НИУ ИТМО

Факультет систем управления и робототехники

Лабораторная работа №5
по дисциплине «Частотные методы»

Выполнил: Гридусов Д.Д

Преподаватель: Перегудин А.А

Санкт-Петербург
2024 г.

Содержание

1 Непрерывное и дискретное Фурье преобразование

1.1 Истиный образ Фурье

$$\Pi(t) = \begin{cases} 1, |t| \leq \frac{1}{2}, \\ 0, |t| > \frac{1}{2} \end{cases}$$

$$\hat{\Pi}(\nu) = \int\limits_{-\infty}^{\infty} \Pi(t) e^{-2i\pi\nu t} dt = \int\limits_{-0.5}^{0.5} e^{-2i\pi\nu t} dt = -\frac{1}{2i\pi\nu} e^{-2i\pi\nu t} |_{-0.5}^{0.5} = -\frac{1}{2i\pi\nu}(e^{-i\pi\nu} - e^{i\pi\nu}) = \frac{e^{i\pi\nu} - e^{-i\pi\nu}}{2i} \frac{1}{\pi\nu} = \frac{\sin \pi\nu}{\pi\nu}$$

$$\hat{\Pi}(\nu) = sinc(\nu)$$

Построим графики функции и образа Фурье:

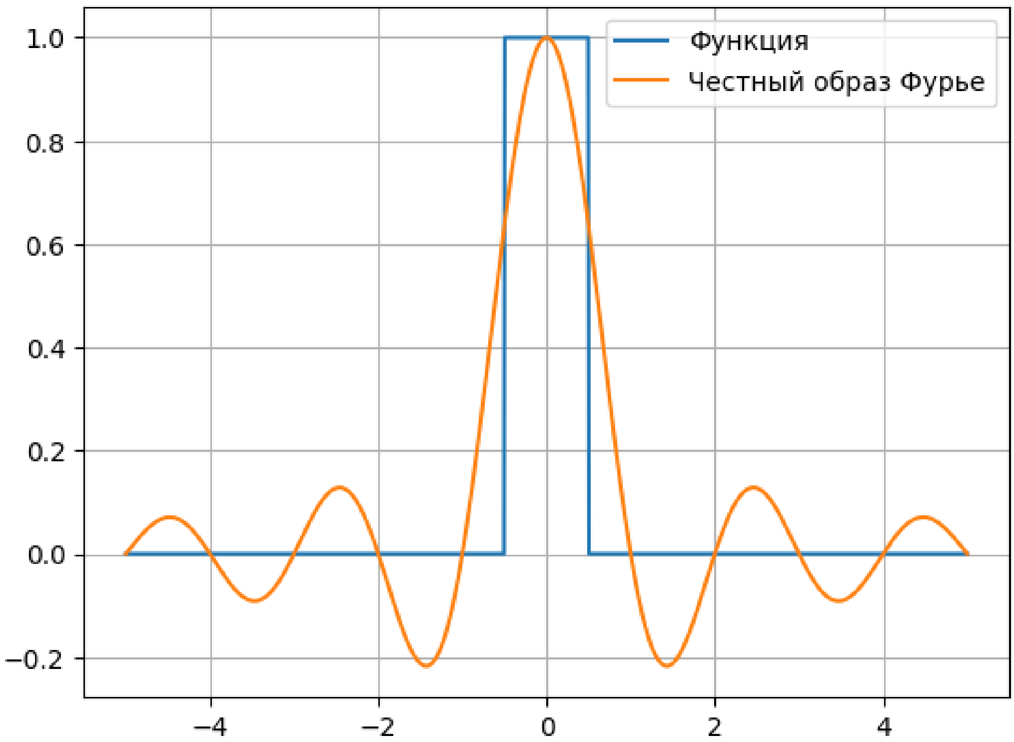

Рис. 1: Аналитически найденный образ

1.2 Образ через численное интегрирование

Посчитаем тоже честный образ, но через аппроксимацию - используем функцию numpy.trapz

Листинг 1.1 Фурье-образ численным интегрированием

```
def sq_wave(t):
    if (np.abs(t) <= 0.5):
        return 1
    return 0


def sq_wave_fourier_image(t):
    return np.sin(np.pi * t)/(np.pi * t)
```

```python
# apply fourier transofrm using numerical integration
def trapz_ft(function, time, dt):
    def integrand(nu):
        integral = lambda t: function(t) * np.exp(-2j * np.pi * nu * t)
        temp = [integral(i) for i in time]
        return np.trapz(temp, time)
    return lambda t: integrand(t)

#apply inverse fourier transform via numerical integration
def trapz_inverse_ft(ft_function, time, dt):
    def integrand(t):
        integral = lambda nu: ft_function(nu) * np.exp(2j * np.pi * nu * t)
        temp = [integral(i) for i in time]
        return np.trapz(temp, time)
    return lambda t: integrand(t)
```

