Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Лабораторная работа №3

Цифровая фильтрация

Выполнил
студент гр. в3530904/00030 В.С. Баганов

Руководитель
доцент, к.т.н. В.С. Тутыгин

«_____» _____ 202__ г.

Санкт-Петербург
2023

Содержание

1. Цель работы. Цифровая фильтрация

Цель данной работы — исследовать зависимость погрешности сигнала на выходе фильтра и коэффициент подавления шума от величины СКО шума на входе и от вида фильтра, найти оптимальное значение полосы пропускания каждого фильтра.

Сравнить эффективность типовых цифровых фильтров оптимального фильтра Колмогорова-Винера.

2. Программа работы

1. Определить зависимости погрешность заданного преподавателем сигнала на выходе фильтра от полосы пропускания фильтра при нескольких (от 0.1 до 0.5) значениях СКО шума ξ на входе.

2. Для каждого значения СКО шума ξ найти оптимальное значение полосы пропускания фильтра Нопт, при котором погрешность сигнала на выходе фильтра минимальна δопт=δ(Нопт).

3. Построить график зависимости δопт =f(ξ).

4. Определить зависимости погрешность заданного преподавателем сигнала на выходе оптимального фильтра Колмогорова-Винера при нескольких (от 0.1 до 0.5) значениях СКО шума ξ на входе.

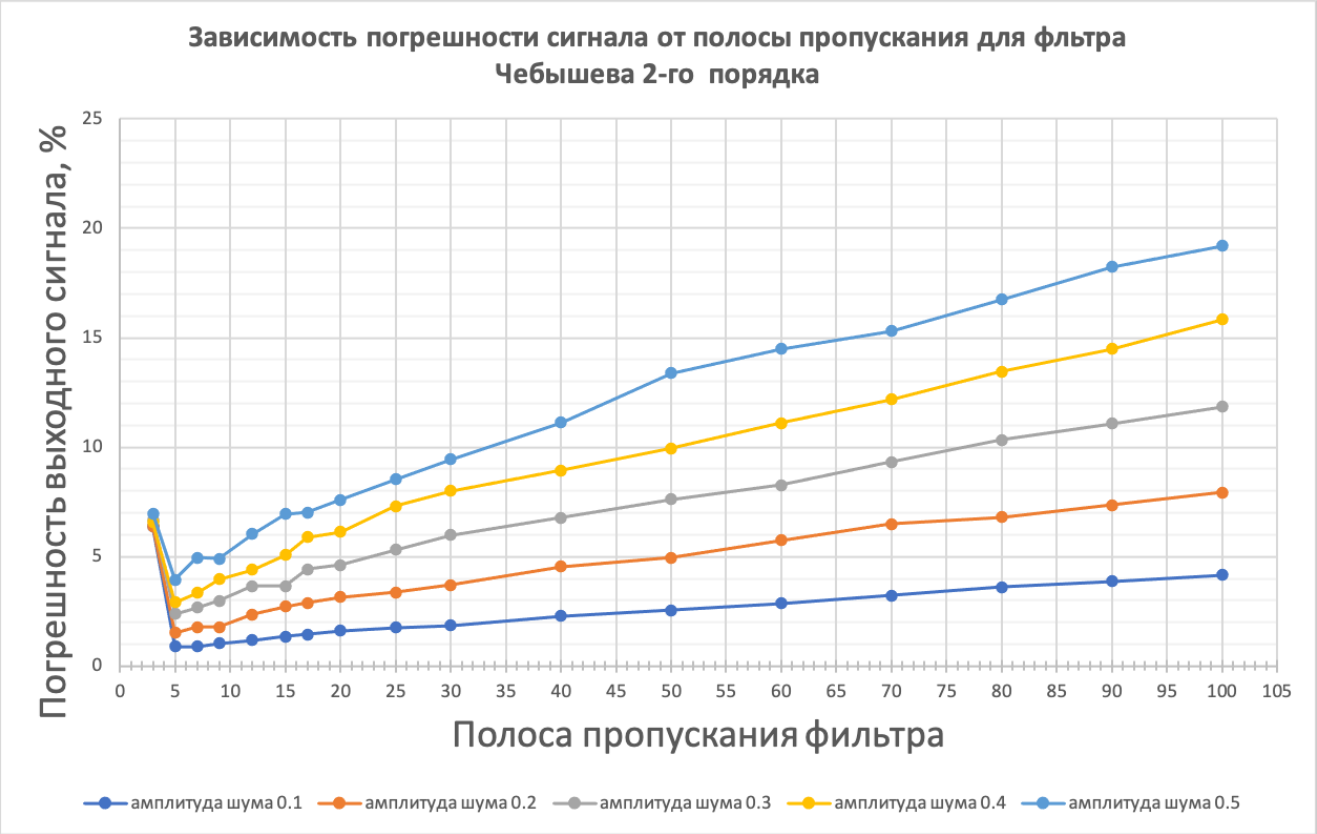
5. Построить график зависимости δ=f(ξ) для оптимального фильтра Колмогорова-Винера.

6. Построить графики зависимости δопт =f(ξ) для всех фильтров.

3. Результаты работы

3.1. График зависимости погрешности заданного сигнала от полосы пропускания фильтра.

График зависимости погрешности заданного сигнала от полосы пропускания фильтра Чебышева 2-го порядка. Поиск оптимального значения полосы пропускания для фильтра Чебышева 2-го порядка.

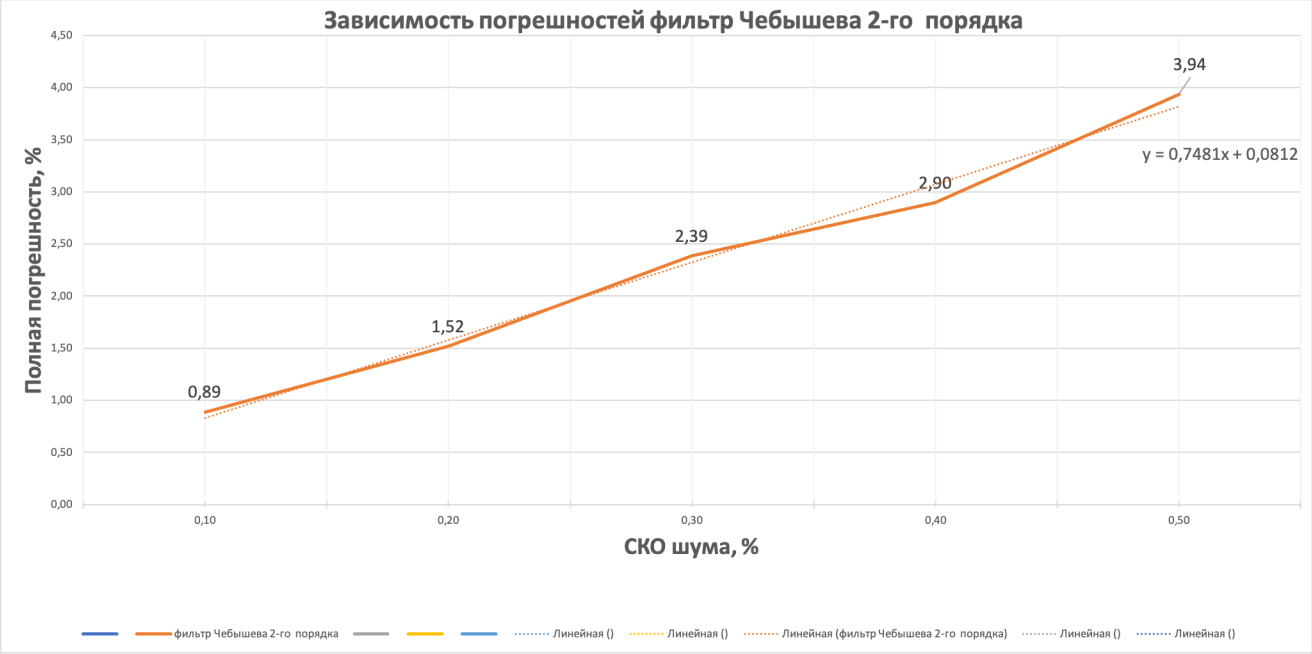


По причине того, что при замерах значения имели большой разброс, было принято решение взять среднее значение из 5 замеров. Средние значения представление на графике и в таблице. Все данные которые были собраны представлены в приложении к лаб. работы.

