Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №2.2

з дисципліни
«Інтелектуальні вбудовані системи»

на тему
«Дослідження параметрів алгоритму швидкого перетворення Фур'є з проріджуванням відліків сигналів у часі»

Виконав:
студент групи ІП-84
Голубов Іван
номер залікової книжки: 8404

Перевірив:
викладач
Регіда Павло Геннадійович

Київ 2021

Основні теоретичні відомості

Швидкі алгоритми ПФ отримали назву схеми Кулі-Тьюкі. Всі ці алгоритми використовують регулярність самої процедури ДПФ і те, що будь-який складний коефіцієнт W_N^{pk} можна розкласти на прості комплексні коефіцієнти.

$$W_N^{pk} = W_N^1 W_N^2 W_N^3$$

Для стану таких груп коефіцієнтів процедура ДПФ повинна стати багаторівневою, не порушуючи загальних функціональних зв'язків графа процедури ДПФ.

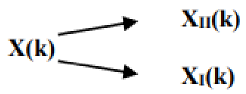
Існують формальні підходи для отримання регулярних графів ДПФ. Всі отримані алгоритми поділяються на 2 класи:

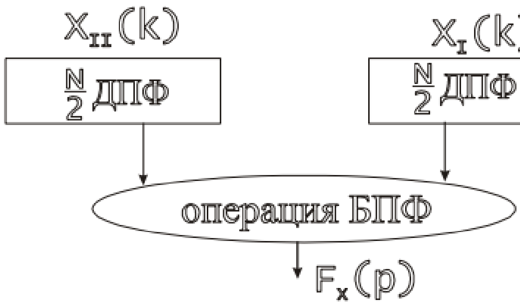
1) На основі реалізації принципу зріджені за часом $\mathbf{X_к}$

2)на основі реалізації принципу зріджені відліків шуканого спектру **F(p).**

Найпростіший принцип зріджені - поділу на парні/непарні пів-послідовності, які потім обробляють паралельно. А потім знаходять алгоритм, як отримати шуканий спектр.

Якщо нам вдасться ефективно розділити, а потім алгоритм отримання спектра, то ми можемо перейти від **N** ДПФ до **N/2** ДПФ.





Розглянемо формальний висновок алгоритму ШПФ, який реалізує в одноразовому застосуванні принцип проріджування по часу:

$$F_x(p) = \sum_{k=0}^{N-1} X(k) W_N^{pk} = \sum_{k=0}^{N-2} X_{II}(k) W_N^{pk} + \sum_{k=1}^{N-2} X_I(k) W_N^{pk}$$

$$X_{II}(k) \to X(2k^*);\; X_I(k) \to X(2k^*+1);\; k^* = 0; \frac{N}{2} - 1$$

$$F_x(p) = \sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*) W_N^{pk} + \sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*+1) W_N^{p(2k^*+1)}$$

$$W_N^{p2k^*} = e^{-j\frac{2\pi}{N}p2k^*} = e^{-j\frac{2\pi}{N/2}pk^*} = W_{\frac{N}{2}}^{pk^*}$$

У цій першій сумі з'явилися коефіцієнти в 2 рази менше.

У другій сумі з'явився множник, який не залежить від k * тобто він може бути винесений за знак суми.

$$W_N^{p(2k^*+1)} = W_N^{p2k^*} \cdot W_N^{p} = W_{\frac{N}{2}}^{pk^*} W_N^{p}$$

$$F_x(p) = \underbrace{\sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*) W_{\frac{N}{2}}^{pk^*}}_{F_{II}(p^*)} + W_N^{p} \underbrace{\sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*+1) W_{\frac{N}{2}}^{pk^*}}_{F_I(p^*)}$$

Ми бачимо, що всі вирази можна розділити на 2 частини, які обчислюються паралельно.

