

УДК 621.38

Составитель: П.А. Безмен

Рецензент

Кандидат технических наук, доцент Юго-Западного государственного
университета *Е.Н. Политов*

Сенсорные системы и методы обработки сигналов: методические указания по выполнению практических и самостоятельных работ для студентов направления 15.03.06 «Мехатроника и робототехника» / Юго-Зап. гос. ун-т; сост. П.А. Безмен. Курск, 2022. 45 с.

Изложены задания для выполнения на практических занятиях и задания для самостоятельной работы студентов по дисциплине «Сенсорные системы и методы обработки сигналов».

Методические указания соответствуют требованиям Федерального государственного образовательного стандарта.

Методические указания предназначены для студентов направления 15.03.06 «Мехатроника и робототехника», а также других направлений технического профиля для всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать 10.02.2022. . Формат 60x84 1/16.
Уч.-изд.л.24. Тираж 30 экз. Заказ 737. Бесплатно.
Юго-Западный государственный университет.
305040 Курск, ул. 50 лет Октября, 94.

Усл.печ.л. 2,6.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ.....	4
1 ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	5
2 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНЫХ РАБОТ	29
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.....	45

ПРЕДИСЛОВИЕ

Цель курса «Сенсорные системы и методы обработки сигналов» – сформировать у студентов направления 15.03.06 «Мехатроника и робототехника» навыки проектирования информационных систем роботов и мехатронных модулей, а также навыки работы с существующими информационными системами. Студент должен приобрести навыки решения таких задач, как выбор набора сенсоров, необходимого для получения нужной информации о состоянии робота и окружающей среды, сбор и обработка информации, поступающей от сенсорной системы робота, обеспечение взаимодействия сенсорной системы робота и его системы автоматического управления и др. Дисциплина формирует знания и навыки, необходимые для дальнейшей деятельности студента в качестве исследователя, инженера-конструктора, инженера-робототехника и в других видах научно-исследовательской и инженерной деятельности по изучению и освоению новых систем.

Задачи дисциплины

Основными задачами изучения дисциплины «Сенсорные системы и методы обработки сигналов» являются:

- изучение базовых принципов проектирования информационных систем роботов;
- рассмотрение моделей и алгоритмов обработки сигналов, применительно к сенсорным системам в области мехатроники и робототехники;
- рассмотрение особенностей практического приложения методов теории информационных систем роботов к частным инженерным и исследовательским задачам с учетом будущей специальности.

Практические занятия по дисциплине «Сенсорные системы и методы обработки сигналов» направления 15.03.06 «Мехатроника и робототехника» проводятся с целью практического закрепления знаний, получаемых студентами в лекционном курсе, и выполняются в специализированных лабораториях кафедры механики, мехатроники и робототехники Юго-Западного государственного университета.

Основная цель самостоятельной работы студента при изучении дисциплины закрепить теоретические знания, полученные в процессе лекционных занятий, а также сформировать практические навыки самостоятельного анализа особенностей дисциплины.

Предлагаемое пособие содержит задачи для решения на практических занятиях и задания для самостоятельной работы студентов.

1 ПРАКТИЧЕСКИЕ ЗАНЯТИЯ

Практическая работа 1. Фильтрация сигнала

Методика выполнения работы

Составим в среде Mathcad сигнал сложив две функции синуса с частотами 1 Гц и 50 Гц и амплитудой 10:

$$f1 := 1$$

$$f2 := 50$$

$$A := 10$$

Зададим общее время моделирования сигнала N в секундах:

$$N := 2$$

Зададим шаг времени $step$ в секундах для формирования сигнала:

$$step := 0.001$$

Добавим сигналу шум, используя функцию $rnd(A)$, возвращающую случайную величину в диапазоне от 0 до A , и зададим переменную $noise$, включающую шум при значении 1 и более:

$$noise := 1$$

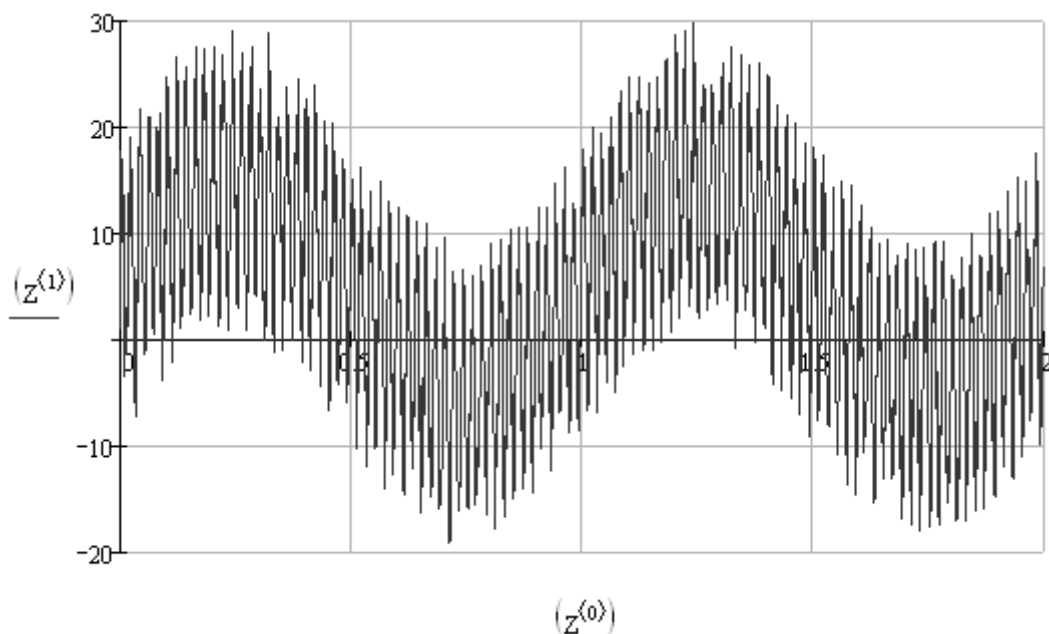
Функция S сигнала от времени t примет вид:

$$S(t) := [\sin[t \cdot (f1) \cdot 2 \cdot \pi] + \sin[t \cdot (f2) \cdot 2 \cdot \pi]] \cdot A + noise \cdot rnd(A)$$

Составим процедуру формирования дискретного сигнала:

$$Z := \begin{cases} n \leftarrow 0 \\ \text{while } n \leq \left(\frac{N}{step} \right) \\ \quad \left| \begin{array}{l} Z_{n,0} \leftarrow step \cdot n \\ Z_{n,1} \leftarrow S(step \cdot n) \\ n \leftarrow n + 1 \end{array} \right. \\ \text{return } Z \end{cases}$$

В результате получим двумерный массив Z с двумя столбцами: $Z^{<0>}$ - столбец значений времени, $Z^{<1>}$ - столбец значений сигнала. Далее построим изображение сигнала на временной диаграмме.



Получим спектр сигнала, используя функцию *fft* – дискретное преобразование Фурье (таблица П4 приложения 6):

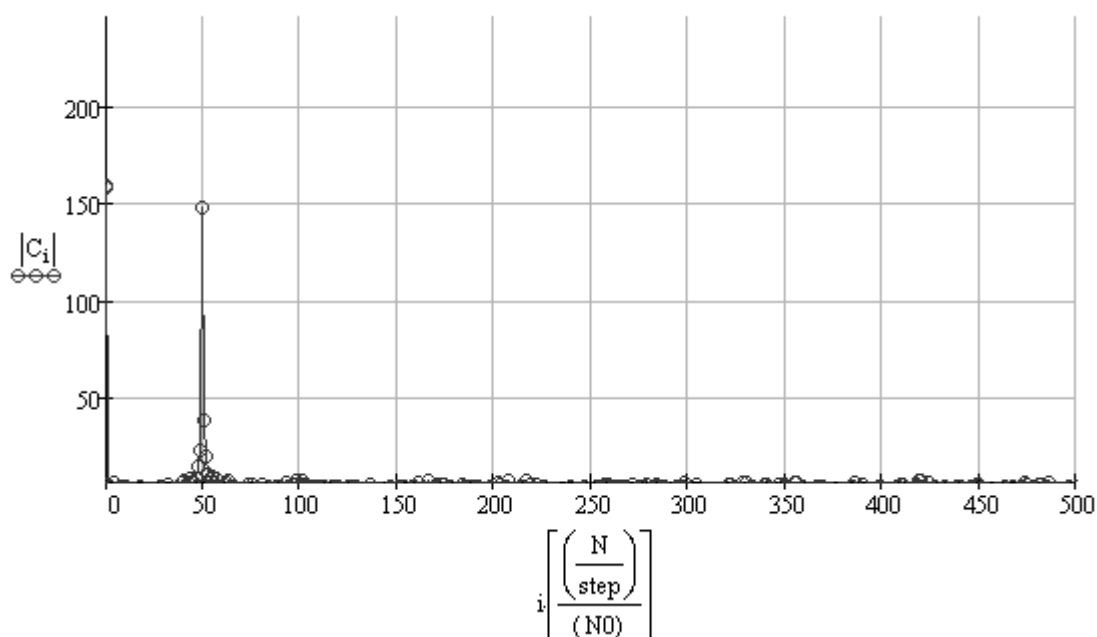
```
NO := 2048
```

```
i := 0..trunc( $\frac{NO - 1}{2}$ )
```

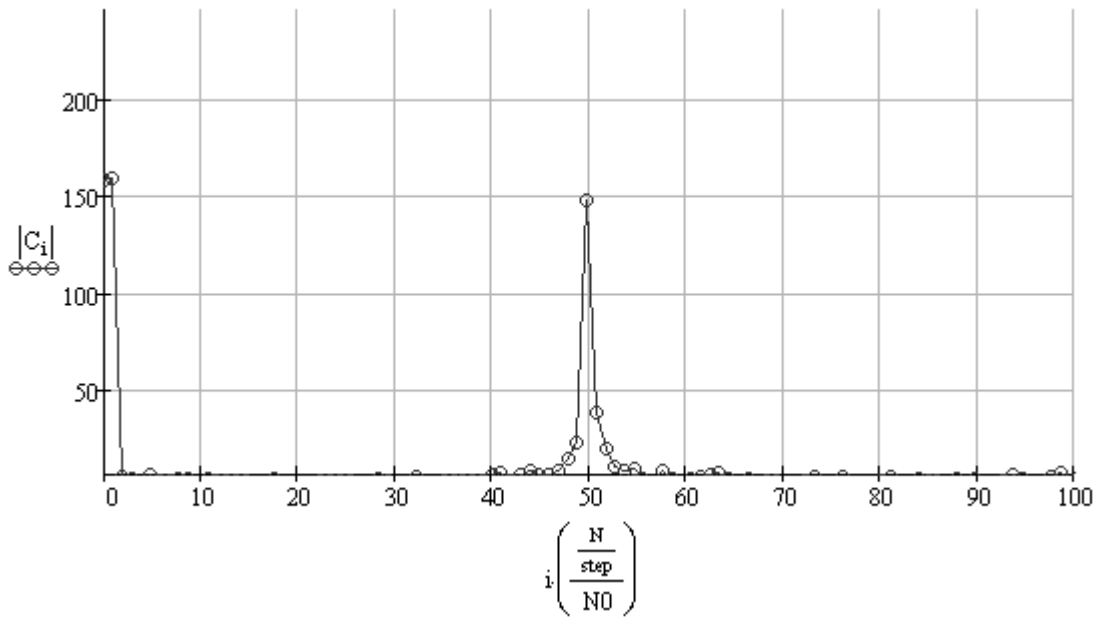
```
ZZi := [z^{(1)}i]
```

```
C := fft(ZZ)
```

После использования функции *fft* массив *C* содержит данные спектра сигнала:



Рассмотрим детально полученный спектр сигнала в диапазоне от 0 до 100 Гц – на изображении спектра видны два пика, соответствующие составляющим сигнала с частотами 1 Гц и 50 Гц:



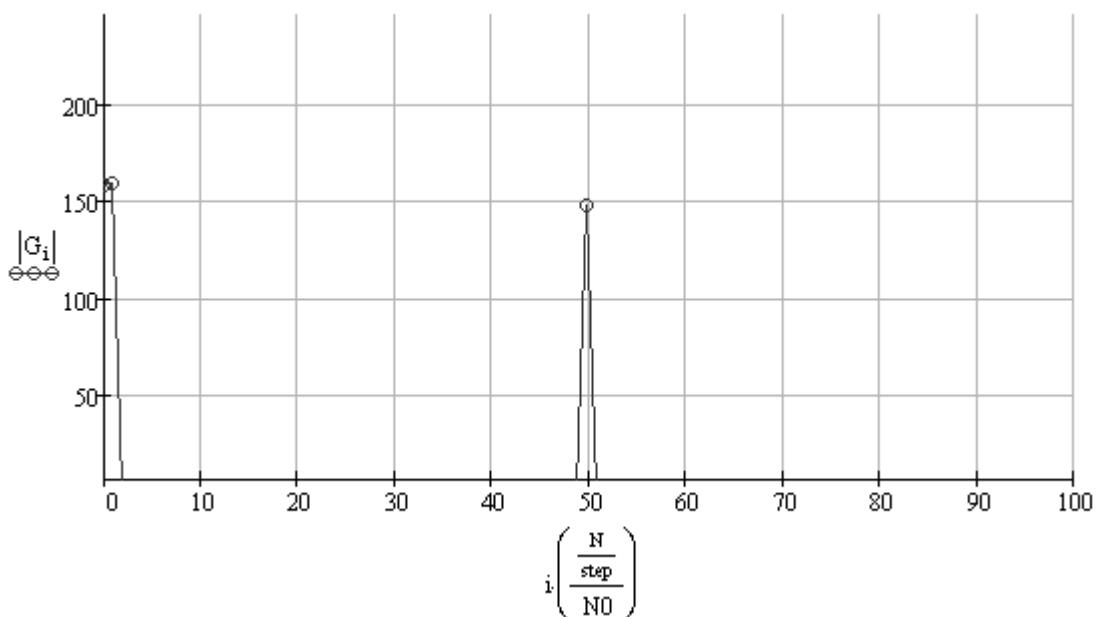
Введем функцию отсечки спектра сигнала (A – массив исходного спектра сигнала, min – порог отсечки спектра сигнала):

$$\text{filt}(A, min) := \begin{cases} n \leftarrow 0 \\ \text{while } n \leq \frac{\left(\text{trunc} \left(\frac{NO}{2} \right) \right)}{2} \\ \quad \left| \begin{array}{l} A_n \leftarrow 0 \text{ if } |A_n| \leq min \\ n \leftarrow n + 1 \end{array} \right. \\ A \end{cases}$$

