



## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	7
ПОСТАНОВКА ЗАДАЧИ .....	8
1.1 Цель разработки, назначение и область применения .....	8
2. ТЕОРИТИЧЕСКАЯ ЧАСТЬ .....	9
2.1 Радиосистемы .....	9
2.2 Гетеродин в составе радиоприемника .....	10
2.3 Влияние характеристик гетеродина на характеристики системы .....	11
2.4 Аналого-цифровое преобразование .....	13
2.5 Выбор генератора частоты .....	14
2.5.2 Косвенный синтез частоты на основе фазовой автоподстройки (PLL) .....	16
2.5.3 Прямой цифровой синтез (DDS) .....	17
2.6 Критерии качества .....	20
2.7 Коэффициент шума .....	21
2.6.3 Динамический диапазон, свободный от паразитных составляющих .....	21
2.7.1 Дискретное преобразование Фурье .....	22
2.7.2 Утечка ДПФ .....	22
2.7.3 Оконные функции .....	23
2.8 Метод CORDIC .....	24
2.9 Табличный метод .....	24
3 ПРАКТИЧЕСКАЯ ЧАСТЬ .....	26
3.1 Выбор графического интерфейса .....	26
3.2 Создание графического интерфейса .....	26
3.2.1 Параметры .....	26
3.2.2. Тест .....	28
3.2.3 LUT vs LI .....	30
3.2.4 Точность .....	30
ЗАКЛЮЧЕНИЕ .....	31
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	32

ПРИЛОЖЕНИЕ А .....	34
ПРИЛОЖЕНИЕ Б .....	35
ПРИЛОЖЕНИЕ В .....	36
ПРИЛОЖЕНИЕ Г .....	37
ПРИЛОЖЕНИЕ Д .....	38
ПРИЛОЖЕНИЕ Е .....	39
ПРИЛОЖЕНИЕ Ж .....	40
ПРИЛОЖЕНИЕ И .....	41
ПРИЛОЖЕНИЕ К .....	44
ПРИЛОЖЕНИЕ Л .....	45
ПРИЛОЖЕНИЕ М .....	46

## ВВЕДЕНИЕ

АО "ГРЦ Макеева" известно в России как стратегический холдинг оборонно-промышленного комплекса страны и ракетно-космической отрасли, головной разработчик жидкостных и твердотопливных ракетных комплексов стратегического назначения с баллистическими ракетами, один из крупнейших научно-конструкторских центров России по разработке ракетно-космической техники. К разрабатываемой аппаратуре предъявляются высокие требования по надежности, условиям эксплуатации, сроку службы.

Одной из задач является разработка малогабаритной радиоприемной аппаратуры, позволяющей получать телеметрическую информацию от объекта испытаний по радиоканалу.

Целью дипломной работы является выбор и оптимизация методов синтеза однотонного цифрового сигнала.

В настоящей дипломной работе разработано программное обеспечение, позволяющее подбирать оптимальный метод генерации синусоидального сигнала с последующим переносом его в микроконтроллер.

					27.03.04.2018.293 ПЗ	Лист
Изм						7
	Лист	№ докум.	Подпись	Дата		

## ПОСТАНОВКА ЗАДАЧИ

Цель разработки, назначение и область применения

Цель выполнения работы – программная реализация устройства, позволяющая сравнивать методы генерации синусоидального сигнала с заданием параметров сигнала и параметров микроконтроллера.

Устройство предназначено для облегчения работы разработчика программного обеспечения, используемого в радиоприемной аппаратуре.

Область применения – работа в процессе разработки модели гетеродина в составе радиоприемного оборудования.

Основные требования к разрабатываемому устройству

Требования к разрабатываемому устройству приведены в техническом задании на разработку ВКР по теме "Цифровой генератор гармонического сигнала", утвержденном начальником отдела 121 АО "ГРЦ Макеева".

Выводы по разделу 1

В разделе 1 определены цель выполнения работы, назначение и область применения, основные требования к устройству.

					27.03.04.2018.293 ПЗ	Лист
Изм						8
	Лист	№ докум.	Подпись	Дата		

## 2. ТЕОРИТИЧЕСКАЯ ЧАСТЬ

### 2.1 Радиосистемы

В соответствии с общепринятой терминологией под радиосвязью понимают связь, осуществляемую с помощью радиоволн. При этом к радиоволнам относят электромагнитные волны, распространяющиеся в пространстве без искусственного волновода, частоты которых условно ограничены 3000 Гц.

Согласно «Регламенту радиосвязи», любые системы связи подразделяются на наземные и космические. Космической считается «...любая связь, при которой используется одна или несколько космических станций или один, или несколько отражающих спутников, или другие космические объекты». Все остальные системы радиосвязи относятся к наземным.

Следует отметить, что в регламенте радиосвязи понятие «связь» трактуется широко, так что и радионавигационные, и радиолокационные системы оказываются системами связи. Регламент радиосвязи ограничивается классификацией систем, основанной только на их назначении. Для целей стандартизации требований системам часто используется дополнительная классификация, учитывающая базовые технические принципы их построения. Именно так выделяется класс систем цифровой радиосвязи. К характерной особенности систем этого класса относится то, что информация в них передается в цифровой форме.

Цифровое радиовещание либо обеспечивает более высокое качество вещания в диапазонах длинных, средних и коротких волн, в той же полосе частот, прежде занимаемой системой на основе амплитудной модуляции, либо повышает эффективность использования радиочастотного спектра на частотах выше 30 МГц.

					27.03.04.2018.293 ПЗ	Лист
Изм						9
	Лист	№ докум.	Подпись	Дата		

## 2.2 Гетеродин в составе радиоприемника

В настоящее время большую распространенность получили программно-определяемые радиосистемы. Это радиопередатчик или радиоприемник, в основе которого лежит технология, позволяющая устанавливать или изменять рабочие радиочастотные параметры, включая, в частности тип модуляции, диапазон частот или выходную мощность с помощью программного обеспечения.

Блок-схема программно-определяемой схемы представлена на рисунке 1.

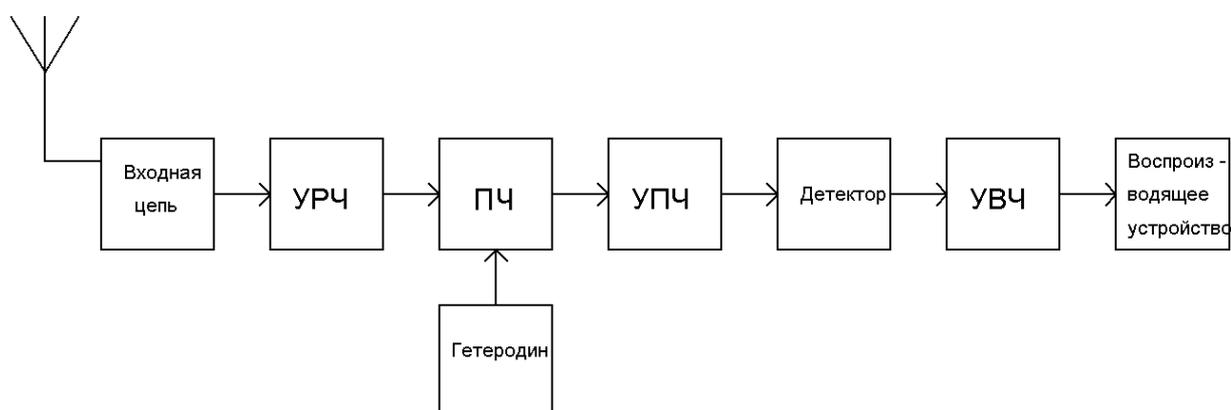


Рис. 1. Блок-схема радиоприемника

Одним из ключевых компонентов радиоприемника является гетеродин — маломощный генератор электрических колебаний, применяемый для преобразования частот сигнала, создающий колебания вспомогательной частоты, которые в блоке смесителя смешиваются с поступающими извне колебаниями высокой частоты. В результате смешения двух частот, входной и гетеродина, образуются ещё две частоты (суммарная и разностная). Разностная частота используется как промежуточная частота, на которой происходит основное усиление сигнала.

## 2.3 Влияние характеристик гетеродина на характеристики системы

От параметров гетеродина зависит качество преобразованного сигнала и в общем случае параметры всего радиоприемного устройства в целом.

Основные параметры гетеродина:

- Частота или диапазон частот выходного сигнала
- Шаг перестройки частоты гетеродина
- Мощность выходного сигнала
- Относительная нестабильность частоты:
  - кратковременная
  - долговременная
- Точность установки частоты
- Относительный уровень гармоник на выходе
- Спектральная плотность мощности фазовых шумов

Рабочая частота или диапазон частот определяются, исходя из полосы частот принимаемого сигнала. Например, для стандарта сотовой связи GSM 900 диапазон принимаемых частот для мобильных аппаратов составляет 935 ... 960 МГц.

Шаг перестройки гетеродина задается шириной радиоканала, принятой в системе связи, в составе которой будет работать приемник. Например, шаг перестройки гетеродина для стандарта сотовой связи GSM будет составлять 200 кГц.

Мощность на выходе гетеродина (в дБм) определяется, исходя из требуемой для работы смесителя мощности плюс запас в 2 ... 4 дБ, необходимый на компенсацию потерь в тракте гетеродин-смеситель и на рассогласование по входу смесителя. Обычно уровень сигнала гетеродина составляет +16 дБм.

Относительная нестабильность частоты — характеристика, показывающая отклонение частоты генератора (уход частоты) от номинального значения. Различают кратковременную (например, за 1 секунду) и долговременную (например, за 1 год) нестабильность частоты.

Точность установки частоты задается резонансной частотой резонатора в цепи обратной связи генератора. В случае применения в качестве гетеродина синтезатора частот, точность установки частоты гетеродина определяется частотой опорного генератора. Обычно это значение составляет 10 ... 1000 Гц.

Относительный уровень гармоник на выходе зависит от схемы генератора и наличия фильтра на его выходе. Обычно на выходе гетеродина требуется, чтобы уровень гармоник не превышал -25 дБ.

Спектральная плотность мощности фазовых шумов характеризует кратковременную фазовую нестабильность частоты гетеродина за счет шумовых свойств генератора.

Наиболее важным параметром гетеродина, влияющим на всю систему, является фазовый шум. Он ограничивает максимальное отношение сигнал/шум, которое может быть достигнуто в приемнике частотно-модулированных сигналов, увеличивает вероятность ошибочного приема бита информации в цифровых системах передачи информации с фазовой манипуляцией. Если сильные сигналы находятся вблизи частоты настройки приемника, то они могут вызывать блокирование смесителя радиоприемного устройства, выражающееся в переносе шумов гетеродина в полосу пропускания приемника и увеличении его собственного шума. Это ограничивает способность приемника принимать слабые сигналы.

## 2.4 Аналого-цифровое преобразование

Аналого-цифровое преобразование заключается в преобразовании информации, содержащейся в аналоговом сигнале, в дискретный (цифровой) код. Иными словами, это процесс преобразования входной физической величины в ее числовое представление. Аналого-цифровой преобразователь – устройство, выполняющее такое преобразование.

Понятие аналого-цифрового преобразования тесно связано с понятием измерения. Под измерением понимается процесс сравнения измеряемой величины с некоторым эталоном, при аналого-цифровом преобразовании происходит сравнение входной величины с некоторой опорной величиной (как правило, с опорным напряжением). Таким образом, аналого-цифровое преобразование может рассматриваться как измерение значения входного сигнала, и к нему применимы все понятия метрологии, такие, как погрешности измерения.

АЦП имеет множество характеристик, из которых основными можно назвать частоту преобразования и разрядность. Частота преобразования обычно выражается в отсчетах в секунду (samples per second, SPS), разрядность – в битах. Современные АЦП могут иметь разрядность до 24 бит и скорость преобразования до единиц GSPS (конечно, не одновременно). Чем выше скорость и разрядность, тем труднее получить требуемые характеристики, тем дороже и сложнее преобразователь. Скорость преобразования и разрядность связаны друг с другом определенным образом, и мы можем повысить эффективную разрядность преобразования, пожертвовав скоростью.

Структурная схема АЦП приведена на рисунке 2:

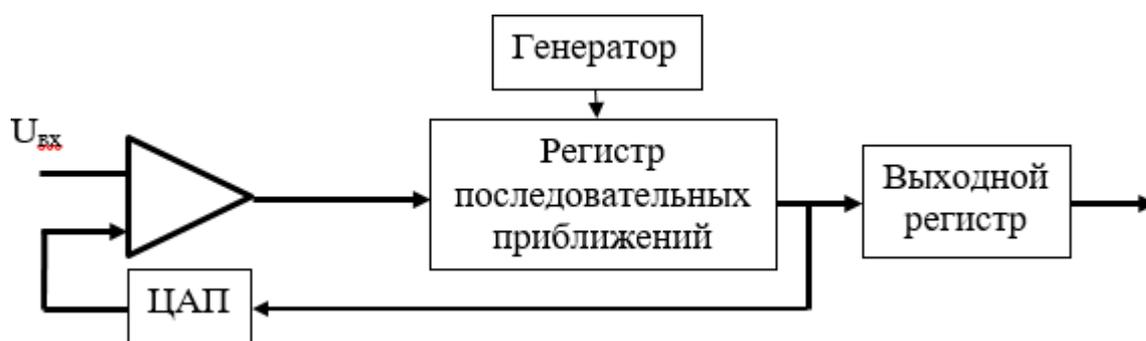


Рис. 2. Структурная схема АЦП

## Выбор генератора частоты

Одним из компонентов является генератор частоты, электронное устройство, позволяющее из опорной частоты получать на выходе требуемую частоту или набор частот, согласно управляющим сигналам. Наибольшее распространение получили следующие методы синтеза частот:

- прямой аналоговый синтез (Direct Analog Synthesis, DAS) на основе структуры смеситель/фильтр/делитель, когда выходная частота получается непосредственно из опорной частоты посредством операций смешения, фильтрации, умножения и деления

- косвенный (indirect) синтез на основе фазовой подстройки частоты (Phase Locked Loop, PLL), когда выходная частота получается с помощью дополнительного генератора (Voltage Controlled Oscillator, VCO), который охвачен петлей фазовой автоподстройки

- прямой цифровой синтез (Direct Digital Synthesis, или DDS), когда выходной сигнал синтезируется цифровыми методами

- гибридный синтез – совместное использование вышеперечисленных методов

Основными параметрами, характеризующими качество синтезатора частоты, являются следующие:

- чистота спектра выходного сигнала (уровень побочных компонентов и уровень шума)
- диапазон перестройки (полоса частот выходного сигнала)
- скорость перестройки
- частотное разрешение
- количество разных генерируемых частот
- гибкость (возможность осуществления различных видов модуляции)
- неразрывность фазы выходного сигнала при перестройке

## 2.5.1 Прямой аналоговый синтез

Структурная схема прямого аналогового синтезатора показана на рис. 3. Этот метод называют прямым по той причине, что отсутствует процесс коррекции ошибки, следовательно, качество выходного сигнала напрямую связано с качеством опорного сигнала. Фазовый шум этого синтезатора достаточно низкий вследствие прямого синтеза. Перестройка по частоте также может быть очень быстрой. Одной из важных особенностей такого синтезатора на основе смесителя/фильтра является возможность вернуться на любую частоту и продолжать работать в той фазе, как и в том случае, если бы перехода вообще не было. Этот эффект называют «фазовой памятью». Для перестройки по частоте используется переключаемый банк опорных генераторов. Это подходит, например, для радиостанций с небольшим количеством каналов. Но для обеспечения широких возможностей по перестройке частоты требуется очень много опорных генераторов, что является весьма дорогостоящим решением. Используя делители частоты (структура смеситель/фильтр/делитель) можно уменьшить количество необходимых опорных генераторов, хотя и в этом случае возможности по перестройке останутся более чем скромными.

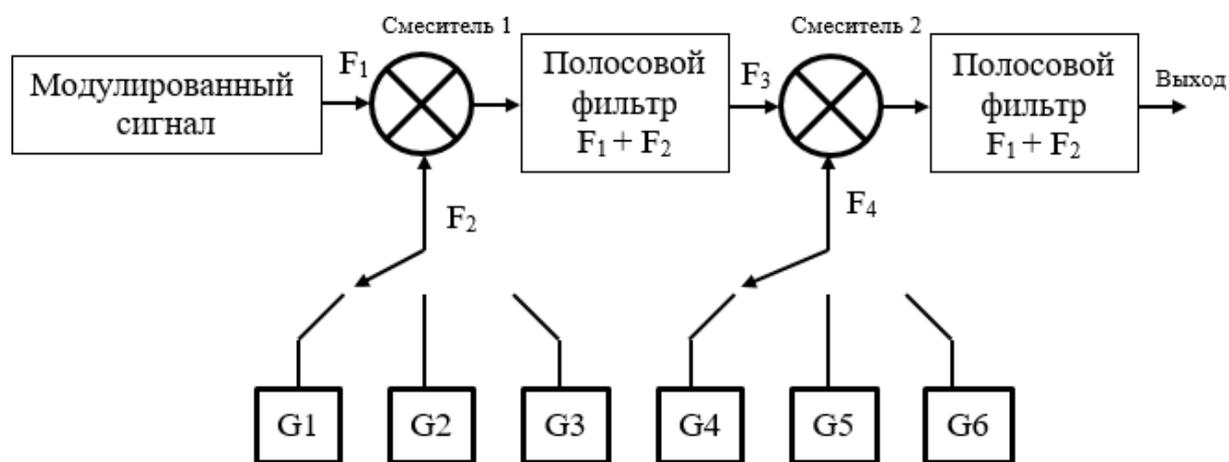


Рис. 3. Структурная схема DAS

## 2.5.2 Косвенный синтез частоты на основе фазовой автоподстройки (PLL)

Этот метод синтеза использует принцип сравнения частоты и фазы выходного сигнала, источником которого служит генератор, управляемый напряжением (VCO), с сигналом опорного генератора. Структурная схема такого синтезатора показана на рисунке 4.

Обнаружение ошибки осуществляется с помощью фазового детектора, который работает на определенной частоте  $F_C$ , называемой частотой сравнения. Эта частота получается путем деления на  $N$  частоты опорного генератора  $G$ . Частота выходного сигнала вначале делится на  $M$ , а потом сравнивается с частотой  $F_C$ . Если частота отклоняется, обнаруженная ошибка вызывает изменение управляющего напряжения VCO, что приводит к уменьшению отклонения.

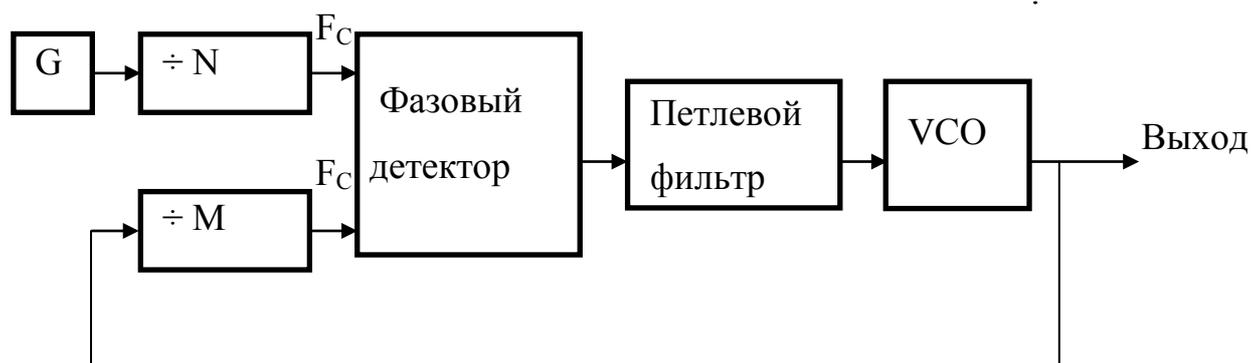


Рис. 4. Синтезатор частоты на основе PLL

Выходная частота определяется по формуле:

$$F_{\text{OUT}} = F_C \cdot M = (F_{\text{CLK}}/N) \cdot M = F_{\text{CLK}} \cdot (N/M),$$

где:

$F_{\text{OUT}}$  – выходная частота,

$F_C$  – частота сравнения,

$M$  – коэффициент деления выходной частоты.

