

智能乒乓球拾取机器人的设计与实现

邱高磊, 李红双, 姜春英, 潘五九

(沈阳航空航天大学 机电工程学院, 辽宁 沈阳 110136)

摘要: 针对在乒乓球运动中人工频繁捡球的问题, 文中设计了一款智能乒乓球拾取机器人。主要介绍了该机器人的整体结构设计和控制算法设计, 并通过实验验证可行性。首先从整体介绍了该机器人的工作流程和三个主要系统的设计, 包括收球系统、存球系统、发球系统, 其中收球系统通过旋转刷式收球机制和摩擦轮实现; 存球系统通过“人”字形存球轨道实现; 发球系统通过送球托杆和摩擦轮实现。接着讨论了目标跟随算法、数据融合算法、路径规划算法的设计, 其中目标跟随算法分别建立摄像头画面模型和全局定位系统模型, 并通过基于全连接神经网络的非线性 PID 算法实现机器人的运动控制, 神经网络通过遗传算法优化参数; 数据融合算法通过经典的卡尔曼滤波融合北斗定位模块、编码器以及 IMU 的数据得到更精确的机器人的位置和速度信息; 对于路径规划算法, 通过实验证明贪心算法、回溯算法、动态规划、分支限界中动态规划是最优秀的, 然后再进行实验证明遗传算法优于动态规划。最后, 在大约 $10\text{m} \times 10\text{m}$ 的场地中由人工随机放置 16 个乒乓球, 由机器人以 8 个为一组进行收取, 重复实验 10 次, 结果表明, 在每次实验中, 机器人均能够成功完成收球任务, 且 10 次实验的平均速度与规定的机器人运动最大速度的比值为 0.82491, 在能够完成工作的基础上, 机器人还具有较高的工作效率。

关键词: 乒乓球拾取机器人; 目标跟随; 非线性 PID 算法; 数据融合; 路径规划

中图分类号: TP242 **文献标识码:** **文章编号:**

Design and implementation of an intelligent table tennis ball picking robot

QIU Gaolei, LI Hongshuang, JIANG Chunying, PAN Wujia

(College of Electromechanical Engineering, Shenyang Aerospace University, Shenyang 110136)

Abstract: To address the issue of frequent manual ball picking in table tennis, an intelligent table tennis ball picking robot is designed. The overall structural design and control algorithm design of the robot are mainly introduced, and the feasibility is verified through experiments. Firstly, the working process of the robot and the design of the three main systems, including the ball collecting system, the ball storing system, and the ball launching system, are introduced. The ball collecting system is realized through a rotating brush ball collecting mechanism and friction wheels; the ball storing system is realized through a “人”-shaped ball storing track; the ball launching system is realized through a ball feeding rod and friction wheels. Then, the design of the target following algorithm, data fusion algorithm, and path planning algorithm is discussed. The target following algorithm establishes the camera image model and the global positioning system model, and the robot's motion control is realized through a nonlinear PID algorithm based on a fully connected neural network, where the neural network parameters are optimized by a genetic algorithm. The data fusion algorithm fuses the data from the Beidou positioning module, encoder, and IMU through the classical Kalman filter to obtain more accurate position and speed information of the robot. For the path planning algorithm, experiments first prove that dynamic programming is the best among the greedy algorithm, backtracking algorithm, dynamic programming, and branch and bound, and then further experiments prove that the genetic algorithm is superior to dynamic programming.

Finally, in a field of approximately 10m×10m, 16 table tennis balls are randomly placed by hand, and the robot collects them in groups of 8. The experiment is repeated 10 times, and the results show that the robot successfully completes the ball collecting task in each experiment, and the average speed of the 10 experiments is 0.82491 times the maximum specified robot movement speed, indicating that the robot not only completes the task but also has high work efficiency.

Key words: Table tennis ball picking robot; Target following; Nonlinear PID algorithm; Data fusion; Path planning.

随着机器人技术的发展, 机器人技术在体育领域的应用层出不穷, 国内外在捡球机器人方面颇有研究^[1-5]。Yeon H S^[6]等设计了自主式乒乓球收集机器人, 其采用了斑点检测和级联分类器的组合, 实现对球体的检测, 并使用 A* 图搜索算法生成最佳路径, 最后通过吸力马达和一个2自由度的主动喷嘴来收集检测到的球; Sato^[7]等使用旋转弹性叶片的独特收球机制, 并使用深度学习算法检测球, 使机器人能够自主导航到最近的球并将其收集起来; 张勇^[8]等通过进球叶轮实现收球, 并设计了存球系统, 可以临时存储收取的球体, 并结合了多种技术, 实现建图定位和路径规划, 高效完成收球任务; Nicolaus^[9]等设计了一款搭载两个摄像头的机器人, 分别用于确定球体的位置和机械臂的反馈控制能够实现精准收球; Wei^[10]等设计了一种利用机械臂夹球和流体吸球功能的乒乓球拾球装置, 可实现两种不同拾球机构的切换, 应用场景更加广泛, 实用性更强。

尽管与捡球机器人相关的研究有很多, 但应用性并不高, 原因主要在于捡球效率存在瓶颈, 主要表现在以下两个方面:

(1) 在捡球方式方面, 类似于机械臂形式的捡球方式在效率上存在挑战。这主要是因为机械臂的动作需要精确的控制来适应不同的球体位置和环境。机械臂捡球效率与人工捡球相比, 优势并不明显。

(2) 在运动控制方面, 现有的捡球机器人设计方案通常集中于机器人身边的球体, 而未充分考虑整个场地内球体的分布情况。这种局限可能导致机

器人在执行任务时, 未能采用较短的路径, 从而增加了不必要的移动时间。

针对以上问题, 本文设计了一款同时具备收球、存球、发球功能的机器人, 其中收球功能通过旋转刷式捡球机制实现, 原理为高速旋转的胶圈轮与球体接触的过程中, 在摩擦力的作用下球体能够加速并运动到存球系统中, 而发球系统的功能是在存球系统中球体数量达到容量上限后, 将球体发射出去, 达到清空存球系统的目的。同时搭载双目相机、IMU、北斗定位模块, 用于计算球场中球体相对于机器人的坐标, 并基于此, 通过遗传算法确定收球顺序。

1 整体结构设计(Overall structural design)

机器人的任务是自动捡取散落在场地上的球体, 并在存球系统达到容量上限时, 到指定位置清空存球系统。为了提高工作效率, 机器人在执行任务时应最大限度地减少时间消耗。整个系统被划分为以下六个关键部分: 收球系统、存球系统、发球系统、控制系统、能源系统、驱动系统, 下面先简述整机工作流程, 再介绍前3个系统的设计。

