

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Санкт-Петербургский горный университет**

Кафедра информатики и компьютерных технологий

**ИНФОРМАТИКА
ПРОГРАММИРОВАНИЕ
РАДИОТЕХНИЧЕСКИХ СИГНАЛОВ
В SCILAB**

*Методические указания к курсовой работе
для студентов специальности 11.05.01*

**САНКТ-ПЕТЕРБУРГ
2019**

УДК 004.424 (073)

ПРОГРАММИРОВАНИЕ РАДИОТЕХНИЧЕСКИХ СИГНАЛОВ В SCILAB: Методические указания к курсовой работе / Санкт-Петербургский горный университет. Сост.: *О.В. Косарев, Е.Г. Водкайло, А.Б. Маховиков*. СПб, 2019. 33 с.

Рассмотрены основные приемы работы в Scilab. Показана реализация линейных, разветвляющихся и циклических вычислительных процессов, построение графиков и обмен данными через файл. Приведены примеры программирования последовательности прямоугольных видеоимпульсов, амплитудно-, частотно- и фазоманипулированных последовательностей радиоимпульсов, записи массива сигнала в файл.

Предназначены для студентов специальности 11.05.01 «Радиоэлектронные системы и комплексы».

Научный редактор доц. *А.Б. Маховиков*

Рецензент канд. техн. наук *К.В. Столяров* («Корпорация Телум Инк»)

© Санкт-Петербургский
горный университет, 2019

ВВЕДЕНИЕ

Курсовая работа посвящена основам программирования в системе компьютерной алгебры Scilab. В ходе выполнения курсовой работы необходимо продемонстрировать знание трех типов вычислительных процессов (линейный, разветвляющийся, циклический), способов обмена данными с программой и работу со средствами визуализации результатов. Студенты, обучающиеся по специальности 11.05.01 «Радиоэлектронные системы и комплексы», должны уметь программировать радиотехнические сигналы, алгоритмы обработки сигналов во временной и частотной области, функциональные модели устройств обработки и др. Цель настоящей курсовой работы – дать студентам навыки программирования радиотехнических сигналов во временной области и записи этих сигналов в файл для последующей обработки. Примеры программирования приведены для Scilab. Для выполнения курсовой работы требуется владеть приемами работы в Scilab на минимальном уровне. Допустимо выполнить работу в другой системе компьютерной алгебры, математическом пакете или на современных языках программирования. Предполагается, что модели сигналов, запрограммированные в курсовой работе и сохраненные в файл, студенты смогут использовать на старших курсах обучения в других компьютерных программах, связанных с обработкой сигналов.

1. ИСХОДНЫЕ ДАННЫЕ

Необходимо запрограммировать в Scilab последовательность прямоугольных радиоимпульсов с амплитудной, частотной и фазовой модуляцией. Модулирующая последовательность видеоимпульсов формируется как двоичный код первых заглавных букв ФИО студента в кодировке ASCII.

Параметры последовательности радиоимпульсов:

- длительность одной дискреты 1 мс;
- амплитуда дискреты 1 мВ;
- скважность последовательности импульсов 1;
- количество импульсов в пачке 24;
- количество пачек 2;
- несущая частота 4 кГц (частота для дискреты 0), 8 кГц для дискреты 1;
- начальная фаза 0 градусов (для дискреты 0), 180 градусов для дискреты 1.

Последовательность радиоимпульсов должна формироваться в цикле. Эпюры сигналов должны быть подписаны. Пояснительная записка должна содержать скриншоты кода и рисунков. Модели сигналов должны быть записаны в файл и считаны в другую переменную.

2. ПОНЯТИЕ ИМПУЛЬСНОГО СИГНАЛА. ПАРАМЕТРЫ ИМПУЛЬСА

Прежде чем говорить об импульсном сигнале, необходимо дать определение просто сигналу. Итак, **СИГНАЛ** (франц. signal, нем. Signal, от лат. signum – знак) – условный знак, физический процесс (или явление), несущий сообщение (информацию) о каком-либо событии, состоянии объекта наблюдения либо передающий команды управления, указания, оповещения и т.д. [1].

Следует обратить внимание на то, что смысл и значение сигнал приобретает только после того, как получатель его интерпретирует (получит, узнает, поймет, что он означает).

Нас, конечно же, интересует сигнал как физический процесс изменения напряжения и тока в электрических цепях. Регистрация изменения напряжения в цепи лежит в основе телефонной связи. Регистрация изменения напряжения на выходе радиоприемного устройства отраженных сигналов лежит в основе радиолокации. Потому далее под термином «сигнал» будем понимать электрический или электромагнитный сигнал.

Сигналы бывают непрерывные и импульсные. **Непрерывный сигнал** — это такой сигнал, амплитуда которого постоянна или изменяется во времени медленно.

Импульсный сигнал, или **электрический импульс** — это кратковременное скачкообразное изменение электрического напряжения или силы тока [2].

Импульсы, имеющие постоянную составляющую и не содержащие высокочастотных колебаний, называются **видеоимпульсами**. По характеру изменения во времени различают видеоимпульсы прямоугольной, пилообразной, трапецеидальной, колоколообразной, экспоненциальной и др. формы (рис. 1, а–г).

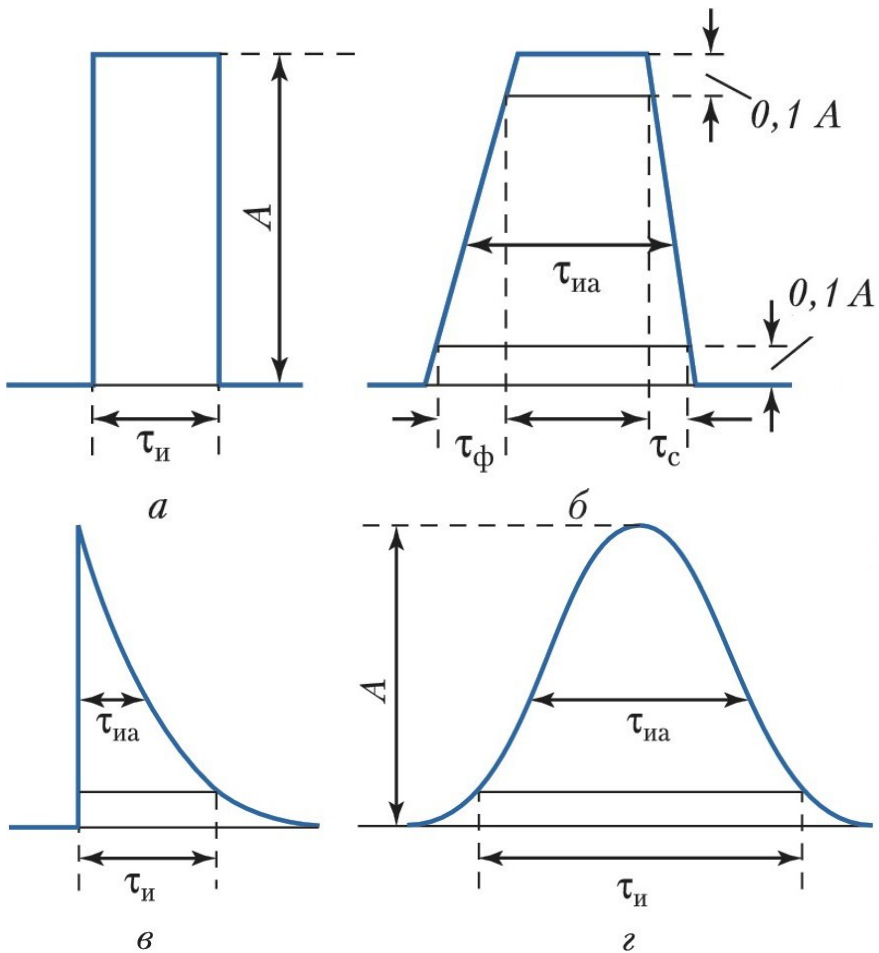


Рис. 1. Электрические импульсы разной формы: а – прямоугольный; б – трапеци-
дальный; в – экспоненциальный; г – колоколообразный; A – амплитуда; $\tau_{и}$ – дли-
тельность импульса; $\tau_{иа}$ – длительность импульса на уровне $0,5 A$; $\tau_{ф}$ и $\tau_{с}$ – длитель-
ность фронта и спада соответственно.