$N$  – коэффициент деления опорной частоты,

Другими словами, PLL синтезатор умножает опорную частоту в  $N/M$  раз.

Коэффициенты  $N$  и  $M$  могут задаваться микроконтроллером, хотя на практике число  $N$  при перестройке меняют редко, так как это влечет за собой изменение частоты сравнения (соответственно, и шага сетки) и требует изменения параметров петлевого фильтра.

Фазовый детектор является источником дополнительных фазовых шумов. Попытки получить малый шаг перестройки частоты вынуждают работать на более низкой частоте сравнения, что требует понижения частоты среза петлевого фильтра. А это еще более увеличивает фазовые шумы. Быстрое переключение частоты в этой структуре также обеспечить очень сложно.

Для получения малого шага перестройки по частоте иногда объединяют в одном синтезаторе несколько петель PLL. Однако многопетлевой PLL синтезатор является весьма дорогим и громоздким устройством, что сдерживает его широкое применение.

### 2.5.3 Прямой цифровой синтез (DDS)

DDS-устройства (генераторы, синтезаторы) уникальны своей цифровой определенностью: генерируемый ими сигнал синтезируется со свойственной цифровым системам точностью. Частота, амплитуда и фаза сигнала в любой момент времени точно известны и подконтрольны. DDS практически не подвержены температурному дрейфу и старению. Единственным элементом, который обладает свойственными аналоговым схемам нестабильностями, является ЦАП. Все это является причиной того, что в последнее время DDS вытесняют обычные аналоговые синтезаторы частот.

Основные преимущества DDS:

- очень высокое разрешение по частоте и фазе
- цифровое управление частотой и фазой выходного сигнала
- экстремально быстрый переход на другую частоту (или фазу), перестройка по частоте без разрыва фазы, без выбросов и других аномалий, связанных с временем установления



- максимальная выходная частота не может быть выше половины тактовой (на практике она еще меньше). Это ограничивает области применения DDS на HF и часть VHF диапазона

- потребляемая DDS мощность практически прямо пропорциональна тактовой частоте и может достигать сотен милливольт. При больших тактовых частотах DDS могут оказаться непригодными для устройств с батарейным питанием

- отдельные побочные компоненты на выходе DDS могут быть значительно большими, чем у других видов синтеза. Спектральная чистота выходного сигнала DDS сильно зависит от качества ЦАП

Структурная схема прямого цифрового синтезатора приведена на рисунке 4:

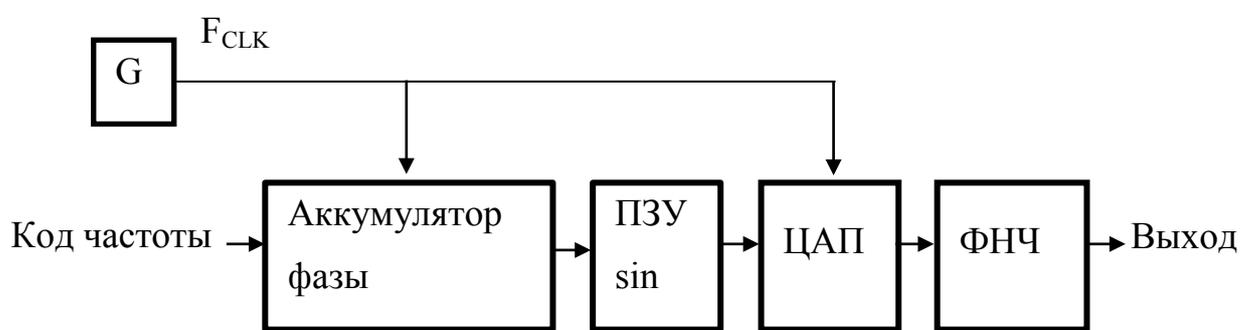


Рис. 4 Структурная схема DDS

## 2.6 Критерии качества

### 2.6.1 Отношение сигнал/шум

Шумовые характеристики – это ключевые параметры для любого радиоприемника. Отношение сигнал/шум (ОСШ, SNR) – это характеристика чувствительности приемника. Любые шумы, поступающие на вход усилителя, будут суммироваться с сигналом и усиливаться в последующих усилительных каскадах радиоприемника.

Отношение сигнал/шум – это отношение между полезным сигналом и нежелательным шумом.

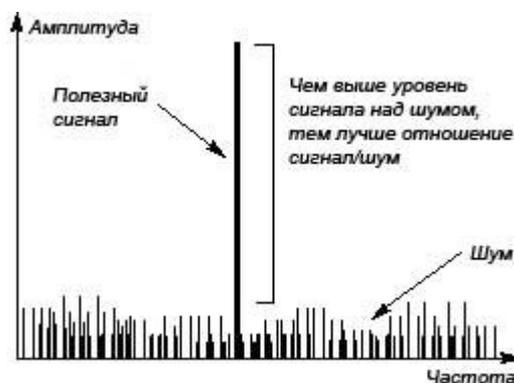


Рис. 5. Графическое изображение ОСШ

Формула отношения сигнал/шум:

$$\text{SNR} = \frac{P_{\text{сигнала}}}{P_{\text{шума}}}$$

Формула отношения сигнал/шум, выраженная в логарифмических единицах с использованием децибел:

$$\text{ОСШ}_{\text{дБ}} = 10 \lg_{10} \frac{P_{\text{сигнала}}}{P_{\text{шума}}}$$

Факторы, влияющие на отношение сигнал/шум:

- Полоса пропускания. Чем шире полоса пропускания, тем выше уровень шума.
- Использование амплитудной модуляции. Чем выше уровень модуляции, тем выше сигнал на выходе радиоприемника.

### Коэффициент шума

Другим, не менее важным показателем качества радиоприемника является коэффициент шума. Он показывает насколько собственные шумы приемника влияют на усиливаемый полезный сигнал. Его определение включает в себя отношение сигнал/шум:

$$K_{ш} = \frac{SNR_i}{SNR_o},$$

где  $SNR_i$  – отношение сигнал/шум входного сигнала, а  $SNR_o$  – выходного сигнала.  $F$  – коэффициент, характеризующий ухудшение усилителем отношения сигнал-шум входного сигнала.

Динамический диапазон, свободный от паразитных составляющих (Spurious-Free Dynamic Range, SFDR) - безразмерная величина, равная отношению мощности полезного узкополосного сигнала (несущей) к мощности наиболее мощной паразитной частотной составляющей (гармоники).

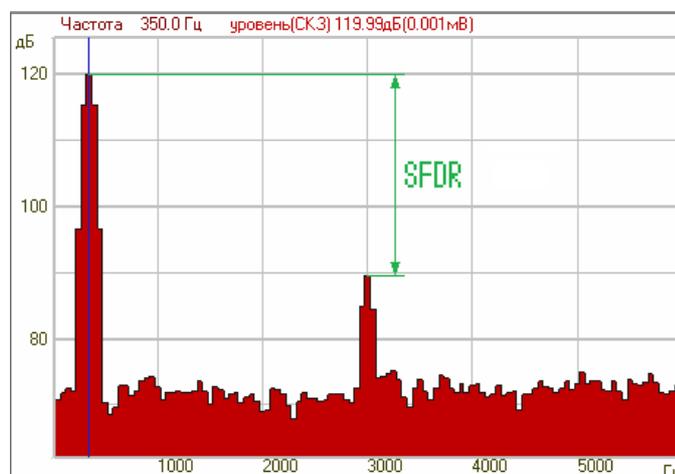


Рис. 6. Измерение SFDR при помощи спектра

## Методы измерения характеристик

### Дискретное преобразование Фурье

Дискретное преобразование Фурье (ДПФ) – одна из двух наиболее распространенных и мощных процедур цифровой обработки сигналов. ДПФ позволяет анализировать, преобразовывать и синтезировать сигналы такими способами, которые невозможны при аналоговой обработке.

ДПФ – это математическая процедура, используемая для определения гармонического состава дискретных сигналов. Истоком ее является непрерывное преобразование Фурье, которое определяется, как

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt,$$

где  $x(t)$  – непрерывный сигнал во временной области.

Преобразование Фурье – наиболее значительный и широко распространенный математический механизм для анализа физических систем. С приходом цифровой обработки сигналов, ДПФ преобразовалось в дискретную последовательность  $X(m)$  в частотной области

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi m n / N},$$

где  $x(n)$  – последовательность значений, полученных дискретизацией во временной области непрерывной переменной  $x(t)$ .

### Утечка ДПФ

ДПФ выдает конкретные результаты лишь тогда, когда входные последовательности  $x(n)$  представляют собой специально подобранные синусоиды. Такая особенность ДПФ, как утечка, приводит к тому, что результаты ДПФ представляют собой только аппроксимацию истинного спектра исходного непрерывного сигнала.

Утечка - это эффект перераспределения энергии в измеренном спектре от одной частотной составляющей к другим. Утечка возникает, если в выборке дискретизированного сигнала, полученной за время сбора данных, не укладывается целое число периодов сигнала. Невозможно полностью устранить утечку, но существуют способы по ее минимизации.

### 2.7.3 Оконные функции

За счет уменьшения уровня боковых лепестков функции, взвешивание окон уменьшает утечку. Это достигается путем постепенного приведения отсчетов в начале и конце последовательности к общему значению. Если мы рассмотрим сигнал бесконечной длительности во временной области, ДПФ можно выполнить только над интервалом конечной длительности (рисунок 7 а, с).

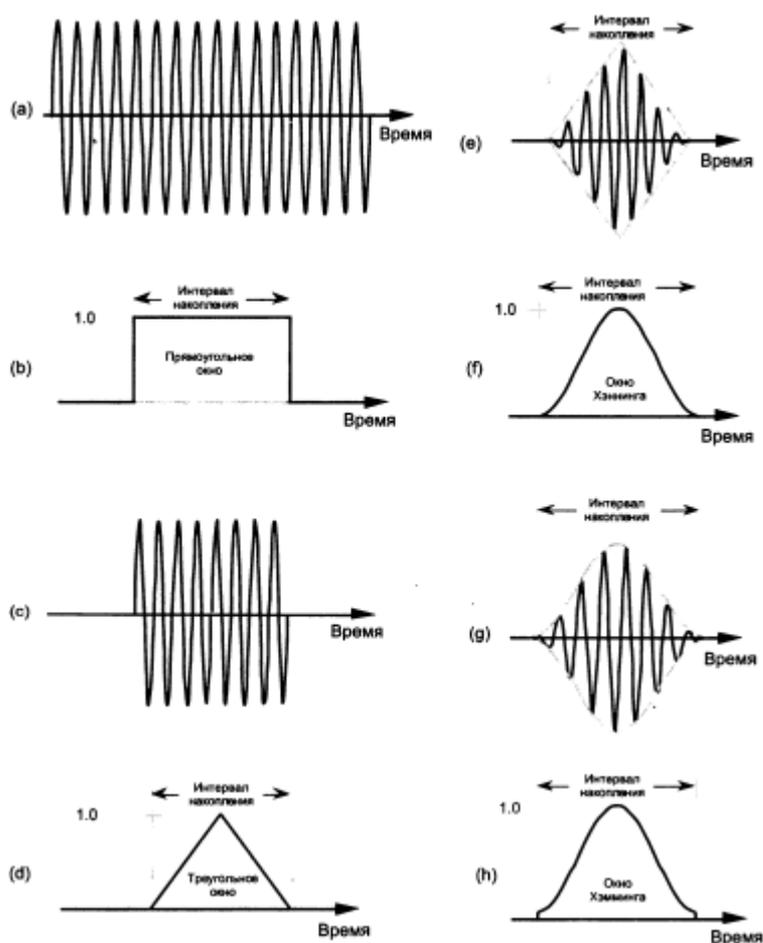


Рис. 7. Окно Хэмминга

## Метод CORDIC

CORDIC (от англ. Coordinate Rotation Digital Computer — цифровой вычислитель поворота системы координат) — итерационный метод сведения прямых вычислений сложных функций к выполнению простых операций сложения и сдвига.

Идея метода заключается в сведении вычисления значений сложных (например, гиперболических) функций к набору простых шагов — сложению и сдвигу. Такой подход особенно полезен при вычислении функций на устройствах с ограниченными вычислительными возможностями, такими как микроконтроллеры или программируемые логические матрицы (FPGA). Кроме того, поскольку шаги однотипны, то при аппаратной реализации алгоритм поддается развёртыванию в конвейер либо свертыванию в цикл.

## Табличный метод

Одна из задач цифровой обработки сигналов - генерация сигналов заданной частоты и формы. Основные методы цифровой генерации сигналов: метод аппроксимации и табличный метод.

Метод аппроксимации использует представление периодического сигнала в аналитическом виде и выполнение в микропроцессоре вычислений по программе. Параметры генерируемого сигнала хранятся в ячейках ОЗУ микропроцессорного устройства (или в ПЗУ, если они не изменяются). Программа должна обеспечивать вычисление отсчетов функции с таким интервалом, чтобы число отсчетов за период сигнала составляло не менее десятка.

В заключение цикла выполнения программы происходит вывод полученного цифрового отсчета сигнала. При необходимости получения аналогового сигнала устройство вывода содержит цифро-аналоговый преобразователь. Цифровые отсчеты сигнала могут использоваться и внутри МП-устройства, например, в другой программе.

					27.03.04.2018.293 ПЗ	Лист
Изм						24
	Лист	№ докум.	Подпись	Дата		

Метод аппроксимации позволяет обходиться небольшим объемом ЗУ, а к его недостаткам относится значительное число арифметических и логических операций, затрачиваемых на формирование одного отсчета сигнала, что снижает максимальную частоту генерируемых сигналов.

Табличный метод генерации предполагает, что предварительно вычисленные цифровые отсчеты функции хранятся в постоянном ЗУ. Формирование очередного отсчета сводится к операциям подготовки адреса и чтения соответствующей ячейки ПЗУ.

Достоинством табличного метода является меньшее время, затрачиваемое на формирование отсчета и, как следствие, возможность генерации сигналов с более высокой частотой. Недостатком метода является необходимость иметь ПЗУ большого объема.

### 3 ПРАКТИЧЕСКАЯ ЧАСТЬ

#### Выбор графического интерфейса

Согласно ТЗ, программа должна быть написана в среде MATLAB. В связи с этим, встал вопрос выбора между двумя графическими интерфейсами, GUIDE и AppDesigner. Образцы окон каждого из приложений приведены в приложениях А и Б. После сравнения данных графических интерфейсов был выбран AppDesigner, поскольку он предоставляет больше возможностей, поддерживает множество версий MATLAB, размещение элементов и работа с ними значительно облегчена в сравнении с GUIDE, присутствуют такие элементы, как List Box (выбор значения из списка), Spinner (поле для ввода чисел с их пошаговым изменением), есть возможность отрисовки графиков внутри окна программы.

#### Создание графического интерфейса

После ознакомления со инструментом «AppDesigner» для более удобной работы было решено использовать в графическом интерфейсе систему вкладок.

##### 3.2.1 Параметры

Первая вкладка «Параметры» содержит две панели и кнопку закрытия приложения. Скриншот окна приведен в приложении В. На первой панели задаются основные параметры, необходимые для генерации сигнала:

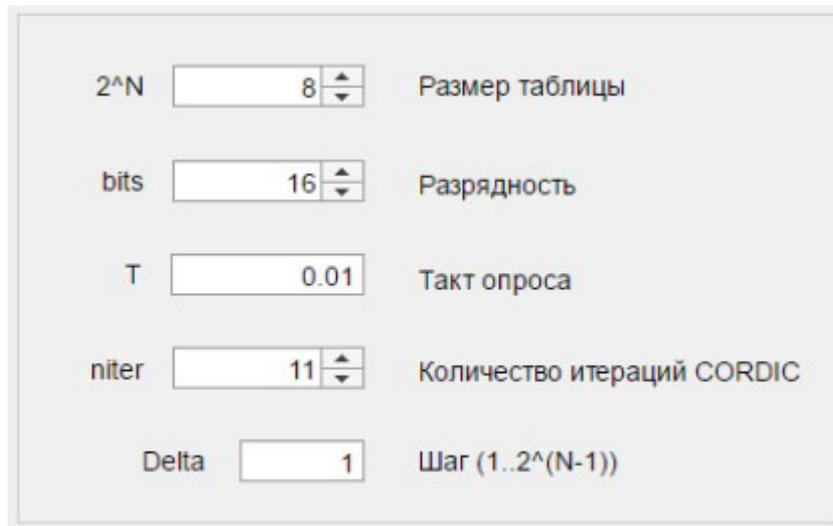


Рис. 8. Панель параметров

«Размер таблицы» - это значение, являющееся степенью двойки, в пределах от  $2^8$  до  $2^{16}$  в таблице, используемой для табличного метода генерации, определяет память процессора.

«Разрядность» - разрядность, необходимая для арифметики с фикс. точкой, для которой выполняется генерация кода (разрядность процессора).

«Такт опроса» задает быстродействие.

«Количество итераций CORDIC» - количество тактов выполнения алгоритма CORDIC.

«Шаг» - изменение частоты синусоиды (шаг по таблице).

На второй панели расположена кнопка «Перенос методов в HDL», осуществляющую генерацию HDL кода для выбранного метода синтеза гармонического сигнала.

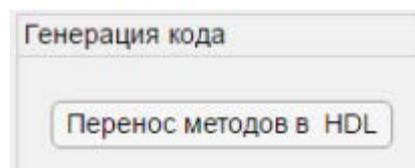


Рис. 9. Панель «Генерация кода»

### 3.2.2. Тест

На вкладке «Тест» расположены основные органы управления программы, а также содержит две панели: «Результаты» и «Пакетный режим». Скриншот окна приведен в приложении Г.

Выпадающий список «Выбор метода» содержит 5 основных методов синтеза синусоидального сигнала:

- «Direct» - табличный метод
- «Linear» - табличный метод с линейной интерполяцией
- «FixPtLinear» - табличный метод с линейной интерполяцией в арифметике с фиксированной точкой
- «Reccurent» - рекуррентный метод
- «CORDIC» - итерационный метод

Флаговая кнопка «Шум на выходе преобразователя» позволяет подмешивать шум к выходному сигналу выбранного метода генерации синусоидального сигнала.

Флаговая кнопка «Шум на входе» позволяет подмешивать шум к выходному сигналу аккумулятора фазы.

Слайдер «Уровень шума» изменяет уровень подмешиваемого шума.

Кнопка «Spectrum» запускает алгоритм расчета коэффициента нарушения гармоничности, значение которого отображается на вкладке «Результаты», также в окне программы строится спектр помех. Скриншот окна после нажатия кнопки приведен в приложении Д.

Кнопка «Test SFDR (Matlab, double)» запускает встроенный в пакет MATLAB алгоритм расчёта динамического диапазона, свободного от паразитных составляющих в арифметике с плавающей точкой, полученное значение отображается на панели «Результаты», построение графика SFDR осуществляется в отдельном окне. Скриншот окна после нажатия кнопки приведен в приложении Е.

Кнопка «Test SFDR (Simulink)» позволяет рассчитать SFDR в среде «Simulink» с помощью тонкой настройки, при этом значения задаются на вкладке «Параметры», построение графика SFDR осуществляется в отдельном окне, полученное значение отображается на панели «Результаты». Скриншот окна после нажатия кнопки приведен в приложении Ж.

Кнопка «Simulink» запускает схему в среде «Simulink», в которой строятся спектры сигналов, генерируемые табличным методом, итерационным методом и табличным методом с линейной интерполяцией. В этом случае значения задаются не во вкладке «Параметры», а непосредственно в среде «Simulink»: при нажатии на подсистему открывается окно, в котором пользователь вводит необходимые значения. Построение спектров сигналов происходит в отдельных окнах, имеющих соответствующие обозначения. Скриншот окна после нажатия кнопки приведен в приложении И.