1.1 整机工作流程

机器人整体结构如图1所示, 在收球的过程中, 收球胶圈轮不断旋转, 球体与之接触后会产生较大的速度, 以此通过上升通道, 进入存球轨道, 在重力的作用下, 球体沿着轨道运动至最低端, 当存球轨道中球体数量达到上限时, 通过发球系统清空存球轨道, 然后继续收球。从开机到关机的完整工作流程如图2所示。



图1 机器人整体结构图

Fig.1 Robot overall structure diagram

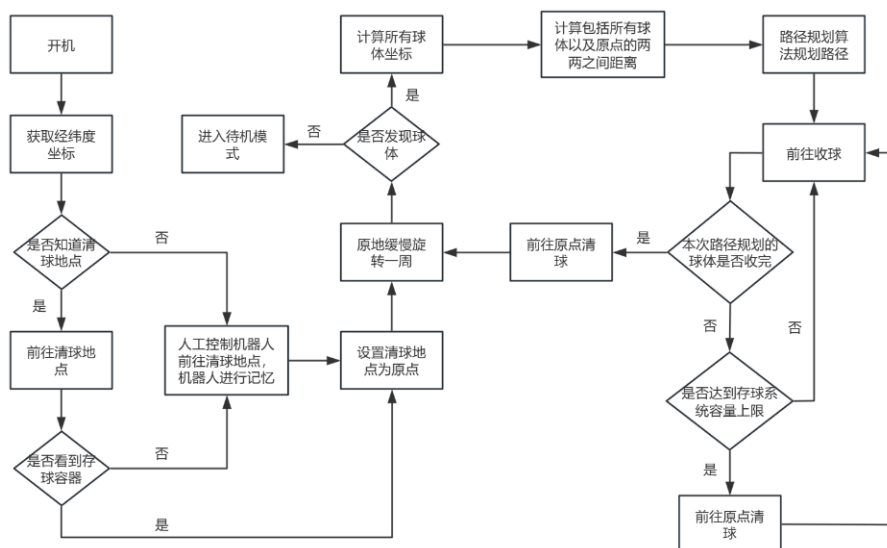
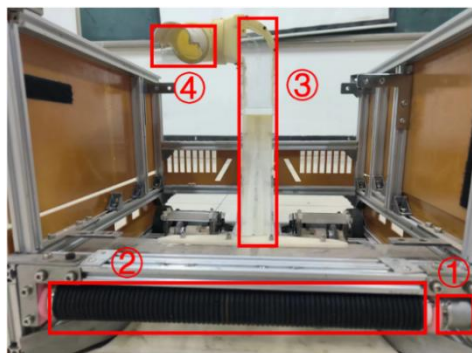


图 2 控制系统工作流程图

Fig.2 Control system workflow diagram

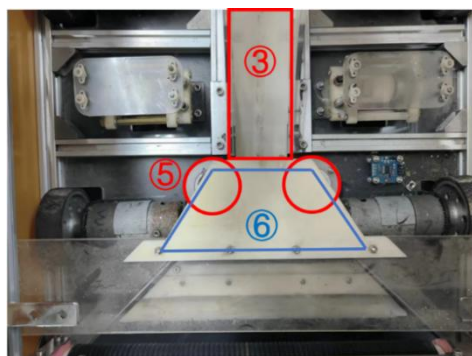
1.2 收球系统设计

收球系统的核心任务是通过机械装置将场地上的球体收集到机器人内部。该系统由收球胶圈轮、上升通道、收球机构出口、升球摩擦轮和挡球板组成，如图3所示。



①收球电机 ②收球胶圈轮
③上升通道 ④收球机构出口

(a) 收球系统正视图



⑤升球摩擦轮 ⑥挡球板

(b) 收球系统俯视图

图3 收球系统结构图

Fig.3 Ball receiving system structure diagram

收球胶圈轮在功能和原理上与摩擦轮相似，即通过旋转使其与物体之间存在相对运动，从而在接触时产生滑动摩擦力，物体受该摩擦力的作用而产生加速度，进而加速运动。收球胶圈轮的作用是通过上述原理将场地上的球体送进通道中，在通道中由升球摩擦轮进行加速。由此可知，收球胶圈轮的速度并不需要很快，选用直流减速电机驱动其转动。考虑到升球摩擦轮与地面有一定距离，故在收球胶圈轮下方安装一个倾斜向上的收球铲。在收球铲的作用下，球体能够和升球摩擦轮在同一高度处，使其能够获得良好的加速效果。为了避免球体在收球胶圈轮的作用下速度过快而直接飞到上升通道中，在收球铲末端和升球摩擦轮之间水平放置了挡球板，限制球体的高度。通过收球胶圈轮、收球铲和挡球板，球体能够顺利地与升球摩擦轮接触，并获得足够的速度通过上升通道和收球机构出口，最后进入存球轨道。

1.3 存球系统设计

存球系统的任务是临时存储收球系统收取的球体。该系统由转盘、转盘电机和存球轨道组成，如图4所示。

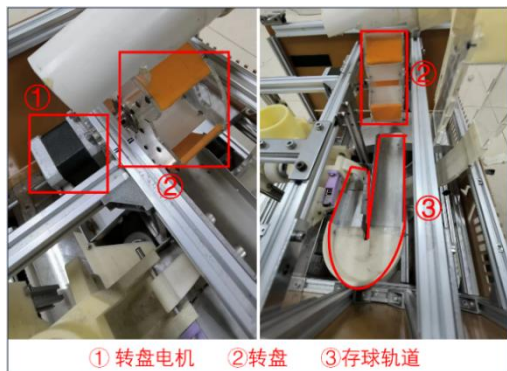


图4 存球系统结构图

Fig.4 Ball storage system structure diagram

转盘设计有若干凹槽，用于接收从收球机构出口滚出的球体。当球体到达转盘上方，如果遇到挡板，球体将暂时停留。在机器人工作时，转盘电机始终驱动转盘旋转，在凹槽对准球体时，球体便落入凹槽中。当转盘继续旋转至一定角度后，球体将滚入“人”字形的存球轨道。轨道整体向下倾斜，使得球体在重力作用下沿轨道下滑，直至到达发球系统入口或被前方的球体阻挡。

在存球轨道始端和末端分别放置有激光测距模块，其作用分别是检测轨道中球体数量是否达到容量上限和检测轨道中是否有剩余球体。

1.4 发球系统设计

当存球系统的容量达到上限后，发球系统需要清空存球系统，使存球系统为后续工作提供足够的容量，结构如图5所示，主要由送球托杆、托杆电机、发球摩擦轮和发球通道组成。当存球系统中的球体数量达到容量上限后，托杆电机将驱动送球托杆不断旋转，带动存球轨道最下端的球体运动，球体沿着发球轨道上升，当接触到高速旋转的升球摩擦轮后，将从发射口发射出去，如图6所示。当存球系统中的球体被清空后，发球系统将停止工作，机器人继续执行捡球任务。