Импульсы, представляющие собой ограниченные во времени высокочастотные (ВЧ) или сверхвысокочастотные (СВЧ) колебания, огибающая которых, имеет форму видеоимпульса (рис. 1, д), называются радиоимпульсами. Длительность и амплитуда радиоимпульсов соответствуют параметрам модулирующих видеоимпульсов. Дополнительным параметром является несущая частота — частота ВЧ-заполнения импульса [3].

Помимо одиночных и нерегулярно следующих во времени импульсов, на практике используют периодические последовательности импульсов. Последовательности описываются периодом повторения T или частотой повторения $f=1/T$ (рис. 1, д). Отношение периода следования импульсов к длительности импульса называется скважностью: $q=T/\tau$.

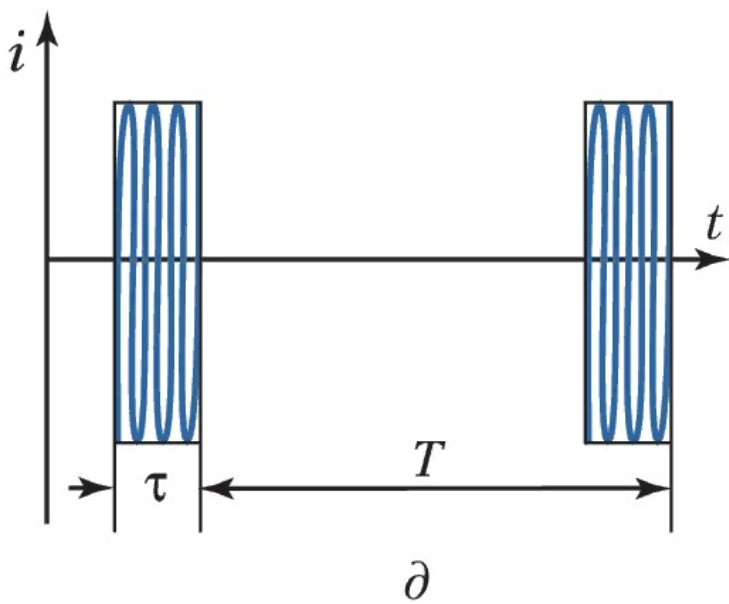


Рис. 2. Электрические импульсы разной формы: ∂ – радиоимпульс; τ – длительность импульса; i – ток; T – период; t – время.

Реальный видеоимпульс может иметь довольно сложную форму (рис. 2), которая характеризуется амплитудой A , длительностью $\tau_{и}$, длительностью фронта $\tau_{ф}$ и спада $\tau_{с}$, скосом вершины ΔA (выражается в % от A).

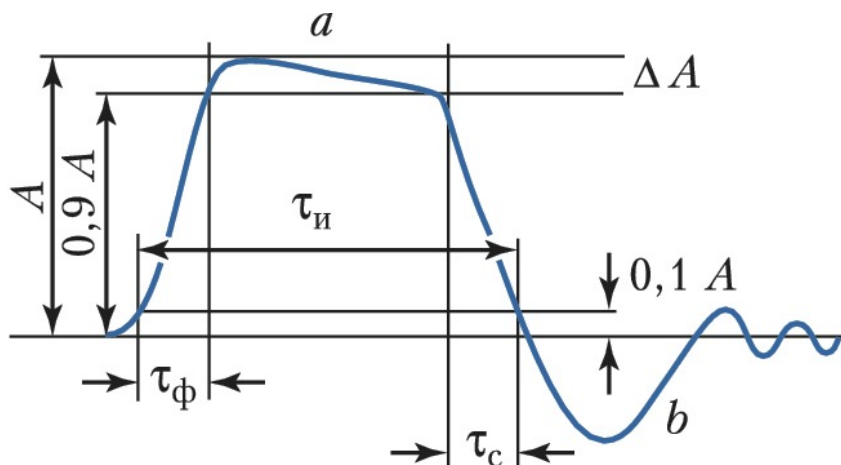


Рис. 3. Видеоимпульс и его основные характеристики: A – амплитуда; a – вершина; ΔA – скос вершины; b – хвост; $\tau_{ф}$ – длительность фронта; $\tau_{и}$ – длительность импульса; $\tau_{с}$ – длительность спада.

В курсовой работе будем программировать в Scilab как одиночные импульсы, так и последовательности импульсов. Импульсы будут как без заполнения (видеоимпульсы), так и с заполнением (радиоимпульсы). И те и другие относятся к классу радиотехнических сигналов. Более подробно с теорией радиотехнических сигналов можно ознакомиться в учебниках Гоноровского И.С. «Радиотехнические цепи и сигналы» [4] и Баскакова С.И. «Радиотехнические цепи и сигналы. Руководство к решению задач» [5]. Найдите эти учебники и всегда держите под рукой — это классика!

3. ПРОГРАММИРОВАНИЕ ВИДЕОИМПУЛЬСОВ

Система компьютерной алгебры Scilab ведет свою историю с 1994 года. На момент написания методических рекомендаций Scilab — это бесплатная программа с открытым кодом для инженеров и ученых. Разработкой и поддержкой Scilab занимается группа французских разработчиков из компании ESI Group. В состав группы разработчиков входят инженеры и ученые в области математики и компьютерных наук, работающие в высших учебных заведениях. Программа доступна для скачивания по адресу <https://www.scilab.org/download>. Примеры программирования будут приведены для версии 6.0.1. Доступны для установки версии под современные версии операционных систем Windows, Linux и Mac OS. Также поддерживается облачная версия, работа в которой осуществляется через веб-интерфейс. Интерфейс программы не русифицирован. Основной Wiki-ресурс по функциям программы находится по адресу <https://ru.smath.info/wiki/>. Форум пользователей находится по адресу <https://en.smath.info/forum/>. Для понимания принципов работы с математическими пакетами будет полезно учебное пособие Сильвашко С.А., Фролов С.С. "Программные средства компьютерного моделирования элементов и устройств электроники" [6]. Это пособие доступно для студентов в электронной библиотеке университета.

Итак, приступим к программированию. В радиотехнике основными представлениями сигналов являются **временное** (зависимость амплитуды сигнала от времени) и **спектральное** (зависимость амплитуды и/или фазы гармонических составляющих сигнала от частоты). В рамках курсовой работы будем рассматривать **только временное представление**.

Начнем с прямоугольного видеоимпульса, зададим его амплитуду и длительность. Длительность импульса — это время, в течение которого импульс существует. Однако в математических пакетах нет переменной «время». Поэтому будем программировать модель сигнала, которая позволит получить временное представление сигнала в виде массива его отсчетов.

Следует понимать, что модель в виде массива — это дискретное представление сигнала. Т.е. описываем значения сигнала в каких-то конкретных точках (моментах времени). Между этими точками сигнал в модели не существует. Поэтому очень важно определиться, насколько точно модель будет повторять моделируемый сигнал. Как правило, для последующей обработки и отображения достаточно 10000 точек [7, 8]. При моделировании необходимо соблюдать баланс между количеством точек модели, количеством точек на период следования сигнала, и соответствием одной дискреты модели реальному времени. Изначально следует исходить из того, что модель сигнала выглядит на графике достоверно приблизительно при 10 точках модели на период следования сигнала. Затем методом подбора можно определить остальные параметры. На рис.4 показан один период гармонического колебания, построенный по 25 точкам.

