**PID智能小车实验报告**

王晨昊2018011202003

1. **PID智能小车控制系统的基本工作原理**

本实验涉及了PID调节的原理，超声波测距的原理，电动机转动的原理以及数码管动态扫描的原理。这些原理在实验代码中分别体现在各个模块的函数中。下面对这四个重要的原理做阐述。

### PID原理

PID调节是连续控制系统中技术最成熟、应用最广泛的调节方式。PID调节实质是根据输入的偏差值，按照比例、积分、微分的函数关系进行运算，运算结果用以控制输出，使系统达到稳定的算法。

连续系统的PID公式为：



其中，为比例系数，为积分时间常数，为微分时间常数，为偏差量，为控制量。

其中比例控制能提高系统的动态响应速度，从而快速减少误差，但是不能消除误差，在误差较小时会发生震荡。为积分控制系数，用于消除稳态误差，只要系统存在误差积分作用就会不断积累，输出控制量来消除稳态误差，如果偏差为零积分停止，但是积分作用太强会使得超调量加大，甚至使系统出现震荡。为微分控制，微分与变化率有关，可以减小超调量来克服震荡，使系统稳定性提高，同时加快响应速度，使系统稳定更快有更好的动态性能。PID公式离散化与增量式PID

计算机控制必须对公式进行离散化，具体就是用求和代替积分，用向后差分来代替微分，使模拟PID离散化为数字形式的差分方程：



将PID公式离散化后就可以用计算机来实现PID算法，但是根据离散PID公式可以看出，输出的数字调节信号与过去所有的偏差信号有关，计算机需要将所有过去的进行累加，运算量很大，这种离散的PID算法一般称为位置式PID算法。

为了减少计算量对位置式PID算法进行改进，使用增量式PID算法。

增量式PID与位置式PID算法没有本质区别，对位置式PID取增量，可以得到：



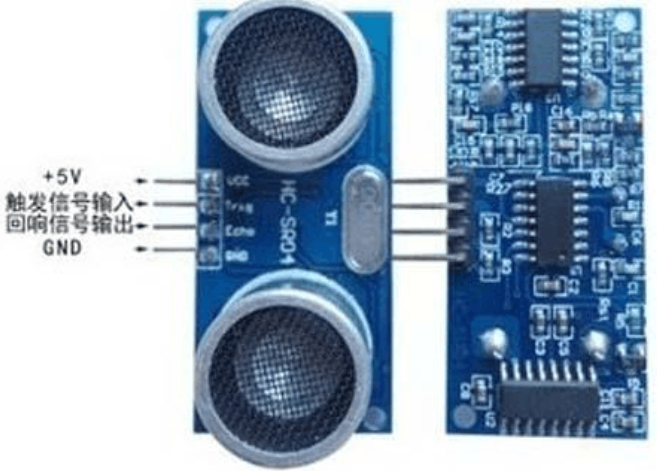
因此当前时刻控制量可以由上次采样时刻的控制量得到：



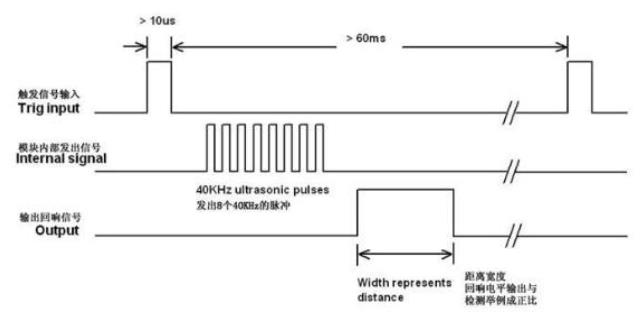
增量式PID算法只需要、、三个时刻的采样值，大幅减少了计算量和内存的使用量，在工程中更为常用。

### 超声波模块测距原理

使用hc-sr04超声波模块采集距离信息，hc-sr04模块具有四个引脚，两个为电源与GND引脚，另外两个用于反馈距离信息，模块的模型如下图所示：



在触发信号输入引脚给至少10us的高电平信号，hc-sr04模块会自动发送8个40kHz的方波信号，当hc-sr04模块检测到回波信号的时候，在回响信号输出引脚会输出一个低电平，低电平的持续时间就是超声波从发射到返回的时间，测试距离=（高电平持续时间×声速）/2。模块工作的时序图如下：



### 驱动电机原理

电机的两个引脚产生电位差时会驱动电机转动，将两个引脚的电平翻转会使电机反转（后退），用一个PWM波信号来驱动电机，改变PWM信号的占空比可以改变电机的驱动速度实现小车的行进速度的调整。