полоса пропускания фильтра	**амплитуда шума 0.1**	**амплитуда шума 0.2**	**амплитуда шума 0.3**	**амплитуда шума 0.4**	**амплитуда шума 0.5**
3	6,37846	6,39294	6,69086	6,56996	6,96022
5	0,88894	1,51912	2,38604	2,8963	3,93874
7	0,88686	1,77064	2,65772	3,33924	4,95502
9	1,02946	1,773	2,9914	3,96834	4,90074
12	1,16744	2,35542	3,65324	4,4056	6,02618
15	1,3465	2,72618	3,6599	5,08046	6,94166
17	1,43944	2,88328	4,42068	5,87438	7,0148
20	1,60118	3,155	4,60452	6,13472	7,59428
25	1,76328	3,3604	5,3081	7,30604	8,53872
30	1,85652	3,69608	5,99404	8,00252	9,44744
40	2,27224	4,53006	6,78664	8,929	11,126
50	2,54538	4,94024	7,62508	9,93914	13,38975
60	2,85504	5,73652	8,27524	11,0943	14,49536
70	3,22336	6,48764	9,31678	12,19116	15,29214
80	3,60332	6,80444	10,32558	13,44962	16,73168
90	3,85994	7,36166	11,0764	14,48628	18,24278
100	4,14688	7,92984	11,8411	15,82372	19,18932

Таблица оптимальных значений полосы пропускания фильтра Нопт, при которых погрешность сигнала на выходе фильтра минимальна.

полоса пропускания	амплитуда шума 0.1	амплитуда шума 0.2	амплитуда шума 0.3	амплитуда шума 0.4	амплитуда шума 0.5
5	0,88894	1,51912	2,38604	2,8963	3,93874

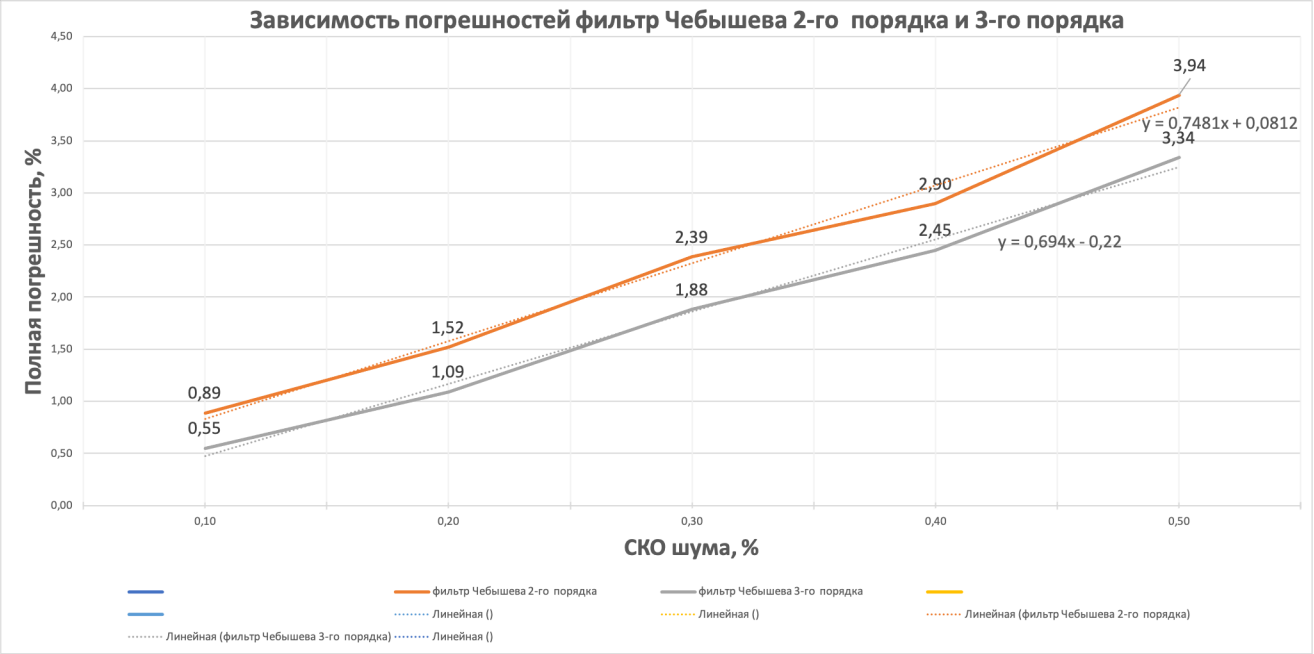


3.2. График зависимости δ=f(ξ) фильтра Чебышева 2-го и 3-го порядка

Таблица зависимости погрешностей заданного преподавателем сигнала на выходе филь-

тра Чебышева 2-го порядка и Чебышева 3-го порядка при нескольких (от 0.1 до 0.5) значениях СКО шума ξ на входе.

Фильтр\шум	**0,10**	**0,20**	**0,30**	**0,40**	**0,50**
фильтр Чебышева 2-го порядка	0,89	1,52	2,39	2,90	3,94
фильтр Чебышева 3-го порядка	0,55	1,09	1,88	2,45	3,34



3.3. График зависимости δ=f(ξ) фильтра Баттерворта

Таблица зависимости погрешностей заданного преподавателем сигнала на выходе фильтра Баттерворта при нескольких (от 0.1 до 0.5) значениях СКО шума ξ на входе.

Фильтр\шум	**0,10**	**0,20**	**0,30**	**0,40**	**0,50**
фильтр Баттерворта	0,90	1,60	2,40	3,20	4,00

3.4. График зависимости δ=f(ξ) Низкочастотного фильтра

Таблица зависимости погрешностей заданного преподавателем сигнала на выходе Низкочастотного фильтра при нескольких (от 0.1 до 0.5) значениях СКО шума ξ на входе.

Фильтр\шум	**0,10**	**0,20**	**0,30**	**0,40**	**0,50**
Низкочастотный фильтр	2,10	4,00	5,30	6,50	7,50

3.5. График зависимости δ=f(ξ) для оптимального фильтра Колмогоров Винера.

Таблица зависимости погрешностей заданного преподавателем сигнала на выходе опти-

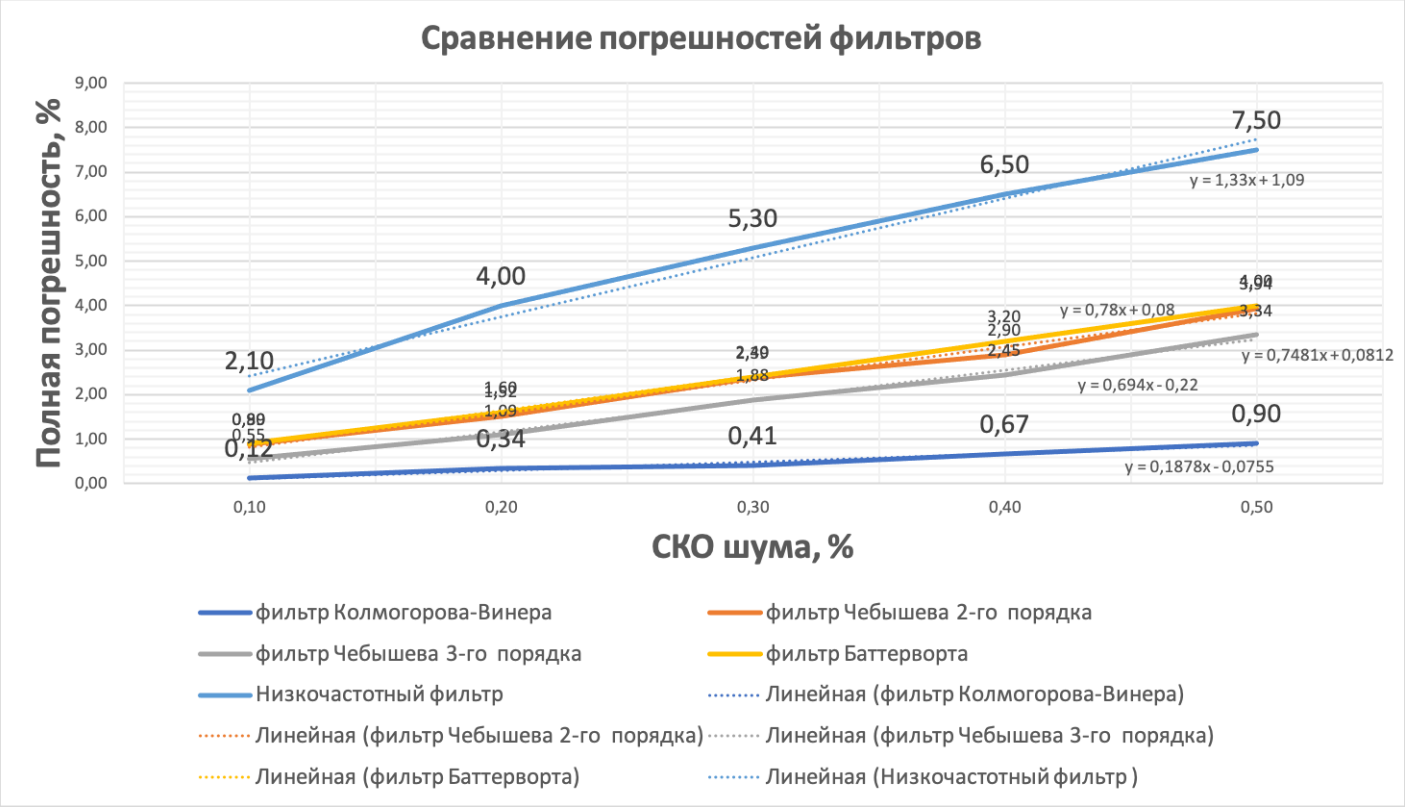
мального фильтра Колмогорова-Винера при нескольких (от 0.1 до 0.5) значениях СКО шума ξ на выходе.

