

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра «Инфокоммуникационных технологий»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой
Даровских С.Н

“ _____ ” _____ 2017 г.

Разработка компьютерной модели для изучения механизмов анализа и синтеза периодических сигналов в тригонометрическом базисе Фурье

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ - Д 11.03.02.17.109.00 ПЗ

Руководитель работы
Вдовина Н.В. _____

“ _____ ” _____ 2017 г.

Автор работы
студент группы КЭ-438
Богомолов В.А. _____

“ _____ ” _____ 2017 г.

Нормоконтролер
Спицына В.Д. _____

“ _____ ” _____ 2017 г.

Челябинск
2017

РЕФЕРАТ

Богомолов В.А. Разработка
компьютерной модели для изучения
механизмов анализа и синтеза
периодических сигналов в
тригонометрическом базисе Фурье –
Челябинск: ЮУрГУ, КЭ-438, 2017. –
45 с., 9 ил., 3 табл., библиогр. список –
8 наим., 1 прил., 2 л. плакатов ф. А1.

Компьютерное моделирование, анализ и синтез сигналов в тригонометрическом базисе Фурье, амплитудный и фазовый спектры.

ВКР представляет собой программную разработку комплекса лабораторных работ по предмету «Математические методы представления информации». При помощи компьютерного моделирования были разработаны алгоритмы анализа и синтеза некоторых периодических функций, которые помогут студентам освоить соответствующий теоретический материал. В работе представлена визуализация результатов в виде различных графиков, что упрощает понимание математических выкладок.

Изложены теоретические положения, на которых основываются алгоритмы данной программы. Данный комплекс лабораторных работ был протестирован на правильность работы и готов к использованию.

					ЮУрГУ - Д 11.03.02.17.109.00 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Разработка компьютерной модели для изучения механизмов анализа и синтеза периодических сигналов в тригонометрическом базисе Фурье. Пояснительная записка	Лит.	Масса	Масштаб
Разраб.		Богомолов В.А.						
Провер.		Вдовина Н.В.						
Т. Контр.								
Реценз.						Лист 3	Листов 45	
Н. Контр.		Спицына В.Д.			2	ЮУрГУ Кафедра ИКТ		
Утверд.		Даровских С.Н.						

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 Анализ технического задания	6
1.1 Выбор и обоснование использования программных средств.....	6
2 Разработка интерфейса программы.....	8
3 Основные теоретические сведения о гармоническом анализе и синтезе периодических сигналов.....	10
4 Программная реализация.....	14
5 Алгоритм программы лабораторной работы.....	16
6 Тестирование разработанной программы.....	19
6.1 Тестирование работы программы с прямоугольным сигналом	19
6.2 Тестирование работы программы с пилообразным сигналом	23
6.3 Тестирование работы программы с треугольным сигналом	25
ЗАКЛЮЧЕНИЕ	27
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	28
ПРИЛОЖЕНИЕ А	29

ВВЕДЕНИЕ

ВКР представляет собой комплекс лабораторных работ, состоящий из двух частей, по дисциплине «Математические методы представления сигналов и процессов». Представленные лабораторные работы позволят студентам наглядно понять принципы анализа и синтеза различных сигналов. Отличительной особенностью проекта является тот факт, что приложение создано с помощью интерактивной среды программирования MATLAB и написано на одноименном языке. Так как в течение обучения по данной специальности студентам не раз придется работать с данным программным продуктом по различным дисциплинам, то его использование при написании этого приложения даст наглядное представление о возможностях и функционале MATLAB.

Разработанное приложение после задания основных параметров сигнала в качестве выполнения команды анализа генерирует сигнал одной из заданных форм: прямоугольной (меандр выведен отдельно от прямоугольного сигнала), треугольной или пилообразной, и строит его амплитудный и фазовый спектры. При выполнении синтеза сигнала предлагается выбрать номер гармоники, которая будет последней в ряде Фурье. Затем синтезируется сигнал с заданным числом гармоник и отображается на одном графике с идеальным сигналом. Также показываются рассчитанные для выбранной гармоники значения коэффициентов ряда Фурье, амплитуда и фаза.

Основной целью дипломного проекта было создание наглядного и понятного приложения для проведения анализа и синтеза сигналов на языке программирования от компании Matrix Laboratory, который все шире используется учеными и инженерами по всему миру.

1 Анализ технического задания

Результатом выполнения дипломного проекта должно стать создание приложения для выполнения комплекса лабораторных работ, состоящего из двух частей: «Анализ сигналов в тригонометрическом базисе Фурье» и «Синтез сигналов в тригонометрическом базисе Фурье», по дисциплине «Математические методы представления сигналов и процессов».

Приложение для выполнения лабораторных работ должно иметь возможности:

- выбора формы сигнала;
- управления параметрами сигнала;
- расчета и построения осциллограмм, расчета и построения спектров графиков, необходимых для визуализации полученных результатов;
- отображения основных формул, используемых для расчетов;
- отображения значений коэффициентов Фурье, амплитуды и фазы для заданной гармоники (для сравнения с значениями, рассчитанными студентами).

Для реализации перечисленных требований необходимо создать соответствующие алгоритмы, основанные на теоретических положениях.

Таким образом, после выполнения заданных требований студенты смогут разобраться с принципами анализа и синтеза сигналов, построив соответствующие графики и сравнив самостоятельно рассчитанные значения с таковыми, рассчитанными программой.

1.1 Выбор и обоснование использования программных средств

Так как на протяжении всего обучения студенты периодически будут работать в среде программирования MATLAB, то программное обеспечение для данного комплекса лабораторных работ имеет смысл выполнить с

использованием этого же языка программирования. Это позволит студентам лучше узнать о возможностях данного языка, понять его преимущества и недостатки.

Данный язык программирования нашел широкое применение в кругах ученых и инженеров, т. к. содержит разнообразный инструментарий, большое количество готовых специализированных математических функций, на разработку которых не нужно тратить время и производительность которых лучше или по крайней мере не хуже аналогов, созданных с помощью традиционных языков программирования, таких как C/C++, Java, Pascal и т.д. Обработка сигналов и связь, обработка изображений и видео, автоматизация тестирования и измерений, системы управления, вычислительная биология, финансовый инжиниринг – это только часть из тех научных областей, программирование в которых на языке MATLAB значительно упрощает и ускоряет процесс разработки.

Этот высокоуровневый язык используется в одноименной среде программирования – MATLAB. Для выполнения дипломного проекта был выбран релиз R2016a данной среды.

Наличие такой встроенной инструментальной среды, как App Designer, позволяет использовать визуальные компоненты интерфейса, которые представлены в достаточном для выполнения поставленных задач объеме.

2 Разработка интерфейса программы

Разберем структуру интерфейса разработанного приложения.

Строка меню лабораторных работ содержит два подменю:

а) «Анализ» – данный пункт выбирается автоматически при включении программы и обязателен к выполнению (именно в нем пользователь задает основные параметры сигнала) даже при необходимости работы в другом подменю. Пример стартового окна с по умолчанию активирующимся подменю «Анализ» показан на рисунке 2.1.

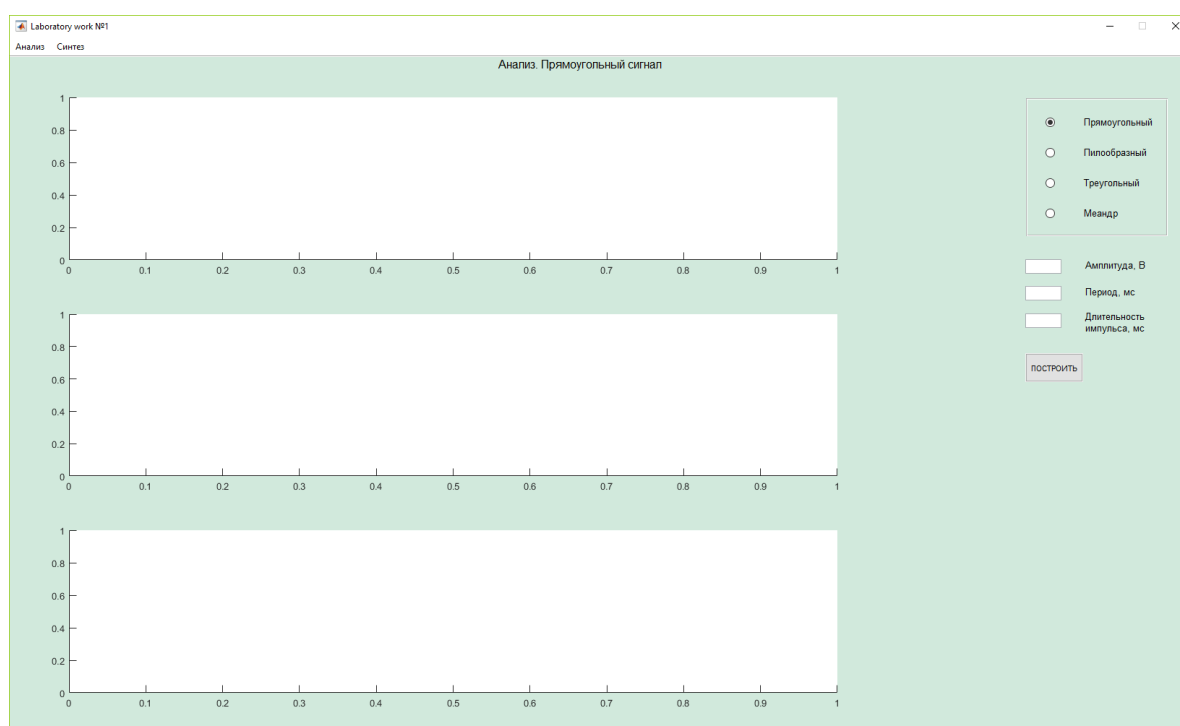


Рисунок 2.1 – Графическое окно подменю «Анализ»

б) «Синтез» – подменю, возможность выбрать которое появляется после выполнения построений в подменю «Анализ». Пример представлен на рисунке 2.2.

