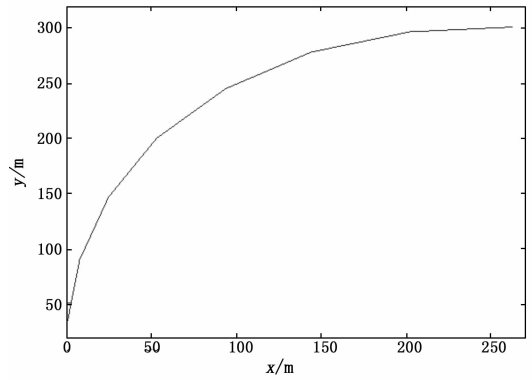


图 5 $\tau = 0.6$

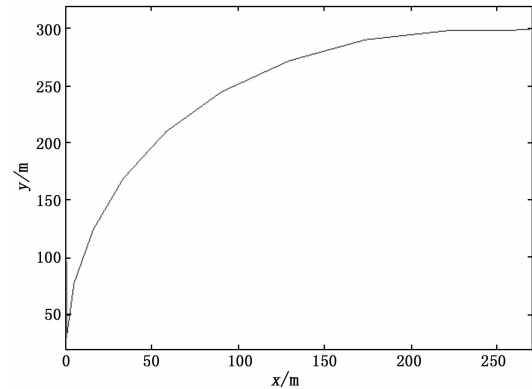


图 6 $\tau = 0.8$

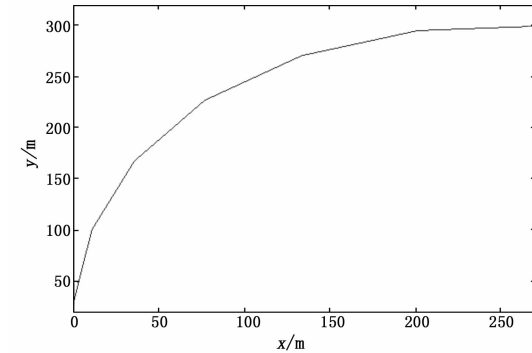


图 7 $\tau = 1.0$

行走, 说明在实际当中运用 Euler 算法计算机器人的实时运动坐标是可行的。并且机器人按照一定的轨迹行走, 可以在一定范围内有效地减少机器人无规律运动形成的误差。当 $\tau = 1.0$ 时, 波形出现抖动, 说明此时 Euler 算法计算出来的坐标已经开始出现误差, 不能平稳地计算出机器人运动的坐标, 所以在实际当中选取正确的 τ 是必须的, 基于 Euler 算法检测出来的机器人运动坐标信息, 也是有一定局限性的。但是在这里结合上面机器人自身测量出的坐标能起到一个很好的位姿坐标检测与矫正的效果。

4 实验与分析

这里全向机器人通过上位机发送指令给定机器人定位坐标, 然后启动机器人行走。一方面机器人通过自身运动测量自身行走坐标, 另一面通过 Euler 算法实现机器人坐标实时测

器故障与执行器故障, 利用 HB_o 技术抑制了系统干扰对故障估计的影响, 同时得到了连续系统中传感器故障与执行器故障的鲁棒渐近估计。本文所提故障诊断观测器不需要对故障、故障导数与系统干扰作上界已知假设, 与已有方法相比, 所提方法适用范围更广。最后通过仿真分析验证了所提方法的有效性。

参考文献:

[1] Witczak M. Fault Diagnosis and Fault-Tolerant Control Strategies for Nonlinear Systems [M]. Berlin: Springer, 2014.
 [2] Huang S, Tan K K, Lee T H. Fault diagnosis and fault-tolerant control in linear drives using the Kalman filter [J]. IEEE Transactions on Industrial Electronics, 2012, 59 (11): 4285 - 4292.
 [3] 许 研, 张 炜. 基于噪声滤波的非线性系统故障诊断方法研究 [J]. 计算机测量与控制, 2014, 22 (9): 2758 - 2759.
 [4] Ríos H, Davila J, Fridman L, et al. Fault detection and isolation for nonlinear systems via high-order-sliding-mode multiple observer [J]. International Journal of Robust and Nonlinear Control, in press.
 [5] Ahmadzadeh S, Zarei J, Karimi H R. Robust unknown input observer design for linear uncertain time delay systems with application to fault detection [J]. Asian Journal of Control, 2014, 16 (3): 1 - 14.

