

Федеральное агентство по образованию
Государственное образовательное учреждение высшего профессионального
образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра компьютерных систем в управлении и проектировании (КСУП)

В. П. Коцубинский, Д. А. Звонков, А. В. Ямшанов

**Информационные системы, сети и телекоммуникации
для студентов специальности
220201 – «Управление и информатика в технических системах»
230104 — «Системы автоматизированного проектирования»**

Учебно-методическое пособие

Томск – 2012

Коцубинский В. П., Звонков Д.А., Ямшанов А. В.

Информационные системы, сети и телекоммуникации (для студентов специальности 220201 – Управление и информатика в технических системах и 230104 – Системы автоматизированного проектирования) : учеб. методич. пособие / В. П. Коцубинский, Д. А. Звонков, А. В. Ямшанов. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2012. – 142 с.

В пособии представлены основные теоретические сведения, необходимые для выполнения комплекса лабораторных работ, задания на лабораторные работы и примеры выполнения по дисциплинам «Сети ЭВМ и телекоммуникации» для специальности 230104 — «Системы автоматизированного проектирования» и «Информационные сети и телекоммуникации» для специальности 220201 – «Управление и информатика в технических системах».

© Коцубинский В. П. ., 2012

© Звонков Д.А. ., 2012

© Ямшанов А. В., 2012

© Том. гос. ун-т систем упр. и
радиоэлектроники, 2012

Содержание

Лабораторная работа №1. Исследование спектров амплитудно-импульсной и широтно-импульсной модуляции	7
1.1 Особенности импульсной модуляции	9
1.2 Анализ спектров импульсной модуляции	11
1.2.1 Спектр сигнала АИМ	11
1.2.2 Спектр сигнала ШИМ	13
1.3 Основы теории разделения сигналов	15
1.3.1 Блок-схема многоканальной системы	15
1.3.2 Временной способ разделения сигналов	17
1.4 Реализация в среде MathCAD	20
1.4.1 Реализация АИМ в среде MathCAD	20
1.4.2 Реализация ШИМ в среде MathCAD	21
1.4.3 Пример реализации АИМ и ШИМ	22
1.5 Контрольные вопросы	24
1.6 Задание	24
1.7 Литература	25
Лабораторная работа №2. Частотно-фазовая модуляция и демодуляция	25
2.1 Модуляция сигнала	25
2.1.1 Фазовая модуляция	25
2.1.2 Фазовая манипуляция	26
2.1.3 Частотная модуляция	27
2.1.4 Частотная манипуляция	29
2.2 Демодуляция сигнала	30
2.2.1 Краткие сведения из теории оптимального приёма сигналов	30
2.2.2 Построение корреляционного приёмника	32
2.2.3 Цифровые частотные демодуляторы	35
2.2.4 Построение частотного приёмника	37
2.3 Контрольные вопросы	39
2.4 Задание	40
2.5 Литература	40
Лабораторная работа №3. Помехозащищающие коды и механизмы избыточности	40
3.1 Цепной код	43

3.2	Применение цепного кода на практике	45
3.3	Контрольные вопросы	50
3.4	Задание	50
3.5	Литература	51
	Лабораторная работа №4. Многоканальные системы передачи данных	51
4.1	Принцип частотного объединения и разделения каналов	54
4.2	Временное разделение каналов	56
4.3	Принцип частотного объединения и разделения каналов на практике.....	60
4.4	Принцип временного объединения и разделения каналов на практике....	65
4.5	Контрольные вопросы	68
4.6	Задание	69
4.7	Литература	69
	Лабораторная работа №5. Линейное предсказание при анализе речевого сигнала	69
5.1	Модель предсказания.....	70
5.2	Автокорреляционный метод	74
5.3	Ковариационный метод	76
5.4	Вычисление коэффициента усиления модели.....	78
5.5	Математическое моделирование	82
5.6	Задание	92
5.7	Литература	92
	Лабораторная работа №6. Введение в IP-сети	92
6.1	Протокол IP.....	93
6.1.1	Протокол IP версии 4.....	93
6.1.2	Протокол IP версии 6.....	93
6.2	IP-Пакет.....	94
6.2.1	IP-Пакет версии 4 (IPv4).....	94
6.2.2	IP-Пакет версии 6 (IPv6).....	95
6.3	Протокол ICMP.....	96
6.3.1	Технические подробности.....	96
6.3.2	Использование ICMP-сообщений.....	97
6.3.3	Формат пакета ICMP.....	98
6.4	Утилита Ping	98
6.4.1	Этимология	99
6.4.2	История.....	99

6.5	Утилита Traceroute	100
6.5.1	Принцип работы traceroute	100
6.6	Утилита WHOIS	101
6.6.1	Централизованная и распределённая модели	102
6.6.2	Интернационализация протокола.....	103
6.7	Утилита Wireshark.....	103
6.8	Контрольные вопросы	105
6.9	Задание	105
6.10	Литература	106
	Лабораторная работа №7. Система доменных имён	106
7.1	Описание протокола DNS	106
7.1.1	Ключевые характеристики DNS	107
7.1.2	Ключевые понятиям DNS:	108
7.1.3	Типы запросов в DNS	111
7.1.4	Записи DNS	113
7.1.5	Балансировка нагрузки	114
7.1.5.1	Round robin DNS.....	115
7.1.5.2	Географическое распределение нагрузки.....	117
7.2	Утилита Wireshark.....	117
7.3	Утилита Nslookup	118
7.3.1	Поиск данных различных типов	118
7.3.2	Получение данных с удалённого сервера имён напрямую.....	119
7.4	Утилита dig	119
7.5	Контрольные вопросы	120
7.6	Задание	120
7.7	Литература	121
	Лабораторная работа №8. Протокол доставки электронной почты.....	121
8.1	Описание протокола SMTP	121
8.2	Пример отправки письма клиентом	123
8.2.1	Подключаемся к Unix-серверу;.....	123
8.2.2	Выясняем адрес SMTP сервера;.....	123
8.2.3	Подключаемся к SMTP серверу.....	124
8.2.4	Передаём серверу команду HELO и наш IP адрес.....	125
8.2.5	Указываем отправителя	125

8.2.6	Указывает получателя	125
8.2.7	Передаём заголовок письма	125
8.2.8	Передаём тело письма;	126
8.2.9	Отключаемся от SMTP сервера.	127
8.2.10	Послесловие	127
8.3	Задание	127
Приложение А. Варианты исходных данных для лабораторной работы №1 ...		129
Приложение Б. Варианты исходных данных для лабораторной работы №2....		131
Приложение В. Варианты исходных данных для лабораторной работы №4. ..		133
Приложение Г. Варианты исходных данных для лабораторной работы №6.		
Список адресов для исследования		135
Приложение Д. Типы пакетов ICMP (полный список)		137

Лабораторная работа №1. Исследование спектров амплитудно-импульсной и широтно-импульсной модуляции

Для построения системы передачи информации необходимо знать свойства сигналов, способы представления и преобразования сигналов различной формы, чтобы уметь передавать их по каналам связи с наименьшими потерями и представлять в наиболее удобной форме. Эти свойства могут быть определены и наглядно представлены, если воспользоваться математическими моделями сигналов в форме, удобной для анализа и физических выводов. В информационных процессах сигналы могут иметь разную форму. Математические модели сигналов позволяют рассчитывать характеристики сигналов во временной и частотной областях и выбирать по этим характеристикам параметры каналов связи для передачи сигналов с минимальными искажениями. Передаваемые сигналы могут иметь разную форму, которую отобразить математической моделью не всегда удобно или возможно. Поэтому для описания свойств сигналов их модели декомпозируют на более простые составляющие, допускающие преобразования, известными в математике методами. В частности модели сигналов можно представить с помощью рядов Фурье, Тейлора, полиномов Лагранжа, Чебышева и т.п. В результате сложный по форме сигнал можно отображать математически более простыми выражениями. Одна из задач теории передачи информации — оценить частотные свойства передаваемых сигналов и дать рекомендации по их согласованию с частотными характеристиками каналов связи. Для решения таких задач модели сигналов с временным аргументом следует представлять функцией в частотной области аргументов. Очевидно, для передачи сигналов без искажений необходимо согласовывать амплитудно-частотные характеристики аппаратуры передачи данных с частотными характеристиками передаваемых сигналов. Если сигнал содержит частотные составляющие, находящиеся в пределах АЧХ, то они будут передаваться с малыми искажениями. Обеспечить такие условия можно, зная свойства сигналов в

частотной области. Эти знания основаны на математическом аппарате рядов Фурье и преобразования Фурье.

