【实验目的】

加深对 IIR 数字滤波器的级联结构的理解，掌握应用 MATLAB 实现

给定系统的级联结构。

【实验内容】

利用 MATLAB 编程将给定直接结构的系统用级联结构来实现，要求如下：编写一个通用的函数，输入参数为直接形式的系统函数分母分子中的各阶系数A=(a0,a1,…,aN) ，B=(b0,b1,…,bN)。输出为矩阵 Numerator 和 Denominator，其中 Numerator 和 Denominator 的每一行分别代表一个二阶节（一阶节可看做二阶节的特例）中分子和分母中的系数。

给定一个系统，利用编写的函数求出级联结构中各个二阶节的系数，画出系统的级联结构图。一个系统由如下的差分方程给出，编写求级联结构各二阶节系数的函数，用主程序调用该函数求出该系统各级二阶节的系数，根据程序运行结果，画出该系统的级联结构。

**13y(n)+8y(n-1)+2y(n-2)-4y(n-3)-y(n-4)=2x(n)-3x(n-1)+9x(n-2)-26x(n-3)+18x(n-4)**

【实验步骤】

%主程序main.m 调用函数TheCalledFunction.m

n = input('输入阶数：');

char coefficient\_Y(i) coefficient\_X(i);

%循环输入Y的系数

for i=1:n

coefficient\_Y(i)=input('输入Y的系数：')

end

%循环输入X的系数

for j=1:n

coefficient\_X(j)=input('输入X的系数：')

end

E=TheCalledFunction(coefficient\_Y)%调用函数,求出Y的多项式结果

F=TheCalledFunction(coefficient\_X)%调用函数,求出X的多项式结果

function cal=TheCalledFunction(xxi)

N=length(xxi);%求多项式长度

root=roots(xxi);%求根

sort=cplxpair(root);%将求出的根按照复共轭对进行排序

k=(N-1)/2;%计算级联个数

%下面的循环确定最终的多项式,poly将一对根转化为对应的多项式

for J=1:k

sortpro=[sort(2\*J-1) sort(2\*J)];

cal(J,:)=poly(sortpro);

endfunction cal=TheCalledFunction(xxi)

N=length(xxi);%求多项式长度

root=roots(xxi);%求根

sort=cplxpair(root);%将求出的根按照复共轭对进行排序

k=(N-1)/2;%计算级联个数

%下面的循环确定最终的多项式,poly将一对根转化为对应的多项式

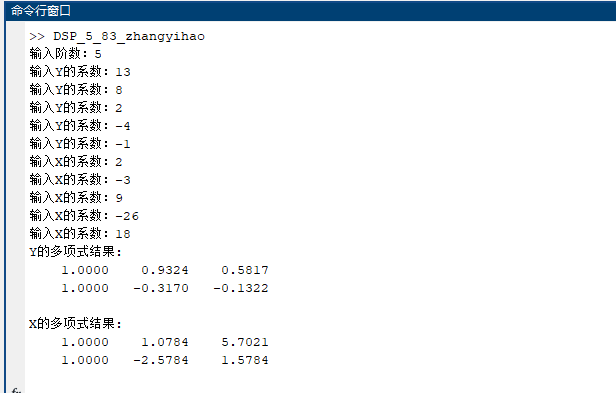
for J=1:k

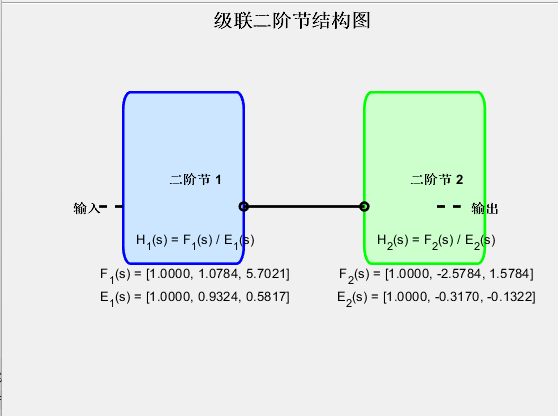
sortpro=[sort(2\*J-1) sort(2\*J)];

cal(J,:)=poly(sortpro);

end

【实验结果及分析】





【实验心得】

这次实验让我IIR数字滤波器的级联结构有了更加深刻的理解，并通过 MATLAB编程实现了从直接结构到级联结构的转换。实验的过程充满了挑战，也让我收获了不少。从输入系统的系数开始，我使用 MATLAB 的 roots 函数计算多项式的根，接着利用cplxpair将复共轭根配对，最终通过 poly函数生成每个二阶节的系数。通过这些步骤，我不仅将复杂的差分方程转化成了更简洁的级联结构，还掌握了如何将滤波器设计转化为实际的 MATLAB 代码。虽然在求根和级联结构的实现过程中遇到了一些困难，但通过不断的调试，我终于成功实现了目标。实验让我深刻体会到数字滤波器设计中的数学与编程技巧如何结合，也让我更加熟悉了 MATLAB 在信号处理中的应用。这次实验不仅让我对滤波器的级联结构有了更直观的认识，也让我感受到了通过编程解决实际问题的成就感。