

# PID 调参

## 2. 手动调参的详细步骤

### (1) 确定初始条件

固定积分和微分增益:

初始  $K_i = 0$ ,  $K_d = 0$ , 仅使用  $K_p$  进行初始测试。

确保控制器的输出范围适配硬件执行器 (例如电机的转向角度或速度范围)。

输入初始偏差信号:

设置机器人偏离目标路径的初始偏差 (例如, 路径为直线, 机器人有一个固定横向偏差)。

观察控制器的响应行为。

### (2) 调整比例增益 ( $K_p$ )

从小值开始:

将  $K_p$  设置为一个较小的值 (例如 0.1 或 1)。

观察系统对偏差的响应:

如果反应过慢, 说明  $K_p$  太小。

如果震荡严重, 说明  $K_p$  太大。

逐步增大  $K_p$ :

每次调整后观察系统的运行轨迹:

系统应快速修正偏差, 但不会因为修正过度而出现剧烈震荡。

如果震荡接近临界 (有小幅波动但不完全不稳定), 停止增加  $K_p$ 。

目标: 找到一个使系统快速响应但不震荡的  $K_p$ 。

### (3) 添加积分增益 ( $K_i$ )

解决稳态误差:

当机器人能跟踪路径但在运行一段时间后仍有轻微偏差 (稳态误差) 时, 逐步增大  $K_i$ 。

初始  $K_i$  值应较小 (例如 0.01 或 0.1), 每次小幅增加。

观察控制效果:

积分增益会累积偏差修正, 消除长时间内的稳态误差。

如果  $K_i$  太大, 会导致:

系统出现超调 (修正过度)。

系统反应过慢, 或因积分饱和导致震荡。

目标: 找到一个  $K_i$ , 既能消除稳态误差, 又不会引入显著的超调或震荡。

### (4) 添加微分增益 ( $K_d$ )

解决震荡问题:

如果系统出现震荡或快速变化时不稳定, 逐步增大  $K_d$ 。

初始  $K_d$  值较小 (例如 0.01 或 0.1), 每次小幅增加。

观察控制效果:

微分增益会根据误差变化率对系统进行修正, 抑制震荡并提高平稳性。

如果  $K_d$  太大, 会导致:

对噪声过于敏感, 系统变得不稳定。

控制器的反应速度减慢。

目标: 找到一个  $K_d$ , 在减少震荡的同时不引入过多延迟。

### 3. 验证参数的有效性

#### (1) 路径跟踪验证

##### 直线路径:

机器人应能快速调整偏差, 平稳地沿直线路径行驶。

检查是否存在偏差修正不足或超调的问题。

##### 弯道测试:

模拟机器人通过弯道时的路径调整。

检查转向是否平滑, 避免急转弯时偏离路径。

#### (2) 十字路口测试

模拟机器人在十字路口切换路径的场景:

机器人应能快速修正路径, 避免因偏差过大导致的跟踪失败。

验证机器人在转向过程中是否出现剧烈震荡。

#### (3) 噪声测试

在仿真中加入随机噪声 (如传感器抖动或路径数据的高频波动)。

验证 PID 参数是否能在噪声条件下保持稳定控制。

### 问题 4: 路径跟踪算法与硬件调试过程中如何实现闭环控制?

#### 预期问题细化:

如何确保路径跟踪算法能够实时闭环控制机器人运动?

跟踪偏差的计算是基于什么标准? 是否采用了特定的目标函数来优化路径误差?

在硬件调试过程中, 遇到哪些闭环控制相关的问题? 如何解决?

#### 解答方向:

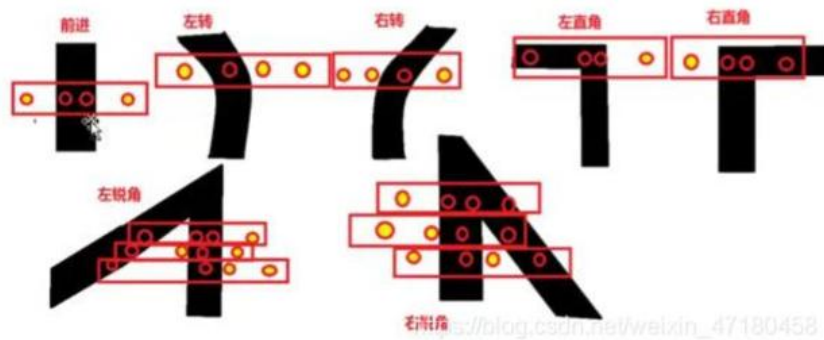
闭环控制通过 PID 算法实时调整电机转速来校正路径偏差。

提到基于摄像头或红外传感器计算机人的实际位置与期望路径之间的偏差值。

通过调整控制周期与数据采样频率, 减少闭环控制中的延迟问题。

## 红外布局原理

我们本次实验采用的是四路红外传感器分别连接在 Arduino 主控板上的 A1,A2,A3,A4 口上。其中中间两路巡线是一直在黑线上, 小车会直行, 当任意一个出来, 则小车会自动纠正, 如果最外面的检测到黑线, 则小车以更大速度纠正到正确黑线上面。



中间两路巡线一直在黑线上, 小车会直行, 当任意一个出来, 小车会自动纠正。如果最外面的检测到黑线。则小车以更大速度纠正到黑线上去。在算法上, 优先处理锐角, 直角等外围传感器检测到黑线的状况