Фильтр\шум	**0,10**	**0,20**	**0,30**	**0,40**	**0,50**
фильтр Колмогорова-Винера	0,12	0,34	0,41	0,67	0,90

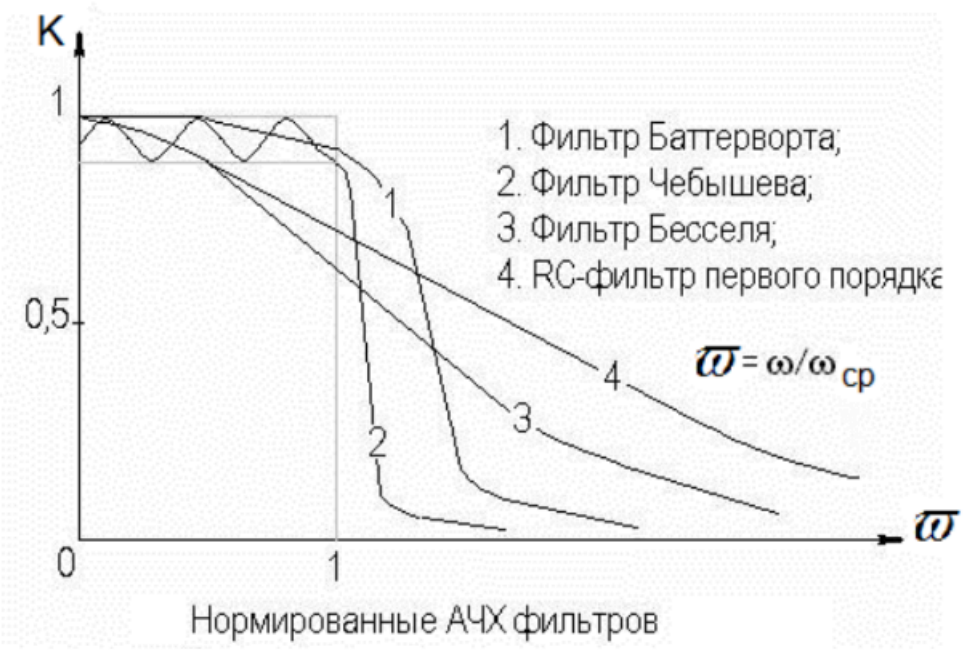
4. Графики зависимости δопт =f(ξ) для всех фильтров.

График зависимости погрешности фильтра Колмогорова-Винера при нескольких (от 0.1 до 0.5) значениях СКО шума. Построение графиков зависимости для различных фильтров.

Фильтр\СКО шума	**0,10**	**0,20**	**0,30**	**0,40**	**0,50**
фильтр Колмогорова-Винера	0,12	0,34	0,41	0,67	0,90
фильтр Чебышева 2-го порядка	0,89	1,52	2,39	2,90	3,94
фильтр Чебышева 3-го порядка	0,55	1,09	1,88	2,45	3,34
фильтр Баттерворта	0,90	1,60	2,40	3,20	4,00
фильтр Низкочастотный	2,10	4,00	5,30	6,50	7,50



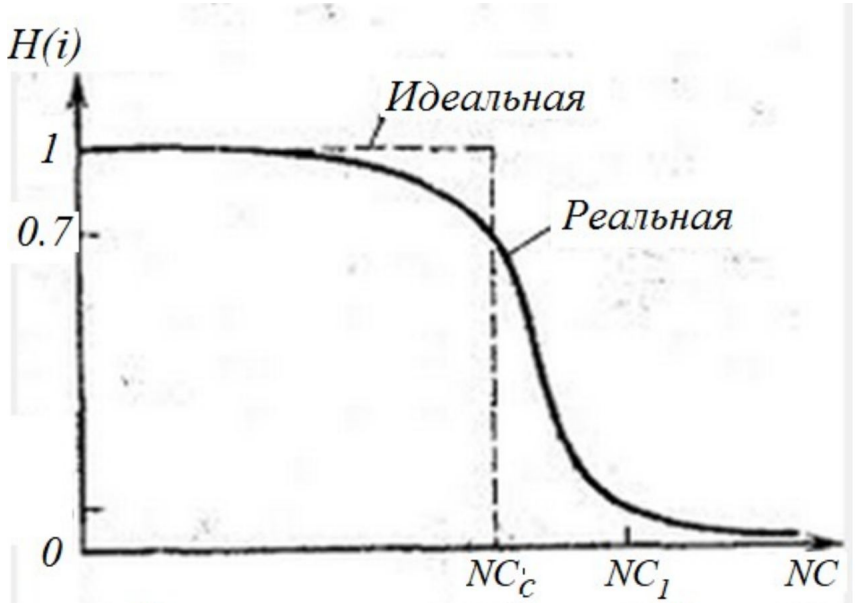
5. Выводы



Нормированные АЧХ фильтров

На графике зависимости погрешности заданного сигнала от полосы пропускания фильтра видно, что фильтр Чебышева 2-го порядка имеет оптимальную полосу пропускания шириной окна 5-7, а дальше начинает сильно увеличиваться погрешность. На картинке выше видно, что фильтры Чебышева и Баттерворта имеют резкий спад в полосе перехода, следовательно у них буду четко выраженные границы оптимальной погрешности на выходе сигнала.

С ростом шума в сигнале растет и погрешность на выходе фитльтра.



NC'_c - количество периодов сигнала, пропускаемых идеальным фильтром без искажений

При фильтрации заданного, гармонического, сигнала самым плохим (с большими погрешностями) оказался Низкочастотный фильтр.

Фильтр Чебышева 2-го порядка по величине минимальной погрешности уступает только фильтру Колмогорова-Виннера. Фильтр Чебышева 3-го порядка чуть лучше 2-го порядка. Можно предположить, что Фильтр Чебышева N-го порядка будет снижать минимальную погрешность, но не сильно.

При фильтрации заданного, гармонического, сигнала самым эффективным оказался фильтр Колмогорова-Виннера.

5.1. Листинг Matlab. Фильтр Чебышёва (Chebishev)

```
%Фильтр Чебышева
clc;
clear;
A=1; %амплитуда сигнала
Q=0.1; %амплитуда шума 0.1 вместо 0.001
KP1=5;% - количество периодов первого сигнала
% KP2=15;% - количество периодов второго сигнала
KP2=15;% - количество периодов второго сигнала
% N=1024;%количество точек расчета
N=1024;
NC=9; %полоса пропускания фильтра по уровню 0,7 амплитуды
% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром
% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде
e=0.1;%параметр фильтра Чебышева
q=randn(1,N); %генерация одномерного массива нормально распределенного
 ↪  шума
for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/N);
% x1(k)=(2*pi*(((k-1-N/2))/N));%%%%%%%%
% s(k)=abs(sin(x1(k)));
x(k)=s(k)+Q*q(k); % суммирование сигнала и шума
end
figure
plot(x);
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Kmax
title('Зашумленный сигнал до фильтра');
Y=fft(x,N); %БПФ сигнала с шумом
i=1:N/2;
figure
% semilogy(i(1:200),2*abs(SS1(1:200)));
%plot(i(1:100),2*abs(SS1(1:100)));
plot(i(1:N/2),abs(Y(1:N/2)));
title('Частотный спектр сигнала с шумом');

for i=1:N;
Tn=2*(i/NC)^2-1;%значение полинома Чебышёва 1 рода 2-го порядка
%  Tn=4*(i/NC)^3-3*(i/NC);%значение полинома Чебышёва 3-го порядка
% Tn=8*(i/NC)^4-8*(i/NC)^2+1;%значение полинома Чебышёва 4-го порядка
% Tn=16*(i/NC)^5-20*(i/NC)^3+5*(i/NC);%значение полинома Чебышёва 5-го
 ↪  порядка
H(i)=1/sqrt(1+e^2*Tn^2);%частотная характеристика фильтра Чебышева
end
h=ifft(H);
 i=1:N;
 plot(i(1:20),abs(h(1:20))); %Импульсная характеристика фильтра
 title('Импульсная характеристика фильтра');
i=1:200;
figure
plot(i,abs(H(1:200)));
%semilogx(i,abs(H(1:200)));%то же, что и plot, но в логарифмическом
%масштабе по X
grid on;
title('Частотная хар-ка цифрового фильтра Чебышева');

i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
```

```
XX2=fft(s,N);%частотный спектр сигнала
Z2=ifft(XX2.*H);%свертка незашумленного сигнала с частотной хар-кой фильтра
DZ(i)=(2*real(Z(i))-2*real(Z2(i)))*100./(max(s)-min(s));%случайная погрешность
DZ1(i)=(2*real(Z(i))-s(i))*100/(max(s)-min(s));%полная погрешность
SKO_random=std(DZ)
SKO_total=std(DZ1)
i=1:N;
yy=A*sin((6.28*KP1*i/N));
figure
plot(i,x); %вывод сигнала до фильтра
title('Сигнал до фильтра');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Амплитуда'); % подпись по оси Y
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
hold on; % "удержание" окна вывода для вывода следующего графика

i=1:N;
plot(i,2*real(Z(1:N)),'r-'),grid; %вывод отфильтрованного сигнала
%представление графика линией красного цвета, отображение сетки
title('Сигнал до и после фильтра');%подпись названия графика
hold off;

i=1:N;
figure
plot(i,DZ(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Случайная погрешность отфильтрованного сигнала');
ylabel('Случайная погрешность, %'); % подпись по оси Y
axis tight;
i=1:N;
figure
plot(i,DZ1(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Полная погрешность отфильтрованного сигнала');
ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;

pause;
close all;
```