Рис. 10. Основные органы управления

Панель «Пакетный режим» содержит кнопку «SFDR/noise (FixPt)», которая запускает алгоритм вычисления зависимости уровня SFDR от шума в арифметике с фиксированной точкой. Построение графика зависимости SFDR от уровня шума производится в окне программы. Данный алгоритм требователен к характеристикам компьютера, следовательно, имеет низкую скорость выполнения, поэтому был введен индикатор готовности программы - слайдер «Выполнение». Скриншот окна после нажатия кнопки «SFDR/noise (FixPt)» приведен в приложении К.

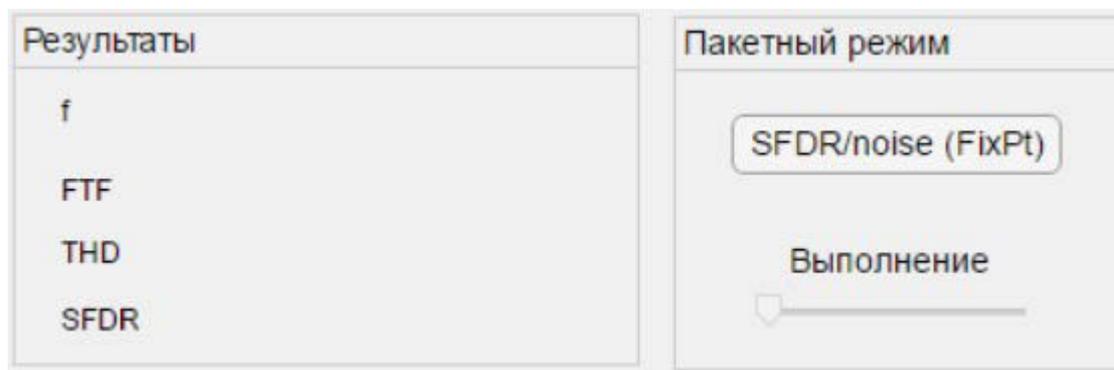


Рис. 11. Панели «Результаты» и «Пакетный режим»

## LUT vs LI

На вкладке «LUT vs LI» показан коэффициент нелинейных искажений для синуса, сгенерированного табличным методом с применением линейной интерполяции и без ее применения. Для построения используется кнопка «График». Так как алгоритм построения графика также требователен к ресурсам ПК, был введен индикатор готовности программы - слайдер «Выполнение». Скриншоты окна до и после нажатия кнопки приведены в приложении Л.

## Точность

На вкладке «Точность» показана ошибка по амплитуде (разница между истинным и сгенерированным сигналом) для каждого из методов. Для построения графика необходимо выбрать соответствующий метод из выпадающего списка и нажать кнопку «Double» (расчет в арифметике с плавающей точкой) или «Fixed» (расчет в арифметике с фиксированной точкой). Скриншоты окна при различных значениях приведены в приложении Е.

## ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе были рассмотрены и про-  
моделированы различные способы генерации синусоидальных сигналов, а также  
разработан графический интерфейс пользователя для упрощения выбора метода  
генерации.

					27.03.04.2018.293 ПЗ	Лист
Изм						31
	Лист	№ докум.	Подпись	Дата		

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Lyons, R. Streamlining Digital Signal Processing / R. Lyons. - Hoboken, New Jersey: John Wiley & Sons, Inc, 2007. – 338 с.
2. Кестер, У. Проектирование систем цифровой и смешанной обработки сигналов / У. Кестер. – Москва: Техносфера, 2010. – 330 с.
3. Дайнеко Д. Реализация CORDIC-алгоритма на ПЛИС / Д. Дайнеко // Компоненты и технологии. – 2011. №12. – С. 36-46.
4. Miao J. Signal processing in digital communications / George J. Miao. – Norwood, MA: ARTECH HOUSE, INC, 2007. – 533 с.
5. Ридико Л. DDS: прямой цифровой синтез частоты / Л. Ридико // Компоненты и технологии. – 2001. №7. – С.11-15.
6. Chrysafis A. Digital Sine-Wave Synthesis Using the DSP56001/2 // Motorola Digital Signal Processors. – 2006. Enclosure 1. – С.1-44.

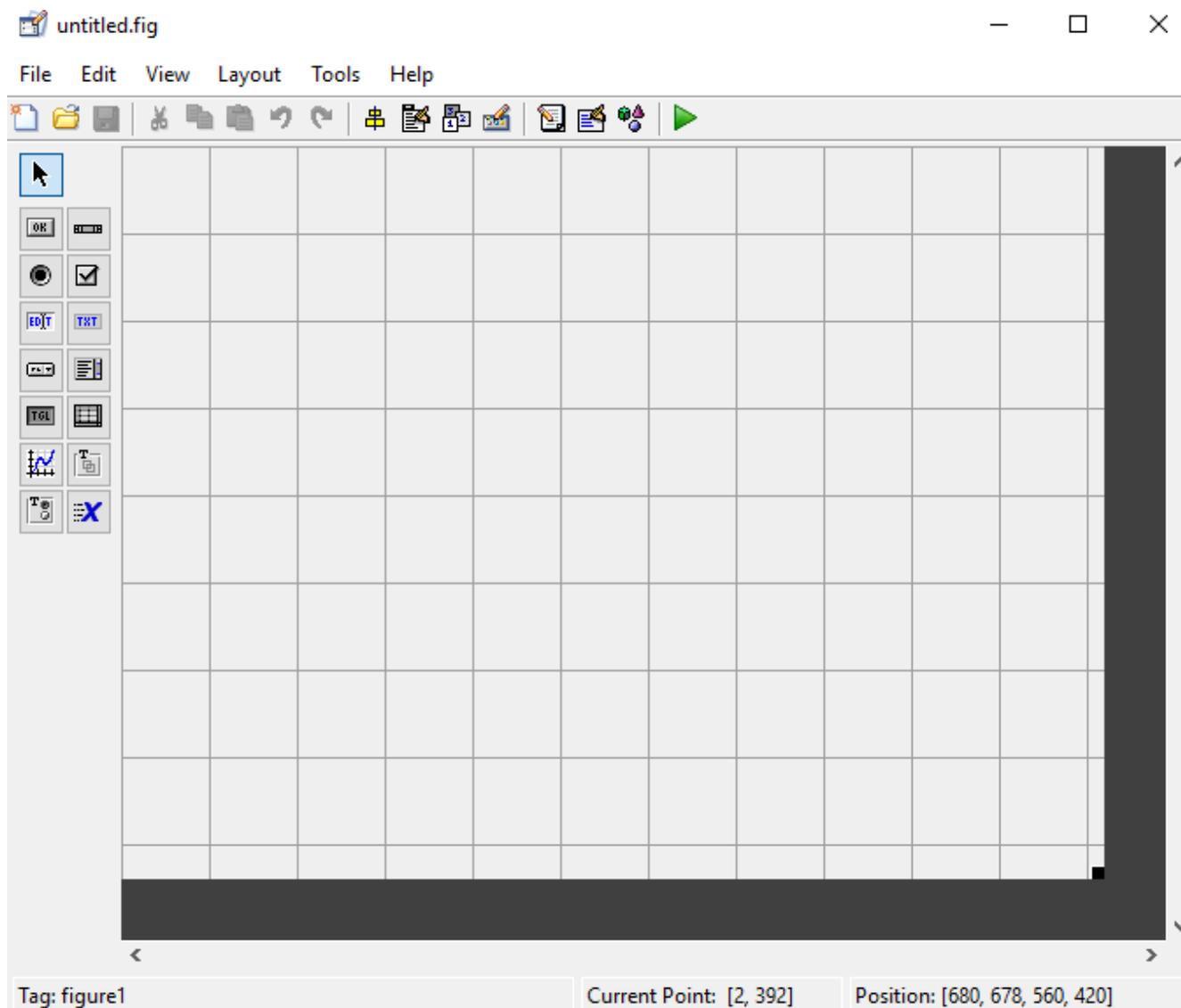
Ресурсы интернет:

7. Методы CORDIC - <https://ru.wikipedia.org/wiki/CORDIC>
8. Коэффициент гармонических искажений - [http://ru.electrical-installation.org/ru-wiki/Суммарный\\_коэффициент\\_гармонических\\_искажений\\_\(THD\)](http://ru.electrical-installation.org/ru-wiki/Суммарный_коэффициент_гармонических_искажений_(THD))
9. Direct Digital Syntesis - <http://ra3ggi.qrz.ru/UZLY/dds.htm>
10. Creating System in Simulink – [https://www.mathworks.com/help/simulink/guide/define-system.html?s\\_cid=learn\\_doc](https://www.mathworks.com/help/simulink/guide/define-system.html?s_cid=learn_doc)
11. AppDesigner – [https://www.mathworks.com/help/matlab/ref/appdesigner.html?s\\_tid=doc\\_ta](https://www.mathworks.com/help/matlab/ref/appdesigner.html?s_tid=doc_ta)
12. GUIDE – [https://www.mathworks.com/help/matlab/ref/guide.html?searchHighlight=guide&s\\_tid=doc\\_srchtile](https://www.mathworks.com/help/matlab/ref/guide.html?searchHighlight=guide&s_tid=doc_srchtile)
13. Compute total harmonic distortion (THD) of signal – [https://www.mathworks.com/help/physmod/sps/powersys/ref/thd.html?searchHighlight=thd&s\\_tid=doc\\_srchtile](https://www.mathworks.com/help/physmod/sps/powersys/ref/thd.html?searchHighlight=thd&s_tid=doc_srchtile)

14. Generate sinewave using CORDIC -  
<https://www.mathworks.com/help/fixedpoint/ref/cordicsin.html>
15. Digital waveform generation approximating a sine wave –  
[https://www.mathworks.com/help/simulink/examples/digital-waveform-generation-approximating-a-sine-wave.html?searchHighlight=tonegenout&s\\_tid=doc\\_srchtile](https://www.mathworks.com/help/simulink/examples/digital-waveform-generation-approximating-a-sine-wave.html?searchHighlight=tonegenout&s_tid=doc_srchtile)
16. SFDR- <https://www.mathworks.com/help/search.html?submitsearch=&qdoc=sfdr>
17. Spectrum - <https://www.mathworks.com/help/ident/ref/spectrum.html>
18. HDL Coder –  
[https://www.mathworks.com/help/hdlcoder/index.html?searchHighlight=hdl&s\\_tid=doc\\_srchtile](https://www.mathworks.com/help/hdlcoder/index.html?searchHighlight=hdl&s_tid=doc_srchtile)

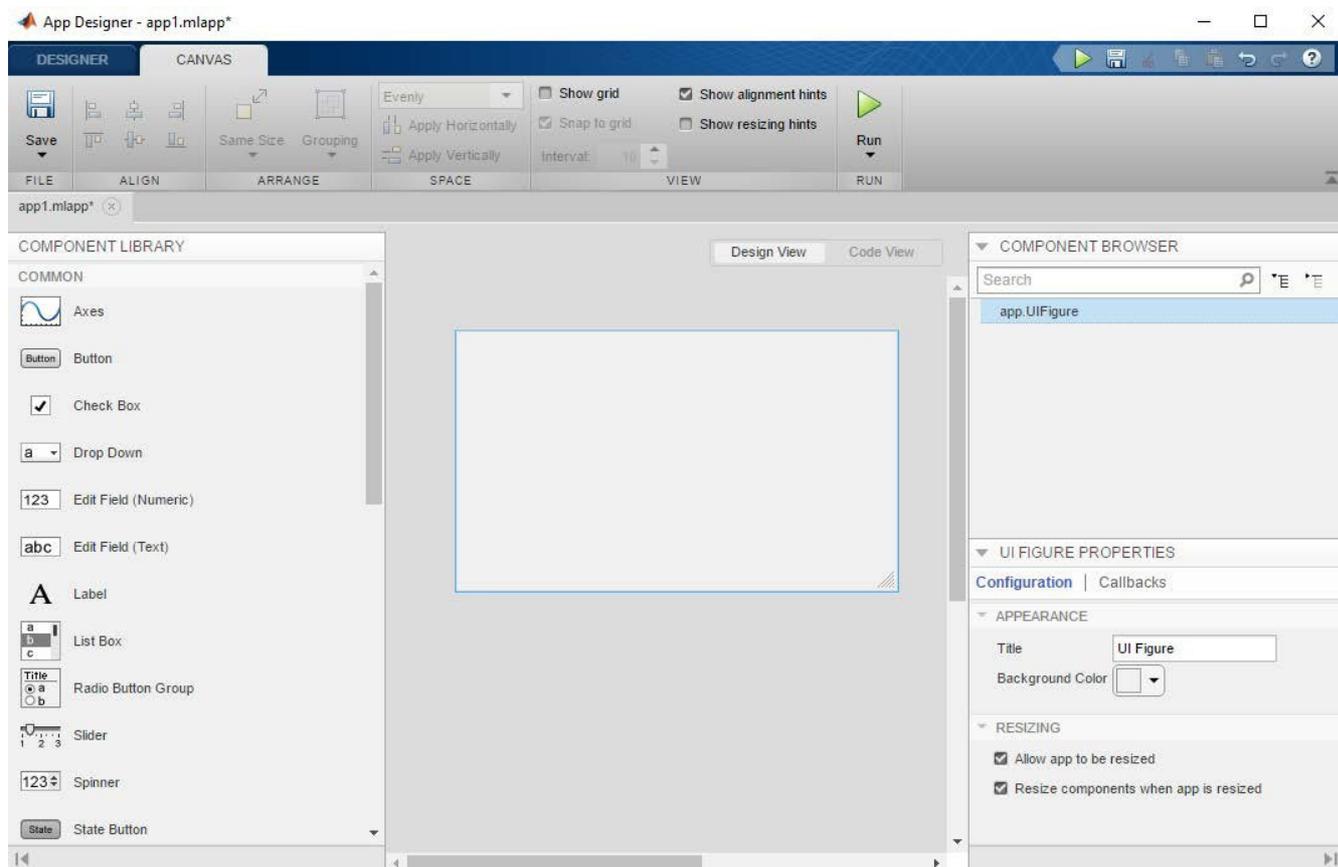
# ПРИЛОЖЕНИЕ А

## Интерфейс программы GUIDE



# ПРИЛОЖЕНИЕ Б

## Интерфейс программы AppDesigner



# ПРИЛОЖЕНИЕ В

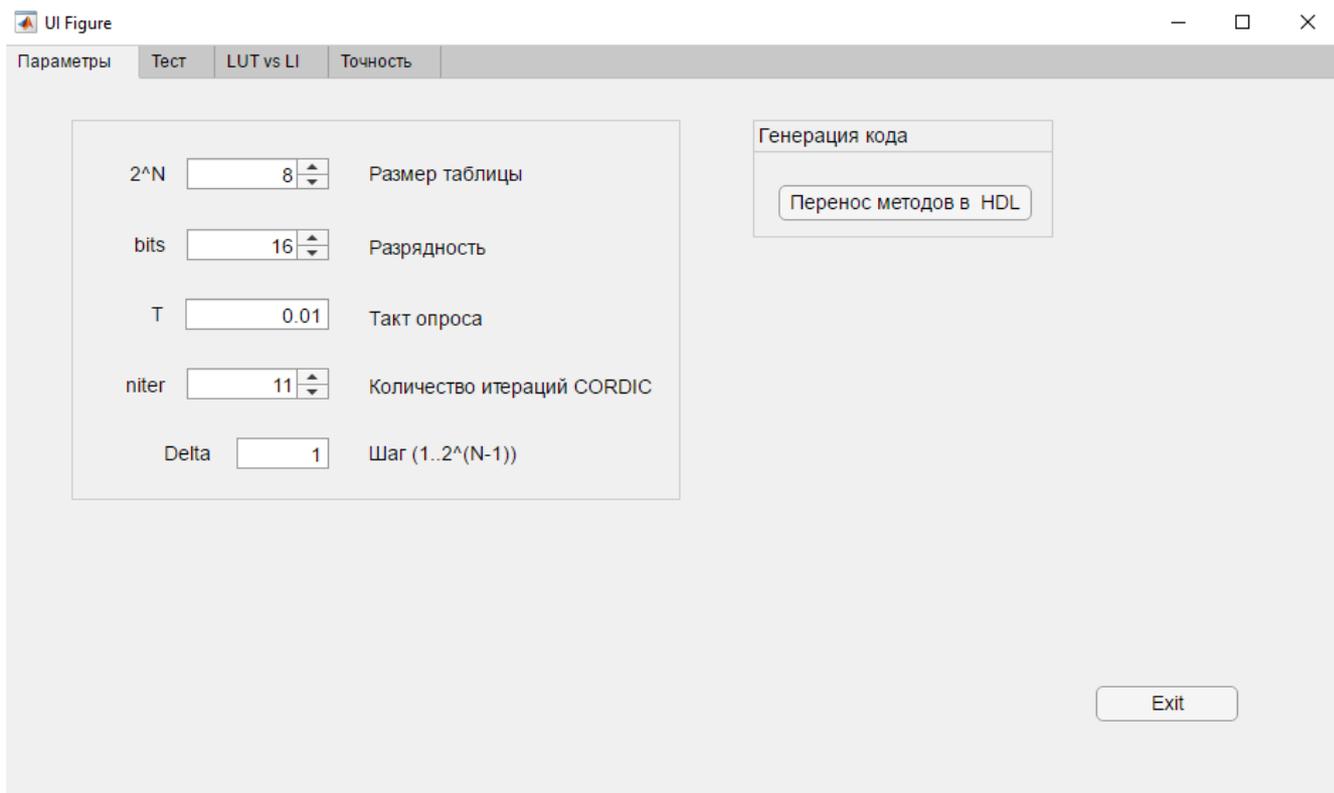


Рис. 1. Вкладка «Параметры»

# ПРИЛОЖЕНИЕ Г

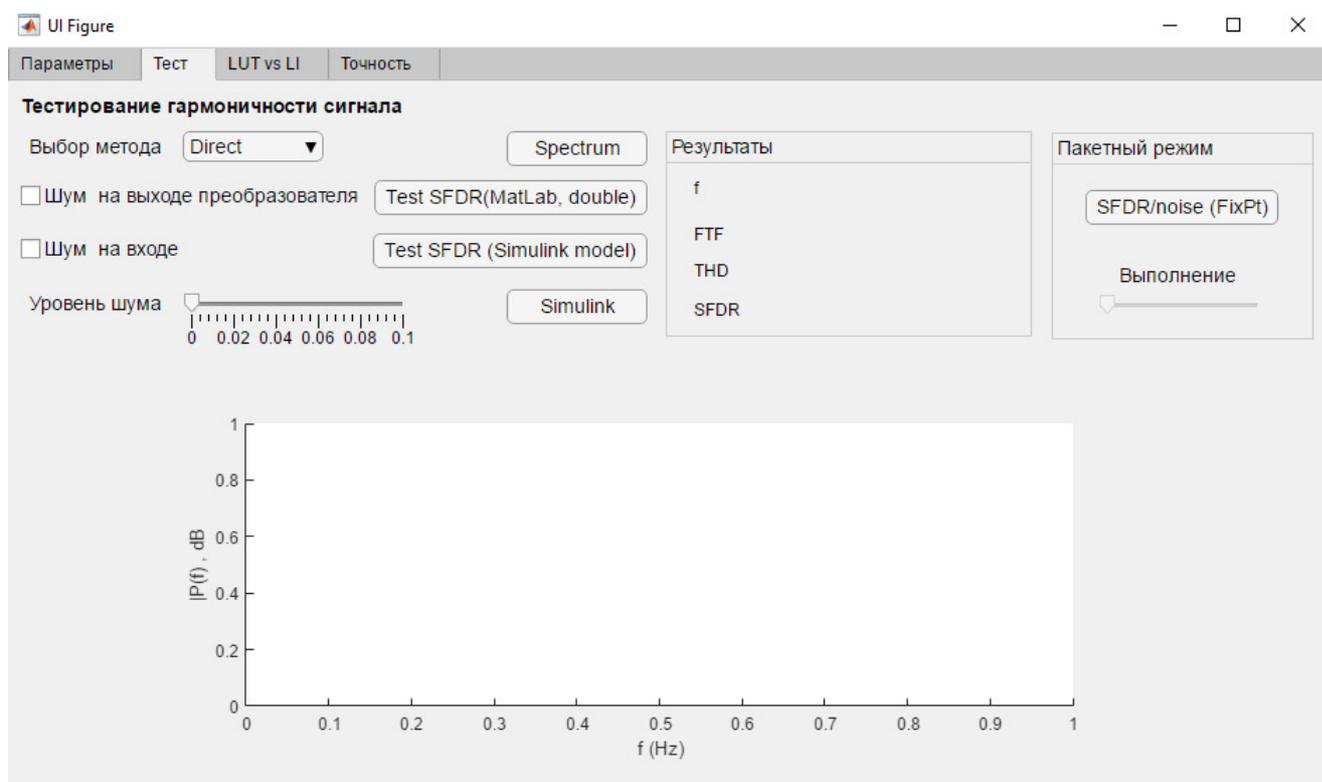


Рис. 1. Вкладка «Тест»