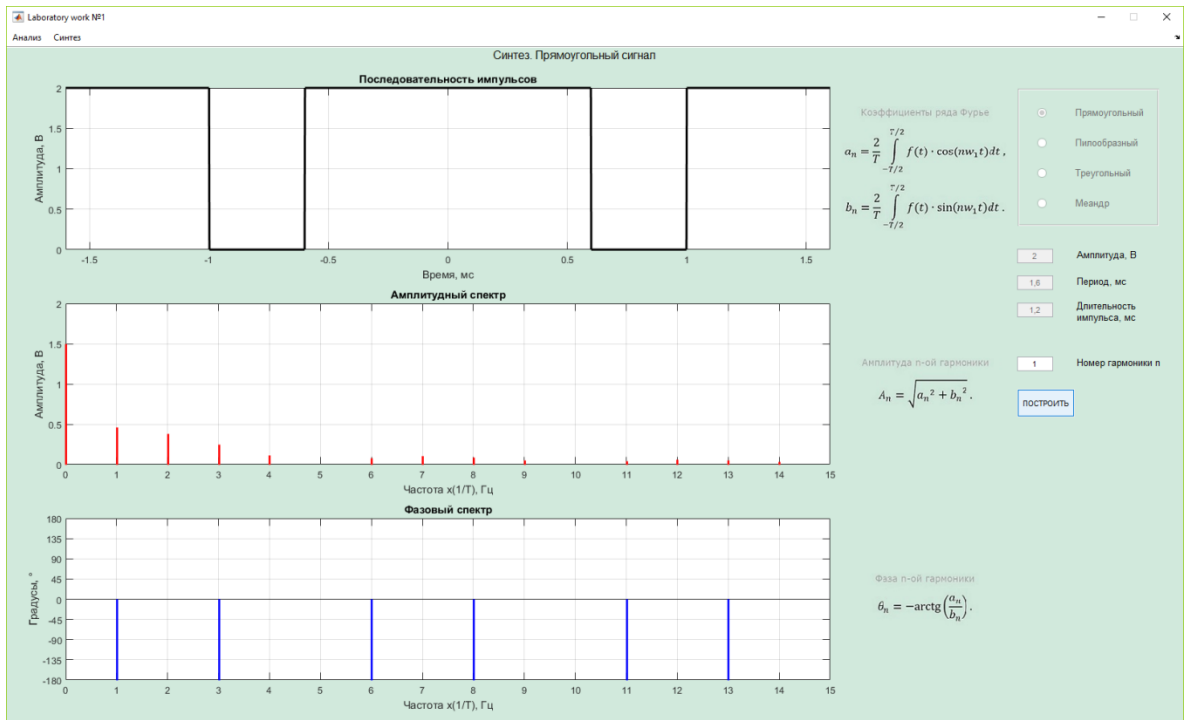


Рисунок 2.2 – Графическое окно подменю “Синтез”

3 Основные теоретические сведения о гармоническом анализе и синтезе периодических сигналов

Программа предоставляет возможности проведения спектрального анализа – разложения сигналов в ряд Фурье в тригонометрическом базисе, и синтеза – восстановления сигналов по их спектру. Далее изложим основные теоретические положения, на основе которых разрабатывались соответствующие алгоритмы.

Сигнал $s(t)$ можно представить в виде обобщенного ряда Фурье

$$s(t) = \sum_{n=0}^{\infty} c_n \varphi_n(t), \quad (3.1)$$

где c_n – коэффициенты ряда Фурье; $\{\varphi_n(t)\}$ – система гармонических колебаний с кратными частотами, взятая в качестве ортогональной. В комплексной форме данная система будет выглядеть следующим образом:

$$\{\varphi_n(t)\} = \{\dots, e^{-i2w_1 t}, e^{-iw_1 t}, 1, e^{iw_1 t}, e^{i2w_1 t}, \dots, e^{inw_1 t}, \dots\}. \quad (3.2)$$

Перепишем выражение (3.1) с учетом вышесказанного:

$$s(t) = \sum_{n=-\infty}^{\infty} c_n e^{inw_1 t}. \quad (3.3)$$

Все коэффициенты c_n ряда Фурье в базисе тригонометрических функций представляют собой частотный спектр периодического сигнала. Определим значения этих коэффициентов.

Зная квадрат нормы функции

$$\|\varphi_n\|^2 = \int_a^b \varphi_n(x) \varphi_n^*(x) dx = \int_a^b |\varphi_n(x)|^2 dx \quad (3.4)$$

и формулу обобщенного ряда Фурье

$$c_n = \frac{1}{\|\varphi_n\|^2} \int_a^b f(x) \varphi_n^*(x) dx, \quad (3.5)$$

где $\varphi_n^*(x)$ – функция, комплексно-сопряженная с функцией $\varphi_n(x)$, получим, что

$$\|\varphi_n\|^2 = \int_{-T/2}^{T/2} e^{inw_1 t} e^{-inw_1 t} dt = T, \quad (3.6)$$

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} s(t) e^{-inw_1 t} dt. \quad (3.7)$$

Подставив в (3.7) $e^{-inw_1 t} = \cos nw_1 t - i \sin nw_1 t$, получим

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} s(t) (\cos nw_1 t - i \sin nw_1 t) dt = c_{nR} - ic_{nI}, \quad (3.8)$$

где действительная косинусная и мнимая синусная части находятся по следующим формулам:

$$c_{nR} = \frac{1}{T} \int_{-T/2}^{T/2} s(t) \cos nw_1 t dt, \quad (3.9)$$

$$c_{nI} = \frac{1}{T} \int_{-T/2}^{T/2} s(t) \sin nw_1 t dt. \quad (3.10)$$

Также коэффициенты c_n можно записать в виде

$$c_n = |c_n| e^{i\theta_n}, \quad (3.11)$$

где

$$|c_n| = \sqrt{c_{nR}^2 + c_{nI}^2}, \quad (3.12)$$

$$\theta_n = -\text{arctg}(c_{nR}/c_{nI}). \quad (3.13)$$

Модуль $|c_n|$ является четной функцией дискретной переменной n , а аргумент θ_n – нечетной: $|c_n| = |c_{-n}|$, $\theta_n = -\theta_{-n}$.

Приведем выражение (3.3) к следующему виду

$$s(t) = \sum_{n=-\infty}^{\infty} |c_n| e^{i(n\omega_1 t + \theta_n)}. \quad (3.14)$$

Теперь выделим два слагаемых для фиксированного $|n|$ и рассмотрим векторную диаграмму этих комплексных составляющих ряда (рисунок 3.1), чтобы перейти к тригонометрической форме ряда Фурье. Угловая частота вращения векторов длиной $|c_n|$ составляет $n\omega_1$, и вращаются эти векторы в противоположных друг другу направлениях. Очевидно, что сумма проекций на мнимую ось равна нулю, а на действительную составляет $2|c_n| \cos(n\omega_1 t + \theta_n)$.

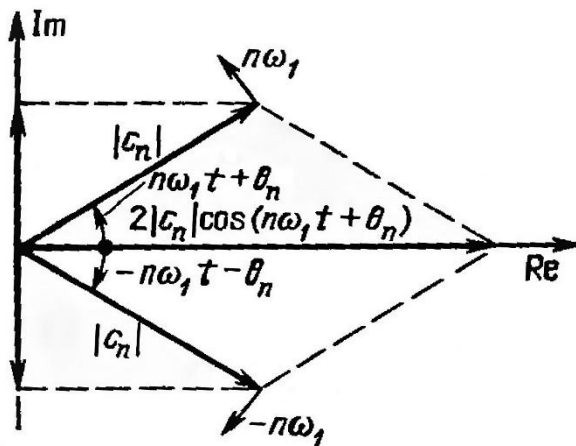


Рисунок 3.1 – Векторная диаграмма

Запишем ряд (3.14) в тригонометрической форме

$$s(t) = c_0 + \sum_{n=1}^{\infty} 2|c_n| \cos(n\omega_1 t + \theta_n). \quad (3.15)$$

Обычно выражение (3.15) записывают в виде

$$\begin{aligned} s(t) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega_1 t + b_n \sin n\omega_1 t) = \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty} A_n \cos(n\omega_1 t + \theta_n), \end{aligned} \quad (3.16)$$

где

$$\theta_n = -\arctg(b_n/a_n). \quad (3.17)$$

Из выражений (3.15) и (3.16) видно, что A_n – амплитуда n -ой гармоники, связана с $|c_n|$ – коэффициентом ряда (3.14), соотношением

$$A_n = 2|c_n|, \quad (3.18)$$

а $a_n = 2c_{nR}$, $b_n = 2c_{nI}$.

Т.е. при $n \geq 0$

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} s(t) \cos nw_1 t dt, \quad (3.19)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} s(t) \sin nw_1 t dt. \quad (3.20)$$

Из этих формул видно, что если сигнал $s(t)$ является четной функцией времени, то коэффициенты b_n равны нулю, и если $s(t)$ – нечетная функция, то нулю равны только коэффициенты a_n .

Амплитудная и фазовая характеристики полностью определяют структуру спектра. И в то же время при известном спектре можно синтезировать его сигнал. Наглядное представление о спектрах можно получить, построив их графики и по спектру синтезировав сигнал.

4 Программная реализация

В соответствии с теоретическими положениями для проведения анализа сигнала необходимо найти коэффициенты ряда a_n, b_n , а также найти и построить амплитудный и фазовый спектры A_n, θ_n . А для проведения процедуры синтеза необходимо знать структуру спектра.

Все эти действия возможно выполнить, используя при компьютерной обработке представление непрерывного сигнала в виде последовательности дискретных отсчетов k , период повторения которых называется периодом дискретизации T_k .

Для всех сигналов алгоритмы нахождения коэффициентов a_n и b_n осуществляются по формулам (3.19) и (3.20), значений амплитуд и фаз – по формулам (3.12), (3.18) и (3.17).

Учитывая, что общее число отсчетов составляет N , то дискретный вид данных формул подразумевает использование значения количества отсчетов N вместо периода T и вместо интегрирования на промежутке $[-T/2; T/2]$ суммирования в пределах $[0; N-1]$ с использованием вместо значений времени t дискретных отсчетов k .

Также, принимая во внимание особенности использования функции нахождения арктангенса на языке Matlab – $\text{arctg}(x)$ (находит арктангенс на промежутке $[-\pi/2; +\pi/2]$), уточним алгоритм нахождения значений фазы гармоник (3.17):

$$\theta_n = \begin{cases} -\operatorname{arctg}(b_n/a_n) & \text{при } a_n > 0, \\ -\operatorname{arctg}(b_n/a_n) - \pi & \text{при } a_n < 0 \text{ и } b_n \geq 0, \\ -\operatorname{arctg}(b_n/a_n) + \pi & \text{при } a_n < 0 \text{ и } b_n < 0, \\ -\pi/2 & \text{при } a_n = 0, b_n > 0, \\ \pi/2 & \text{при } a_n = 0, b_n < 0, \\ 0 & \text{при } a_n = 0, b_n = 0, \end{cases} \quad (4.1)$$

Алгоритмы анализа и синтеза для всех наших сигналов одинаковые и уже определены, поэтому остается найти только сами функции сигналов $s(t)$.

а) Прямоугольный сигнал (меандр)

Сам одиночный прямоугольный импульс зададим формулой

$$s(t) = \begin{cases} U_0 + E, & -ti/2 \leq t \leq ti/2, \\ 0, & t < -ti/2 \text{ или } t > ti/2, \end{cases} \quad (4.2)$$

где ti – длительность импульса, E – амплитуда, U_0 – сдвиг по амплитуде.

Для меандра – прямоугольного сигнала со скважностью 2, формула, задающая форму импульса, будет такой же.

б) Треугольный сигнал

$$s(t) = \begin{cases} (2E/ti)t + E, & -ti/2 \leq t \leq 0 \\ -(2E/ti)t + E, & 0 < t \leq ti/2 \\ 0, & t < -ti/2 \text{ или } t > ti/2 \end{cases} . \quad (4.3)$$

в) Пилообразный сигнал

$$s(t) = \begin{cases} (2E/ti)t + E, & -ti/2 \leq t \leq ti/2 \\ 0, & t < -ti/2 \text{ или } t > ti/2 \end{cases} . \quad (4.4)$$

5 Алгоритм программы лабораторной работы

Алгоритм работы программы приведен на рисунке 5.1. После запуска программы по умолчанию появляется окно с уже активным подменю «Анализ» с пунктами, обязательными к выполнению. После указания значений параметров сигнала и нажатия на кнопку «Построить» выполняются построения графиков самого сигнала, амплитудного и фазового спектров единичного импульса выбранного сигнала. Также приводятся формулы и их расчетные значения для спектральных коэффициентов. Выполнив построения в данном подменю, пользователь может перейти к подменю «Синтез», в котором предлагается выбрать номер гармоники, которая будет последней в ряде Фурье при восстановлении сигнала. Коэффициенты ряда Фурье и значения амплитуд и фаз берутся из предыдущего пункта. После нажатия кнопки «Построить» на одном графике с идеальным сигналом появляется синтезированный, с заданным нами числом гармоник.

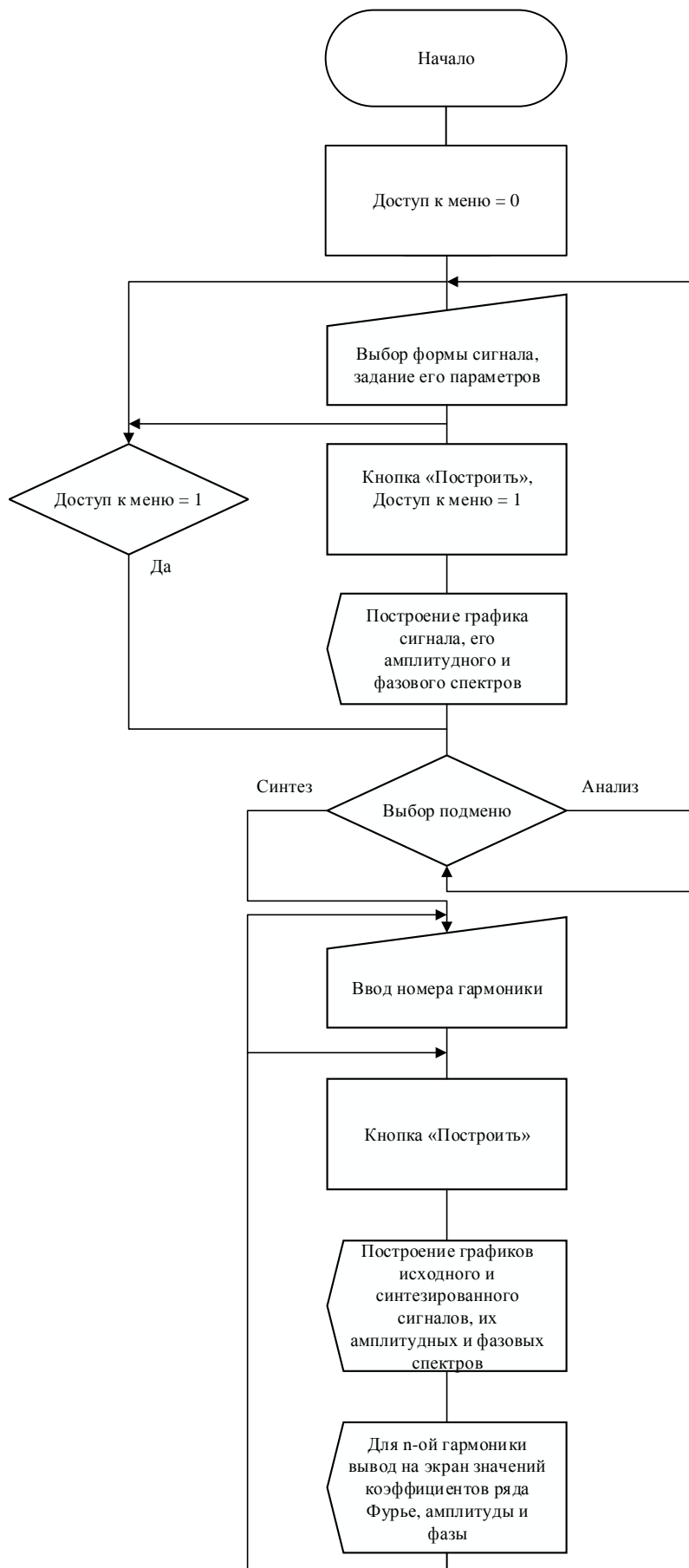


Рисунок 5.1 – Алгоритм работы программы

6 Тестирование разработанной программы

Работа данного приложения была протестирована с каждым из представленных сигналов и при различных параметрах. Теперь построим графики последовательностей импульсов, амплитудных и фазовых спектров, рассчитаем спектральные коэффициенты и сравним их с расчетными, а также восстановим сигналы по их спектрам.

6.1 Тестирование работы программы с прямоугольным сигналом

В подменю «Анализ» выберем сигнал прямоугольной формы и зададим следующие значения параметров сигнала:

- амплитуда $E = 1$ В;
- длительность импульса $t_i = 0,2$ мс;
- период повторения $T = 1$ мс.

Данные параметры будем использовать для тестирования и остальных сигналов.

Нажмем кнопку «Построить», чтобы вывести графики самой последовательности импульсов, амплитудного и фазового спектров. Результат этих действий представлен на рисунке 6.1.

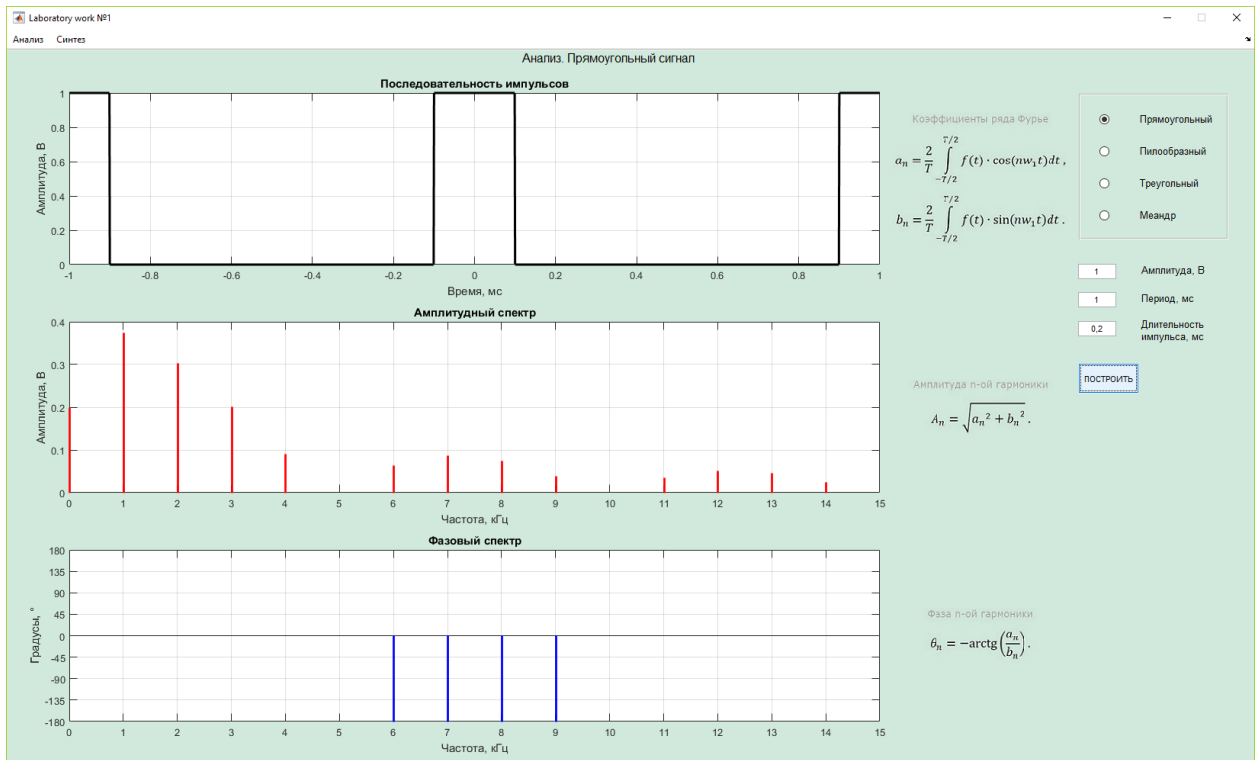


Рисунок 6.1 – Тестирование прямоугольного сигнала в подменю «Анализ»

Далее перейдем к подменю «Синтез» и синтезируем сигнал при указанном числе гармоник (проверим все: от 1 до 15). Затем нажмем на кнопку «Построить», после чего появятся графики синтезированного сигнала, соответствующие амплитудный и фазовый спектры, а также будут показаны коэффициенты ряда Фурье и значения фазы и амплитуды последней n-ой гармоники (рисунок 6.2). На данном рисунке число гармоник выбрано равным восьми.

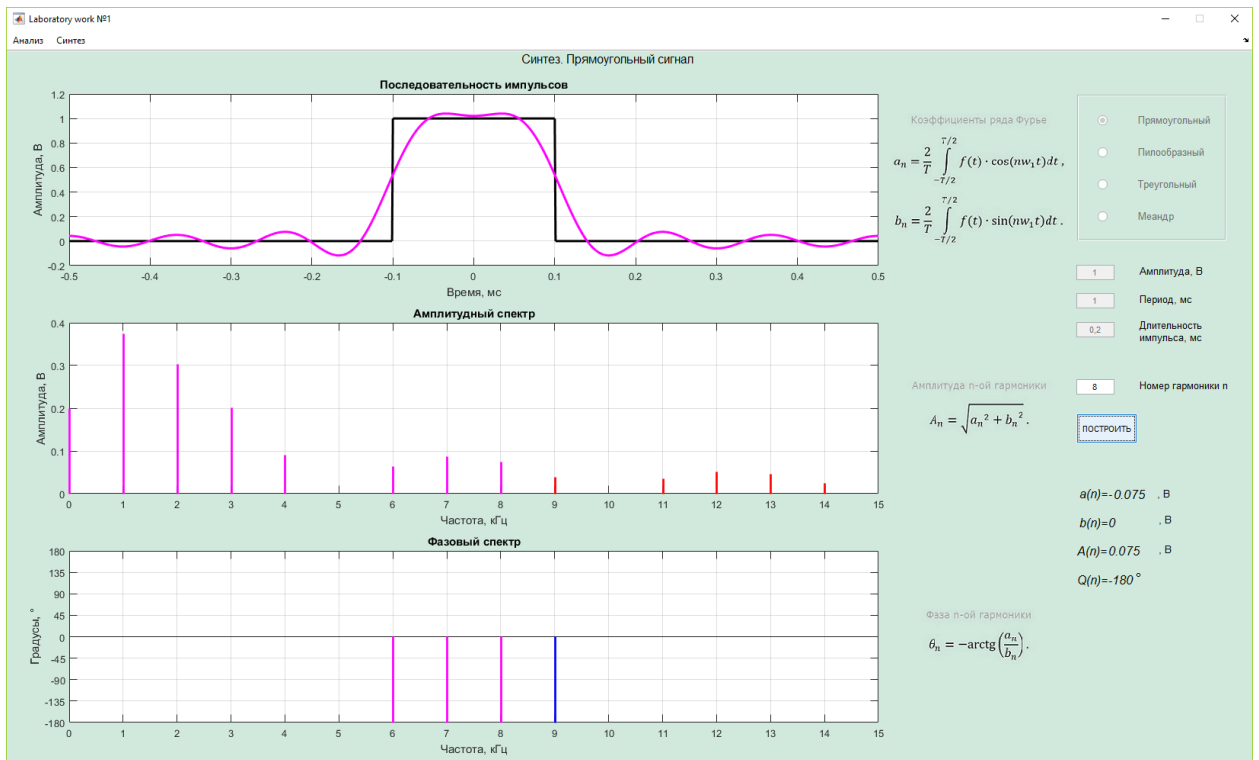


Рисунок 6.2 – Тестирование прямоугольного сигнала в подменю «Синтез»

Составим сводную таблицу значений амплитуд и фаз гармоник, полученных программным путем и рассчитанных теоретически (таблица 6.1).

Данная таблица составляется для того, чтобы оценить точность компьютерных вычислений.

Таблица 6.1 – Тестирование прямоугольного сигнала

n	Расчет программы		Теоретический расчет	
	An, B	$\theta n, ^\circ$	An, B	$\theta n, ^\circ$
0	0,2008	0	0,2008	0
1	0,3754	0	0,3754	0
2	0,303	0	0,303	0
3	0,201	0	0,201	0
4	0,0918	0	0,0918	0
5	0	0	0	0
6	0,0639	-180	0,0639	-180
7	0,087	-180	0,087	-180
8	0,075	-180	0,075	-180
9	0,0399	-180	0,0399	-180
10	0	0	0	0
11	0,0356	0	0,0356	0
12	0,051	0	0,051	0
13	0,0459	0	0,0460	0
14	0,0251	0	0,0251	0
15	0	0	0	0

Из таблицы видно, что значения совпадают как минимум до третьего знака. Это говорит о высокой точности производимых программой расчетов.

Так как меандр является частным случаем прямоугольного сигнала (скважность $Q = 2$), то алгоритмы анализа и синтеза одинаковые. Разница заключается лишь в реализации способа ввода значений: при выборе меандра в подменю «Анализ» поле ввода длительности импульса недоступно и заполняется автоматически.

6.2 Тестирование работы программы с пилообразным сигналом

Параметры сигнала возьмем прежними. По аналогии с предыдущим пунктом произведём анализ и синтез сигнала. Для синтеза возьмем 11 гармоник. Выполнение этих действий проиллюстрировано на рисунках 6.3 и 6.4.

Помимо этого, как и для прямоугольного сигнала, составим сводную таблицу значений амплитуд и фаз, полученных программно или теоретически (таблица 6.2).

Таблица 6.2 – Тестирование пилообразного сигнала

n	Расчет программы		Теоретический расчет	
	A_n, B	$\theta_n, ^\circ$	A_n, B	$\theta_n, ^\circ$
0	0,1004	0	0,1004	0
1	0,1921	-12,264	0,1921	-12,264
2	0,1678	-25,463	0,1678	-25,463
3	0,133	-40,943	0,133	-40,943
4	0,0953	-61,191	0,0953	-61,191
5	0,0636	-90,9	0,0636	-90,9
6	0,0467	-133,21	0,0467	-133,21
7	0,0436	-175,71	0,0436	-175,71
8	0,0428	151,031	0,0428	151,031
9	0,0384	121,285	0,0384	121,285
10	0,0318	88,2	0,0318	88,2
11	0,027	48,8086	0,0271	48,808
12	0,0258	8,6752	0,0258	8,6752
13	0,0256	-26,332	0,0257	-26,332
14	0,024	-58,52	0,024	-58,52
15	0,0212	-92,7	0,0213	-92,7

После сравнения данных в таблице можно говорить о высокой точности вычислений программы. Как минимум до третьего знака после запятой результаты совпадают.

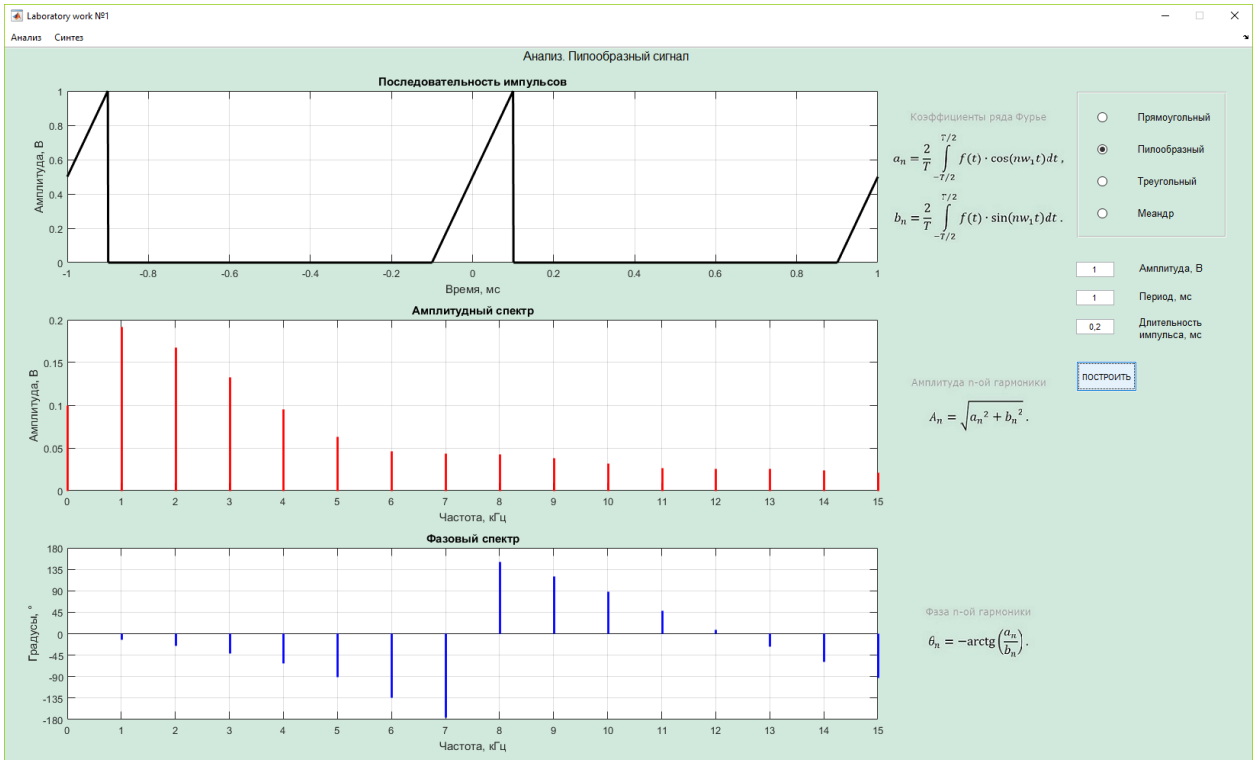


Рисунок 6.3 – Тестирование пилообразного сигнала в подменю «Анализ»

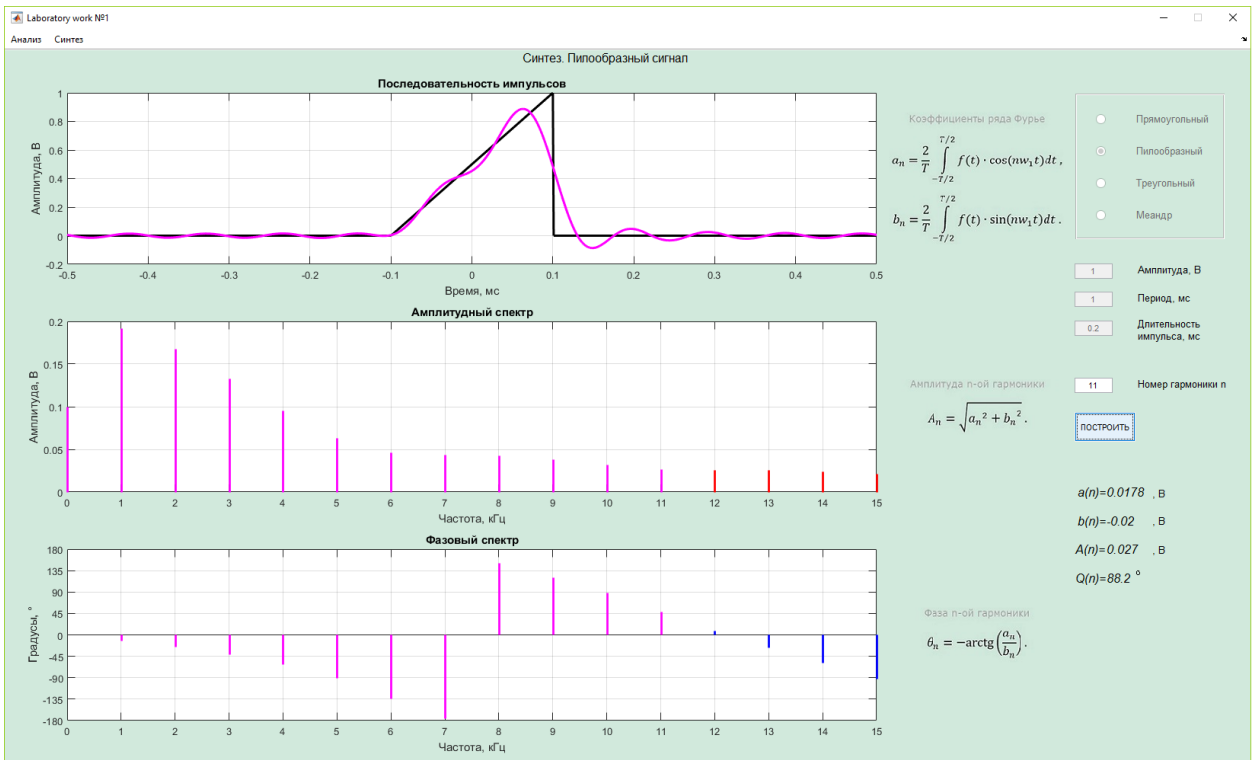


Рисунок 6.4 – Тестирование пилообразного сигнала в подменю «Синтез»

6.3 Тестирование работы программы с треугольным сигналом

Как и в предыдущих пунктах, воспользуемся теми же параметрами сигнала, за исключением формы. Проведем анализ сигнала и синтез (возьмем три гармоники). Результаты работы программы показаны на рисунках 6.5 и 6.6.

Составим сводную таблицу значений амплитуд и фаз.

Таблица 6.3 – Тестирование треугольного сигнала

n	Расчет программы		Теоретический расчет	
	$A_n, В$	$\theta_n, ^\circ$	$A_n, В$	$\theta_n, ^\circ$
0	0,0999	0	0,0999	0
1	0,1933	0	0,1933	0
2	0,1749	0	0,1749	0
3	0,1472	0	0,1472	0
4	0,1144	0	0,1144	0
5	0,081	0	0,081	0
6	0,0509	0	0,0509	0
7	0,027	0	0,027	0
8	0,0109	0	0,0109	0
9	0,0024	0	0,0024	0
10	0	0	0	0
11	0	0	0	0
12	0,0049	0	0,0049	0
13	0,0078	0	0,0078	0
14	0,0093	0	0,0094	0
15	0,009	0	0,0091	0

Результаты практически не отличаются.

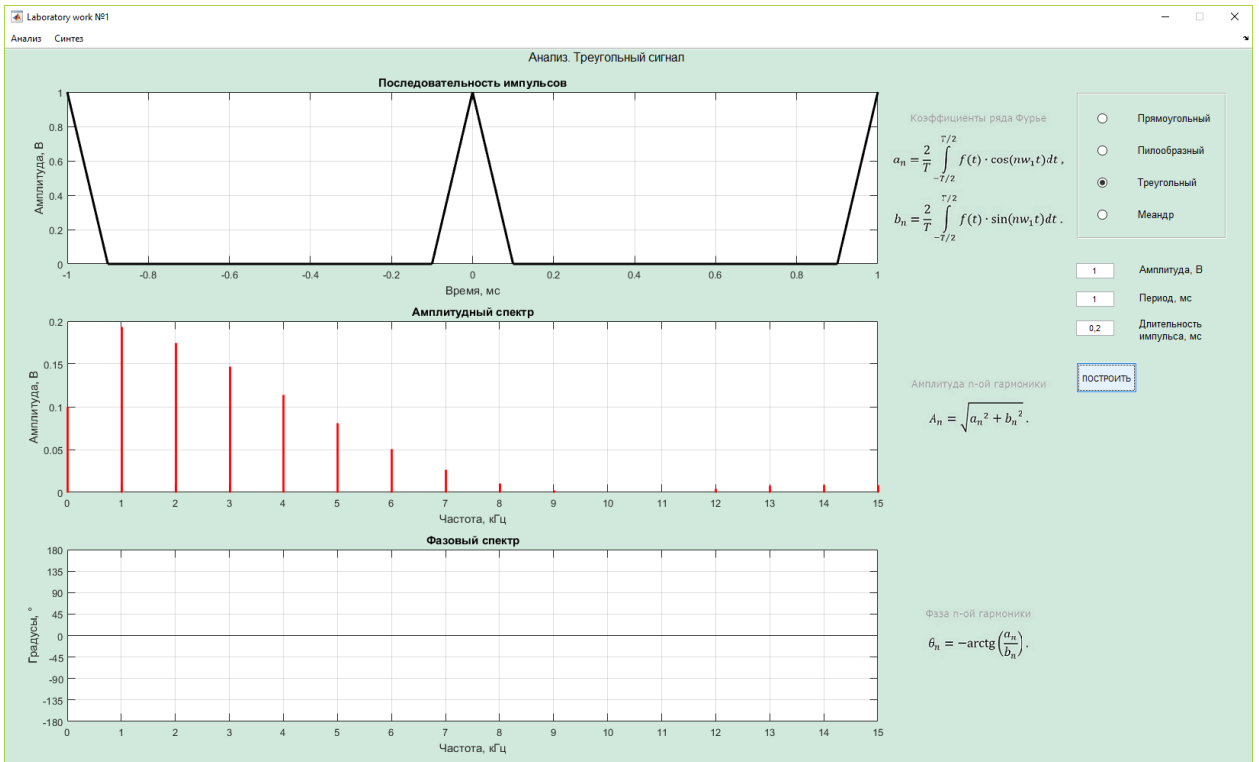


Рисунок 6.5 – Тестирование треугольного сигнала в подменю «Анализ»

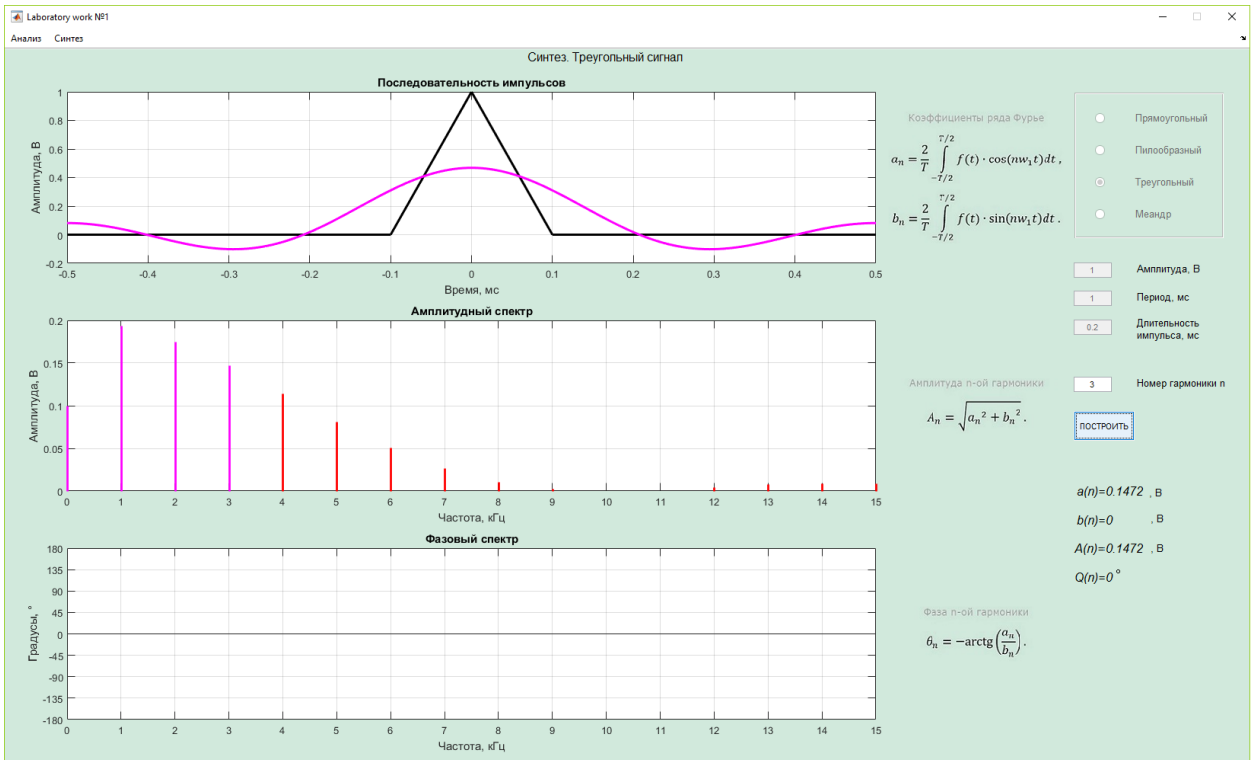


Рисунок 6.6 – Тестирование треугольного сигнала в подменю «Синтез»