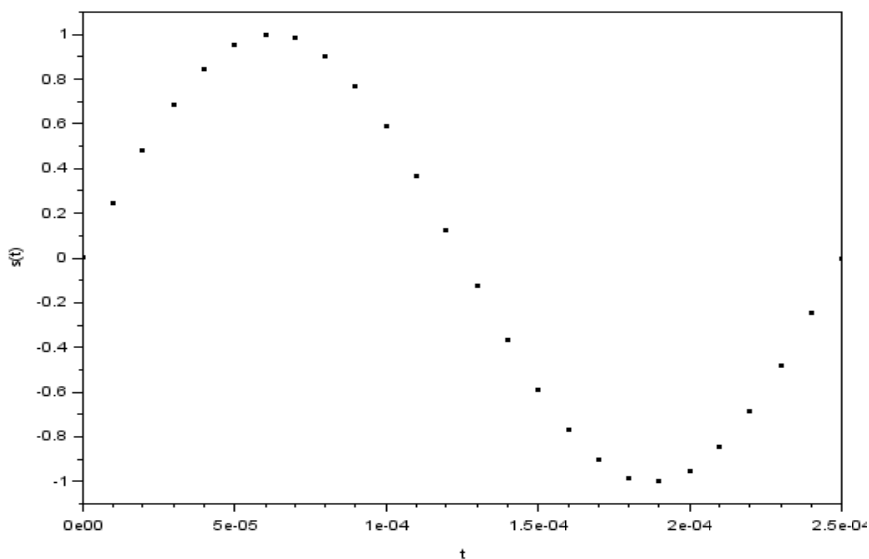


Рис.4. Один период гармонического колебания

Если в программе предполагается использовать и видео- и радиоимпульсы, то при выборе количества точек модели (дискретности модели) следует исходить из корректности представления радиоимпульса. Следует задать некоторое количество точек, необходимых для построения радиоимпульса, и по этим точкам строить видеоимпульс. Опишем начальные условия из задания и зададим количество точек модели. Листинг программы показан на рис.5.

Рассмотрим введенные переменные. Во второй строке кода на рис. 5 задана длительность одной дискретности Td в секундах. Одна дискретность — это один прямоугольный видеоимпульс, соответствует одной логической единице (или логическому нулю). В зависимости от практической реализации, в реальных устройствах логическому нулю может соответствовать не ноль, а импульс отрицательной полярности. В рамках этой работы логическому нулю будет соответствовать отсутствие напряжения на эфире. Мы будем кодировать первые буквы своего полного имени по таблице ASCII. Например, первые буквы полного имени автора задания — *КОВ* в кириллической раскладке. Смотрим по таблице ASCII, какому коду соответствует каждая буква, и переводим в двоичный код. Каждый знак таблицы кодируется одним байтом (8 бит). Последовательно записываем код трех букв ($3 \times 8 = 24$ бита) в переменную-вектор **Code** (строка 9 на рис. 5). Количество дискрет — 24. Если будет другой вектор **Code** — то количество дискрет изменится. Количество дискрет будет храниться в переменной **Nd**. Чтобы не переписывать код программы, зададим значение этой переменной как количество столбцов вектора **Code** (строка 10 на рис.5). Таблица ASCII для кириллического сегмента показана на рис. 6.

Последовательность прямоугольных импульсов (24 бита) — это пачка. Пачка может передаваться несколько раз, например для исключения ошибок передачи. Зададим количество пачек $Np=2$ (строка 11 на рис.5). Отношение периода следования импульса (пачки импульсов) к его длительности называется скважностью q . Скважность задается в строке 12 кода на рис.5.

В строках 13-16 заданы две частоты и две начальных фазы. Эти параметры будут использоваться при формировании радиоим-

пульсов. Переменные с индексом 1 используются формирования дискрет с логической 1, а с индексом 2 — логического нуля.

Длительность T_p и период следования Per пачки импульсов заданы в строках 17 и 18. Длительность пачки задана как произведение длительности одной дискреты на количество дискрет, период следования пачки импульсов выражен через длительность пачки и скважность. Так как скважность у нас по условию равна 1, то период следования пачки импульсов будет равен длительности пачки. Однако в общем случае скважность не равна единице, и поэтому следует программировать универсальный код.

```

Курсовая РСК.sce
1 //НАЧАЛЬНЫЕ УСЛОВИЯ
2 Td=1*10^-3//длительность одной дискреты
3 //Кодовая последовательность
4 //КОВ - кодовая последовательность, 3 буквы (XX) h=
5 //3 буквы (YYYY.YYYY) b.бит = 24.бит
6 //Буква К: 138d=8Ah=10001010b
7 //Буква О: 142d=8Eh=10001110b
8 //Буква В: 130d=80h=10000010b
9 Code=[1,0,0,0,1,0,1,0,1,0,0,0,1,1,1,0,1,0,0,0,0,1,0]
10 Nd=length(Code(1,:))//количество дискрет в пачке
11 Np=2//Количество пачек импульсов
12 q=1//скважность
13 f1=4*10^3//несущая частота, частота дискреты 1
14 f2=8*10^3//частота дискреты 0
15 phi1=0//фаза дискреты 1
16 phi2=%pi//фаза дискреты 0
17 Tp=Nd*Td//длительность пачки
18 Per=Tp*q//период следования пачки импульсов
19 Nt=100//количество точек моделирования на одну дискрету сигнала
20 ts=[0:Nt*Nd*Np*q]//вектор точек моделирования
21 t=ts*Td/Nt//ось времени, вектор с учетом "веса" Td/Nt одной точки модели

```

Рис. 5. Начальные условия

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	А	160	A0	а	192	C0	Л	224	E0	р
129	81	Б	161	A1	б	193	C1	л	225	E1	с
130	82	В	162	A2	в	194	C2	Т	226	E2	т
131	83	Г	163	A3	г	195	C3	т	227	E3	у
132	84	Д	164	A4	д	196	C4	—	228	E4	ф
133	85	Е	165	A5	е	197	C5	т	229	E5	х
134	86	Ж	166	A6	ж	198	C6	т	230	E6	ц
135	87	З	167	A7	з	199	C7	т	231	E7	ч
136	88	И	168	A8	и	200	C8	т	232	E8	ш
137	89	Й	169	A9	й	201	C9	т	233	E9	щ
138	8A	К	170	AA	к	202	CA	т	234	EA	ъ
139	8B	Л	171	AB	л	203	CB	т	235	EB	ы
140	8C	М	172	AC	м	204	CC	т	236	EC	ь
141	8D	Н	173	AD	н	205	CD	т	237	ED	э
142	8E	О	174	AE	о	206	CE	т	238	EE	ю
143	8F	П	175	AF	п	207	CF	т	239	EF	я
144	90	Р	176	B0	т	208	D0	т	240	F0	Ё
145	91	С	177	B1	т	209	D1	т	241	F1	ё
146	92	Т	178	B2	т	210	D2	т	242	F2	Є
147	93	У	179	B3	т	211	D3	т	243	F3	е
148	94	Ф	180	B4	т	212	D4	т	244	F4	ї
149	95	Х	181	B5	т	213	D5	т	245	F5	і
150	96	Ц	182	B6	т	214	D6	т	246	F6	Ў
151	97	Ч	183	B7	т	215	D7	т	247	F7	ў
152	98	Ш	184	B8	т	216	D8	т	248	F8	°
153	99	Щ	185	B9	т	217	D9	т	249	F9	·
154	9A	Ъ	186	BA	т	218	DA	т	250	FA	·
155	9B	Ы	187	BB	т	219	DB	т	251	FB	√
156	9C	Ь	188	BC	т	220	DC	т	252	FC	№
157	9D	Э	189	BD	т	221	DD	т	253	FD	□
158	9E	Ю	190	BE	т	222	DE	т	254	FE	■
159	9F	Я	191	BF	т	223	DF	т	255	FF	

Рис. 6. Таблица кодировки IBM CP866 (ASCII)

В строке 20 на рис.5 задан вектор точек моделирования *ts*. Элементы вектора изменяются от нуля до максимального количества с шагом один. Количество точек в векторе (количество точек моделирования, точек модели сигнала) определяется как произведение константы **Nt** на количество дискрет **Nd**, количество пачек импульсов **Np** и скважность **q**. После того, как модель заработает, само-

стоятельно определите, какое количество точек моделирования будет приходиться на одно высокочастотное колебание и сравните с рис.4. Исследуйте, что произойдет при уменьшении/увеличении количества точек в 10 раз. Для удобства работы с графиками сигналов необходимо по горизонтальной оси откладывать отметки времени. Для этого необходимо сделать нормировку вектора ts с помощью нормирующего коэффициента Td/Nt .