В настоящее время широкое распространение получают системы, в которых осуществляется физическая дискретизация сигнала по времени и по значению, например, при использовании импульсных методов передачи сообщений радиосвязи. Операция дискретизации по времени обратима, если непрерывная функция времени $\alpha(t)$ заменяется последовательностью своих отсчётов $\{a_j\}$ огибающей и фазы (или ортогональных компонент комплексной огибающей), взятых в соответствии с теоремой Котельникова (теорема отсчётов): *если наивысшая частота в спектре сигнала $\alpha(t)$ меньше, чем f_m , то сигнал полностью определяется последовательностью своих значений в моменты, отстоящие друг от друга не больше чем на $\frac{1}{2}f_m$ секунд.*

Дискретизация по времени осуществляется амплитудным импульсным модулятором (линейный безынерционный четырёхполюсник с переменными параметрами). Обратная операция, полностью восстанавливающая функцию должна представлять собой пропускание последовательности отсчётов через фильтр низких частот. Практически такой преобразователь реализован быть не может. Поэтому в реальных условиях можно говорить лишь о приблизительном восстановлении непрерывной функции после дискретизации во времени. Дискретизация по значениям, или квантование непрерывного сигнала состоит в замене, по тем или иным правилам, его значений, принадлежащих непрерывному множеству, дискретными значениями. Чаще всего при квантовании шкала возможных значений сообщения разбивается на равные интервалы (квантование с равномерным шагом Δa) и непрерывное значение заменяется ближайшим дискретным. Однако в общем случае шаг шкалы квантования может быть неравномерным, и непрерывное значение может заменяться не ближайшим дискретным. Во всех случаях каждому дискретному значению соответствует множество непрерывных, поэтому операция квантования является необратимой. Искажения, возникающие при квантовании,

часто характеризуют так называемый шум квантования, понимая под ним разность исходного и квантованного значений. Шум квантования представляет собой пример детерминированных, но не компенсируемых искажений (типа отсечки). Очевидно, чем меньше шаг шкалы квантования, тем меньше шум квантования.

1.1 Особенности импульсной модуляции

Передача непрерывных сообщений по дискретному каналу, называемая цифровым способом передачи, обладает рядом особенностей. При непрерывной модуляции воздействие $x(t)$, являющееся в этом случае модулирующим сигналом или сообщением $a(t)$, взаимодействует с переносчиком $f(t)$. Импульсный переносчик — периодическая последовательность импульсов.

$$f(t) = A_0 \cdot \sum_{i=-\infty}^{\infty} f_1(t - t_0 - i \cdot T_0), \quad (1.1)$$

где $f_1(t)$ — функция, описывающая форму одиночного импульса. Эта последовательность описывается тремя параметрами:

- Амплитудой (высотой) A_0 ;
- Начальной фазой (сдвигом относительно выбранного начала отсчёта) t_0 ;
- Частотой следования $1/T_0$ (или непосредственно периодом следования T_0);
- В общем случае другими параметрами формы импульса $f_1(t)$;

Модуляция состоит в изменении одного или нескольких параметров переносчика в соответствии с воздействием. Для импульсного переносчика возможны следующие виды модуляции:

- Амплитудно-импульсная модуляция (АИМ);

- Широтно-импульсная или модуляция по длительности (ШИМ);
- Дельта-модуляция (ДМ);

Характерной особенностью импульсных систем передачи является то, что энергия сигнала излучается не непрерывно, а в виде коротких импульсов, длительность которых обычно составляет незначительную часть периода их повторения. Благодаря этому энергия импульсного сигнала во много раз меньше энергии непрерывного сигнала (при одинаковых пиковых значениях). Различия в энергиях импульсного и непрерывного сигналов зависит от соотношения между длительностью τ и периодом повторения T_0 , т.е. от соотношения скважности $\gamma = T_0 / \tau$. Обычно $\gamma \gg 1$. Большие временные интервалы между импульсами используются для размещения импульсов других каналов, т.е. для осуществления многоканальной связи с временным разделением каналов. Частоту повторения импульсов определяют, исходя из допустимой точности восстановления непрерывного сообщения при его демодуляции. Минимальная частота повторения импульсов:

$$F_0 = \frac{1}{T_{0\text{ макс}}} = 2 \cdot F_a, \quad (1.2)$$

где F_a — максимальная частота в спектре передаваемого непрерывного сообщения $a(t)$.

В большинстве случаев сигнал импульсной модуляции создаётся в два этапа:

- Сообщение моделирует тот или иной параметр периодической последовательности импульсов;
- Импульсы модулируют непрерывное высокочастотное колебание;

1.2 Анализ спектров импульсной модуляции

При рассмотрении спектров сигналов импульсной модуляции будем исследовать лишь спектры модулированных импульсов.

1.2.1 Спектр сигнала АИМ

При АИМ сигнал записывается в следующем виде:

$$\begin{aligned} s(t) &= [A_0 + \Delta A a(t)] \cdot \sum_{n=-\infty}^{\infty} f_1(t - nT_0 - t_0) = \\ &= A_0 [1 + ma(t)] \cdot \sum_{n=-\infty}^{\infty} f_1(t - nT_0 - t_0) \end{aligned} \quad (1.3)$$

где $f_1(t)$ — функция описывающая одиночный импульс последовательности.

Немодулированную последовательность импульсов, выполняющую роль промежуточного переносчика, можно представить рядом Фурье:

$$s(t) = A_0 \sum_{n=-\infty}^{\infty} f_1(t - nT_0 - t_0) = \sum_{k=-\infty}^{\infty} \dot{C}_k e^{ik\Omega_0 t}, \quad (1.4)$$

где $\Omega_0 = 2\pi / T_0$ — частота повторения импульсов. Спектр сигнала $s(t)$ находится по формуле:

$$\begin{aligned} \dot{S}(\omega) &= \int_{-\infty}^{\infty} s(t) e^{-i\omega t} dt = \int_{-\infty}^{\infty} [1 + ma(t)] \sum_{k=-\infty}^{\infty} \dot{C}_k e^{-i(\omega - k\Omega_0)t} dt = \\ &= 2\pi \sum_{k=-\infty}^{\infty} \dot{C}_k \delta(\omega - k\Omega_0) + m \sum_{k=-\infty}^{\infty} \dot{C}_k \dot{S}_a(\omega - k\Omega_0) \end{aligned} \quad (1.5)$$

Первая сумма, не зависящая от m , представляет собой спектр немодулированных сигналов. АИМ вызывает появление около каждой из составляющих этого спектра боковых полос, повторяющих спектр сообщения $\dot{S}(\omega)$. Таким образом, спектр сигнала АИМ представляет собой как