5.2. Листинг Matlab. Оптимальный фильтр Колмогорова-Винера (KolmVinfilter)

```
%Фильтр Колмогорова-Винера

A=1; %амплитуда сигнала
Q=0.5; %СКО шума
N=1024;%количество точек расчета
% kp1=5;%количество переиодов сигнала
kp1=5;%количество переиодов сигнала
clc;
q=Q*randn(1,N);%генерация одномерного массива нормально распределенного
 ↪  шума с СКО=Q
for k=1:N %цикл вычисления зашумленного сигнала
%s1(k)=A*exp(-0.0003*(k-200)^2.0); %колоколообразный сигнал
s(k)=A*sin(2*pi*kp1*k/N);%+A*sin(2*pi*kp2*k/N);%гармонический сигнал
% x1(k)=(2*pi*(((k-1-N/2))/N));%%%%%%%
%s(k)=abs(sin(x1(k)));
% s(k)=x1(k)*exp(x1(k));
% s1(k)=0; % сигнал прямоугольной формы
% if (k>100)&(k<300) % сигнал прямоугольной формы
% s1(k)=A;
% end
x(k)=s(k)+q(k); % суммирование сигнала и шума
end
%x=s-mean(s)+q;%центрирование сигнала
figure
plot(x(1:N));
title('Зашумленный сигнал до фильтра');
axis tight;

Y=fft(x,N)/N; %БПФ сигнала с шумом
SS1=Y.*conj(Y)/N; %спектр мощности
i=1:200;
figure
%plot(i,SS1(1:200));
semilogy(i,SS1(1:200)); %вывод спектра мощности сигнала с шумом
title('Частотный спектр сигнала с шумом');

Y=fft(s,N)/N; %БПФ сигнала без шума
SS1=Y.*conj(Y)/N; %спектр мощности сигнала без шума

Y1=fft(q,N)/N; %БПФ шума
SS2=Y1.*conj(Y1)/N; %спектр мощности шума

for i=1:N
H(i)=SS1(i)/(SS1(i)+SS2(i));%частотная характеристика оптимального
 ↪  фильтра
end
i=1:200;
figure
%plot(i,abs(H(1:200)));
semilogx(i,abs(H(1:200)));
%hold on
title('Частотная характеристика оптимального фильтра');

i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
```

```
55  axis tight;
56
57  figure
58  plot(i,s(1:N)); %вывод незашумленного сигнала до фильтра сигнала
59  title('Незашумленный сигнал до фильтра');
60  axis tight;
61  figure
62  plot(i,Z(1:N)); %вывод отфильтрованного сигнала
63  title('Сигнал после свертки с част. хар-кой опт. фильтра');
64  axis tight;
65  figure
66  plot(i,x); %вывод сигнала до фильтра
67  title('Сигнал до фильтра');
68  xlabel('Номер отсчета'); % подпись по оси X
69  ylabel('Амплитуда'); % подпись по оси Y
70  axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
71  hold on; % "удержание" окна вывода для вывода следующего графика
72
73  i=1:N;
74  plot(i,real(Z(1:N)),'r-'),grid; %вывод отфильтрованного сигнала
75  %представление графика линией красного цвета, отображение сетки
76  title('Сигнал до и после фильтра');%подпись названия графика
77  hold off;
78  i=1:N;
79  DZ(i)=Z(i)-s(i);
80  DZ1=DZ*100/(max(s)-min(s));
81  SKO_total=std(DZ1)
82
83  i=1:N;
84  figure
85  plot(i,DZ1(1:N)); %вывод погрешности отфильтрованного сигнала
86  title('Погрешность отфильтрованного сигнала');
87  ylabel('Полная погрешность, %'); % подпись по оси Y
88  axis tight;
89
90  pause;
91  close all;
92  clear;
```

5.3. Листинг Matlab. Фильтр скользящего среднего (Smoothing)

```
%Фильтр медианный и скользящего среднего
%выбор одного из двух фильтров - в строках 23 и 24
clc;
A=1; %амплитуда сигнала
Q=0.5 ; %СКО шума
KP1=5;% - количество периодов первого сигнала
%KP2=15;% - количество периодов второго сигнала
N=1024;%количество точек расчета
W=3;%ширина окна сглаживания

q=Q*randn(1,N); %генерация одномерного массива шума с СКО=Q
H=(W+1)/2;%вычисление полуширины окна сглаживания
for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);
x(k)=(2*pi*(((k-1-N/2))/N));%%%%%%%
s(k)=x(k)*exp(x(k));
% s(k)=exp(x(k)); %%%%%%%%
x(k)=s(k)+q(k); % суммирование сигнала и шума
end;

for i=1:N-W %сглаживание зашумленного сигнала
for j=1:W
z(j)=x(j+i-1);
end
%y(i-1+H)=median(z);%вычисление медианы в скользящем окне
y(i-1+H)=mean(z); %вычисление скользящего среднего
end
i=H:N-H;
figure
plot(i,y(i));
title('Сигнал после сглаживания');
for i=H:N-H %H - половина ширины окна сглаживания +0.5
DZ(i)=s(i)-y(i);%уровень зашумления в сигнале после фильтра
end
DZ=DZ*100/(max(s)-min(s));%полная погрешность в %
SKO_total=std(DZ)

i=H:N-H;
figure;
plot(x);
axis tight;
xlabel('Номер отсчета'); % подпись по оси X
%hold on;
title('Зашумленный сигнал до сглаживания');
figure
plot(i,y(H:N-H));
title('Сигнал после сглаживания');
xlabel('Номер отсчета'); % подпись по оси X
axis tight;
%hold off;
figure
plot(i,DZ(H:N-H)); %вывод погрешности отфильтрованного сигнала
axis tight;
title('Погрешность отфильтрованного сигнала');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Полная погрешность, %'); % подпись по оси Y

```

```
58 pause;
59 close all;
60 clear;
```

5.4. Листинг Matlab. Медианный фильтр (Median)

```
%Фильтр медианный и скользящего среднего
%выбор одного из двух фильтров - в строках 23 и 24
clc;
A=1; %амплитуда сигнала
Q=0.5 ; %СКО шума
KP1=5;% - количество периодов первого сигнала
%KP2=15;% - количество периодов второго сигнала
N=1024;%количество точек расчета
W=3;%ширина окна сглаживания

q=Q*randn(1,N); %генерация одномерного массива шума с СКО=Q
H=(W+1)/2;%вычисление полуширины окна сглаживания
for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);
x(k)=(2*pi*(((k-1-N/2))/N));%%%%%%%
s(k)=x(k)*exp(x(k));
% s(k)=exp(x(k)); %%%%%%%%
x(k)=s(k)+q(k); % суммирование сигнала и шума
end;

for i=1:N-W %сглаживание зашумленного сигнала
for j=1:W
z(j)=x(j+i-1);
end
%y(i-1+H)=median(z);%вычисление медианы в скользящем окне
y(i-1+H)=mean(z); %вычисление скользящего среднего
end
i=H:N-H;
figure
plot(i,y(i));
title('Сигнал после сглаживания');
for i=H:N-H %H - половина ширины окна сглаживания +0.5
DZ(i)=s(i)-y(i);%уровень зашумления в сигнале после фильтра
end
DZ=DZ*100/(max(s)-min(s));%полная погрешность в %
SKO_total=std(DZ)

i=H:N-H;
figure;
plot(x);
axis tight;
xlabel('Номер отсчета'); % подпись по оси X
%hold on;
title('Зашумленный сигнал до сглаживания');
figure
plot(i,y(H:N-H));
title('Сигнал после сглаживания');
xlabel('Номер отсчета'); % подпись по оси X
axis tight;
%hold off;
figure
plot(i,DZ(H:N-H)); %вывод погрешности отфильтрованного сигнала
axis tight;
title('Погрешность отфильтрованного сигнала');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Полная погрешность, %'); % подпись по оси Y