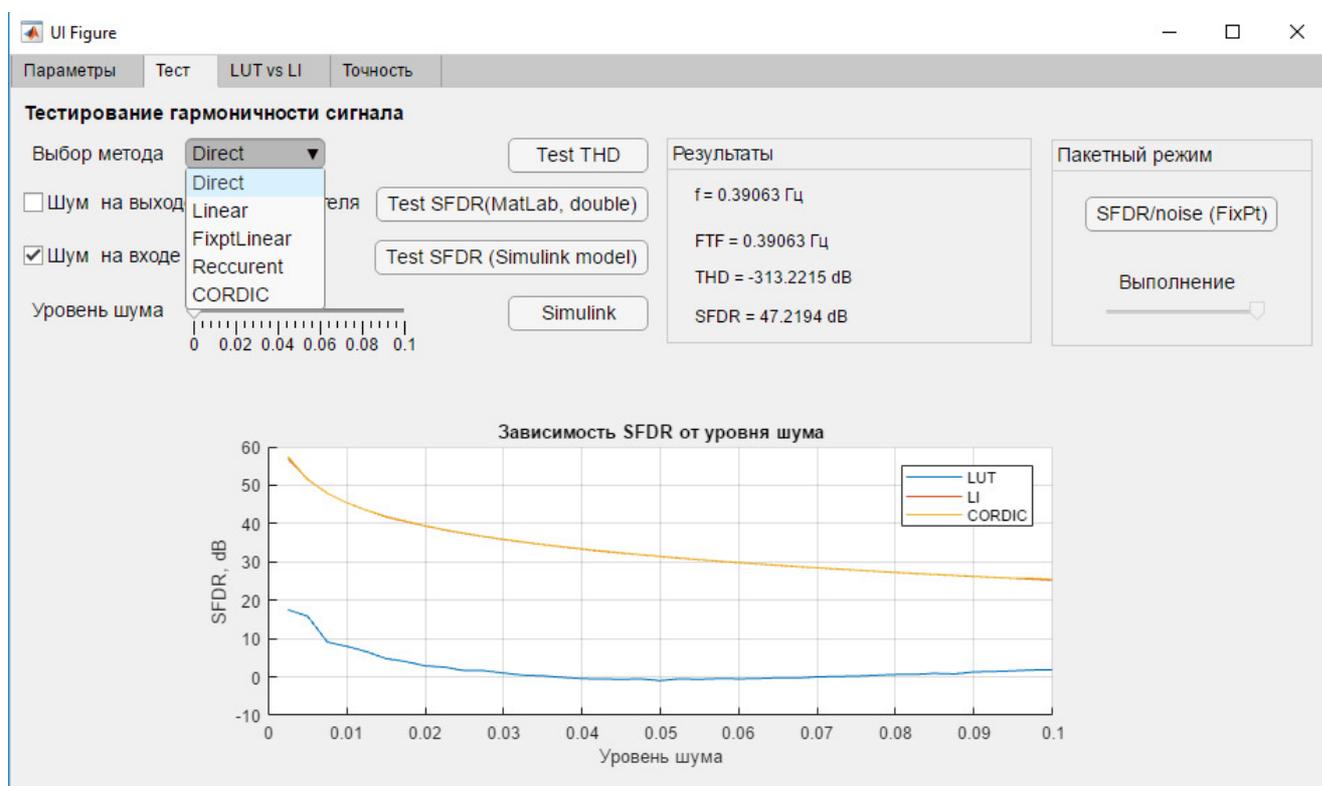


Рис. 2. Выпадающий список «Выбор метода»

# ПРИЛОЖЕНИЕ Д

## Работа кнопки «Spectrum»

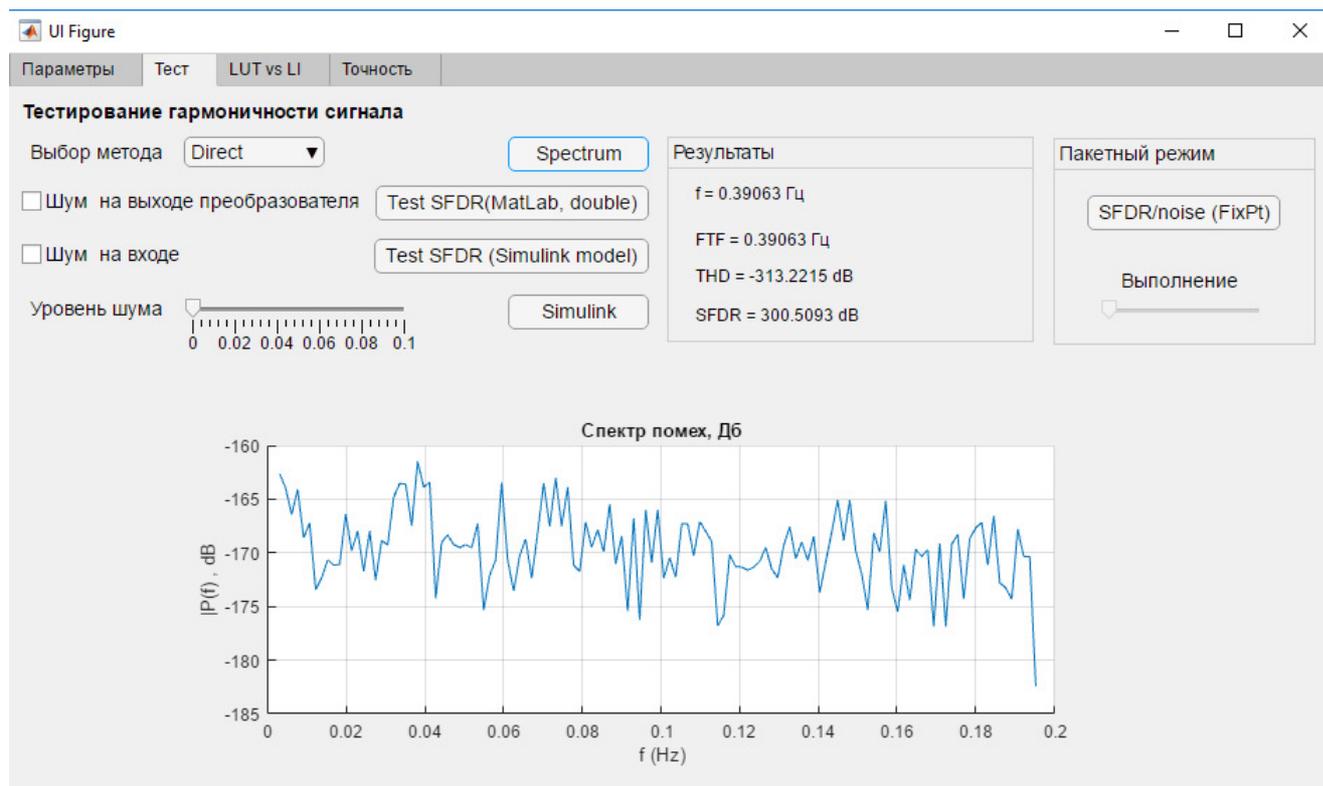


Рис. 1. Работа кнопки «Spectrum»

# ПРИЛОЖЕНИЕ Е

## Работа кнопки «Test SFDR (Matlab, double)»

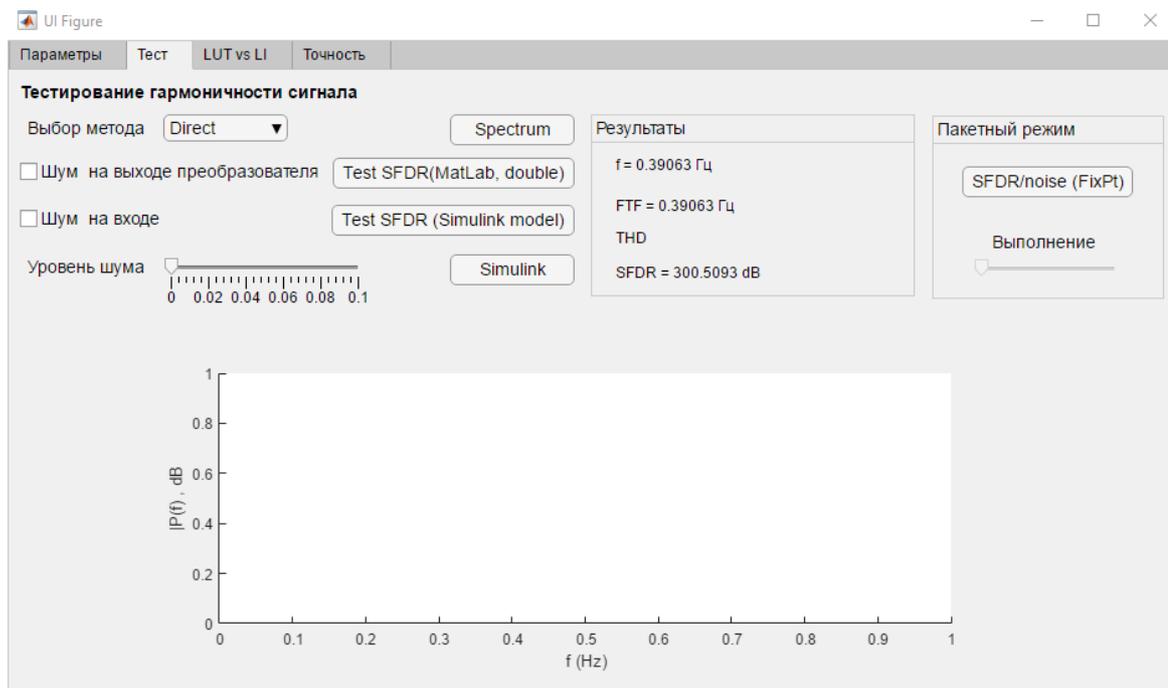


Рис. 1 Окно программы

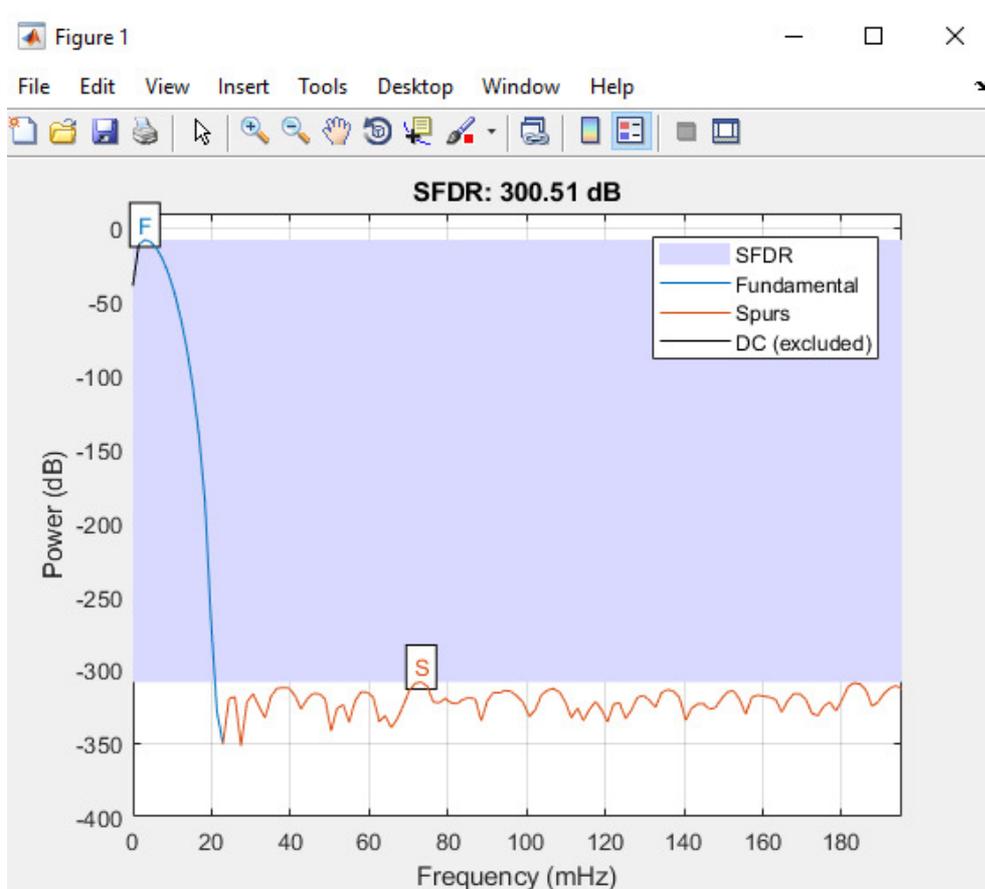


Рис. 2 График SFDR

Изм				
Лист	№ докум.	Подпись	Дата	

27.03.04.2018.293 ПЗ

# ПРИЛОЖЕНИЕ Ж

## Работа кнопки «Test SFDR (Simulink)»

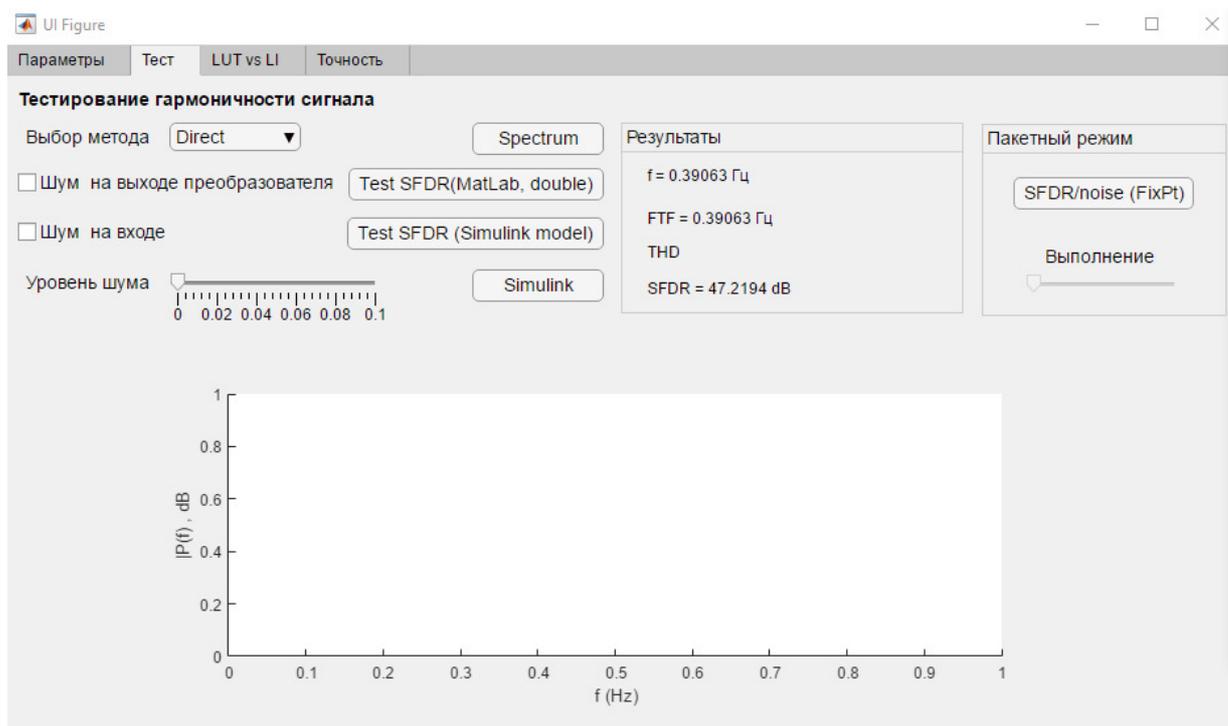


Рис. 1 Окно программы

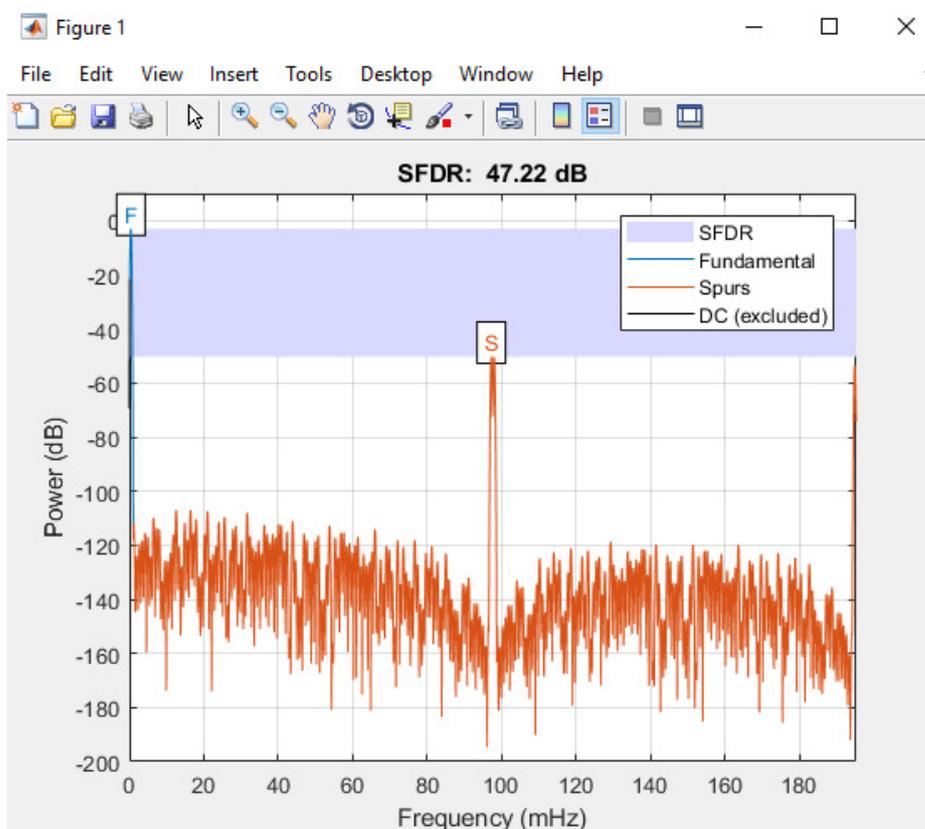


Рис. 2 График SFDR

Изм				
Лист	№ докум.	Подпись	Дата	

27.03.04.2018.293 ПЗ

Лист  
40

## ПРИЛОЖЕНИЕ И

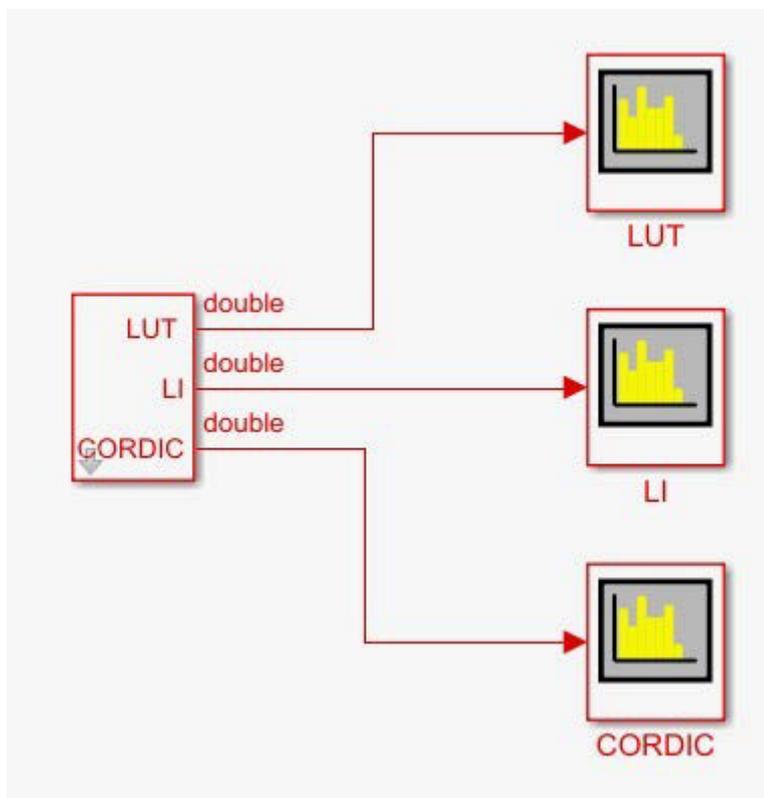


Рис. 1. Работа кнопки «Test SFDR (Simulink)»

