

## 2. Представление сигналов при помощи ряда Котельникова

Андрей Валиков

### 1 Построение заданного сигнала

Исходный сигнал:

$$x(t) = \begin{cases} Ae^{-t/\tau} & \text{if } 0 \leq t \leq T_c \\ 0 & \text{else} \end{cases}$$

Константы имеют следующие значения:

$$A = 2$$

$$\tau = 0.5$$

$$T_c = 0.7$$

$$N = 800$$

$$T = .3$$

$$T_{\max} = 1$$

$$\text{om}_{\max} = 100$$

$$t = \text{np.linspace}(0, T_{\max}, N)$$

$$\text{om} = \text{np.linspace}(0, \text{om}_{\max}, N)$$

$$n = 7$$

$$\text{func} = \text{np.array}(\text{list}(\text{map}(\text{lambda } x: \_ \text{func}(x, T_{\max}), t)))$$

$$\text{sp}_1 = \text{plt.subplot}(221)$$

$$\text{sp}_1.\text{plot}(t, \text{func}, \text{color}=(0, .9, 0))$$

## 2 Расчет спектральной характеристики

Используется тригонометрическая форма преобразования Фурье:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} x(t) \cos(\omega t) dt - i \int_{-\infty}^{\infty} x(t) \sin(\omega t) dt$$

Модуль полученной спектральной характеристики будет составлять амплитудную характеристику сигнала:

$$A(\omega) = |\hat{f}|$$

```
amp_func = np.abs(_fourier(func, t, om))
sp_2 = plt.subplot(223)
sp_2.plot(om, amp_func, color=(0, .5, 0))
```

Рис. 1:

Рис. 2:

## 3 Отсчётная функция и её спектральная характеристика

$$\phi(t) = \frac{\sin \omega_r(t - nT)}{(t - nT)}$$

```
count_f = CDF(t, T, n)
sp_3 = plt.subplot(222)
sp_3.plot(t, count_f, color=(0, 0, .9))

fou = _fourier(count_f, t, om)
sp_4 = plt.subplot(224)
sp_4.plot(t, np.real(fou), color=(0, 0, .5))
```

