《模块化机器人综合设计实践》

实验报告

学生A姓名： 学号：

学生B姓名： 学号：

学生C姓名： 学号：

一、控制系统设计

1.3执行器设计

在整体的硬件性能上，我们提出以下要求：方便转动方向的调节；保证足够的爬坡能力，并且不发生打滑；保证小车以当前方向行驶时不发生偏移或偏移量较少。

基于硬件要求，我们提出以下三种方案:

方案一：前轮转向，后轮驱动。执行器选择一个180°舵机和一个360°舵机。其中，180°舵机控制前轮转向，360°舵机通过齿轮与轴控制后轮转动。

方案二：前轮转向，后轮驱动，执行器选择一个180°舵机和两个360°舵机。其中，180°舵机控制前轮转向，两个360°舵机分别单独控制对应后轮的转动。

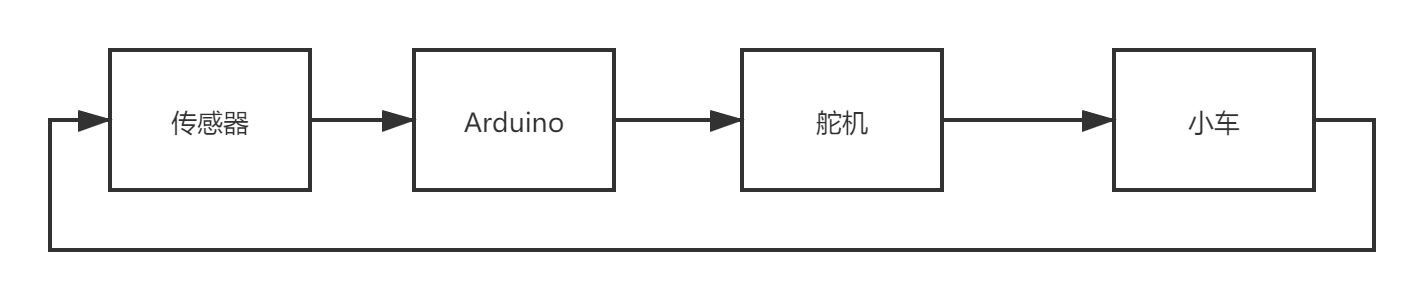
方案三：后轮驱动，差速转向。执行器选择两个360°舵机，分别单独与左右后轮相连。

经比较，方案一的优点在于调节转动方向的灵敏性和行驶时较小的方向偏移，但爬坡能力相对较弱。方案二弥补了方案一驱动能力不足的缺点，但由于左右舵机在制造时的差异，难以保证两个后轮转速相同，从而存在一定的行驶方向偏移。方案三使用差速转向，小车转向性能受物理环境影响较大，且调整方向的复杂性略高于方案一和方案二。最后，我们决定使用方案一，通过优化小车的机械机构来减轻驱动的负载。

在使用执行器（舵机）时，我们需要对两种执行器进行单独控制。其中，对于180°舵机，需要控制它的转角。对于360°舵机，需要控制它的转速和转向。使用时，需要对舵机的参数进行预校正。180°舵机在安装后需要校准相对于跷跷板的正方向，360°舵机在安装前需要测量在arduino舵机函数库中的转动停止时的PWM占空比。

1.4控制算法设计

控制系统框图如图所示



控制系统框图

基于控制系统框图，我们对控制系统提出以下性能要求：响应速度较快；超调量较小；震荡次数较少。由于跷跷板触地对整个系统的影响较大，所以控制系统对响应速度和超调量具有较高要求。

在采用方案一作为执行器选择方案后，控制系统将受到以下外界干扰：跷跷板转动时惯性的影响，跷跷板转动轴摩擦力的影响等，同时，传感器在使用过程中会随时间产生较大的漂移量。惯性和摩擦的影响可以通过提高响应速度和减小超调量来削弱，传感器的漂移可以通过滤波算法进行校正，