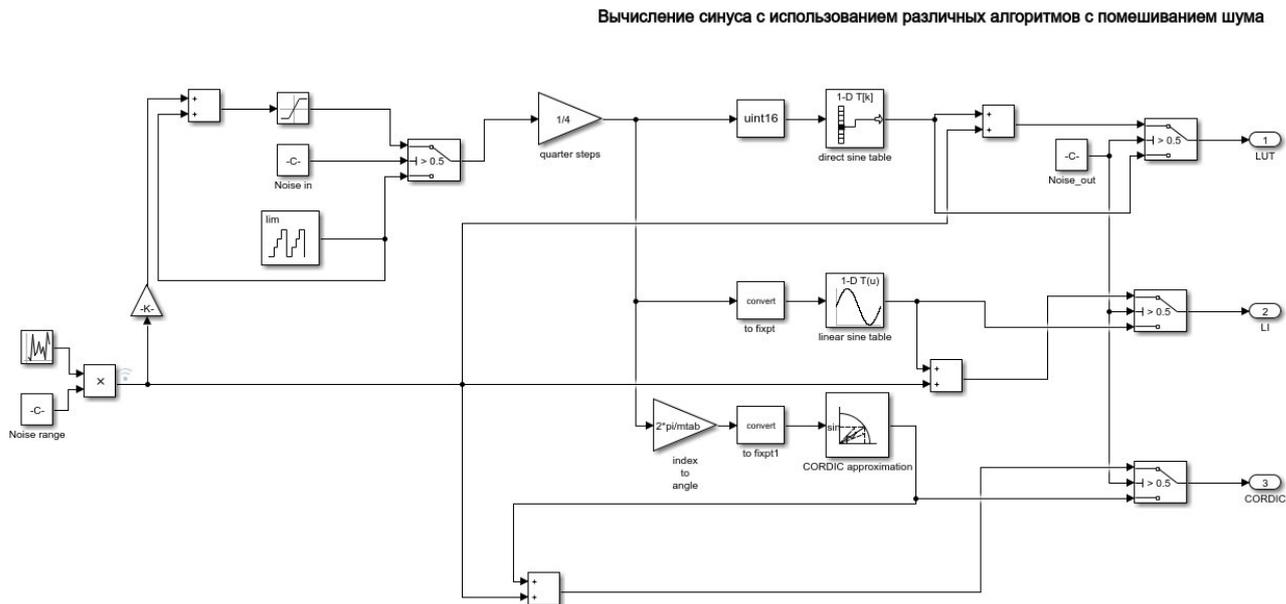


Рис. 2. Подсистема в среде «Simulink»

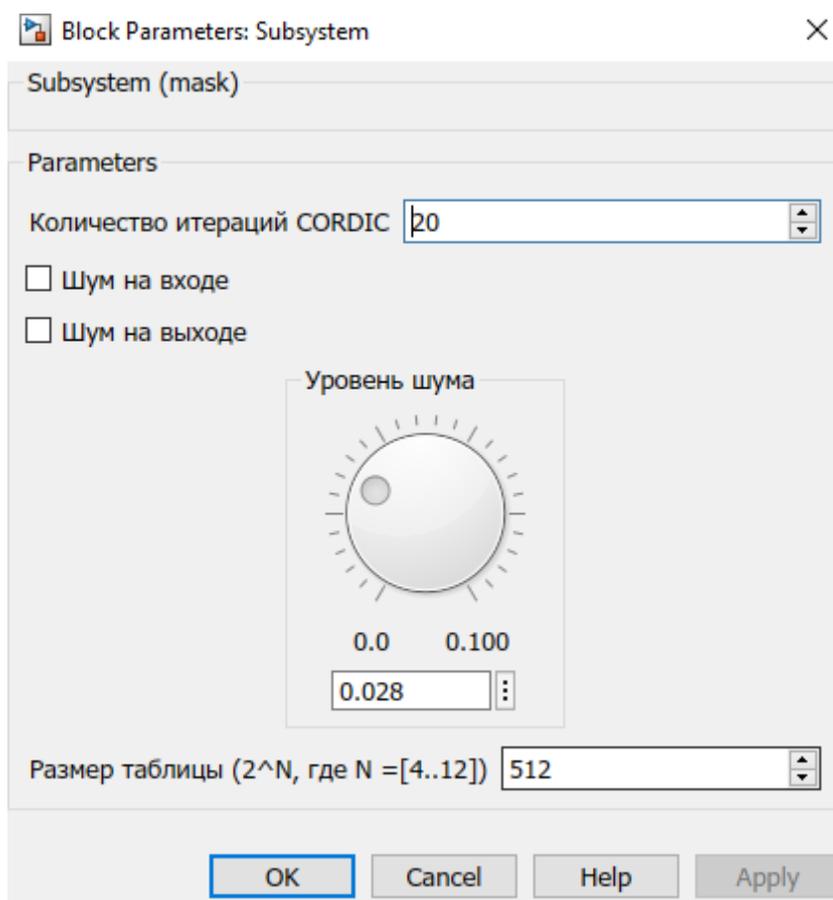


Рис. 3. Задание параметров подсистемы

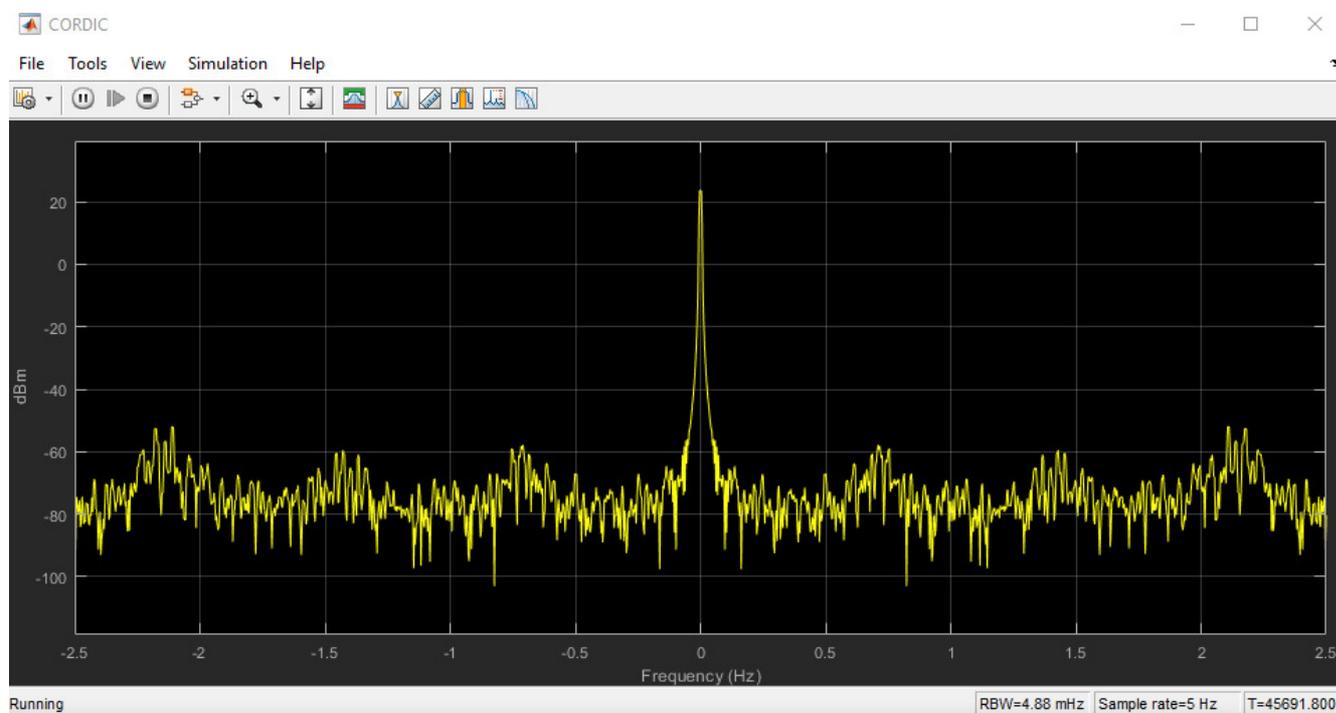


Рис. 4 Спектр сигнала метода CORDIC

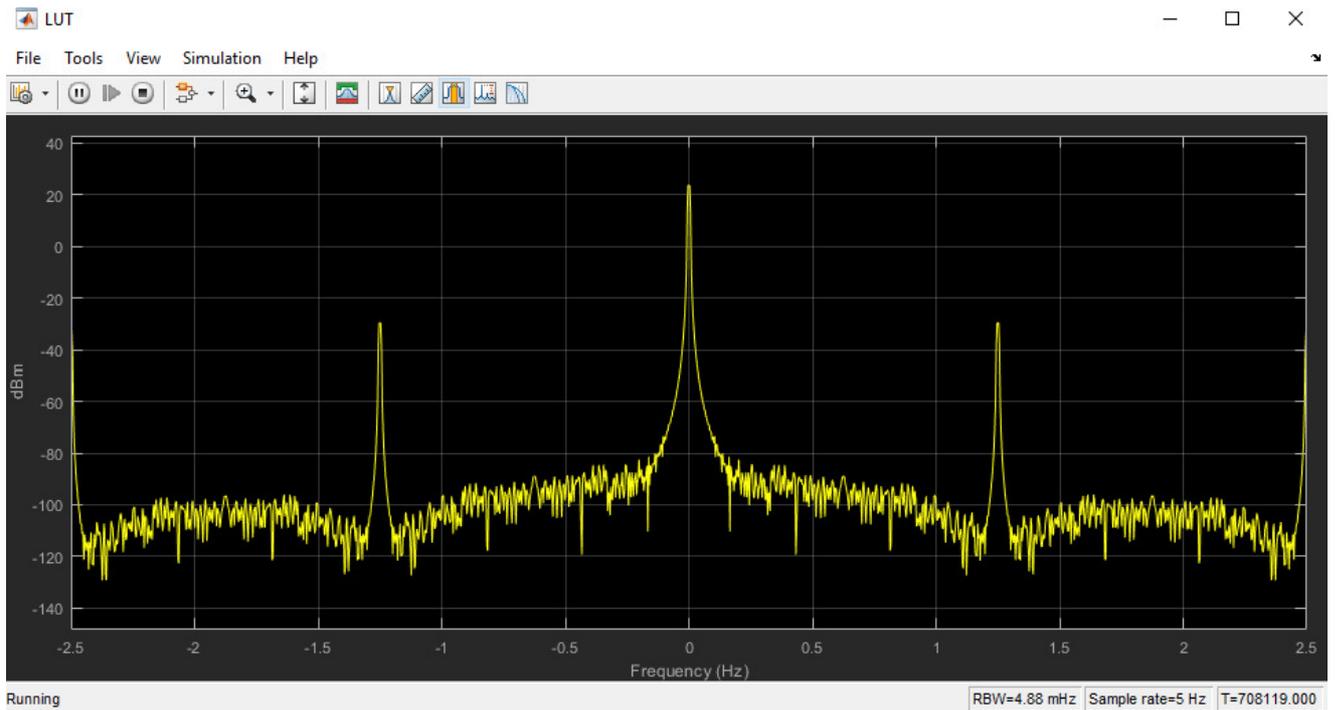


Рис. 5 Спектр сигнала табличного метода

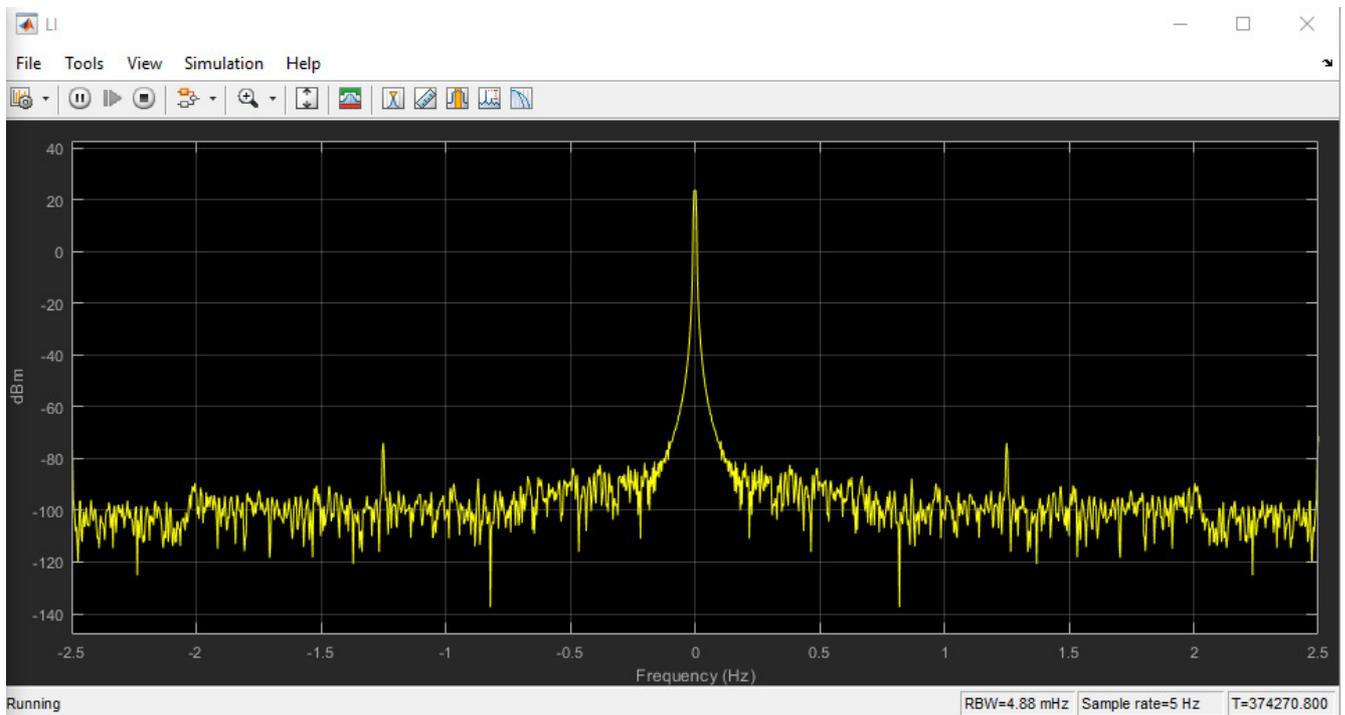
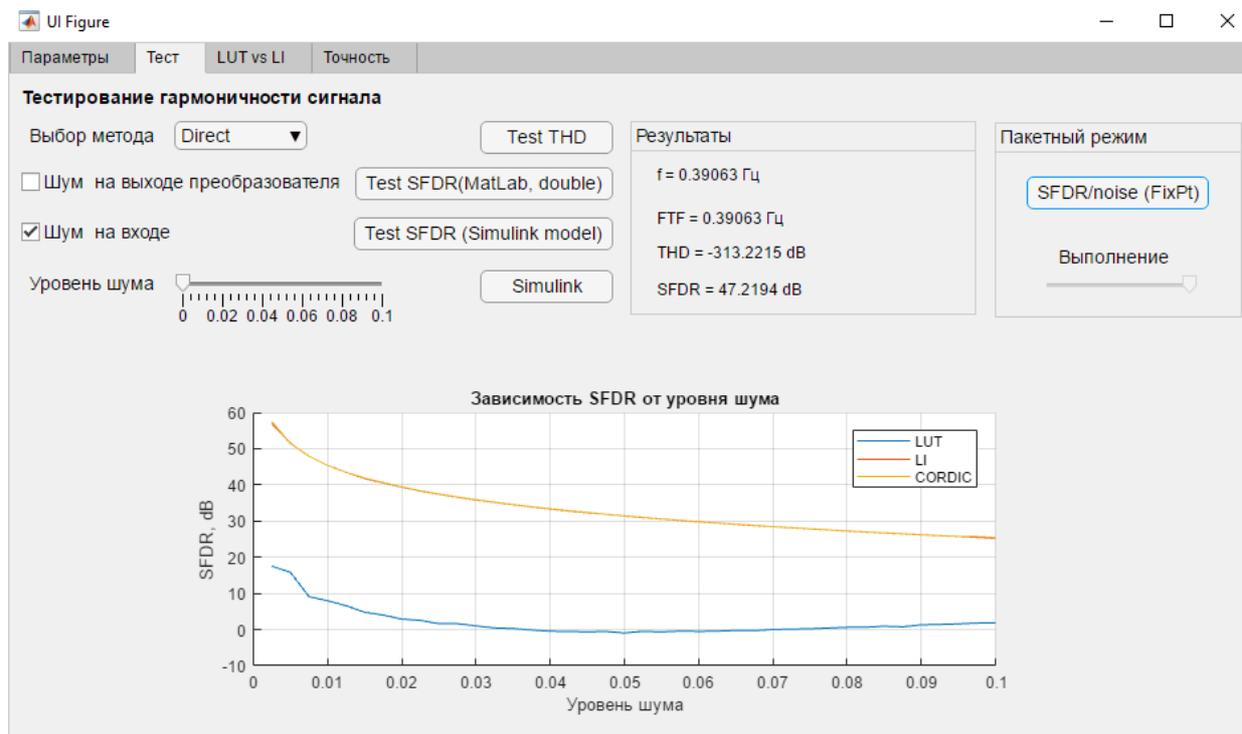