Посмотрим на результаты:

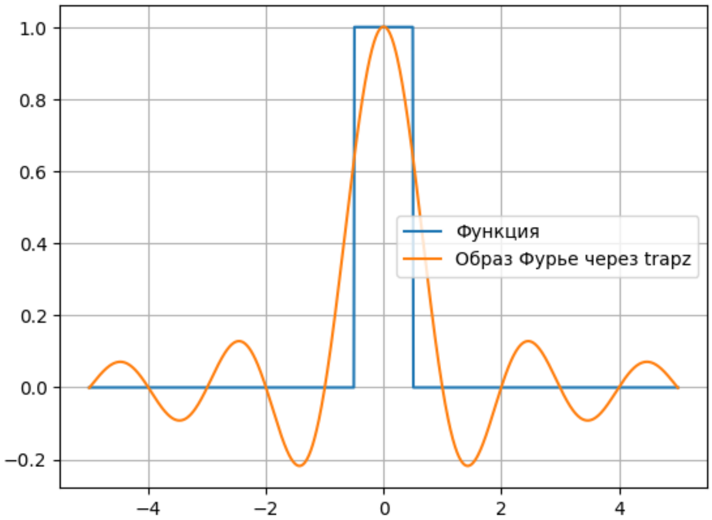

Рис. 2: Образ через численный подсчет

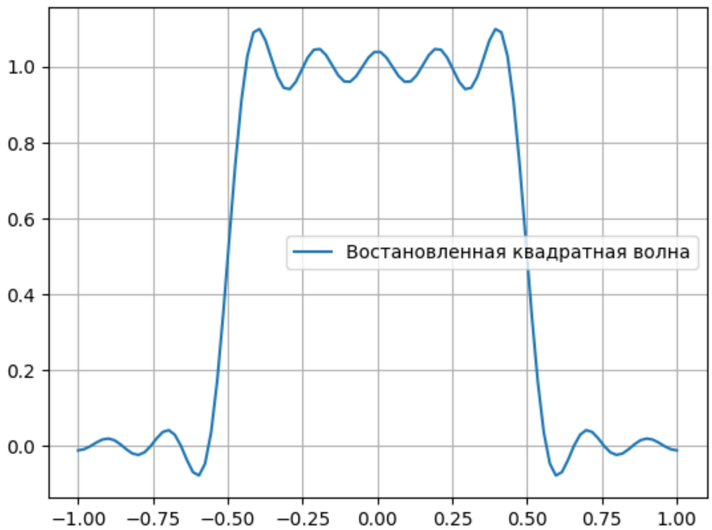

Рис. 3: Обратное Фурье преобразование

Образ Фурье был найден очень и очень точно, а с обратным возникли проблемы: во-первых, оно рассчитывалось минуты четыре, а во-вторых есть заметные отличия от исходной волны. Почему так произошло - будем разбираться позже, в отдельном пункте.

1.3 FFT

Воспользуемся встроенной в nupmy быстрым преобразованием Фурье (FFT) и выведем результат:

