Министерство Образования, Культуры,

Исследований Республики Молдова

Технический Университет Молдовы

Департамент Программная Инженерия и Автоматика

**Отчёт**

по лабораторной работе №6

**по дисциплине «PS»**

Выполнил: ст.гр. TI-197

Шарафудинов Н.

Проверил: Romanenco A.

Кишинёв - 2022

**Лабораторная работа №6**

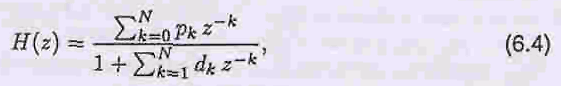
Тема:Структуры цифровых фильтров

Цель: Научиться строить различные структуры фильтров в зависимости от желаемых свойств будущей структуры.

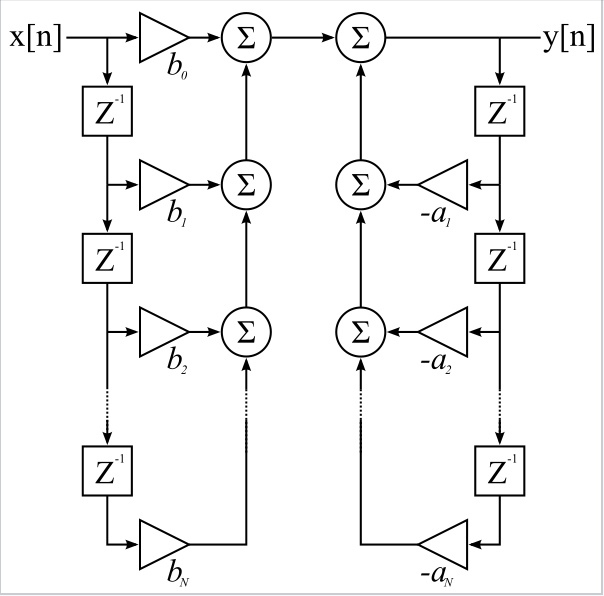
# Краткая теория:

В данной лабораторной работе рассматриваются различные структуры фильтров (параллельная и каскадная). Определённая структура фильтров позволяет добиться подходящих отношений между внутренними переменными, а, соответственно, должного выходного сигнала.

Любой каузальный IIR (Infinite Impulse Response – фильтр с бесконечной импульсной характеристикой) фильтр характеризуется функцией преобразования:



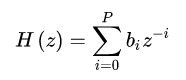
(8)



**Фильтр с конечной импульсной характеристикой** один из видов линейных цифровых фильтров, характерной особенностью которого является ограниченность по времени его импульсной характеристики (с какого-то момента времени она становится точно равной нулю). Такой фильтр называют ещё нерекурсивным из-за отсутствия обратной связи. Знаменатель передаточной функции такого фильтра — константа.

КИХ-фильтр обладает рядом полезных свойств, из-за которых он иногда более предпочтителен в использовании, чем БИХ-фильтр. Вот некоторые из них:

* КИХ-фильтры устойчивы.
* КИХ-фильтры при реализации не требуют наличия обратной связи.
* Фаза КИХ-фильтров может быть сделана линейной





Цифровые фильтры являются частным случаем линейных инвариантных систем. Существенное ограничение связано с физической реализуемостью системы.

**Определение**. Система называется физически реализуемой, если сигнал на выходе в момент времени *t* зависит от входных сигналов в моменты времени .

Пусть имеется ЛИС (линейная инвариантная система) . Рассмотрим сосредоточенную в одной точке последовательность . Пусть , а по определению . Для произвольной последовательности  справедливо разложение . В силу линейности  а в силу инвариантности . Окончательно, если , то  (9)

Другими словами, реакция на любую последовательность получается с помощью свертки этой последовательности и последовательности , называемой импульсной реакцией, или функцией отклика.



**Используемые команды MATLAB**

#### Команды общего назначения

disp length

#### Операторы и специальные символы

: . + - \* / ; %

#### Элементарные матрицы и действия над ними

ones pi

#### Функции пакета обработки сигналов

latc2tf poly2rc residue residuez tf2latc zp2sos

**Программа 6\_1**

При помощи функции MATALAB zp2sos данная программа вычисляет коэффициенты *H(z)* путём обращения переходной функции.