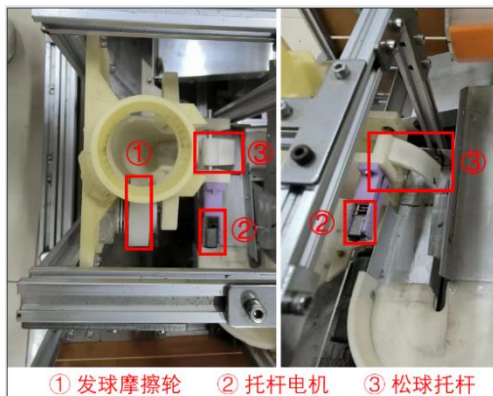


图5 发球系统结构示意图 (1)

Fig.5 Schematic diagram of the serving system structure (1)

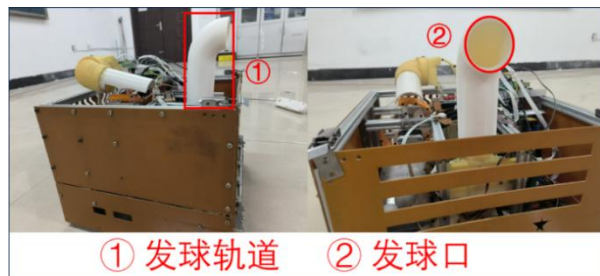


图6 发球系统结构示意图 (2)

Fig.6 Schematic diagram of the serving system structure (2)

2 算法设计 (Algorithm Design)

本文设计的机器人搭载全局定位系统，其本质就是固结于地面的参考系，以清球地点为原点，以沿地磁场方向和垂直于地磁场方向为坐标轴的正方向，通过该参考系来描述机器人的位姿和球体的绝对坐标，进而计算球体相对于机器人的相对坐标。

在机器人的运动中，目标跟随算法发挥着关键作用，它根据摄像头捕获的实时画面信息，或根据全局定位系统中机器人与球体的绝对坐标计算出的球体相对于机器人坐标系的相对坐标，动态调整左右车轮的转速，以确保机器人能够快速修正速度方向，从而提高捡取球体的精度和效率。

此外，为了有效进行全局定位并优化路径规划，机器人需要准确知晓自身的位置和速度。因此，使用数据融合算法，将来自多个传感器的数据集成，以获得更精确的位置和速度信息。同时，机器人需要能够进行合适的路径规划，以减少不必要的时间开销。

2.1 目标跟随算法

目标跟随算法的任务是在机器人靠近目标球体的过程中，不断调整车轮转速以修正机器人运动方向，提高收球准确率和效率。算法的输入为球体在机器人坐标系中的方位，输出为机器人两侧车轮速度差。相对方位可以通过摄像头画面信息和全局定位系统两个方式计算得到，下面建立相关的数学模型。

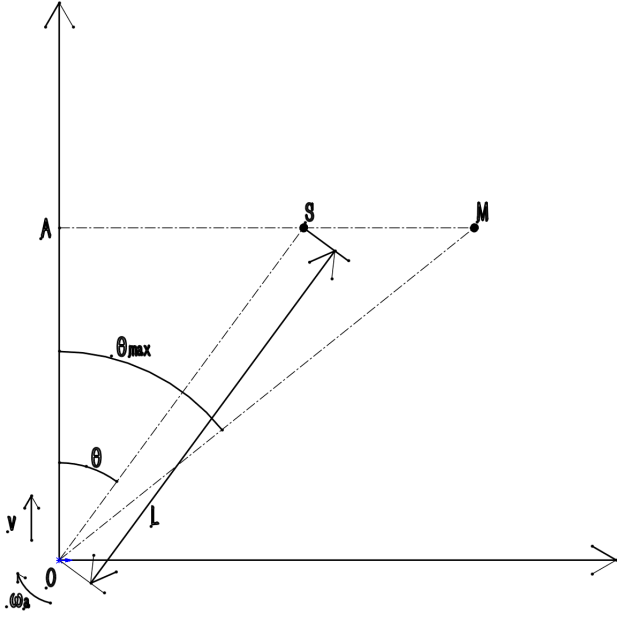


图7 机器人运动模型

Fig.7 Robot motion model

2.1.1 摄像头画面信息模型

如图7所示，坐标系固结于机器人，设机器人位于点O，其形心相对于地面的速度大小为 v ，角速度大小为 ω_a ，目标地点为点S，距离为 $|OS| = L$ ，与速度方向的夹角为 θ ，假设相机能够捕捉的最大范围为 θ_{max} 。相机画面模型如图8所示，矩形框为画面边界，以画面中心为原点，以一个像素作为单位长度建立坐标系，设点S与中轴线（即Y轴）的距离为 x ，画面最大距离为 x_{max} 。

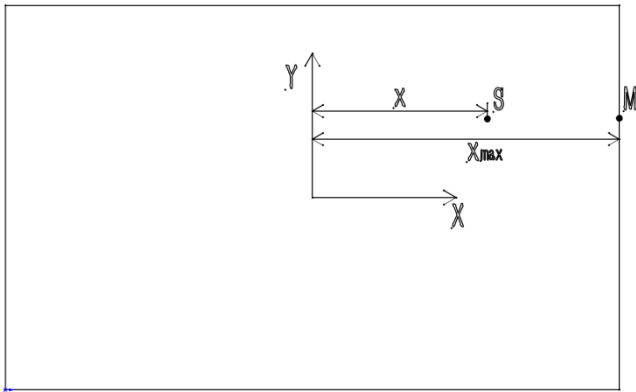


图8 相机画面模型

Fig.8 Camera screen model

根据图7的模型易知

$$\begin{cases} |AS| = |OA|\tan\theta \\ |AM| = |OA|\tan\theta_{max} \end{cases} \quad (1)$$

运动模型和画面模型通常有以下关系：

$$\frac{|AS|}{|AM|} = \frac{x}{x_{max}} \quad (2)$$

联立式(1)(2)，得

$$x = \frac{x_{max}}{\tan\theta_{max}} \tan\theta \quad (3)$$

等式两边同时对时间求导得

$$\dot{x} = \frac{x_{max}}{\tan\theta_{max}} \sec^2\theta \cdot \dot{\theta} \quad (4)$$