$F_I(p^*)$ - проміжний спектр, побудований на парних відліку. У цьому алгоритмі передбачається, щоб отримати спектр **F(p)** треба виконати 2 незалежних **N/2** ШПФ.

$$1)\quad F_{II}(p^*) = \sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*) W_{\frac{N}{2}}^{pk^*} \qquad p^* = 0, \frac{N}{2}-1$$

$$2)\quad F_I(p^*) = \sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*+1) W_{\frac{N}{2}}^{pk^*}$$

А на наступному кроці буде реалізована швидка збірка, тобто ШПФ з зрідженим за часом за формулою:

$F_\tau(p^*) = F_{II}(p^*) + W_N^{p^*} F_I(p^*)$

Але в цьому виразі різні **p** для зв'язку їх

Якщо $p < N/2$, то $p = p^*$ 1-а половина спектру

Якщо $p \geq N/2$, то $p = p^* + N/2$ 2-а половина спектру

В алгоритмі БПФ вже використовуються 2 рівня

$$F_x(p^*) = F_{II}(p^*) + W_N^{p^*} F_I(p^*)$$

$$F_x\left(p^* + \frac{N}{2}\right) = F_{II}(p^*) + W_N^{p^* + \frac{N}{2}} F_I(p^*)$$

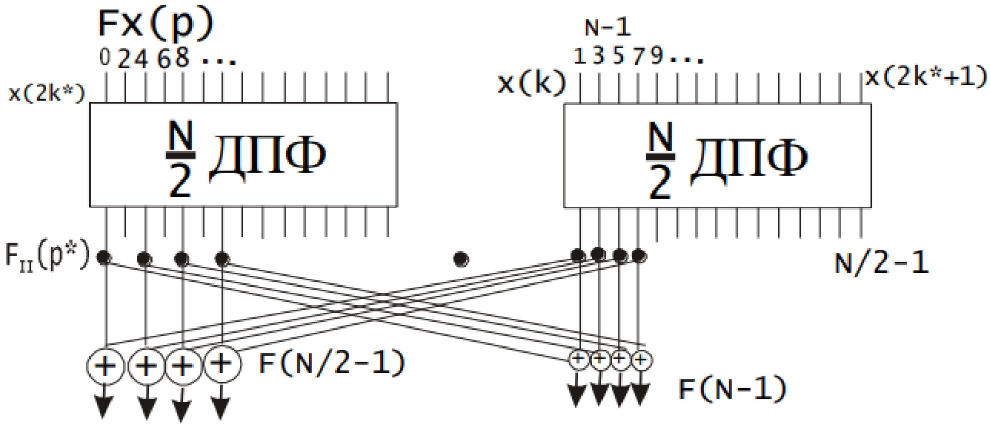
Алгоритм ШПФ з зрідженим по часу:

$$F_x(p^*) = F_{II}(p^*) + W_N^{p^*} F_I(p^*) \qquad p^* = 0, \frac{N}{2}-1$$

$$F_x\left(p^* + \frac{N}{2}\right) = F_{II}(p^*) - W_N^{p^*} F_I(p^*) \qquad W_N^{p^*}$$

$\frac{N}{2}$ помножений на комплексний коефіцієнт.

Загальна схема самого ДПФ змінилася замість однорівневого перетворення.



Буде потрібно: $\frac{N^2}{2}+\frac{N}{2}$ множень. Точно такий же прийом можна виконати і для $\frac{N}{2}$ ДПФ перетворень:

$$X(2k^*) \to X_{II}(2k^*), \quad X(2k^*) \to X_{I}(2k^*)$$

тоді

$$X(2\cdot 2k^*) \to X(4k^{**}),\ X(2\cdot(2k^*+1)) \to X(4k^*+2)$$

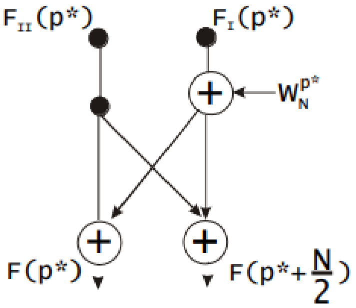
$$X_I(2k^*+1) \to X(2\cdot 2k^{**}+1) = X(4k^{**}+1);$$