Для формирования пачки прямоугольных импульсов и последовательности пачек используем цикл со счетчиком. На рис.7 показан код для формирования последовательность прямоугольных импульсов *Sam* с помощью вложенных циклов *For* и функции *ЕСЛИ (if)*. Обратите внимание на синтаксис функции *if*. Одновременное соответствие значения переменной t двум интервалам задается через логическое умножение (знак $\&$). Последовательность прямоугольных импульсов формируется как сумма значений вспомогательной переменной s . Вспомогательная переменная s принимает значение вектора *Code* для текущей дискреты, если точка сигнала находится в пределах дискреты, или нуля в обратном случае. Внешний цикл последовательно «пробегает» по точкам модели сигнала, а два внутренних по дискретам и периодам следования пачки. Внутренние циклы повторяются для каждой точки модели. Внутри циклов проверяется условие нахождения временной точки модели в пределах конкретной дискреты. Если условие нахождения внутри дискреты выполняется, то точке модели присваивается соответствующее значение информационного сообщения (24-х битного кода) из переменной *Code*. Формирование пачки импульсов — основополагающий для дальнейшего программирования и наиболее сложный момент в работе. Для более глубокого понимания студентами работы циклов необходимо **дополнительно самостоятельно** реализовать формирование вектора *Sam* с помощью конструкции *while* и нарисовать блок-схему алгоритма формирования вектора модели сигнала.

На рис.8 показаны две пачки последовательности прямоугольных видеоимпульсов из 24 бит для букв КОВ. По оси « x » отложены отсчеты времени. На рис.9 показан код формирования рис.8.

```

47 //Формирование пачки видеоимпульсов
48 //Sam=0
49 for i=1:length(ts(1,:))//Перебор по элементам вектора сигнала
50 ... sm1=0//Обнуление промежуточного счетчика
51 ... for k=0:(Np-1)//Перебор по пачкам
52 ...     for n=0:(Nd-1)//Перебор по дискретам
53 ...         //Текущий момент времени внутри дискреты?
54 ...         if ((Tp*n/Nd+Per*k)<t(i)) & (t(i)<(Tp/Nd+Tp*n/Nd+Per*k)) then
55 ...             //Да. Сигнал = Элемент кода (вектор Code)
56 ...             s=Code(n+1)
57 ...         else//Нет. Дискрета закончилась.
58 ...             s=0
59 ...         end
60 ...         sm1=sm1+s//Хранение значения сигнала текущей дискреты
61 ...     end
62 ... end
63 ... Sam(:,i)=sm1//Формирование вектора сигнала
64 end

```

Рис. 7. Формирование пачки видеоимпульсов

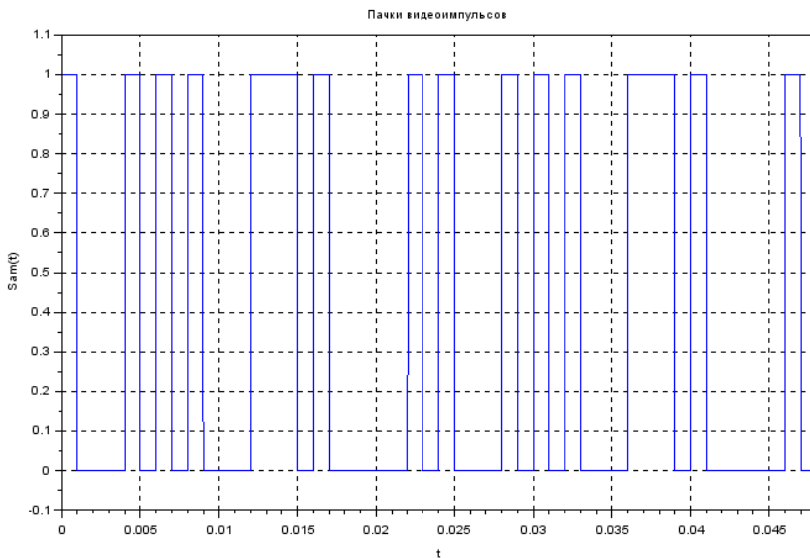


Рис. 8. Две пачки последовательности прямоугольных видеоимпульсов из 24 бит для букв КОВ

```

66 //Построение графика
67 clf() //Очистка графической формы
68 //Определение значений по осям и диапазонов построения
69 plot2d(t, Sam, 2, rect=[0, -1, 2*Per, 1.1])
70 //Описание параметров рисунка вручную через дескриптор
71 g=get('current_axes'); ..... //Получение дескриптора осей
72 g.box="on" ..... //Рамка рисунка
73 g.title.text='Пачки видеоимпульсов'; ..... //Название графика
74 g.x_label.text='t'; ..... //Подпись оси абсцисс (X)
75 g.y_label.text='Sam(t)'; ..... //Подпись оси ординат (Y)
76 g.grid=[0 0]; ..... //Цвет и стиль сетки по осям X и Y

```

Рис. 9. Код формирования последовательности прямоугольных видеоимпульсов

Последовательность прямоугольных видеоимпульсов является базовой для дальнейшего программирования. Измените элементы или длину вектора *Code* и посмотрите, что произойдет с графиком. Используем эту последовательность в качестве модулирующей для формирования радиоимпульсов с амплитудной, фазовой и частотной модуляцией.

4. РАДИОИМПУЛЬСЫ С АМПЛИТУДНОЙ МОДУЛЯЦИЕЙ

Для формирования радиоимпульсов необходимо запрограммировать несущее колебание с параметрами, заданными в условии на рис.5. Запрограммируем вектор гармонического колебания с частотой f_1 . На рис.10 показаны код гармонического колебания *Scar* и его график для первых 8-ми бит. Определите по графику, соответствует ли частота сформированного колебания условию. На этом же графике показан модулирующий сигнал *Sam*.

Формирование амплитудно-модулированного сигнала заключается в перемножении опорного колебания и модулирующего. При этом на временных отрезках, соответствующих логической 1 в опорном колебании, будет радиоимпульс, а на отрезках логического 0 — колебания не будет. Создадим в программе отдельную переменную *Sam2*, значение которой равно произведению *Scar * Sam*, и построим график этой функции. На рис.11 показан код, и график нашего сообщения для первых восьми бит. На этом программирование радиоимпульсов с амплитудной модуляцией закончено. Следует отметить, что модуляция, при которой величина модулируемого параметра изменяется скачкообразно, называется *манипуляцией*. В рамках настоящей курсовой работы понятия «модуляция» и «манипуляция» являются равнозначными.

```

87 //Формирование несущего колебания частоты f1
88 Scar=sin(2*pi*f1*t+phi1)
89 //Построение графика
90 clf() //Очистка графической формы
91 //Определение значений по осям и диапазонов построения
92 plot2d(t, Sam, 2, rect=[0, -1, 8*Td, 1.1])
93 plot2d(t, Scar, 5, rect=[0, -1.1, 8*Td, 1.1])
94 //Описание параметров рисунка вручную через дескриптор
95 g=get('current_axes'); ..... //Получение дескриптора осей
96 g.box="on" ..... //Рамка рисунка
97 g.title.text='Модулирующий сигнал и несущее колебание'; //Название графика
98 g.x_label.text='t'; ..... //Подпись оси абсцисс (X)
99 g.y_label.text='Sam(t), Scar(t)'; ..... //Подпись оси ординат (Y)
100 g.grid=[0 0]; ..... //Цвет и стиль сетки по осям X и Y