Рис. 4 Спектр сигнала табличного метода с линейной интерполяцией

# ПРИЛОЖЕНИЕ К

## Работа кнопки «SFDR/noise (FixPt)»

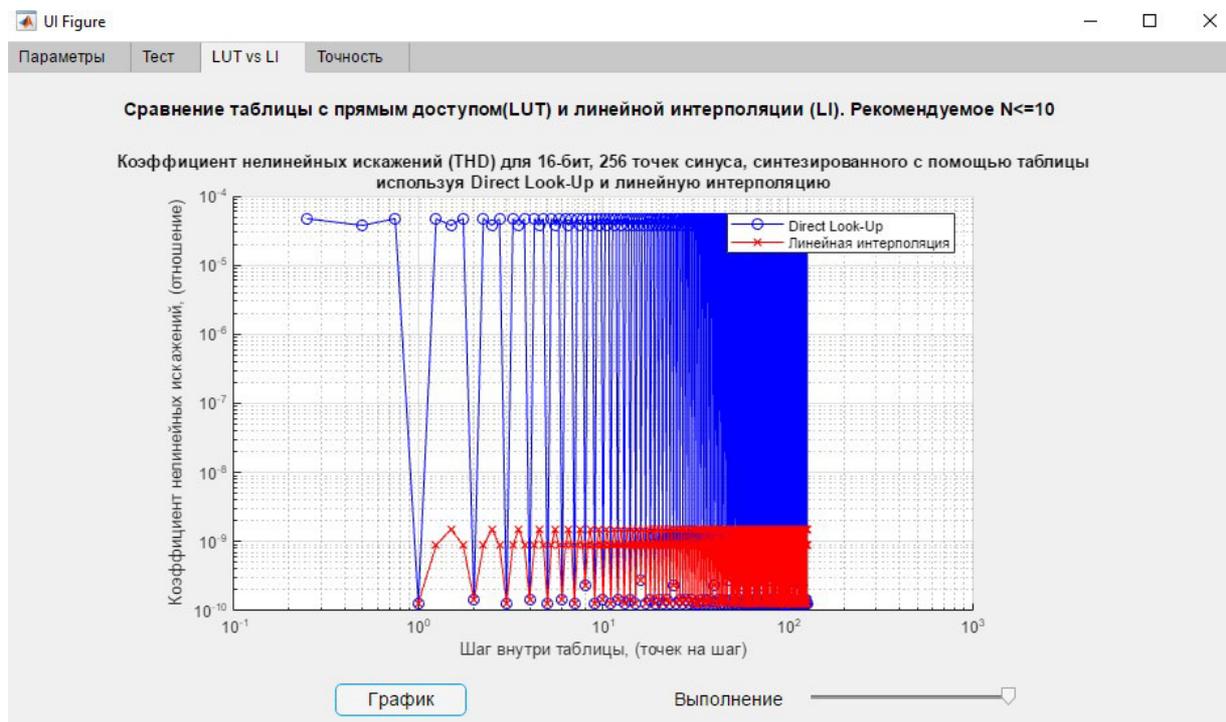


Изм				
Лист	№ докум.	Подпись	Дата	

27.03.04.2018.293 ПЗ

# ПРИЛОЖЕНИЕ Л

## Сравнение таблиц с линейной интерполяцией и без нее



Изм				
Лист	№ докум.	Подпись	Дата	

27.03.04.2018.293 ПЗ

## ПРИЛОЖЕНИЕ М

### Вкладка «Точность»

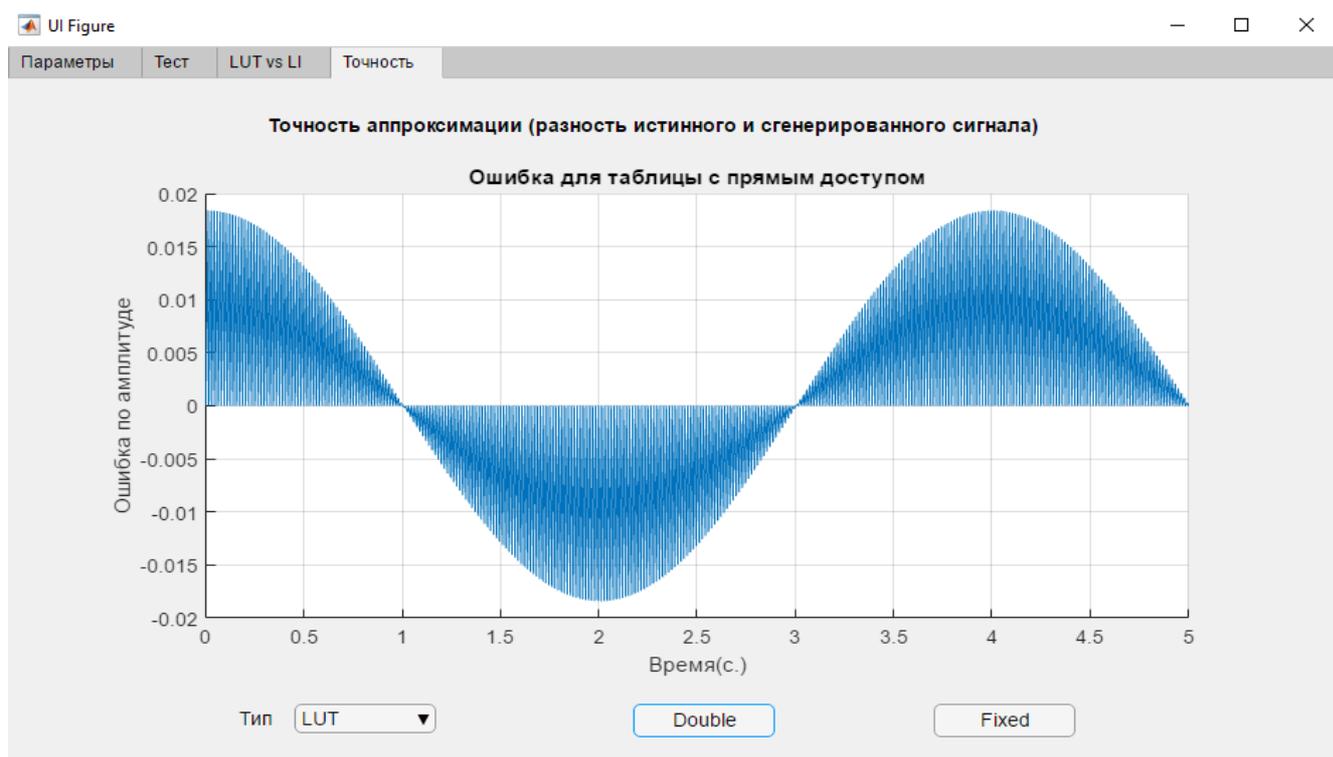


Рис. 1 Ошибка для таблицы с прямым доступом в арифметике с плавающей точкой

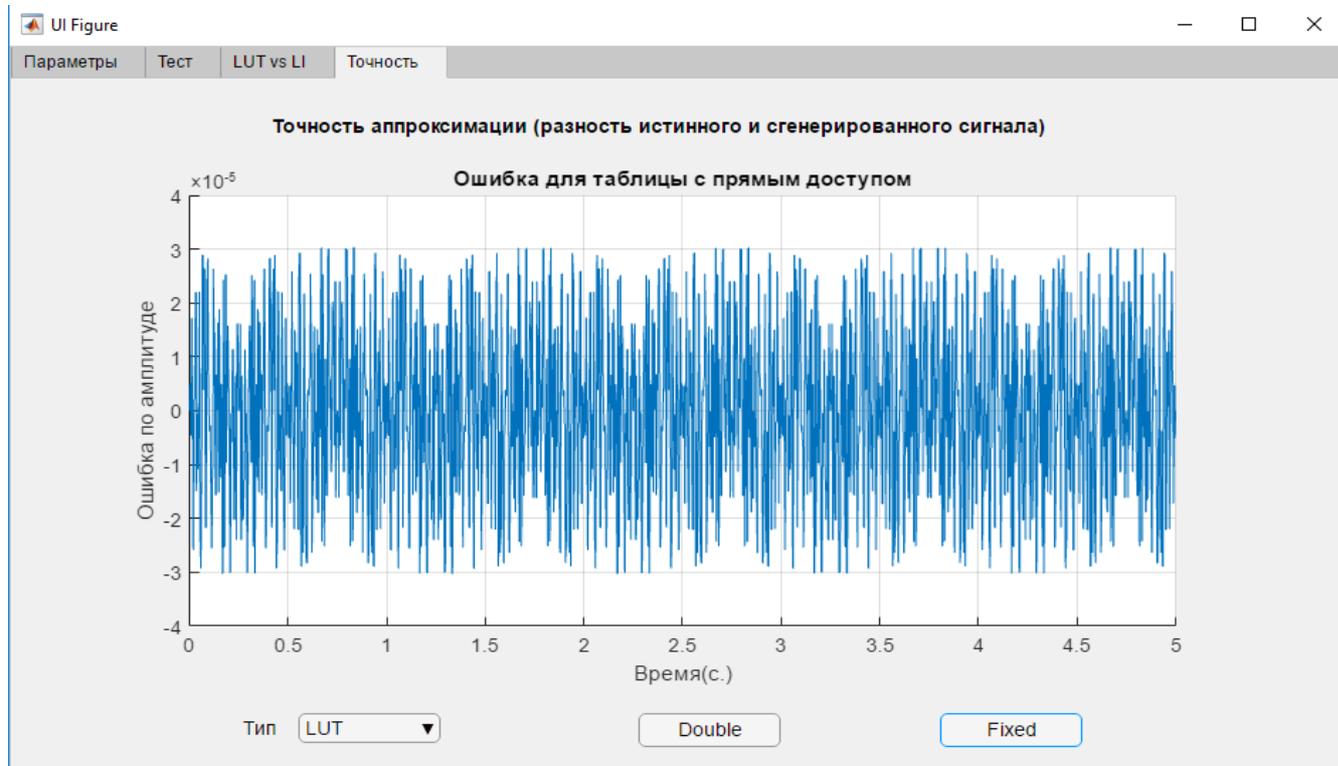


Рис. 2 Ошибка для таблицы с прямым доступом в арифметике с фиксированной точкой

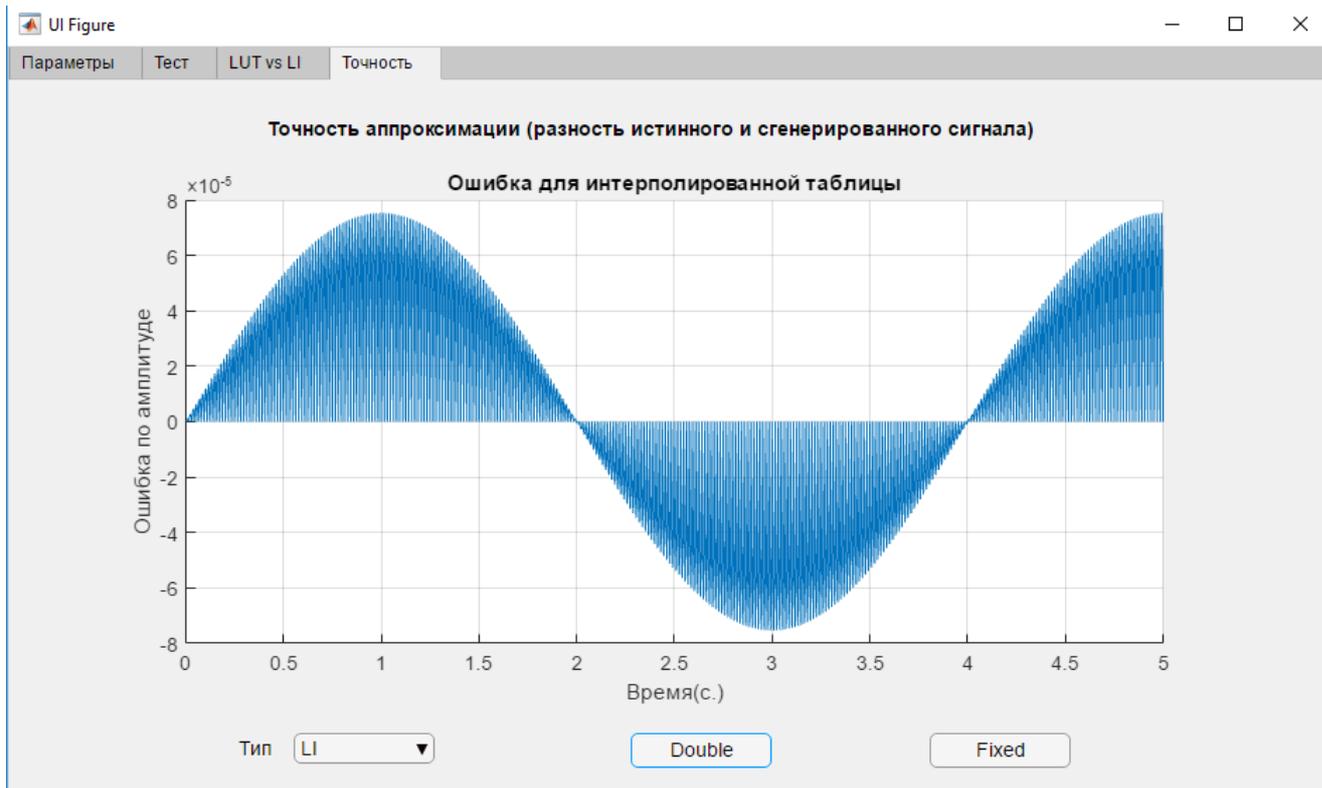


Рис. 3 Ошибка для таблицы с линейной интерполяцией в арифметике с плавающей точкой

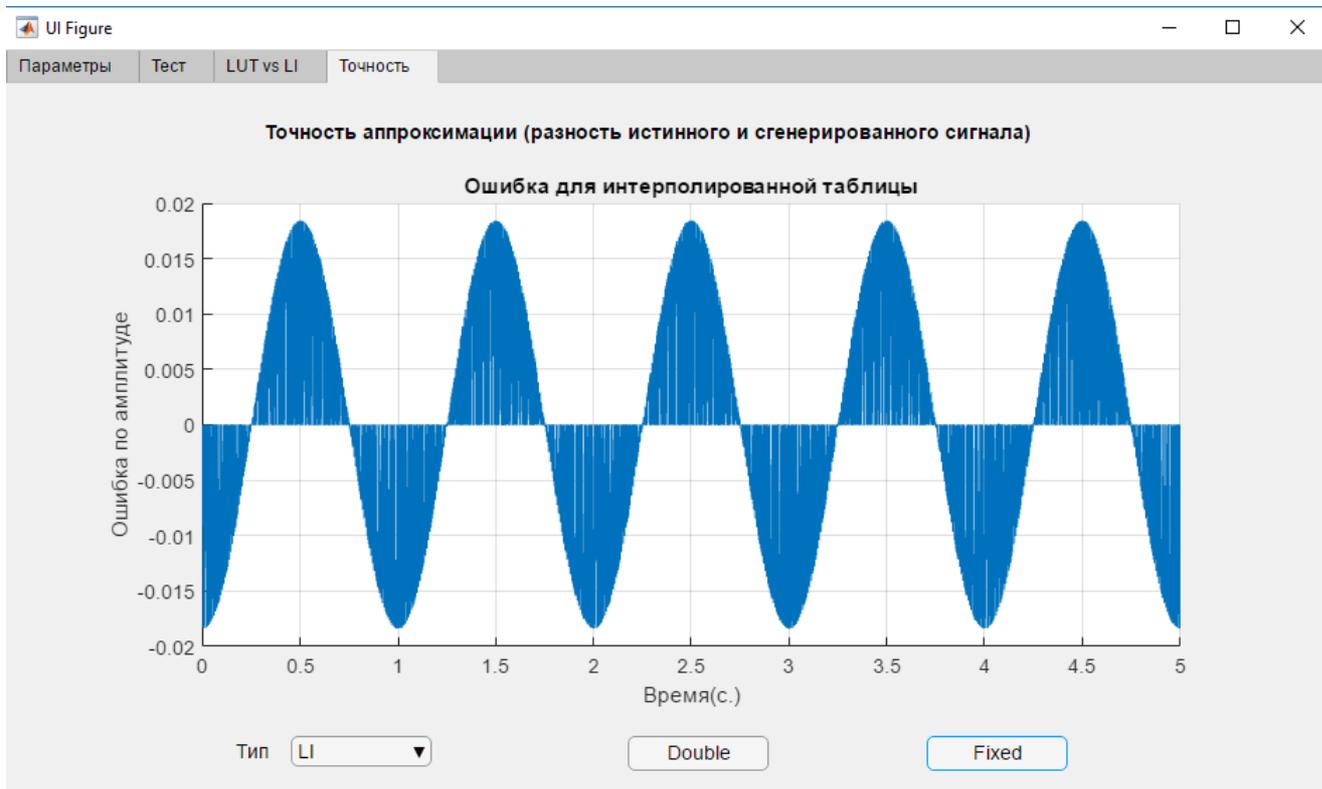


Рис. 4 Ошибка для таблицы с линейной интерполяцией в арифметике с фиксированной точкой

