**#include**"math.h"

**#define** IIRNUMBER 2 //IIRNUMBER=滤波器阶数n+1，对应输入输出的延迟序列的个数n+1(1代表当前时刻序列),例如y[n]=0.7757y[n-1]+0.1122x[n]+0.1122x[n-1]中，n=1

**#define** SIGNAL1F 1000 //仿真信号由1个正弦信号(其频率为SIGNAL1F=1000)和1个余弦信号(其频率为SIGNAL2F=4500)

**#define** SIGNAL2F 4500 //仿真信号由1个余弦信号(其频率为SIGNAL2F=4500)和1个正弦信号(其频率为SIGNAL1F=1000)

**#define** SAMPLEF 10000 //在本程序中没有用到采样频率为SAMPLEF=10000

**#define** PI 3.1415926

**float** **InputWave**();

**float** **IIR**();

**float** fBn[IIRNUMBER]={ 0.0,0.7757 };// fBn是系统函数H(z)的分母的系数，且注意fBn[0]=0。

**float** fAn[IIRNUMBER]={ 0.1122,0.1122 };//fAn是系统函数H(z)的分子的系数，

**float** fXn[IIRNUMBER]={ 0.0 };//fXn初始化值[0,0]

**float** fYn[IIRNUMBER]={ 0.0 };//fYn初始化值[0,0]

**float** fInput,fOutput; //单值(当前时刻)滤波器输入输出值，滤波前滤波器输入波形值fInput或波形产生电路输出值，滤波后输出波形值fOutput，

**float** fSignal1,fSignal2; //单值(当前时刻)正弦函数和余弦函数的自变量的值(是频率和时刻的乘积)，

**float** fStepSignal1,fStepSignal2;//是每次让正弦函数和余弦函数的自变量值变化的步进值(即将弧度周期2\*PI分成N等份离散化后的间隔)，

**float** f2PI; //用来表示弧度周期2\*PI：f2PI=2\*PI

**int** i;

**float** fIn[256],fOut[256];//数组记录(前256个时刻)滤波器输入输出值，滤波前滤波器输入波形值fIn，滤波后输出波形值fOut，

**int** nIn,nOut;//nIn,nOut用来计数滤波器滤波的次数，使得次数超过256时，nIn,nOut复位到0，以便将数组fIn[nIn]和fOut[nOut]的数据更新（滤波256次后滤波数据覆盖以前的记录），

**main**()

{

 nIn=0; nOut=0;

 fInput=fOutput=0;

 f2PI=2\*PI;

 fSignal1=0.0;

 fSignal2=PI\*0.1;

// fStepSignal1=2\*PI/30;

// fStepSignal2=2\*PI\*1.4;

 fStepSignal1=2\*PI/50;

 fStepSignal2=2\*PI/2.5;

 **while** ( 1 )

 {

 fInput=InputWave();

 fIn[nIn]=fInput;

 nIn++; nIn%=256;

 fOutput=IIR(); //单值(当前时刻)滤波后滤波器输出滤波值fOutput，

 fOut[nOut]=fOutput;//将单值(当前时刻)滤波器输出滤波值fOutput记录到数组中，

 nOut++; // break point

 **if** ( nOut>=256 )

 {

 nOut=0;

 }

 }

}

**float** **InputWave**()

{

 **for** ( i=IIRNUMBER-1;i>0;i-- )

 {

 fXn[i]=fXn[i-1]; //输入信号序列向后延迟一个时刻, 即上次调用InputWave()时的当前时刻fXn[0],变成当前调用InputWave()时的前一时刻fXn[1]

 fYn[i]=fYn[i-1];//输入信号序列向后延迟一个时刻, 即上次调用InputWave()时的当前时刻fYn[0],变成当前调用InputWave()时的前一时刻fYn[1]

 }

 fXn[0]=**sin**((**double**)fSignal1)+**cos**((**double**)fSignal2)/6.0;//产生当前时刻[0]的仿真信号，以便输入到滤波器进行滤波

 //fYn[0]=0.0;

 fYn[0]=fOutput;//单值(当前时刻[0])滤波后滤波器输出滤波值fYn[0]=fOutput，原实验箱程序fYn[0]=0错;

 fSignal1+=fStepSignal1;

 **if** ( fSignal1>=f2PI ) fSignal1-=f2PI;

 fSignal2+=fStepSignal2;

 **if** ( fSignal2>=f2PI ) fSignal2-=f2PI;

 **return**(fXn[0]);

}

**float** **IIR**()

{

 **float** fSum;

 fSum=0.0;

 **for** ( i=0;i<IIRNUMBER;i++ )

 {

 fSum+=(fXn[i]\*fAn[i]); //为迭代计算输出y[n]=0\*y[n]+0.7757y[n-1]+0.1122x[n]+0.1122x[n-1]，先累加0.1122x[n]+0.1122x[n-1]

 fSum+=(fYn[i]\*fBn[i]); //再累加0\*y[n]+0.7757y[n-1]，最后输出y[n]=0\*y[n]+0.7757y[n-1]+0.1122x[n]+0.1122x[n-1]

 }

 **return**(fSum);

}