### 4．数码管动态扫描原理

本实验需要数码管显示小车与障碍物之间的距离，数码管动态显示是将所有数码管的 8个显示笔划“a,b,c,d,e,f,g,dp”的同名端连在一起，另外为每个数码管的公共极 COM 增加位选通控制电路，位选通由各自独立的 I/O 线控制，当单片机输出字形码时，所有数码管都接收到相同的字形码，但哪个数码管会显示出字形，取决于单片机对位选通 COM 端电路的控制，所以我们只要将需要显示的数码管的选通控制打开，该位就显示出字形，没有选通的数码管则不会点亮。通过分时轮流控制各个数码管COM端，就使各个数码管轮流受控显示，这就是动态驱动。在轮流显示过程中，每位数码管的点亮时间很短，由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位数码管并非同时点亮，但只要扫描的速度足够快就不会有闪烁感。

1. **实验结果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 比例（Proportion）值 | 积分（Integral）值 | 微分（Derivative）值 | 实验现象 | 结果分析 |
| 0.2 | 8 | 0.01 | 8.6-14.6 | 超调量增大，系统出现震荡 |
| 0.5 | 2 | 0.5 | 8.6-14.1 | 消除误差，但如果存在系统误差，会不断累积 |
| 0.8 | 1.8 | 0.8 | 6.5-10.8 | 更快的动态性能，减少超调量，使系统稳定，加快速度 |
| 1 | 1.2 | 1 | 7.3-15.5 | 减小误差，但不能消除误差，小车较稳定 |
| 1 | 1 | 1 | 6.8-9.2 | 减小误差，但不能消除误差，小车较稳定 |
| 0.5 | 0.02 | 0.7 | 稳定在十 | 小车稳定 |

1. **对实验结果和理论分析结果进行对比分析和讨论**

在第一次实验中，我们将积分参数调得非常大，验证了PID调节积分部分的作用，发现效果非常明显，超调量增大，系统出现非常大的振荡，导致小车永远无法找到平衡点。这样的现象与实验原理中PID积分调节的作用是吻合的。

在第二次实验中，我们增大了比例部分和微分部分的参数，大比例减小了积分部分的参数，发现现象和之前相差很大，由于积分部分仍然很大，所以系统可以消除误差，但是由于比例部分和微分部分过大，导致积分在消除误差的速度追赶不上新产生的误差，导致系统误差的不断积累不能完全消除。我们对比实验原理，发现这一现象和PID中三者的相互作用是吻合的。

在第三次实验中，我们主要增大了微分部分的参数，发现在比例部分较大时，系统的动态性能更快，使超调量减小，进而验证了实验原理中PID微分部分的作用，和原理相吻合。

在接下来的几次实验中，我们基本将PID三者的大小在前三次实验的范围内进行寻找，通过分别改变三者之间的比例（相对大小）和三者各自的大小（绝对大小）来寻找一个能使小车稳定的点，最终发现在PID参数分别为0.5，0.02，0.7时达到稳定。由于每一个小车的具体情况以及实验的阻力粗糙度等外部因素都不一样，导致我们的数据不一定是一个完全合理稳定的数值。只能做一定的参考。

1. **心得体会**

通过这次实验，验证了PID的原理。比例控制系数能提高系统的动态响应速度，只有比例系数时最终会有误差。积分控制系数，用于消除稳态误差，但是积分作用太强会使得超调量加大，甚至使系统出现震荡。微分控制系数，可以减小超调量来克服震荡，使系统稳定性提高，同时加快响应速度。

这次实验中，小车经常没电，且超声波模块有时会出问题，测出的距离为0不变，给测试带来了障碍。

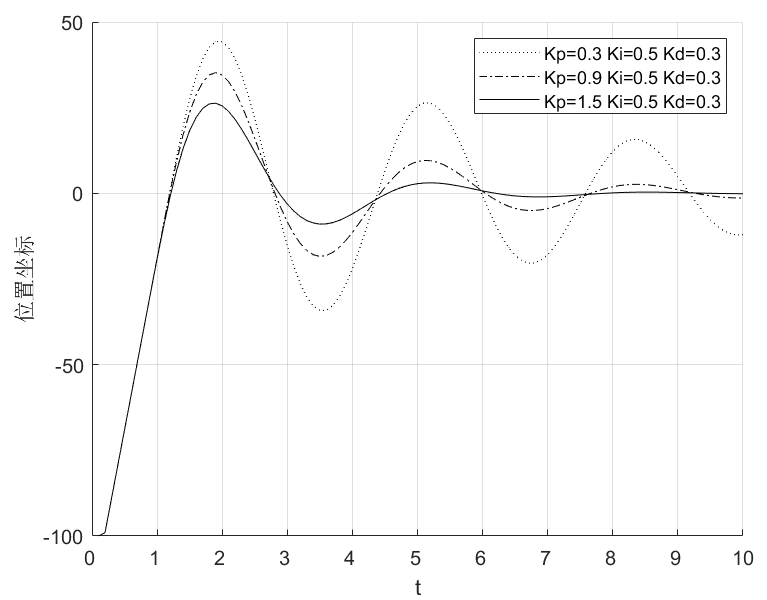
建议可以先通过Matlab仿真确定大致的参数范围，再用小车进行验证，提高实验效率。