```

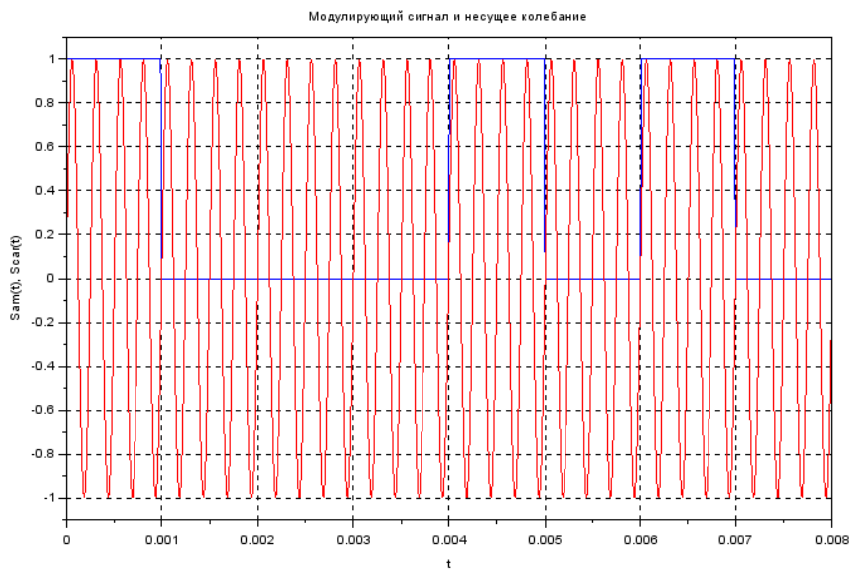


Рис. 10. Код и график несущего гармонического колебания *Scar* и модулирующего сигнала *Sam*

```

102 //Формирование АМ-колебания частоты f1
103 for i=1:length(ts(1,:))//Перебор по элементам вектора сигнала
104     Sam2(:,i)=Sam(:,i)*Scar(:,i)//Формирование вектора сигнала
105 end
106 //Построение графика
107 clf()//Очистка графической формы
108 //Определение значений по осям и диапазонов построения
109 plot2d(t, Sam2, 2, rect=[0, -1.1, 8*Td, 1.1])
110 //Описание параметров рисунка вручную через дескриптор
111 g=get('current_axes'); .....//Получение дескриптора осей
112 g.box="on" .....//Рамка рисунка
113 g.title.text='Радиопульсы с амплитудной модуляцией'; //Название графика
114 g.x_label.text='t'; .....//Подпись оси абсцисс (X)
115 g.y_label.text='Sam2(t)'; .....//Подпись оси ординат (Y)
116 g.grid=[0 0]; .....//Цвет и стиль сетки по осям X и Y

```

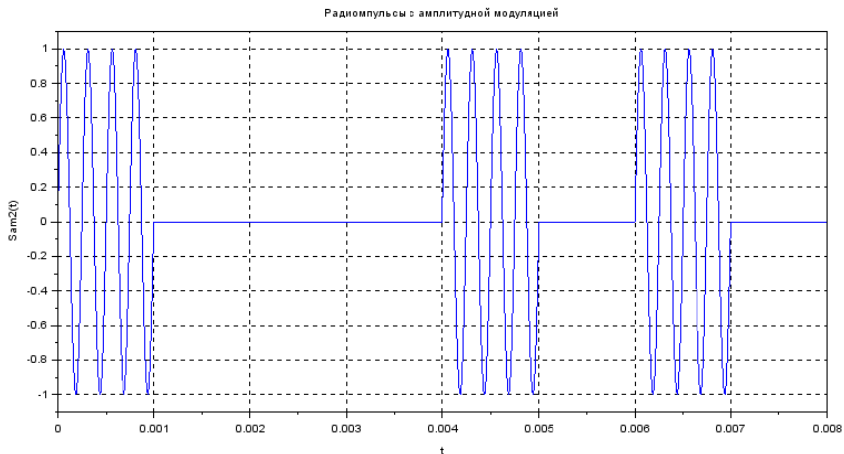


Рис. 11. Радиопульсы с амплитудной модуляцией

5. РАДИОИМПУЛЬСЫ С ФАЗОВОЙ МОДУЛЯЦИЕЙ

Фазовая модуляция подразумевает изменение начальной фазы высокочастотного гармонического колебания в момент начала импульса по какому-либо закону. Простейший случай — начальная фаза радиоимпульса логической 1 — 0 градусов, а логического 0 — 180 градусов. На практике применяются более сложные виды модуляции, когда количество разных начальных фаз больше, чем две, и начальная фаза последующей дискретности зависит от состояния предыдущей. Запрограммируем простейший случай.

Т.к. начальных фаз всего две, и это 0 и 180 градусов, возможны два способа формирования. Первый способ заключается в том, что в кодовой последовательности видеоимпульсов значению логического 0 ставится в соответствие не нулевая амплитуда, а отрицательная. Т.е. логическая 1 кодируется прямоугольным видеоимпульсом с амплитудой «+1», а логический 0 прямоугольным видеоимпульсом с амплитудой «-1». Затем кодовая последовательность перемножается с несущим колебанием и получается пачка радиоимпульсов с фазовой модуляцией. Формирование кодовой последовательности для фазовой модуляции программируется аналогично функции *Sam*. Для этого необходимо задать новый вектор *Code2*, равный вектору *Code*. Перекодировка информационного сообщения *Code2* выполняется в цикле со счетчиком *For* при помощи функции *ЕСЛИ (if)*. Листинг программы формирования кодовой последовательности и модулирующих импульсов для фазовой модуляции показан на рис.12. Как видно на рисунке, для формирования модулирующих видеоимпульсов используется уже известный код, аналогичный коду на рисунке 7.

Для первых 8-ми бит на рис.13 показаны несущее колебание и кодовая последовательность видеоимпульсов для фазовой модуляции, а на рис.14 код формирования их произведения и график получившегося сигнала.

```

121 //Фазовая модуляция. Способ №1. Поворот на Пи
122 //Перекодировка вектора кодовой последовательности
123 Code2=Code
124 for i=1:Nd//Перебор по элементам вектора Code
125 ... if Code(:,i)==1 then
126 ..... Code2(:,i)=1//Формирование вектора Code2
127 ... else
128 ..... Code2(:,i)=-1//Формирование вектора Code2
129 ... end//Конец цикла перекодировки
130 end
131 //Формирование пачки видеосигналов
132 //Sam=0
133 for i=1:length(ts(1,:))//Перебор по элементам вектора сигнала
134 ... sm1=0//Обнуление промежуточного счетчика
135 ... for k=0:(Np-1)//Перебор по пачкам
136 ..... for n=0:(Nd-1)//Перебор по дискретам
137 ..... //Текущий момент времени внутри дискреты?
138 ..... if ((Tp*n/Nd+Per*k)<t(i)) & (t(i)<(Tp/Nd+Tp*n/Nd+Per*k)) then
139 ..... //Да. Сигнал = Элемент кода (вектор Code)
140 ..... s=Code2(n+1)
141 ..... else//Нет. Дискрета закончилась.
142 ..... s=0
143 ..... end
144 ..... sm1=sm1+s//Хранение значения сигнала текущей дискреты
145 ..... end
146 ... end
147 ... Sam3(:,i)=sm1//Формирование вектора сигнала
148 end
149 //Формирование ФМ-колебания
150 for i=1:length(ts(1,:))//Перебор по элементам вектора сигнала
151 ... S1Phm(:,i)=Sam3(:,i)*Scar(:,i)//Формирование вектора сигнала
152 end

```

Рис. 12. Формирования кодовой последовательности и модулирующих импульсов для фазовой модуляции. Способ 1