многократно повторённый спектр обычной амплитудной модуляции, в котором роль несущих частот выполняют гармоники частоты следования импульсов (см. рис. 1.1). Значение $F_0 = 2 \cdot F_a$ определяет то минимальное значение частоты повторения, при котором не происходит наложение спектров соседних боковых полос. Характерной особенностью спектров сигналов импульсной модуляции является наличие около $\omega = 0$ составляющих, соответствующих частотам передаваемого сообщения. Это указывает на возможность демодуляции фильтром нижних частот, пропускающим на выход лишь составляющие с частотами от 0 до $2\pi F_a$ и отфильтровывающим все остальное. Демодуляция будет сопровождаться искажениями, если в полосу пропускания ФНЧ попадут составляющие ближайшей боковой полосы. Из рассмотрения спектра сигнала АИМ видно, что искажения при демодуляции будут отсутствовать, когда спектры соседних боковых полос не перекрываются. А для этого надо чтобы частота повторения импульсов была бы $F_0 \geq 2F_a$. Из этого рассмотрения вытекает также необходимость предварительной фильтрации передаваемого сообщения $a(t)$ таким образом, чтобы ширина спектра его ограничивалась некоторой частотой F_a .

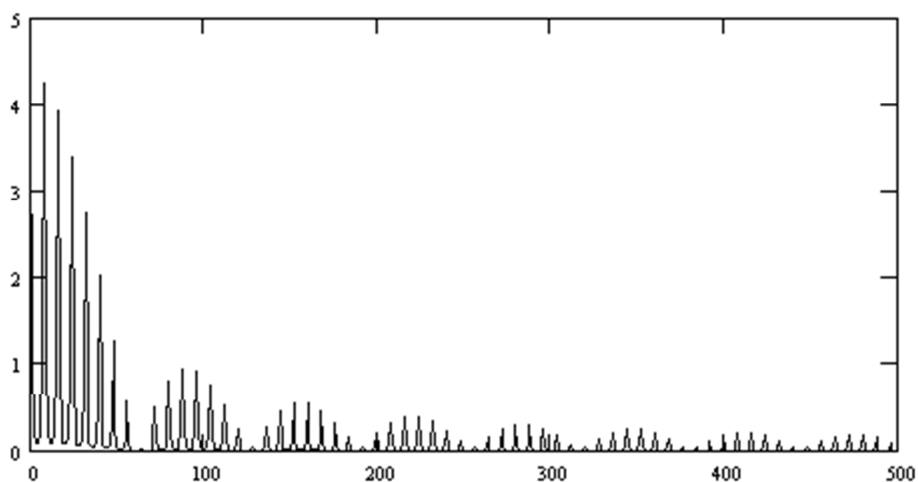


Рисунок 1.1 — Спектр АИМ

1.2.2 Спектр сигнала ШИМ

Рассмотрим сигнал широтно-импульсной модуляции (ШИМ). Для определённости. Не модулированные импульсы имеют прямоугольную форму и длительность τ . За начало отсчёта времени принимается момент, соответствующий середине импульса, так что на интервале равном периоду повторения T_0 , сигнал

$$f_1(t) = \begin{cases} A_0, & t_1 < t < t_2 \\ 0, & t < t_1, t > t_2 \end{cases} \quad (1.6)$$

где t_1 и t_2 — определяют положение переднего и заднего фронтов каждого импульса.

Длительность импульса $\tau = t_1 - t_2$. Пусть при модуляции ширины импульса по закону $a(t) = \sin(\Omega t)$ соответственно изменяется лишь временное положение переднего фронта импульса, а положение заднего фронта остаётся неизменным. Такой вид модуляции называется односторонней модуляцией длительности импульса. Обозначим максимальное отклонение переднего фронта импульса через $\Delta\tau$. Очевидно, что $\Delta\tau \leq \tau$. Тогда положение фронтов импульсов при модуляции выражаются соотношениями:

$$t_1 = -0.5\tau - \Delta\tau \sin(\Omega t); t_2(t) = 0.5\tau \quad (1.7)$$

Представим не модулированную последовательность импульсов прямоугольной формы рядом Фурье. При выбранном начале отсчёта времени

$$f_1(t) = \frac{A_0(t_2 - t_1)}{T_0} + \frac{A_0}{\pi} \sum_{k=1}^{\infty} \frac{1}{k} [\sin k\Omega_0(t_2 - t_1)] - \sin k\Omega_0(t - t_2) \quad (1.8)$$

получаем

$$s(t) = \frac{A_0 \tau}{T_0} + \frac{A_0 \Delta \tau}{T_0} \sin \Omega t + \frac{A_0}{\pi} \sum_{k=1}^{\infty} \frac{1}{k} \left[\sin k \Omega_0 \left(t + \frac{\tau}{2} + \Delta \tau \sin t \Omega \right) \right] - \sin k \Omega_0 \left(t - \frac{\tau}{2} \right) \quad (1.9)$$

Это выражение показывает, что при модуляции коэффициенты ряда Фурье не остаются неизменными, а являются некоторыми периодическими функциями времени. С учётом известного из теории бесселевых функций соотношения

$$\sin(x + y \sin z) = \sum_{m=-\infty}^{\infty} J_m(y) \sin(x + mz) \quad (1.10)$$

выражение для сигнала ШИМ приводится к виду

$$s(t) = A_0 \left\{ \frac{1}{\gamma} + \frac{\Delta \tau}{\gamma \tau} \sin \Omega t + \sum_{k=1}^{\infty} \frac{1}{k \pi} \sin k \Omega_0 \left(t - \frac{\tau}{2} \right) + \sum_{k=1}^{\infty} \sum_{m=-\infty}^{\infty} \frac{1}{k \pi} J_m \left(\frac{2 k \pi \Delta \tau}{\gamma \tau} \right) \sin \left((k \Omega_0 + m \Omega) t + \frac{k \pi}{\gamma} \right) \right\} \quad (1.11)$$

Построенный в соответствии с предыдущей формулой спектр сигнала ШИМ представлен на рис. 1.2. Он имеет более сложную структуру, чем спектр сигнала АИМ при том же законе модуляции. Теперь около гармоник частоты повторения импульсов имеется теоретически бесконечное множество боковых частот. Их амплитуды определяются значениями бесселевой функции m -го порядка от аргумента $\frac{2 k \pi \Delta \tau}{\gamma \tau}$, т.е. зависят от соотношения $\frac{\Delta \tau}{\tau}$, которое можно назвать коэффициентом глубины модуляции длительности импульсов. Этот же коэффициент определяет и значения амплитуды составляющей спектра, имеющей частоту сообщения Ω . Предельное значение $\frac{\Delta \tau}{\tau} = 1$. При $\gamma \gg 1$ для составляющих с небольшим значением k (которые и следует учитывать, т.к.