ЗАКЛЮЧЕНИЕ

Результатом выполнения выпускной квалификационной работы является разработанная компьютерная модель на языке MATLAB, оформленная в виде приложения. Компьютерное моделирование анализа и синтеза некоторых периодических сигналов в тригонометрическом базисе Фурье, представленное данным приложением, нацелено на использование в учебных целях для облегчения понимания материала студентами. Помимо этого, код программы, написанный на данном языке, дает возможность студенту углубить свои знания по программированию в MATLAB.

В данной программе процессы анализа и синтеза не содержат встроенных специализированных функций для нахождения спектров. Это сделано для того, чтобы студент при необходимости мог проследить каждый этап алгоритмов анализа и синтеза периодических сигналов. Программа предоставляет возможности визуализации полученных результатов, что значительно способствует усвоению материала.

Таким образом, был создан удобный и наглядный программный продукт, отвечающий требованиям ТЗ, для лабораторных работ по дисциплине «Математические методы представления сигналов и процессов».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гоноровский И. С. Радиотехнические цепи и сигналы / Гоноровский И.С., Демин М.П. – М.: Радиотехнические цепи и сигналы: Учеб. пособие для вузов. – 5-е изд., перераб. и доп. – М.: Радио и связь, 1994. – 480 с.
2. Дьяконов, В. MATLAB обработка сигналов и изображений. Специальный справочник / В. Дьяконов, И. Абраменкова. – СПб.: Питер, 2002. — 608 с.
3. Баскаков С. И. Радиотехнические цепи и сигналы / С.И. Баскаков. – М.: Высшая школа, 1988. – 504 с.
4. Потемкин В.Г. Инструментальные средства Matlab 5.x. / В. Г. Потемкин. – М.: Диалог-МИФИ, 2000. – 206 с.
5. Дьяконов, В. П. MATLAB. Анализ, идентификация и моделирование систем / В. П. Дьяконов, В. В. Круглов. – СПб.: Питер, 2001. — 448 с.
6. Лазарев, Ю. Моделирование процессов и систем в MATLAB / Юрий Лазарев; Учебный курс. – СПб.: Питер; Киев: Издательская группа ВНУ, 2005. – 512 с.
7. Сергиенко А. Б. Цифровая обработка сигналов. / А. Б. Сергиенко. – СПб.: Питер, 2002. – 507 с.
8. Марпл С. Л. Цифровой спектральный анализ / С. Л. Марпл. – М.: Мир, 1990. – 378 с.

ПРИЛОЖЕНИЕ А

Листинг программы

```
function furia

clc; clear; close all;

scrsz = get(0, 'ScreenSize');
X1 = scrsz(1)+scrsz(3)/13.5; Y1 = scrsz(2)+scrsz(4)/15;
X2 = scrsz(3)-2*scrsz(3)/13.5; Y2 = scrsz(4)-2*scrsz(4)/15;

figure('MenuBar', 'None', ...
    'Name', 'Laboratory work №1', ...
    'NumberTitle', 'Off', ...
    'Color', [0.819608 0.913725 0.862745], ...
    'Position', [X1, Y1, X2, Y2]', ...
    'Resize', 'off');

uimenu('Label','Анализ', 'Callback', @Analysis);
uimenu('Label','Синтез', 'Callback', @Synthesis);

    global E;
    global ti;
    global T;
    global u1;
    global u2;
    global u3;
    global n;
    global p;
    global t;
    global score;
    global a;
    global an;
    global anValue;
    global bn;
    global bnValue;
    global A;
    global Q;
    global StemHandle1;
    global StemHandle2;
    global StemHandle3;
    global Availability;
    global CC;
    global S;
    global Sintplot;

%=====

Availability = 0;
score = 1;
```

```

T = 0;
ti = 0;
E = 0;
P_s = uibuttongroup('Position', [0.86, .7345, .12,
.205], ...
    'BackgroundColor', [0.819608 0.913725 0.862745]);

Bre = uicontrol('Style', 'RadioButton', ...
    'String', '        Прямоугольный', ...
    'Position', [25, 148 , 150, 20], ...
    'Parent', P_s, ...
    'FontSize', 10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Callback', @Rec_chosen);

Bme = uicontrol('Style', 'RadioButton', ...
    'String', '        Меандр', ...
    'Position', [25, 22 , 150, 20], ...
    'Parent', P_s, ...
    'FontSize', 10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Callback', @M_chosen);

Btr = uicontrol('Style', 'RadioButton', ...
    'String', '        Треугольный', ...
    'Position', [25, 64 , 150, 20], ...
    'Parent', P_s, ...
    'FontSize', 10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Callback', @Tr_chosen);

Bsa = uicontrol('Style', 'RadioButton', ...
    'String', '        Пилообразный', ...
    'Position', [25, 106 , 150, 20], ...
    'Parent', P_s, ...
    'FontSize', 10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Callback', @Saw_chosen);

Name_sig = uicontrol('Style', 'Text', ...
    'String', 'Анализ. Прямоугольный сигнал', ...
    'HorizontalAlignment', 'center', ...
    'Visible', 'on', ...
    'FontSize', 12, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [X2/2-scrcsize(3)/13.5, 0.98*Y2 , 230,
20]);

```

```
function Rec_chosen(h, eventdata, handles)
```

```

        set(Name_sig, 'String', 'Анализ. Прямоугольный
сигнал');
        score = 1;
        set(ti_but, 'Enable', 'on');
    end

    function Tr_chosen(h, eventdata, handles)
        set(Name_sig, 'String', 'Анализ. Треугольный
сигнал');
        score = 2;
        set(ti_but, 'Enable', 'on');
    end

    function Saw_chosen(h, eventdata, handles)
        set(Name_sig, 'String', 'Анализ. пилообразный
сигнал');
        score = 3;
        set(ti_but, 'Enable', 'on');
    end

    function M_chosen(h, eventdata, handles)
        set(Name_sig, 'String', 'Анализ. Меандр');
        score = 4;

        if T > 0
            a = num2str(T/2);
            set(ti_but, 'String', a);
            ti = T/2;
        else
            set(ti_but, 'String', '');
        end

        set(ti_but, 'Enable', 'off');
    end

end

%=====

E_but = uicontrol('Style', 'Edit', ...
    'String', '', ...
    'Position', [0.86*X2, 0.68*Y2 , 50, 20], ...
    'Callback', @Take_E);

uicontrol('Style', 'Text', ...
    'String', 'Амплитуда, В', ...
    'FontSize', 10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'HorizontalAlignment', 'left', ...
    'Position', [0.91*X2, 0.68*Y2 , 100, 20]);

T_but = uicontrol('Style', 'Edit', ...
    'String', '', ...
    'Position', [0.86*X2, 0.64*Y2, 50, 20], ...

```

```

        'Callback', @Take_T);

uicontrol('Style', 'Text', ...
    'String', 'Период, мс', ...
    'FontSize',10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'HorizontalAlignment', 'left', ...
    'Position', [0.91*X2, 0.64*Y2 , 100, 20]);

ti_butt = uicontrol('Style', 'Edit', ...
    'String', '', ...
    'Position', [0.86*X2, 0.6*Y2, 50, 20], ...
    'Callback', @Take_ti);

uicontrol('Style', 'Text', ...
    'String', 'Длительность импульса, мс', ...
    'HorizontalAlignment', 'left', ...
    'FontSize',10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.91*X2, 0.583*Y2 , 100, 40]);

function Take_E(h, eventdata, handles)
    E0 = get(E_butt, 'String');
    E0 = strrep(E0, ',', '.', '');
    E = str2num(E0);
end

function Take_T(h, eventdata, handles)
    T0 = get(T_butt, 'String');
    T0 = strrep(T0, ',', '.', '');
    T = str2num(T0);

    if score == 4
        a = num2str(T/2);
        set(ti_butt, 'String', a);
        ti = T/2;
    else
    end

end

function Take_ti(h, eventdata, handles)

    if score == 4
    else
        ti0 = get(ti_butt, 'String');
        ti0 = strrep(ti0, ',', '.', '');
        ti = str2num(ti0);
    end

end
end