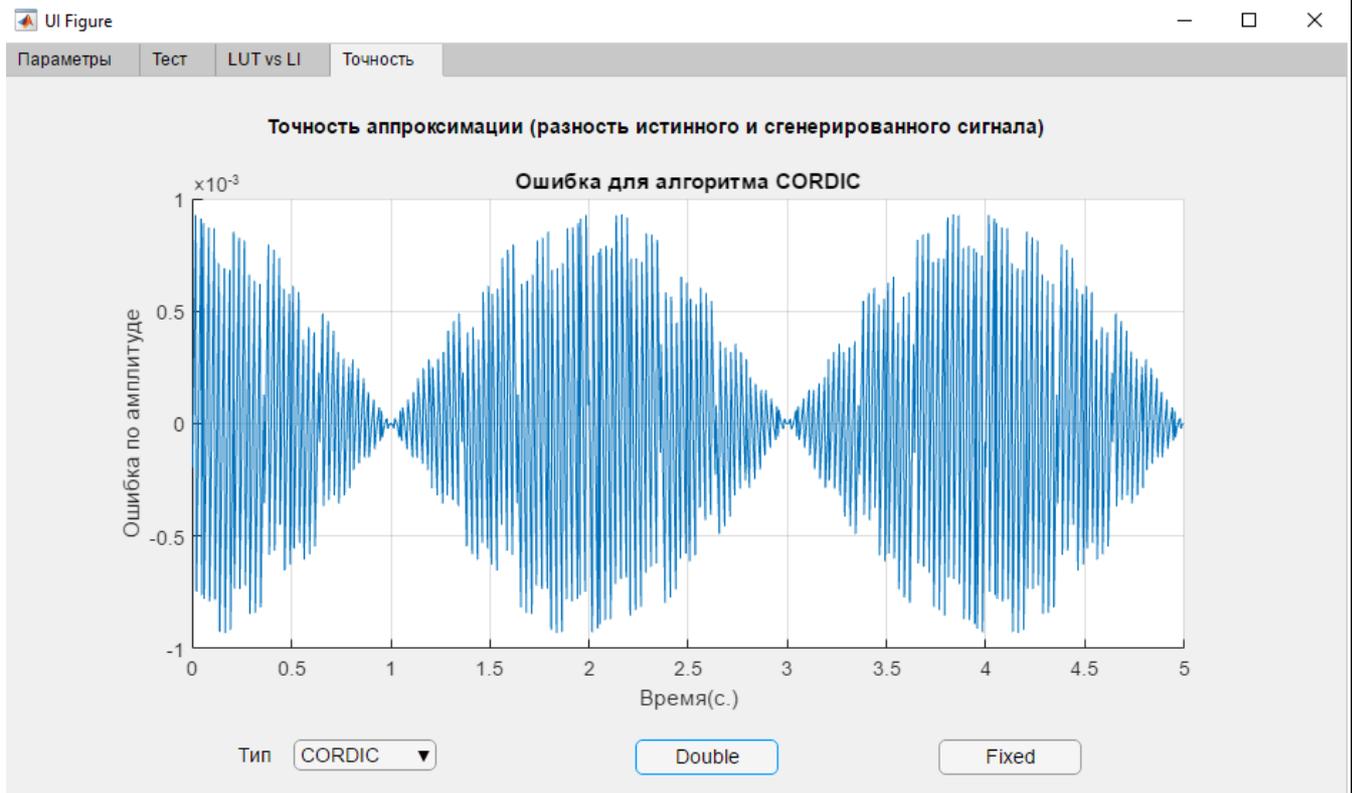


Рис. 5 Ошибка для метода CORDIC в арифметике с плавающей точкой

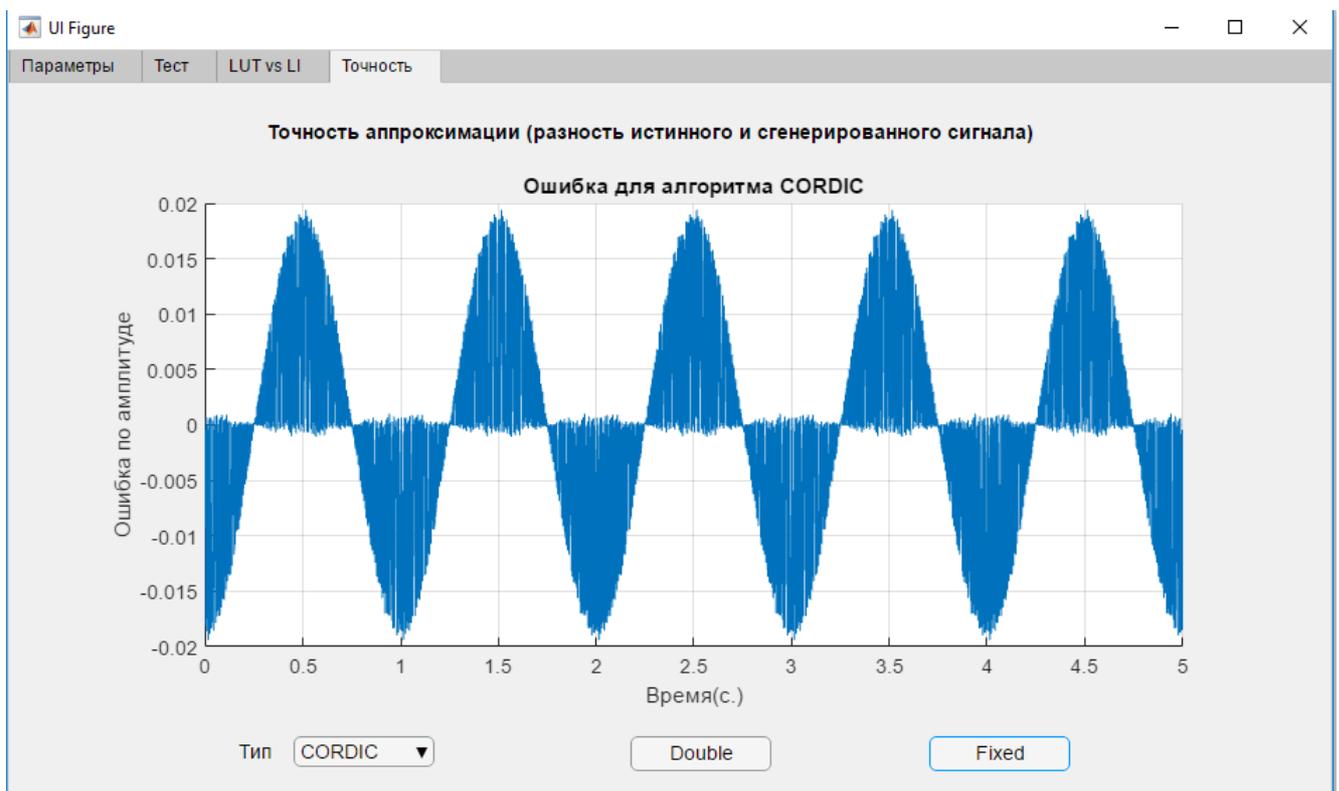


Рис. 6 Ошибка для метода CORDIC в арифметике с фиксированной точкой

## ПРИЛОЖЕНИЕ Н

### Листинг программы

```
classdef app12017b < matlab.apps.AppBase
% Properties that correspond to app components
properties (Access = public)
    UIFigure          matlab.ui.Figure
    TabGroup          matlab.ui.container.TabGroup
    Tab               matlab.ui.container.Tab
    Panel_3           matlab.ui.container.Panel
    HDLButton         matlab.ui.control.Button
    Panel_4           matlab.ui.container.Panel
    Label_4           matlab.ui.control.Label
    Label_5           matlab.ui.control.Label
    Label_9           matlab.ui.control.Label
    N1Label           matlab.ui.control.Label
    TEditFieldLabel  matlab.ui.control.Label
    T                 matlab.ui.control.NumericEditField
    DeltaLabel        matlab.ui.control.Label
    Delta             matlab.ui.control.NumericEditField
    NSpinnerLabel     matlab.ui.control.Label
    N                 matlab.ui.control.Spinner
    bitsSpinnerLabel  matlab.ui.control.Label
    bits              matlab.ui.control.Spinner
    niterLabel        matlab.ui.control.Label
    NIter             matlab.ui.control.Spinner
    CORDICLabel       matlab.ui.control.Label
    ExitButton        matlab.ui.control.Button
    Tab_3             matlab.ui.container.Tab
end
```

Label	matlab.ui.control.Label
TipOut	matlab.ui.control.DropDown
Noise	matlab.ui.control.CheckBox
Label_3	matlab.ui.control.Label
SliderNoise	matlab.ui.control.Slider
UIAxesSpectr	matlab.ui.control.UIAxes
TestTHDButton	matlab.ui.control.Button
Label_6	matlab.ui.control.Label
TestSFDRMatLabdoubleButton	matlab.ui.control.Button
TestSFDRSimulinkmodelButton	matlab.ui.control.Button
Error	matlab.ui.control.Label
Noise_in	matlab.ui.control.CheckBox
SimulinkButton	matlab.ui.control.Button
Panel	matlab.ui.container.Panel
SFDRLabel	matlab.ui.control.Label
FTFLabel	matlab.ui.control.Label
THD	matlab.ui.control.Label
f	matlab.ui.control.Label
Panel_2	matlab.ui.container.Panel
Label_11	matlab.ui.control.Label
Slider_busy_SFDR_noise	matlab.ui.control.Slider
SFDRnoiseFixPtButton	matlab.ui.control.Button
LUTvsLITab	matlab.ui.container.Tab
LUTLIN10Label	matlab.ui.control.Label
UIAxes_LUTvsLI	matlab.ui.control.UIAxes
Button	matlab.ui.control.Button
Label_10	matlab.ui.control.Label
Slider_busy	matlab.ui.control.Slider

```

Tab_2          matlab.ui.container.Tab
Label_7        matlab.ui.control.Label
UIAxes         matlab.ui.control.UIAxes
DoubleButton   matlab.ui.control.Button
Label_8        matlab.ui.control.Label
DropDown       matlab.ui.control.DropDown
FixedButton    matlab.ui.control.Button

```

```
end
```

```
methods (Access = private)
```

```
% Работа кнопки TestTHDButton
```

```
function TestTHDButtonPushed(app, event)
```

```
    global Ts k4 fs
```

```
    app.Error.Text = ' ';
```

```
    N = 2^app.N.Value;
```

```
    T = app.T.Value;
```

```
    FTF = 1/N/T; % Фундаментальная частота
```

```
    app.FTFLabel.Text = ['FTF = ' num2str(FTF) ' Гц'];
```

```
    B = app.Delta.Value;
```

```
    f = B*FTF; % частота сигнала
```

```
    app.f.Text = ['f = ' num2str(f) ' Гц'];
```

```
    [A,B1] = rat(B); % КОЛ-ВО ЦИКЛОВ
```

```
    angle = 2*pi * (0:(N-1))/N;
```

```
    if app.Noise_in.Value
```

```
        angle = angle + app.SliderNoise.Value*(rand(size(angle))-0.5)/N*10/2;
```

```
    end
```

```
    s = sin( angle )';
```

```
    tipout = app.TipOut.Value;
```

```
    switch tipout
```

```
        case 'FixLinear'
```

```

s = num2fixpt( s, sfrac(app.bits.Value), [], 'Nearest', 'on');
case 'Reccurent'
    Ts = T;
    fs = f;
    theta = 2*pi*Ts*f;
    k4= sin(theta);
    open_system('Reccurent','loadonly');
    set_param('Reccurent', 'StopFcn','');
    sim('Reccurent');
    s = ReccurentOut{1}.Values.Data;
    tipout = 'linear';
    bdclose('Reccurent');
case 'CORDIC'
    niter = app.NIter.Value;
    s = cordicsin( angle, niter );
    tipout = 'direct';

end
if app.Noise.Value
    s = s + app.SliderNoise.Value*(rand(size(s))-0.5);
end
[thd,P1] = ssinthd( s, B, B1*N, A, tipout, app.bits.Value);

app.THD.Text = ['THD = ' num2str(10*log10(thd)) ' dB'];

L = B1*N;
f = FTF*(0:(L/2))/L;
switch app.TipOut.Value

```

```

case 'CORDIC'
    ind = P1>0;
    f = f(ind);
    P1 = P1(ind);
end
plot(app.UIAxesSpectr,f,10*log10(P1)), grid(app.UIAxesSpectr, 'on')
title(app.UIAxesSpectr, 'Спектр помех, Дб')

```

```
end
```

```
% Работа кнопки DoubleButton
```

```
function DoubleButtonPushed(app, event)
```

```
    global gN gbits gniter
```

```
    gN = 2^app.N.Value;
```

```
    gbits = app.bits.Value;
```

```
    gniter = app.NIter.Value;
```

```
    open_system('sldemo_tonegen','loadonly');
```

```
    set_param('sldemo_tonegen', 'StopFcn','');
```

```
    sim('sldemo_tonegen');
```

```
    switch app.DropDown.Value
```

```
        case 'LUT'
```

```
            plot(app.UIAxes,tonegenOut.time, tonegenOut.signals(1).values);
```

```
            grid(app.UIAxes, 'on')
```

```
            title(app.UIAxes,{'Ошибка для таблицы с прямым доступом'});
```

```
        case 'LI'
```

```
            plot(app.UIAxes,tonegenOut.time, tonegenOut.signals(2).values);
```

```
            grid(app.UIAxes, 'on')
```

```

title(app.UIAxes, {'Ошибка для интерполированной таблицы'});
case 'CORDIC'
    plot(app.UIAxes, tonegenOut.time, tonegenOut.signals(3).values);
grid(app.UIAxes, 'on')
    title(app.UIAxes, {'Ошибка для алгоритма CORDIC'});
case 'LUT8'
    plot(app.UIAxes, tonegenOut.time, tonegenOut.signals(4).values);
grid(app.UIAxes, 'on')
    title(app.UIAxes, {'Ошибка для таблицы с сжатием в 8 раз'});
end
bdclose('sldemo_tonegen');
end
% Работа кнопки FixedButton
function FixedButtonPushed(app, event)
    global gN gbits gniter
    gN = 2^app.N.Value;
    gbits = app.bits.Value;
    gniter = app.NIter.Value;
    % Используются преднастроенные блоки генерации синуса
    % Также Симулинк включает в себя блок генерации синуса с аналоговым и
    дискретным режимами работы, также с блоками синуса и косинуса в арифметике
    с фиксированной точкой, которые осуществляют аппроксимацию функции с ли-
    нейно интерполированной таблицей, которая использует четверть синусои-
    ды/косинусоиды
    % Данная схема использует источник синусоидальной волны в качестве
    эталонного сигнала и сравнивает ее с сигналом таблицы с линейной интерполяци-
    ей/без нее, также сравнивает с синусоидой, сгенерированной методом CORDIC в
    арифметике с фиксированной точкой

```

```
open_system('sldemo_tonegen_fixpt', 'loadonly');
    set_param('sldemo_tonegen_fixpt', 'StopFcn','');
    sim('sldemo_tonegen_fixpt');
    switch app.DropDown.Value
        case 'LUT'
            plot(app.UIAxes, tonegenOut.time, tonegenOut.signals(1).values);
            grid(app.UIAxes, 'on')
            title(app.UIAxes, {'Ошибка для таблицы с прямым доступом'});
        case 'LI'
            plot(app.UIAxes, tonegenOut.time, tonegenOut.signals(2).values);
            grid(app.UIAxes, 'on')
            title(app.UIAxes, {'Ошибка для интерполированной таблицы'});
        case 'CORDIC'
            plot(app.UIAxes, tonegenOut.time, tonegenOut.signals(3).values);
            grid(app.UIAxes, 'on')
            title(app.UIAxes, {'Ошибка для алгоритма CORDIC'});

        case 'LUT8'
            plot(app.UIAxes, tonegenOut.time, tonegenOut.signals(8).values);
            grid(app.UIAxes, 'on')
            title(app.UIAxes, {'Ошибка для таблицы с сжатием в 8 раз'});
    end
    bdclose('sldemo_tonegen_fixpt');
end
% Работа кнопки Button
function ButtonPushed(app, event)
    % Обзор поведения таблицы с прямым доступом и таблицы с линейной
    интерполяцией
```

% Файл <sldemo\_sweepable\_thd.m> выполняет полную частотную развертку таблиц с фиксированной точкой и позволяет более подробно понять поведение этой конструкции. Общее искажение 24-битной дискретной таблицы неподвижных точек измеряется при каждом размере шага, проходя через D точек за один раз, где D-число от 1 до  $N / 2$  с приращением на 0,25. N-256 пунктов в данном примере. Частота дискретна и поэтому является функцией частоты дискретизации. Обратите внимание на режимы поведения искажений на графике. При извлечении значения из таблицы точно в точке ошибка наименьшая; линейная интерполяция имеет меньшую ошибку, чем таблица с прямым доступом, во время извлечения значения не в точке. Что не интуитивно, так это то, что ошибка относительно постоянна для каждого из режимов до частоты Найквиста.

```
global gN gbits gniter
gN = 2^app.N.Value;
gbits = app.bits.Value;
gniter = app.NIter.Value;
```

```
tic, sldemo_sweepable_thd(gbits, gN, app.UIAxes_LUTvsLI, app.Slider_busy),
toc
end
```

```
% Работа кнопки TestSFDRMatLabdoubleButton
```

```
function TestSFDRMatLabdoubleButtonPushed(app, event)
```

```
global k4 fs Ts
app.Error.Text = ' ';
N = 2^app.N.Value;
T = app.T.Value;
FTF = 1/N/T; % Fundamental Table Frequency
app.FTFLabel.Text = ['FTF = ' num2str(FTF) ' Гц'];
B = app.Delta.Value;
```

```

f = B*FTF;
app.f.Text = ['f = ' num2str(f) ' Гц'];
[A,B1] = rat(B); % КОЛ-ВО ЦИКЛОВ
angle = 2*pi * (0:(N-1))/N;
%angle = repmat(angle,1,10);
if app.Noise_in.Value
    angle = angle + app.SliderNoise.Value*(rand(size(angle))-0.5)/N*10/2;
end
is = sin( angle )';
numPts = length(is);
delta = B;
bn = N;
bits = app.bits.Value;
a = A;
switch app.TipOut.Value
    case 'CORDIC'
        niter = app.NIter.Value;
        is = cordicsin( angle, niter )';
        index = floor(mod(delta*(0:(bn-1)),numPts));
        s = reshape( is(index+1), length(index), 1);
    case 'Direct'
        % Таблица с прямым доступом
        index = floor(mod(delta*(0:(bn-1)),numPts));
        s = reshape( is(index+1), length(index), 1);
    case 'Linear'
        % Линейная интерполяция
        xs = repmat(is, a, 1);
        nxs = length(xs);
        upts = (0:(nxs-1))/nxs;

```

```

index = (0:delta:a*numPts)';
index = index(1:end-1);
s = interp1( upts, xs, index/nxs );
case 'FixptLinear'
% Таблица с линейной интерполяцией в арифметике с фикс. точкой
xs = repmat(is, a, 1);
nxs = length(xs);
upts = (0:(nxs-1))/nxs;
index = (0:delta:a*numPts)';
index = index(1:end-1);
if exist('fixpt_interp1.m','file')
    s = fixpt_interp1( upts, xs, index/nxs, ...
        sfrac(bits),[], sfrac(bits),[],'nearest');
else
    error(message('simdemos:ssinthd:noSimulinkFixedPointLicense'));
end
case 'Reccurent'
Ts = T;
fs = f;
theta = 2*pi*Ts*f;
k4= sin(theta);
open_system('Reccurent','loadonly');
set_param('Reccurent', 'StopFcn','');
sim('Reccurent');
s = ReccurentOut{1}.Values.Data;
tipout = 'linear';
bdclose('Reccurent');

otherwise

```

```

    error(message('simdemos:ssinthd:noOption'));
end
if app.Noise.Value
    s = s + app.SliderNoise.Value*(rand(size(s))-0.5);
end
SFDR = sfdr(s,f);
app.SFDRLabel.Text = ['SFDR = ' num2str(SFDR) ' dB'];
sfdr(s,f);
%ВЫВОДЫ:
% Шум на вх. положительно влияет для Linear, FixptLinear
% Шум на вых. положительно влияет на Direct, CORDIC, Reccurent

end
% Работа кнопки TestSFDRSimulinkmodelButton
function TestSFDRSimulinkmodelButtonPushed(app, event)
    global ntab niter range_noise Ts noise_out noise_in gbits
    app.Error.Text = '';
    N = 2^app.N.Value;
    ntab = N;
    T = app.T.Value;
    Ts = T;
    FTF = 1/N/T; % Fundamental Table Frequency
    fs = FTF;
    app.FTFLabel.Text = ['FTF = ' num2str(FTF) ' Гц'];
    app.f.Text = ['f = ' num2str(FTF) ' Гц'];
    niter = app.NIter.Value;
    noise_in = app.Noise_in.Value;
    range_noise = app.SliderNoise.Value;
    noise_out = app.Noise.Value;

```

```

gbits = app.bits.Value;
niter = min([niter gbits]);
switch app.TipOut.Value
    case {'Direct','Linear','FixLinear','CORDIC'}
        open_system('tonegen_fixpt_noise_param','loadonly');
        set_param('tonegen_fixpt_noise_param','StopFcn','');
        sim('tonegen_fixpt_noise_param');
        switch app.TipOut.Value
            case 'Direct'
                s = tonegenOut.signals(1).values;
            case 'Linear'
                s = tonegenOut.signals(2).values;
            case 'FixLinear'
                s = tonegenOut.signals(2).values;
                s = num2fixpt( s, sfrac(app.bits.Value), [], 'Nearest', 'on');
            case 'CORDIC'
                s = tonegenOut.signals(3).values;
        end
        bdclose('tonegen_fixpt_noise_param');
    case 'Reccurent'
        Ts = T;
        fs = FTF;
        theta = 2*pi*Ts*fs;
        k4= sin(theta);
        open_system('Reccurent','loadonly');
        set_param('Reccurent','StopFcn','');
        sim('Reccurent');
        s = ReccurentOut{1}.Values.Data;
        tipout = 'linear';

```

```

    bdclose('Reccurent');

end

SFDR = sfdr(s,fs);
app.SFDRLabel.Text = ['SFDR = ' num2str(SFDR) ' dB'];
sfdr(s,fs);

% Direct не переводится в фикс. точку в матлабе
% В симулинке переводится в фикс. точку
% Для устранения спура нужно переводить в double

end

% Работа кнопки SimulinkButton
function SimulinkButtonPushed(app, event)
    open_system('tonegen_fixpt_noise_param_mask');
end

% Выпадающий список DropDown
function DropDownValueChanged(app, event)
    value = app.DropDown.Value;

end

% Работа кнопки SFDRnoiseFixPtButton
function SFDRnoiseFixPtButtonPushed(app, event)
    %Зависимость SFDR от уровня шума
    global ntab niter range_noise Ts noise_out noise_in gbits
    app.Error.Text = '';
    N = 2^app.N.Value;
    ntab = N;
    T = app.T.Value;
    Ts = T;
    FTF = 1/N/T; % Fundamental Table Frequency
    app.FTFLabel.Text = ['FTF = ' num2str(FTF) ' Гц'];