控制算法分为两部分，传感器信号处理与舵机驱动。由于传感器在测量数据时产生漂移，我们使用滤波算法进行校核，其中，常见的滤波算法有均值滤波、互补滤波、卡尔曼滤波等。

均值滤波的优点在于计算简单，通过计算多次数据的平均值，将平均值作为真实值。相应地，对滤波效果一般，对噪声的抑制能力较弱。

互补滤波是通过基于传感器对角度、角速度、加速度的测量灵敏性不同而设计的滤波算法。传感器对小车的加速度比较敏感，取瞬时值计算倾角误差比较大；而陀螺仪积分得到的角度不受小车加速度的影响，但是随着时间的增加积分漂移和温度漂移带来的误差比较大。由于加速度计的低频特性较好，加速度计的角度可以直接得出，所以没有累计误差，长时间内性能较好，可以通过高通滤波来抑制低频。而陀螺仪的角速度积分由于误差的积累，长时间的性能较差而短时间内性能较好，通过低通来抑制高频。互补滤波则需要选择低通与高通的切换频率。互补滤波算式为



卡尔曼滤波是一种利用线性系统状态方程，通过系统输入输出观测数据，对系统状态进行最优估计的算法。由于观测数据中包括系统中的噪声和干扰的影响，所以最优估计也可看作是滤波过程。卡尔曼滤波需要对系统进行建模，同时需要干扰噪声的方差等数据。

比较三种滤波方法，均值滤波计算简单但无法达到理想滤波效果。卡尔曼滤波较为精确，但对建模的精确度较高，受模型准确性的影响较大。同时，噪声方差等数值的要求难以满足，设计难度较大。互补滤波具有较好的滤波效果，同时相对于卡尔曼滤波，设计要求相对简单。综合考虑，我们决定使用互补滤波作为传感器信号处理的滤波算法。

在驱动部分中，我们使用PID算法作为调平衡的核心算法。在PID算法中，定值和实际输出值构成控制偏差，将偏差按比例、积分和微分通过线性组合构成控制量，对被控对象进行控制。