амплитуды спектральных составляющих обратно пропорциональны k)

получаем $\frac{2k\pi}{\gamma} \ll 1$. При малых значениях аргумента

$$J_1(y) \approx 0.5y; J_2(y), J_3(y), \dots \ll J_1(y) \quad (1.12)$$

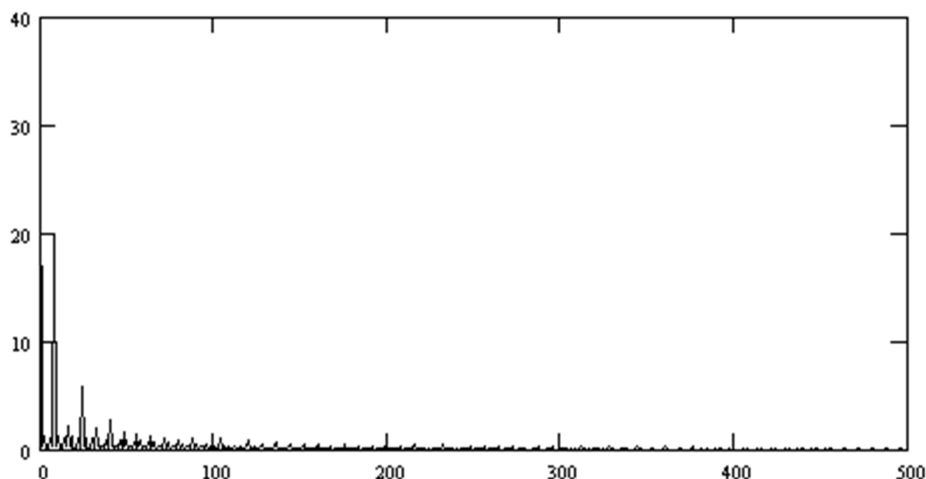


Рисунок 1.2 — Спектр ШИМ

Следовательно, при сделанных предположениях спектр сигнала ШИМ на частотах вблизи $\omega = 0$ совпадает по структуре со спектром сигнала АИМ. Это означает, что возможна демодуляция с помощью фильтра нижних частот, причём для устранения искажений в выходном сообщении должно выполняться условие $F_0 \geq 2F_a$. Если параметры модуляции таковы, что аргумент бесселевой функции при малых значениях k имеет порядок единиц, то составляющие спектра с частотами $k\Omega_0 - m\Omega$ имеют заметные амплитуды, и для устранения искажений при демодуляции необходимо увеличить частоту повторения импульсов по сравнению с $2F_a$.

1.3 Основы теории разделения сигналов

1.3.1 Блок-схема многоканальной системы

Задача увеличения числа одновременно действующих каналов решается с помощью многоканальных систем связи, предназначенные обеспечить одновременную и независимую передачу сообщений от многих отправителей к

такому же числу получателей. В таких системах общая линия связи «уплотняется» индивидуальными каналами, каждый из которых обеспечивает передачу сообщений единственной пары абонентов (отправителя и получателя).

На рис. 1.3 приведена блок-схема системы многоканальной связи. Сообщение каждого канала $a_1(t), a_2(t), \dots, a_k(t), \dots, a_n(t)$ с помощью индивидуальных передатчиков (модуляторов) $M_1, M_2, \dots, M_k, \dots, M_n$ преобразуются в соответствующие каналные сигналы $s_1(t), s_2(t), \dots, s_k(t), \dots, s_n(t)$. Совокупность каналных сигналов на выходе суммирующего устройства Σ образует групповой сигнал $s(t)$. Наконец, в групповом передатчике M сигнал $s(t)$ преобразуется в сигнал $s_n(t)$, который и поступает в линию связи.

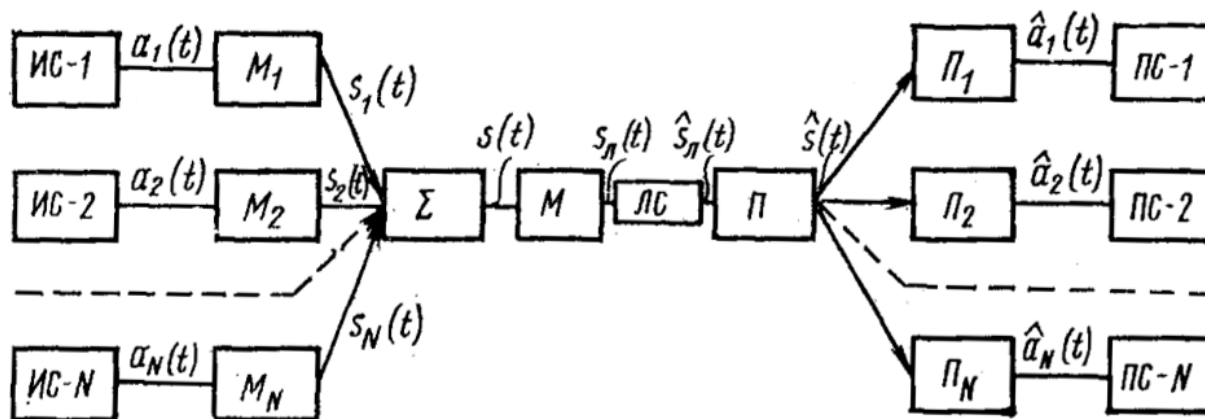


Рисунок 1.3 — Блок-схема системы многоканальной связи

Предположим, что линия пропускает сигнал без искажений и не имеет шумов. Тогда на приёмном конце линии связи сигнал $s_n(t)$ с помощью группового приёмника Π , может быть вновь преобразован в групповой сигнал $s(t)$. Канальными или индивидуальными приёмниками $\Pi_1, \Pi_2, \dots, \Pi_k, \dots, \Pi_n$ из группового сигнала $s(t)$ выделяются соответствующие каналные сигналы $s_1(t), s_2(t), \dots, s_k(t), \dots, s_n(t)$ и затем преобразуются в предназначенные получателям сообщения $a_1(t), a_2(t), \dots, a_k(t), \dots, a_n(t)$.

Канальные передатчики вместе с суммирующим устройством Σ образуют аппаратуру уплотнения. Групповой передатчик M , линия связи и

групповой приёмник Π составляют групповой сигнал. Групповой канал связи вместе с аппаратурой уплотнения и индивидуальными приёмниками объединяются в систему многоканальной связи.

Особо следует обратить внимание на то обстоятельство, что в отличие от одноканальной системы связи индивидуальные приёмники системы многоканальной связи Π_k наряду с выполнением обычной операции преобразования сигнала $S_k(t)$ в соответствующие сообщения $a_k(t)$ должны обеспечивать выделение сигнала $S_k(t)$ из группового сигнала $s(t)$. Иначе говоря, в составе технических устройств на приёмном конце системы многоканальной связи должна быть предусмотрена специальная аппаратура разделения каналов, предназначенная для выделения канальных сигналов $S_k(t)$ из группового сигнала $s(t)$.

1.3.2 Временной способ разделения сигналов

Принцип временного разделения весьма прост и давно применяется в телеграфии. Он состоит в том, что с помощью электромеханических или электронных коммутаторов $K_{пер}$ групповой канал связи предоставляется поочерёдно для передачи сигналов каждого канала многоканальной системы. Сначала передаётся сигнал 1-го канала, затем следующего канала и т.д. до последнего канала за номером N , после чего опять включается 1-ый канал, и процесс периодически повторяется (см. рис. 1.4).