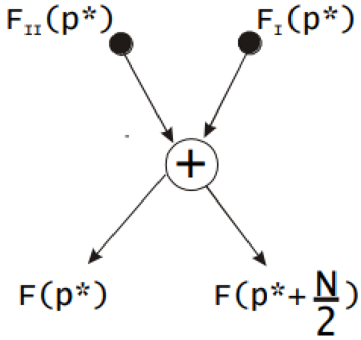
$$\to X(2\cdot(2k^*+1)+1) = X(4k^{**}+3);$$

<u>Особливості апаратної реалізації базової операції БПФ з проріджуванням за часом.</u>

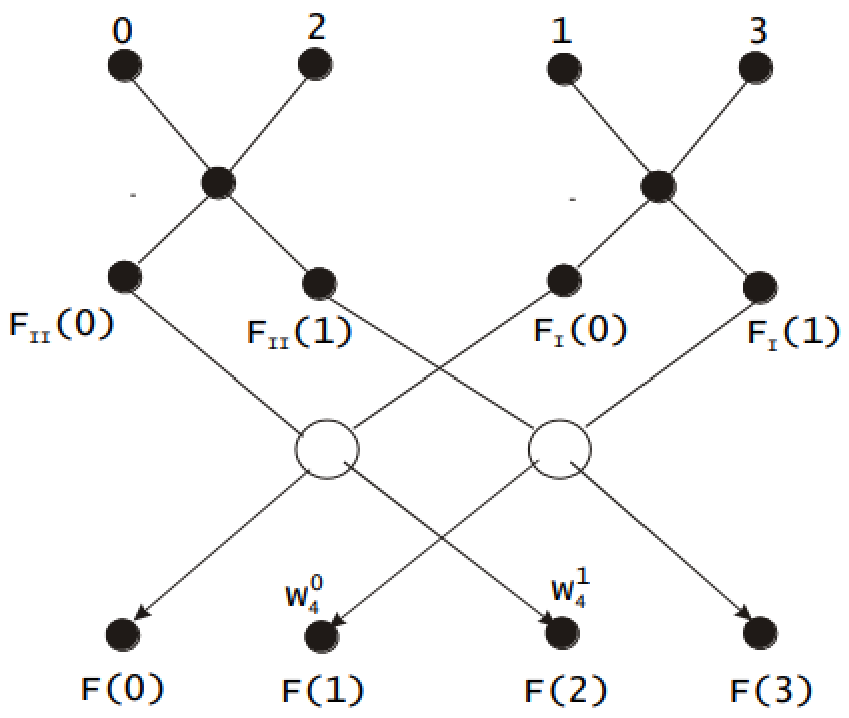
В процесорах цифрової обробки сигналів спеціальне АЛУ орієнтоване на виконання реалізації ШПФ або за часом, або по частоті. Функції, реалізовані базовими операціями БПФ, визначаються базовими алгоритмами. Операнди представлені як комплексні величини. Особливість операції - 2 операнда на вході, на виході так само 2 результату.

Графічно базову операцію БПФ описують так:





Цю базову операцію називають метеликом через форми графа. Для спрощеного аналізу процедури БПФ використовують символічне зображення.



Завдання на лабораторну роботу

Для згенерованого випадкового сигналу з Лабораторної роботи N 1 відповідно до заданого варіантом (Додаток 1) побудувати його спектр, використовуючи процедуру швидкого перетворення Фур'є з проріджуванням відліків сигналу за часом. Розробити відповідну програму і вивести отримані значення і графіки відповідних параметрів.

Варіант-04

Число гармонік в сигналі: 12.

Гранична частота: 2400.

Кількість дискретних відліків: 1024

Примітка

Використовуючи особливості ДПФ задля зменшення часу обчислень, розраховується лише N/8+1 коефіцієнт.