```

```
58 pause;
59 close all;
60 clear;
```

5.5. Листинг Matlab. Фильтр Чебышёва инверсный (Inv_Chebishev)

```
%Фильтр Чебышёва инверсный
clc;
clear;
A=1; %амплитуда сигнала
Q=0.1; %амплитуда шума 0.1 вместо 0.001
% KP1=5;% - количество периодов первого сигнала
KP1=5;% - количество периодов первого сигнала
%KP2=2;% - количество периодов второго сигнала
KP2=15;% - количество периодов второго сигнала
N=1024;%количество точек расчета
NC=15;
%NC - полоса пропускания фильтра по уровню 0,7 амплитуды
% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром
% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде
e=0.1;%параметр фильтра Чебышёва %вместо 0.001
q=Q*randn(1,N); %генерация одномерного массива нормально распределенного
 ↪  шума с СКО=Q
for k=1:N % генерация сигнала и шума
s(k) = A*sin(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/N);
x(k)=s(k)+q(k); % суммирование сигнала и шума
end
figure
plot(x);
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Kmax
title('Зашумленный сигнал до фильтра');

Y=fft(x,N); %БПФ сигнала с шумом
i=1:N/2;
figure
% semilogy(i(1:200),2*abs(SS1(1:200)));
%plot(i(1:100),2*abs(SS1(1:100)));
plot(i(1:N/2),abs(Y(1:N/2)));
title('Частотный спектр сигнала с шумом');

for i=1:N;
Tn=2*(NC/i)^2-1;%значение полинома Чебышёва 1 рода 2-го порядка
%H(i)=1/sqrt(1+e^2*Tn^2);%частотная характеристика фильтра Чебышёва
H(i)=sqrt((e^2*Tn^2)/((1+e^2*Tn^2)));%частотная характеристика фильтра
 ↪  Чебышёва инверсный
end
h=ifft(H);
i=1:N;
plot(i(1:20),abs(h(1:20))); %Импульсная характеристика фильтра
title('Импульсная характеристика фильтра');
i=1:200;
figure
plot(i,abs(H(1:200)));
% semilogx(i,abs(H(1:200)));%то же, что и plot, но в логарифмическом
%масштабе по X
grid on;
title('Частотная хар-ка цифрового фильтра инверсного Чебышёва');

i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
XX2=fft(s,N);%частотный спектр сигнала
```

```
Z2=ifft(XX2.*H);%свертка незашумленного сигнала с частотной хар-кой
 ↪  фильтра
DZ(i)=(2*real(Z(i))-2*real(Z2(i)))*100./(max(s)-min(s));%случайная
 ↪  погрешность
DZ1(i)=(2*real(Z(i))-s(i))*100./(max(s)-min(s));%полная погрешность
SKO_random=std(DZ)
SKO_total=std(DZ1)
i=1:N;
yy=A*sin((6.28*KP1*i/N));
figure
plot(i,x); %вывод сигнала до фильтра
title('Сигнал до фильтра');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Амплитуда'); % подпись по оси Y
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
hold on; % "удержание" окна вывода для вывода следующего графика

i=1:N;
plot(i,2*real(Z(1:N)),'r-'),grid; %вывод отфильтрованного сигнала
%представление графика линией красного цвета, отображение сетки
title('Сигнал до и после фильтра');%подпись названия графика
hold off;

i=1:N;
figure
plot(i,DZ(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Случайная погрешность отфильтрованного сигнала');
ylabel('Случайная погрешность, %'); % подпись по оси Y
axis tight;
i=1:N;
figure
plot(i,DZ1(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Полная погрешность отфильтрованного сигнала');
ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;

pause;
close all;
```

5.6. Листинг Matlab. Линейный низкочастотный фильтр (Lowpass filter)

```
%Низкочастотный фильтр 1-го порядка
clc;
A=1; %амплитуда сигнала
Q=0.5; %СКО шума
KP1=5; % - количество периодов первого сигнала
KP2=15;% количество периодов второго сигнала
N=1024;%количество точек расчета
NC=6;% полоса пропускания фильтра по уровню 0,7 8амплитуды
% выражена в количестве отчетов спектра БПФ, пропускаемых фильтром
% остальные отсчеты (в частотном спектре!) будут ослабляться по амплитуде
q=Q*randn(1,N); %генерация одномерного массива шума. Q-СКО.
for k=1:N % генерация сигнала и шума
s(k) = A*cos(2*pi*KP1*k/N);%+ A*sin(2*pi*KP2*k/N);
x(k)=s(k)+q(k); % суммирование сигнала и шума
end
figure
plot(x);
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Kmax
title('Зашумленный сигнал до фильтра');

Y=fft(x,N); %БПФ сигнала с шумом
i=1:N;
figure
%plot(i(1:200),2*abs(Y(1:200)));
semilogy(i(1:200),2*abs(Y(1:200)));%то же, что и plot, но в
 ↪  логарифмическом
%масштабе по X
title('Частотный спектр сигнала с шумом');
xlabel('Номер элемента массива частотного спектра');
%pause;

for i=1:N
H(i)=1/((1+i/(NC))); %передаточная функция цифрового фильтра НЧ 1-го
 ↪  порядка
%в частотной области
end
h=ifft(H);
i=1:N;
plot(i(1:20),abs(h(1:20))); %Импульсная характеристика фильтра
title('Импульсная характеристика фильтра');
i=1:200;
figure
%plot(i,abs(H(1:200)));
semilogx(i,abs(H(1:200)));%то же, что и plot, но в логарифмическом
%масштабе по X
%grid on;
title('Частотная хар-ка цифрового НЧ-фильтра');

i=1:N;
XX1=fft(x,N); %частотный спектр сигнала с шумом
Z=ifft(XX1.*H);%свертка зашумленного сигнала с частотной хар-кой фильтра
XX2=fft(s,N);%частотный спектр сигнала
Z2=ifft(XX2.*H);%свертка незашумленного сигнала с частотной хар-кой
 ↪  фильтра
i=1:N;
```

```
DZ(i)=(2*real(Z(i))-2*real(Z2(i)))*100./(max(s)-min(s));%случайная
 ↪  погрешность
DZ1(i)=(2*real(Z(i))-s(i))*100./(max(s)-min(s));%полная погрешность
SKO_random=std(DZ)
SKO_total=std(DZ1)
i=1:N;
figure
plot(i,x); %вывод сигнала до фильтра
title('Сигнал до фильтра');
xlabel('Номер отсчета'); % подпись по оси X
ylabel('Амплитуда'); % подпись по оси Y
axis tight; %диапазон X и Y по осям точно соответствует Xmax и Ymax
hold on; % "удержание" окна вывода для вывода следующего графика
plot(i,2*real(Z(1:N)),'r-'),grid; %вывод отфильтрованного сигнала
%представление графика линией красного цвета, отображение сетки
title('Сигнал до и после фильтра');%подпись названия графика
hold off;

i=1:N;
figure
plot(i,DZ(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Случайная погрешность отфильтрованного сигнала');
ylabel('Случайная погрешность, %'); % подпись по оси Y
axis tight;
i=1:N;
figure
plot(i,DZ1(1:N)); %вывод случайной погрешности отфильтрованного сигнала
title('Полная погрешность отфильтрованного сигнала');
ylabel('Полная погрешность, %'); % подпись по оси Y
axis tight;

pause;
close all;
clear;
```