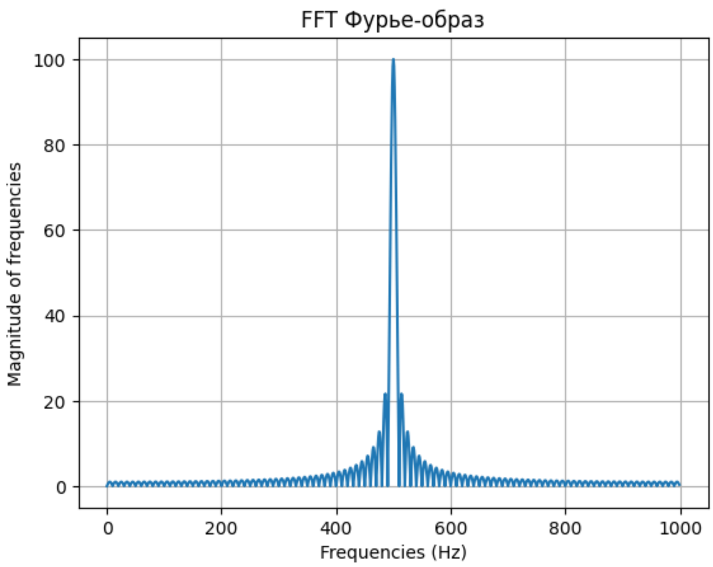

Рис. 4: Образ через DFT

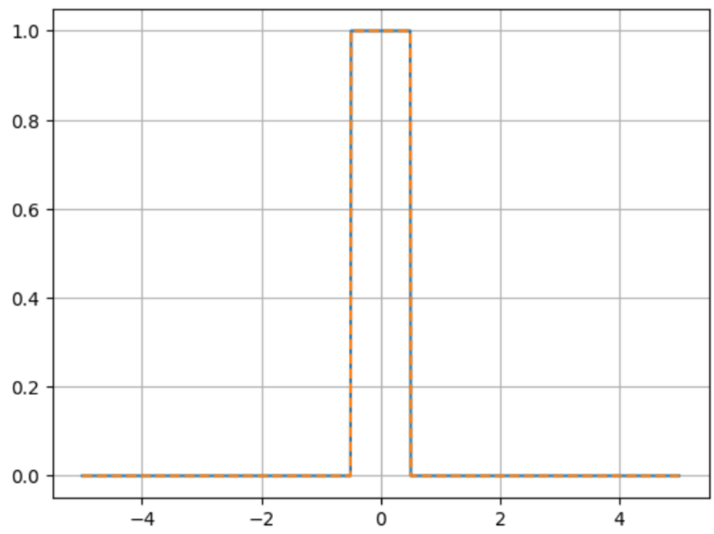

Рис. 5: Обратное Фурье преобразование IFFT

Видно, что получился точно не кардинальный синус. Зато обратное преобразование Фурье сработало идеально (пунктирный оранжевый сигнал - восстановленый, синий - исходный)

1.4 Почему так?

Отличие образов: аналитически найденный образ Фурье от квадратной волны - кардинальный синус. Кардинальный синус - это непрерывная функция. И используя численное интегрирование мы честно (с небольшой погрешностью) искали образ Фурье от непериодической функции непрерывного аргумента. В таком случае образом Фурье будет непрервная функци я - sinc, в данном случае. Используя дискретное преобразование Фурье (DFT) мы можем получить лишь функцию дискретного аргумента в качестве образа Фурье - то есть точно не кардинальный синус.

Отличие в обратном Фурье преобразовании

Видно, что DFT прекрасно справляется с задачей восстановления сигнала. Объясняется это видимо теоремой Найквиста-Шеннона-Котельникова. Что касается неудачи численного метода, я думаю, что причина его неточности в том, что обратное преобразование фурье должно выдавать непрерывную функцию, а квадратная волна имеет разрыв. Поэтому с помощью численного интегрирования мы можем лишь приблизить исходную квадратную волну, но не можем получить в её точности.

2 Непрерывный образ через DFT

3 Сэмплирование

3.1 Синусы

Листинг 2.1 Зададимся параметрами и построим сигналы

```
a1 = 2
a2  = 3
w1 = 0.3
```

```
w2 = .5
phi1 = pi/2
phi2= pi/4

y = lambda t: a1 * sin(w1 * t + phi1) + a2 * sin(w2 * t + phi2)
dt = 0.0001
time = np.linspace(-100, 100, int(1/dt))
short_time = np.linspace(-5*pi, 5*pi, int(1/dt))
Y = [y(t) for t in short_time]
```

Построим непрерывную функцию на небольшом промежутке, добавив сэмплированные точки (70 штук).

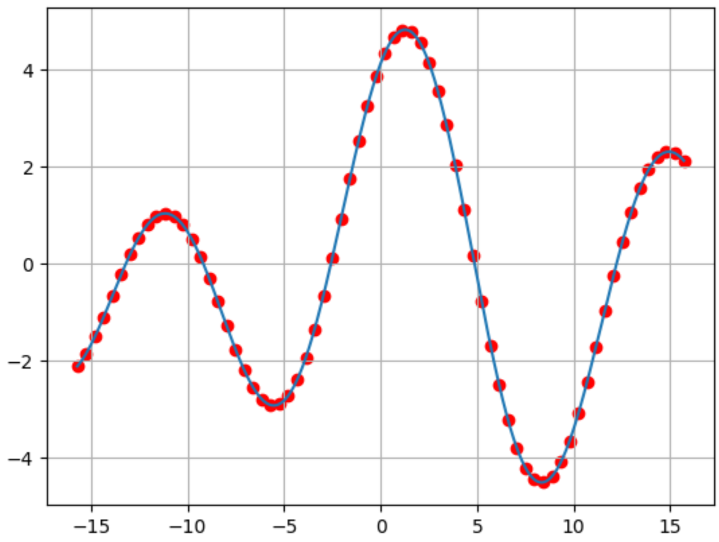

Рис. 6: Функция и ее дискретное разбиение, кол-во точек = 70

Примени дискретное преобразование Фурье

3.2 Кардинальные синусы