其中 $\dot{\theta}$ 表示点S相对于机器人的角速度，由机器人相对于地面的平动速度和旋转速度共同决定。设机器人前进时平动速度为正，顺时针旋转时旋转速度为正，以图7所示情况为例，由式(3)可知，当前进时， θ 增大， x 增大；当顺时针旋转时， θ 减小， x 减小。设 ω_t 和 ω_r 分别为由于平动和旋转产生的点S相对于机器人的角速度分量，显然

$$\dot{\theta} = \omega_t + \omega_r \quad (5)$$

而根据上述分析和图7模型可知

$$\begin{cases} \omega_t = \frac{v\sin\theta}{L} \\ \omega_r = -\omega_a \end{cases} \quad (6)$$

故

$$\dot{\theta} = \frac{v\sin\theta}{L} - \omega_a \quad (7)$$

若设 v_l 和 v_r 分别为左右两侧车轮转速， $2\Delta v$ 为转速差， h 为两侧轮距，则满足

$$\begin{cases} v_l = v + \Delta v \\ v_r = v - \Delta v \\ \omega_a = \frac{2\Delta v}{h} \end{cases} \quad (8)$$

为避免机器人相对于地面角速度过大而导致所需的向心力过大，考虑使机器人的平动速度随角速度增大而减小，定义如下：

$$v = v_0 - p|\Delta v| \quad (9)$$

其中 v_0 为基础速度， p 为无量纲的常数，综上所述

$$x = \frac{x_{max}}{\tan\theta_{max}} \tan\theta \quad (10)$$

$$\dot{x} = \frac{x_{max}}{\tan\theta_{max}} \sec^2\theta \left(\frac{(v_0 - p|\Delta v|)\sin\theta}{L} - \frac{2\Delta v}{h} \right) \quad (11)$$

$$\dot{L} = (v_0 - p|\Delta v|)\sin\theta \quad (12)$$

$$\dot{\theta} = \frac{(v_0 - p|\Delta v|)\sin\theta}{L} - \frac{2\Delta v}{h} \quad (13)$$

2.1.2 全局定位系统模型

本文所指全局定位系统可理解为只定位而不建图的SLAM（同步定位与建图）。SLAM是一类强大而实用的算法，在多个领域都是研究和应用^[11-13]。但完整的SLAM对算力要求较高，且该机器人通常

在单一而简单的环境中工作，建图的意义并非很大，故该机器人只有定位功能。

如图9所示，坐标系 O 为世界坐标系，坐标系 R 为机器人坐标系，点 S 为目标点，各个变量如图9所示，根据机器人运动学，从坐标系 O 到坐标系 R 的齐次变换矩阵为

$${}^O_R T = \begin{bmatrix} \cos {}^O\theta_R & -\sin {}^O\theta_R & {}^Ox_R \\ \sin {}^O\theta_R & \cos {}^O\theta_R & {}^Oy_R \\ 0 & 0 & 1 \end{bmatrix}$$

由图9可知，目标点 S 在坐标系 O 中的坐标为

$${}^OP_S = \begin{bmatrix} {}^Ox_S \\ {}^Oy_S \\ 1 \end{bmatrix}$$

则点 S 在机器人坐标系中的坐标为

$$\begin{aligned} {}^RP_S &= ({}^O_R T)^{-1} {}^OP_S \\ &= \begin{bmatrix} \cos {}^O\theta_R & -\sin {}^O\theta_R & {}^Ox_R \\ \sin {}^O\theta_R & \cos {}^O\theta_R & {}^Oy_R \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} {}^Ox_S \\ {}^Oy_S \\ 1 \end{bmatrix} = \begin{bmatrix} {}^Rx_S \\ {}^Ry_S \\ 1 \end{bmatrix} \end{aligned}$$

整理得

$$\begin{bmatrix} {}^Rx_S \\ {}^Ry_S \end{bmatrix} = A \begin{bmatrix} {}^Ox_S & {}^Oy_S & {}^Ox_R & {}^Oy_R \end{bmatrix}^T \quad (14)$$

其中

$$A = \begin{bmatrix} \cos {}^O\theta_R & \sin {}^O\theta_R & -\cos {}^O\theta_R & -\sin {}^O\theta_R \\ -\sin {}^O\theta_R & \cos {}^O\theta_R & \sin {}^O\theta_R & -\cos {}^O\theta_R \end{bmatrix}$$

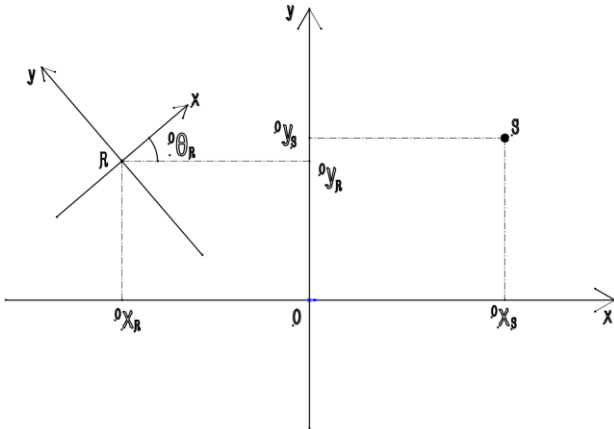


图9 全局定位系统世界坐标系示意图

Fig.9 Schematic diagram of the world coordinate system of the global positioning system

上述基于世界坐标系的模型能够将点 S 在世界坐标系中的坐标映射到机器人坐标系中，下面建立基于机器人坐标系的模型，能够根据点 S 的坐标计算方位。

如图10所示，规定当 ${}^Rx_S \neq 0$ 时， ${}^R\theta_S \cdot {}^Rx_S > 0$ ；

当 ${}^Ry_S < 0$ 时， $|{}^R\theta_S| > \frac{\pi}{2}$ ；当 ${}^Ry_S < 0$ 且 ${}^Rx_S = 0$ 时，

${}^R\theta_S = \pi$ 。基于此，可以得到 ${}^R\theta_S$ 的计算公式为

$${}^R\theta_S = \begin{cases} \theta_1({}^Rx_S, {}^Ry_S), {}^Ry_S > 0 \\ \theta_2({}^Rx_S, {}^Ry_S), {}^Ry_S < 0 \text{ and } {}^Rx_S \neq 0 \\ \theta_3({}^Rx_S, {}^Ry_S), {}^Ry_S < 0 \text{ and } {}^Rx_S = 0 \\ \theta_4({}^Rx_S, {}^Ry_S), {}^Ry_S = 0 \end{cases} \quad (15)$$