Уберем из сигнала шум – сформируем новый массив G :

$$G := \text{filt}(C, 100)$$

Изобразим полученный спектр сигнала:

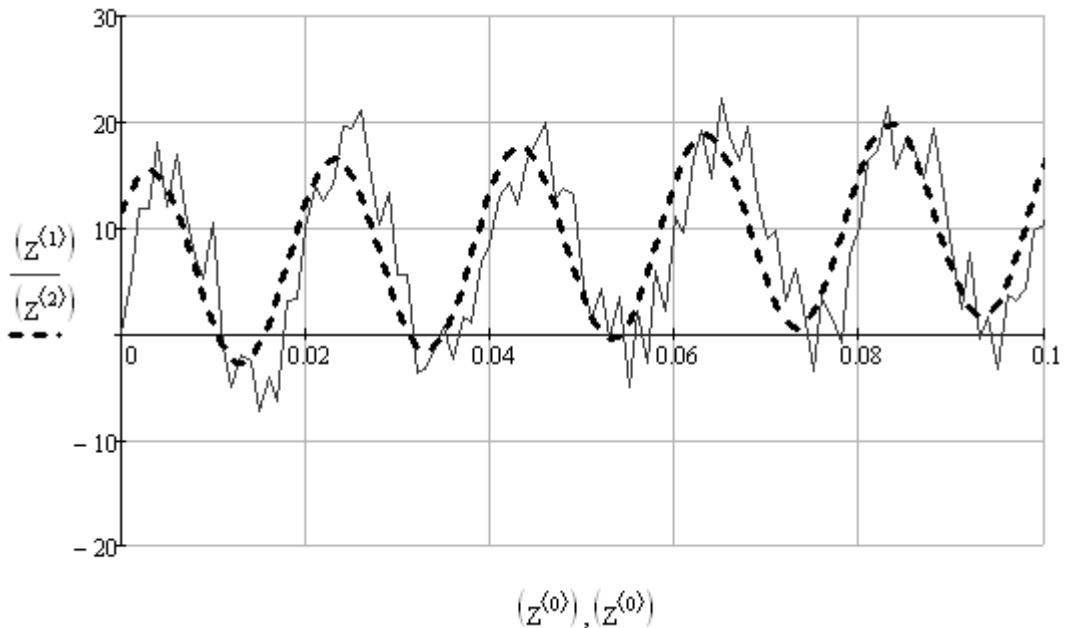


Воспользуемся функцией *ifft* – обратным дискретным преобразованием Фурье (таблица П4 приложения 6) для получения сигнала из его спектра:

$$H := \text{ifft}(G)$$

$$Z^{(2)} := H$$

В результате получим двумерный массив Z с тремя столбцами: $Z^{(0)}$ - столбец значений времени, $Z^{(1)}$ - столбец значений сигнала, $Z^{(2)}$ - столбец значений отфильтрованного от шума сигнала:



Варианты заданий

1. Составить в среде Mathcad сигнал, сложив функции синуса с частотами f_i Гц и амплитудами A_i соответственно, следуя варианту задания (таблица 1), номер варианта задания равен порядковому номеру студента в списке группы.
2. Сформировать дискретный сигнал.
3. Используя преобразование Фурье и обратное преобразование Фурье выполнить фильтрацию сигнала в соответствии с номером варианта задания.
4. Выполнить построение исходного сигнала и сигнала, полученного после фильтрации.
5. Оформить отчет о работе.

Таблица 1.1 – Варианты заданий

Вариант	Задание
1	Частоты и амплитуды складывающихся сигналов: $f_1 = 15$ Гц, $A_1 = 10$, $f_2 = 36$ Гц, $A_2 = 5$. Отфильтровать сигнал от частот выше 20 Гц
2	Частоты и амплитуды складывающихся сигналов: $f_1 = 500$ Гц, $A_1 = 5$, $f_2 = 1050$ Гц, $A_2 = 10$. Отфильтровать сигнал от частот ниже 600 Гц
3	Частоты и амплитуды складывающихся сигналов: $f_1 = 1$ Гц, $A_1 = 10$, $f_2 = 5$ Гц, $A_2 = 5$, $f_3 = 10$ Гц, $A_3 = 10$. Отфильтровать сигнал от частот от 4 до 6 Гц
4	Частоты и амплитуды складывающихся сигналов: $f_1 = 5$ Гц, $A_1 = 10$, $f_2 = 10$ Гц, $A_2 = 5$, $f_3 = 50$ Гц, $A_3 = 10$. Отфильтровать сигнал от частот от 9 до 11 Гц
5	частоты и амплитуды складывающихся сигналов: $f_1 = 5$ Гц, $A_1 = 5$, $f_2 = 10$ Гц, $A_2 = 10$, $f_3 = 50$ Гц, $A_3 = 5$. Выделить сигнал с частотами от 9 до 11 Гц
6	Частоты и амплитуды складывающихся сигналов: $f_1 = 1$ Гц, $A_1 = 5$, $f_2 = 3$ Гц, $A_2 = 10$. Отфильтровать сигнал от частот ниже 2,5 Гц
7	Частоты и амплитуды складывающихся сигналов: $f_1 = 0,5$ Гц, $A_1 = 10$, $f_2 = 1$ Гц, $A_2 = 5$. Отфильтровать сигнал от частот выше 0,8 Гц
8	Частоты и амплитуды складывающихся сигналов: $f_1 = 1$ Гц, $A_1 = 10$, $f_2 = 10$ Гц, $A_2 = 5$, $f_3 = 100$ Гц, $A_3 = 10$, $f_4 = 1000$ Гц, $A_4 = 5$, $f_5 = 10000$ Гц, $A_5 = 10$. Отфильтровать сигнал от частот от 9 до 11 Гц и от частот от 999 Гц до 1001 Гц
9	Частоты и амплитуды складывающихся сигналов: $f_1 = 1$ Гц, $A_1 = 5$, $f_2 = 100$ Гц, $A_2 = 5$, $f_3 = 500$ Гц, $A_3 = 5$, $f_4 = 1000$ Гц, $A_4 = 10$, $f_5 = 5000$ Гц, $A_5 = 5$. Выделить сигнал с частотами от 999 Гц до 1001 Гц
10	Частоты и амплитуды складывающихся сигналов: $f_1 = 3$ Гц, $A_1 = 10$, $f_2 = 50$ Гц, $A_2 = 5$, $f_3 = 1000$ Гц, $A_3 = 5$. Выделить сигнал с частотами от 2 до 4 Гц
11	Частоты и амплитуды складывающихся сигналов: $f_1 = 1$ Гц, $A_1 = 5$, $f_2 = 10$ Гц, $A_2 = 10$, $f_3 = 100$ Гц, $A_3 = 5$. Выделить сигнал с частотами от 9 до 11 Гц
12	Частоты и амплитуды складывающихся сигналов: $f_1 = 10$ Гц, $A_1 = 5$, $f_2 = 500$ Гц, $A_2 = 20$, $f_3 = 1000$ Гц, $A_3 = 10$. Выделить сигнал с частотами от 499 до 501 Гц
13	Частоты и амплитуды складывающихся сигналов: $f_1 = 10$ Гц, $A_1 = 10$, $f_2 = 20$ Гц, $A_2 = 5$, $f_3 = 30$ Гц, $A_3 = 15$, $f_4 = 40$ Гц, $A_4 = 5$, $f_5 = 50$ Гц, $A_5 = 10$. Выделить сигнал с частотами от 29 до 31 Гц
14	Частоты и амплитуды складывающихся сигналов: $f_1 = 2$ Гц, $A_1 = 10$, $f_2 = 10$ Гц, $A_2 = 5$, $f_3 = 100$ Гц, $A_3 = 5$. Отфильтровать сигнал от частот выше 5 Гц
15	Частоты и амплитуды складывающихся сигналов: $f_1 = 20$ Гц, $A_1 = 5$, $f_2 = 25$ Гц, $A_2 = 15$, $f_3 = 30$ Гц, $A_3 = 10$. Выделить сигнал с частотами от 24 до 26 Гц

Таблица 1.1 – Варианты заданий (продолжение)

16	Частоты и амплитуды складывающихся сигналов: $f_1 = 1$ Гц, $A_1 = 10$, $f_2 = 10$ Гц, $A_2 = 5$, $f_3 = 50$ Гц, $A_3 = 10$, $f_4 = 100$ Гц, $A_4 = 5$. Отфильтровать сигнал от частот от 9 до 11 Гц и от частот выше 50 Гц
17	Частоты и амплитуды складывающихся сигналов: $f_1 = 2$ Гц, $A_1 = 8$, $f_2 = 4$ Гц, $A_2 = 10$. Отфильтровать сигнал от частот ниже 4 Гц
18	Частоты и амплитуды складывающихся сигналов: $f_1 = 100$ Гц, $A_1 = 2$, $f_2 = 400$ Гц, $A_2 = 1$. Отфильтровать сигнал от частот выше 150 Гц
19	Частоты и амплитуды складывающихся сигналов: $f_1 = 3$ Гц, $A_1 = 10$, $f_2 = 8$ Гц, $A_2 = 1$, $f_3 = 10$ Гц, $A_3 = 5$. Отфильтровать сигнал от частот от 7 до 9 Гц
20	Частоты и амплитуды складывающихся сигналов: $f_1 = 15$ Гц, $A_1 = 5$, $f_2 = 20$ Гц, $A_2 = 10$. Отфильтровать сигнал от частот ниже 20 Гц

Практическая работа 2. Дифференцирование сигнала

Методика выполнения работы

Составим в среде Mathcad цифровой треугольный сигнал, кусочно задав функцию U :

$$\begin{aligned}
 U(t, t_1, t_2, t_3, t_4, U_1, U_2) := & \left[\begin{array}{l}
 v \leftarrow \frac{2 \cdot U_1}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{3 \cdot U_1}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{4 \cdot U_1}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow U_1 \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow U_1 \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{4 \cdot U_1}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{3 \cdot U_1}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{2 \cdot U_1}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{U_1}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{2 \cdot U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{3 \cdot U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{4 \cdot U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow U_2 \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow U_2 \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{4 \cdot U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{3 \cdot U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{2 \cdot U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{U_2}{5} \text{ if mod} \left[t \cdot \left(\frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[\frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 v \leftarrow \frac{U_1}{5} \text{ otherwise}
 \end{array} \right.
 \end{aligned}$$

Определим следующие величины:

T_1 – длительность первой четверти сигнала U в секундах:

$$T_1 := 0.040$$

$U1$ – амплитуда сигнала U в первом полупериоде:

$$U1 := 12$$

$T2$ – длительность второй четверти сигнала U в секундах:

$$T2 := 0.040$$

$U2$ – амплитуда сигнала U во втором полупериоде:

$$U2 := -12$$

$T3$ – длительность третьей четверти сигнала U в секундах:

$$T3 := 0.040$$

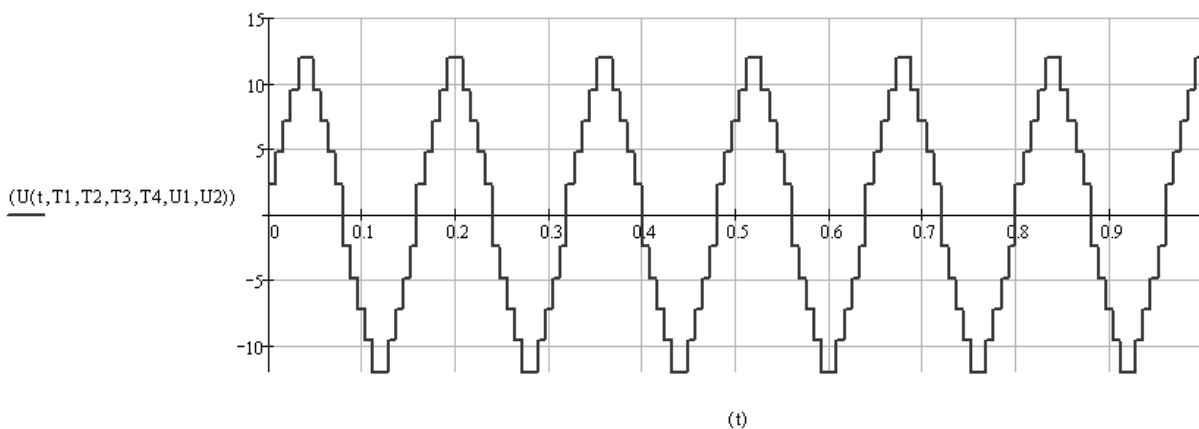
$T4$ – длительность четвертой четверти сигнала U в секундах:

$$T4 := 0.040$$

Общее время периода T составит:

$$T := T1 + T2 + T3 + T4$$

Далее построим изображение сигнала U на временной диаграмме:



Зададим функцию $FtoA$ для перевода функции сигнала U в двумерный массив, состоящий из двух столбцов, где первый столбец – столбец значений времени, второй – столбец значений дискретного сигнала. Аргументы функции $FtoA$: $totaltime$ – общее время моделирования сигнала, сек., $points$ – количество отсчетов (точек). $Step$ – шаг времени, сек.

```

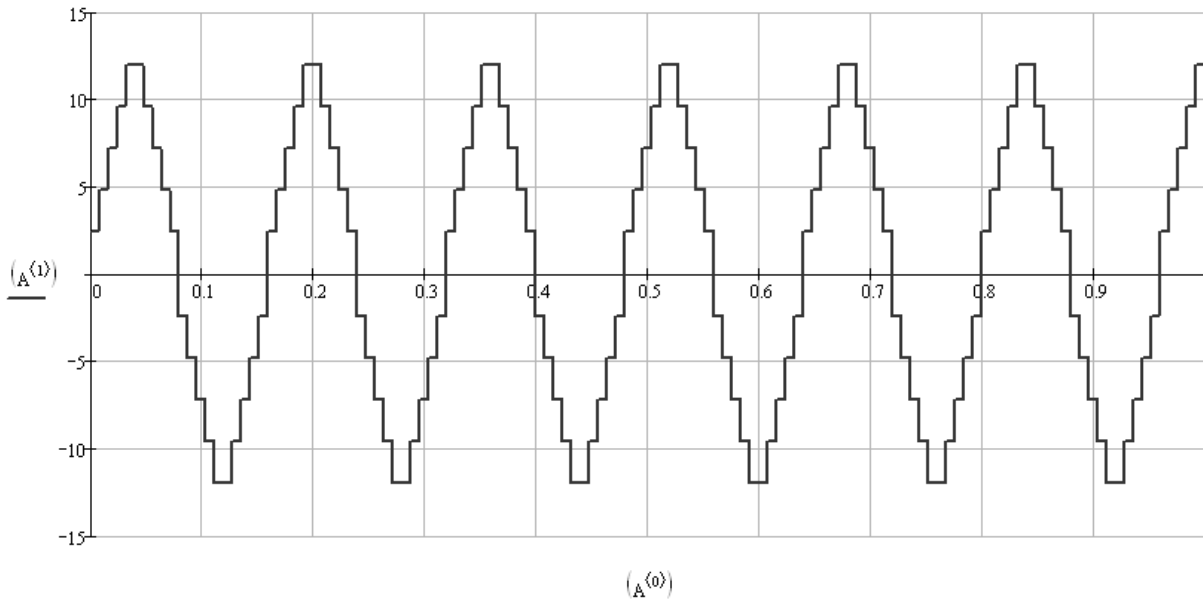
FtoA(totaltime, points) :=
  Z ← 0
  n ← 0
  step ← totaltime / points
  while n ≤ points
    Zn,0 ← step · n
    Zn,1 ← U[Zn,0, T1, T2, T3, T4, U1, U2]
    n ← n + 1
  return Z

```

Сформируем массив A со значениями сигнала U :

$$A := \text{FtoA}(1, 10000)$$

Далее построим изображение дискретного сигнала на временной диаграмме:



К полученному дискретному сигналу применим метод сглаживания, использующий алгоритм нахождения скользящего среднего значения сигнала:

$$w := 250$$

$$n := \text{rows}(A)$$

$$n = 1 \times 10^4$$

$$i := 0..n - 1$$

$$z1_i := \begin{cases} \frac{\sum_{j=0}^i (A^{(1)})_j}{i+1} & \text{if } i < w \\ \frac{\sum_{j=i-w+1}^i (A^{(1)})_j}{w} & \text{otherwise} \end{cases},$$

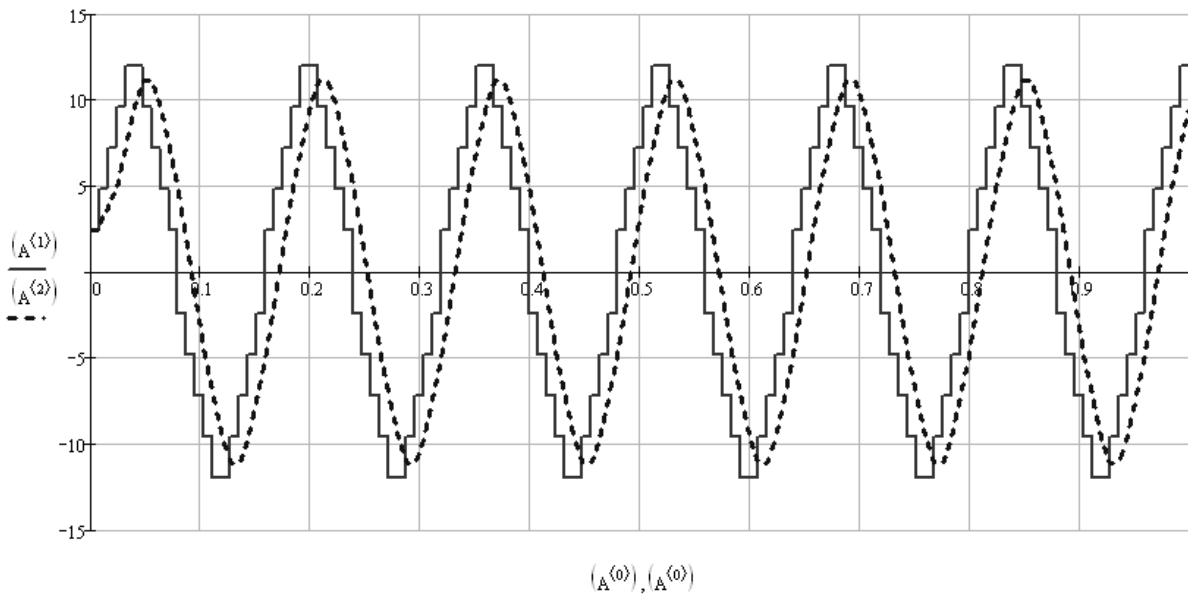
где: A_j – массив A исходного (несглаженного) сигнала со значением сигнала в точке времени j , i – текущая обрабатываемая точка времени, w – константа, определяющая при сглаживании пропуск точек времени в количестве w ($0 < w < n$, n – общее количество точек времени (элементов массива A)), $z1$ – массив со сглаженным уровнем сигнала.

Полученный одномерный массив (матрицу-столбец) $z1$ присвоим третьему столбцу $A^{(2)}$:

$$A^{(2)} := z1$$

	0	1	2
0	0	2.4	2.4
1	1·10 ⁻⁴	2.4	2.4
2	2·10 ⁻⁴	2.4	2.4
3	3·10 ⁻⁴	2.4	2.4
4	4·10 ⁻⁴	2.4	2.4
5	5·10 ⁻⁴	2.4	2.4
6	6·10 ⁻⁴	2.4	2.4
7	7·10 ⁻⁴	2.4	2.4
8	8·10 ⁻⁴	2.4	2.4
9	9·10 ⁻⁴	2.4	2.4
10	1·10 ⁻³	2.4	2.4
11	1.1·10 ⁻³	2.4	2.4
12	1.2·10 ⁻³	2.4	2.4
13	1.3·10 ⁻³	2.4	2.4
14	1.4·10 ⁻³	2.4	2.4
15	1.5·10 ⁻³	2.4	2.4

Далее построим на одной временной диаграмме изображения несглаженного и сглаженного дискретных сигналов:



Зададим функцию *Diff* для получения дискретного продифференцированного по времени сигнала U , где: S – двумерный массив с исходным сигналом, N – номер столбца в массиве S с данными для дифференцирования ($0 < N$):

```

Diff(S, N) :=
  Y ← 0
  t ← 0
  a ← 0
  a0 ← 0
  for Y ∈ 0..last(S<sup>(0)</sup>)
    dt ← SY,0 - t
    t ← SY,0
    a0 ← a
    a ← SY,N
    dt ← 0.000000000000001 if dt ≤ 0
    SY,N+1 ← (a - a0) / dt
  S

```

Продифференцируем данные в третьем столбце $A^{<2>}$ и получим в том же массиве A четвертый столбец $A^{<3>}$ с продифференцированным дискретным сигналом U :

```
A := Diff(A,2)
```

К полученному дискретному сигналу в четвертом столбце $A^{<3>}$ применим метод сглаживания:

```
w := 250
```

```
n := rows(A)
```

```
n = 1 × 104
```

```
i := 0..n - 1
```

$$z1_i := \begin{cases} \frac{\sum_{j=0}^i (A^{(3)})_j}{i+1} & \text{if } i < w \\ \frac{\sum_{j=i-w+1}^i (A^{(3)})_j}{w} & \text{otherwise} \end{cases}$$

$$A^{(3)} := z1$$

Продифференцируем данные в четвертом столбце $A^{<3>}$ и получим в том же массиве A пятый столбец $A^{<4>}$ с дважды продифференцированным дискретным сигналом U :

```
A := Diff(A,3)
```

К полученному дискретному сигналу в пятом столбце $A^{<4>}$ применим метод сглаживания:

```
w := 300
```

```
n := rows(A)
```

```
n = 1 × 104
```

```
i := 0..n - 1
```

$$z1_i := \begin{cases} \frac{\sum_{j=0}^i (A^{(4)})_j}{i+1} & \text{if } i < w \\ \frac{\sum_{j=i-w+1}^i (A^{(4)})_j}{w} & \text{otherwise} \end{cases}$$

$$A^{(4)} := z1$$

Обрезаем левый край сигнала, удаляя первые (верхние) 500 строк из массива A (тем самым удаляя первые 500 точек дискретного сигнала):

CUT := 550

TIMESHIFT := $A_{CUT,0}$

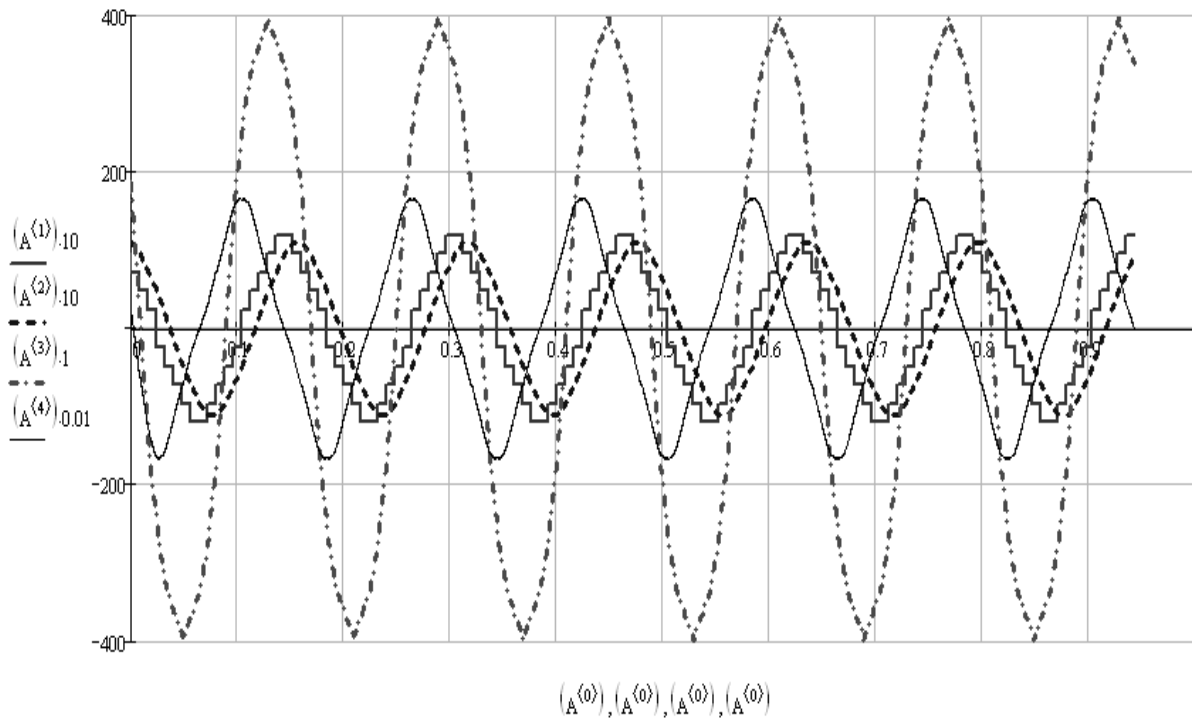
$A := \text{submatrix}[A, \text{CUT}, (\text{last}(A^{(0)})), 0, 4]$

Смещаем влево шкалу времени после обрезки дискретного сигнала:

$A^{(0)} := (A^{(0)}) - \text{TIMESHIFT}$

	0	1	2	3	4
0	0	9.6	11.136	188.16	2.816·10 ³
1	1·10 ⁻⁴	9.6	11.136	187.008	2.739·10 ³
2	2·10 ⁻⁴	9.6	11.136	185.856	2.662·10 ³
3	3·10 ⁻⁴	9.6	11.136	184.704	2.586·10 ³
4	4·10 ⁻⁴	9.6	11.136	183.552	2.509·10 ³
5	5·10 ⁻⁴	9.6	11.136	182.4	2.432·10 ³
6	6·10 ⁻⁴	9.6	11.136	181.248	2.355·10 ³
7	7·10 ⁻⁴	9.6	11.136	180.096	2.278·10 ³
8	8·10 ⁻⁴	9.6	11.136	178.944	2.202·10 ³
9	9·10 ⁻⁴	9.6	11.136	177.792	2.125·10 ³
10	1·10 ⁻³	9.6	11.136	176.64	2.048·10 ³
11	1.1·10 ⁻³	7.2	11.126	175.104	1.958·10 ³
12	1.2·10 ⁻³	7.2	11.117	173.568	1.869·10 ³
13	1.3·10 ⁻³	7.2	11.107	172.032	1.779·10 ³
14	1.4·10 ⁻³	7.2	11.098	170.496	1.69·10 ³
15	1.5·10 ⁻³	7.2	11.088	168.96	1.6·10 ³

Построим на одной временной диаграмме изображения несглаженного и сглаженного дискретных сигналов U , а также одно- и двукратно продифференцированные сигналы U :



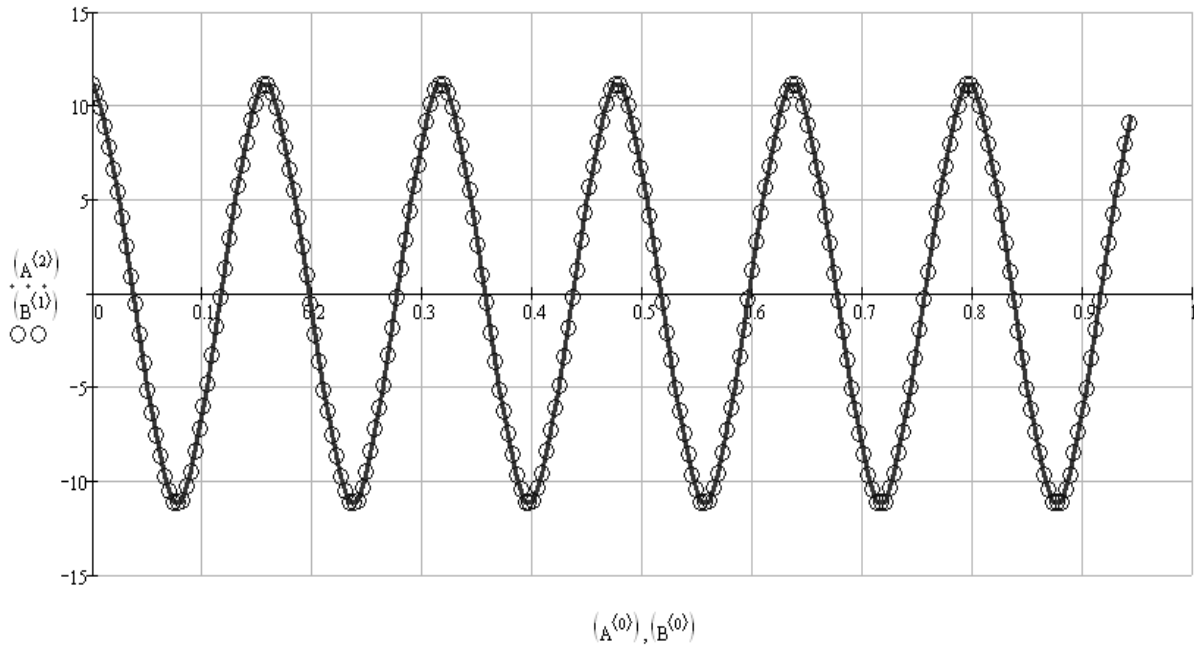
Для сглаженного дискретного сигнала U в столбце $A^{<2>}$ применим метод сокращения точек, используя функцию `rdots` с аргументами: S - двумерный массив с исходным сигналом, d - выборка (в новом массиве присутствует каждая d -точка исходного массива S).

```
rdots(S, d) :=
  Y ← 0
  N ← 0
  while N ≤ last(S<sup>(0)</sup>)
    XY,0 ← SN,0
    XY,1 ← SN,2
    N ← N + (d - 1)
    Y ← Y + 1
  X
```

Сформируем новый массив B :

$$B := \text{rdots}(A, 40)$$

Используя массив B , построим изображение полученного дискретного сигнала на временной диаграмме:



Варианты заданий

1. Составить в среде Mathcad цифровой (квантованный) сигнал, кусочно задав функцию U заданной формы в соответствии с вариантом задания (таблица 2), где номер варианта равен порядковому номеру студента в списке группы.
2. Сформировать дискретный сигнал.
3. К полученному дискретному сигналу применить метод сглаживания, использующий алгоритм нахождения скользящего среднего значения сигнала.
4. Дважды продифференцировать полученный сигнал.
5. Выполнить построение исходного цифрового сигнала, дискретного сигнала, сглаженного дискретного сигнала, сигнала после однократного дифференцирования, сигнала после двукратного дифференцирования.
6. Оформить отчет о работе.

Таблица 1.2 – Варианты заданий

Вариант	Форма сигнала
1	двухполярный сигнал с положительной пилообразной полуволной и отрицательной прямоугольной полуволной
2	двухполярный треугольный сигнал
3	двухполярный сигнал с положительной синусоидальной полуволной и отрицательной прямоугольной полуволной
4	двухполярный пилообразный сигнал
5	синусоидальный сигнал без отрицательной полуволны
6	синусоидальный сигнал без положительной полуволны
7	однополярный пилообразный сигнал
8	однополярный треугольный сигнал
9	двухполярный сигнал с положительной прямоугольной полуволной и отрицательной синусоидальной полуволной
10	двухполярный сигнал с положительной прямоугольной полуволной и отрицательной пилообразной полуволной
11	двухполярный сигнал с положительной синусоидальной полуволной и отрицательной трапециевидной полуволной
12	двухполярный сигнал с положительной прямоугольной полуволной и отрицательной трапециевидной полуволной
13	двухполярный трапециевидный сигнал
14	двухполярный сигнал с положительной трапециевидной полуволной и отрицательной прямоугольной полуволной
15	двухполярный сигнал с положительной трапециевидной полуволной и отрицательной синусоидальной полуволной
16	однополярный трапециевидный сигнал
17	двухполярный сигнал с положительной трапециевидной полуволной и отрицательной пилообразной полуволной
18	двухполярный сигнал с положительной пилообразной полуволной и отрицательной трапециевидной полуволной
19	двухполярный сигнал с положительной трапециевидной полуволной и отрицательной треугольной полуволной
20	двухполярный сигнал с положительной треугольной полуволной и отрицательной трапециевидной полуволной

Практическая работа 3. Интегрирование сигнала

Методика выполнения работы

Составим в среде Mathcad прямоугольный сигнал, кусочно задав функцию U и используя следующие параметры: $T1$ – длительность первого полупериода сигнала, сек., $U1$ – значение сигнала в первом полупериоде, $T2$ – длительность второго полупериода сигнала, сек., $U2$ – значение сигнала во втором полупериоде, T – длительность периода сигнала, сек.:

$$T1 := 0.080$$

$$U1 := 5$$

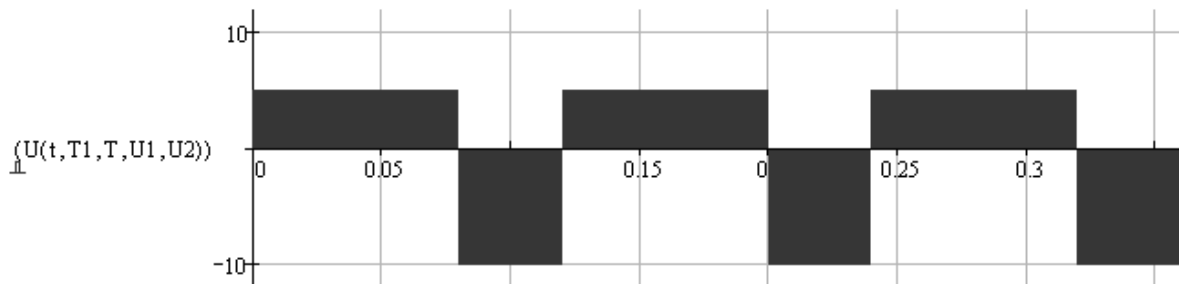
$$T2 := 0.040$$

$$U2 := -10$$

$$T := T1 + T2$$

$$U(t, t1, T, Umax, Umin) := \begin{cases} V \leftarrow Umax & \text{if } \text{mod}\left[t \cdot \left(\frac{T}{t1}\right), (t1) \cdot \left(\frac{T}{t1}\right)\right] \leq T \cdot \left(\frac{T}{t1}\right) \\ V \leftarrow Umin & \text{if } \text{mod}\left[t \cdot \left(\frac{T}{t1}\right), (T) \cdot \left(\frac{T}{t1}\right)\right] \geq t1 \cdot \left(\frac{T}{t1}\right) \\ \text{return } V \end{cases}$$

Далее построим изображение сигнала U на временной диаграмме:



(t)

Зададим функцию $FtoA$ для перевода функции сигнала U в двумерный массив, состоящий из двух столбцов, где первый столбец – столбец значений времени, второй – столбец значений дискретного сигнала. Аргументы функции $FtoA$: $totaltime$ – общее время моделирования сигнала, сек., $points$ – количество отсчетов (точек). $Step$ – шаг времени, сек.

```

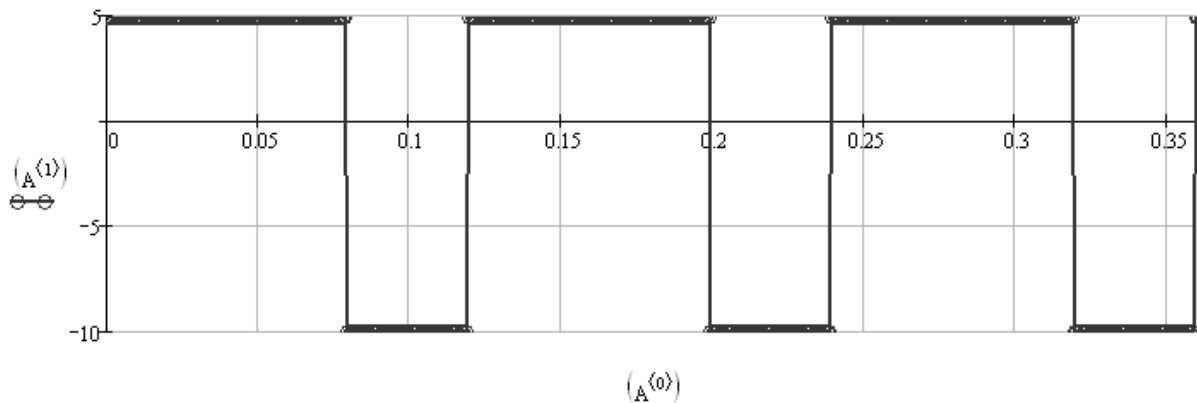
FtoA(totaltime, points) :=
  Z ← 0
  n ← 0
  step ← totaltime / points
  while n ≤ points
    Zn,0 ← step · n
    Zn,1 ← U[Zn,0, T1, T, U1, U2]
    n ← n + 1
  return Z

```

Сформируем массив A со значениями сигнала U :

```
A := FtoA(1, 1000)
```

Далее построим изображение дискретного сигнала на временной диаграмме:



Зададим функцию *Integ* для получения дискретного проинтегрированного по времени сигнала U , где: S – двумерный массив с исходным сигналом, N – номер столбца в массиве S с данными для интегрирования ($0 < N$):

```

Integ(S, N) :=
  Y ← 1
  INT ← 0
  L ← last(S<sup>(0)</sup>)
  SY, N+1 ← 0
  while Y ≤ L
    INT ← [(SY, N) · [(SY, 0) - (SY-1, 0))] + INT
    SY, N+1 ← INT
    Y ← Y + 1
  S

```

Проинтегрируем данные во втором столбце $A^{<1>}$ и получим в том же массиве A третий столбец $A^{<2>}$ с проинтегрированным дискретным сигналом U :

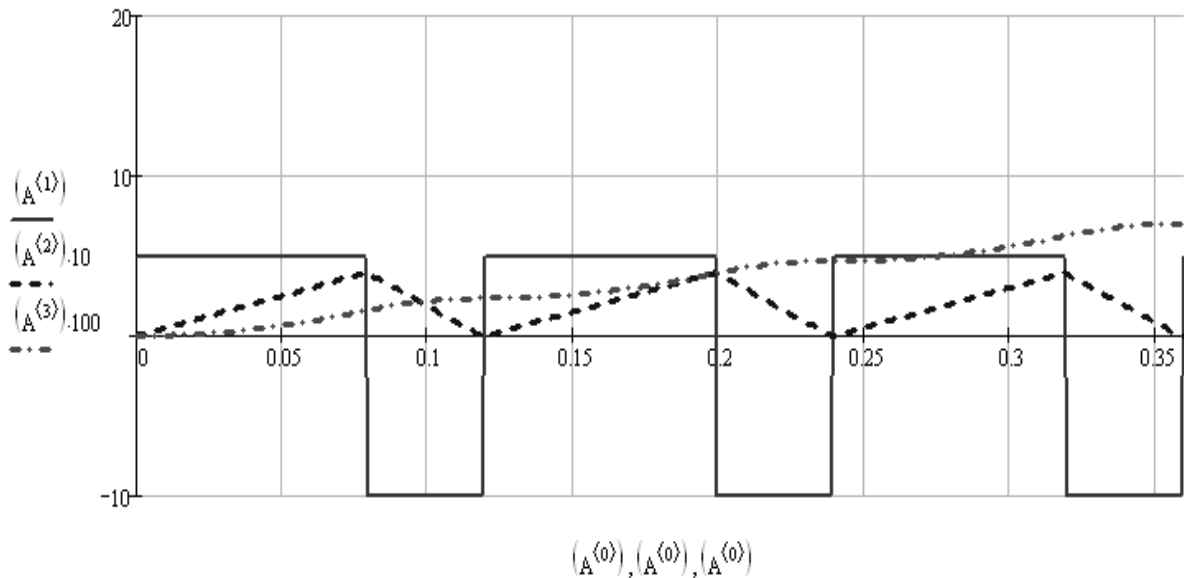
```
A := Integ(A, 1)
```

Проинтегрируем данные в третьем столбце $A^{<2>}$ и получим в том же массиве A четвертый столбец $A^{<3>}$ с дважды проинтегрированным дискретным сигналом U :

$$A := \text{Integ}(A, Z)$$

	0	1	2	3
985	0.985	5	0.125	0.189
986	0.986	5	0.13	0.189
987	0.987	5	0.135	0.189
988	0.988	5	0.14	0.189
989	0.989	5	0.145	0.189
990	0.99	5	0.15	0.19
991	0.991	5	0.155	0.19
992	0.992	5	0.16	0.19
993	0.993	5	0.165	0.19
994	0.994	5	0.17	0.19
995	0.995	5	0.175	0.19
996	0.996	5	0.18	0.191
997	0.997	5	0.185	0.191
998	0.998	5	0.19	0.191
999	0.999	5	0.195	0.191
1000	1	5	0.2	0.191

Построим на одной временной диаграмме изображение дискретного сигнала U , а также одно- и двукратно проинтегрированные сигналы U :



Варианты заданий

1. Составить в среде Mathcad прямоугольный сигнал, кусочно задав функцию U , используя следующие параметры: $T1$ – длительность первого полупериода сигнала, сек., $U1$ – значение сигнала в первом полупериоде, $T2$ – длительность второго полупериода сигнала, сек., $U2$ – значение сигнала во втором полупериоде, при этом $T1 = N \cdot 0,01$, $U1 = N \cdot 2$, $T2 = (T1 \cdot U1) / 5$, $U2 = -(T1 \cdot U1) / T2$, где N – номер варианта задания (соответствует порядковому номеру студента в списке группы).
2. Сформировать дискретный сигнал.
3. Дважды проинтегрировать полученный сигнал.

4. Выполнить построение исходного цифрового сигнала, дискретного сигнала, сигнала после однократного интегрирования, сигнала после двукратного интегрирования.
5. Дважды продифференцировать сигнал, полученный после двукратного интегрирования.
6. Выполнить построение исходного цифрового сигнала, дискретного сигнала, полученного после двукратного дифференцирования. Получить на временной диаграмме разность этих сигналов и объяснить полученный результат.
7. Оформить отчет о работе.

Практическая работа 4. Работа с изображениями

Методика выполнения работы

Сформируем двумерный массив M , содержащий данные графического файла SAM.BMP в цветовой модели RGB:

$$M := \text{READRGB}("C:\SAM.BMP")$$

	0	1	2	3	4	5	6	7	8	9
0	64	64	64	64	96	96	96	96	96	96
1	64	96	64	64	96	96	96	96	96	96
2	64	64	96	96	96	96	96	96	96	96
3	64	64	96	96	96	96	64	64	96	96
4	96	96	96	96	96	96	96	96	96	96
5	96	96	96	64	96	96	96	96	96	64
6	96	96	96	64	96	96	96	96	64	96
7	96	96	96	96	96	96	96	64	64	64
8	96	96	96	96	96	96	64	64	64	32
9	64	64	96	96	64	64	64	64	32	32
10	64	64	64	96	64	64	64	64	32	32
11	64	64	64	64	64	32	32	32	32	32
12	64	64	64	64	32	32	32	0	32	32
13	32	32	32	32	32	32	32	32	32	32
14	32	32	32	32	32	32	32	32	32	64
15	32	32	0	0	32	32	0	64	64	64

Выделим из массива M массивы RM , GM и BM , соответствующие красному, зеленому и синему оттенкам графического изображения.

$$\text{rows}(M) = 480$$

$$\text{cols}(M) = 1.92 \times 10^3$$

$$w := \frac{\text{cols}(M)}{3}$$

$$h := \text{rows}(M)$$

Для изменения порядка индексирования массивов присвоим встроенной переменной $ORIGIN$ значение «1»:

$$ORIGIN := 1$$

$$RM := \text{submatrix}(M, 1, h, 1, w)$$

$$GM := \text{submatrix}(M, 1, h, w + 1, 2 \cdot w)$$

$$BM := \text{submatrix}(M, 1, h, 2 \cdot w + 1, 3 \cdot w)$$

Уменьшим изображение в 2 раза, сократив количество точек:

$$CUT := 2$$

$$\text{newr} := 1.. \left(\frac{\text{rows}(RM)}{CUT} \right)$$

$$\text{newc} := 1.. \left(\frac{\text{cols}(\text{RM})}{\text{CUT}} \right)$$

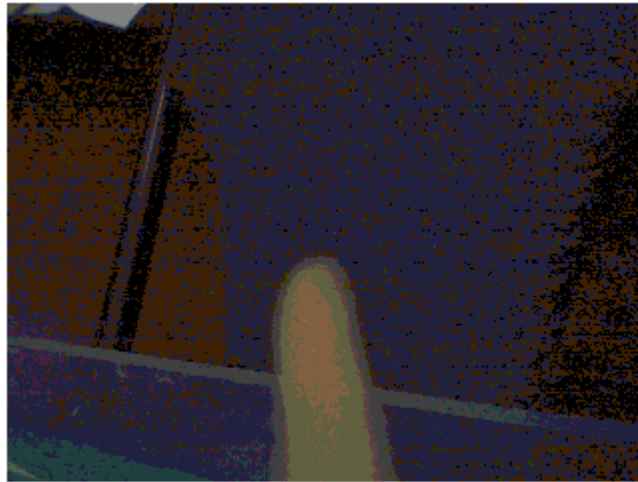
Создадим массивы оттенков NRM, NGM и NBM с уменьшенным изображением:

$$\text{NRM}_{\text{newr}, \text{newc}} := \text{RM}_{\text{CUT} \cdot \text{newr}, \text{CUT} \cdot \text{newc}}$$

$$\text{NGM}_{\text{newr}, \text{newc}} := \text{GM}_{\text{CUT} \cdot \text{newr}, \text{CUT} \cdot \text{newc}}$$

$$\text{NBM}_{\text{newr}, \text{newc}} := \text{BM}_{\text{CUT} \cdot \text{newr}, \text{CUT} \cdot \text{newc}}$$

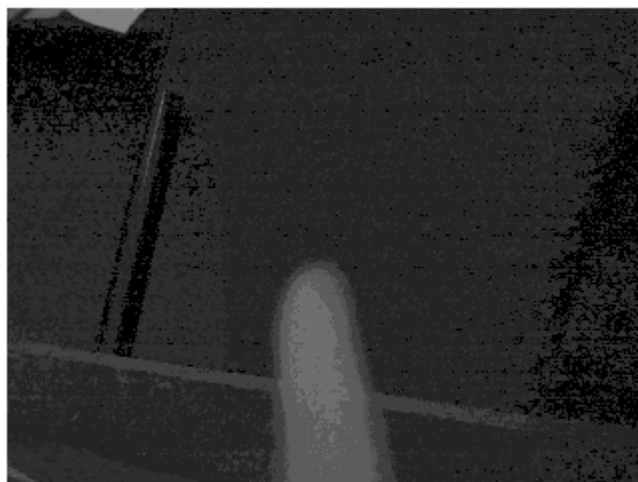
Создадим наложение оттенков NRM, NGM и NBM в компоненте Picture:



NRM, NGM, NBM

Получим графическое изображение в оттенках серого цвета:

$$\text{GRAY} := \text{NRM} \cdot 0.56 + \text{NGM} \cdot 0.33 + \text{NBM} \cdot 0.11$$



GRAY

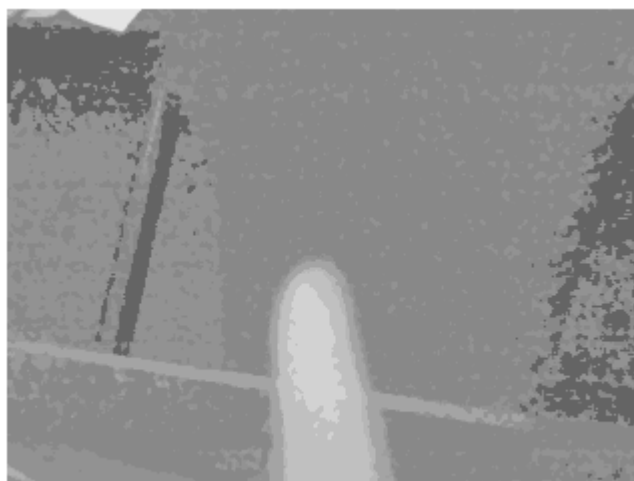
Применим функцию medflt (медианный фильтр изображения) для уменьшения уровня шума:

$$\text{medflt}(M, k) := \left| \begin{array}{l} \text{for } i \in \frac{k-1}{2} \dots \text{rows}(M) - \frac{k+1}{2} \\ \quad \text{for } j \in \frac{k-1}{2} \dots \text{cols}(M) - \frac{k+1}{2} \\ \qquad \left| \begin{array}{l} \text{for } r \in 1 \dots k \\ \qquad \text{for } s \in 1 \dots k \\ \qquad \qquad A_{r,s} \leftarrow M_{i+r-\frac{k-1}{2}, j+s-\frac{k-1}{2}} \\ \qquad \qquad N_{i,j} \leftarrow \text{median}(A) \end{array} \right. \\ \qquad \qquad \qquad N \end{array} \right.$$

GRAY := medflt(GRAY, 3)

Осветлим полученное изображение:

GRAY := GRAY + 100



GRAY

Применим фильтр для выделения некоторого диапазона цветов изображения (цвета изображения меняются от абсолютно черного до абсолютно белого, т.е. от 0 до 255):

$$\text{Filter}(x, x_{\min}, x_{\max}) := \begin{cases} x & \text{if } x_{\min} < x \leq x_{\max} \\ 255 & \text{otherwise} \end{cases}$$

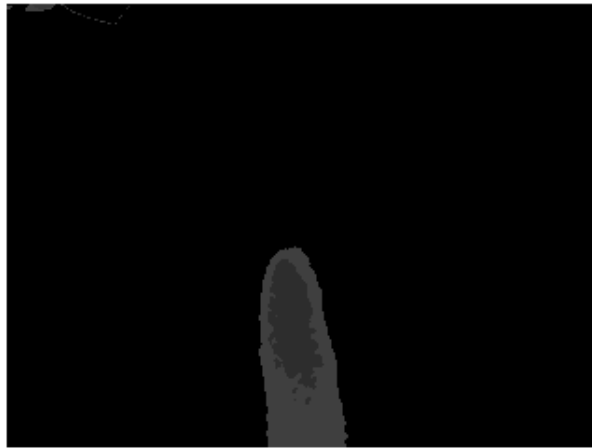
→
GRAY2 := Filter(GRAY, 190, 220)



GRAY2

Получим инверсное изображение:

```
INVERSION := 255 - GRAY2
```



INVERSION

Применим функцию `medflt` к инверсному изображению:

```
INVERSION2 := medflt(INVERSION, 9)
```



INVERSION2

Варианты заданий

1. Сформировать в среде Mathcad двумерный массив M , содержащий данные некоторого графического файла в цветовой модели RGB.
2. Получить графическое изображение в оттенках серого цвета.
3. Применить функцию `medflt`.
4. Применить фильтр для выделения диапазона цветов изображения: от $N \cdot 10$ до $N \cdot 25$, где N – номер варианта задания (соответствует порядковому номеру студента в списке группы). Остальная область изображения закрашивается абсолютно белым цветом.
5. Оформить отчет о работе.

2 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНЫХ РАБОТ

Применение фильтра Гаусса для восстановления зашумленного сигнала

Цель работы: изучение методов применения фильтра Гаусса для восстановления зашумленного сигнала.

Аппаратные средства: программный комплекс MATLAB.

Краткие теоретические сведения

Современные системы автоматического управления во многих случаях полагаются на информацию, поступающую датчиков, для выработки управляющего сигнала. Точность информации, поступающей с датчика, оказывает существенное влияние на качество работы системы автоматического управления. При этом на практике сигналы с датчиков зачастую оказываются «зашумлены». Под «зашумленным» сигналом подразумевают сигнал, представляющий собой смесь содержащего в себе полезную информацию сигнала и шума, сигнала не несущего полезной информационной нагрузки. Использование зашумленного сигнала в цепи обратной связи системы управления может вызвать ухудшение работы этой системы.

Для решения проблемы зашумленности сигнала рассмотрим метод фильтрации сигнала, использующий преобразование Фурье и фильтр Гаусса. Суть метода заключается в том, что над зашумленным сигналом производится преобразование Фурье для перехода к частотному домену, где на сигнал накладывается фильтр Гаусса. Подразумевается, что мы знаем, какая область частот нас интересует. Над полученным сигналом производится обратное преобразование Фурье. В результате описанных действий получаем сигнал во временном домене, очищенный от шума.

Напомним, что преобразование Фурье описывается следующим образом:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) \cdot e^{-ix\omega} dx \quad (2.1)$$

Применение фильтра Гаусса равносильно умножению исходного сигнала на функцию Гаусса, описываемую следующим выражением:

$$\varpi(t) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{t-t_0}{2\sigma^2}} \quad (2.2)$$

Данный метод подходит для борьбы с белым шумом. При наличии других видов шумов следует проанализировать применимость и эффективность метода перед его применением.

Методика выполнения работы

Работа выполняется в среде MATLAB. До начала работы с кодом рекомендуется выбрать рабочую папку и создать новый m-файл в этой папке.

Сначала зададим функцию, имитирующую незашумленный сигнал с датчика:

```
N = 10000;
dt = 0.001;
for i=1:N,
A(i) = 2 * sin(i*dt);
end
plot(A);
```

Исполнение данного участка кода выведет на экран заданную функцию – в данном случае синусоиду – рисунок 2.1.

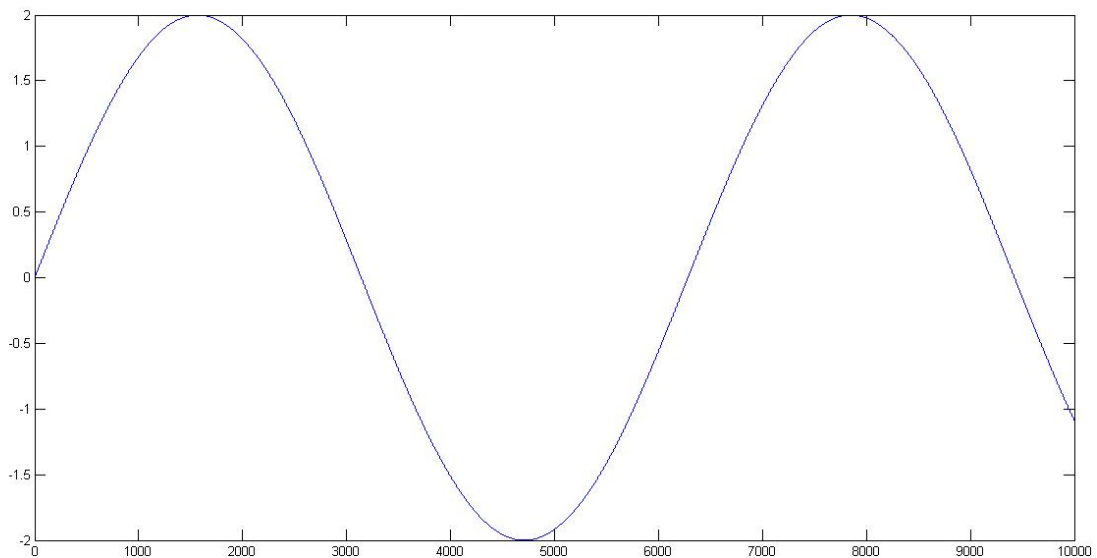


Рисунок 2.1 Исходный сигнал

Добавим к сигналу имитацию белого шума. Для этого прибавим к каждому элементу вектора A случайную величину в диапазоне от 0 до 2:

```
N = 10000;
dt = 0.001;

for i=1:N,
A(i) = 2 * sin(i*dt);
end

for i=1:N,
A(i) = A(i) + 2*randn;
end
```

```
plot(A);
```

Исполнение данного участка кода выведет на экран отображение зашумленного сигнала – рисунок 2.2.

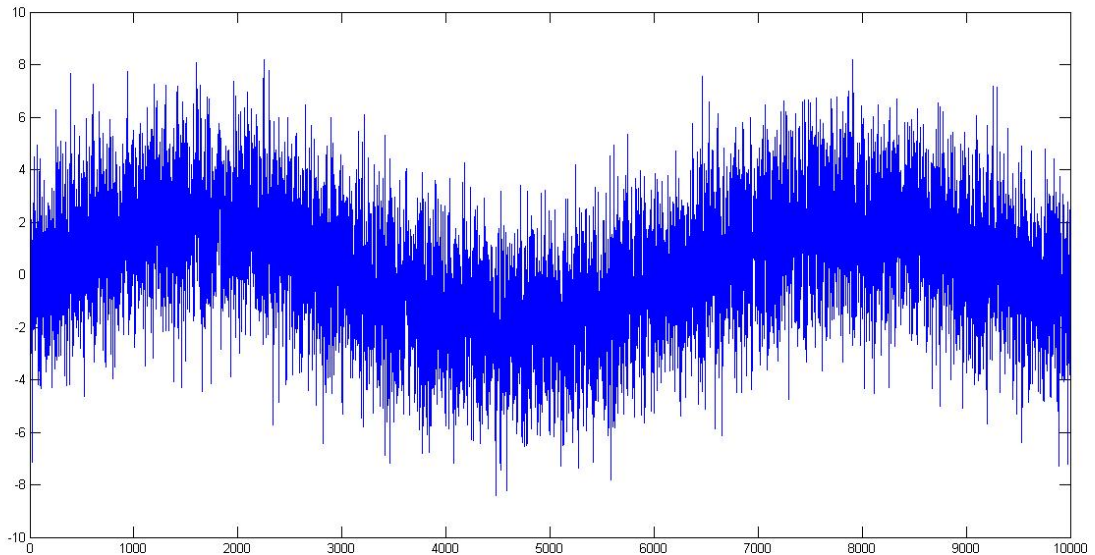


Рисунок 2.2 Зашумленный сигнал

Используем быстрое преобразование Фурье для перехода в частотный домен. Воспользуемся встроенными функциями пакета MATLAB – `fft()` и `fftshift()`.

```
N = 10000;
dt = 0.001;

for i=1:N,
A(i) = 2 * sin(i*dt);
end

for i=1:N,
A(i) = A(i) + 2*randn;
end

F = fft(A);
F = fftshift(F);
plot(abs(F));
```

Выполнение данного участка кода выведет на экран спектрограмму зашумленного сигнала – рисунок 2.3.

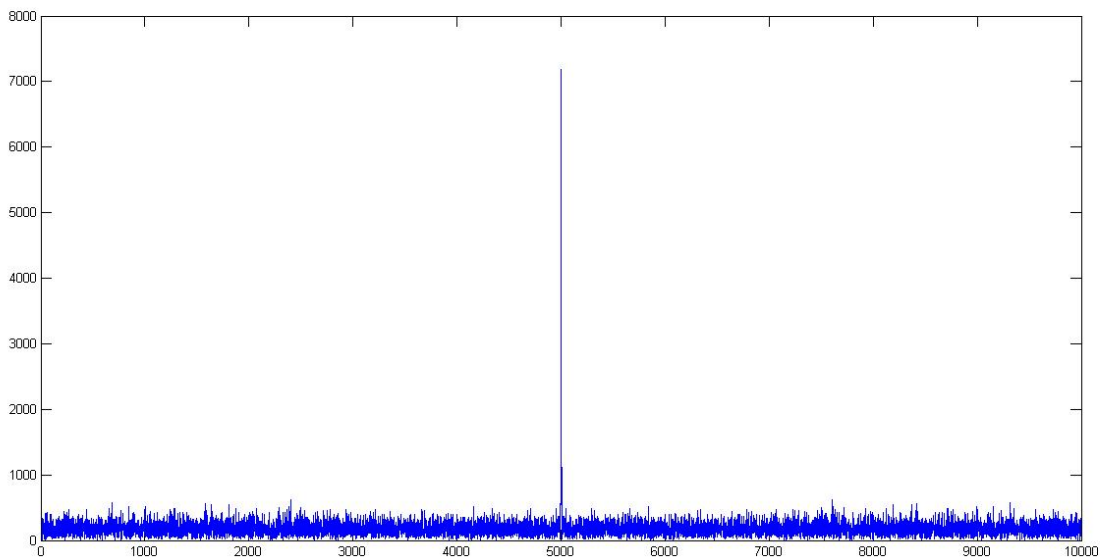


Рисунок 2.3 Спектрограмма зашумленного сигнала

Обратим внимание, что на спектрограмме присутствует один пик, соответствующий частоте полезного сигнала. Остальные частоты представлены равномерно, что характерно для белого шума.

Представленный ниже код генерирует фильтр Гаусса (функцию Гаусса представленную в виде вектора той же размерности что и вектор, содержащий спектр зашумленного сигнала).

```
N = 10000;
dt = 0.001;

for i=1:N,
A(i) = 2 * sin(i*dt);
end

for i=1:N,
A(i) = A(i) + 2*randn;
end

F = fft(A);
F = fftshift(F);
GWP = 0.0001;      %Это параметр определяет ширину гауссовой
                  %функции
Fsize = size(F,2);
for i=1:Fsize,
W(i) = exp(-GWP*(i - (Fsize/2)).^2);
end
plot(W);
```


Выполнение данного участка кода выведет на экран Гауссову функцию – рисунок 2.4.

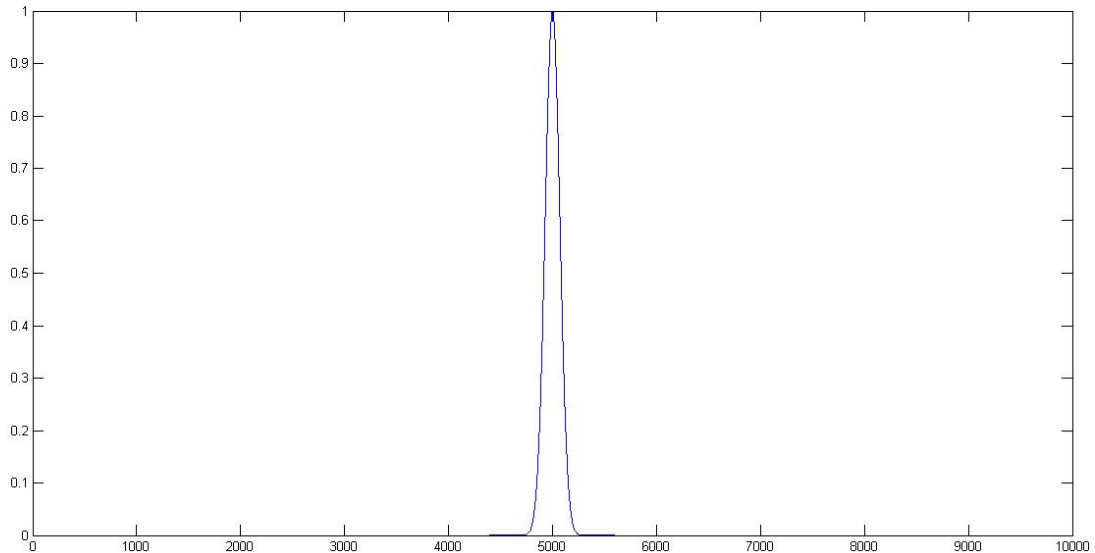


Рисунок 2.4 Функция Гаусса

Умножим спектр зашумленного сигнала на полученную Гауссову функцию и применим к результату обратное преобразование Фурье:

```

N = 8*1024;
dt = 0.005;

B = 1:N;
A = 1:N;

for j=1:N,
B(j) = 2 * sin(j*dt)+5;
end

for j=1:N,
A(j) = B(j) + 2*randn;
end

F = fft(A);
F = fftshift(F);
GWP = 0.01;
Fsize = size(F,2);

for j=1:Fsize,
W(j) = exp(-GWP*(j - (Fsize/2)).^2) + 1i*exp(-GWP*(j -
(Fsize/2)).^2);
end

```

```

F_filt = F.*W;
Res = ifftshift(F_filt);
Res = ifft(Res);
Scale = abs( exp(-GWP*(0).^2) + 1i*exp(-GWP*(0).^2) );
Summ = 0;

for j=1:size(Res,2),
Summ = Summ + abs(Res(j));
end

Summ = Summ / size(Res,2);

for j=1:size(Res,2),
Res(j) = abs(Res(j)) + Summ*(1/Scale - 1);
end

subplot (2,3,1), plot(B);
subplot (2,3,2), plot(A);
subplot (2,3,3), plot(Res);
subplot (2,3,4), plot(abs(W));
subplot (2,3,5), plot(abs(F));

```

Выполнение данной программы выведет на экран пять графиков – исходный сигнал, зашумленный сигнал, очищенный сигнал, Гауссову функцию, спектрограмму зашумленного сигнала.

Как не сложно заметить, очищенный от шума сигнал не полностью совпадает с исходным сигналом. Это характерно для сигналов, полученных очисткой от белого шума с помощью наложения Гауссовой функции в частотном домене.

Задание на выполнение работы

Задание на выполнение работы – фильтрация гармонического сигнала от белого шума. Вид сигнала (функция $F(t)$) и амплитуду накладываемого белого шума η следует выбирать из таблицы 2.1.

Таблица 2.1– Варианты заданий

№	Задание
1	$F(t) = 0.35 \sin(10 \cdot t) + 0.15 \sin(8 \cdot t) + 0.35 \sin(6 \cdot t), \eta = 3$
2	$F(t) = 0.15 \sin(2 \cdot t) + 0.45 \sin(8 \cdot t) + 0.85 \sin(12 \cdot t) + 1.05 \sin(0.5 \cdot t), \eta = 5$
3	$F(t) = 0.25 \sin(13 \cdot t) + 0.35 \sin(12 \cdot t) + 0.75 \sin(5 \cdot t) + 1.05 \sin(4 \cdot t), \eta = 3$
4	$F(t) = 0.45 \sin(2 \cdot t) + 0.65 \sin(0.5 \cdot t) + 0.05 \sin(18 \cdot t) + 0.5 \sin(9 \cdot t), \eta = 1.5$
5	$F(t) = 0.85 \sin(7 \cdot t) + 0.9 \sin(4 \cdot t) + 0.1 \sin(10 \cdot t) + 0.2 \sin(12 \cdot t), \eta = 4.5$
6	$F(t) = 0.6 \cos(14 \cdot t) + 0.95 \sin(2 \cdot t) + 0.5 \sin(t) + 0.7 \sin(3 \cdot t) + 0.8 \sin(12 \cdot t),$ $\eta = 5$
7	$F(t) = 0.1 \cos(4 \cdot t) + 0.7 \sin(6 \cdot t) + 0.6 \sin(t) + 0.85 \sin(12 \cdot t) + 0.9 \sin(8 \cdot t),$ $\eta = 3$
8	$F(t) = 0.5 \cos(12 \cdot t) + 0.55 \sin(17 \cdot t) + 0.65 \sin(3 \cdot t) + 0.3 \sin(15 \cdot t) + 0.45 \sin(t),$ $\eta = 4.5$
9	$F(t) = 0.5 \sin(2 \cdot t) + 0.3 \sin(12 \cdot t) + 0.1 \sin(0.5 \cdot t), \eta = 1.5$
10	$F(t) = 0.2 \sin(7 \cdot t) + 0.1 \cos(6 \cdot t) + 0.3 \cos(3 \cdot t), \eta = 3$

ОКОННОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ И ПРЕОБРАЗОВАНИЕ ГАБОРА

Цель работы: изучить методику использования оконного преобразования Фурье и преобразования Габора для обработки сигналов

Аппаратные средства: математический пакет MathCAD.

Краткие теоретические сведения

Преобразование Фурье позволяет перевести сигнал из временного домена в частотный домен, что может быть использовано для проведения анализа частотного спектра сигнала.

В динамических системах зачастую важным оказывается время появления той или иной гармоники, входящей в состав сигнала. Примером такой задачи является детектирование момента возникновения и/или прекращения вибраций корпуса устройства. При этом преобразование Фурье не позволяет оценить изменение спектра сигнала во времени. Для оценки изменения спектра сигнала во времени могут быть использованы методы оконного преобразования Фурье. В англоязычной литературе для обозначения оконного преобразования Фурье используется термин «short-time Fourier transform»; термин «windowed Fourier transform» используется значительно реже.

В общем виде выражение, описывающее оконное преобразование Фурье для функции $f(x)$, выглядит следующим образом:

$$\hat{f}(\omega, \tau) = \int_{-\infty}^{\infty} f(t) \cdot \varpi(t - \tau) \cdot e^{-it\omega} dt \quad (2.3)$$

где $\omega(t - \tau)$ – так называемая оконная функция, $f(t)$ – исходный сигнал, а $\hat{f}(\omega, \tau)$ – частотный спектр участка сигнала, попавшего в «окно».

Величина τ определяет, для какого момента времени мы строим спектрограмму.

Обратим внимание на то, что существенным отличием оконного преобразования Фурье от обычного преобразования Фурье является наличие функции $\omega(t - \tau)$. Задача этой функции состоит в том, чтобы отсечь сигнал, находящийся вне пределов некоторого окна. Существует множество различных функций, используемых в роли оконной функции. Зачастую используются функции, имеющие область значений в диапазоне от 0 до 1, что позволяет избежать усиления или изменения знака сигнала. В рамках данной работы рассмотрим два варианта реализации оконной функции – прямоугольное окно и гауссову функцию.

Оконное преобразование Фурье с использованием гауссовой функции в качестве окна называется преобразованием Габора (названо в честь венгерского физика Денеша Габора, лауреата нобелевской премии по физике 1971 года).

Функция прямоугольного окна выглядит следующим образом:

$$\varpi(t) = \begin{cases} 1 & \text{если } t \in [t - a; t + a] \\ 0 & \text{если } t \notin [t - a; t + a] \end{cases} \quad (2.4)$$

После наложения данной функции весь сигнал, находящийся вне пределов окна, обратится в ноль. Данная функция относительно проста в реализации, что

является одним из её основных преимуществ. При этом использование данной оконной функции может внести нежелательные искажения в спектрограмму, связанные с резкими границами окна. При использовании данной оконной функции выражение (2.3) может быть переписано следующим образом:

$$\hat{f}(\omega, \tau) = \int_{\tau-a}^{\tau+a} f(t) \cdot e^{-it\omega} dt \quad (2.5)$$

Гауссова функция может быть представлена в следующем виде:

$$\varpi(t) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{t-t_0}{2\sigma^2}} \quad (2.6)$$

На практике при использовании гауссовой функции предполагается, что существует некоторый диапазон $[t - b; t + b]$ за пределами которого значение этой функции пренебрежимо мало. Таким образом, становится возможным записать выражение (2.3) следующим образом:

$$\hat{f}(\omega, \tau) = \int_{\tau-b}^{\tau+b} f(t) \cdot \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{t-\tau}{2\sigma^2}} \cdot e^{-it\omega} dt \quad (2.7)$$

Можно заметить, полученная функция является некоторой комбинацией гауссовой и прямоугольной оконных функций.

Методика выполнения работы

Зададимся некоторыми полигармоническими функциями $F(t)$ и $F_1(t)$, а также зададим частоту дискретизации Δ .

$$k := 1$$

$$F_1(t) := 0.03 \sin(15 \cdot k \cdot t)$$

$$F(t) := 0.02 \sin(2 \cdot k \cdot t) + 0.01 \sin(10 \cdot k \cdot t) + 0.015 \sin(25 \cdot k \cdot t)$$

$$\Delta := \frac{0.0001}{k}$$

Ниже приведен код, накладывающий гармонический сигнал $F_1(t)$ на сигнал $F(t)$ на промежутке $(T_1; T_2)$.

```

SinFunc(Time, T1, T2, Δ) :=
  Count ← Time / Δ
  for i ∈ 0..Count
    t ← i · Δ
    Ai,0 ← t
    Ai,1 ← F(t)
    Ai,1 ← Ai,1 + F1(t) if (t > T1) ∧ (t < T2)
  A

```

Ниже показан пример вызова написанной функции:

```
Signal := SinFunc(40, 20, 30, Δ)
```

Полученный в результате вызова функции сигнал показан на рисунке 2.5.

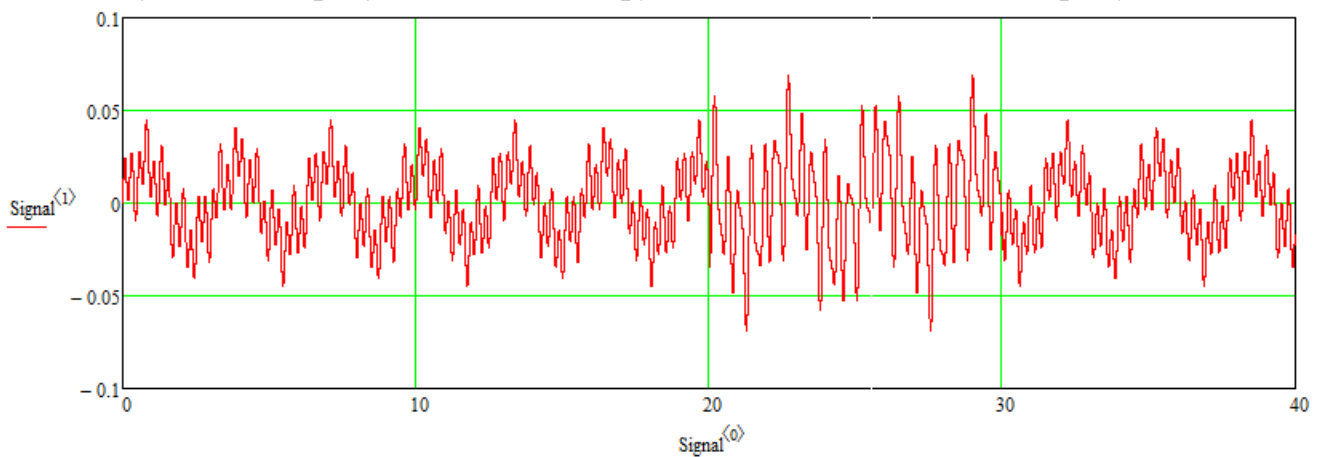


Рисунок 2.5 Гармонический сигнал $F(t)$ с наложенным на него сигналом $F_1(t)$ на промежутке $(T_1 ; T_2)$

Рассмотрим другой способ сложения гармонических сигналов на заданном временном диапазоне. Для этого мы можем записать следующее выражение:

$$F2(t, t_0, \alpha) := F(t) + 1.5 \cdot F1(t) \cdot e^{-\pi \alpha (t-t_0)^2}$$

$$t := 0, 0.001.. 40$$

Этот метод использует гауссову функцию, в результате чего результирующий сигнал не имеет резких границ в области, где появляется (и исчезает) добавленная гармоника. Результат сложения сигналов данным методом показан на рисунке 2.6.

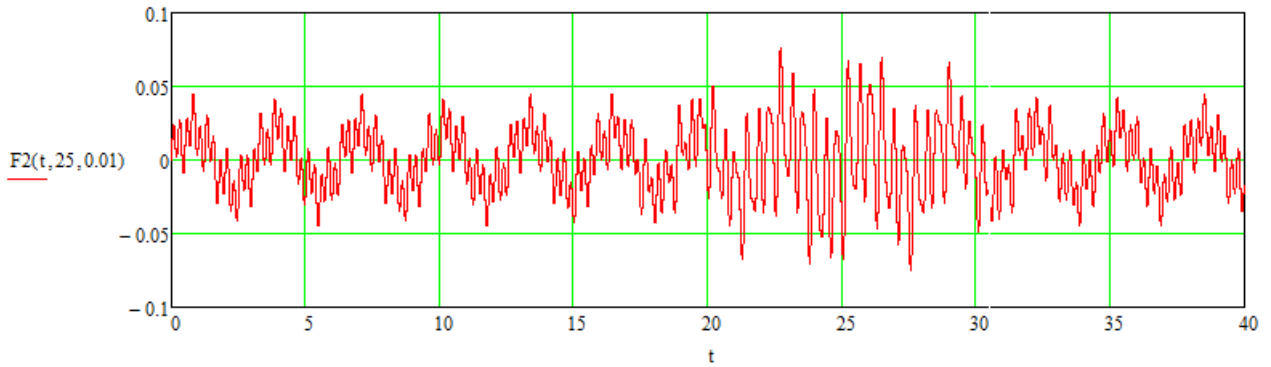


Рисунок 2.6 Гармонический сигнал $F(t)$ с наложенным на него сигналом $F_1(t)$, умноженным на гауссову функцию

Запишем аналитическую функцию, реализующую преобразование Фурье для заданного диапазона частот.

```

Fourier(A, ωMAX, ωMIN, Δω, Δ) :=
  Count ←  $\frac{\omega_{MAX} - \omega_{MIN}}{\Delta\omega}$ 
  CountTime ← длина(A<1>) - 1
  for i ∈ 0..Count
    W ← 0
    ω ← i · Δω + ωMIN
    for j ∈ 0..CountTime
      t1 ← j · Δ
      W ← W + Δ · Aj,1 · e-2·π·t1·i(ω)
    Bi,0 ← ω
    Bi,1 ←  $\sqrt{\text{Im}(W)^2 + \text{Re}(W)^2}$ 
  B
  
```

Ниже показан пример вызова данной функции:

$$\text{Res} := \text{Fourier}(\text{Signal}, 6 \cdot k, 0, 0.02 \cdot k, \Delta)$$

Результат работы функции представлен на рисунке 2.7.

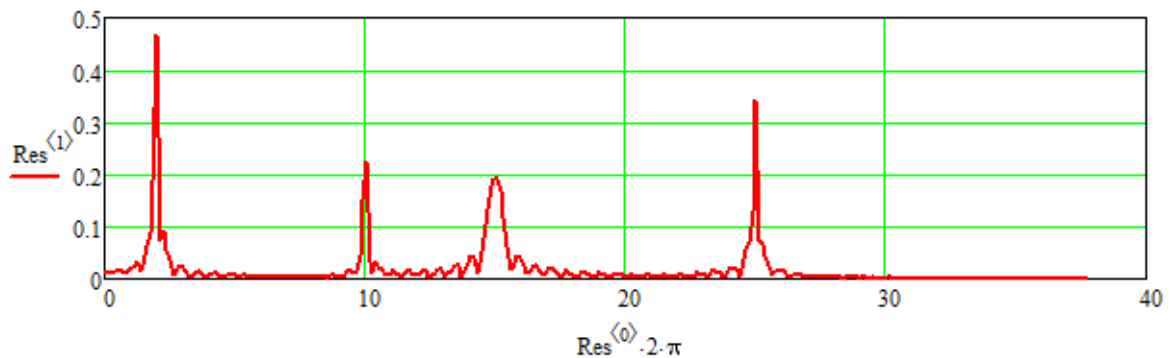


Рисунок 2.7 Частотный спектр сигнала

Следует обратить внимание, что хотя мы и получили информацию о частотном спектре сигнала, в результате преобразования мы потеряли информацию

о характере изменения сигнала. Так, анализируя полученный график, мы можем сделать вывод о наличии в спектре гармонической составляющей с круговой частотой 15 радиан в секунду, при этом мы не можем оценить момент появления этой составляющей в спектре.

Рассмотрим пример реализации оконного преобразования Фурье с использованием прямоугольной оконной функции.

$$\text{FourierWindow}(A, \omega_{\text{MAX}}, \omega_{\text{MIN}}, \Delta\omega, T1, T2, \Delta) := \left. \begin{array}{l} \text{Count} \leftarrow \frac{\omega_{\text{MAX}} - \omega_{\text{MIN}}}{\Delta\omega} \\ \text{for } i \in 0.. \text{Count} \\ \quad \left. \begin{array}{l} W \leftarrow 0 \\ \omega \leftarrow i \cdot \Delta\omega + \omega_{\text{MIN}} \\ \text{for } j \in \frac{T1}{\Delta} .. \frac{T2}{\Delta} \\ \quad \left. \begin{array}{l} t1 \leftarrow j \cdot \Delta \\ W \leftarrow W + \Delta \cdot A_{j,1} \cdot e^{-2 \cdot \pi \cdot t1 \cdot i(\omega)} \end{array} \right. \\ B_{i,0} \leftarrow \omega \\ B_{i,1} \leftarrow \sqrt{\text{Im}(W)^2 + \text{Re}(W)^2} \end{array} \right. \\ B \end{array} \right\}$$

Ниже показан пример вызова написанной функции.

Time := 0

Res := FourierWindow(Signal, 6 · k, 0, 0.05 · k, 0 + Time, 5 + Time, Δ)

Результат оконного преобразования для промежутка от 0 до 5 секунд показан на рисунке 2.8:

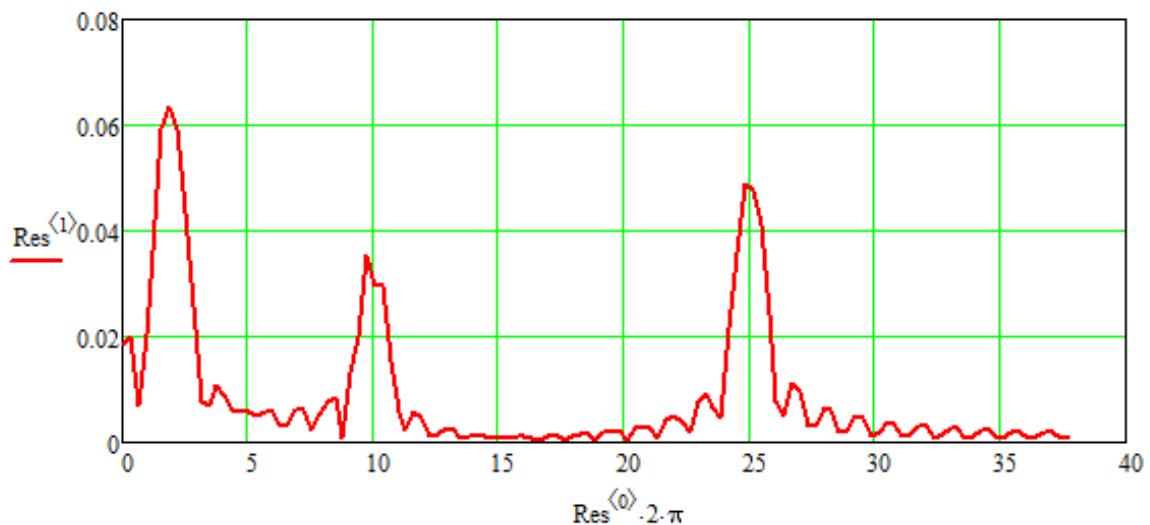


Рисунок 2.8 Спектрограмма, полученная с помощью оконного преобразования Фурье с прямоугольной оконной функцией на промежутке от 0 до 5 с

Запишем выражение для выполнения преобразования Габора:

$$G(\omega, t, \alpha) := \int_{t-2}^{t+2} e^{-\pi \alpha (\tau-t)^2} \cdot e^{-2 \cdot \pi \cdot \tau \cdot i \cdot (\omega)} \cdot F2(\tau, 25, 0.01) d\tau$$

$$gFT(\omega) := \sqrt{\text{Im}(G(\omega, 2, 0.01))^2 + \text{Re}(G(\omega, 2, 0.01))^2}$$

$$\omega := 0, 0.02 \dots 6$$

Представленный выше код использует встроенные средства пакета Mathcad для реализации численного интегрирования. Результат применения данной функции показан на рисунке 2.9.

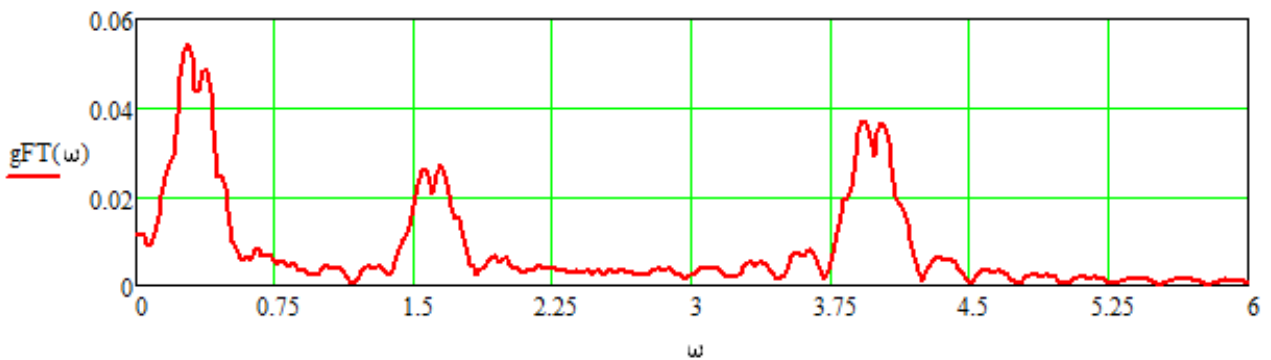
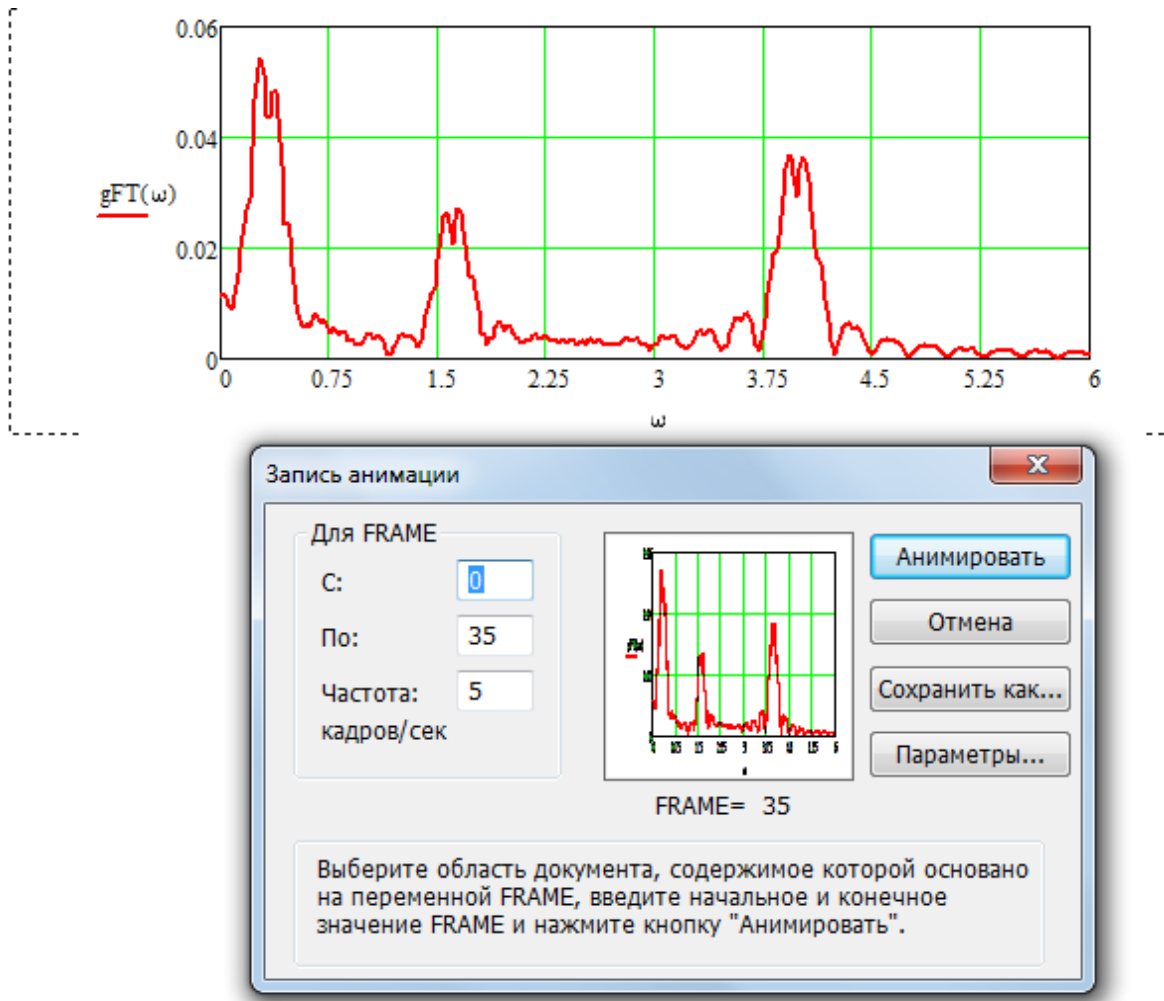


Рисунок 2.9 Спектрограмма, полученная с помощью преобразования Габора для момента времени 2 с

Одним из эффективных способов визуализации результата применения преобразования Габора является анимация изменения спектра по мере перемещения оконной функции. Ниже представлен код и настройки встроенной функции «Запись анимации» среды Mathcad.

$$\text{FRAME1} := 0$$

$$gFT(\omega) := \sqrt{\text{Im}(G(\omega, \text{FRAME1} + 2, 0.01))^2 + \text{Re}(G(\omega, \text{FRAME1} + 2, 0.01))^2}$$



Рассмотрим реализация преобразования Габора программными методами. Для реализации метода численного интегрирования будем использовать метод прямоугольников.

```

Gabor(A, ωMAX, ωMIN, Δω, T, α, Δ) :=
  Count ←  $\frac{\omega_{MAX} - \omega_{MIN}}{\Delta\omega}$ 
  for i ∈ 0..Count
    W ← 0
    ω ← i · Δω + ωMIN
    for j ∈  $\frac{T-2}{\Delta} .. \frac{T+2}{\Delta}$ 
      t ← j · Δ
      W ← W + Δ · Aj,1 · e-2·π·t·i·ω · e-π·α·(t-T)2
    Bi,0 ← ω
    Bi,1 ←  $\sqrt{\text{Im}(W)^2 + \text{Re}(W)^2}$ 
  B
  
```

Ниже показан пример вызова написанной функции и полученные результаты:

```
Res := Gabor(Signal, 6, 0, 0.05, 23, 0.01, Δ)
```

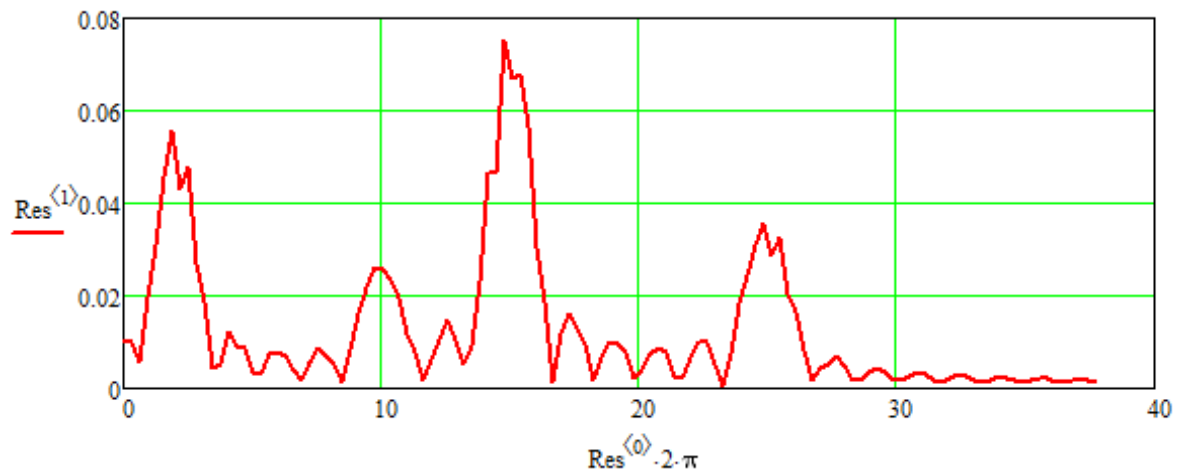


Рисунок 2.10 Спектрограмма, полученная в результате преобразования Габора для момента времени 23 с

С помощью описанных методов может быть построена диаграмма для любого участка сигнала, при этом от ширины окна зависит как точность определения составляющих спектра, так и точность определения времени возникновения и исчезновения отдельных гармоник.

Задание на выполнение работы

Задание на выполнение работы состоит в нахождении времени появления и времени исчезновения гармоники сигнала используя один из описанных в работе методов оконного преобразования Фурье.

Метод оконного преобразования Фурье и исходные данные следует выбирать из таблицы 2.2.

Таблица 2.2 – Варианты заданий

№	Задание
1	Оконное преобразование Фурье с прямоугольным окном, $T_1 = 3$ с, $T_2 = 15$ с, $k = 2$
2	Преобразование Габора, $T_1 = 23$ с, $T_2 = 26$ с, $k = 3$
3	Преобразование Габора, $T_1 = 4$ с, $T_2 = 25$ с, $k = 0.5$
4	Оконное преобразование Фурье с прямоугольным окном, $T_1 = 1$ с, $T_2 = 20$ с, $k = 0.2$
5	Преобразование Габора, $T_1 = 12$ с, $T_2 = 13$ с, $k = 5$
6	Преобразование Габора, $T_1 = 31$ с, $T_2 = 33$ с, $k = 12$
7	Оконное преобразование Фурье с прямоугольным окном, $T_1 = 5$ с, $T_2 = 20$ с, $k = 0.1$
8	Преобразование Габора, $T_1 = 5$ с, $T_2 = 8$ с, $k = 20$
9	Преобразование Габора, $T_1 = 7$ с, $T_2 = 13$ с, $k = 6$
10	Оконное преобразование Фурье с прямоугольным окном, $T_1 = 0$ с, $T_2 = 12$ с, $k = 3$

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Яцун, С.Ф. Датчики и обработка сигналов в мехатронике : учебное пособие / С.Ф. Яцун, П.А. Безмен ; Юго-Зап. гос. ун-т. - Курск : ЮЗГУ, 2014. - 238 с.
2. Жданов, С.А. Информационные системы : учебник / С.А. Жданов, М.Л. Соболева, А.С. Алфимова. – Москва : Прометей, 2015. – 302 с.
3. Воротников, С.А. Информационные устройства робототехнических систем : учебное пособие / С.А. Воротников. - М. : МГТУ им. Н. Э. Баумана, 2005. - 384 с.
4. Каменев, С.В. Автоматизация контрольно-измерительных операций : учебное пособие / С.В. Каменев, К.В. Марусич. – Оренбург : Оренбургский государственный университет, 2014. – 102 с.
5. Дворкович, В.П. Метрологическое обеспечение видеоинформационных систем / В.П. Дворкович, А.В. Дворкович. – Москва : Техносфера, 2015. – 784 с.
6. Прэт, У. Цифровая обработка изображений // М. –Мир.-1982. -т.1. - 495 с.-т.2. - 480с.
7. Рабинер, Л., Гоулд, Б. Теория и применения цифровой обработки сигналов // М. – Мир. -1978. - 545 с.
8. Уидроу, Б., Стирнз, С. Адаптивная обработка сигналов // М. - Радио и связь. - 1989, - 440 с.