PID控制器各校正环节的作用如下：

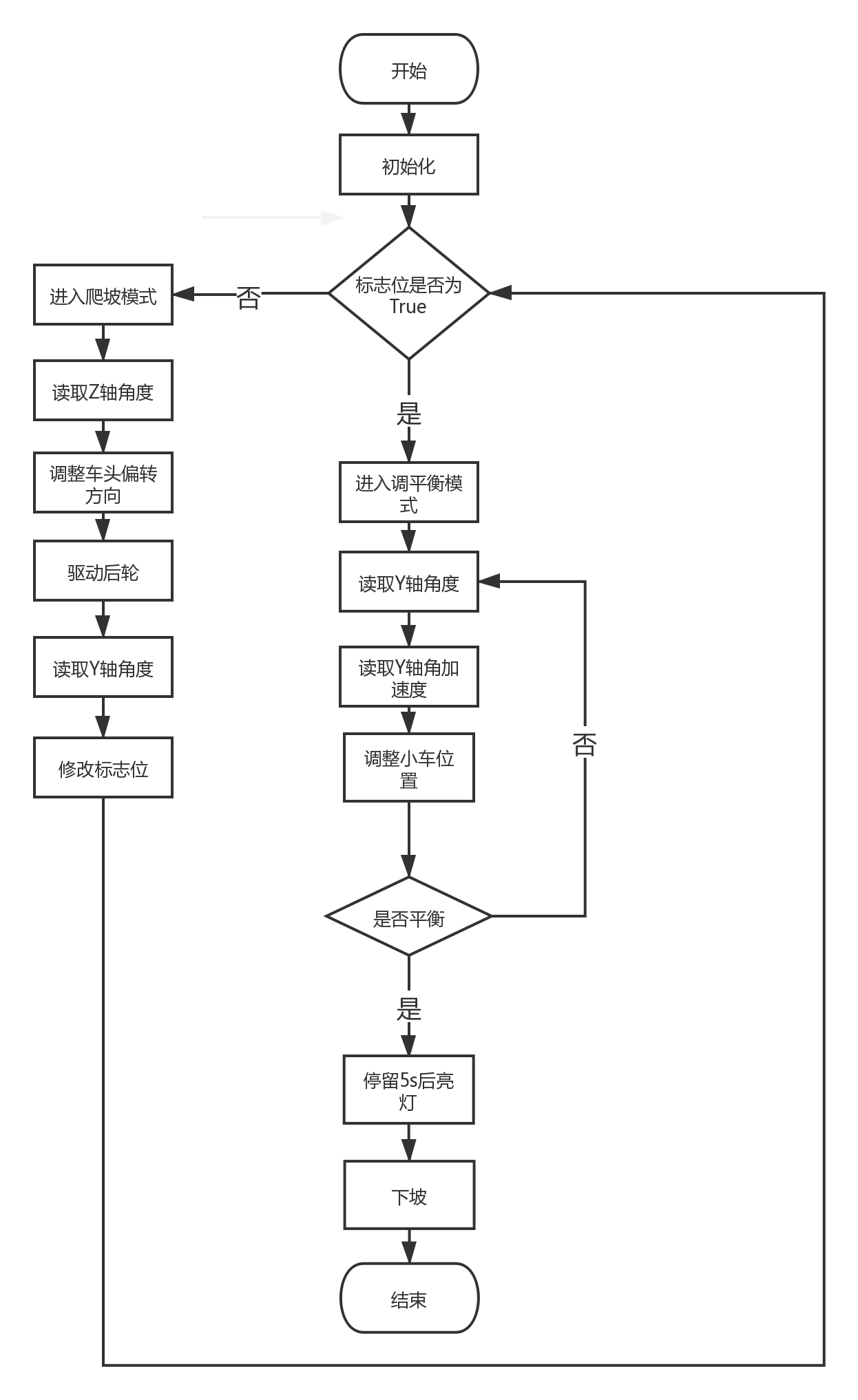
比例环节：即时成比例地反应控制系统的偏差信号e(t)，偏差一旦产生，控制器立即产生控制作用以减小误差。当偏差e=0时，控制作用也为0。因此，比例控制是基于偏差进行调节的，即有差调节。

积分环节：能对误差进行记忆，主要用于消除静差，提高系统的无差度，积分作用的强弱取决于积分时间常数T，T越大，积分作用越弱，反之则越强。

微分环节：能反映偏差信号的变化趋势(变化速率)，并能在偏差信号值变得太大之前，在系统中引入一个有效的早期修正信号，从而加快系统的动作速度，减小调节时间。

同时，通过比较PD、PID、PI的使用效果。考虑到系对超调量和响应速度的要求较高，我们最终使用PI，该算法在满足方案要求的同时，能够减少震荡。

控制流程图如下：



控制流程图

图1.4.1 软件结构

开机后，系统初始化，读取JY901数据。通过判断标志位来选择小车工作模式。

在爬坡模式中，通过不断读取Z轴角度，和设定的标准值对比来实时修正小车的偏转。同时，不断读取Y轴角度，通过判断Y轴角度是否小于设定值来修改标志位。当标志位为False时，重复执行爬坡模式。

在调平衡模式中，根据9轴传感器的摆放位置，读取Y轴的角速度和角加速度作为调平衡算法的其中两个输入参数。当达到平衡并持续5秒之后，LED灯亮起，开始执行下坡任务，启动后轮，后轮倒转。

二、控制系统的制作与调试

2.3控制程序的编写与调试

控制程序分为两个模块，传感器信号处理模块与舵机驱动模块。信号处理模块使用驱动库文件，通过设定arduino与JY901的通信波特率实现读取X、Y、Z轴加速度、角度、角速度。之后通过互补滤波对角度数据进行处理。