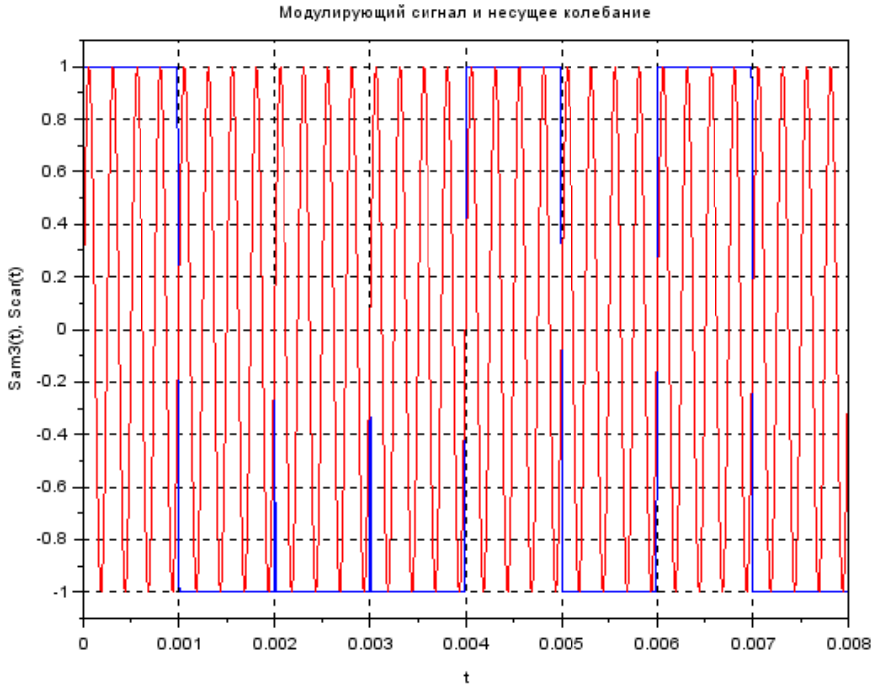


Рис. 13. График несущего гармонического колебания *Scar* и модулирующего сигнала *SIPhm*

Обратите внимание на начальную фазу гармонического колебания в начале каждого импульса. Проверьте правильность формирования сигнала.

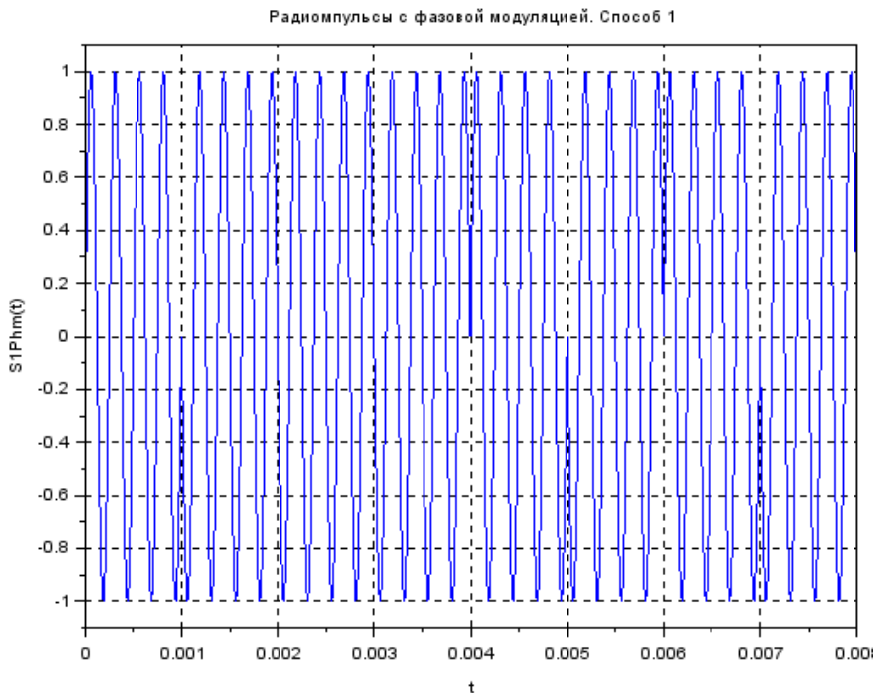


Рис. 14. Радиоимпульсы с фазовой модуляцией. Способ 1

Второй способ формирования радиоимпульсов с фазовой модуляцией является универсальным для любой начальной фазы, заданной заранее. Способ заключается в следующем. В цикле формируется вектор модулирующего сигнала *Sam3*. Вектор *Sam3* аналогичен вектору *Sam*. Далее в этом же цикле проверяется значение элемента вектора *Sam3* и формирование элемента вектора фазомодулированного сигнала *S2Phm*. Значениям *Sam3* больше 0 (т.е. точкам, соответствующим логическим 1 информационного вектора *Code*) соответствуют значения гармонического колебания с фазой *phi1*. В противном случае - с фазой *phi2*. Проверка условия организована таким образом, что на местах логического 0 и 1 сразу формируются гармонические колебания с нужной начальной фазой. Код программы и график для первых 8-ми бит показаны на рис.15.

```

181 //Фазовая модуляция. Способ №2
182 for i=1:length(ts(1,:)) //Перебор по элементам вектора сигнала
183     sm2=0 //Обнуление промежуточного счетчика
184     for k=0:(Np-1) //Перебор по пачкам
185         for n=0:(Nd-1) //Перебор по дискретам
186             //Текущий момент времени внутри дискреты?
187             if ((Tp*n/Nd+Per*k)<t(i)) & (t(i)<(Tp/Nd+Tp*n/Nd+Per*k)) then
188                 //Да. Сигнал = Элемент кода (вектор Code)
189                 s2=Code(n+1)
190             else //Нет. Дискрета закончилась.
191                 s2=0
192             end
193             sm2=sm2+s2 //Хранение значения сигнала текущей дискреты
194         end
195     end
196     Sam3(:,i)=sm2 //Формирование вектора сигнала
197     //Формирование ФМ-колебания
198     if Sam3(:,i)>0 then S2Phm(:,i)=sin(2*pi*f1*t(i)+phi1)
199     else S2Phm(:,i)=sin(2*pi*f1*t(i)-phi2)
200     end
201 end

```

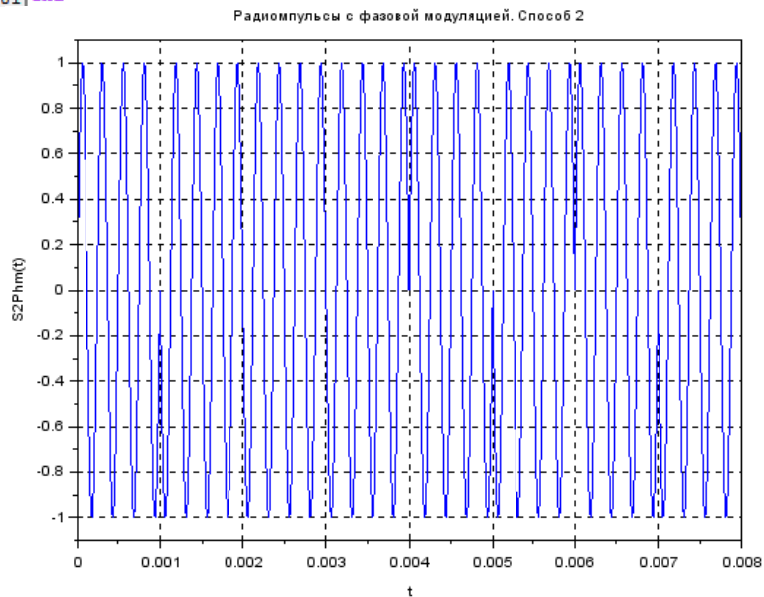


Рис. 15. Радиоимпульсы с фазовой модуляцией. Способ 2

Нарисуйте блок-схему алгоритма формирования вектора SPhm2 и приведите ее в работе. Это алгоритм мы будем использовать для формирования радиоимпульсов с частотной модуляцией. На рис.16 показаны первые две дискреты, сформированные Способом 1 и Способом 2. Какие есть различия на графиках?