Лістинг програми

lab2.m

```
X=zeros(3,1024);
x(1,:)=signal_generator(12,1024);
x(2,:)=signal_generator(12,1024);
x(3,:)=signal_generator(12,1024);
N = 1024;
Fd=1024;% Частота дискретизации
[F_DFT, title_DFT] = FT.DFT(x(1,:),N);
[F_FFT_h_W, title_FFT_h_W]= FT.FFT_handmade_without_W  (x(1,:),N);
```

```
[F_FFT_h, title_FFT_h]= FT.FFT_handmade (x(1,:),N);
[F_FFT_m, title_FFT_m]=  FT.FFT_matlab(x(1,:));
[A_lenH_DFT, H1]= FT.plot_FT (F_DFT,title_DFT,Fd,N);
[A_lenH_FFT, H2]= FT.plot_FT (F_FFT_h,title_FFT_h,Fd,N);
FT.plot_FT (F_FFT_h_W,title_FFT_h_W,Fd,N);
FT.plot_FT (F_FFT_m,title_FFT_m,Fd,N);
FT.compare_deltaA_FFT_DFT (A_lenH_DFT, A_lenH_FFT, H1);
f_DFT=@() FT.DFT(x(1,:),N);
f_FFT=@() FT.FFT_handmade(x(1,:),N);
t_DFT=MD.time_f(f_DFT,'ДПФ');
t_FFT=MD.time_f(f_FFT,'БПФ');
MD.difference_time (t_DFT,t_FFT, 'ДПФ', 'БПФ');
[numRows,numCols ] = size(X);
for i=1:numRows
   FT.compare_time_FFT(X(i,:), N);
End
```

FT.m

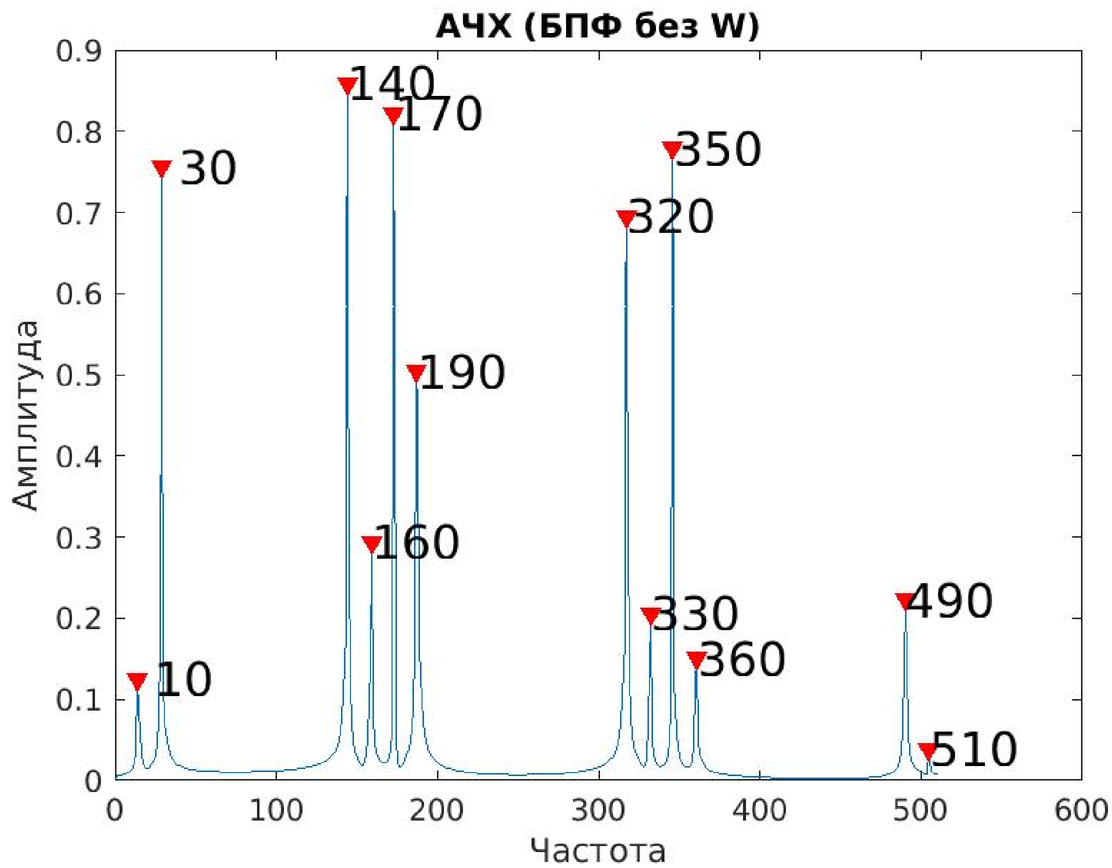
```
classdef FT
   methods (Static)
      function [F, title_FT] = DFT(x,N)
         F = complex(zeros(1,N));
         title_FT = 'АЧХ (ДПФ)';
         for p=0:N-1
            for k = 0:N-1
               F(1,p+1) = F(1,p+1) + x(k+1)*(cos(2*pi*p*k/N)-sin(2*pi*p*k/N)*1i);
            end
         end
      end
      function [F, title_FT]=FFT_handmade_without_W (x,N)
         F = complex(zeros(1,N));
         title_FT= 'АЧХ (БПФ без W)';
         w=zeros(2,N);
         for i = 0:N-1
            if i<=N/8
               w(1,i+1)=cos(2*pi*i/N);
               w(2,i+1)=sin(2*pi*i/N);
            elseif i <=N/4
               w(1,i+1)=w(2,N/4-i+1);
               w(2,i+1)=w(1,N/4-i+1);
```