驱动模块的核心为PI控制器，将角速度值作为偏差，控制量 。其中， 为角度值互补滤波处理后的结果。根据与测量结果，当 时，舵机正转，小车前进；反之舵机反转，小车后退。

在信号处理模块中，需要调用函数测得单词程序的运行周期，作为互补滤波公式中的 ,并 设置为0.98。互补滤波公式如下：

在该模块中，需要注意JY901的波特率设置，并注意根据传感器的摆放位置合理选用X、Y、Z轴的数据。通过arduino接受传感器数据并发送到上位机，可以检测信号读取是否正确。

在驱动模块中，需要注意小车在向前和向后行驶时，驱动能力不同，需要设计两套不同的 参数。同时，在小车于跷跷板直行并即将进入调平衡时，应当使速度尽量慢，减小动静摩擦不同以及惯性带来的影响。基于对任务时间限制的考量，我们使速度随时间降低，降低的幅度需要预先测量。

三、控制系统的测试与分析

3.1测试方法

评价小车性能有以下几个要素：行驶角度调整时间（调整角度为45°）；调平衡的次数（前后移动一次为调平衡一次）；调平衡的时间；调平衡总过程跷跷板平均倾角。其中，行驶角度调整时间代表180°舵机的灵敏性；调平衡的次数反映系统的震荡特性；调平衡的时间反映系统的综合性能；平均倾角反映系统的超调量。

同时，为测试滤波效果，测试工作将在滤波和不滤波两种条件下进行。

为测量以上要素，我们使用秒表来计算时间，并通过量角器测量倾角，人工记录角度大小和调平衡的次数。

3.2测试数据与现象

在测试时，我们发现小车在调整过程中会使跷跷板触地，从而严重影响系统的平衡。成功的测试均呈现震荡幅度由大到小变化并最终不变的现象。经过性能比较，我们选择测试中表现最理想的一组参数作为使用参数。参数性能测试表如下。

首先是使用互补滤波的数据。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 行驶角度调整时间 | 2.7 | 2.2s | 3s | 2.3s | 2.2s |

行驶角度调整时间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 调平衡次数 | 7 | 5 | 2 | 8 | 6 |

调平衡次数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 调平衡时间 | 20 | 15 | 31 | 17 | 22 |

调平衡时间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 跷跷板倾角 | 4.2° | 3.3° | 3.7° | 2.8° | 2.6° |

平均角度

以下是不使用滤波的数据。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 行驶角度调整时间 | 3.2s | 3.3s | 4s | 3.6s | 3.7s |

行驶角度调整时间3.56

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 调平衡次数 | 10 | 7 | 9 | 13 | 11 |

调平衡次数10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 调平衡时间 | 22 | 30 | 27 | 39 | 34 |

调平衡时间30.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试次数 | 1 | 2 | 3 | 4 | 5 |
| 跷跷板倾角 | 3.9° | 4.7° | 4.1° | 3.4° | 3.7° |

平均角度3.96

在多次调平衡测试中，测试量围绕在某个值在某个范围内上下波动。并且，使用滤波的效果明显强于不使用滤波。

3.3结果分析

使用滤波后，行驶角度调整平均时间T=2.48s，调平衡平均次数N=5.6，调平衡平均时间t=21s，调平衡总过程跷跷板平均倾角=3.32°。

相对于其他数据，调平衡总过程的平均倾角较大，说明系统存在较大超调，即驱动模块中，KI参数较大。需要适当减小来抑制系统超调。

将使用滤波前与使用滤波后对比，行驶角度调整平均时间T=3.56s，调平衡平均次数N=10，调平衡平均时间t=30.4s，调平衡总过程跷跷板平均倾角=3.96°。

可以看出，使用滤波将提高系统整体性能，包括响应速度、超调、震荡特性等。

四、总结与心得

文博正

欧亚非

曹啸