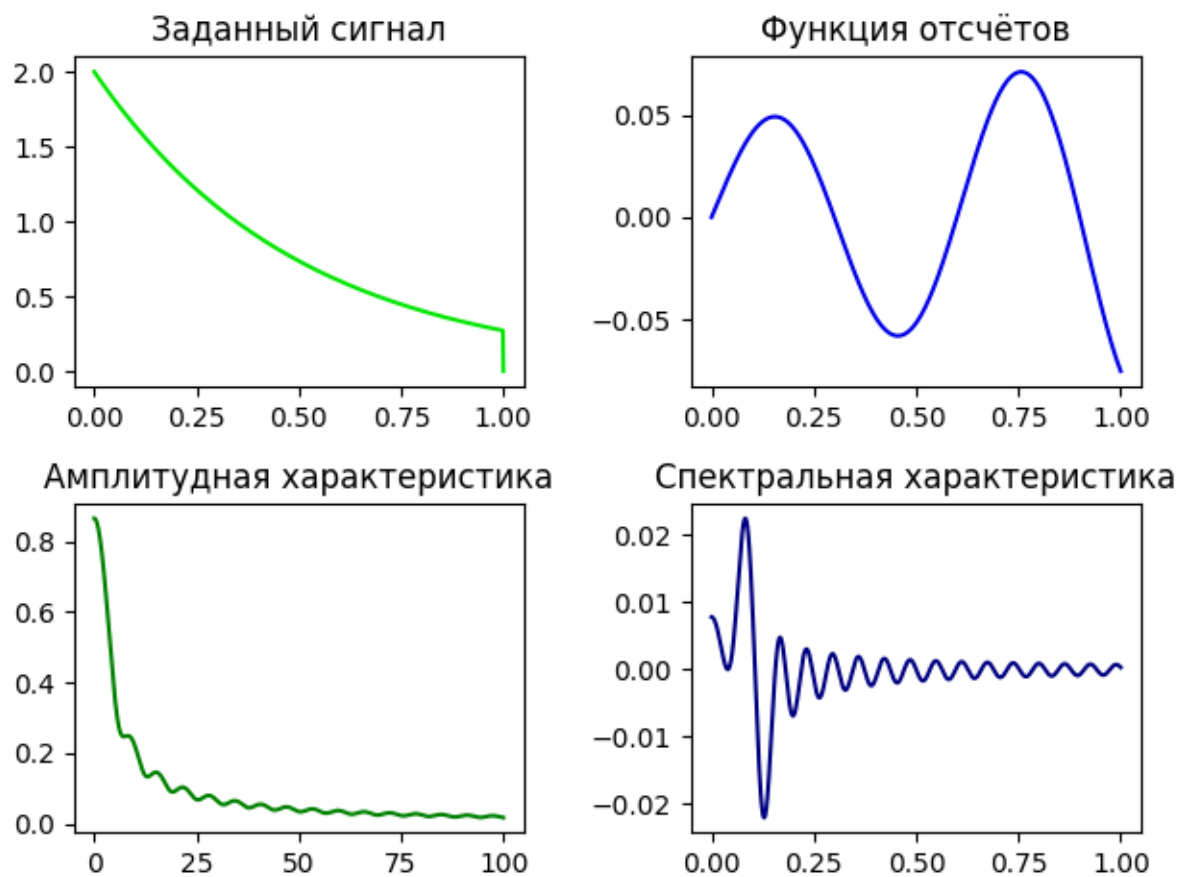


Рис. 3:

## 4 Определение граничной частоты

```
max_amp = .05 * max(amp_func)
for i in range(N):
    if amp_func[i] <= max_amp:
        freq = om[i]

print(freq)
```

Она оказывается равной 100.

## 5 Определение периода дискретизации и построение аппроксимирующего сигнала

```
T_disc = np.pi / freq
NN = int(np.fix(T_max / T_disc))

func_recovered = np.zeros(N)
for i in range(N):
    for nn in range(NN + 1):
        func_recovered[i] += _func(T * (nn - 1), T_max) * CDF(t[i], T, nn - 1)

plt.plot(t, func)
plt.plot(t, func_recovered)
plt.show()
```