[6] Alwi H, Edwards C, Tan C P. Sliding mode estimation schemes for incipient sensor faults [J]. Automatica, 2009, 45 (7): 1679 - 1685.
 [7] Ichalal D, Marx B, Ragot J, et al. New fault tolerant control strategies for nonlinear Takagi-Sugeno systems [J]. International Journal of Applied Mathematics and Computer Science, 2012, 22 (1): 197 - 210.
 [8] Park T G. Estimation strategies for fault isolation of linear systems with disturbances [J]. IET Control Theory & Applications, 2010, 4 (12): 2781 - 2792.
 [9] Spong M W. Modeling and control of elastic joint robots [J]. Journal of dynamic systems, measurement, and control, 1987, 109 (4): 310 - 318.
 [10] Yan X G, Edwards C. Nonlinear robust fault reconstruction and estimation using a sliding mode observer [J]. Automatica, 2007, 43 (9): 1605 - 1614.
 [11] Jiang B, Staroswiecki M, Cocquempot V. Fault accommodation for a class of nonlinear systems [J]. IEEE Transactions on Automatic Control, 2006, 51 (9): 1578 - 1583.
 [12] 俞 立. 鲁棒控制: 线性矩阵不等式处理方法 [M]. 北京: 清华大学出版社, 2002.

(上接第 2297 页)

量, 通过与机器人自身坐标相比较实现位姿及时调整。整个系统启动流程图如图 8 所示。

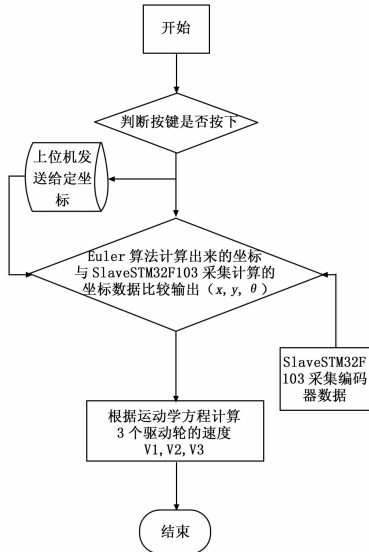


图 8 机器人启动流程图

实验环境选在 400 cm * 400 cm 的开阔场地进行, 上位机设定坐标 (250, 300), 机器人速度 $V_t = 60$ cm/s, 假设机器人给定坐标 $y = 300$ cm 不动, x 坐标沿着 X 轴正方向移动, 移动速度为 $V = 35$ cm/s, 中间移动的坐标可以表示为 $(35 k\tau, 300)$ 到达给定坐标 (250, 300)。因为之前类似仿实现过, 实验只需将机器人自身数字坐标与 Euler 算法计算出来的坐标实时比较, 并加入适量的增减参数, 调整机器人行走姿态, 最终整合程序一起调试, 就可以实现机器人位姿调整和定位。由于机器人追踪的速度不是很稳定, 这其中涉及到场地环境的影响, 还有机器人运动协调性, 动力输出等问题^[8]。所以在追踪速度上我们可以认为我同一速度 60 cm/s 运行。其次机器人的速度, 虽然在直线可以很好地调节, 但是在弯道或者地面摩擦

不够的情况下, 比较难去控制, 这里就在小车转弯的过程中适当加入参数量进行补充。实验中, 采取多次取值微调机器人速度尽量靠近真实值。试验经过多次调试, 基本上可以实现机器人位姿矫正和定位。关于修正误差还可以加入其它一些调节修正模块实现精确定位, 例如地盘 PID 反馈调节等^[9]。

5 结束语

通过对全向轮机器人运动分析, 以及将 Euler 算法运用到实验中, 达到了全向轮机器人定位的预期效果。在工业需求中可以推广到无参照物导航系统中去, 满足工业机器人自动搬运货物, 减少人力财力的消耗。

参考文献:

[1] 陶 敏, 陈 新, 孙振平. 移动机器人定位技术 [J]. 火力与指挥控制, 2010, 7 (7): 169 - 172.
 [2] Stephan K D, Michael K, Michael M G, Jacob L, Anesta EP. Social implications of technology: the past, the present, and the future. Proceedings of the IEEE, 2012, 100 (Special Centennial Issue): 1752 - 1781.
 [3] Sun T H, Tian W J, Wang H, et al. Kine-matic calibration of 3-DOF spindle head using double-ball bar [J]. Journal of Mechanical Engineering, 2012, 48 (5): 22 - 27.
 [4] 陈 伟, 何 伟. 基于 Bezier 曲线的机器人路径规划 [J]. 机器人技术与应用, 2009, 9 (5): 56 - 59.
 [5] 熊家新, 刘 丽, 徐 鹤, 等. 全向轮式移动机器人的定位导航算法 [J]. 长春理工大学学报, 2014, 8 (4): 83 - 87.
 [6] 江 杰, 李 强. 嵌入式机器人视觉定位及地图构建系统设计 [J]. 计算机测量与控制, 2012, 20 (9): 2497 - 2500.
 [7] 向隆万, 乐经良. 导弹跟踪问题 [J]. 高等数学研, 1999, 4: 35 - 44.
 [8] 笞 杰, 蔡宗琰, 梁 虎, 等. 基于 Bezier 曲线的自主移动机器人最优路径规划 [J]. 兰州大学学报, 2013, 4 (2): 249 - 254.
 [9] 刘祚时 李向誉, 胡发焕, 等. 基于 Bezier 曲线的全向机器人运动控制系统 [J]. 制造业自动化, 2014, 7 (7).