其中

$$\theta_1(x, y) = \arctan \frac{y}{x}$$

$$\theta_2(x, y) = \left(\pi + \arctan \frac{y}{x} \right) \text{sign}(x)$$

$$\theta_3(x, y) = \pi$$

$$\theta_4(x, y) = \frac{\pi}{2} \text{sign}(x)$$

在全局定位系统的模型中，机器人的运动状态对点 S 在机器人坐标系中的坐标作用效果与式(12)(13)同理。

根据以上模型， x_{\max} 、 θ_{\max} 和 h 是机器人本身的参数，而 v_0 和 p 是超参数，理论上，这些参数确定之后，系统就是确定的，尽管存在噪声，但由于机器人采用的是旋转刷式收球机制，不需要非常精确，故可以先在仿真环境中确定好具体的算法，再移植到控制系统中。

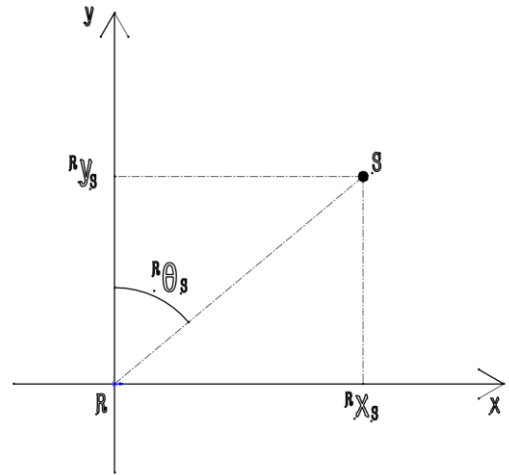


图10 全局定位系统机器人坐标系示意图

Fig.10 Schematic diagram of the global positioning system robot coordinate system

2.1.3 算法设计

根据模型，显然这是非线性时不变系统，可以考虑非线性PID算法。相较于传统的PID算法，非线性PID算法可以根据误差，通过非线性函数来调整比例、积分和微分三者的参数^[14-16]，如式(16)所示。

$$u(k) = K_p(e(k)) \cdot e(k) + K_i(e(k)) \cdot T \sum_{i=0}^k e(i) + K_d(e(k)) \cdot \frac{(e(k) - e(k-1))}{T} \quad (16)$$

神经网络具有强大的拟合能力，而遗传算法具有强大的搜索能力，本文通过简单的全连接神经网络来计算PID参数，并通过遗传算法来优化神经网络参数。定义适应度函数如下

$$Fitness = \begin{cases} \frac{S}{v_0 t}, & \text{successful} \\ \frac{S_{flag}}{L} - 1, & \text{unsuccessful} \end{cases} \quad (17)$$

其中 S 为初始距离， t 为机器人实际运行时间， v_0 为机器人基础速度， L 为结束时机器人与目标点的距离，当在一定时间内机器人与目标点的距离小于一定范围时，即 S_{flag} ，则定义为“成功”，否则为“失败”。

机器人的速度、角速度和角加速度有范围限制，当接收到非线性PID计算的原始角速度后，会进一步计算最终角速度和速度，伪代码如下所示

算法 1 计算线速度 v 和角速度 w

Input: 在第 k 次控制时，由非线性PID算法计算得到的原始的角速度
Output: 在第 k 次控制时，根据原始的角速度计算得到的最终的角速度和线速度

```

1: a_max := 1.9           //最大加速度的大小
2: T := 0.005             //采样周期
3: p := 1                 //线速度衰减系数
4: h := 0.3               //两侧车轮间距
5: v_0 := 0.5             //基础线速度
6: w_max := v_0 / p / h   //最大角速度
7:
8: a := (current_w - last_w) / T
9: if abs(a) >= a_max then
10:    current_w := last_w + a_max * T * sign(a)
11: end if
12: if abs(current_w) >= w_max then
13:    current_w := w_max * sign(current_w)
14: end if
15: current_v := v_0 - abs(p * current_w * h)
16:
17: return current_w, current_v

```

全连接神经网络和遗传算法的参数设置如表 1 所示，其中神经网络的输入层考虑了上一次 PID 控制器的输出。经过 100 次迭代，最优个体适应度达到**0.9904**，优化过程如图 11 所示。对该个体进行**100**次的随机测试（对于单次测试，当误差小

于**0.5 cm**时停止本次测试），平均误差为**0.004308241m**，最大的**10**次误差如表 2 所示，随机选择**1**次模拟的结果，包含路径、角度与时间、距离与时间、速度与时间、角速度与时间的关系，如图 12 所示。

表 1 非线性 PID 的神经网络和遗传算法的参数设置

Tab.1 Parameter Setting of Neural Network and Genetic Algorithm for Nonlinear PID

项目	内容
每层神经元数量	(2, 4, 5, 3)
种群中个体数	200
迭代次数	100
选择算子	规模为 3 的锦标赛算子
交叉算子	模拟二进制算子
突变算子	高斯突变

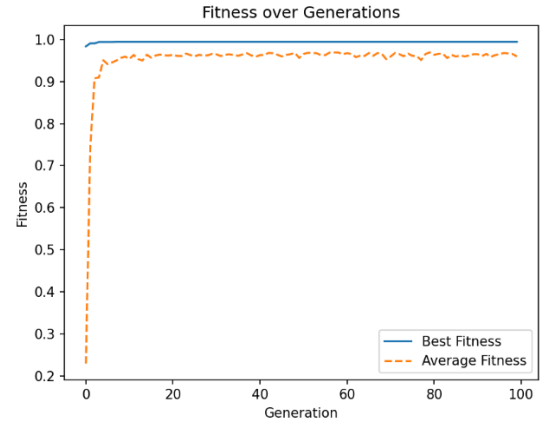


图 11 神经网络优化过程

Fig.11 Neural network optimization process

表 2 最优个体在随机测试中最劣 10 次表现

Tab.2 The worst 10 performances of the best individual in random tests

序号	误差/ m	序号	误差/ m
1	0.00498744	6	0.00499400
2	0.00499038	7	0.00499519
3	0.00499195	8	0.00499535
4	0.00499221	9	0.00499846
5	0.00499378	10	0.00499859