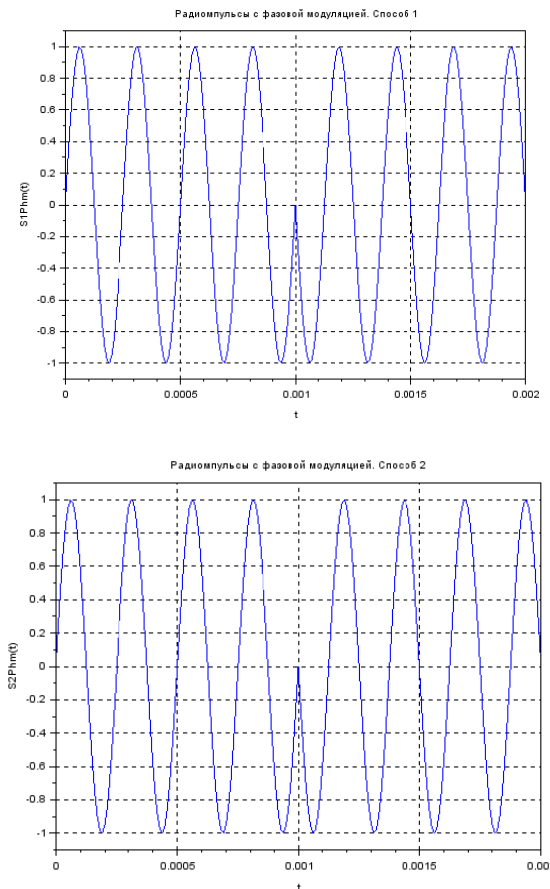


Рис. 16. Радиоимпульсы с фазовой модуляцией. Способ 1 и способ 2

6. РАДИОИМПУЛЬСЫ С ЧАСТОТНОЙ МОДУЛЯЦИЕЙ

При частотной модуляции логической 1 будет соответствовать радиоимпульс с частотой f_1 , а логическому 0 — радиоимпульс с частотой f_0 . Значение частот заданы на рис.5 в строках 13 и 14. Формирование пачки радиоимпульсов с частотной модуляцией аналогично формированию пачки с фазовой модуляцией. Отличие заключается в том, что вместо сигналов с разными фазами формируются сигналы с разными частотами. Код программы формирования радиоимпульсов с частотной модуляцией и график для первых 8-ми бит показаны на рис.17. Нарисуйте блок-схему алгоритма формирования вектора SFm и приведите ее в работе.

Алгоритм формирования импульсов с частотной модуляцией учитывает изменение начальной фазы сигнала также, как и при фазовой модуляции. В данном конкретном случае в этом нет необходимости. Учет фазы оставлен для сохранения единства подхода к формированию радиоимпульсов.

```

216 //Частотная модуляция
217 for i=1:length(ts(1,:)) //Перебор по элементам вектора сигнала
218     sm3=0 //Обнуление промежуточного счетчика
219     for k=0:(Np-1) //Перебор по пакетам
220         for n=0:(Nd-1) //Перебор по дискретам
221             //Текущий момент времени внутри дискреты?
222             if ((Tp*n/Nd+Per*k)<t(i)) & (t(i)<(Tp/Nd+Tp*n/Nd+Per*k)) then
223                 //Да. Сигнал.= Элемент кода (вектор Code)
224                 s3=Code(n+1)
225             else //Нет. Дискрета закончилась.
226                 s3=0
227             end
228             sm3=sm3+s3 //Хранение значения сигнала текущей дискреты
229         end
230     end
231     Sam4(:,i)=sm3 //Формирование вектора сигнала
232     //Формирование ЧМ-колебания
233     if Sam4(:,i)>0 then SFm(:,i)=sin(2*pi*f1*t(i)+phi1)
234     else SFm(:,i)=sin(2*pi*f2*t(i)+phi1)
235     end
236 end

```

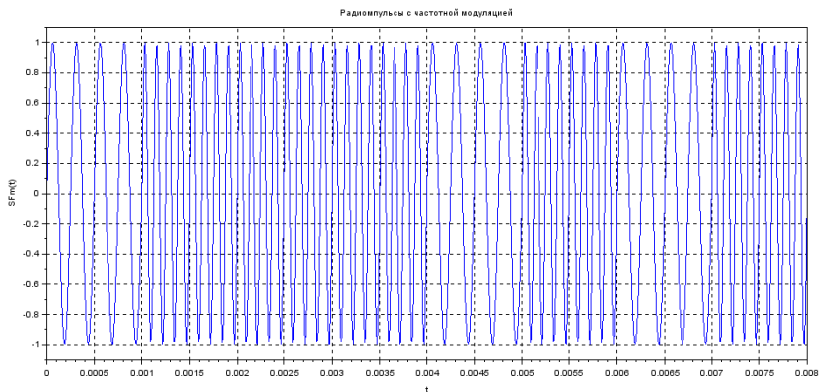


Рис. 17. Радиоимпульсы с частотной модуляцией

На рис.18 для иллюстрации различий в видах модуляции показаны первые два бита последовательности радиоимпульсов с частотной и фазовой модуляцией. Необходимо проверить по графикам частоты в каждом бите и сравнить с исходными данными.

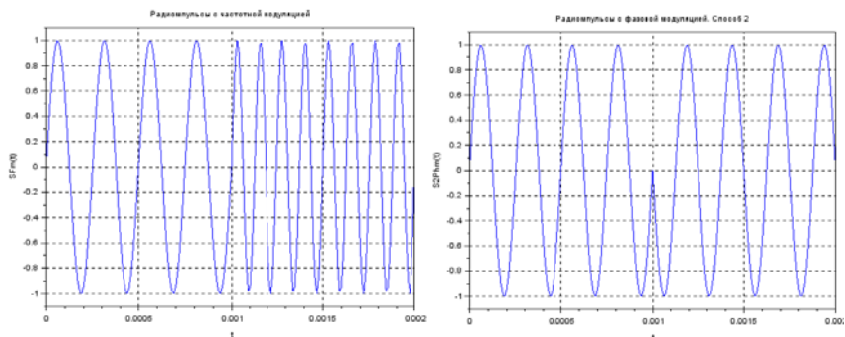


Рис. 18. Радиоимпульсы с частотной и фазовой модуляцией

На этом формирование сигналов закончено. Приступим к записи сигналов в файл и считыванию из файла в другие переменные.

7. ЗАПИСЬ РАДИОИМПУЛЬСОВ В ФАЙЛ

Для того чтобы можно было использовать полученные модели сигналов в других расчетах и в других пакетах моделирования, запишем полученные модели в файл. Т.к. мы запрограммировали временные модели сигналов, то в файл будем записывать два вектора: вектор отсчетов сигнала и вектор отсчетов времени. Каждому отсчету сигнала в файле будет соответствовать отсчет времени. Вектор отсчетов сигнала и вектор отсчетов времени необходимо объединить в один массив, и уже этот массив записать в файл. Объединение векторов происходит с помощью цикла со счетчиком. Запись в файл происходит при выполнении команды `csvWrite(Имя_массива_для_записи, "Имя_файла")`, а считывание `csvRead("Имя_файла")`. Объединим вектора в массив *SigFM*, запишем этот массив в файл *ЧМ.dat*. Затем выполним обратные дейст-

вия, построим график сигнала и сравним с рисунком 18. На рис.19 показан код программы.

```
252 //Объединение массивов сигнала и времени
253 //Столбец 1 -- сигнал, столбец 2 -- время
254 for i=1:length(ts(1,:))//Перебор по элементам вектора сигнала
255     for j=1:2//Перебор по столбцам объединенного массива
256         if j==1 then
257             SigFm(i,j)=SFm(1,i)
258         else
259             SigFm(i,j)=t(1,i)
260         end
261     end
262 end
263 //Запись сигнала в файл
264 csvWrite(SigFm,"ЧМ.dat")
265 //Чтение сигнала из файла
266 SigFm2=csvRead("ЧМ.dat")
267 //Определение размера считанного файла
268 [n1,n2]=size(SigFm2)
269 //Формирование вектора сигнала и времени из считанного файла
270 //Столбец 1 -- сигнал, столбец 2 -- время
271 for i=1:n1//Перебор по строкам считанного массива
272     for j=1:n2//Перебор по столбцам считанного массива
273         if j==1 then
274             SFm2(1,i)=SigFm2(i,j)
275         else
276             t2(1,i)=SigFm2(i,j)
277         end
278     end
279 end
```

Рис. 19. Код для записи массива в файл и считывания из файла в переменную

Файл с записанным сигналом можно посмотреть через стандартный Блокнот в Windows. На рис. 20 показан файл, открытый в Блокноте.