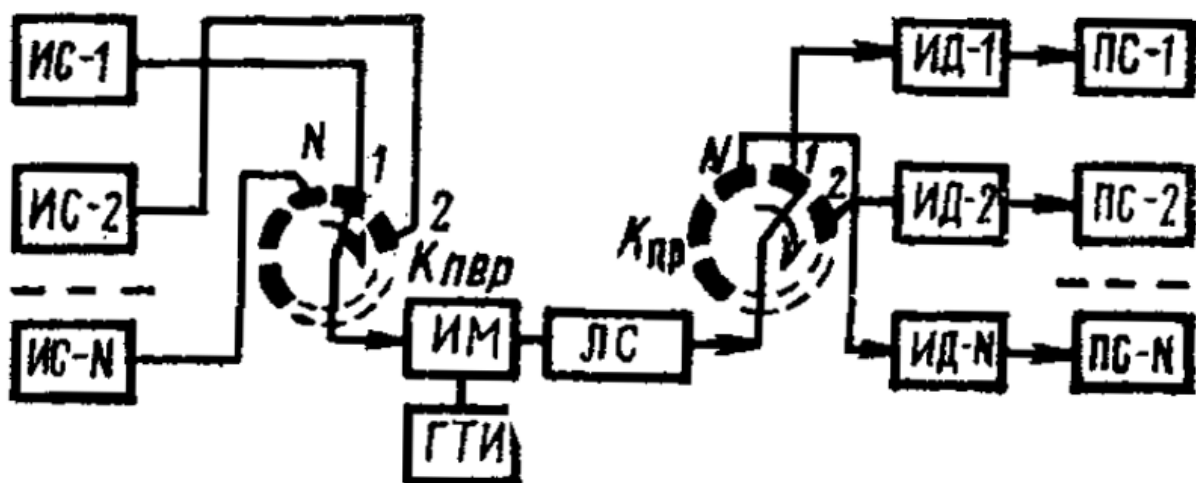


Рисунок 1.4 — Блок-схема системы многоканальной связи с временным разделением сигналов

На приёмном конце устанавливается аналоговый переключатель $K_{пр}$, который подключает групповой канал поочерёдно к приёмникам разных каналов. Приёмник каждого k -го канала должен быть включён только на время передачи k -го сигнала и выключен все остальное время, пока передаются сигналы других каналов. Это означает, что для нормальной работы многоканальной системы с временным разделением необходима синхронная и синфазная работа коммутаторов на приёмной и передающей стороне. Часто для этого один канал занимается под передачу специальных импульсов синхронизации, предназначенных для согласованной работы $K_{пер}$ и $K_{пр}$.

Таким образом, сущность временного способа организации многоканальной связи в том, что сигналы различных каналов передаются поочерёдно и в той же последовательности подключаются к приёмникам. Переносчиком сообщений здесь является последовательность импульсов (с периодом T_0), поступающая на импульсный модулятор $ИМ$ от генератора тактовых импульсов $ГТИ$. Пусть для простоты передача ведётся с помощью сигналов АИМ. Групповой сигнал $s(t)$ поступает на коммутатор $K_{пр}$. Последний выполняет роль «временных» фильтров или ключей, проводимость

которых g_k изменяется синхронно (с периодом T_0) и синфазно с изменениями проводимости $K_{пер}$:

$$g_k = g_k(t - k\tau) = \begin{cases} 1, & t \in \Delta t_k \\ 0, & t \notin \Delta t_k \end{cases}$$

Это означает, что в пределах каждого временного интервала Δt_k линия связи соединяется только с k -м импульсным детектором ИД- k . Полученные в результате детектирования $S_k(t)$ сообщения поступают к получателю сообщений ПС- k .

При временном разделении взаимные помехи в основном обусловлены двумя причинами. Первая из них состоит в том, что за счёт ограниченности полосы частот всякой физически осуществимой системы связи нарушается импульсный характер сигналов. Действительно, передаче модулированных импульсов конечной длительности ограничить спектр, то импульсы «расплывутся» и вместо импульсов конечной длительности мы получим процессы, бесконечно протяжённые во времени. При временном разделении сигналов это приведёт к тому, что импульсы одного канала будут накладываться на импульсы других каналов. Иначе говоря, между каналами возникают взаимные переходные помехи (взаимная интерференция символов).

Кроме того, взаимные помехи могут возникать из-за несовершенства синхронизации.

Для снижения уровня взаимных помех приходится вводить «защитные» временные интервалы, что соответствует некоторому расширению спектр сигнала. Так, в многоканальной телефонии полоса эффективно передаваемых частот устанавливается $F = 3100 \text{ Гц}$ в соответствии с теоремой Котельникова, минимальное значение периода следования тактовых импульсов можно было

взять $T_0 = \frac{1}{2F} = \frac{1}{6200} \approx 160 \text{ мксек}$. Однако, в реальных системах период следования импульсов выбирается с некоторым запасом, его принимают

равным $T_0' = 125 \text{ мксек}$, что соответствует частоте $F = 4 \text{ кГц}$. Это естественно приводит к снижению эффективности использования спектра. При временном разделении сигнал каждого канала занимает одинаковую полосу частот, определяемую из соотношения (без учёта канала синхронизации):

$$\Delta t_k = \frac{T_0}{N} = \frac{1}{N2F} = \frac{1}{2F_{\text{общ}}},$$

где $F_{\text{общ}} = NF$

Системы с временным разделением имеют неоспоримое преимущество, связанное с тем, что благодаря поочерёдной передаче сигналов разных каналов отсутствуют взаимные помехи, обусловленные нелинейностью тракта передачи.

1.4 Реализация в среде MathCAD

1.4.1 Реализация АИМ в среде MathCAD

Передача непрерывных сообщений по дискретному каналу, называемая цифровым способом передачи, обладает рядом особенностей. Как указывалось выше для передачи по дискретному каналу непрерывное сообщение должно быть преобразовано в дискретную форму. Дискретизация производится следующей программой:

$$\begin{array}{l} \underline{S} := \left| \begin{array}{l} k \leftarrow 0 \\ \text{for } i \in 0 \dots \frac{T_c}{T} - 1 \\ \quad \text{for } j \in \frac{i \cdot T \cdot 2 - t_n}{2 \cdot \Delta} \dots \frac{(i+1) \cdot T \cdot 2 - t_n}{2 \cdot \Delta} - 1 \\ \quad \quad \left| \begin{array}{l} R_k \leftarrow \text{sp}(i \cdot T) \cdot \left(\Phi \left(j - \frac{i \cdot T \cdot 2 - t_n}{2 \cdot \Delta} \right) - \Phi \left(j - \frac{i \cdot T \cdot 2 + t_n}{2 \cdot \Delta} \right) \right) \\ k \leftarrow k + 1 \end{array} \right. \\ \end{array} \right. \\ R \end{array}$$

где: S — полученный массив дискретных значений;

T_c — время процесса дискретизации;

T — период дискретизации;

tn — длительность импульса;

Δ — элементарный квант времени;

$sp(t)$ — входной сигнал.

Весь интервал времени Tc разбит на элементарные шаги Δ малые по сравнению с длительностью импульса tn . Программа построена на двух циклах:

- Цикл по периоду квантования T ;
- Вложенный цикл по импульсу tn который лежит в интервале $\left[T_i - \frac{tn}{2}, T_{i+1} - \frac{tn}{2} \right]$

На данном интервале выполняется следующее условие:

$$R_k = \begin{cases} s(t), T_i - \frac{tn}{2} \leq t < T_i + \frac{tn}{2} \\ 0, T_i + \frac{tn}{2} \leq t < T_{i+1} + \frac{tn}{2} \end{cases}$$

реализовано это следующим образом:

$$R_k \leftarrow sp(i \cdot T) \cdot \left(\Phi \left(j - \frac{i \cdot T \cdot 2 - tn}{2 \cdot \Delta} \right) - \Phi \left(j - \frac{i \cdot T \cdot 2 + tn}{2 \cdot \Delta} \right) \right)$$

В итоге работы программы формируется массив S значений входной функции в определённые моменты времени. Разложение в ряд Фурье производится с помощью функции MathCAD:

$$Sf := cfft(S)$$

1.4.2 Реализация ШИМ в среде MathCAD

Построение спектра ШИМ отличается от построения спектра АИМ тем, что значение tn зависит от входной функции, а амплитуда A на всем промежутке

дискретизации остаётся постоянной. Получаем следующую программу дискретизации сигнала:

```

.S :=
  k ← 0
  tn ← 0
  for i ∈ 0..  $\frac{T_c}{T} - 1$ 
    tn ←  $\frac{sp(i \cdot T)}{MAX} \cdot T$ 
    for j ∈  $\frac{i \cdot T \cdot 2 - tn}{2 \cdot \Delta} .. \frac{(i+1) \cdot T \cdot 2 - tn}{2 \cdot \Delta} - 1$ 
       $R_k \leftarrow A \cdot \left( \Phi \left( j - \frac{i \cdot T \cdot 2 - tn}{2 \cdot \Delta} \right) - \Phi \left( j - \frac{i \cdot T \cdot 2 + tn}{2 \cdot \Delta} \right) \right)$ 
      k ← k + 1
  R

```

Значение tn находится следующим образом:

$$tn \leftarrow \frac{sp(i \cdot T)}{MAX} \cdot T$$

где MAX — максимальное значение, которое может принимать сигнал.