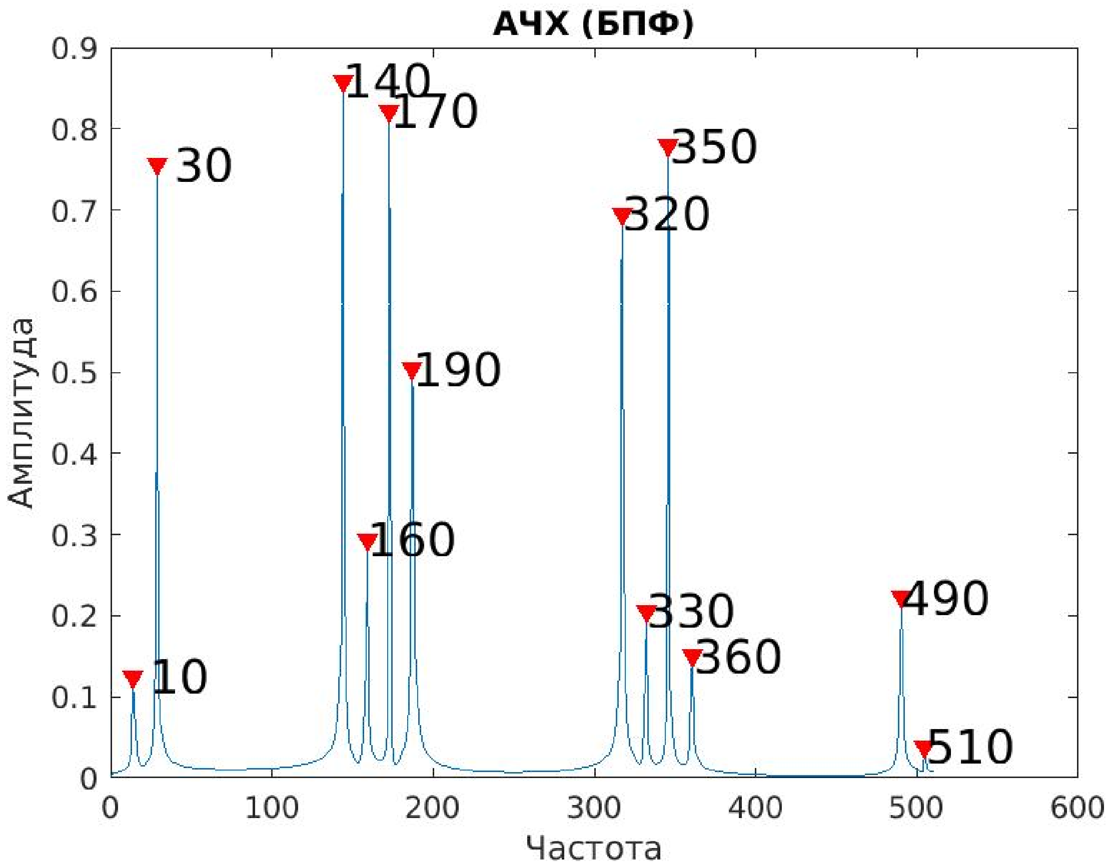
```
        elseif i<=N/2
            w(1,i+1)=-w(2,i-N/4+1);
            w(2,i+1)=+w(1,i-N/4+1);
        else
            w(1,i+1)=-w(1,i-N/2+1);
            w(2,i+1)=-w(2,i-N/2+1);
        end
    end
    for p=0:N-1
        for k = 0:N-1
            a=mod(p*k,N);
            F(1,p+1) = F(1,p+1) + x(k+1)*(w(1,a+1)-w(2,a+1)*1i);
        end
    end
end
function [F, title_FT]=FFT_handmade (x,N)
    F = complex(zeros(1,N));
    title_FT= 'АЧХ (БПФ)';
    w=zeros(2,N/4+1);
    W=zeros(1,N);
    for i = 0:N/4
        if i<=N/8
            w(1,i+1)=cos(2*pi*i/N);
            w(2,i+1)=sin(2*pi*i/N);
        else
            w(1,i+1)=w(2,N/4-i+1);
            w(2,i+1)=w(1,N/4-i+1);
        end
    end
    for b= 0:N-1
        if b <=N/4
            W(1,b+1)= w(1,b+1)-w(2,b+1)*1i;
        elseif b<=N/2
            W(1,b+1)=W(1,b-N/4+1)*-1*1i;
        else
            W(1,b+1)=-W(1,b-N/2+1);
        end
    end
    for p=0:N-1
```