6. Приложение. Данные

```
+=========================================+
Q=0.1;
KP1=5;
+=========================================+
NC=3
SKO_random = 0.5032; SKO_total = 6.2747
SKO_random = 0.4521; SKO_total = 6.4159
SKO_random = 0.6375; SKO_total = 6.5473
SKO_random = 0.7399; SKO_total = 6.3272

NC=5
SKO_random = 0.5924; SKO_total = 0.8829
SKO_random = 0.5787; SKO_total = 0.9368
SKO_random = 0.6732; SKO_total = 0.8242
SKO_random = 0.7077; SKO_total = 0.9436
SKO_random = 0.6503; SKO_total = 0.8572

NC=7
SKO_random = 0.8791; SKO_total = 0.8799
SKO_random = 1.0183; SKO_total = 1.0169
```

```
SKO_random = 0.8916; SKO_total = 0.8885
SKO_random = 0.8340; SKO_total = 0.8357
SKO_random = 0.8102; SKO_total = 0.8133

NC=9
SKO_random = 0.9756; SKO_total = 0.9755
SKO_random = 1.0869; SKO_total = 1.0870
SKO_random = 0.9753; SKO_total = 0.9740
SKO_random = 1.0024; SKO_total = 1.0020
SKO_random = 1.1085; SKO_total = 1.1088

NC=12;
SKO_random = 1.2524; SKO_total = 1.2529
SKO_random = 1.2648; SKO_total = 1.2634
SKO_random = 1.1912; SKO_total = 1.1938
SKO_random = 1.1095; SKO_total = 1.1103
SKO_random = 1.0153; SKO_total = 1.0168

NC=15
SKO_random = 1.3925; SKO_total = 1.3979
SKO_random = 1.5698; SKO_total = 1.5667
SKO_random = 1.3205; SKO_total = 1.3230
SKO_random = 1.1742; SKO_total = 1.1805
SKO_random = 1.2694; SKO_total = 1.2644

NC=17
SKO_random = 1.4740; SKO_total = 1.4752
SKO_random = 1.5361; SKO_total = 1.5349
SKO_random = 1.3989; SKO_total = 1.3960
SKO_random = 1.3926; SKO_total = 1.3946
SKO_random = 1.3932; SKO_total = 1.3965

NC=20
SKO_random = 1.5506; SKO_total = 1.5520
SKO_random = 1.7345; SKO_total = 1.7353
SKO_random = 1.5598; SKO_total = 1.5641
SKO_random = 1.4997; SKO_total = 1.5020
SKO_random = 1.6525; SKO_total = 1.6525

NC=25
SKO_random = 1.8340; SKO_total = 1.8347
SKO_random = 1.8371; SKO_total = 1.8357
SKO_random = 1.7727; SKO_total = 1.7756
SKO_random = 1.7441; SKO_total = 1.7451
SKO_random = 1.6237; SKO_total = 1.6253

NC=30
SKO_random = 1.9370; SKO_total = 1.9369
SKO_random = 1.8468; SKO_total = 1.8468
SKO_random = 1.8569; SKO_total = 1.8567
SKO_random = 1.7868; SKO_total = 1.7874
SKO_random = 1.8546; SKO_total = 1.8548

NC=40
SKO_random = 2.3487; SKO_total = 2.3536
SKO_random = 2.2595; SKO_total = 2.2554
SKO_random = 2.2468; SKO_total = 2.2574
SKO_random = 2.1795; SKO_total = 2.1812
SKO_random = 2.3161; SKO_total = 2.3136
```

```
NC=50;
SKO_random = 2.4288; SKO_total = 2.4535
SKO_random = 2.4598; SKO_total = 2.4949
SKO_random = 2.4726; SKO_total = 2.4804
SKO_random = 2.6102; SKO_total = 2.6290
SKO_random = 2.6490; SKO_total = 2.6691

NC=60;
SKO_random = 2.8281; SKO_total = 2.8800
SKO_random = 2.7759; SKO_total = 2.7881
SKO_random = 2.7759; SKO_total = 2.7881
SKO_random = 2.9562; SKO_total = 2.9789
SKO_random = 2.8124; SKO_total = 2.8401

NC=70;
SKO_random = 3.0865; SKO_total = 3.1223
SKO_random = 3.3197; SKO_total = 3.3761
SKO_random = 3.2321; SKO_total = 3.3170
SKO_random = 3.1861; SKO_total = 3.1717
SKO_random = 3.0572; SKO_total = 3.1297

NC=80;
SKO_random = 3.5385; SKO_total = 3.6807
SKO_random = 3.3886; SKO_total = 3.5750
SKO_random = 3.3886; SKO_total = 3.5750
SKO_random = 3.5195; SKO_total = 3.6465
SKO_random = 3.4353; SKO_total = 3.5394

NC=90;
SKO_random = 3.4353; SKO_total = 3.5394
SKO_random = 3.7185; SKO_total = 3.9139
SKO_random = 3.7185; SKO_total = 3.9139
SKO_random = 3.7185; SKO_total = 3.9139
SKO_random = 3.7893; SKO_total = 4.0186

NC=100;
SKO_random = 3.8329; SKO_total = 4.1076
SKO_random = 3.7513; SKO_total = 3.9933
SKO_random = 3.8488; SKO_total = 4.1880
SKO_random = 3.9453; SKO_total = 4.2252
SKO_random = 3.9135; SKO_total = 4.2203



+=========================================+
Q=0.2;
KP1=5;
+=========================================+
NC=3
SKO_random = 1.0063; SKO_total = 6.195
SKO_random = 0.9042; SKO_total = 6.4708
SKO_random = 0.9042; SKO_total = 6.4708
SKO_random = 1.1837; SKO_total = 6.3276
SKO_random = 0.8660; SKO_total = 6.5005

NC=5
SKO_random = 1.3456; SKO_total = 1.5280
SKO_random = 1.2959; SKO_total = 1.4442
SKO_random = 1.3594; SKO_total = 1.4874
SKO_random = 1.2320; SKO_total = 1.4649
```

```
SKO_random = 1.5240; SKO_total = 1.6711

NC=7
SKO_random = 1.9553; SKO_total = 1.9547
SKO_random = 1.7568; SKO_total = 1.7578
SKO_random = 1.9736; SKO_total = 1.9708
SKO_random = 1.6811; SKO_total = 1.6868
SKO_random = 1.4812; SKO_total = 1.4831

NC=9
SKO_random = 1.7197; SKO_total = 1.7189
SKO_random = 1.9628; SKO_total = 1.9626
SKO_random = 1.8458; SKO_total = 1.8461
SKO_random = 1.6701; SKO_total = 1.6687
SKO_random = 1.6701; SKO_total = 1.6687

NC=12;
SKO_random = 2.2480; SKO_total = 2.2469
SKO_random = 2.4375; SKO_total = 2.4368
SKO_random = 2.2269; SKO_total = 2.2293
SKO_random = 2.3729; SKO_total = 2.3704
SKO_random = 2.4933; SKO_total = 2.4937

NC=15
SKO_random = 2.7851; SKO_total = 2.7901
SKO_random = 3.1396; SKO_total = 3.1362
SKO_random = 2.6411; SKO_total = 2.6431
SKO_random = 2.3485; SKO_total = 2.3543
SKO_random = 2.7106; SKO_total = 2.7072

NC=17
SKO_random = 2.8537; SKO_total = 2.8548
SKO_random = 2.7534; SKO_total = 2.7478
SKO_random = 2.9480; SKO_total = 2.9488
SKO_random = 3.0723; SKO_total = 3.0706
SKO_random = 2.7978; SKO_total = 2.7944

NC=20
SKO_random = 3.0198; SKO_total = 3.0213
SKO_random = 3.0560; SKO_total = 3.0586
SKO_random = 3.1013; SKO_total = 3.1022
SKO_random = 3.4691; SKO_total = 3.4694
SKO_random = 3.1196; SKO_total = 3.1235

NC=25
SKO_random = 3.3595; SKO_total = 3.3604
SKO_random = 3.3595; SKO_total = 3.3604
SKO_random = 3.3595; SKO_total = 3.3604
SKO_random = 3.3595; SKO_total = 3.3604
SKO_random = 3.3595; SKO_total = 3.3604

NC=30
SKO_random = 3.6253; SKO_total = 3.6251
SKO_random = 3.8739; SKO_total = 3.8739
SKO_random = 3.6936; SKO_total = 3.6936
SKO_random = 3.7138; SKO_total = 3.7136
SKO_random = 3.5736; SKO_total = 3.5742

NC=40
SKO_random = 4.3812; SKO_total = 4.3862
```

```
SKO_random = 4.6974; SKO_total = 4.7010
SKO_random = 4.6974; SKO_total = 4.7010
SKO_random = 4.4936; SKO_total = 4.5028
SKO_random = 4.3590; SKO_total = 4.3593

NC=50;
SKO_random = 5.2861; SKO_total = 5.2819
SKO_random = 4.6159; SKO_total = 4.6496