1.4.3 Пример реализации АИМ и ШИМ

Рассмотрим следующий пример:

Время процесса дискретизации $T_c=4$;

Элементарный квант времени $\Delta=0.001$;

Период дискретизации $T=0.5$;

Входной сигнал $s(t) = \sin(t^{\cos(t)})$;

Длительность импульса $tn=1/16$.

При амплитудно-импульсной модуляции получаем (рис. 1.5 и рис. 1. 6):

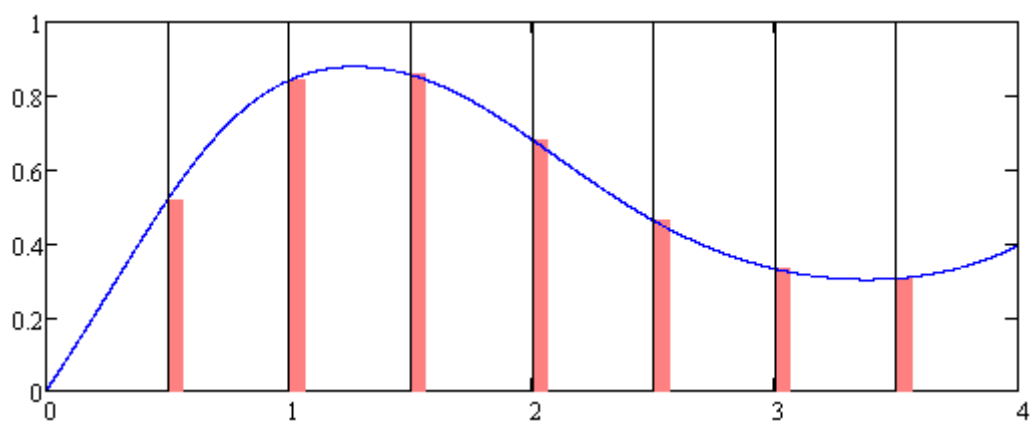


Рисунок 1.5 — График входного сигнала и АИМ сигнала

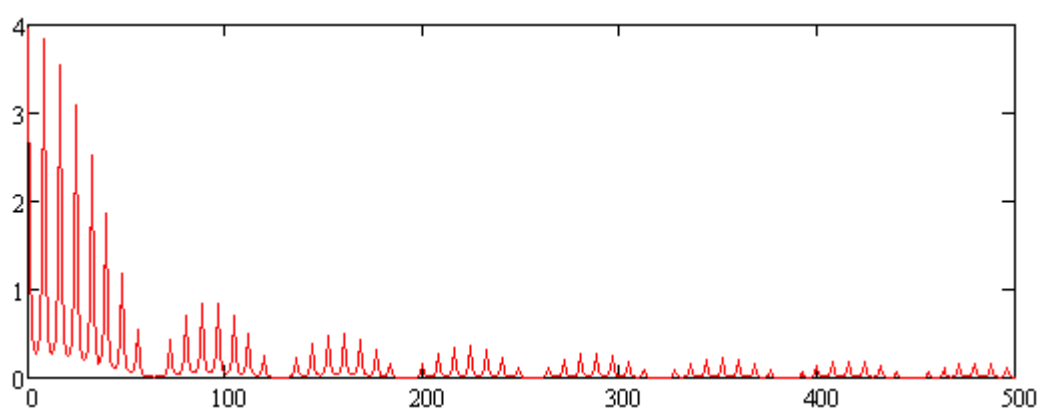


Рисунок 1.6 — Спектр АИМ сигнала

При широтно-импульсной модуляции получаем (рис. 1.7 и рис. 1.8):

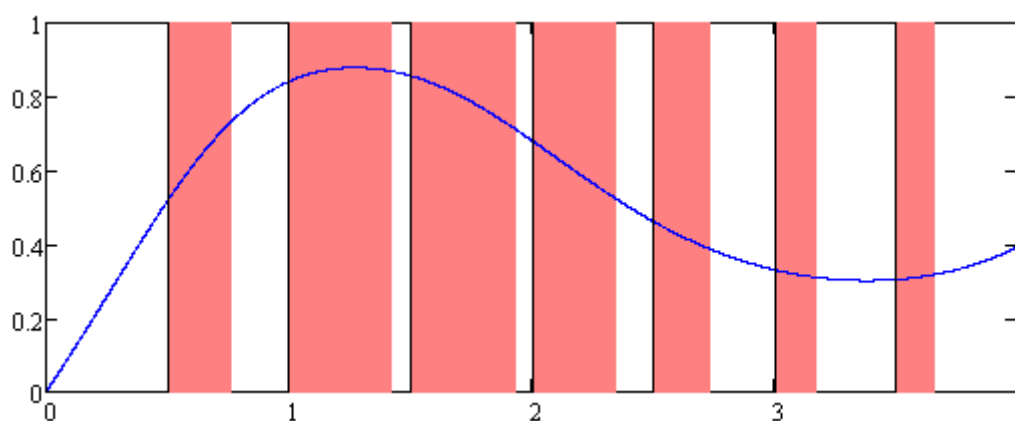


Рисунок 1.7 — График входного сигнала и ШИМ сигнала

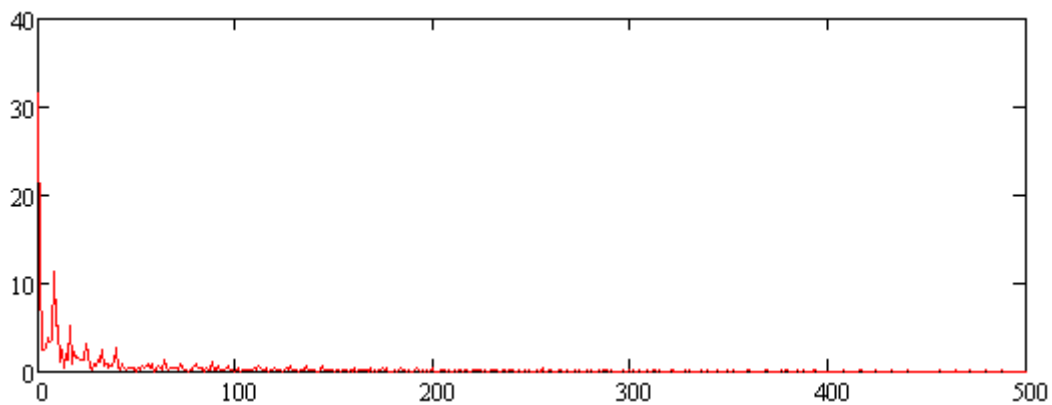


Рисунок 1.8 — Спектр ШИМ сигнала

1.5 Контрольные вопросы

- Определение теоремы Котельникова;
- Импульсный переносчик;
- Суть импульсной модуляции сигналов;
- Виды импульсной модуляции и характерные особенности импульсных систем;
- Спектры сигналов импульсной модуляции;
- Характерные особенности спектров сигналов импульсной модуляции;
- Блок-схема многоканальной системы связи;
- Принцип временного разделения сигналов многоканальной системы связи;
- Блок-схема системы многоканальной связи с временным разделением сигналов;
- Методы устранения рассинхронизации при временном разделении сигналов;
- Преимущество системы с временным разделением сигналов;
- Реализация АИМ в среде MathCAD;
- Реализация ШИМ в среде MathCAD.