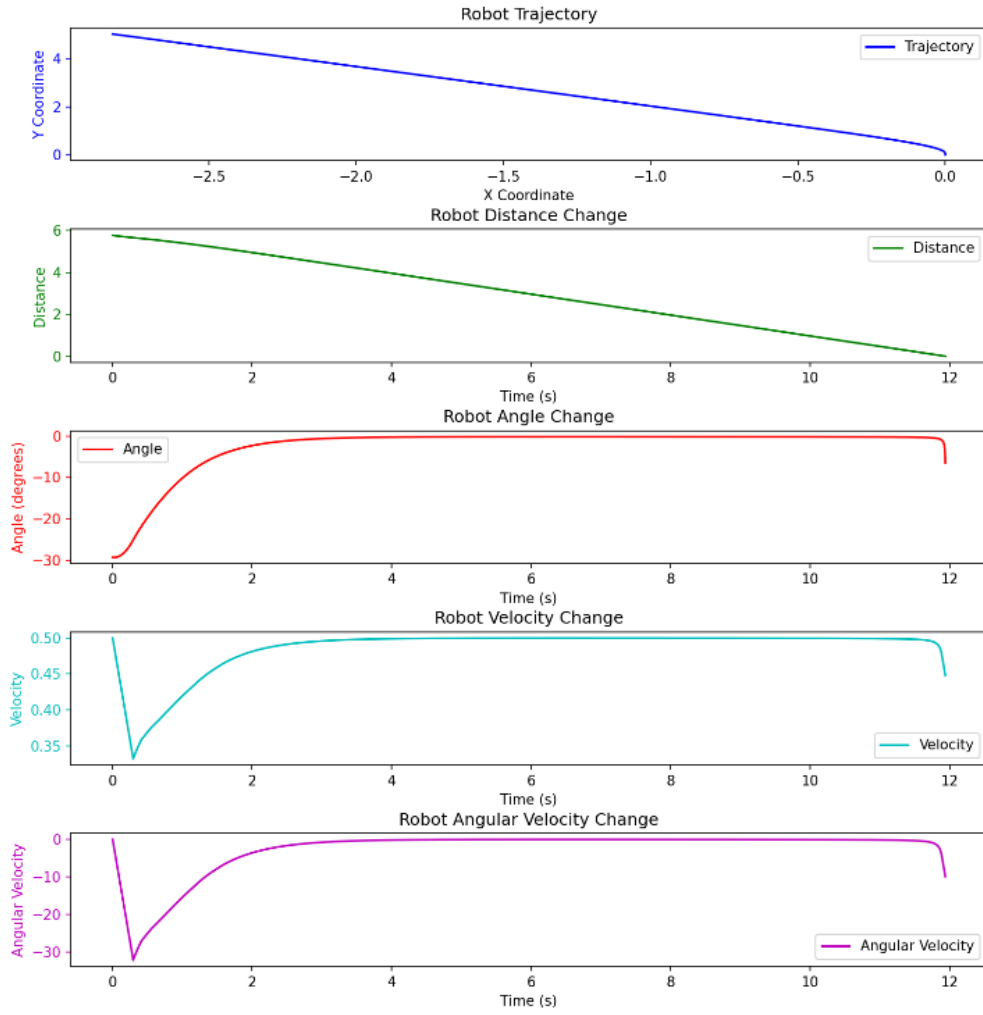


图 12 模拟结果示意图

Fig.12 Schematic diagram of simulation results

2.2 数据融合算法

在机器人中，需要知道机器人的空间位置、运动速度和运动方向，其搭载有IMU，北斗定位模块和编码器。根据文献[17]，若已知地球上两点A和B的经纬度分别为 (x_1, y_1) 和 (x_2, y_2) ，其中 x 为纬度， y 为经度， R 为地球半径，则两点距离 Δ 满足

$$\Delta = 2R \cdot \arcsin\left(\frac{\sqrt{2m}}{2}\right) \quad (18)$$

其中

$$f(x, y) = \cos(x - y)$$

$$g(x, y) = \cos x \cdot \cos y$$

$$m = 1 - f(x_1, x_2) + g(x_1, x_2) \cdot (1 - f(y_2, y_1))$$

通常机器人的工作范围相对于地球而言非常微小，尽管当机器人在不同纬线上沿纬线运动时，其经度变化量相同时位移不同，但该差别可以忽略不计，因此，如果物体O从点A运动到点B，其在纬线和经线方向上的位移分别近似为

$$\begin{cases} \Delta_{warp} = 111.199|x_1 - x_2| \\ \Delta_{weft} = \sqrt{\Delta^2 - \Delta_{warp}^2} \end{cases} \quad (19)$$

然后可以根据当地的磁偏角计算沿地磁场方向和垂直于地磁场方向的位移分量。速度大小可以通过编码器计算得到，结合IMU的数据，可以知道速度沿地磁场方向和垂直于地磁场方向的分量，同理，也可以将IMU中的加速度投影到沿地磁场方向和垂直于地磁场方向上。

卡尔曼滤波是一种广泛使用的数据融合算法[18-20]，本文使用该算法来融合北斗导航模块和编码器的数据，获取更加准确的位置和速度信息。经典的卡尔曼滤波有如下5个公式：

$$\hat{X}_k^- = A\hat{X}_{k-1} + B\hat{U}_{k-1} \quad (20)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (21)$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (22)$$

$$\hat{X}_k = \hat{X}_k^- + K_k(Z_k - H\hat{X}_k^-) \quad (23)$$

$$P_k = (I - K_k H) P_k^- \quad (24)$$

显然该系统的状态方程和观测方程分别为

$$\begin{bmatrix} s_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(\Delta t)^2 \\ \Delta t \end{bmatrix} a_{k-1} \quad (25)$$

$$\begin{bmatrix} s_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_k \\ v_k \end{bmatrix} \quad (26)$$

则 A, B, H 的值分别为 $\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \frac{1}{2}(\Delta t)^2 \\ \Delta t \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

初始状态变量 \hat{x}_0 可取传感器测量值,对于超参数 P_0, Q 和 R 的值,在不同的系统中合适的取值也不同,需要通过反复实验来确定。

2.3 路径规划

对于小规模TSP问题,常用的算法有动态规划算法、回溯法、分支限界、贪心算法和智能优化算法^[21]等。下面先比较前4种算法的时间开销和效果。

假设球场上有30个球待捡取,而机器人每次最多一次性收取10个球,且每次均从原点出发,当存球系统达到容量上限后,回到原点清空系统,而后继续工作。对于贪心算法,每次均捡取距离机器人最近的球,对于其他3个算法,先将30个球随机分为3组,每组10个球,将原点考虑在内,分别对3组进行路径规划,最后得到整体的收球顺序。在第 k 轮测试中,将算法 i 所得的路径规划对应的路程 $s_{k,i}$ 与该轮测试中4个算法中最短路程 $s_{k,min}$ 的差值 $e_{k,i}$ 定义为“误差”,将 $e_{k,i}$ 与 $s_{k,i}$ 的比值 $r_{k,i}$ 定义为“误差率”,如式(27)所示。

$$r_{k,i} = \frac{s_{k,i} - s_{k,min}}{s_{k,i}} \quad (27)$$

经过80轮随机测试,得到数据如表3所示。

表3 不同路径规划算法性能比较

Tab.3 Performance comparison of different path planning algorithms

算法	平均时间/s	平均路程/m	平均误差率
贪心算法	0.0004816	191.11898	0.12245
动态规划	0.0488536	176.20623	0.05048
分支限界	387.1770149	175.12481	0.04399
回溯算法	18.3319739	176.24461	0.05056

理论上,动态规划、分支限界、回溯算法都能够得到TSP问题的最优解,所得的路程是一致的,但由于引入了随机性,故测试结果略有差别,但随机性对时间开销几乎没有影响,根据实验结果,动态规划是这4个算法中最优秀的。

下面比较动态规划算法与遗传算法的性能。测试方法与上述同理,动态规划算法不做变化,而遗传算法的分组通过优化得到(每组依然10个球),参数设置如表4所示,动态规划与遗传算法性能比较如表5所示。

表4 路径规划遗传算法参数设置

Tab.4 Path planning genetic algorithm parameter setting

项目	内容
种群中个体数	43
迭代次数	500
交叉率	0.75
变异率	0.25
精英保留规模	1
交叉算子	部分交叉匹配
变异算子	逆转变异
选择算子	规模为4的锦标赛算子

表5 动态规划和遗传算法性能比较

Tab.5 Performance comparison between dynamic programming and genetic algorithm

算法	平均时间/s	平均路程/m	平均误差率
动态规划	0.02282	177.87215	0.30719
遗传算法	1.05514	122.82506	0.0

由表5的实验结果可知,在所有情况中,遗传算法所得到的解均优于动态规划的,且在时间上,仅仅比动态规划慢约1s。在机器人实际工作中,即使以最大速度 v_0 运动1s,其路程也只有0.5m,远小于两个算法平均路程之差55.04709m,显然遗传算法是最优选择。

3 实验 (Experiment)

作为一个在球场中工作的机器人,首先要能够顺利地完成任务,其次再考虑工作效率。定义如下两个评价指标以量化性能:

(1)若某次实验中,计算得到的路径为 p ,其理论路程(即收球顺序不变,而任意两点之间仅考虑直线距离)为 s_p ,则规定工作时间内上限为

$$T_{limit} = \frac{2s_p}{v_0} \quad (28)$$

若 t 为单次实验中实际运行时间(不考虑清空存球系统所用时间,下文同理), N 为实验次数,定义 $isSuccess$ 为单次实验中是否成功收球的标志, $SuccessTimes$ 为成功收球的次数, $SuccessRate$ 为整个实验中,能够成功收球的次数与总实验次数的

比值, 则

$$isSuccess = \begin{cases} 1, & t < T_{limit} \\ 0, & t \geq T_{limit} \end{cases} \quad (29)$$

$$SuccessTimes = \sum_{k=1}^N isSuccess_k \quad (30)$$

$$SuccessRate = \frac{1}{N} SuccessTimes \quad (31)$$

(2)工作效率仅考虑成功收球的情况。若某次实验中, 机器人的速度与时间的关系为 $v = v_k(t)$, 则该次实验的平均速度为

$$\bar{v}_k = \frac{1}{t} \int v_k(t) dt \quad (32)$$

由于仅考虑成功收球的情况, 且理论上在多次独立同分布实验中, 每次实验的权重是一样的, 因此定义整个实验中的平均速度为

$$\bar{v} = \frac{1}{SuccessTimes} \sum_{k=1}^N \bar{v}_k \cdot isSuccess_k \quad (33)$$

定义 $SpeedRate$ 为整个实验的平均速度与基础速度的比值, 即

$$SpeedRate = \frac{\bar{v}}{v_0} \quad (34)$$

如图13所示, 使机器人做10随机实验, 在每次实验中, 由人工抛掷16个球体, 随后让机器人以8个球为一组进行收球。

在每收完8个球之后, 机器人返回原点, 通过发球系统清空存球系统, 如图14所示。



图 13 实验照片

Fig.13 Experimental photos



图 14 发球系统清空存球系统

Fig.14 Ball serving system clears the ball storage system

每次实验结束后记录机器人的 $isSuccess$ 和 \bar{v}_k , 实验结果如表6所示

表 6 机器人实验结果

Tab.6 Robot Experiment Results

序号	$isSuccess$	$\bar{v}_k / (m/s)$	序号	$isSuccess$	$\bar{v}_k / (m/s)$
1	1	0.39359	6	1	0.39040
2	1	0.41032	7	1	0.40815
3	1	0.40393	8	1	0.41150
4	1	0.44104	9	1	0.30159
5	1	0.42672	10	1	0.42509

根据实验数据, 计算 $SuccessRate$ 和 $SpeedRate$ 如下所示

$$SuccessRate = 1.0$$

$$SpeedRate = 0.82491$$

在这10次实验中随机选择一次实验的过程数据, 绘制其路径图, 如图15所示。

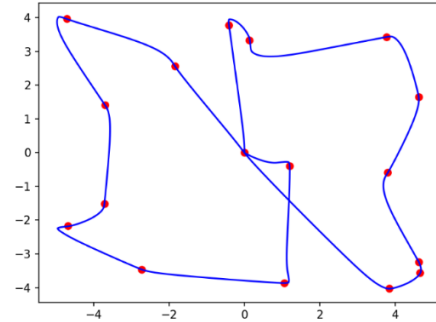


图 15 实验的路径图

Fig.15 Experimental Path Map

以该路径图为例分析机器人的工作效果和算法的有效性。首先, 从路径图中可以看出, 在每一个工作周期内 (即相邻两次位于原点附近的时间间隔), 机器人都是准确地收取8个球体, 并接着运动到原点附近清空存球系统, 直接说明了收球系统、存球系统和发球系统这些主要由机械结构组成的系统都能够正常工作, 间接说明了能源系统和控制系统这些主要由电子元件和软件组成的系统能够正常工作; 其次, 机器人能够准确地收取球体并返回原点附近, 说明了基于卡尔曼滤波的数据融合算法、基于神经网络的非线性PID算法以及目标跟随的数学模型在实际工作中是有效的。

4 结论 (Conclusion)

针对现在人工捡拾乒乓球效率较低的问题,设计了一款智能乒乓球拾取机器人。在系统整体设计方面,提出了旋转刷式收球机制,并设计了存球装置,且当容量达到上限后通过发球系统清空存球装置。这种工作方式不需要对机器人的位置进行非常精确地控制,相较于机械臂式捡球方式,提高了捡球效率,且具备存球和清空存球系统的功能,为机器人持续工作提供条件。为了机器人能够顺利工作,设计了目标跟随算法、数据融合算法以及路径规划算法。首先建立了机器人目标跟随的模型,通过基于全连接神经网络的非线性PID算法实现运动控制,其中神经网络通过遗传算法来优化参数;接着,通过经纬度坐标、编码器测得的速度以及IMU测得的角度和加速度,利用卡尔曼滤波计算机器人位置和速度;最后对比5个路径规划算法的性能,选择效果最好的遗传算法作为实际的路径规划算法。实验表明该机器人能够高效率地完成乒乓球拾取的任务。