SKO_random = 4.8576; SKO_total = 4.8755
SKO_random = 4.9197; SKO_total = 4.9480
SKO_random = 4.9452; SKO_total = 4.9462

NC=60;
SKO_random = 5.8115; SKO_total = 5.8376
SKO_random = 5.8240; SKO_total = 5.8525
SKO_random = 5.6562; SKO_total = 5.6910
SKO_random = 5.5518; SKO_total = 5.5462
SKO_random = 5.7118; SKO_total = 5.7553

NC=70;
SKO_random = 6.4485; SKO_total = 6.4681
SKO_random = 6.4485; SKO_total = 6.4681
SKO_random = 6.6394; SKO_total = 6.6631
SKO_random = 6.4642; SKO_total = 6.5159
SKO_random = 6.3723; SKO_total = 6.3230

NC=80;
SKO_random = 7.0771; SKO_total = 7.1619
SKO_random = 6.5653; SKO_total = 6.6143
SKO_random = 6.4690; SKO_total = 6.4909
SKO_random = 6.7100; SKO_total = 6.8053
SKO_random = 6.8942; SKO_total = 6.9498

NC=90;
SKO_random = 7.3474; SKO_total = 7.4614
SKO_random = 7.2918; SKO_total = 7.4476
SKO_random = 7.1095; SKO_total = 7.2085
SKO_random = 7.3789; SKO_total = 7.4987
SKO_random = 7.0522; SKO_total = 7.1921

NC=100;
SKO_random = 7.6033; SKO_total = 7.7437
SKO_random = 7.6910; SKO_total = 7.8151
SKO_random = 7.6297; SKO_total = 7.7752
SKO_random = 8.2224; SKO_total = 8.3643
SKO_random = 7.8468; SKO_total = 7.9509









+========================================+
Q=0.3;
KP1=5;
+========================================+
NC=3
SKO_random = 1.6469; SKO_total = 6.5428
SKO_random = 1.4005; SKO_total = 6.6001
SKO_random = 1.8501; SKO_total = 6.4216
```

```
SKO_random = 1.2700; SKO_total = 6.8679
SKO_random = 1.9126; SKO_total = 7.0219

NC=5
SKO_random = 2.8795; SKO_total = 2.8613
SKO_random = 2.8795; SKO_total = 2.8613
SKO_random = 1.7360; SKO_total = 1.9775
SKO_random = 2.0196; SKO_total = 2.0064
SKO_random = 2.1232; SKO_total = 2.2237

NC=7
SKO_random = 2.4289; SKO_total = 2.4228
SKO_random = 2.6372; SKO_total = 2.6377
SKO_random = 3.0550; SKO_total = 3.0533
SKO_random = 2.6748; SKO_total = 2.6713
SKO_random = 2.5021; SKO_total = 2.5035

NC=9
SKO_random = 2.8389; SKO_total = 2.8381
SKO_random = 2.9268; SKO_total = 2.9266
SKO_random = 3.2608; SKO_total = 3.2609
SKO_random = 2.9258; SKO_total = 2.9245
SKO_random = 3.0073; SKO_total = 3.0069

NC=12;
SKO_random = 3.8111; SKO_total = 3.8108
SKO_random = 3.7571; SKO_total = 3.7575
SKO_random = 3.7943; SKO_total = 3.7928
SKO_random = 3.5735; SKO_total = 3.5760
SKO_random = 3.3284; SKO_total = 3.3291

NC=15
SKO_random = 3.5117; SKO_total = 3.5144
SKO_random = 3.5117; SKO_total = 3.5144
SKO_random = 3.7573; SKO_total = 3.7569
SKO_random = 3.7573; SKO_total = 3.7569
SKO_random = 3.7573; SKO_total = 3.7569

NC=17
SKO_random = 4.0908; SKO_total = 4.0885
SKO_random = 4.3821; SKO_total = 4.3808
SKO_random = 4.4489; SKO_total = 4.4543
SKO_random = 4.9342; SKO_total = 4.9304
SKO_random = 4.2473; SKO_total = 4.2494

NC=20
SKO_random = 4.2528; SKO_total = 4.2583
SKO_random = 4.8402; SKO_total = 4.8368
SKO_random = 4.6649; SKO_total = 4.6658
SKO_random = 4.6799; SKO_total = 4.6787
SKO_random = 4.5882; SKO_total = 4.5830

NC=25
SKO_random = 5.4509; SKO_total = 5.4511
SKO_random = 5.6114; SKO_total = 5.6104
SKO_random = 5.2022; SKO_total = 5.2004
SKO_random = 5.0598; SKO_total = 5.0605
SKO_random = 5.2168; SKO_total = 5.2181

NC=30
```

```
SKO_random = 6.4512; SKO_total = 6.4512
SKO_random = 5.8573; SKO_total = 5.8571
SKO_random = 5.5337; SKO_total = 5.5336
SKO_random = 6.1428; SKO_total = 6.1429
SKO_random = 5.9855; SKO_total = 5.9854

NC=40
SKO_random = 7.1195; SKO_total = 7.1250
SKO_random = 6.9142; SKO_total = 6.9078
SKO_random = 6.7722; SKO_total = 6.7712
SKO_random = 6.4618; SKO_total = 6.4593
SKO_random = 6.6698; SKO_total = 6.6699

NC=50;
SKO_random = 7.4935; SKO_total = 7.4872
SKO_random = 7.4065; SKO_total = 7.4363
SKO_random = 7.5284; SKO_total = 7.5401
SKO_random = 7.9340; SKO_total = 7.9429
SKO_random = 7.7284; SKO_total = 7.7189

NC=60;
SKO_random = 8.3757; SKO_total = 8.4126
SKO_random = 8.2211; SKO_total = 8.2294
SKO_random = 8.8256; SKO_total = 8.8204
SKO_random = 7.6740; SKO_total = 7.7277
SKO_random = 8.1573; SKO_total = 8.1861

NC=70;
SKO_random = 8.9279; SKO_total = 8.9985
SKO_random = 9.0549; SKO_total = 9.0670
SKO_random = 9.5694; SKO_total = 9.6050
SKO_random = 9.5112; SKO_total = 9.5503
SKO_random = 9.3157; SKO_total = 9.3631
SKO_random = 9.1844; SKO_total = 9.1765

NC=80;
SKO_random =10.1797; SKO_total =10.2675
SKO_random =10.4523; SKO_total =10.4739
SKO_random = 9.9967; SKO_total =10.0261
SKO_random =10.0938; SKO_total =10.0903
SKO_random =10.7432; SKO_total =10.7701

NC=90;
SKO_random =11.0320; SKO_total =10.9641
SKO_random =10.7075; SKO_total =10.7775
SKO_random =11.3909; SKO_total =11.4863
SKO_random =10.9012; SKO_total =11.0500
SKO_random =10.9743; SKO_total =11.1041

NC=100;
SKO_random =12.1020; SKO_total =12.2123
SKO_random =11.8351; SKO_total =11.8727
SKO_random =12.0985; SKO_total =12.2470
SKO_random =11.4988; SKO_total =11.5787
SKO_random =11.2539; SKO_total =11.2948



+=========================================+
Q=0.4;
```

```
KP1=5;
+=======================================+
NC=3
SKO_random =1.6690; SKO_total =6.2232
SKO_random =2.9597; SKO_total =6.6409
SKO_random =1.5653; SKO_total =7.0395
SKO_random =1.6690; SKO_total =6.2232
SKO_random =1.8700; SKO_total =6.7230

NC=5
SKO_random =2.6012; SKO_total =2.6299
SKO_random =2.6977; SKO_total =2.6173
SKO_random =2.7558; SKO_total =2.8289
SKO_random =3.4453; SKO_total =3.4539
SKO_random =2.9752; SKO_total =2.9515

NC=7
SKO_random =3.3362; SKO_total =3.3375
SKO_random =3.2409; SKO_total =3.2435
SKO_random =3.2935; SKO_total =3.2942
SKO_random =3.2766; SKO_total =3.2809
SKO_random =3.5385; SKO_total =3.5401

NC=9
SKO_random =4.4338; SKO_total =4.4341
SKO_random =4.4108; SKO_total =4.4118
SKO_random =3.9636; SKO_total =3.9618
SKO_random =3.5960; SKO_total =3.5955
SKO_random =3.4394; SKO_total =3.4385

NC=12;
SKO_random =4.6821; SKO_total =4.6824
SKO_random =4.3842; SKO_total =4.3839
SKO_random =4.1436; SKO_total =4.1456
SKO_random =4.3255; SKO_total =4.3212
SKO_random =4.4960; SKO_total =4.4949

NC=15
SKO_random =5.4638; SKO_total =5.4625
SKO_random =5.1217; SKO_total =5.1264
SKO_random =5.1744; SKO_total =5.1782
SKO_random =4.9995; SKO_total =4.9916
SKO_random =4.6472; SKO_total =4.6436