1.6 Задание

1. Варианты исходных данных приведены в приложении А;

2. На примере АИМ и ШИМ реализации, реализовать программу в среде MathCAD и построить спектр сигнала;
3. Объяснить принцип работы программы дискретизации;
4. Оформить отчет.

1.7 Литература

1. Назаров М.В., Кувшинов Б.И., Ионов О.В. Теория передачи сигналов. Москва, «Связь» 1970г.
2. Зюко А.Г., Кловский Д.Д., Назаров М.В., Финк Л.М. Теория передачи сигналов. Москва, «Связь» 1980г.
3. Гоноровский И.С. Радиотехнические цепи и сигналы

Лабораторная работа №2. Частотно-фазовая модуляция и демодуляция

Модуляция — преобразование первичного сообщения (сигнала) в сигнал, пригодный для передачи по линии связи.

Демодуляция — это не только операция, обратная модуляции, выполняемые над пришедшем из канала сигналом. В результате различных искажений и воздействия помех, пришедший сигнал может существенно отличаться от переданного сигнала. Поэтому демодуляция должна включать анализ приходящего сигнала, и принимать решение о переданном сообщении.

Дискретную модуляцию часто называют *манипуляцией*, и устройство его осуществляющую — *манипулятор* или *генератор сигналов*.

2.1 Модуляция сигнала

2.1.1 Фазовая модуляция

При частотной модуляции (ЧМ) общее выражение для высокочастотного колебания, амплитуда которого постоянна, а аргумент $\psi(t)$ модулирован по определённому закону (частота колебания также остаётся неизменной), можно записать в виде:

$$y(t) = A_0 \cdot \sin(\omega_0 t + a(t) + \theta_0), \quad (2.1)$$

где A_0 — амплитуда модулированного колебания;
 ω_0 — несущая частота колебания;
 $a(t)$ — модулирующий сигнал;
 θ_0 — начальная фаза колебания.

Поясним это соотношение на примере простейшей гармонической частотной модуляции, когда модулирующий сигнал представлен в виде:

$$a(t) = \theta_D \cdot \cos(\Omega t), \quad (2.2)$$

где θ_D — амплитуда отклонения периода.

Полученная фазовая модуляция представлена на рис. 2.1.

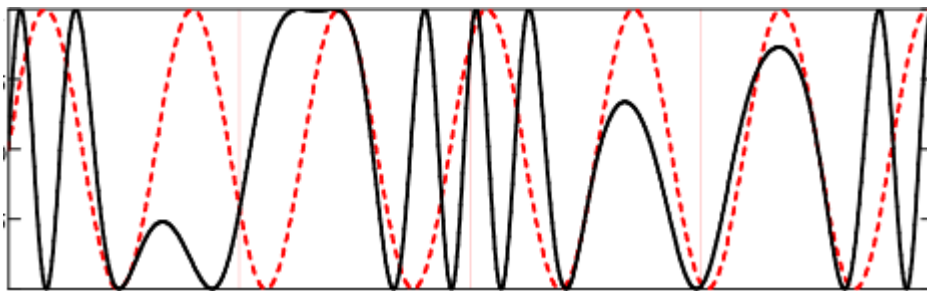


Рисунок 2.1 — Фазовая модуляция сигнала

2.1.2 Фазовая манипуляция

Фазовая манипуляция применяется намного чаще чем фазовая модуляция, и проще для понимания. Простая фазовая манипуляция показана в примерах дальше. Исходный набор данных представляет собой последовательность из $DataLeng=8$ бит, которые записаны в массив $Data$. При фазовой манипуляции используется следующая формула:

$$s_k := A \cdot \sin \left(f_{sin} \cdot \pi \cdot k \cdot C + Data_{\text{floor} \left(\frac{k}{N} \right)} \cdot \frac{\pi}{2} \right), \quad (2.3)$$

где $Data$ — исходный массив данных

$A = 1$ — амплитуда модулированного колебания;

f_{sin} — несущая частота колебаний;

$C = \frac{T_{сигн}}{K}$ — интервал дискретизации;

$N = 2^7$ — количество точек несущей, на которые при передаче накладывается один бит данных.

Полученная фазовая манипуляция приведена на рис. 2.2. Спектр фазовой манипуляции приведён на рис. 2.3.

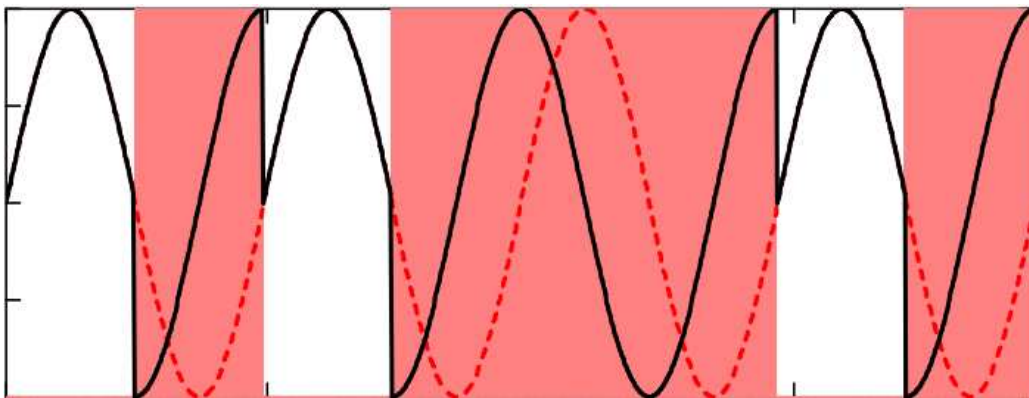


Рисунок 2.2 — Фазовая манипуляция

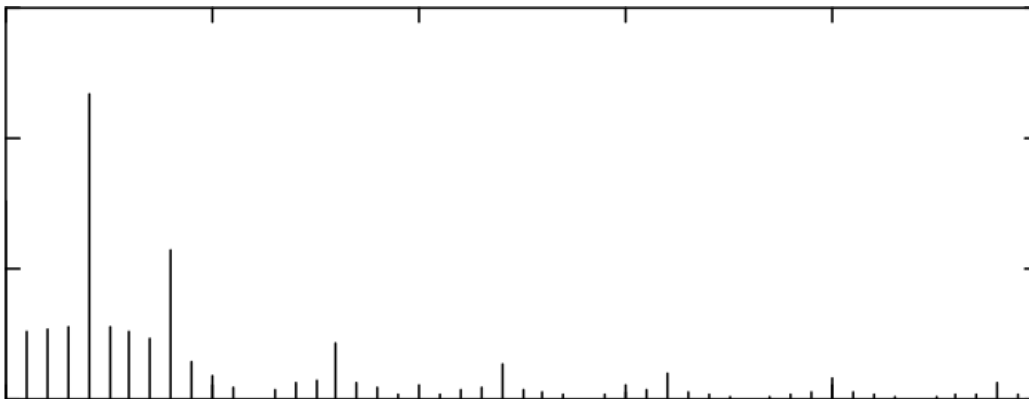


Рисунок 2.3 — Спектр фазовой манипуляции

2.1.3 Частотная модуляция

При частотной модуляции (ЧМ) общее выражение для высокочастотного колебания, амплитуда которого постоянна, а аргумент $\psi(t)$ модулирован по

определённому закону (фаза колебания также остаётся неизменной), можно записать в виде:

$$y(t) = A_0 \cdot \sin\left(\omega_0 t + \int_0^t a(\tau) d\tau + \theta_0\right), \quad (2.4)$$

Поясним это соотношение на примере простейшей гармонической частотной модуляции, когда модулирующий сигнал представлен в виде:

$$a(t) = \omega_D \cdot \cos(\Omega t), \quad (2.5)$$

где ω_D — амплитуда частотного отклонения (*частота девиации*);

Ω — модулирующая частота.