**补充内容**

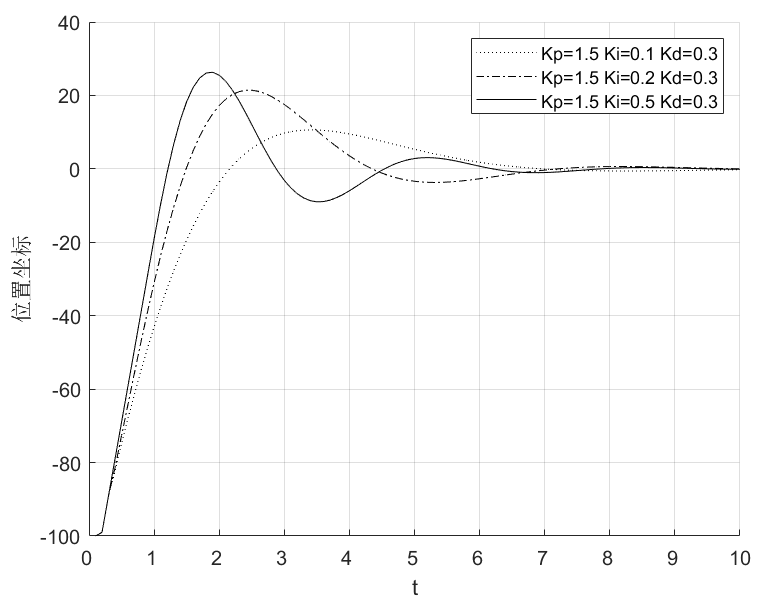
仿照单片机的程序，自己用matlab编写了增量式PID的代码，希望仿真小车的运动。但对于小车的运动系统，不知其系统函数，这导致了仿真的局限性。仿真时不易模拟电机的滞后作用及小车的惯性，所以微分调节在仿真时几乎没有作用。

仿真思路：小车的位置坐标x（一维的）为受控对象，目标位置为x=0, 初始位置x=-100. 设仿真步长dt, 位置坐标的迭代公式为x(k)=x(k-1)+v(k-1)\*dt. 由于dt在仿真过程中为常数，所以速度v即为调节量。PID控制的输出即为速度v, 输入为位置的偏差（不同时刻的，用于计算调节量的增量）。

仿真结果：



增大比例调节系数Kp，系统更快稳定。



增大积分调节系数Ki，波动更剧烈，变化更快

Matlab代码：

function test\_PID

%画不同参数下位置变化图

close all

global dt lim\_v

% PID=[0.3 0.5 0.3;%Kp Ki Kd

% 0.9 0.5 0.3;%第二组参数

% 1.5 0.5 0.3];

PID=[1.5 0.1 0.3;%Kp Ki Kd

1.5 0.2 0.3;%第二组参数

1.5 0.5 0.3];

% PID=[0.9 0.9 0;%Kp Ki Kd

% 0.9 0.9 0.5;%第二组参数

% 0.9 0.9 0.9];

D=0;%目标位置

lim\_v=1;%是否有速度限制

dt=0.1;%仿真步长

t=10; %仿真时长

N=t/dt;

x=zeros(1,N);%初始化位置坐标

v=zeros(1,N);%速度

e=zeros(1,N);%位置误差 被调量为位置

tlist=1:N;

text=cell(1,size(PID,1));%用于画图显示参数

line=cell(1,size(PID,1));%用于画图显示线型

line{1}='k:';

line{2}='k-.';

line{3}='k-';

x(1)=-100;%初始位置

v(1)=10;%初始速度

e(1)=D-x(1);

hold on

for i=1:size(PID,1)%遍历每组参数

Kp=PID(i,1);

Ki=PID(i,2);

Kd=PID(i,3);

x(2)=x(1)+v(1)\*dt;

e(2)=D-x(2);

v(2)=Kp\*e(2)+Ki\*(e(1)+e(2))+Kd\*(e(2)-e(1))/dt;

if lim\_v==1 %速度限制

if v(2)>100

v(2)=100;

end

if v(2)<-100

v(2)=-100;

end

end

for k=3:N

x(k)=x(k-1)+v(k-1)\*dt;

e(k)=D-x(k);

v(k)=cal\_v(Kp,Ki,Kd,v(k-1),e(k),e(k-1),e(k-2));

end

plot(tlist,x,line{i})

text{i}=['Kp=',num2str(Kp),' Ki=',num2str(Ki),' Kd=',num2str(Kd)];

end

legend(text{1},text{2},text{3})

xlabel('t')

ylabel('位置坐标')

set(gca, 'XTick',0:N/10:N,'XTicklabel',dt\*[0:N/10:N]);

grid on

function vn=cal\_v(Kp,Ki,Kd,v,e,el,ell)

%增量式PID

%e为偏差 el为前一时刻偏差， ell为再前一时刻

%if dis>=115 && dis<=135

global dt lim\_v

vn=v+Kp\*(e-el)+Ki\*e+Kd\*(e-2\*el+ell)/dt;

if lim\_v==1 %速度限制

if vn>100

vn=100;

end

if vn<-100

vn=-100;

end

end