NC=17
SKO_random =5.7624; SKO_total =5.7555
SKO_random =5.5493; SKO_total =5.5515
SKO_random =5.8748; SKO_total =5.8730
SKO_random =6.1537; SKO_total =6.1514
SKO_random =6.0439; SKO_total =6.0405

NC=20
SKO_random =5.8256; SKO_total =5.8308
SKO_random =6.0453; SKO_total =6.0400
SKO_random =6.3643; SKO_total =6.3667
SKO_random =6.4321; SKO_total =6.4337
SKO_random =5.9967; SKO_total =6.0024

NC=25
SKO_random =7.8978; SKO_total =7.9041
```

```
SKO_random =6.7560; SKO_total =6.7560
SKO_random =7.3892; SKO_total =7.3876
SKO_random =6.9842; SKO_total =6.9800
SKO_random =7.5055; SKO_total =7.5025

NC=30
SKO_random =7.9093; SKO_total =7.9094
SKO_random =7.7865; SKO_total =7.7871
SKO_random =8.0500; SKO_total =8.0498
SKO_random =8.1986; SKO_total =8.1983
SKO_random =8.0680; SKO_total =8.0680

NC=40
SKO_random =8.4294; SKO_total =8.4337
SKO_random =9.0631; SKO_total =9.0536
SKO_random =9.1613; SKO_total =9.1616
SKO_random =8.4766; SKO_total =8.4666
SKO_random =9.5208; SKO_total =9.5295
SKO_random =9.3686; SKO_total =9.3701

NC=50;
SKO_random =9.8039; SKO_total =9.7946
SKO_random =10.2621; SKO_total =10.2621
SKO_random =9.9478; SKO_total =9.9715
SKO_random =9.8774; SKO_total =9.8807
SKO_random =9.7808; SKO_total =9.7868

NC=60;
SKO_random =11.6291; SKO_total =11.6488
SKO_random =10.6178; SKO_total =10.6245
SKO_random =11.0245; SKO_total =10.9889
SKO_random =10.9238; SKO_total =10.9294
SKO_random =11.2791; SKO_total =11.2799

NC=70;
SKO_random =12.7172; SKO_total =12.7578
SKO_random =11.7274; SKO_total =11.7374
SKO_random =11.7616; SKO_total =11.7643
SKO_random =12.2905; SKO_total =12.3221
SKO_random =12.3535; SKO_total =12.3742

NC=80;
SKO_random =13.6566; SKO_total =13.7155
SKO_random =13.8744; SKO_total =13.8573
SKO_random =12.8513; SKO_total =12.9120
SKO_random =13.4775; SKO_total =13.5090
SKO_random =13.1524; SKO_total =13.2543

NC=90;
SKO_random =14.0277; SKO_total =14.0983
SKO_random =14.2485; SKO_total =14.3079
SKO_random =14.6522; SKO_total =14.6640
SKO_random =14.5450; SKO_total =14.6241
SKO_random =14.6765; SKO_total =14.7371
SKO_random =15.0121; SKO_total =15.0617

NC=100;
SKO_random =15.0101; SKO_total =15.0584
SKO_random =15.8717; SKO_total =15.9176
SKO_random =15.9390; SKO_total =15.9788
```

```
SKO_random =15.1904; SKO_total =15.3082
SKO_random =16.7351; SKO_total =16.8556
SKO_random =15.5650; SKO_total =15.6828
```

```
+=========================================+
Q=0.5;
KP1=5;
+=========================================+
NC=3
SKO_random = 2.1047; SKO_total = 6.5492
SKO_random = 3.8840; SKO_total = 7.1935
SKO_random = 2.4314; SKO_total = 6.7406
SKO_random = 3.8697; SKO_total = 7.0889
SKO_random = 2.1560; SKO_total = 7.2289

NC=5
SKO_random = 4.1288; SKO_total = 4.2740
SKO_random = 3.7164; SKO_total = 3.7919
SKO_random = 4.0959; SKO_total = 4.1125
SKO_random = 3.5722; SKO_total = 3.6412
SKO_random = 3.7720; SKO_total = 3.8741

NC=7
SKO_random = 5.7578; SKO_total = 5.7632
SKO_random = 4.6666; SKO_total = 4.6681
SKO_random = 5.6338; SKO_total = 5.6352
SKO_random = 4.1412; SKO_total = 4.1339
SKO_random = 4.5708; SKO_total = 4.5747

NC=9
SKO_random = 5.0822; SKO_total = 5.0845
SKO_random = 5.0240; SKO_total = 5.0240
SKO_random = 4.7564; SKO_total = 4.7547
SKO_random = 4.6605; SKO_total = 4.6609
SKO_random = 4.9803; SKO_total = 4.9796

NC=12;
SKO_random = 6.2535; SKO_total = 6.2483
SKO_random = 5.4607; SKO_total = 5.4575
SKO_random = 6.1886; SKO_total = 6.1885
SKO_random = 6.5371; SKO_total = 6.5405
SKO_random = 5.6958; SKO_total = 5.6961

NC=15
SKO_random = 7.1192; SKO_total = 7.1117
SKO_random 6.1287; SKO_total = 6.1271
SKO_random 6.3150; SKO_total = 6.3157
SKO_random = 7.0155; SKO_total = 7.0164
SKO_random = 6.7026; SKO_total = 6.6969

NC=17
SKO_random = 7.2164; SKO_total = 7.2155
SKO_random = 6.8305; SKO_total = 6.8244
SKO_random = 6.9642; SKO_total = 6.9712
SKO_random = 6.9754; SKO_total = 6.9736
SKO_random = 7.0943; SKO_total = 7.0893
```

```
NC=20
SKO_random = 7.2846; SKO_total = 7.2974
SKO_random = 7.6901; SKO_total = 7.6914
SKO_random = 7.0315; SKO_total = 7.0290
SKO_random = 7.8516; SKO_total = 7.8514
SKO_random = 8.1123; SKO_total = 8.1022

NC=25
SKO_random = 8.7957; SKO_total = 8.7985
SKO_random = 8.5014; SKO_total = 8.5020
SKO_random = 8.1015; SKO_total = 8.0985
SKO_random = 8.5001; SKO_total = 8.4979
SKO_random = 8.8018; SKO_total = 8.7967

NC=30
SKO_random = 9.6059; SKO_total = 9.6068
SKO_random = 9.1637; SKO_total = 9.1634
SKO_random = 9.5142; SKO_total = 9.5145
SKO_random = 9.8370; SKO_total = 9.8372
SKO_random = 9.1150; SKO_total = 9.1153

NC=40
SKO_random =10.8779; SKO_total =10.8818
SKO_random =11.4947; SKO_total =11.4855
SKO_random =11.7233; SKO_total =11.7175
SKO_random =11.1289; SKO_total =11.1260
SKO_random =11.4574; SKO_total =11.4670
SKO_random =11.5831; SKO_total =11.5964

NC=50;
SKO_random =12.3674; SKO_total =12.3797
SKO_random =13.2088; SKO_total =13.2280
SKO_random =12.5688; SKO_total =12.5637
SKO_random =12.5500; SKO_total =12.5415
SKO_random =13.5294; SKO_total =13.5515
SKO_random =12.8066; SKO_total =12.8054

NC=60;
SKO_random =14.1448; SKO_total =14.1394
SKO_random =13.8896; SKO_total =13.8786
SKO_random =14.2767; SKO_total =14.2513
SKO_random =14.6135; SKO_total =14.6650
SKO_random =15.5721; SKO_total =15.5425

NC=70;
SKO_random =14.4146; SKO_total =14.4476
SKO_random =15.9943; SKO_total =16.0110
SKO_random =14.7613; SKO_total =14.7425
SKO_random =15.8581; SKO_total =15.8841
SKO_random =15.3762; SKO_total =15.3755
SKO_random =16.2715; SKO_total =16.2514

NC=80;
SKO_random =17.0639; SKO_total =17.1080
SKO_random =17.0716; SKO_total =17.1570
SKO_random =16.8914; SKO_total =16.8378
SKO_random =16.3895; SKO_total =16.4210
SKO_random =16.1013; SKO_total =16.1346
```

```
NC=90;
SKO_random =17.9378; SKO_total =18.0264
SKO_random =17.6949; SKO_total =17.7005
SKO_random =18.3477; SKO_total =18.3717
SKO_random =18.0882; SKO_total =18.0964
SKO_random =18.8956; SKO_total =19.0189

NC=100;
SKO_random =18.7728; SKO_total =18.9262
SKO_random =19.5564; SKO_total =19.5764
SKO_random =19.8899; SKO_total =19.9440
SKO_random =18.8198; SKO_total =18.8280
SKO_random =18.6391; SKO_total =18.672
```