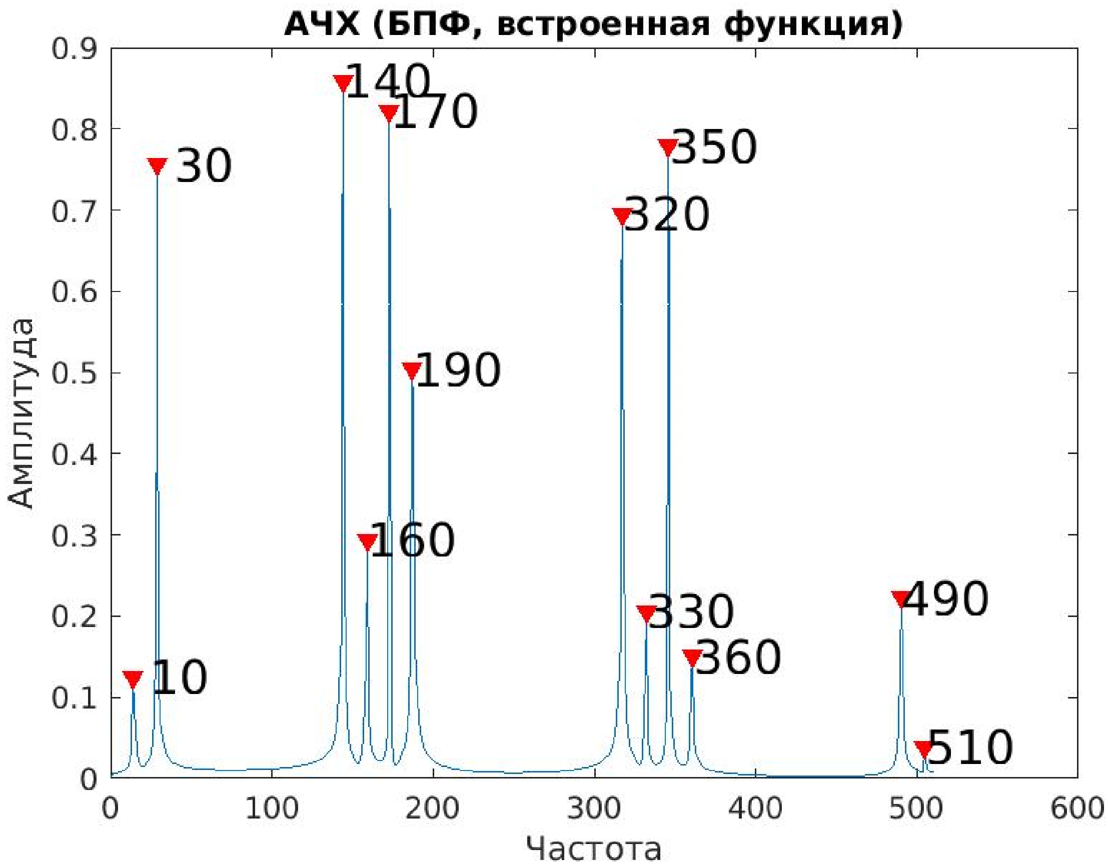
```
        for k = 0:N-1
            a=mod(p*k,N);
            F(1,p+1) = F(1,p+1) + x(k+1)*W(1,a+1);
        end
    end
end
function [F, title_FT]=FFT_matlab(x)
    F=fft(x);
    title_FT = 'АЧХ (БПФ, встроенная функция)';
end
function [A_lenH, H]=plot_FT (F,title_FT,Fd,N)
    A=abs(F);% Амплитуды преобразования Фурье сигнала
    A=2*A./N;% Нормировка спектра по амплитуде
    A(1)=A(1)/2;% Нормировка постоянной составляющей в спектре
    H=0:Fd/N:Fd/2-1/N;% Массив частот вычисляемого спектра Фурье
    A_lenH = A(1:length(H));

    figure
    plot(H,A_lenH);% Построение спектра Фурье сигнала
    hold on;
    title(title_FT);
    xlabel('Частота');
    ylabel('Амплитуда');
    [~,locs1] = findpeaks(A_lenH,'MinPeakHeight',0.02,...
        'MinPeakDistance',2);
    plot(H(locs1),A_lenH(locs1),'rv','MarkerFaceColor','r');
    cellpeaks = cellstr(num2str(round(H(locs1)',-1)));
    text(H(locs1),A_lenH(locs1),cellpeaks,'FontSize',16);
    hold off;
    saveas(gcf, sprintf('./res/%s.jpg',title_FT))
end
function compare_deltaA_FFT_DFT (A_lenH_DFT, A_lenH_FFT, H)
    figure
    plot(H,A_lenH_FFT-A_lenH_DFT);
    hold on;
    title('Отклонение БПФ от ДПФ');
    xlabel('Частота');
    ylabel('Отклонение');
    saveas(gcf,'./res/compare_deltaA_FFT_DFT.jpg');
```

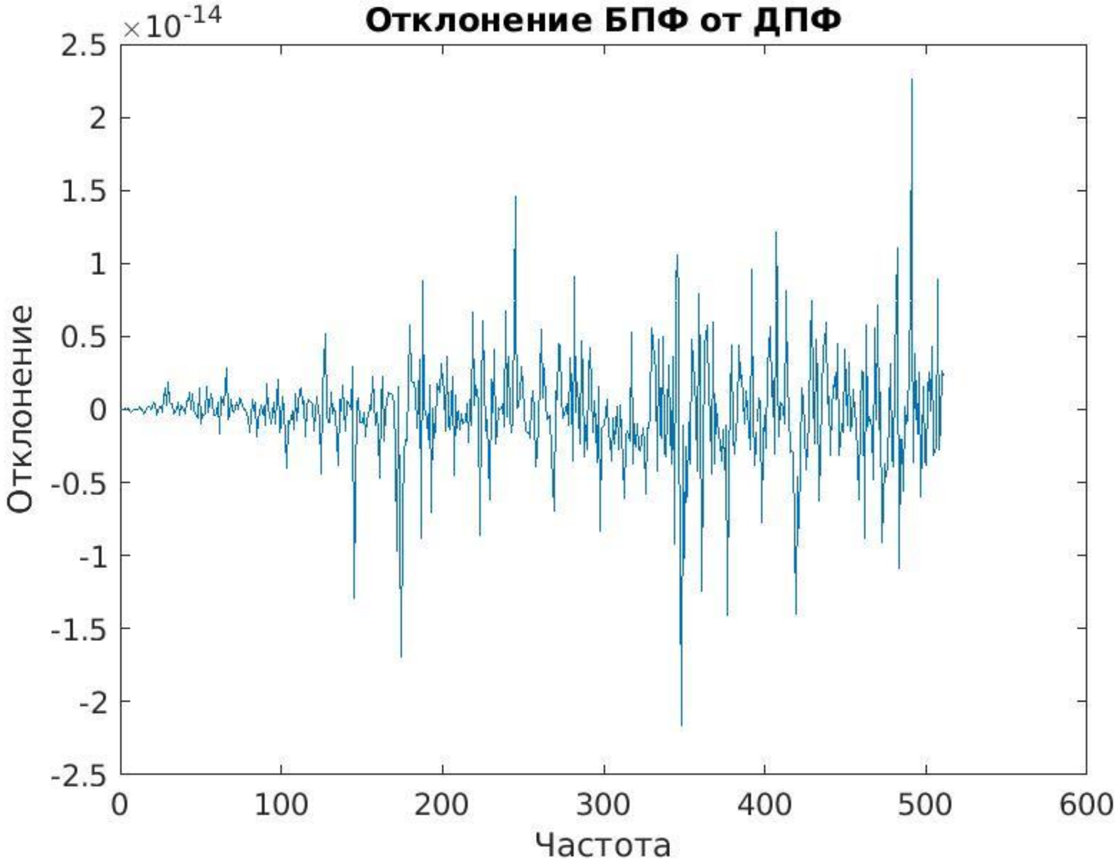
```
        end
        function compare_time_FFT(y,N)
            f1=@() FT.FFT_handmade_without_W(y,N);
            f2=@() FT.FFT_handmade(y,N);
            t1=MD.time_f(f1,'БПФ без W');
            t2=MD.time_f(f2,'БПФ');
            MD.difference_time (t1,t2, 'БПФ без W', 'БПФ');
        end
    end
end
```

