

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №3
з дисципліни «Системи реального часу»
на тему *«Дослідження параметрів алгоритму дискретного перетворення
Фур'є»*

Виконав:
студент гр. ПІ-83
Бойко Андрій

Перевірив:
Регіда П.Г.

Завдання

Мета роботи - ознайомлення з принципами реалізації спектрального аналізу випадкових сигналів на основі алгоритму перетворення Фур'є, вивчення та дослідження особливостей даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.

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

Програмний код

dft.js

```
const { complex, exponent } = require('./utils');

const dft = vector => {
  const X = [];
  const N = vector.length;

  for (let k = 0; k < N; k++) {
    X[k] = [0, 0];

    for (let i = 0; i < N; i++) {
      const exp = exponent(k * i, N);
      let term;
      if (Array.isArray(vector[i])) {
        term = complex.multiply(vector[i], exp);
      } else {
        term = complex.multiply([vector[i], 0], exp);
      }
      X[k] = complex.add(X[k], term);
    }
  }

  return X;
```

```
};
```

```
module.exports = { dft };
```

utils.js

```
const complexAdd = (a, b) => [a[0] + b[0], a[1] + b[1]];
```

```
const complexSubtract = (a, b) => [a[0] - b[0], a[1] - b[1]];
```

```
const complexMultiply = (a, b) => [a[0] * b[0] - a[1] * b[1],  
  a[0] * b[1] + a[1] * b[0]];
```

```
const complexMagnitude = c => Math.sqrt(c[0] * c[0] + c[1] * c[1]);
```

```
const mapExponent = {},  
  exponent = (k, N) => {  
    const x = -2 * Math.PI * (k / N);  
  
    mapExponent[N] = mapExponent[N] || {};  
    mapExponent[N][k] = mapExponent[N][k] || [Math.cos(x), Math.sin(x)];  
  
    return mapExponent[N][k];  
  };
```

```
const convert = arr => arr.map(([x, y]) => Math.sqrt(Math.pow(x, 2) +  
  Math.pow(y, 2)));
```

```
module.exports = {  
  complex: {  
    add: complexAdd,  
    subtract: complexSubtract,  
    multiply: complexMultiply,  
    magnitude: complexMagnitude,  
  },  
  exponent,
```

```
    convert,  
};
```

```
index.js
```

```
const { generateSignal } = require('../1/generateSignal');  
const { dft } = require('../dft');  
const plt = require('matplotnode');  
const path = require('path');  
const { convert } = require('../utils');
```

```
const harmonics = 6;  
const frequency = 1200;  
const discreteCalls = 64;
```

```
const signal = generateSignal(  
    harmonics,  
    frequency,  
    discreteCalls,  
);
```

```
const spectrumDft = dft(signal);
```

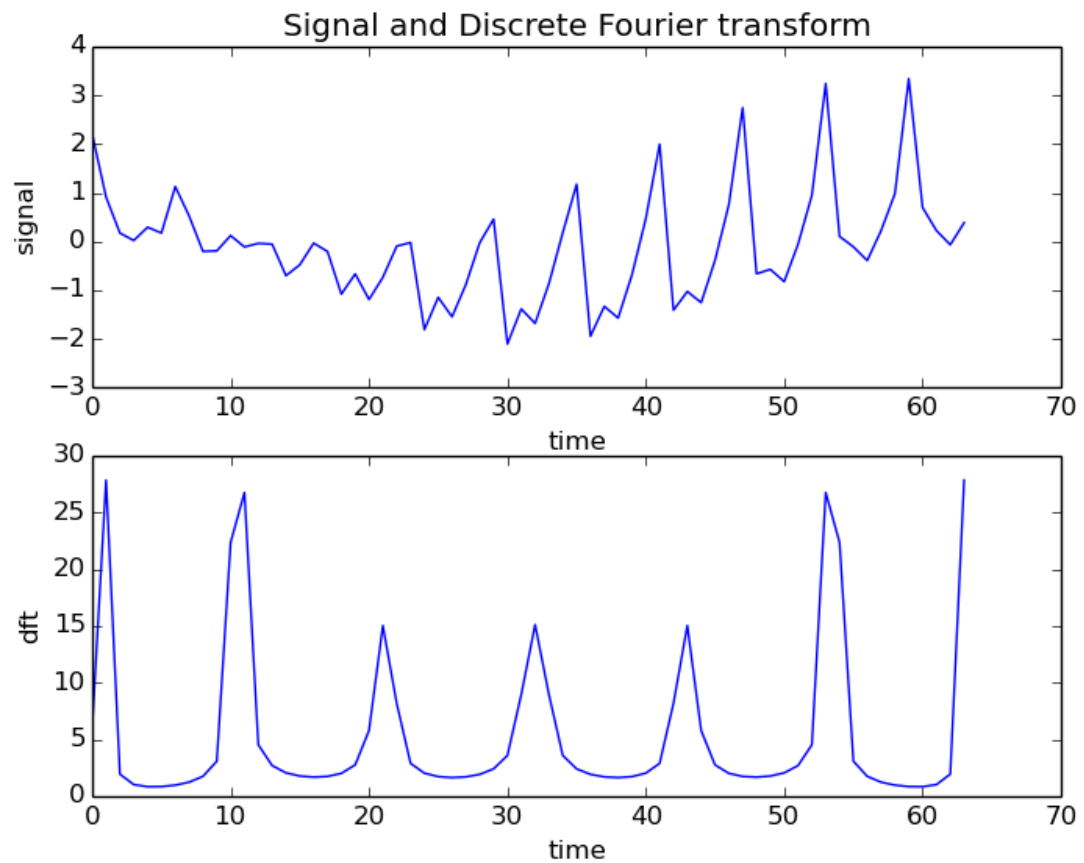
```
plt.subplot('211');  
plt.title('Signal and Discrete Fourier transform');  
plt.plot([...Array(discreteCalls).keys()], signal);  
plt.xlabel('time');  
plt.ylabel('signal');  
plt.legend();
```

```
plt.subplot('212');  
plt.plot([...Array(discreteCalls).keys()], convert(spectrumDft));  
plt.xlabel('time');  
plt.ylabel('dft');  
plt.legend();
```

```
const currentDir = path.join(__dirname, '/2.1.png');
```

```
plt.save(currentDir);
```

Результати виконання програми



Висновки

Під час виконання лабораторної роботи, було розкладено сигнал за допомогою дискретної трансформації Фур'є.