参考文献 (References)

- [1] Mao Q-C, Sun H-M, Liu Y-B, Jia R-S. [Fast and Efficient Non-Contact Ball Detector for Picking Robots](#)[J]. IEEE Access, 2019, 7: 175487-175498.
- [2] 刁跃虎,吴涛,杨刚,等. [一种机械臂结构及其构建的捡球机器人](#)[P]. 云南省:CN201811073534.1,2024-03-29.
Diao Yuehu, Wu Tao, Yang Gang, et al. A Mechanical Arm Structure and Its Constructed Ball Collecting Robot [P]. Yunnan Province: CN201811073534.1, 2024-03-29.
- [3] Shen, C., Sun, Z. [Research on target localization recognition of automatic mobile ball-picking robot](#). J Opt 51, 866-873 (2022).
- [4] Zhang P. [A route planning algorithm for ball picking robot with maximum efficiency](#)[C]//Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV). Los Angeles, CA, USA: IEEE, 2017: 11-14.
- [5] Faizah F, Triwiyatno A, Isnanto R R. [Fuzzy Logic Implementation on Motion of Tennis Ball Picker Robot](#)[C]//2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT). Purwokerto, Indonesia: IEEE, 2021: 57-63.
- [6] Yeon H S ,Kim D ,Ryou G , et al.[System Design for Autonomous Table Tennis Ball Collecting Robot](#)[J].제어로봇시스템학회국제학술대회 논문집,2017,
- [7] T. Sato, K. Irie, S. Takaishi and H. Kawamoto. [Design and System Integration of Table Tennis Ball-Collecting Robot](#)[J]. 2024 IEEE/SICE International Symposium on System Integration (SII), Ha Long, Vietnam, 2024, pp. 1357-1362.
- [8] 张勇,周飞翔,王伟,等. [网球捡球机器人](#)[P]. 江苏省:CN202410614100.7,2024-07-26.
Zhang Yong, Zhou Feixiang, Wang Wei, et al. Tennis Ball Collecting Robot [P]. Jiangsu Province: CN202410614100.7, 2024-07-26.
- [9] Nicolaus K, Hooper J, Wood R, Ham C. [Development of an Autonomous Ball-Picking Robot](#)[C]//2016 International Conference on Collaboration Technologies and Systems (CTS). Orlando, FL, USA: IEEE, 2016: 373-378.
- [10] Wei M, Zhao C, Wang C, Chen Z. [Control Structure Design and Simulation Analysis of Table Tennis Picking Robot](#)[C]//2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT). Jilin, China: IEEE, 2023: 1558-1562.
- [11] 黄泽霞,邵春莉. [深度学习下的视觉 SLAM 综述](#)[J]. 机器人, 2023,45(06):756-768.
YAO Erliang, SONG Haitao, ZHAO Jing, et al. LiDAR-inertial SLAM Algorithm Based on Point Cloud Structure and Appearance[J]. Robot, 2023, 45(06):756-768.
- [12] 路春晓,钟焕,刘威,等. [复杂地形环境下的多传感器融合 SLAM 技术](#)[J]. 机器人, 2024, 46(04): 425-435.
LU Chunxiao, ZHONG Huan, LIU Wei, et al. Multi-sensor Fusion SLAM in Complex Terrain Environments[J]. Robot, 2024, 46(04): 425-435.
- [13] 姚二亮,宋海涛,赵婧,等. [考虑点云结构和表观信息的激光雷达-惯性 SLAM 算法](#)[J]. 机器人, 2024, 46(04): 436-449.
YAO Erliang, SONG Haitao, ZHAO Jing, et al. LiDAR-inertial SLAM Algorithm Based on Point Cloud Structure and Appearance[J]. Robot, 2024, 46(04): 436-449.
- [14] 程苗苗,翟朋辉,张英杰,等. [基于自学习非线性 PID 的音圈电机精密定位系统](#)[J]. 电工技术学报,2023,38(06):1519-1530.
CHENG Miaomiao, ZHAI Penghui, ZHANG Yingjie, et al. A Voice Coil Motor-Driven Precision Positioning System Based on Self-Learning Nonlinear PID[J]. Transactions of China Electrotechnical Society, 2023, 38(06):1519-1530.
- [15] 苏杰,曾喆昭. [非线性时变系统的自耦 PID 控制方法](#)[J]. 控制理论与应用,2022,39(02):299-306.

- SU Jie, ZENG Zhe-zhao. Auto-coupling PID control method for nonlinear time varying systems[J]. Control Theory & Applications, 2022, 39(02):299-306.
- [16] 曾喆昭, 刘文珏. [自耦 PID 控制器](#) [J]. 自动化学报, 2021, 47(02):404-422.
- ZENG ZheZhao, LIU WenJue. Self-coupling PID Controllers[J]. Acta Automatica Sinica, 2021, 47(02):404-422.
- [17] 卫宇. [考虑地球曲率情况下两点距离问题的求解](#) [J]. 航空兵器, 2008, (03):7-12.
- WEI Yu. Solution of Distance Between Two Points Considering Curvature of the Earth[J]. Aero Weaponry, 2008, (03):7-12.
- [18] 蒋帅, 陈铖, 段砚州, 等. [温度和老化影响下的锂离子电池荷电状态和健康状态联合估计方法研究](#) [J/OL]. 机械工程学报, 1-10[2024-09-06].
- JIANG Shuai, CHEN Cheng, DUAN Yanzhou, et al. Joint Estimation of State-of-charge and State-of-health for Lithium-ion Batteries under the Influence of Temperature and Aging[J]. Journal of Mechanical Engineering, 1-10[2024-09-06].
- [19] 南子寒, 刘大禹, 董明, 等. [GNSS 拒止下多源自主导航鲁棒滤波方法研究](#) [J/OL]. 航空学报, 1-13[2024-09-06].
- NAN Zihan, LIU Dayu, DONG Ming, et al. Research on Improved Robust Filtering Method for GNSS Denied Multisource Autonomous Navigation[J]. Acta Aeronautica et Astronautica Sinica, 1-13[2024-09-06].
- [20] 石饶桥, 李健, 张彪, 等. [基于卡尔曼滤波的烟气 NO_x 浓度融合测量方法](#) [J]. 机械工程学报, 2017, 53(24):63-69.
- SHI Raoqiao, LI Jian, ZHANG Biao, et al. Fusion Measurement Method for NO_x Concentration of Flue Gas Based on Kalman Filter[J]. Journal of Mechanical Engineering, 2017, 53(24):63-69.
- [21] Bladimir Toaza, Domokos Esztergár-Kiss. [A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems](#) [J]. Applied Soft Computing, 2023, 148: 110908.