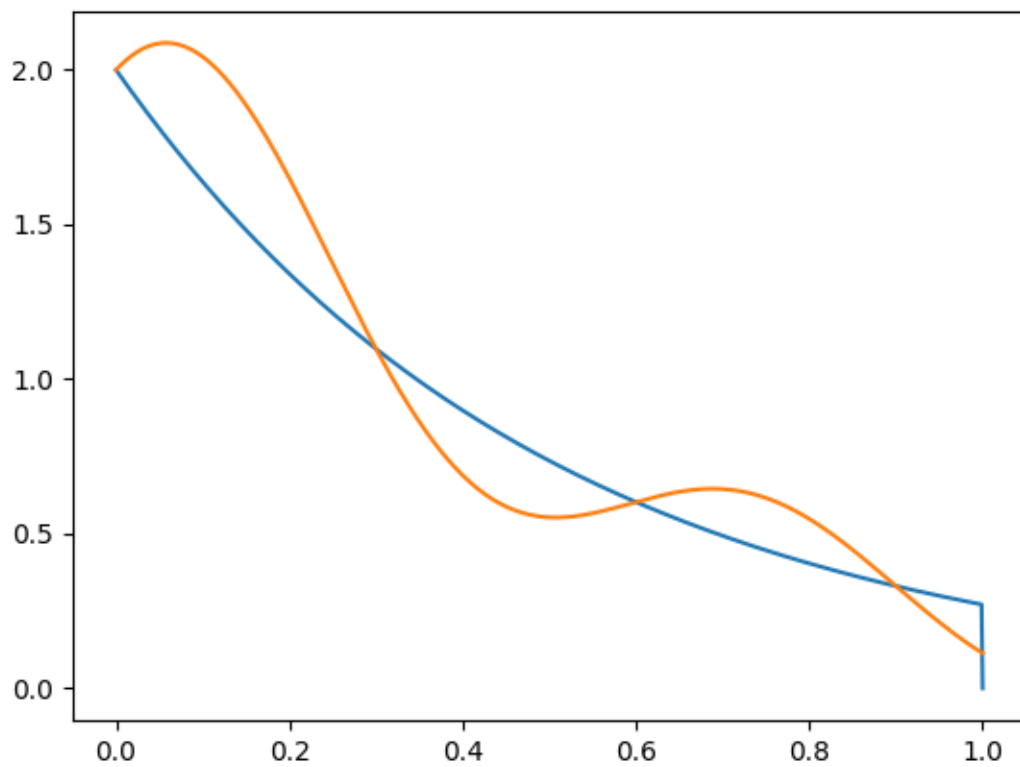


Рис. 4:

## 6 Нахождение сигнала ошибки

`err = abs(func - func_recovered)`

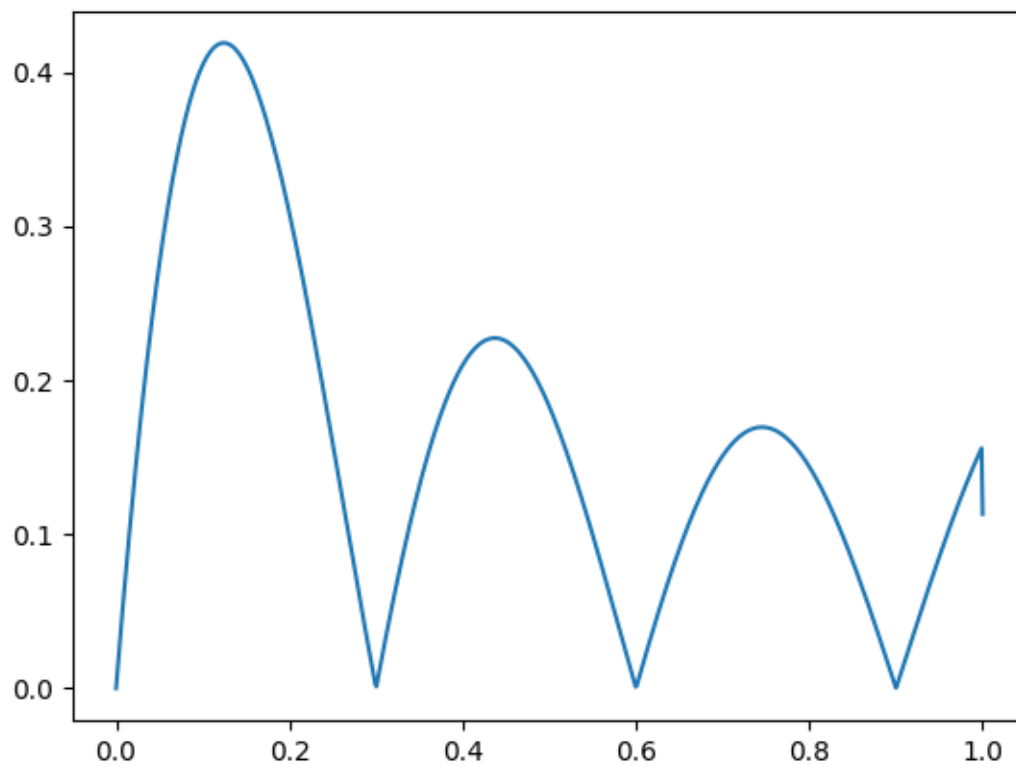


Рис. 5: