

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторной работы
**«Формирование сигналов и их преобразование при цифровой
обработке»**
по дисциплине «Цифровая обработка сигналов»
для студентов всех форм обучения
специальности «Электронные системы»

Утверждено
редакционно-издательским
советом университета,
протокол № 1 от 04.06.2014 г.

Харьков
НТУ «ХПИ»
2015

Методические указания к выполнению лабораторной работы «Формирование сигналов и их преобразование при цифровой обработке» по дисциплине «Цифровая обработка сигналов» для студентов всех форм обучения специальности «Электронные системы» /Состав. Л. В. Фетюхина, О. А. Бутова. – Харьков: НТУ «ХПИ», 2015. – 41 с.

Составители Л. В. Фетюхина,
 О. А. Бутова

Рецензент В. В. Замаруев

Кафедра промышленной и биомедицинской электроники

ВСТУПЛЕНИЕ

В настоящее время к общепризнанным стандартам в области компьютерных технологий относится система MATLAB, предназначенная для компьютерного моделирования в самых разных областях науки и техники. К основным достоинствам программной среды MATLAB относятся: алгоритмический язык с матричной обработкой данных; большая библиотека стандартных функций; огромное разнообразие графических средств; удобные средства создания и отладки программ; широкий набор программных средств (Toolbox), сгруппированных по специализированным приложениям, средств графического интерфейса пользователя без использования алгоритмического языка в явном виде и средств пакета Simulink общего и специального назначения для блочного моделирования динамических систем.

Актуальная проблема подготовки современных научно-технических специалистов, владеющих навыками моделирования цифровой обработки сигналов (ЦОС) в MATLAB, требует обеспечения учебного процесса необходимой методической литературой. Базовая подготовка по ЦОС подразумевает изучение теории ЦОС, технологий компьютерного моделирования и реализации алгоритмов ЦОС на современной цифровой элементной базе.

В данной лабораторной работе студенты получают навыки формирования последовательностей типовых временных сигналов для дальнейшей их обработки спроектированными фильтрами в пакете Signal Processing Toolbox и Simulink.

ЛАБОРАТОРНАЯ РАБОТА

ФОРМИРОВАНИЕ СИГНАЛОВ И ИХ ПРЕОБРАЗОВАНИЕ ПРИ ЦИФРОВОЙ ОБРАБОТКЕ

Цель: формирование типовых сигналов с заданными параметрами, используя основные возможности среды MATLAB (пакетов SIGNAL PROCESSING TOOLBOX & SIMULINK).

НЕОБХОДИМЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ. ФУНКЦИИ ГЕНЕРАЦИИ СИГНАЛОВ

Сигнал (от лат. *signum* – знак) – физический процесс, который осуществляет перенос информации во времени и пространстве. Физическая природа сигнала различна. Часто это напряжение (ток), но могут быть и другие величины. Сигнал обладает различными параметрами и характеристиками и может быть представлен математической моделью. Как правило, математическая модель упрощает сигнал, поскольку не все свойства сигнала существенны при решении той или иной задачи. В рамках данной лабораторной работы сигнал будет рассматриваться как функция времени, хотя возможны варианты (например, зависимость светового потока от пространственных координат).

В зависимости от того известен ли сигнал точно, различают детерминированные и случайные сигналы.

Детерминированный сигнал отличается тем, что закон изменения во времени известен, а следовательно, и математическая модель не содержит неизвестных параметров. Способы задания таких сигналов: математическая формула; вычислительный алгоритм; словесное описание.

Строго говоря, детерминированных сигналов в природе не существует, так как существуют хаотические тепловые явления, взаимосвязь с окружающей средой, неполнота знаний об объекте и т.п.

Квазидетерминированным сигналом называется сигнал с частично известным законом изменения во времени, то есть с одним или несколькими случайными параметрами.

Случайный сигнал имеет случайный характер изменения во времени и может быть описан с помощью статистических характеристик (спектральная плотность закона распределения). Его нельзя представить детерминированной функцией, поскольку конкретные значения сигнал принимает с некоторой вероятностью.

В теории и практике обработки сигналов часто встречаются *периодические* сигналы, для которых выполняется соотношение:

$$S(t + nT) = s(t)$$

при любом t , где n – произвольное число; T – период. Величина обратная периоду, – частота f , часто используется понятие круговой частоты $\omega = 2\pi f$ (рад/сек). Периодические сигналы имеют бесконечную энергию. Кроме синусоиды к таким периодическим сигналам относят часто встречающиеся на практике прямоугольный, пилообразный и треугольный сигнал.

Сигналы, концентрирующие энергию в коротком интервале времени, называются *импульсом*. Сигналы, исчезающие в течение длительного времени при ограниченной энергии источника, называются *затухающими*. Это *финитные* сигналы, которые относят к *непериодическим* или *одиночным*.

Сигналы описываются общими математическими моделями. В общем случае эта модель определяется математическими зависимостями:

$$S = f(t, a, b, c...),$$

где t – основной независимый аргумент, a, b, c – параметры сигнала. Модели сигналов в зависимости от основного независимого аргумента бывают временными (аргумент t) и частотными (аргумент ω). Частотная область удобна при изображении частотного состава сигналов. Применяются также векторные модели. Вид модели выбирается из конкретных условий.

Аналоговый сигнал с математической точки зрения представляет собой *функцию*, и при его дискретизации получаются отсчеты, которые являются значениями этой функции, взятые в отдельные дискретные моменты времени. Дискретный сигнал – это по существу результат преобразования аналогового сигнала в последовательность чисел, которые являются значениями сигнала в отдельные (дискретные) моменты времени, называются отсчетами. Как правило, отсчеты берутся через равные промежутки времени t_s , называемые периодом дискретизации. Величина, обратная периоду дискретизации, называется частотой дискретизации: $f_s = 1/t_s$. Круговая частота определяет-

ся как $\omega_s = 2\pi/t_s$. Для того чтобы гармонический сигнал мог быть адекватно представлен дискретными отсчетами, его частота не должна превышать половины частоты дискретизации: $f_N = f_s/2$ (частота Найквиста).

В пакете SIGNAL PROCESSING TOOLBOX, который работает в среде MATLAB, разработанной компанией MathWorkInc., имеется целый ряд функций, предназначенных для генерации сигналов стандартной формы, часто встречающихся при решении различных задач обработки сигнала.

Генерация периодических сигналов

Функции, относящиеся к данной группе, задают в виде тех же параметров (вектор моментов времени и дополнительные аргументы, описывающие параметры формируемого импульса). Период формируемых сигналов равен 2π . Для формирования сигналов с иным периодом необходимо соответствующим образом масштабировать передаваемый функции временной аргумент. Возвращаемым результатом является вектор отсчетов результирующего сигнала. Имеются функции для генерации периодических сигналов следующей формы:

Синтаксис: $y = \text{square}(t, \text{duty})$.

Описание: *square* – генерация периодической последовательности прямоугольных импульсов. Дополнительным параметром является коэффициент заполнения импульсов *duty*. Он определяется как отношение длительности импульса к периоду их следования в процентах. По умолчанию этот параметр равен 50, то есть это меандр. Последовательность является двухполярной, то есть диапазон равен ± 1 .

Синтаксис: $y = \text{sawtooth}(t)$ или $y = \text{sawtooth}(t, \text{width})$.

Описание: Функция $y = \text{sawtooth}(t)$ формирует пилообразный сигнал с амплитудой ± 1 и периодом 2π . Функция $y = \text{sawtooth}(t, \text{width})$ формирует модифицированный пилообразный сигнал. Параметр *width* задается в диапазоне от 0 до 1 и определяет часть периода, в котором возрастает сигнал. Сигнал возрастает от -1 до 1 на интервале от 0 до $2\pi \text{width}$, а затем убывает от 1 до -1 на интервале от $2\pi \text{width}$ до 2π . Если $\text{width} = 0.5$, то формируется симметричное пилообразное колебание. Функция $\text{sawtooth}(t, l)$ эквивалентна функции $\text{sawtooth}(t)$.

Синтаксис: $y = \text{dirict}(t, n)$.

Описание: *diric* – функция Дирихле или периодическая *sinc* – функция. Дополнительным параметром является целочисленный порядок функции n . Функция Дирихле рассчитывается по формуле
$$diric(x) = \frac{\sin(nx/2)}{(n \cdot \sin(x/2))}$$
. Параметр n должен быть целым положительным числом.

Формирование колебаний, состоящих из конечного числа гармоник (полигармонических сигналов), можно осуществить при помощи обычных процедур $\sin(x)$ и $\cos(x)$.

Синтаксис: $y = \sin(x)$ или $y = \cos(x)$.

Описание: Период формируемых сигналов равен 2π .

Генерация колебаний с изменяющейся частотой

Синтаксис: $y = chirp(t, f0, t1, f1, 'method', phi)$.

Описание: Функция *chirp* генерирует колебания, мгновенная частота которых изменяется по одному из трех возможных законов – линейному, квадратичному или экспоненциальному. Числовые параметры $f0$, $t1$, $f1$ определяют узловые точки расчета. В нулевой момент времени частота равняется $f0$, а в момент $t1$ частота будет равна $f1$. Переменная *'method'* определяет тип зависимости частоты от времени: *'linear'*, *'quadratic'*, *'logarithmic'*. По умолчанию $f0=0$, $t1=1$, $f1=100$, $phi=0$ и тип изменения *'linear'*. Звук такого сигнала напоминает визг – откуда и его название.

Более широкими возможностями обладает функция *vco* (*Voltage Controlled Oscillator* – генератор, управляемый напряжением), которая позволяет формировать колебания с произвольным законом изменения мгновенной частоты. Можно сказать, что данная функция осуществляет частотную модуляцию.

Синтаксис: $y = vco(x, fc, fs)$, или $y = vco(x, [F \min F \max], fs)$.

Описание: $vco(x, fc, fs)$ генерирует сигнал, мгновенная частота которого меняется по закону, заданному отсчетами из входного вещественного вектора или матрицы x . Частота дискретизации отсчетов x задается входным параметром fs . Входной параметр fc задает несущую (опорную) частоту в герцах; при $x = 0$ созданный сигнал y представляет собой гармоническое колебание с частотой fc Гц и единичной амплитудой, дискретизированное с

частотой f_s Гц. Значения элементов x должны лежать в диапазоне от -1 до 1 , при этом:

значение $x = -1$ дает сигнал с нулевой частотой;

значение $x = 0$ дает сигнал с частотой f_c ;

значение $x = 1$ дает сигнал с частотой $2 \cdot f_c$.

В данном варианте синтаксиса $y = vco(x, [F \min F \max], f_s)$ диапазон изменения мгновенной частоты масштабируется таким образом, что значения ± 1 из вектора x соответствуют частотам F_{min} и F_{max} Гц соответственно. Для получения наилучших результатов значения F_{min} и F_{max} должны лежать в диапазоне от 0 до $f_s/2$.

Генерация неперiodических (одиночных) сигналов

Все функции генерации неперiodических сигналов задают в виде параметров (вектор моментов времени и дополнительные аргументы, описывающие параметры формируемого импульса). Возвращаемым результатом является вектор отсчетов результирующего сигнала. Имеются функции для генерации сигналов следующей формы:

Синтаксис: $y = rectpuls(t)$, или $y = rectpuls(t, w)$.

Описание: $rectpuls$ – генерация одиночного прямоугольного импульса, единственным дополнительным параметром является длительность импульса. Обращение вида $y = rectpuls(t)$ позволяет образовать вектор y значений сигнала такого импульса единичной амплитуды, шириной w , центрированного относительно $t = 0$ по заданному вектору t . Если ширина импульса w не указана, ее значение по умолчанию принимается равным единице.

Синтаксис: $y = tripuls(t, w, as)$.

Описание: $tripuls$ – генерация одиночного треугольного импульса, дополнительными параметрами являются длительность импульса и коэффициент его асимметрии. Аргументы y , t и w имеют тот же смысл. Аргумент as определяет наклон боковых сторон треугольника и изменяется в диапазоне $-1 < a < 1$. Если $as = 0$ или не указан, треугольный импульс имеет симметричную форму.

Синтаксис: $y = sinc(x)$.

Описание: $\text{sinc}(x)$ – генерация импульса $\sin(x)/x$ (от лат. *sinus cardinalis* – «кардинальный синус»). Эта функция представляет собой обратное непрерывное преобразование Фурье от прямоугольного импульса шириной 2 и высотой 1: Дополнительных параметров данная функция не имеет. Последовательность отсчетов аргумента задается в массиве x . Последовательность значений функции sinc формируется в векторе y .

Синтаксис: $y = \text{gauspuls}(t, fc, bw)$.

$y = \text{gauspuls}(t, fc, bw)$, или $y = \text{gauspuls}(t, fc, bw, bwr)$.

$[y, yq] = \text{gauspuls}(\dots)$, $[y, yq, ye] = \text{gauspuls}(\dots)$.

$tc = \text{gauspuls}('cutoff', fc, bw, bwr, tpe)$.

Описание: Группа функций gauspuls генерирует синусоидальный сигнал, модулированный функцией Гаусса. Функция $y = \text{gauspuls}(t, fc, bw)$ возвращает последовательность отсчетов сигнала, вычисленных в моменты времени, заданные в векторе t ; fc определяет частоту синусоиды (Гц), а bw – относительную ширину полосы частот сигнала. По умолчанию $fc = 1000$ Гц, а $bw = 0.5$.

Функция $y = \text{gauspuls}(t, fc, bw, bwr)$ возвращает последовательность отсчетов сигнала, имеющего следующие параметры: амплитуда равна 1, ширина полосы частот bw равна 0.5, границы полосы частот определяются уровнем затухания bwr (в децибелах) по отношению к нормализованной амплитуде сигнала. Параметр bwr должен быть отрицательным. Если он не указан, то по умолчанию $bwr = -6$ дБ. Функции $[y, yq] = \text{gauspuls}(\dots)$ возвращают два вектора: y и yq . Вектор y содержит отсчеты исходного сигнала, а вектор yq – отсчеты сигнала, синусоида в котором сдвинута по фазе на 90° .

Функция $[y, yq, ye] = \text{gauspuls}(\dots)$ дополнительно возвращают огибающую сигнала. Функция $tc = \text{gauspuls}('cutoff', fc, bw, bwr, tpe)$ вычисляет время отсечения tc (больше или равно 0); tc соответствует моменту времени, в который амплитуда огибающей сигнала снижается до tpe дБ. Величина параметра tpe должна быть отрицательной, поскольку он определяет относительный уровень сигнала, меньший чем пиковое значение (равное 1). По умолчанию $tpe = -60$ дБ.

Синтаксис: $y = \text{gmonopuls}(t, fc)$.

$tc = \text{gmonopuls}('cutoff', fc)$.

Описание: $gmonopuls(t,fc)$ возвращает отсчеты гауссова моноимпульса с единичной амплитудой, рассчитанные в моменты времени, заданные входным вектором t . Дополнительным параметром является несущая частота генерируемого сигнала (Гц), которая задается входным параметром fc . По умолчанию $fc = 1000$ Гц.

При использовании $tc = gmonopuls('cutoff', fc)$ производится только определение расстояния по времени между точками, где моноимпульс принимает минимальное и максимальное значения. Значения по умолчанию используются, если параметр задан в виде пустой матрицы [] или если при вызове функции один или несколько последних входных параметров отсутствуют.

Генерация последовательности импульсов

Функция $pulstran$ служит для генерации конечной последовательности импульсов одинаковой формы с произвольно задаваемыми задержками и амплитудными множителями. Форма импульсов может задаваться одним из двух способов: именем функции, генерирующей импульс, либо уже рассчитанным вектором отсчетов.

Синтаксис: $y = pulstran(t,d,'func',p1,p2,...)$.

Описание: Функция $y = pulstran(t,d,'func',p1,p2,...)$ определяет d – вектор значений тех моментов времени, где должны быть центры соответствующих импульсов; параметр $'func'$ определяет форму импульсов. Он может иметь одно из следующих значений: $rectpuls$ (для прямоугольного импульса), $tripuls$ (для треугольного импульса) и $gauspuls$ (для гауссового импульса); параметры $p1, p2, \dots$ определяют необходимые параметры импульса в соответствии с формой обращения к процедуре, определяющей этот импульс. Функция $y = pulstran(t,d,p,Fs)$ позволяет определять импульс последовательностью отсчетов, заданных в векторе p . Частота дискретизации задается параметром Fs . По умолчанию частота дискретизации полагается равной 1 Гц.

Выходной сигнал y рассчитывается для значений аргумента, заданных в векторе t , по формуле $y = func(t - d(1)) + func(t - d(2)) + \dots$. Число импульсов в заданном диапазоне значений аргумента равно $length(d)$.

ПРАКТИЧЕСКАЯ ЧАСТЬ 1

1.1. ПРИМЕРЫ СОЗДАНИЯ СИГНАЛОВ

Далее приведены примеры генерирования некоторых сигналов, которые студентам рекомендуется выполнить в командной строке пакета SIGNAL PROCESSING TOOLBOX.

1) *Прямоугольный импульс*

В пакете сигнал задаётся в виде вектора, сопоставленного с вектором моментов времени. Перед вводом модели сигнала нужно указать частоту дискретизации и сформировать вектор-столбец моментов времени. Например, командную строку можно записать так:

```
>> fs = 1e3; t=-1:1/fs:1; t=t';
```

В данном случае введена частота дискретизации fs 1 кГц. Сигнал будет задан на интервале времени 1 с. Это соответствует 1001 отсчёту. Последний оператор означает преобразование вектора-строки в вектор-столбец (апостроф ' означает операцию транспонирования матрицы). Не следует забывать ставить точку с запятой в конце каждого оператора, чтобы подавить вывод значений на экран монитора. При вводе этого оператора либо нужно предварительно задать значения амплитуды A и длительности tau , либо в самом операторе вместо идентификаторов A и tau поставить численные значения.

Нужный импульс будет получен как:

```
>> y=A*rectpuls (t, tau);
```

Для вывода графиков может быть использована команда *plot*. Команда *plot(y)* изображает вектор y как функцию индекса. Задав вектор времени t , фактически получается график аналогового сигнала, который получен путем линейной интерполяции.

Возможно задание цвета графика, осевых пределов, используемых символов (сплошная черта, точка, пунктир и т.д.), а также текстовых надписей. Например, изображение графика зависимости $y(t)$ красным цветом 'r' (red) с соответствующими надписями по осям и сеткой.

```
>> plot(t,y,'r'); title('График функции RECTPULS');
>> xlim([-1 1]);
>> ylim([-0.1 1.5]);
>> xlabel('Время ( с )');
>> ylabel('Прямоугольный сигнал y(t)');
>> grid
```

2) Треугольный импульс

```
>> y= A * tripuls (t, tau);
```

3) Экспоненциальный импульс

```
>> y = A * exp (- t / tau);
```

Подразумевается, что вектор t задан для моментов времени $t \geq 0$.

4) Синусоидальный импульс

```
>> y = A * sin (pi * t / tau) .* (t>=0) .* (t<= tau);
```

Здесь используется тот факт, что операции сравнения возвращают 1, если неравенство выполняется, или 0 в противном случае. Обратите внимание, что операция умножения представлена здесь как `.*` (точка перед знаком `*`). Это означает поэлементное умножение векторов (в противном случае производилась бы операция матричного умножения). В тех случаях, когда осуществляется умножение скаляров или матрицы (вектора) на скаляр, можно использовать просто символ `*`. То же самое относится к операции деления и возведения в степень. Поэлементное деление матриц задаётся оператором `./`, поэлементное возведение в степень `.^`. Число π задаётся в MATLAB как `pi`.

5) Радиои импульсы

Модулированные импульсы получаются при умножении импульса s на гармоническое колебание. Предварительно нужно задать значение несущей частоты f_0 и начальной фазы ϕ .

```
>> s = A * sin (pi * t / tau) .* (t>=0) .* (t<= tau);
>> sr = s .* cos (2*pi*f0*t + phi);
```

б) Последовательности импульсов

Для генерации конечной последовательности импульсов одинаковой формы с произвольно задаваемыми задержками и амплитудами используется функция *pulstran*. Например, нужно задать следующую последовательность прямоугольных импульсов:

```
>> Fs= 1e3; t= -0.1:1/Fs:1; t= t';  
>> tau= 0.1;  
>> d = [0.05 0.25 0.45 0.65 0.85 ; 2 1.5 3 1 2.5]' ;  
>> y= pulstran (t, d, 'rectpuls', tau);
```

Так же этот процесс, состоящий из трех последовательных прямоугольных импульсов разной высоты и ширины, можно получить по такой последовательности команд

```
>> t = 0 : 0.01 : 10;  
>> s = 0.75*rectpuls(t-2, 1.2)+0.5*rectpuls(t-8, 0.4)  
+1.35*rectpuls(t-5, 0.8); plot(t,s, 'b');
```

или последовательность треугольных импульсов

```
>> y=0.75*tripuls(t-2,0.5)+0.5*tripuls(t-8,0.5,-1)  
+1.35*tripuls(t-5,0.8,1); plot(t,y, 'b');
```

Если нужно построить последовательность импульсов произвольной формы, причём отсчёты одиночного импульса записаны в векторе *s1*, то используют следующую форму задания функции *pulstran*:

```
>> s= pulstran(t, d, s1, Fs);
```

Например, нужно задать последовательность (пачку) из четырёх синусоидальных импульсов:

```
>> Fs=1e4;  
>> t=0:1/Fs:2e-2;t=t';  
>> tau=2e-3;A=2;  
>> s1=sin(pi*t/tau) .* (t<=tau);  
>> d =[(0:3)'*5e-3;A*ones(4,1)];  
>> s= pulstran(t, d, s1, Fs);  
>> plot(t,s, 'b');
```

Функция *sinc* является обратным преобразованием Фурье прямоугольного импульса шириной 2π и ее максимальное значение равно единице. Пример использования функции, сдвинутой относительно начала координат, приведен ниже.

```
>> Fs=1e4;
>> t=10:1/Fs:40;
>> s2=0.7*sinc(pi*(t-25)/5);
>> plot(t,s2);
```

Рассмотрим применение функций *pulstran* и *sinc* для восстановления аналогового сигнала по его дискретным отсчетам.

Обратите внимание на команду *stem*, позволяющую получить график в виде «стебельков». Команда *hold on* включает режим сохранения текущего содержимого координатных осей при выводе в них новых графиков. Для выключения этого режима используют команду *hold off*.

```
>> t=-5:0.1:10;           % время для аналогового сигнала
>> k=0:5;                 % номера отсчетов дискретного сигнала
>> sd=[2 2 3 3 2.5 2.5]; % дискретный сигнал
>> sa=pulstran(t,[k' sd'],'sinc'); %восстановленный сигнал
>> stem(k, sd)            % график дискретного сигнала
>> hold on                % в виде «стебельков»
>> plot(t, sa, 'r')      % график аналогового сигнала
>> hold off
```

В результате получим сигнал, восстановленный по заданным дискретным значениям (рис. 1):

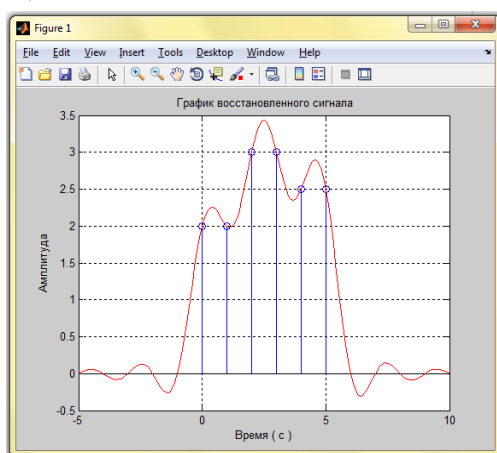


Рисунок 1

7) Последовательности периодических сигналов

Приведем пример использования функций *square* и *sawtooth*. Для вывода нескольких графиков в разных областях одного окна используют команду *subplot*, имеющую следующий синтаксис:

```
>> subplot (Row, Columns, N) ;
```

Графическое окно будет разбито на клетки, где *Row* – количество строк; *Columns* – количество столбцов; *N* – номер текущей клетки (счет ведется по строкам). Приведенные ниже команды осуществляют одновременный вывод трех сигналов в матрице (2:2).

```
>> t=0:0.01:30;           % задание временного интервала
>> x=square(t);           % определение прямоугольного сигнала
>> y=sawtooth(t,0.5);     % определение треугольного сигнала
>> z=sawtooth(t,1);       % определение пилообразного сигнала
>> subplot(2,2,1); plot(t,x);
>> subplot(2,2,2); plot(t,y);
>> subplot(2,2,3); plot(t,z);
```

В результате получаем процесс, изображенный на рис. 2.

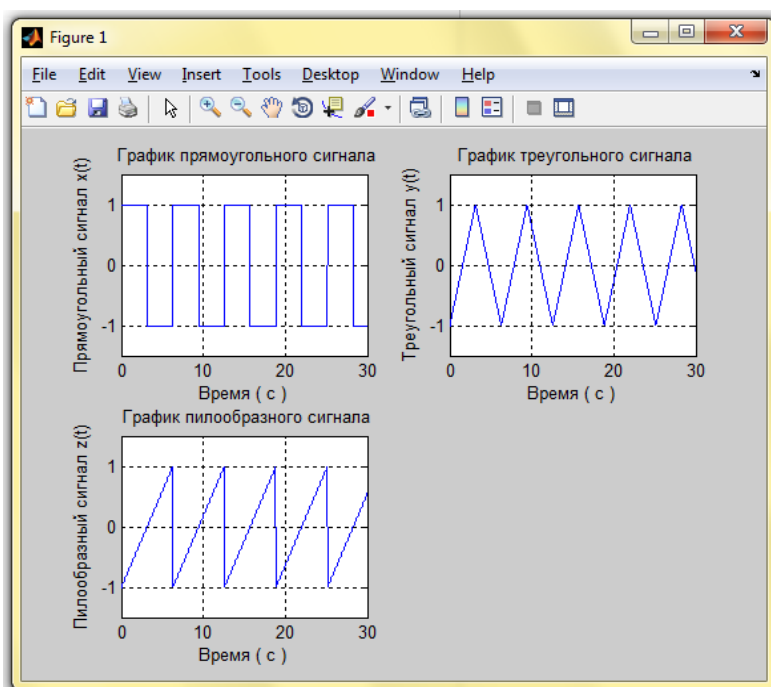


Рисунок 2

8) Последовательности периодических сигналов с шумом

Для генерации случайных чисел служат команды *rand* (равномерное распределение) и *randn* (нормальное распределение). При обращении к функции *randn* формируется массив из *m* строк и *n* столбцов, состоящий из псевдослучайных чисел, имеющих нормальное распределение, с нулевым средним значением и единичным стандартным отклонением. В задачах, связанных с обработкой сигналов, иногда используют функцию *wgn* (*white Gausse noise*).

В качестве примера рассмотрим такую задачу. Допустим имеется некоторый «полезный» сигнал, имеющий синусоидальную форму с известным периодом $T_1 = 1$ с и амплитудой $A_1 = 0.75$. Сформируем этот сигнал как вектор его значений в дискретные моменты времени с $T_s = 0.001$ с:

```
>> Ts=0.001; t=0:Ts:20; A1=0.75; T1=1;  
>> Yp=A1*sin(2*pi*t/T1);  
>> plot(t(10002:end), Yp(10002:end))
```

Допустим, что вследствие прохождения через датчик к полезному сигналу добавился шум в виде более высокочастотной синусоиды с периодом $T_2 = 0.2$ с и амплитудой $A_2 = 5$, а в результате измерения к нему еще добавился белый гауссовый шум измерителя с интенсивностью $A_{sh} = 5$.

В результате создался такой сигнал $x(t)$ (рис. 3):

```
>> T2=0.2; A2=10; eps=pi/4; Ash=5;  
>> x=A1.*sin(2*pi*t./T1)+A2.*sin(2*pi.*t./T2+eps)  
+Ash*randn(1,length(t)); plot(t(10002:end), x(10002:end))
```

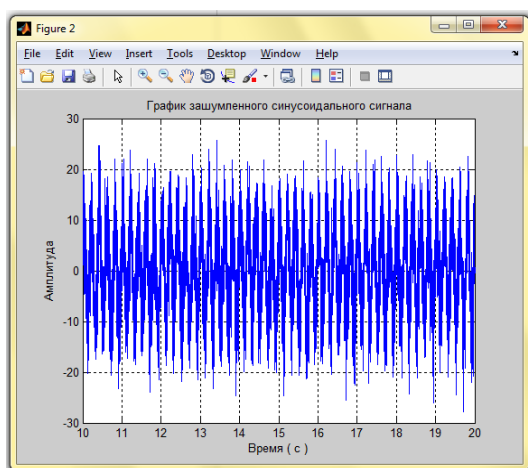


Рисунок 3

Следующие команды формируют три периода гармонического сигнала со случайной начальной фазой и последующим смещением с шумом. Для выравнивания количества отсчетов используют команду *ceil* – округление вверх.

```
>> A=0.05;  
>> B = 500; % верхняя граничная частота шума  
>> fi = rand(1)*2*pi; % случайная начальная фаза  
>> f0B = f0/B; % относительная частота сигнала  
>> Ngraf = ceil(3/f0B)*2+1;  
>> i = 1: Ngraf;  
>> S = A*cos(pi*f0B*i+fi); % Ngraf отсчетов сигнала  
>> ksi = randn(1, Ngraf); % Ngraf отсчетов шума  
>> Sksi = S + ksi; % Ngraf отсчетов смеси  
>> subplot(3,1,1); plot(i,S); % график сигнала  
>> subplot(3,1,2); plot(i,ksi); % график шума  
>> subplot(3,1,3); plot(i,Sksi,'r'); % график Сигнал+шум
```

В результате получаем сигналы, изображенные на рис. 4.

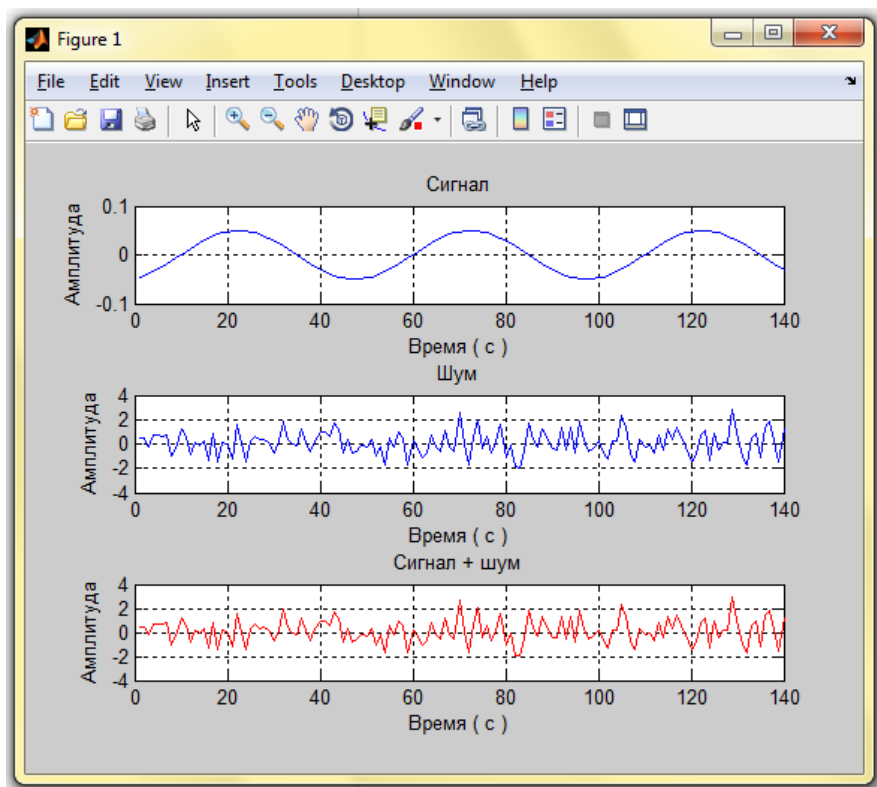


Рисунок 4

9) Получение данных из внешних источников

Преобразование аналогового сигнала из внешних источников в цифровой и обратно – это процессы, которые выполняются аппаратными средствами. MATLAB же, будучи программным пакетом, может лишь взаимодействовать с соответствующим оборудованием. Кроме того, в MATLAB предусмотрены средства для воспроизведения и записи звука, а также для работы со звуковыми файлами формата WAV. (Waveform Audio File Format WAVE, WAV, от англ. waveform – «в форме волны» – формат файла-контейнера для хранения записи оцифрованного аудиопотока).

Для считывания WAV-файлов в MATLAB имеется функция *wavread*. В простейшем случае она может быть использована следующим образом:

```
>> y = wavread ('filename');
```

Здесь *filename* – имя звукового файла (расширение .wav указывать обязательно).

В результате выполнения функции в переменную *y* будет перемещено всё содержимое указанного файла. Строки матрицы *y* соответствуют отсчётам сигнала, столбцы – каналам, которых в WAV-файле может быть несколько.

В звуковом файле отсчёты сигнала представлены целыми числами, лежащими в диапазоне $-128..+127$ (8 бит на отсчёт), либо $-32768...+32767$ (16 бит на отсчёт). Управлять нормировкой считываемых отсчётов можно с помощью дополнительного строкового параметра '*fmt*', добавляемого в конце списка входных параметров функции *wavread*. При принятом по умолчанию варианте 'double' значения отсчётов приводятся в диапазоне $-1..+1$, а при значении 'native' функция возвращает целые числа в том виде, в котором они хранятся в WAV-файле.

Помимо собственно отсчётов, в WAV-файле хранится служебная информация о частоте дискретизации, количество бит на отсчёт и т.п. Узнать частоту дискретизации можно, используя при вызове функции второй выходной параметр:

```
>> y = wavread ('filename');
```

Например, необходимо разработать скрипт MATLAB, позволяющий считать данные из WAV-файла и вывести их в графическом виде.

```
>> [y, Fs]=wavread('E:\Speech Off.wav');  
>> subplot(2,1,1); %функция вывода нескольких графиков  
>> plot(y(:,1)) %Канал 1  
>> grid on;  
>> subplot(2,1,2);  
>> plot(y(:,2)) %Канал 2  
>> grid on;
```

Результат выполнения скрипта (рис. 5):

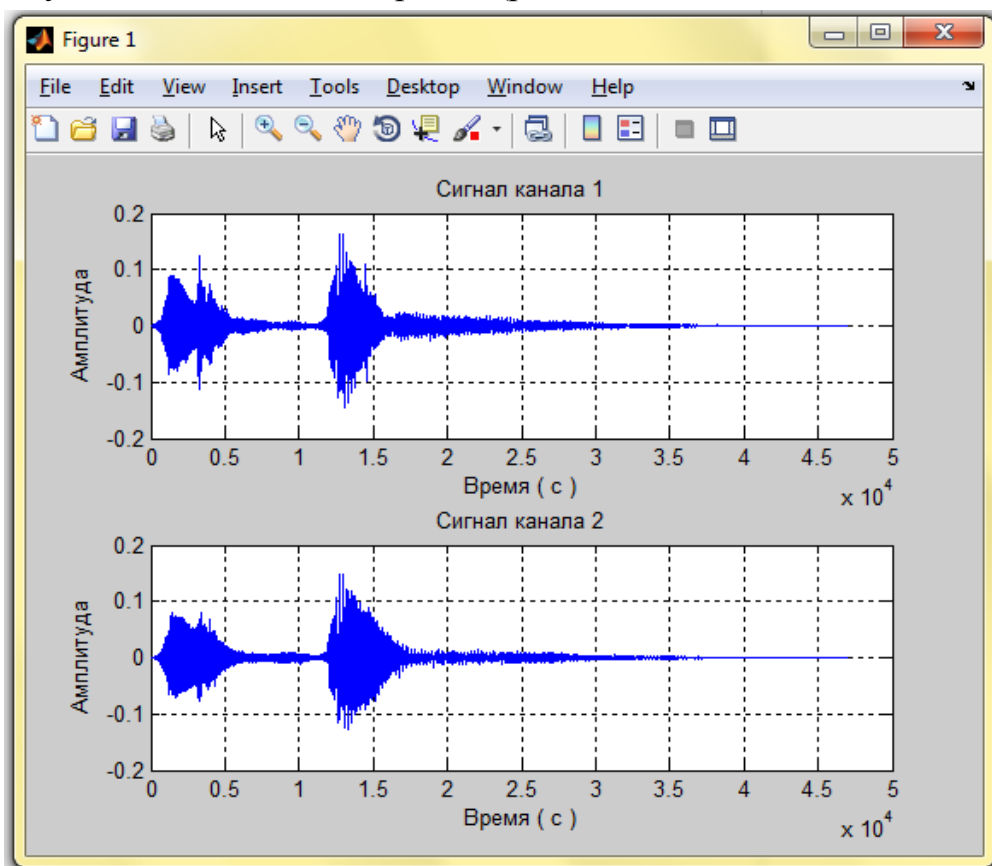


Рисунок 5

Чтобы записать вектор (или матрицу) на диск в виде WAV-файла, используется функция *wavwrite*:

```
>> wavwrite(y, Fs, N, 'filename') ,
```

где y – записываемые данные (вектор для монофонической записи, двух-столбцовая матрица – для создания стереофайла); F_s – частота дискретизации в герцах; N – число бит на отсчёт (8 или 16); ‘filename’ – имя создаваемого файла.

Выходных параметров у данной функции нет. Параметры N и F_s можно опускать, при этом используются значения по умолчанию – $N = 16$ и $F_s = 8000$.

Создадим скрипт MATLAB, производящий операцию удвоения амплитуды сигнала WAV-файла и записывающий изменения в новый файл. Исходный код представить ниже:

```
>> [y,Fs,b]=wavread('E:\Speech Off.wav');
>> subplot(2,1,1);    %функция вывода нескольких графиков
>> plot(y(:,1))       %Канал 1
>> title('Сигнал канала 1');
>> xlabel('Время ( с )')
>> ylabel('Амплитуда')
>> grid on;
>> subplot(2,1,2);
>> plot(y(:,2))       %Канал 2
>> grid on;
>> title('Сигнал канала 2');
>> xlabel('Время ( с )')
>> ylabel('Амплитуда')
>> y2=y*5;           %изменяем амплитуду
>> Fs2=Fs*5;         %изменяем частоту
>> figure;           %показать график в новом окне
>> plot(y2)
>> title('Измененный сигнал');
>> xlabel('Время ( с )')
>> ylabel('Амплитуда')
>> grid on;
>> wavwrite(y2,Fs,b,'E:\Speech Off-new.wav')
```

Данный пример удваивает амплитуду (громкость) исходного файла и записывает изменения в файл с именем ‘Speech Off-new.wav’.

Результаты выполнения скрипта представлены на рис. 6, 7.

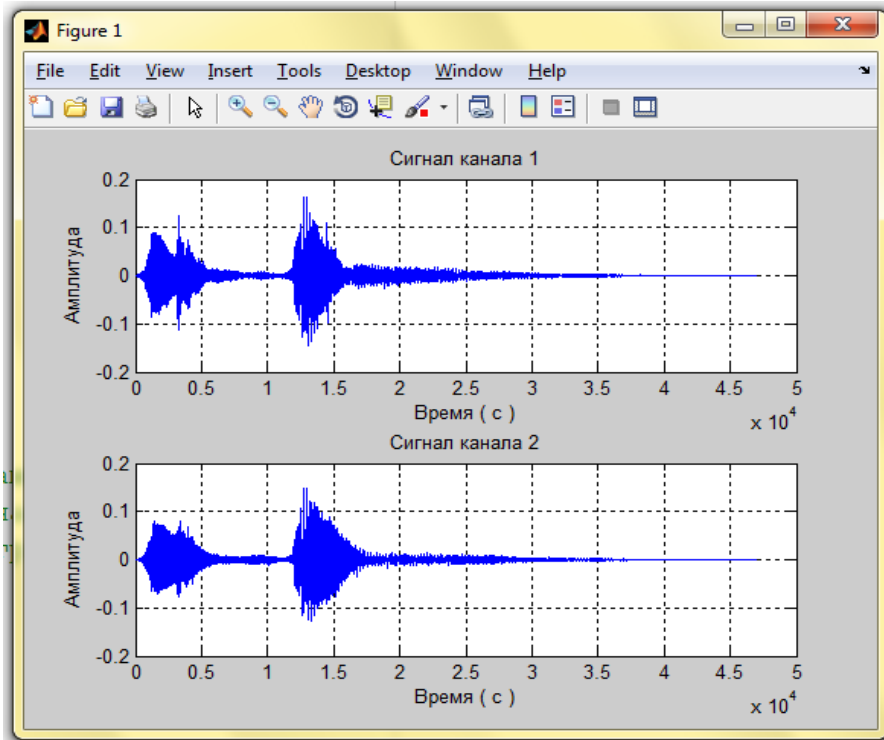


Рисунок 6

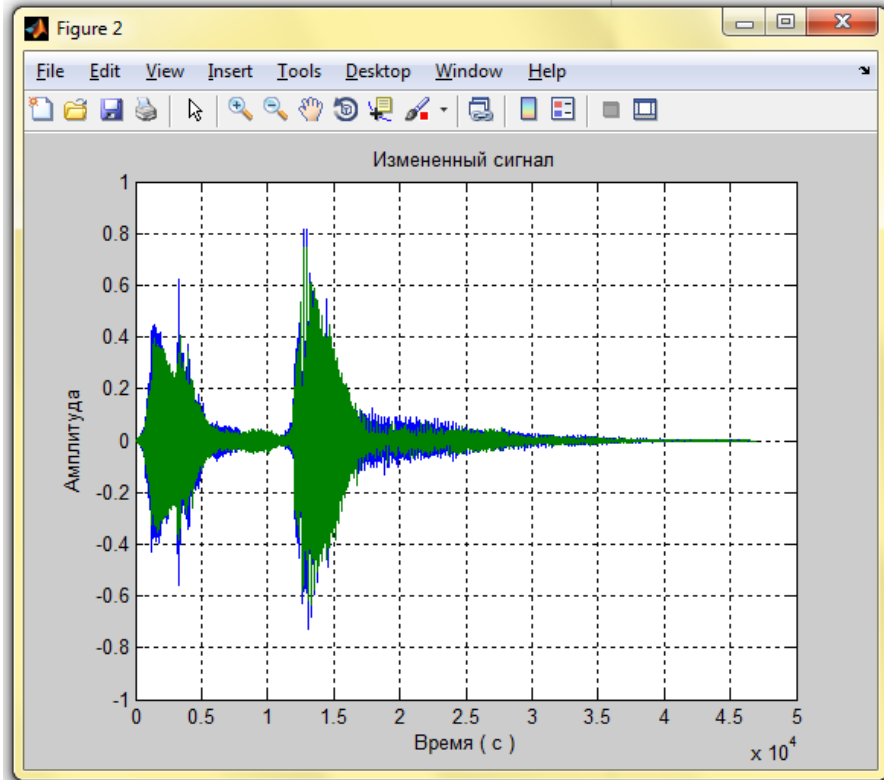


Рисунок 7

Приложение *SPTool* (Signal Processing Tool)

Программа *SPTool* предоставляет для пользователя графическую среду для просмотра графиков, спектров сигналов, а также выполняет расчет фильтров. Для вызова программы нужно в командном окне MATLAB набрать:

```
>> sptool
```

Появится окно, изображенное на рис. 8:

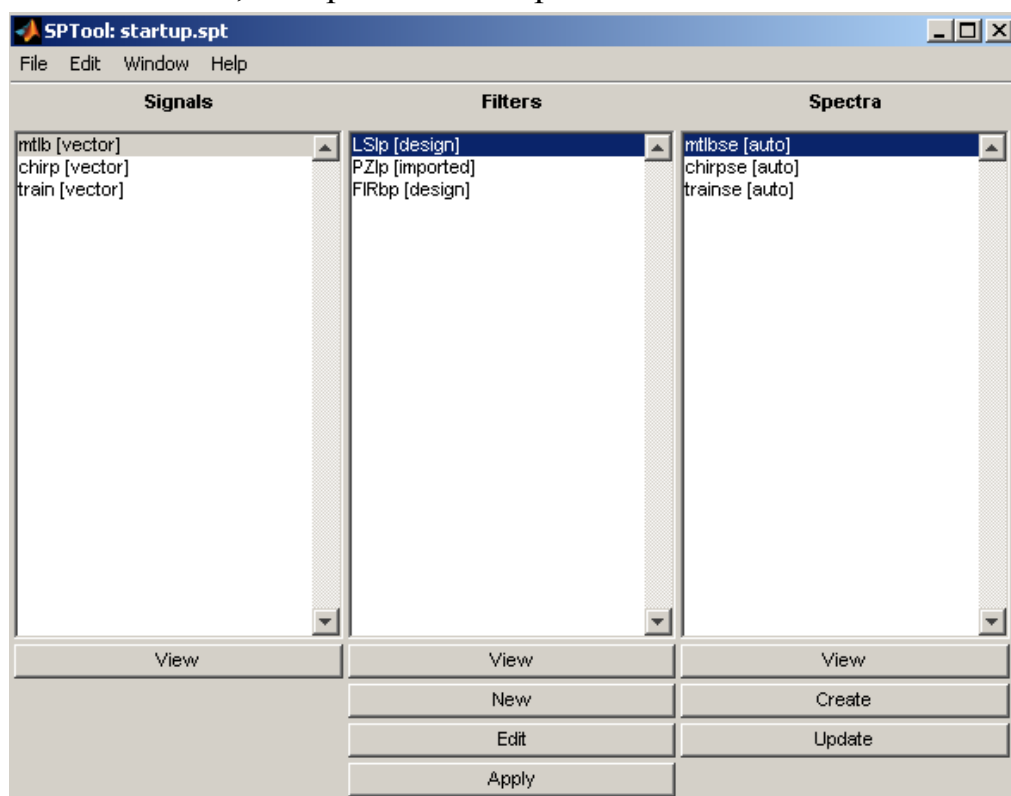


Рисунок 8

Импортируйте в *SPTool* сформированный согласно заданию сигнал. С этой целью выберите команду *Import* в меню *File* главного окна программы *SPTool*. Появится окно *Import to SPTool*. Переключатель *Source* установите в положение *From Workspace*. В списке *Workspace Contents* перечислены переменные, имеющиеся в рабочей области MATLAB. В раскрывающемся списке *Import As* нужно выбрать строку *Signal*. Далее выберите в списке идентификатор вектора, содержащего отсчёты сигнала, и нажмите кнопку --> напротив поля *Data*. Затем аналогичным образом введите значение частоты дискретизации F_s в поле *Sampling Frequency*. Задайте имя сигнала в поле *Name*. Под этим именем он будет помещён в список сигналов *SPTool*. Вслед за этим следует нажать кнопку ОК. Просмотрите график сигнала. Для этого

выделите имя сигнала в списке сигналов и нажмите кнопку *View*, расположенную под этим списком. При просмотре графиков можно изменять масштаб, укрупняя отдельные участки. Для измерения значений в отдельных точках используются маркеры, которые можно перетаскивать мышью.

1.2. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Загрузите MATLAB. В командном окне задайте частоту дискретизации, интервал времени, в течение которого будет длиться сигнал (учитывается время, отводимое на импульсы и на паузы между ними) и сформируйте вектор отсчётов сигнала, задайте необходимые параметры сигнала, руководствуясь индивидуальным заданием. Согласно заданию сформируйте необходимые сигналы:

- затухающий гармонический сигнал;
- пару разнополярных прямоугольных импульсов с амплитудой 2 В и длительностью 10 мс, расположенных справа и слева от начала отсчёта времени;
- импульс вида $\sin(\pi t)/(\pi t)$;
- симметричный трапецевидный импульс с амплитудой 5 В и размерами верхнего и нижнего оснований 10 мс и 40 мс;
- три симметричных треугольных импульсов, интервалы между которыми линейно увеличиваются, а амплитуды экспоненциально уменьшаются;
- последовательность из 4 импульсов, имеющих форму одного периода функции $\sin^2(t)$. Длительность импульсов равна 30 мс. Расстояние между центрами импульсов будет равным 35 мс. Импульсы будут с возрастанием номера;
- последовательность однополярных прямоугольных импульсов с амплитудой 5 В, частотой следования 100 Гц и длительностью 10 мс.

Импортируйте в SPtool зашумленную версию одного из заданных сигналов.

Запишите любой WAV-файл и выполните изменение его параметров.

ПРАКТИЧЕСКАЯ ЧАСТЬ 2

2.1. ФУНКЦИИ ГЕНЕРАЦИИ СИГНАЛОВ В SIMULINK

Пакет **Simulink** позволяет осуществлять исследование (моделирование) поведения динамических систем. Запуск пакета **Simulink** можно произвести из командного окна MATLAB, нажав пентаграмму в панели инструментов и открыв новую модель. При этом открываются два окна: пустое окно *untitled* (окно для создания блок-диаграммы модели) и окно **Library Simulink** (библиотека) с перечнем основных разделов библиотеки. В открывшееся окно *untitled* необходимо добавить блоки, моделирующие работу источников сигналов, измерительных приборов и аналоговых систем.

БИБЛИОТЕКА БЛОКОВ SIMULINK

Источник синусоидального сигнала Sine Wave

Назначение: формирует синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением. Для формирования выходного сигнала блоком могут использоваться два алгоритма. Вид алгоритма определяется параметром *Sine Type* (способ формирования сигнала): *Time-based* – по текущему времени; *Sample-based* – по величине шага модельного времени.

При формировании выходного сигнала по текущему значению времени для непрерывных систем выходной сигнал источника в этом режиме соответствует выражению:

$$y = \textit{Amplitude} * \sin(\textit{frequency} * \textit{time} + \textit{phase}) + \textit{bias}.$$

Параметры:

Amplitude – амплитуда.

Bias – постоянная составляющая сигнала.

Frequency (rads/sec) – частота (рад/с).

Phase (rads) – начальная фаза (рад).

Sample time – шаг модельного времени. Используется для согласования работы источника и других компонентов модели во времени. Параметр может принимать следующие значения:

0 (по умолчанию) – используется при моделировании непрерывных систем;

> 0 (положительное значение) – задается при моделировании дискретных систем. В этом случае шаг модельного времени можно интерпретировать как шаг квантования по времени выходного сигнала;

– 1 – шаг модельного времени устанавливается таким же, как и в предшествующем блоке, то есть блоке, откуда приходит сигнал в данный блок.

Этот параметр может задаваться для большинства блоков библиотеки **Simulink**. В дальнейшем он рассматриваться не будет.

При расчетах для очень больших значений времени точность расчета выходных значений сигнала падает вследствие значительной ошибки округления.

Осциллограф **Scope**

Назначение: Строит графики исследуемых сигналов в функции времени. Позволяет наблюдать за изменениями сигналов в процессе моделирования. Изображение блока и окно для просмотра графиков показаны на рис. 9.

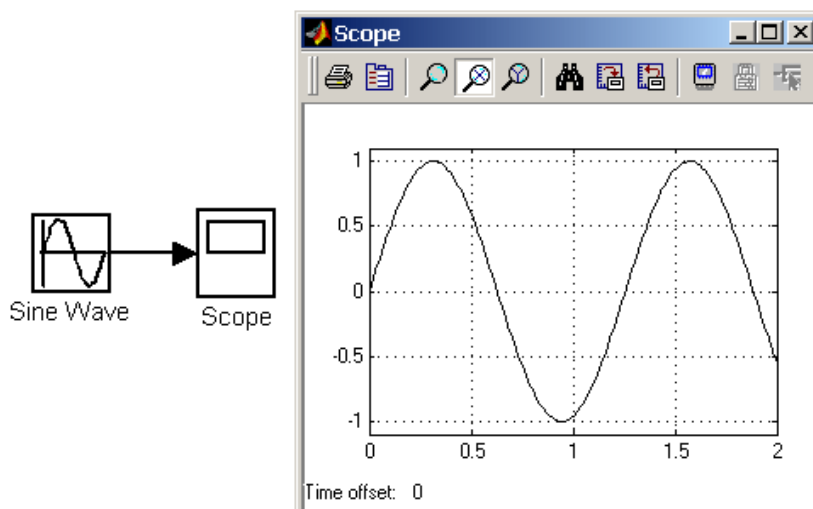
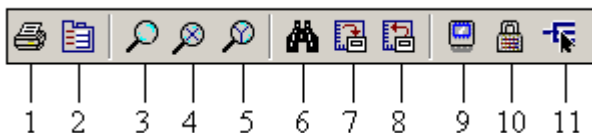


Рисунок 9

Для того чтобы открыть окно просмотра сигналов, необходимо выполнить двойной щелчок левой клавишей «мыши» на изображении блока. Это можно сделать на любом этапе расчета (как до начала расчета, так и после него, а также во время расчета). В случае если на вход блока поступает

векторный сигнал, кривая для каждого элемента вектора строится отдельным цветом.

Настройка окна осциллографа выполняется с помощью панелей инструментов:



Панель инструментов содержит 11 кнопок:

Print – печать содержимого окна осциллографа.

Parameters – доступ к окну настройки параметров.

Zoom – увеличение масштаба по обеим осям.

Zoom X-axis – увеличение масштаба по горизонтальной оси.

Zoom Y-axis – увеличение масштаба по вертикальной оси.

Autoscale – автоматическая установка масштабов по обеим осям.

Save current axes settings – сохранение текущих настроек окна.




Restore saved axes settings – установка ранее сохраненных настроек окна.




Floating scope – перевод осциллографа в «свободный» режим.

Lock/Unlock axes selection – закрепить/разорвать связь между текущей координатной системой окна и отображаемым сигналом. Инструмент доступен, если включен режим **Floating scope**.

Signal selection – выбор сигналов для отображения. Инструмент доступен, если включен режим **Floating scope**.

Изменение масштабов отображаемых графиков можно выполнять несколькими способами:

1. Нажать соответствующую кнопку (,  или ) и щелкнуть один раз левой клавишей «мыши» в нужном месте графика. Произойдет 2,5 кратное увеличение масштаба.

2. Нажать соответствующую кнопку (,  или ) и, нажав левую клавишу «мыши», с помощью динамической рамки или отрезка указать область графика для увеличенного изображения.

3. Щелкнуть правой клавишей «мыши» в окне графиков и выбрать команду **Axes properties...** в контекстном меню. Откроется окно свойств

графика, в котором с помощью параметров **Y-min** и **Y-max** можно указать предельные значения вертикальной оси. В этом же окне можно указать заголовок графика (**Title**), заменив выражение **%<SignalLabel>** в строке ввода. Окно свойств показано на рис. 10.

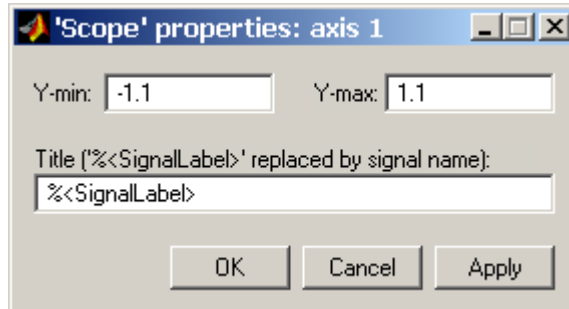


Рисунок 10

Параметры блока устанавливаются в окне *'Scope' parameters*, которое открывается с помощью инструмента (**Parameters**) панели инструментов. Окно параметров имеет две вкладки:

General – общие параметры;

Data history – параметры сохранения сигналов в рабочей области MATLAB.

Вкладка общих параметров показана на рис. 11.

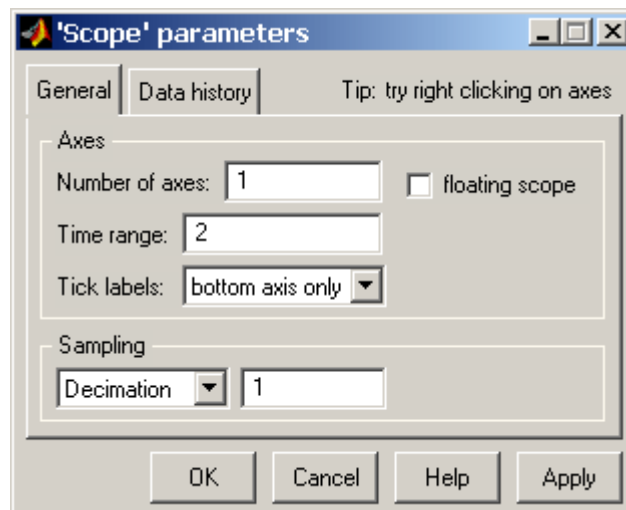


Рисунок 11

На вкладке **General** задаются следующие параметры:

Number of axes — число входов (систем координат) осциллографа. При изменении этого параметра на изображении блока появляются дополнительные входные порты.

Time range — величина временного интервала, для которого отображаются графики. Если время расчета модели превышает заданное параметром *Time range*, то вывод графика производится порциями, при этом интервал отображения каждой порции графика равен заданному значению *Time range*.

Tick labels — вывод/скрытие осей и меток осей. Может принимать три значения (выбираются из списка): *all* – подписи для всех осей; *none* – отсутствие всех осей и подписей к ним; *bottom axis only* – подписи горизонтальной оси только для нижнего графика.

Sampling — установка параметров вывода графиков в окне. Задаёт режим вывода расчетных точек на экран. При выборе **Decimation** кратность вывода устанавливается числом, задающим шаг выводимых расчетных точек. На рис. 12 и 13 показаны графики синусоидальных сигналов рассчитанных с фиксированным шагом 0.1 с. На рис. 12 в окне блока **Scope** выводится каждая расчетная точка (параметр *Decimation* равен 1). На рис. 13 показан вывод каждого второго значения (параметр *Decimation* равен 2). Маркерами на графиках отмечены расчетные точки.

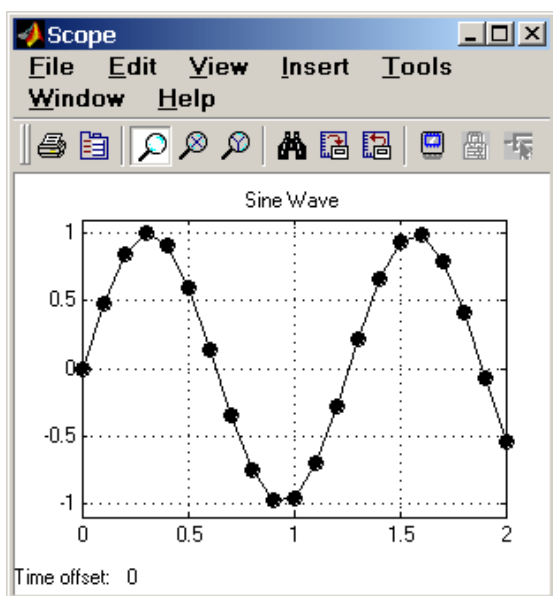


Рисунок 12

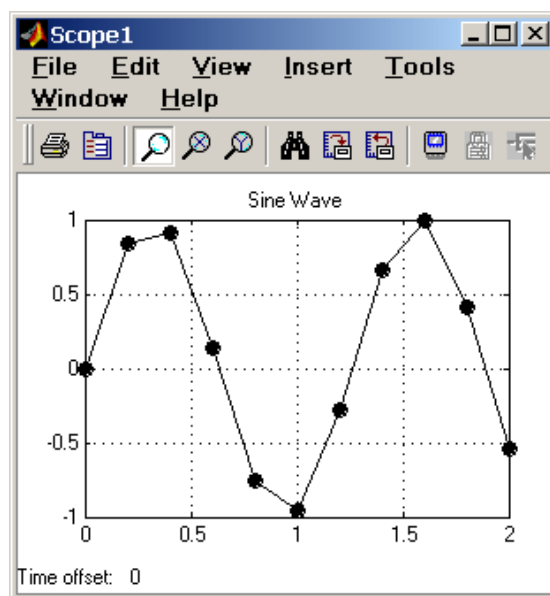


Рисунок 13

floating scope – перевод осциллографа в «свободный» режим (при установленном флажке).

В случае если режим вывода расчетных точек задается как **Sample time**, его числовое значение определяет интервал квантования при отображении сигнала. На рис. 14 показан график синусоидального сигнала для случая, когда значение параметра **Sample time** равно 0.1.

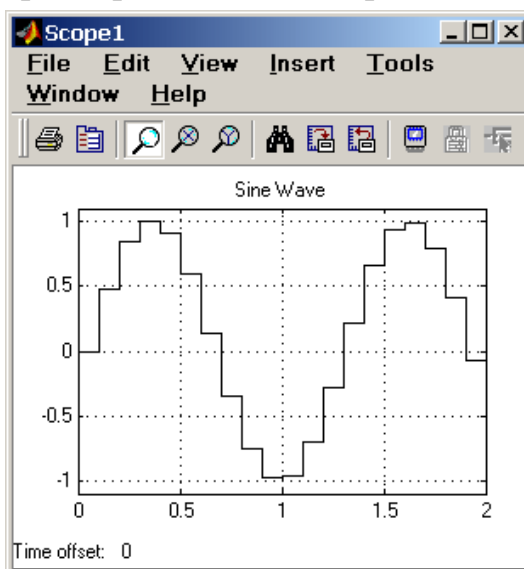


Рисунок 14

На вкладке **Data history** (рис. 15) задаются следующие параметры:

Limit data points to last – максимальное количество отображаемых расчетных точек графика. При превышении этого числа начальная часть графика обрезается. В случае если флажок параметра **Limit data points to last** не установлен, **Simulink** автоматически увеличит значение этого параметра для отображения всех расчетных точек.

Save data to workspace – сохранение значений сигналов в рабочей области MATLAB.

Variable name – имя переменной для сохранения сигналов в рабочей области MATLAB.

Format – формат данных при сохранении в рабочей области MATLAB. Может принимать значения:

Array – массив,

Structure – структура,

Structure with time – структура с дополнительным полем «время».

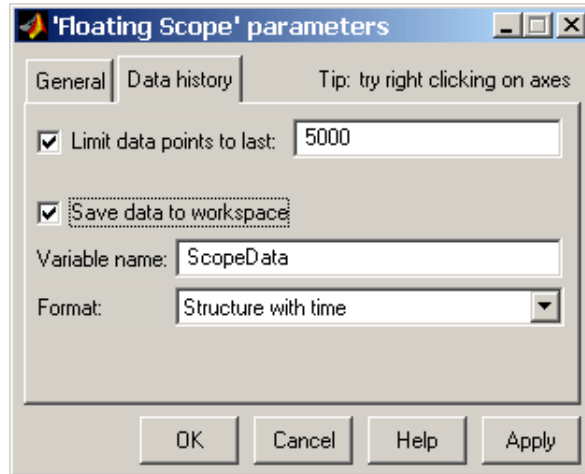


Рисунок 15

При формировании выходного сигнала по величине модельного времени и количеству расчетных шагов на один период выходной сигнал источника в этом режиме соответствует выражению:

$$y = \textit{Amplitude} * \sin[(k + \textit{Number of offset samples}) / \textit{Samples per period}] + \textit{bias} ,$$

где k – номер текущего шага расчета.

Параметры:

Amplitude – амплитуда.

Bias – постоянная составляющая сигнала.

Samples per period – количество расчетных шагов на один период синусоидального сигнала:

$$\textit{Samples per period} = 2p / (\textit{frequency} * \textit{Sample time}).$$

Number of offset samples – начальная фаза сигнала. Задается количеством шагов модельного времени:

$$\textit{Number of offset samples} = \textit{Phase} * \textit{Samples per period} / (2\pi).$$

Sample time – шаг модельного времени.

В данном режиме ошибка округления не накапливается, поскольку **Simulink** начинает отсчет номера текущего шага с нуля для каждого периода.

На рис. 16 показано применение блока с разными значениями шага модельного времени ($\textit{Sample time} = 0$ для блока **Sine Wave 1** и $\textit{Sample time} = 0.1$ для блока **Sine Wave 2**). Для отображения графиков выходных сигналов в модели использован виртуальный осциллограф (**Scope**).

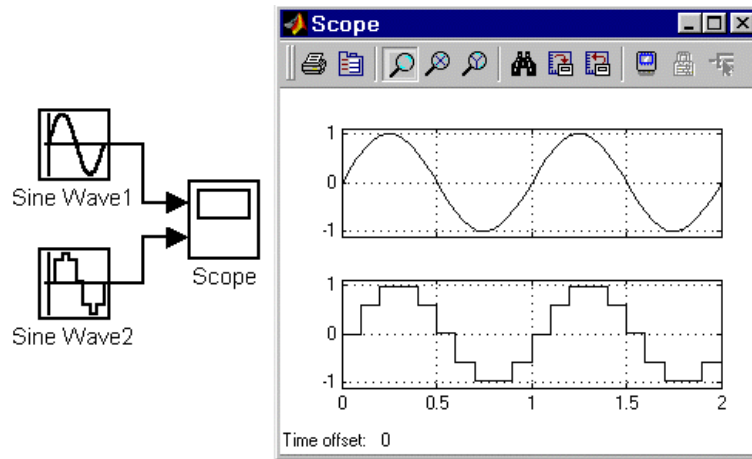


Рисунок 16

Источник линейно изменяющегося воздействия Ramp

Назначение: Формирует линейный сигнал вида

$$Y = Slope*time + Initial\ value.$$

Параметры:

Slope – скорость изменения выходного сигнала.

Start time – время начала формирования сигнала.

Initial value – начальный уровень сигнала на выходе блока.

На рис. 17 показано использование данного блока.

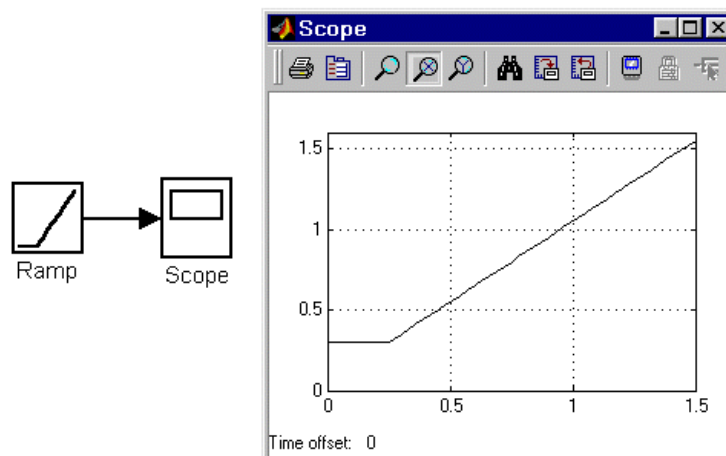


Рисунок 17

Генератор ступенчатого сигнала Step

Назначение: Формирует ступенчатый сигнал.

Параметры:

Step time – время наступления перепада сигнала (с).

Initial value – начальное значение сигнала.

Final value – конечное значение сигнала.

Перепад может быть как в большую сторону (конечное значение больше чем начальное), так и в меньшую (конечное значение меньше чем начальное). Значения начального и конечного уровней могут быть не только положительными, но и отрицательными (например, изменение сигнала с уровня -5 до уровня -3).

На рис. 18 показано использование генератора ступенчатого сигнала.

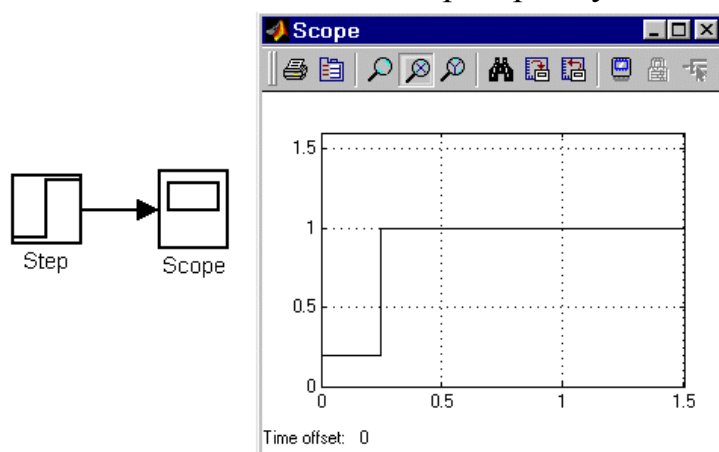


Рисунок 18

Генератор сигналов **Signal Generator**

Назначение: Формирует один из 4 видов периодических сигналов:

sine — синусоидальный сигнал;

square — прямоугольный сигнал;

sawtooth — пилообразный сигнал;

random — случайный сигнал.

Параметры:

Wave form – вид сигнала;

Amplitude – амплитуда сигнала;

Frequency – частота (рад/с);

Units – единицы измерения частоты. Может принимать два значения: Hertz (Гц) или rad/sec (рад/с).

На рис. 19 показано применение этого источника при моделировании прямоугольного сигнала.

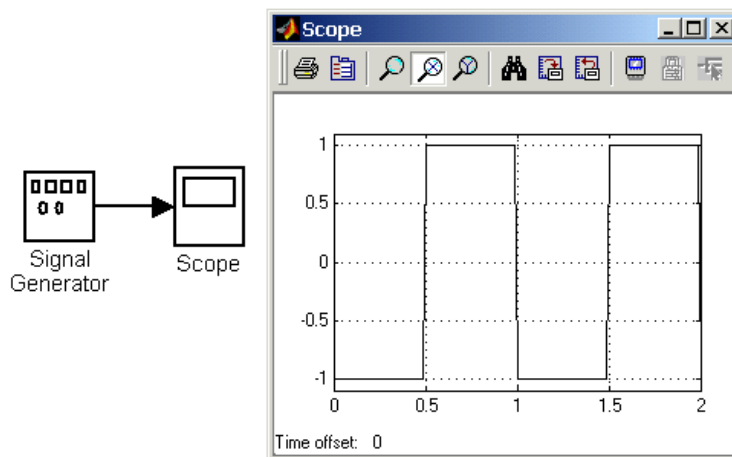


Рисунок 19

***Источник случайного сигнала с равномерным распределением
Uniform Random Number***

Назначение: Формирование случайного сигнала с равномерным распределением.

Параметры:

Minimum – минимальный уровень сигнала;

Maximum – максимальный уровень сигнала;

Initial seed – начальное значение.

***Источник случайного сигнала с нормальным распределением
Random Number***

Назначение: Формирование случайного сигнала с нормальным распределением уровня сигнала.

Параметры:

Mean – среднее значение сигнала;

Variance – дисперсия (среднеквадратическое отклонение);

Initial seed – начальное значение.

Источник импульсного сигнала Pulse Generator

Назначение: Формирование прямоугольных импульсов.

Параметры:

Pulse Type – способ формирования сигнала. Может принимать два значения:

Time-based – по текущему времени;

Sample-based – по величине модельного времени и количеству расчетных шагов.

Amplitude – амплитуда.

Period – период. Задается в секундах для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.

Pulse width – ширина импульсов. Задается в процентах по отношению к периоду для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.

Phase delay – фазовая задержка. Задается в секундах для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.

Sample time – шаг модельного времени. Задается для **Sample-based Pulse Type**.

Пример использования Pulse Generator показан на рис. 20.

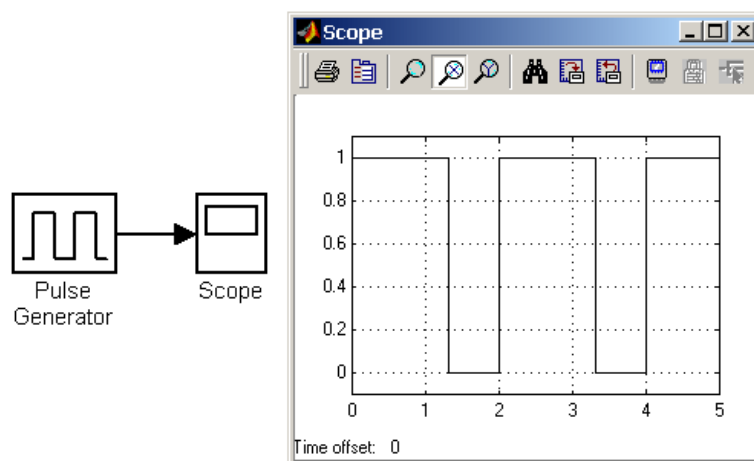


Рисунок 20

Генератор линейно-изменяющейся частоты Chirp Generator

Назначение: Формирование синусоидальных колебаний, частота которых линейно изменяется.

Параметры:

Initial frequency – начальная частота (Гц);

Target time – время изменения частоты (с);

Frequency at target time – конечное значение частоты (Гц).

Пример использования блока показан на рис. 21.

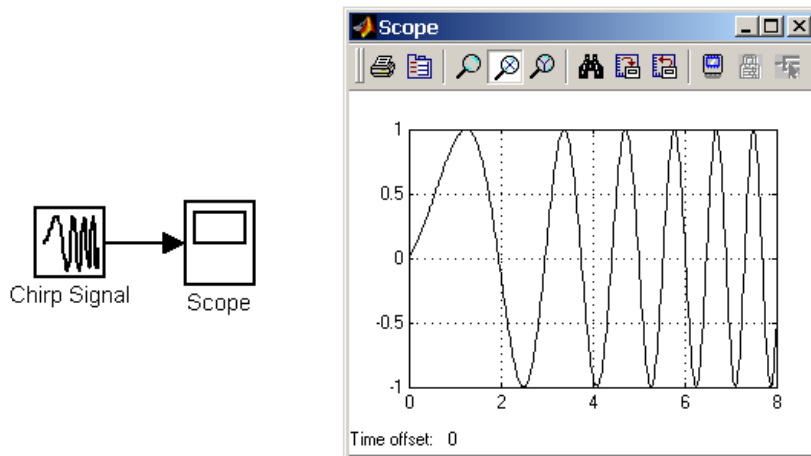


Рисунок 21

Генератор белого шума Band-Limited White Noise

Назначение: Создает сигнал заданной мощности, равномерно распределенной по частоте.

Параметры:

Noise Power – мощность шума;

Sample Time – модельное время;

Seed – число, необходимое для инициализации генератора случайных чисел.

На рис. 22 показана работа этого генератора.