```



```

%=====
%=====

hA1 = axes('Position', [0.05 0.70 0.65 0.24]);
hA2 = axes('Position', [0.05 0.38 0.65 0.24]);
hA3 = axes('Position', [0.05 0.06 0.65 0.24]);

GenGr = uicontrol('Style', 'PushButton', ...
    'String', 'ПОСТРОИТЬ', ...
    'Position', [0.86*X2, 0.52*Y2 , 80, 40], ...
    'Callback', {@Make_Gr, hA1, hA2, hA3});

function Make_Gr(h, eventdata, hA1, hA2, hA3)

if ((E == 0) || (T == 0) || (ti == 0))
    msgbox ('Введите значения параметров сигнала!', ...
        'Warning', 'warn', 'Modal');
else
    F1 = axes('Position', [0.71 0.70 0.14 0.24], ...
        'Visible', 'off');
    Upload1 = imread('1111.jpg');
    imshow(Upload1);
    F2 = axes('Position', [0.71 0.38 0.14 0.24], ...
        'Visible', 'off');
    Upload2 = imread('2222.jpg');
    imshow(Upload2);
    F3 = axes('Position', [0.71 0.06 0.14 0.24], ...
        'Visible', 'off');
    Upload3 = imread('3333.jpg');
    imshow(Upload3);

%=====

if ((score == 1) || (score == 4))

%-----

axes (hA1);

r = -T : T : T;
tt = -T : T/1000 : T;
signal = E*pulstran(tt,r,'rectpuls',ti);
axes(hA1);
plot(tt, signal, 'LineWidth', 2, 'Color',
'black');

xlim([-T T]);
title('Последовательность импульсов');
ylabel('Амплитуда, В');
xlabel('Время, мс');
grid on;

dt = 0;
U0 = 0;

```

```

n = 1;

for i = -T/2 : T/1000 : T/2
    t(n) = i;
    if ((i >= -ti/2+dt)&&(i <= ti/2+dt))
        u1(n) = E + U0;
    else
        u1(n) = 0;
    end
    n = n + 1;

end

N = 1001;

```

%-----

```

for n = 0 : 1 : 15

    % Расчет коэффициента a1
    an = (1/N)*sum(u1.*cos(2*pi*n.*t));
    if abs(an) <= 0.001
        an = 0;
    end;
    disp(an);
    %Расчет коэффициента b1
    bn = (1/N)*sum(u1.*sin(2*pi*n.*t));
    if abs(bn) <= 0.001
        bn = 0;
    end;
    disp(bn);
    % Расчет составляющих амплитудного спектра
    A(n+1) = 2*sqrt(an*an+bn*bn);

    if n == 0
        A(n+1)=0.5*A(n+1);
    end;

    % Расчет составляющих фазового спектра
    % Вычисляем фазу Qn
    if (an>0)
        Q(n+1)=-atan(bn/an);      %Первый и
четвертый квадранты

    elseif (an<0)

        if (bn>=0) %Второй квадрант
            Q(n+1)=-atan(bn/an)-pi;
        end;

        if (bn<0) %Третий квадрант

```

```

        Q(n+1)=-atan(bn/an)+pi;
    end;

elseif (an==0) % Особая ситуация an=0

    if (bn>0)
        Q(n+1)=-pi/2;
    end;

    if (bn<0)
        Q(n+1)=pi/2;
    end;

    if (bn==0)
        Q(n+1)=0;
    end;
end;

% disp(Q(n+1));

%Устанавливаем порог коэффициентов A и Q
if (A(n+1)<0.01)
    Q(n+1)=0;
    A(n+1)=0;
end;

if abs(Q(n+1))<0.001
    Q(n+1)=0;
end;

if (abs(Q(n+1))>pi-pi/100)
    Q(n+1)=-pi;
end;

%Переводим фазу в градусы
Q(n+1)=Q(n+1)*57.7;

k = k + 1;
end

axes(hA2);
f = 0 : 1 : 15;
stem(f,A, 'LineWidth', 2, 'Color', 'red',
'Marker', 'none');
set(gca, 'XTick', [0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15]);
xlim([0 15]);
title('Амплитудный спектр');
ylabel('Амплитуда, В');
xlabel('Частота x(1/T), Гц');
grid on;

axes(hA3);

```

```

StemHandle1 = stem(f,Q, 'LineWidth', 2, 'Color',
'b', 'Marker', 'none');
set(gca, 'XTick', [0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15]);
xlim([0 15]);
set(gca, 'YTick', [-180 -135 -90 -45 0 45 90 135
180]);
ylim([-180 180]);
title('Фазовый спектр');
xlabel('Частота  $\omega$  (1/T), Гц');
ylabel('Градусы,  $^\circ$ ');
grid on;

```

```

%=====

```

```

elseif (score == 2)

```

```

%-----

```

```

i = -T : T : T;
t = -T : T/1000 : T;
signal = E*pulstran(t,i,'tripuls',ti,0);
axes(hA1);
plot(t, signal, 'LineWidth', 2, 'Color',
'black');
title('Последовательность импульсов');
ylabel('Амплитуда, В');
xlabel('Время, мс');
grid on;

```

```

%-----

```

```

p = 1;
dt = 0;

for i = -T/2 : T/1000 : T/2
    t(p) = i;
    %находим u2 через прямые
    if ((i >= -ti/2+dt)&&(i <= 0))
        u2(p) = t(p)*2*E/ti + E;
    elseif ((i > 0 + dt)&&(i <= ti/2+dt))
        u2(p) = t(p)*(-2)*E/ti + E;
    else
        u2(p) = 0;
    end
    p = p + 1;
end

```

```

%-----

```

```

N = 1001;
for p = 0 : 1 : 15

```

```

    % Расчет коэффициента a1

```

```

an = (1/N)*sum(u2.*cos(2*pi*p.*t));
if abs(an) <= 0.001
    an = 0;
end;
%
    disp(an);
%Расчет коэффициента b1
bn = (1/N)*sum(u2.*sin(2*pi*p.*t));
if abs(bn) <= 0.031
    bn = 0;
end;
%
    disp(bn);
% Расчет составляющих амплитудного спектра
A(p+1) = 2*sqrt(an*an+bn*bn);

if p == 0
    A(p+1)=0.5*A(p+1);
end;

% Расчет составляющих фазового спектра
% Вычисляем фазу Qn
if (an>0)
    Q(p+1)=-atan(bn/an);    %Первый и
четвертый квадранты

elseif (an<0)

    if (bn>=0)    %Второй квадрант
        Q(p+1)=-atan(bn/an)-pi;
    end;

    if (bn<0)    %Третий квадрант
        Q(p+1)=-atan(bn/an)+pi;
    end;

elseif (an==0)    % Особая ситуация an=0

    if (bn>0)
        Q(p+1)=-pi/2;
    end;

    if (bn<0)
        Q(p+1)=pi/2;
    end;

    if (bn==0)
        Q(p+1)=0;
    end;

end;
%Устанавливаем порог коэффициентов A и Q
if (A(p+1)<0.01)
    Q(p+1)=0;

```

```

        A(p+1)=0;
    end;

    if abs(Q(p+1))<0.001
        Q(p+1)=0;
    end;

    if (abs(Q(p+1))>pi-pi/100)
        Q(p+1)=-pi;
    end;

    %Переводим фазу в градусы
    Q(p+1)=Q(p+1)*57.7;

end

axes(hA2);
f = 0 : 1 : 15;
stem(f,A, 'LineWidth', 2, 'Color', 'red',
'Marker', 'none');
set(gca, 'XTick', [0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15]);
xlim([0 15]);
title('Амплитудный спектр');
ylabel('Амплитуда, В');
xlabel('Частота x(1/T), Гц');
grid on;
axes(hA3);
StemHandle2 = stem(f,Q, 'LineWidth', 2, 'Color',
'b', 'Marker', 'none');
set(gca, 'XTick', [0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15]);
xlim([0 15]);
set(gca, 'YTick', [-180 -135 -90 -45 0 45 90 135
180]);
ylim([-180 180]);
title('Фазовый спектр');
xlabel('Частота x(1/T), Гц');
ylabel('Градусы, °');
grid on;
%=====

elseif (score == 3)
%-----

    i = -T : T : T;
    t = -T : T/1000 : T;
    signal = E*pulstran(t,i,'tripuls',ti,1);
    axes(hA1);
    plot(t, signal, 'LineWidth', 2, 'Color',
'black');
    title('Последовательность импульсов');

```

```

ylabel('Амплитуда, В');
xlabel('Время, мс');
grid on;
%-----

p = 1;
dt = 0;
for i = -T : T/1000 : T
    t(p) = i;
    %находим u3 через уравнение отрезка прямой
(гипотенузы) треугольника
    if ((i >= -ti/2+dt)&&(i <= ti/2+dt))
        u3(p) = t(p)*E/ti + E/2;
    else
        u3(p) = 0;
    end
    p = p + 1;
end

N = length(u3);
for p = 0 : 1 : 15

    % Расчет коэффициента a1
    an = (1/N)*sum(u3.*cos(2*pi*p.*t));
    if abs(an) <= 0.0001
        an = 0;
    end;
    disp(an);
    %Расчет коэффициента b1
    bn = (1/N)*sum(u3.*sin(2*pi*p.*t));
    if abs(bn) <= 0.0001
        bn = 0;
    end;
    disp(bn);
    % Расчет составляющих амплитудного спектра
    A(p+1) = 2*sqrt(an*an+bn*bn);
    if p == 0
        A(p+1)=0.5*A(p+1);
    end;

    % Расчет составляющих фазового спектра
    % Вычисляем фазу Qn
    if (an>0)
        Q(p+1)=-atan(bn/an);    %Первый и
четвертый квадранты

    elseif (an<0)

        if (bn>=0)    %Второй квадрант
            Q(p+1)=-atan(bn/an)-pi;
        end;
    end;
end;

```

```

        if (bn<0)    %Третий квадрант
            Q(p+1)=-atan(bn/an)+pi;
        end;

elseif (an==0)    % Особая ситуация an=0

    if (bn>0)
        Q(p+1)=-pi/2;
    end;

    if (bn<0)
        Q(p+1)=pi/2;
    end;

    if (bn==0)
        Q(p+1)=0;
    end;

end;

%Устанавливаем порог коэффициентов А и Q
if (A(p+1)<0.01)
    Q(p+1)=0;
    A(p+1)=0;
end;

if abs(Q(p+1))<0.001
    Q(p+1)=0;
end;

if (abs(Q(p+1))>pi-pi/100)
    Q(p+1)=-pi;
end;

%Переводим фазу в градусы
Q(p+1)=Q(p+1)*57.7;

end
axes(hA2);
f = 0 : 1 : 15;
stem(f,A, 'LineWidth', 2, 'Color', 'red',
'Marker', 'none');
set(gca, 'XTick', [0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15]);
xlim([0 15]);
title('Амплитудный спектр');
ylabel('Амплитуда, В');
xlabel('Частота x(1/T), Гц');
grid on;

axes(hA3);
StemHandle3 = stem(f,Q, 'LineWidth', 2, 'Color',
'b', 'Marker', 'none');