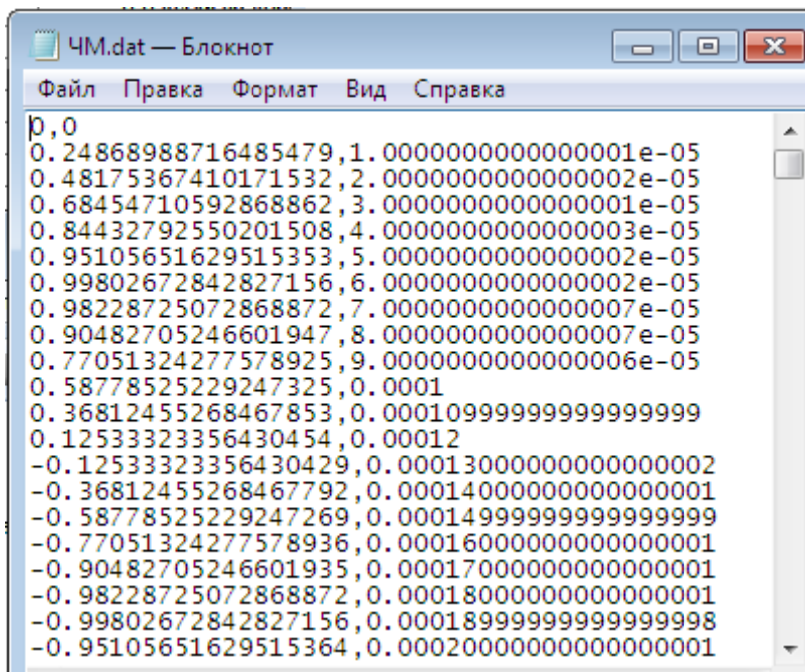
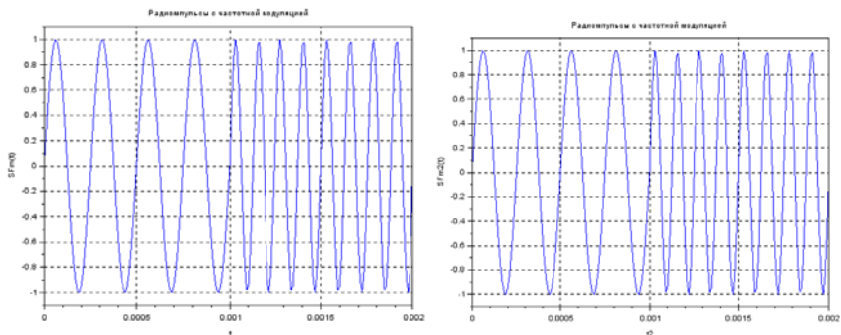


Рис. 20. Файл ЧМ.dat, открытый в Блокноте

На рис.21 показан исходный сигнал и сигнал, считанный из файла. Как видно на эпюрах, сигналы одинаковые, значит все сделано правильно. Обратите внимание на переменную оси времени на эпюре считанного из файла сигнала.

Запишите самостоятельно модели всех сигналов в разные файлы и считайте их в переменные в Scilab. Постройте графики этих сигналов и сравните с исходными сигналами.



ЗАКЛЮЧЕНИЕ

После того, как все расчеты и программирование выполнены, необходимо заполнить пояснительную записку смыслом и правильно оформить в текстовом редакторе. Оформление курсовой работы должно соответствовать государственным требованиям, предъявляемым к оформлению документации [9-12]. Внимательно прочитайте следующие строки и следуйте им при оформлении курсовой работы.

Содержание курсовой работы отражает ход мысли и способности студента. Оформление работы демонстрирует отношение студента к процессу обучения, преподавателю и лично к себе. Все разделы работы последовательно раскрывают этапы достижения цели. В работе содержится только информация, непосредственно связанная с этапами работы. Все иллюстрации раскрывают содержание выполненных действий. Текстовые пояснения помогают студенту отвечать на вопросы на защите, а не порождают новые уточняющие вопросы. Выводы по работе содержат личные впечатления студента и констатируют достижение цели работы, а не перечисляют очевидные факты. Переписывать теорию из учебника в курсовую работу не нужно.

Удачи!

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сигнал // Большая российская энциклопедия. Том 30. Москва, 2015, с. 144-145.
2. *Ештокин Е.Н.* Импульс электрический // Большая российская энциклопедия. Том 11. Москва, 2008, стр. 163.
3. Несущая частота // Большая российская энциклопедия. Том 22. Москва, 2013, с. 523.
4. *Гоноровский И.С.* Радиотехнические цепи и сигналы: Учебник для ВУЗов. — 4-е изд., перераб. и доп. М.: Радио и связь, 1986, 512 с.
5. *Баскаков С.И.* Радиотехнические цепи и сигналы. Руководство к решению задач. М.: Высшая школа, 2002, 214 с.
6. *Сильвашко С.А.* Программные средства компьютерного моделирования элементов и устройств электроники: учебное пособие / С.А. Сильвашко, С.С. Фролов. - Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Оренбургский государственный университет», Кафедра промышленной электроники и информационно-измерительной техники. Оренбург: ОГУ, 2014, 170с.
7. *Антипенский Р.В.* Разработка моделей первичных сигналов в программной среде MathCAD // Компоненты и технологии, 2007, № 3, с.162-165.
8. *Антипенский Р.В.* Разработка моделей первичных сигналов в программной среде MathCAD // Компоненты и технологии, 2007, № 6, с.147-151.
9. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления. Минск, 2001, 22 с.
10. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам. Минск, 1995, 26 с.
11. ГОСТ Р 7.0.5-2008. Библиографическая ссылка. Общие требования и правила составления. М.: Стандартинформ, 2005, 20 с.
12. ГОСТ 19.701-90 (ИСО 5807-85). Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. М.: Стандартинформ, 2010, 22 с.

СОДЕРЖАНИЕ

Введение.....	3
1. Исходные данные	4
2. Понятие импульсного сигнала. Параметры импульса.....	5
3. Программирование видеоимпульсов.....	9
4. Радиоимпульсы с амплитудной модуляцией.....	17
5. Радиоимпульсы с фазовой модуляцией.....	20
6. Радиоимпульсы с частотной модуляцией.....	26
7. Запись радиоимпульсов в файл.....	28
Заключение	31
Библиографический список.....	32

ИНФОРМАТИКА
ПРОГРАММИРОВАНИЕ
РАДИОТЕХНИЧЕСКИХ СИГНАЛОВ
В SCILAB

*Методические указания к курсовой работе
для студентов специальности 11.05.01*

Сост.: *О.В. Косарев, Е.Г. Водкайло, А.Б. Маховиков*

Печатается с оригинал-макета, подготовленного кафедрой
информатики и компьютерных технологий

Ответственный за выпуск *Е.Г. Водкайло*

Лицензия ИД № 06517 от 09.01.2002

Подписано к печати 10.06.2019. Формат 60×84/16.

Усл. печ. л. 1,9. Усл.кр.-отт. 1,9. Уч.-изд.л. 1,5. Тираж 50 экз. Заказ 535. С 191.

Санкт-Петербургский горный университет
РИЦ Санкт-Петербургского горного университета
Адрес университета и РИЦ: 199106 Санкт-Петербург, 21-я линия, 2