% Программа P6\_1

% Обращение рациональной переходной функции в форму коэффициентов

num = [20,30,40];

den= [12,34,45];

[z,p,k]=tf2zp(num,den);

sos=zp2sos(z,p,k)

*Результат исполнения программы*

*sos =*

*1.6667 2.5000 3.3333 1.0000 2.8333 3.7500*

**Программа 6\_2**

Существуют две параллельных формы реализации каузиальной IIR функции преобразования. Первая форма основывается на кусочно-заданном пространстве, зависящем от *z-1*. Она реализуется при помощи функции MATLAB residuez. Вторая же форма зависит от *z* и реализуется функцией MATLAB

% Программа P6\_2

% Реализации в параллельной форме переходной IIR функции

num=input('Вектор коэффициентов числителя: ');

den=input('Вектор коэффициентов знаменателя: ');

[r1,p1,k1]=residuez(num,den);

[r2,p2,k2]=residue(num,den);

disp('Параллельная форма 1')

disp('Остатки: '); disp(r1);

disp('Полюса: '); disp(p1);

disp('Константа: '); disp(k1);

disp('Параллельная форма 2')

disp('Остатки: '); disp(r2);

disp('Полюса: '); disp(p2);

disp('Константа: '); disp(k2);

*Результат исполнения программы*

Enter numinator koefficients:[10, 20, 30]

Enter denuminator koefficients:[12, 34, 45]

Parallel Form 1

Rest:

0.0833 + 0.1736i

0.0833 - 0.1736i

Poles:

-1.4167 + 1.3202i

-1.4167 - 1.3202i

Const:

0.6667

Parallel Form 2

Rest:

-0.3472 - 0.1359i

-0.3472 + 0.1359i

Poles:

-1.4167 + 1.3202i

-1.4167 - 1.3202i

Const:

0.8333

**Программа 6\_3**

Здесь реализуется алгоритм Gray-Markel. Решетчатая каскадная реализация Gray-Markel переходной функции H(z) N-ого порядка, основывается на решетчатой каскадной реализации промежуточной переходной функции *AN(z)*, имеющей тот же знаменатель, что и функция H(z).

% Программа P6\_3

% Решетчатая каскадная структура Gray-Markel

% k - вектор параметров решётки

% alpha - вектор множителей прямой связи

format long

% Запрос коэффициентов

num=input('Вектор коэффициентов числителя: ');

den=input('Вектор коэффициентов знаменателя: ');

N=length(den)-1; % Порядок (степень) многочлена знаменателя

k=ones(1,N);

a1=den/den(1);

alpha=num(N+1:-1:1)/den(1);

for ii=N:-1:1,

alpha(N+2-ii:N+1)=alpha(N+2-ii:N+1)-alpha(N-ii+1)\*a1(2:ii+1);

k(ii)=a1(ii+1);

a1(1:ii+1)=(a1(1:ii+1)-k(ii)\*a1(ii+1:-1:1))/(1-k(ii)\*k(ii));

end

disp('Параметры решётки: '); disp(k)

disp('Множители прямой связи: '); disp(alpha)

subplot(1,2,1); title('Параметры решётки');

bar(k);

subplot(1,2,2); title('Множители прямой связи');

bar(alpha);

*Результат исполнения программы:*

Enter numinator koefficients:[10, 20, 30]

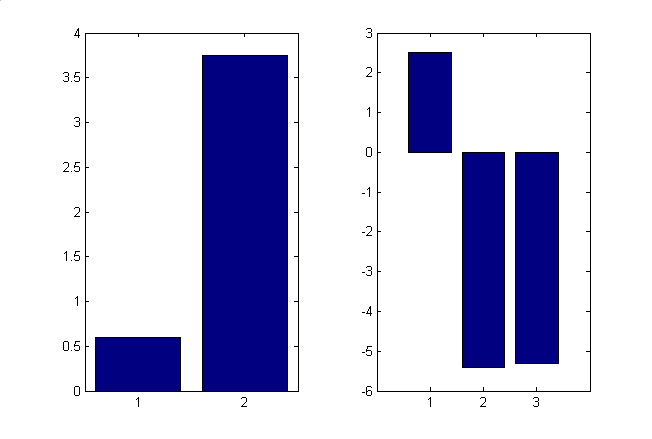
Enter denuminator koefficients:[12, 34, 45]

Grid Parametrs:

0.59649122807018 3.75000000000000

Mults of direct connection:

1. -5.41666666666667 -5.31067251461988



# Вывод:

В данной лабораторной работе была рассмотрена cтруктуры цифровых фильтров. Так же были изучено как строить различные структуры фильтров в зависимости от желаемых свойств будущей структуры.