```



```

        set(gca, 'XTick', [0 1 2 3 4 5 6 7 8 9 10 11 12
13 14 15]);
        xlim([0 15]);
        set(gca, 'YTick', [-180 -135 -90 -45 0 45 90 135
180]);
        ylim([-180 180]);
        title('Фазовый спектр');
        xlabel('Частота x(1/T), Гц');
        ylabel('Градусы, °');
        grid on;
    end

    Availability = 1;
end

end

%=====

Name_sigSint = uicontrol('Style', 'Text', ...
    'String', 'Анализ. Прямоугольный сигнал', ...
    'HorizontalAlignment', 'center', ...
    'Visible', 'off', ...
    'FontSize', 12, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [X2/2-scsize(3)/13.5, 0.98*Y2 , 230, 20]);

Number = uicontrol('Style', 'Edit', ...
    'String', '1', ...
    'Visible', 'off', ...
    'Position', [0.86*X2, 0.52*Y2, 50, 20], ...
    'Callback', @Take_C);

NumberWords = uicontrol('Style', 'Text', ...
    'String', 'Номер гармоники n', ...
    'HorizontalAlignment', 'left', ...
    'Visible', 'off', ...
    'FontSize', 10, ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.91*X2, 0.52*Y2 , 150, 20]);

function Take_C (h, eventdata, handles)
    CC = str2num(get(Number, 'String'));
end

Sintez = uicontrol('Style', 'PushButton', ...
    'String', 'ПОСТРОИТЬ', ...
    'HorizontalAlignment', 'center', ...
    'Visible', 'off', ...
    'Position', [0.86*X2, 0.45*Y2 , 80, 40], ...
    'Callback', @Make_SintGr);

%=====

```

```

anbutt = uicontrol('Style', 'Text', ...
    'String', 'a(n)=', ...
    'Visible', 'off', ...
    'FontSize',12, ...
    'FontAngle', 'italic', ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.86*X2, 0.37*Y2, 45, 20]);

anbuttcontent = uicontrol('Style', 'Text', ...
    'String', '', ...
    'Visible', 'off', ...
    'HorizontalAlignment', 'left', ...
    'FontSize',12, ...
    'FontAngle', 'italic', ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.885*X2, 0.37*Y2, 100, 20]);
%-----

bnbutt = uicontrol('Style', 'Text', ...
    'String', 'b(n)=', ...
    'Visible', 'off', ...
    'FontSize',12, ...
    'FontAngle', 'italic', ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.86*X2, 0.33*Y2, 45, 20]);

bnbuttcontent = uicontrol('Style', 'Text', ...
    'String', '', ...
    'Visible', 'off', ...
    'HorizontalAlignment', 'left', ...
    'FontSize',12, ...
    'FontAngle', 'italic', ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.885*X2, 0.33*Y2, 40, 20]);
%-----

Anbutt = uicontrol('Style', 'Text', ...
    'String', 'A(n)=', ...
    'Visible', 'off', ...
    'HorizontalAlignment', 'left', ...
    'FontSize',12, ...
    'FontAngle', 'italic', ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.86*X2, 0.29*Y2, 45, 20]);

Anbuttcontent = uicontrol('Style', 'Text', ...
    'String', '', ...
    'Visible', 'off', ...
    'HorizontalAlignment', 'left', ...
    'FontSize',12, ...
    'FontAngle', 'italic', ...

```

```

    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.885*X2, 0.29*Y2, 40, 20]);
%-----

Qnbutt = uicontrol('Style', 'Text', ...
    'String', 'Q(n)=', ...
    'Visible', 'off', ...
    'HorizontalAlignment', 'left', ...
    'FontSize', 12, ...
    'FontAngle', 'italic', ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.86*X2, 0.25*Y2, 45, 20]);

Qnbuttcontent = uicontrol('Style', 'Text', ...
    'String', '', ...
    'Visible', 'off', ...
    'HorizontalAlignment', 'left', ...
    'FontSize', 12, ...
    'FontAngle', 'italic', ...
    'BackgroundColor', [0.819608 0.913725 0.862745], ...
    'Position', [0.885*X2, 0.25*Y2, 40, 20]);
%=====

function Make_SintGr(h, eventdata, handles)
    t = -T/2 : T/1000 : T/2;
    S = E/2;

    for d = 1: 1: CC
        S = S + A(d)*cos(2*pi*d*(1/T).*t + Q(d)/57.7);
    end

    axes(hA1);
    if (score == 1 || score==4)
        Sintplot = plot(t, u1, 'black',t, S, 'g');
        xlim([-T T]);
        title('Последовательность импульсов');
        ylabel('Амплитуда, В');
        xlabel('Время, мс');
        grid on;

    elseif score == 2
        Sintplot = plot(t, u2, 'black',t, S, 'g');
        xlim([-T T]);
        title('Последовательность импульсов');
        ylabel('Амплитуда, В');
        xlabel('Время, мс');
        grid on;

    elseif score == 3
        Sintplot = plot(t, u3, 'black',t, S, 'g');
        xlim([-T T]);
        title('Последовательность импульсов');

```

```

        ylabel('Амплитуда, В');
        xlabel('Время, мс');
        grid on;
    end

    if (score == 1 || score == 4)
        anValue = (2/N)*sum(u1.*cos(2*pi*CC.*t));
        bnValue = (2/N)*sum(u1.*sin(2*pi*CC.*t));
    elseif score == 2
        anValue = (2/N)*sum(u2.*cos(2*pi*CC.*t));
        bnValue = (2/N)*sum(u2.*sin(2*pi*CC.*t));

    elseif score == 3
        anValue = (2/N)*sum(u3.*cos(2*pi*CC.*t));
        bnValue = (2/N)*sum(u3.*sin(2*pi*CC.*t));
    end

    set(anbuttcontent, 'String', anValue);
    set(bnbuttcontent, 'String', bnValue);
    set(Anbuttcontent, 'String', A(CC));% или CC+1?
    set(Qnbuttcontent, 'String', Q(CC));% или CC+1?
    set(anbutt, 'Visible', 'on');
    set(anbuttcontent, 'Visible', 'on');
    set(bnbutt, 'Visible', 'on');
    set(bnbuttcontent, 'Visible', 'on');
    set(Anbutt, 'Visible', 'on');
    set(Anbuttcontent, 'Visible', 'on');
    set(Qnbutt, 'Visible', 'on');
    set(Qnbuttcontent, 'Visible', 'on');
end

```

```
function Analysis(h, eventdata, handles)
```

```
    if Availability == 1
```

```

        set(E_but, 'Enable', 'on');
        set(T_but, 'Enable', 'on');
        set(ti_but, 'Enable', 'on');
        set(Bre, 'Enable', 'on');
        set(Bme, 'Enable', 'on');
        set(Btr, 'Enable', 'on');
        set(Bsa, 'Enable', 'on');
        set(GenGr, 'Visible', 'on');
        set(Number, 'Visible', 'off');
        set(Sintez, 'Visible', 'off');
        set(Name_sig, 'Visible', 'on');
        set(Name_sigSint, 'Visible', 'off');
        set(NumberWords, 'Visible', 'off');

```

```
%-----
```

```

        set(anbutt, 'Visible', 'off');
        set(anbuttcontent, 'Visible', 'off');
        set(bnbutt, 'Visible', 'off');

```

```

set(bnbuttcontent, 'Visible', 'off');
set(Anbutt, 'Visible', 'off');
set(Anbuttcontent, 'Visible', 'off');
set(Qnbutt, 'Visible', 'off');
set(Qnbuttcontent, 'Visible', 'off');
%-----

if ((score == 1) || (score == 4))
    plot(t, u1, 'LineWidth', 2, 'Color', 'black');
    xlim([-T T]);
    title('Последовательность импульсов');
    ylabel('Амплитуда, В');
    xlabel('Время, мс');
    grid on;
elseif score == 2
    plot(t, u2, 'LineWidth', 2, 'Color', 'black');
    xlim([-T T]);
    title('Последовательность импульсов');
    ylabel('Амплитуда, В');
    xlabel('Время, мс');
    grid on;
elseif score == 3
    plot(t, u3, 'LineWidth', 2, 'Color', 'black');
    xlim([-T T]);
    title('Последовательность импульсов');
    ylabel('Амплитуда, В');
    xlabel('Время, мс');
    grid on;
end

else
    msgbox ('Чтобы перейти к пунктам меню, введите значения параметров сигнала и постройте графики!', ...
        'Warning', 'warn', 'Modal');
end

end

function Synthesis(h, eventdata, handles)

if Availability == 1

    set(E_but, 'Enable', 'off');
    set(T_but, 'Enable', 'off');
    set(ti_but, 'Enable', 'off');
    set(Bre, 'Enable', 'off');
    set(Bme, 'Enable', 'off');
    set(Btr, 'Enable', 'off');
    set(Bsa, 'Enable', 'off');
    set(GenGr, 'Visible', 'off');
    set(Number, 'Visible', 'on');
    set(Sintez, 'Visible', 'on');

```

```

        set(Name_sig, 'Visible', 'off');
        set(Name_sigSint, 'Visible', 'on');
        set(NumberWords, 'Visible', 'on');
%-----

        if score == 1
            set(Name_sigSint, 'String', 'Синтез.
Прямоугольный сигнал');
        elseif score == 2
            set(Name_sigSint, 'String', 'Синтез. Треугольный
сигнал');
        elseif score == 3
            set(Name_sigSint, 'String', 'Синтез.
Пилообразный сигнал');
        elseif score == 4
            set(Name_sigSint, 'String', 'Синтез. Меандр');
        end
    else
        msgbox ('Чтобы перейти к пунктам меню, введите
значения параметров сигнала и постройте графики!', ...
            'Warning', 'warn', 'Modal');
    end
end
end
end

```