

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ТЕЛЕКОММУНИКАЦИЙ
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ ПО ДИСЦИПЛИНЕ «Цифровая обработка сигналов»
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Выполнил студент группы ПМ-51 _____ Кислицын И.К.
Проверил преподаватель кафедры РЭС _____ Лесников В.А.

Цель работы:

Синтез и моделирование цепной и параллельной форм ЦФ на основе входных данных ПФ Чебышёва из Л.Р. 1.

Данные:

Коэффициенты a_i .

a_0	a_1	a_2
1.0	-2.0	1.0
1.0	2.0	1.0
1.0	-2.0	1.0
1.0	2.0	1.0

Коэффициенты b_i .

b_1	b_2
-0.5255105661730298	-0.9282002484842149
1.553115895306827	-0.9541037161358508
0.1802644442576899	-0.8012100199005483
1.1158326558729976	-0.8379957820342729

Коэффициенты $Scale_i$.

$Scale$
0.40970269370333545
0.6092773811532036
0.11066419565223808
0.423055206452789

Листинг:

```
#Внесение исходных данных
restart;
with(PolynomialTools):with(plots):
numElements := 4: # количество звеньев в цепи
degElement := 2: # степень звена
deg := numElements * degElement: # степень фильтра
a := [[1.0, -2.0, 1.0],
[1.0, 2.0, 1.0],
[1.0, -2.0, 1.0],
[1.0, 2.0, 1.0]]:
```

```

b := [[1, 0.5255105661730298, 0.9282002484842149],
[1, -1.553115895306827, 0.9541037161358508],
[1, -0.1802644442576899, 0.8012100199005483],
[1, -1.1158326558729976, 0.8379957820342729]]:

scale := [0.40970269370333545,
0.6092773811532036,
0.11066419565223808,
0.423055206452789]:

zeros := [1.0 ,
1.0,
1.0,
1.0,
-1.0,
-1.0,
-1.0,
-1.0]:

poles := [-0.2627552830865149 + I * 0.9269087925434414,
-0.2627552830865148 - I * 0.9269087925434415,
0.7765579476534135 - I * 0.5925044051078179,
0.7765579476534137 + I * 0.592504405107818,
0.09013222212884495 + I * 0.8905538739653346,
0.09013222212884495 - I * 0.8905538739653346,
0.5579163279364988 - I * 0.7257583296498401,
0.5579163279364987 + I * 0.7257583296498401]:

# Построение коэффициентов для параллельной формы
for k from 1 to deg do
  coeffC[k] := 1;
  for j from 1 to deg do
    coeffC[k] := coeffC[k] * (poles[k] - zeros[j]);
    if j <> k then coeffC[k] := coeffC[k] / (poles[k] - poles[j]); fi;
  od;
od:
parElnumer := Array(1..numElements):
parEldenom := Array(1..numElements):
testPar := 0:
for k from 1 to numElements do
  temp := coeffC[2*k - 1] / (z - poles[2*k - 1]) + coeffC[2*k] / (z - poles[2*k]);
  testPar := testPar + temp;
  parElnumer[k] := expand(numer(temp));
  parEldenom[k] := expand(denom(temp));
  parCoeffB0 := Re(coeff(parEldenom[k], z, degElement));
  for degr from 0 to degElement do
    parCoeffA[k, degr] := Re(coeff(parElnumer[k], z, degElement - degr)) / parCoeffB0;
    parCoeffB[k, degr] := Re(coeff(parEldenom[k], z, degElement - degr)) / parCoeffB0;
  od;
od:

# Заполнение начальных данных для синтеза
numValues := 200: # количество отсчётов
x := Array(1..numValues + deg):
printX := Array(1..numValues + deg):
x[deg+1] := 1:
for k from 1 to numValues + deg do printX[k] := k: od:

#Main calculations for chain form
y_chain := Array(1..numValues + deg):
trX := Array(0..numElements, 1..numValues + deg):
for k from 1 to numValues + deg do trX[0,k] := x[k]; od:
w := Array(1..numElements, 1..numValues + deg):
for k from deg + 1 to numValues + deg do
  for element from 1 to numElements do
    localSumW := 0:
    localSumY := 0:
    for shift from 2 to degElement + 1 do
      localSumW := localSumW - b[element][shift] * w[element, k - shift+1];
      localSumY := localSumY + a[element][shift] * w[element, k - shift+1];
    od;
  od;
od:

```

```

od;
localSumW := localSumW + trX[element - 1, k];
w[element, k] := localSumW;
localSumY := localSumY + a[element][1] * w[element, k];
trX[element, k] := localSumY * scale[element];
od;
y_chain[k] := trX[numElements, k];
od;
#Main calculations for parallel form
y_parallel := Array(1..numValues + deg):
w := Array(1..numElements, 1..numValues + deg):
for k from deg + 1 to numValues + deg do
  totalSum := coeffA[0] * x[k];
  for element from 1 to numElements do
    localSumW := 0:
    localSumY := 0:
    for shift from 1 to degElement do
      localSumW := localSumW - parCoeffB[element,shift] * w[element, k - shift];
      localSumY := localSumY + parCoeffA[element,shift] * coeffA[0] * w[element, k - shift];
    od;
    localSumW := localSumW + x[k];
    w[element, k] := localSumW;
    localSumY := localSumY + parCoeffA[element,0] * coeffA[0] * w[element, k];
    totalSum := totalSum + localSumY;
  od;
  y_parallel[k] := totalSum;
od;

# Построение результата:
plot([seq([printX[i],y_transposed[i]],i=deg..numValues)],style=line);
plot([seq([printX[i],y_transposed_canonical[i]],i=deg..numValues)],style=line);

```

Выводы

Весовые функции, соответствующие данному полиному (для цепной и параллельной форм) приведены ниже и совпадают друг с другом и с результатами, полученными в Л.Р. 3-4.