Результати роботи програми




АЧХ (БПФ)
Амплитуда
Частота
10
30
140
160
170
190
320
330
350
360
490
510


АЧХ (БПФ, встроенная функция)
Амплитуда
Частота
10
30
140
160
170
190
320
330
350
360
490
510



>> lab2
Время выполнения ДПФ функции
t(ДПФ) = 0.234096

Время выполнения БПФ функции
t(БПФ) = 0.139782

Соотношение времени выполнения ДПФ и БПФ функций
t(ДПФ)/t(БПФ) = 1.674720

Время выполнения БПФ без W функции
t(БПФ без W) = 0.169992

Время выполнения БПФ функции
t(БПФ) = 0.102964

Соотношение времени выполнения БПФ без W и БПФ функций
t(БПФ без W)/t(БПФ) = 1.650982

Время выполнения БПФ без W функции
t(БПФ без W) = 0.173733

Время выполнения БПФ функции
t(БПФ) = 0.105139

Соотношение времени выполнения БПФ без W и БПФ функций
t(БПФ без W)/t(БПФ) = 1.652410

Время выполнения БПФ без W функции
t(БПФ без W) = 0.171553

Время выполнения БПФ функции
t(БПФ) = 0.103219

Соотношение времени выполнения БПФ без W и БПФ функций
t(БПФ без W)/t(БПФ) = 1.662026

Висновки

Під час даної лабораторної роботи я ознайомився з принципами реалізації прискореного спектрального аналізу випадкових сигналів на основі алгоритму швидкого перетворення Фур'є, вивчив та дослідив особливості даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.