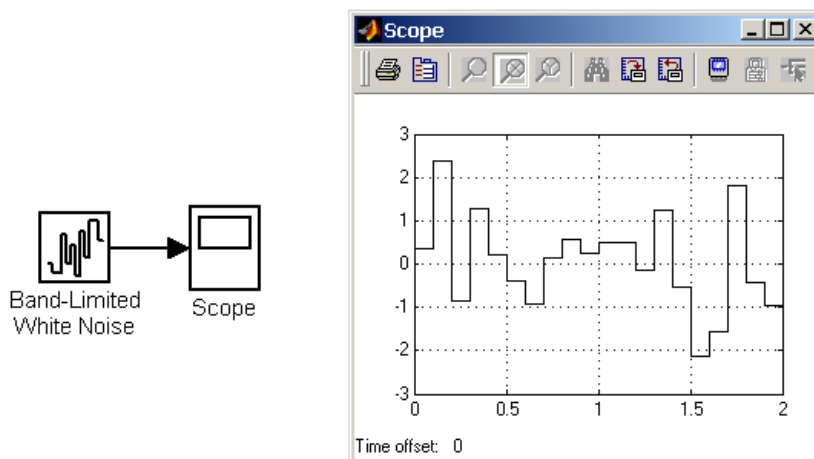


Рисунок 22

Источник временного сигнала Clock

Назначение: Формирует сигнал, величина которого на каждом шаге расчета равна текущему времени моделирования.

Параметры:

Decimation – шаг, с которым обновляются показания времени на изображении источника (в том случае, если установлен флажок параметра **Display time**). Параметр задается как количество шагов расчета. Например, если шаг расчета модели в окне диалога **Simulation parameters** установлен равным 0.01 с, а параметр **Decimation** блока **Clock** задан равным 1000, то обновление показаний времени будет производиться каждые 10 с модельного времени.

Display time – отображение значения времени в блоке источника.

На рис. 23 показан пример работы данного источника.

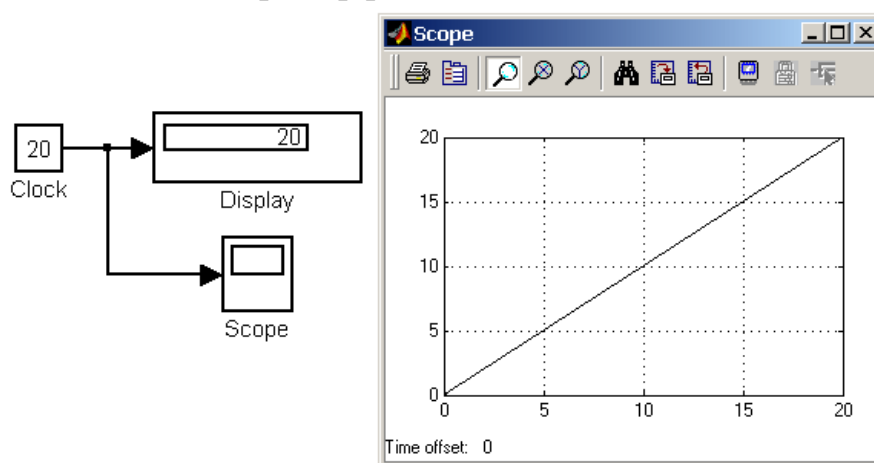


Рисунок 23

Цифровой источник времени Digital Clock

Назначение: Формирует дискретный временной сигнал.

Параметры:

Sample time – шаг модельного времени (с).

На рис. 24 показана работа источника Digital Clock.

Блок считывания данных из файла From File

Назначение: Получение данных из внешнего файла.

Параметры:

File Name – имя файла с данными;

Sample time – шаг изменения выходного сигнала блока.

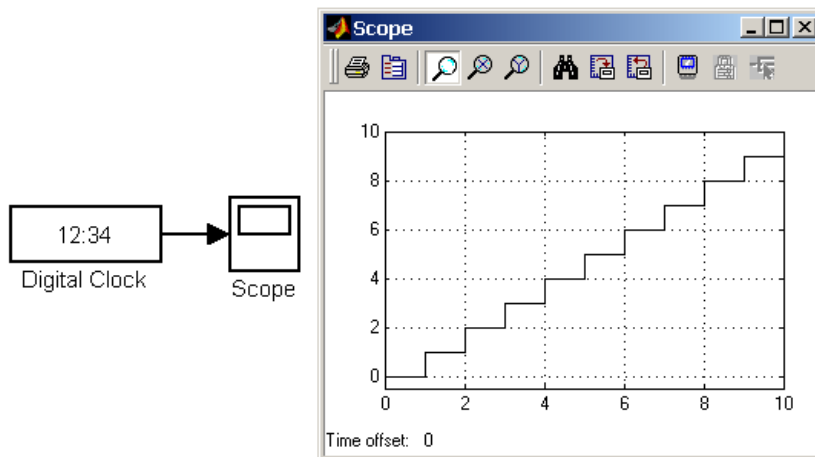


Рисунок 24

Данные в файле представлены в виде матрицы. Она состоит, как минимум, из двух строк. Значения времени записаны в первой строке матрицы, а в остальных строках находятся значения сигналов, соответствующие данным моментам времени. Значения времени записываются в возрастающем порядке. Выходной сигнал блока содержит только значения сигналов, а значения времени в нем отсутствуют. Если шаг расчета текущей модели не совпадает с отсчетами времени в файле данных, то **Simulink** выполняет линейную интерполяцию данных. Файл данных (mat-файл), из которого считываются значения, не является текстовым. Структура файла подробно описана в справочной системе MATLAB. Пользователям **Simulink** удобнее всего создавать mat-файл с помощью блока **To File** (библиотека **Sinks**).

На рис. 25 показан пример использования данного блока. Из файла data.mat считываются значения синусоидального сигнала.

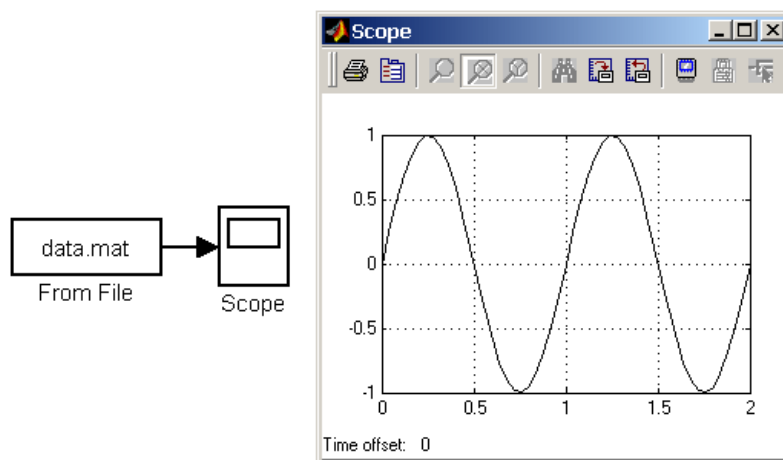


Рисунок 25

Блок периодического сигнала Repeating Sequence

Назначение: Формирование периодического сигнала.

Параметры:

Time values – вектор значений модельного времени;

Output values – вектор значений сигнала для моментов времени заданных вектором **Time values**.

Блок выполняет линейную интерполяцию выходного сигнала для моментов времени не совпадающих со значениями, заданными вектором **Time values**. На рис. 26 показан пример использования блока для формирования пилообразного сигнала. Значения модельного времени заданы вектором [0 3], а значения выходного сигнала-вектором [0 2].

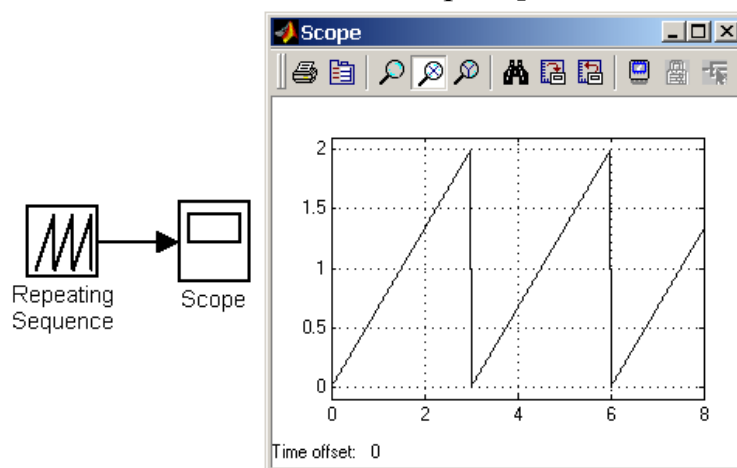


Рисунок 26

2.2. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Запустите пакет **Simulink** из командного окна MATLAB, нажав пентаграмму в панели инструментов и открыв новую модель. Согласно заданию создайте модель сигнала.

СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать: цель работы; задание на формирование сигналов, полученное от преподавателя; тексты программ (m-файлы) и графики построенных сигналов; схему модели сигнала, построенную в пакете **Simulink**; выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение сигнала. Какие существуют виды сигналов?
2. Какие функции MATLAB используются для генерации периодических сигналов?
3. Какие функции MATLAB используются для генерации одиночных импульсов?
4. Какие существуют способы задания последовательности импульсов для функции `pulstran`?
5. В чем заключается цифровая обработка сигналов?
6. Что такое частота дискретизации?
7. Что произойдет при нарушении условий теоремы Котельникова?
8. Как в среде MATLAB сгенерировать шум?
9. Как импортировать в SPtool сформированный сигнал?
10. Как построить модель сигнала в пакете Simulink?

СПИСОК ЛИТЕРАТУРЫ

1. Сергиенко А. Б. Цифровая обработка сигналов / А. Б. Сергиенко. – СПб. : Питер, 2006. – 604 с.: ил.
2. Солонина А. И. Цифровая обработка сигналов. Моделирование в MATLAB / А. И. Солонина, С. М. Арбузов. – СПб. : БХВ-Петербург, 2008. – 816 с.: ил.
3. Курбатова Е. А. MATLAB 7 / Е. А. Курбатова. – Самоучитель. М. : Вильямс, 2006. – 256 с.
4. Дьяконов В. П. Matlab 6.5 SP1/7 + Simulink 5/6. Обработка сигналов и проектирование фильтров / В. П. Дьяконов. – М. : Солон-Пресс, 2005. – 576 с.
5. Поршнева С. В. MATLAB 7. Основы работы и программирования : учебник. / С. В. Поршнева. – М. : Бином. Лаборатория знаний, 2006. – 320 с.

СОДЕРЖАНИЕ

Вступление	3
Лабораторная работа	
Формирование сигналов и их преобразование при цифровой обработке	4
Практическая часть 1	11
1.1. Примеры создания сигналов	
1.2. Индивидуальное задание	23
Практическая часть 2.	
2.1. Функции генерации сигналов в SIMULINK	24
2.2. Индивидуальное задание	38
Содержание отчета	38
Контрольные вопросы	39
Список литературы	39