```

```

B = app.Delta.Value;
f = B*FTF;
app.f.Text = ['f = ' num2str(f) ' Гц'];
[A,B1] = rat(B); % КОЛ-ВО ЦИКЛОВ
niter = app.NIter.Value;
noise_in = app.Noise_in.Value;
range_noise = app.SliderNoise.Value;
noise_out = app.Noise.Value;
gbits = app.bits.Value;

niter = app.NIter.Value; %Количество итераций для CORDIC
max_noise = app.SliderNoise.Limits(2);
delta = .0025;
n = fix(max_noise/delta);
R = zeros(n,3);
app.Slider_busy_SFDR_noise.Value = 0;
niter = min([niter gbits]);
open_system('tonegen_fixpt_noise_param','loadonly');
for i = 1:n
    range_noise = i*delta;
    sim('tonegen_fixpt_noise_param')
    t = ScopeData{1}.Values.Time;
    for j = 1:3
        s = ScopeData{j}.Values.Data;
        R(i,j) = sfdr(s,f);
    end
    app.Slider_busy_SFDR_noise.Value = fix(100*i/n);
end
bdclose('tonegen_fixpt_noise_param');

```

```

plot(app.UIAxesSpectr,(1:n)*delta,R)
title(app.UIAxesSpectr, 'Зависимость SFDR от уровня шума');
xlabel(app.UIAxesSpectr, 'Уровень шума');
ylabel(app.UIAxesSpectr, 'SFDR, dB');
legend(app.UIAxesSpectr, 'LUT', 'LI', 'CORDIC');

```

```
end
```

```
% Значение Slider_busy_SFDR_noise
```

```
function Slider_busy_SFDR_noiseValueChanged(app, event)
```

```
    value = app.Slider_busy_SFDR_noise.Value;
```

```
end
```

```
% Работа кнопки HDLButton
```

```
function HDLButtonPushed(app, event)
```

```
    global gbits niter
```

```
    gbits = app.bits.Value;
```

```
    niter = app.NIter.Value;
```

```
    open_system('HDL_all');
```

```
end
```

```
% Callback
```

```
function mps_ReccPushed(app, event)
```

```
    global gbits Ts
```

```
    gbits = app.bits.Value;
```

```
    Ts = app.T.Value;
```

```
end
```

```
% Работа кнопки ExitButton
```

```
function ExitButtonPushed(app, event)
```

```
    delete(app)
```

```

end

% Заккрытие приложения: UIFigure
function UIFigureCloseRequest(app, event)
    delete(app)

end

end

% Инициализация приложения
methods (Access = private)
    % Создание UIFigure и ее компонентов
function createComponents(app)
    % Создание UIFigure
    app.UIFigure = uifigure;
    app.UIFigure.Position = [100 100 940 532];
    app.UIFigure.Name = 'UI Figure';
    app.UIFigure.CloseRequestFcn = createCallbackFcn(app,
@UIFigureCloseRequest, true);
    % Создание TabGroup
    app.TabGroup = uitabgroup(app.UIFigure);
    app.TabGroup.Position = [1 1 940 532];
    % Create Tab
    app.Tab = uitab(app.TabGroup);
    app.Tab.Title = 'Параметры';
    % Create Panel_3
    app.Panel_3 = uipanel(app.Tab);
    app.Panel_3.Title = 'Генерация кода ';
    app.Panel_3.FontSize = 14;
    app.Panel_3.Position = [527 396 211 83];
    % Создание HDLButton

```

```
app.HDLButton = uibutton(app.Panel_3, 'push');
app.HDLButton.ButtonPushedFcn = createCallbackFcn(app,
@HDLButtonPushed, true);
app.HDLButton.FontSize = 14;
app.HDLButton.Position = [18 12 178 25];
app.HDLButton.Text = 'Перенос методов в HDL';
% Создание Panel_4
app.Panel_4 = uipanel(app.Tab);
app.Panel_4.Position = [48 211 428 268];
% Создание Label_4
app.Label_4 = uilabel(app.Panel_4);
app.Label_4.VerticalAlignment = 'center';
app.Label_4.FontSize = 14;
app.Label_4.Position = [209 219 113 22];
app.Label_4.Text = 'Размер таблицы';
% Создание Label_5
app.Label_5 = uilabel(app.Panel_4);
app.Label_5.VerticalAlignment = 'center';
app.Label_5.FontSize = 14;
app.Label_5.Position = [209 167 88 22];
app.Label_5.Text = 'Разрядность';
% Создание Label_9
app.Label_9 = uilabel(app.Panel_4);
app.Label_9.FontSize = 14;
app.Label_9.Position = [209 119 85 18];
app.Label_9.Text = 'Такт опроса';
% Создание N1Label
app.N1Label = uilabel(app.Panel_4);
app.N1Label.FontSize = 14;
```

```
app.N1Label.Position = [209 24 109 18];
app.N1Label.Text = 'Шар (1..2^(N-1))';
% Создание TEditFieldLabel
app.TEditFieldLabel = uilabel(app.Panel_4);
app.TEditFieldLabel.HorizontalAlignment = 'right';
app.TEditFieldLabel.VerticalAlignment = 'center';
app.TEditFieldLabel.FontSize = 14;
app.TEditFieldLabel.Position = [40 120 25 22];
app.TEditFieldLabel.Text = 'T';
% Create T
app.T = uieditfield(app.Panel_4, 'numeric');
app.T.Limits = [0.0001 1];
app.T.FontSize = 14;
app.T.Position = [81 120 100 22];
app.T.Value = 0.01;
% Создание DeltaLabel
app.DeltaLabel = uilabel(app.Panel_4);
app.DeltaLabel.HorizontalAlignment = 'right';
app.DeltaLabel.VerticalAlignment = 'center';
app.DeltaLabel.FontSize = 14;
app.DeltaLabel.Position = [33 22 38 22];
app.DeltaLabel.Text = 'Delta';
% Создание Delta
app.Delta = uieditfield(app.Panel_4, 'numeric');
app.Delta.Limits = [1 10];
app.Delta.FontSize = 14;
app.Delta.Position = [93 22 88 22];
app.Delta.Value = 1;
% Создание NSpinnerLabel
```

```
app.NSpinnerLabel = uilabel(app.Panel_4);
app.NSpinnerLabel.HorizontalAlignment = 'right';
app.NSpinnerLabel.VerticalAlignment = 'center';
app.NSpinnerLabel.FontSize = 14;
app.NSpinnerLabel.Position = [36 219 30 22];
app.NSpinnerLabel.Text = '2^N';
% Create N
app.N = uispinner(app.Panel_4);
app.N.Limits = [4 16];
app.N.ValueDisplayFormat = '%.0f';
app.N.FontSize = 14;
app.N.Position = [81 219 100 22];
app.N.Value = 8;
% Создание bitsSpinnerLabel
app.bitsSpinnerLabel = uilabel(app.Panel_4);
app.bitsSpinnerLabel.HorizontalAlignment = 'right';
app.bitsSpinnerLabel.VerticalAlignment = 'center';
app.bitsSpinnerLabel.FontSize = 14;
app.bitsSpinnerLabel.Position = [39 169 27 22];
app.bitsSpinnerLabel.Text = 'bits';
% Create bits
app.bits = uispinner(app.Panel_4);
app.bits.Limits = [14 64];
app.bits.ValueDisplayFormat = '%.0f';
app.bits.FontSize = 14;
app.bits.Position = [81 169 100 22];
app.bits.Value = 16;
% Создание niterLabel
app.niterLabel = uilabel(app.Panel_4);
```

```
app.niterLabel.HorizontalAlignment = 'right';
app.niterLabel.FontSize = 14;
app.niterLabel.Position = [33 72 33 18];
app.niterLabel.Text = 'niter';
% Create NIter
app.NIter = uispinner(app.Panel_4);
app.NIter.Limits = [6 25];
app.NIter.FontSize = 14;
app.NIter.Position = [81 71 100 22];
app.NIter.Value = 11;
% Create CORDICLabel
app.CORDICLabel = uilabel(app.Panel_4);
app.CORDICLabel.FontSize = 14;
app.CORDICLabel.Position = [209 71 210 18];
app.CORDICLabel.Text = 'Количество итераций CORDIC ';
% Create ExitButton
app.ExitButton = uibutton(app.Tab, 'push');
app.ExitButton.ButtonPushedFcn = createCallbackFcn(app,
@ExitButtonPushed, true);
app.ExitButton.FontSize = 14;
app.ExitButton.Position = [768 55 100 25];
app.ExitButton.Text = {'Exit'; ''};
% Create Tab_3
app.Tab_3 = uitab(app.TabGroup);
app.Tab_3.Title = 'Тест';
% Create Label
app.Label = uilabel(app.Tab_3);
app.Label.HorizontalAlignment = 'right';
app.Label.VerticalAlignment = 'center';
```

```
app.Label.FontSize = 14;
app.Label.Position = [10 450 100 22];
app.Label.Text = 'Выбор метода';
% Create TipOut
app.TipOut = uidropdown(app.Tab_3);
app.TipOut.Items = {'Direct', 'Linear', 'FixptLinear', 'Reccurent', 'CORDIC'};
app.TipOut.FontSize = 14;
app.TipOut.Position = [125 450 100 22];
app.TipOut.Value = 'Direct';
% Create Noise
app.Noise = ucheckbox(app.Tab_3);
app.Noise.Text = 'Шум на выходе преобразователя';
app.Noise.FontSize = 14;
app.Noise.Position = [10 413 247 22];
% Create Label_3
app.Label_3 = uilabel(app.Tab_3);
app.Label_3.HorizontalAlignment = 'right';
app.Label_3.VerticalAlignment = 'center';
app.Label_3.FontSize = 14;
app.Label_3.Position = [10 338 100 22];
app.Label_3.Text = 'Уровень шума';
% Create SliderNoise
app.SliderNoise = uislider(app.Tab_3);
app.SliderNoise.Limits = [0 0.1];
app.SliderNoise.Position = [131 347 150 3];
% Create UIAxesSpectr
app.UIAxesSpectr = uiaxes(app.Tab_3);
xlabel(app.UIAxesSpectr, 'f (Hz)')
ylabel(app.UIAxesSpectr, 'P(f) , dB')
```

```

app.UIAxesSpectr.PlotBoxAspectRatio = [1 0.344051446945338
0.344051446945338];
app.UIAxesSpectr.Position = [23 22 881 249];
% Create TestTHDButton
app.TestTHDButton = uibutton(app.Tab_3, 'push');
app.TestTHDButton.ButtonPushedFcn = createCallbackFcn(app,
@TestTHDButtonPushed, true);
app.TestTHDButton.FontSize = 14;
app.TestTHDButton.Position = [355 447 100 25];
app.TestTHDButton.Text = 'Test THD';
% Create Label_6
app.Label_6 = uilabel(app.Tab_3);
app.Label_6.VerticalAlignment = 'center';
app.Label_6.FontSize = 14;
app.Label_6.FontWeight = 'bold';
app.Label_6.Position = [10 471 326 37];
app.Label_6.Text = 'Тестирование гармоничности сигнала';
% Create TestSFDRMatLabdoubleButton
app.TestSFDRMatLabdoubleButton = uibutton(app.Tab_3, 'push');
app.TestSFDRMatLabdoubleButton.ButtonPushedFcn = createCallbackFcn(app,
@TestSFDRMatLabdoubleButtonPushed, true);
app.TestSFDRMatLabdoubleButton.FontSize = 14;
app.TestSFDRMatLabdoubleButton.Position = [261 412 194 25];
app.TestSFDRMatLabdoubleButton.Text = 'Test SFDR(MatLab, double)';
% Create TestSFDRSimulinkmodelButton
app.TestSFDRSimulinkmodelButton = uibutton(app.Tab_3, 'push');
app.TestSFDRSimulinkmodelButton.ButtonPushedFcn = createCallback-
Fcn(app, @TestSFDRSimulinkmodelButtonPushed, true);
app.TestSFDRSimulinkmodelButton.FontSize = 14;

```

```
app.TestSFDRSimulinkmodelButton.Position = [260 374 195 25];
app.TestSFDRSimulinkmodelButton.Text = 'Test SFDR (Simulink model)';
% Create Error
app.Error = uilabel(app.Tab_3);
app.Error.FontSize = 14;
app.Error.FontWeight = 'bold';
app.Error.FontColor = [1 0 0];
app.Error.Position = [564 300 212 18];
app.Error.Text = "";
% Create Noise_in
app.Noise_in = uicheckbox(app.Tab_3);
app.Noise_in.Text = 'Шум на входе';
app.Noise_in.FontSize = 14;
app.Noise_in.Position = [10 375 117 22];
% Create SimulinkButton
app.SimulinkButton = uibutton(app.Tab_3, 'push');
app.SimulinkButton.ButtonPushedFcn = createCallbackFcn(app,
@SimulinkButtonPushed, true);
app.SimulinkButton.FontSize = 14;
app.SimulinkButton.Position = [355 334 100 25];
app.SimulinkButton.Text = 'Simulink';
% Create Panel
app.Panel = uipanel(app.Tab_3);
app.Panel.Title = 'Результаты';
app.Panel.FontSize = 14;
app.Panel.Position = [468 325 260 147];
% Create SFDRLabel
app.SFDRLabel = uilabel(app.Panel);
app.SFDRLabel.Position = [20 10 208 17];
```

```
app.SFDRLabel.Text = 'SFDR';
% Create FTFLabel
app.FTFLabel = uilabel(app.Panel);
app.FTFLabel.VerticalAlignment = 'center';
app.FTFLabel.Position = [20 62 193 22];
app.FTFLabel.Text = 'FTF';
% Create THD
app.THDL = uilabel(app.Panel);
app.THDL.VerticalAlignment = 'center';
app.THDL.Position = [20 36 193 22];
app.THDL.Text = 'THD';
% Create f
app.f = uilabel(app.Panel);
app.f.VerticalAlignment = 'center';
app.f.Position = [20 95 193 22];
app.f.Text = 'f';
% Create Panel_2
app.Panel_2 = uipanel(app.Tab_3);
app.Panel_2.Title = 'Пакетный режим';
app.Panel_2.FontSize = 14;
app.Panel_2.Position = [742 323 186 148];
% Create Label_11
app.Label_11 = uilabel(app.Panel_2);
app.Label_11.HorizontalAlignment = 'right';
app.Label_11.FontSize = 14;
app.Label_11.Position = [43 37 88 18];
app.Label_11.Text = 'Выполнение';
% Create Slider_busy_SFDR_noise
app.Slider_busy_SFDR_noise = uislider(app.Panel_2);
```

```

app.Slider_busy_SFDR_noise.MajorTicks = [];
app.Slider_busy_SFDR_noise.MajorTickLabels = {};
app.Slider_busy_SFDR_noise.ValueChangedFcn = createCallbackFcn(app,
@Slider_busy_SFDR_noiseValueChanged, true);
app.Slider_busy_SFDR_noise.MinorTicks = [];
app.Slider_busy_SFDR_noise.Enable = 'off';
app.Slider_busy_SFDR_noise.FontSize = 14;
app.Slider_busy_SFDR_noise.Position = [39 23 107 3];
% Create SFDRnoiseFixPtButton
app.SFDRnoiseFixPtButton = uibutton(app.Panel_2, 'push');
app.SFDRnoiseFixPtButton.ButtonPushedFcn = createCallbackFcn(app,
@SFDRnoiseFixPtButtonPushed, true);
app.SFDRnoiseFixPtButton.FontSize = 14;
app.SFDRnoiseFixPtButton.Position = [24 82 137 25];
app.SFDRnoiseFixPtButton.Text = 'SFDR/noise (FixPt)';
% Create LUTvsLITab
app.LUTvsLITab = uitab(app.TabGroup);
app.LUTvsLITab.Title = 'LUT vs LI';
% Create LUTLIN10Label
app.LUTLIN10Label = uilabel(app.LUTvsLITab);
app.LUTLIN10Label.VerticalAlignment = 'center';
app.LUTLIN10Label.FontSize = 14;
app.LUTLIN10Label.FontWeight = 'bold';
app.LUTLIN10Label.Position = [90 469 730 22];
app.LUTLIN10Label.Text = 'Сравнение таблицы с прямым доступом(LUT)
и линейной интерполяции (LI). Рекомендуемое  $N \leq 10$ ';
% Create UIAxes_LUTvsLI
app.UIAxes_LUTvsLI = uiaxes(app.LUTvsLITab);
xlabel(app.UIAxes_LUTvsLI, 'X')

```

```
ylabel(app.UIAxes_LUTvsLI, 'Y')
app.UIAxes_LUTvsLI.PlotBoxAspectRatio = [1 0.566787003610108
0.566787003610108];
app.UIAxes_LUTvsLI.Position = [11 57 888 392];
% Create Button
app.Button = uibutton(app.LUTvsLITab, 'push');
app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);
app.Button.FontSize = 14;
app.Button.Position = [251 13 100 25];
app.Button.Text = 'График';
% Create Label_10
app.Label_10 = uilabel(app.LUTvsLITab);
app.Label_10.HorizontalAlignment = 'right';
app.Label_10.FontSize = 14;
app.Label_10.Position = [504 17 88 18];
app.Label_10.Text = 'Выполнение';
% Create Slider_busy
app.Slider_busy = uislider(app.LUTvsLITab);
app.Slider_busy.MajorTicks = [];
app.Slider_busy.MajorTickLabels = {};
app.Slider_busy.MinorTicks = [];
app.Slider_busy.FontSize = 14;
app.Slider_busy.Position = [613 26 150 3];
% Create Tab_2
app.Tab_2 = uitab(app.TabGroup);
app.Tab_2.Title = 'Точность';
% Create Label_7
app.Label_7 = uilabel(app.Tab_2);
app.Label_7.VerticalAlignment = 'center';
```

```

app.Label_7.FontSize = 14;
app.Label_7.FontWeight = 'bold';
app.Label_7.Position = [184 462 560 22];
app.Label_7.Text = 'Точность аппроксимации (разность истинного и сгенерированного сигнала)';
% Create UIAxes
app.UIAxes = uiaxes(app.Tab_2);
xlabel(app.UIAxes, 'Время(с.)')
ylabel(app.UIAxes, 'Ошибка по амплитуде')
app.UIAxes.PlotBoxAspectRatio = [1 0.457326892109501 0.457326892109501];
app.UIAxes.FontSize = 14;
app.UIAxes.Position = [32 65 863 381];
% Create DoubleButton
app.DoubleButton = uibutton(app.Tab_2, 'push');
app.DoubleButton.ButtonPushedFcn = createCallbackFcn(app, @DoubleButtonPushed, true);
app.DoubleButton.FontSize = 14;
app.DoubleButton.Position = [442 19 100 25];
app.DoubleButton.Text = 'Double';
% Create Label_8
app.Label_8 = uilabel(app.Tab_2);
app.Label_8.HorizontalAlignment = 'right';
app.Label_8.VerticalAlignment = 'center';
app.Label_8.FontSize = 14;
app.Label_8.Position = [158 22 30 22];
app.Label_8.Text = 'Тип';
% Create DropDown
app.DropDown = uidropdown(app.Tab_2);

```

```

app.DropDown.Items = {'LUT', 'LI', 'CORDIC', 'LUT8'};
app.DropDown.ValueChangedFcn = createCallbackFcn(app,
@DropDownValueChanged, true);
app.DropDown.FontSize = 14;
app.DropDown.Position = [203 22 100 22];
app.DropDown.Value = 'LUT';
% Create FixedButton
app.FixedButton = uibutton(app.Tab_2, 'push');
app.FixedButton.ButtonPushedFcn = createCallbackFcn(app,
@FixedButtonPushed, true);
app.FixedButton.FontSize = 14;
app.FixedButton.Position = [654 19 100 25];
app.FixedButton.Text = 'Fixed';
end
end
methods (Access = public)
% Construct app
function app = app12017b
% Create and configure components
createComponents(app)
% Register the app with App Designer
registerApp(app, app.UIFigure)
if nargin == 0
clear app
end
end
% Code that executes before app deletion
function delete(app)
% Delete UIFigure when app is deleted

```

```
delete(app.UIFigure)  
end  
end  
end
```

					27.03.04.2018.293 ПЗ	Лист
Изм						77
	Лист	№ докум.	Подпись	Дата		