Выполнив интегрирование, получим:

$$y(t) = A_0 \cdot \sin(\omega_0 t + m \sin(\Omega t) + \theta_0), \quad (2.6)$$

Где $m = \frac{\omega_D}{\Omega}$ — *индекс модуляции*.

Установлено, что при *быстрой модуляции* ($m \ll 1$) ширина спектра частотно-модулированного колебания близка к величине 2Ω , при *медленной модуляции* ($m \gg 1$) ширина спектра близка к величине $2\omega_D$.

При модуляции частоты колебания по закону, отличающемуся от гармонического, нахождение спектра сигнала усложняется и зависит от вида модулирующей функции.

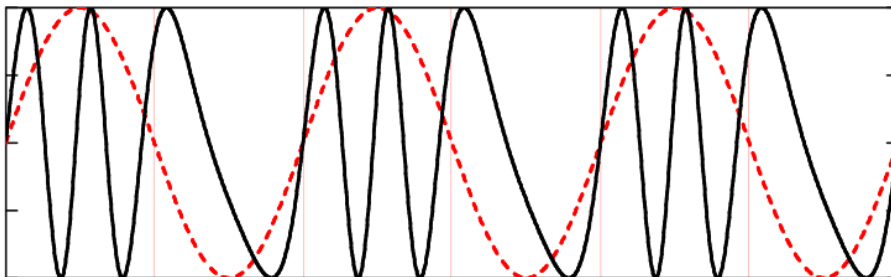


Рисунок 2.4 — Частотная модуляция

2.1.4 Частотная манипуляция

Набор начальных данных тот же что и в предыдущем примере. При моделировании частотной манипуляции используется следующая функция:

$$s := \left| \begin{array}{l} \text{int} \leftarrow 0 \\ \text{for } i \in 0..K-1 \\ \quad \left| \begin{array}{l} \text{int} \leftarrow \text{int} + \pi \cdot C \cdot Wd \cdot \text{Data} \left\lfloor \frac{i}{N} \right\rfloor \\ R_i \leftarrow A \cdot \sin(fsin \cdot \pi \cdot i \cdot C + \text{int}) \end{array} \right. \\ R \end{array} \right. , \quad (2.7)$$

где $Data$ — исходный массив данных

$A = 1$ — амплитуда модулированного колебания;

$fsin$ — несущая частота колебаний;

Wd — амплитуда частотного отклонения (частота девиации);

$K = N \cdot DataLng$ — количество точек в выходном сигнале

$C = \frac{T_{сигн}}{K}$ — интервал дискретизации;

$T_{сигн} = \frac{DataLng}{fsin}$ — длительность манипулированного сигнала;

$N = 2^7$ — количество точек несущей, на которые при передаче накладывается один бит данных.

Манипулированный сигнал представлен на рис. 2.5. На рис. 2.6 изображён спектр этого сигнала.

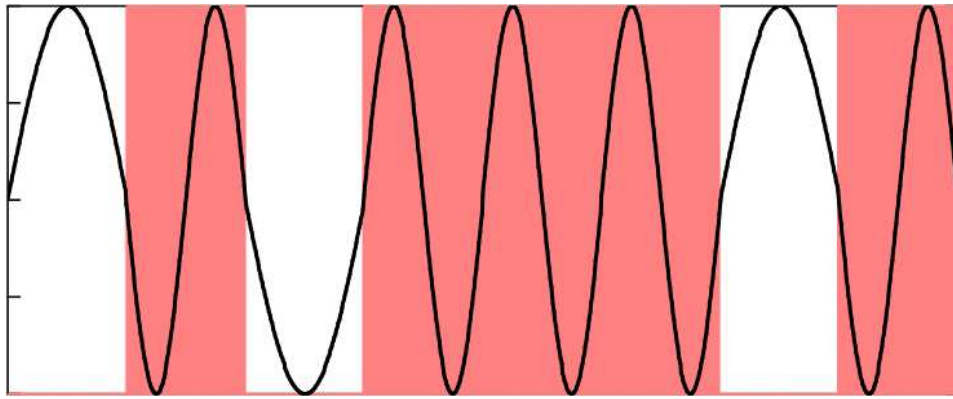


Рисунок 2.5 — Частотная манипуляция

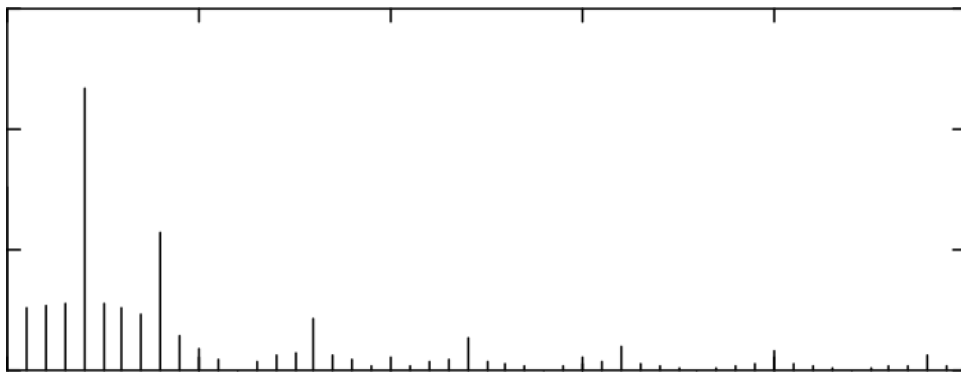


Рисунок 2.6 — Спектр частотной манипуляции

2.2 Демодуляция сигнала

2.2.1 Краткие сведения из теории оптимального приёма сигналов

Пусть на вход приёмника поступает сигнал $x(t)$, равный сумме полезного сигнала $u(t)$ и помехи $n(t)$. Тогда оптимальный приёмник вычисляет корреляционный интеграл, а затем сравнивает его величину с порогом z_0 . Если помеха представляет собой белый шум, то корреляционный интеграл имеет вид:

$$z = \int_0^T x(t)u(t)dt. \quad (2.8)$$

Пределы интегрирования в этом случае учитывают отрезок времени, в котором может появиться обнаруживаемый сигнал $u(t)$.

Значение корреляционного интервала находят с помощью *коррелятора* или *согласованного фильтра*. Основными элементами коррелятора являются

перемножитель, генератор ожидаемого сигнала $u(t)$ и интегратор. На перемножитель поступают входной сигнал $x(t)$ и сигнал $u(t)$, произведение $x(t)$ и $u(t)$ интегрируется с момента прихода ($t=0$) и до момента окончания обнаруживаемого сигнала ($t=T$). Коррелятор является устройством с переменными параметрами, так как режим его работы зависит от изменения $u(t)$ во времени.

Найдём пассивный фильтр, который в момент окончания сигнала создаёт напряжения, пропорциональное z . Напряжение на выходе фильтра в момент t можно найти с помощью интеграла Дюамеля:

$$v(t) = \int_{-\infty}^t x(\tau)h(t-\tau)d\tau, \quad (2.9)$$

где $h(t)$ — импульсная характеристика фильтра.

В момент окончания сигнала $v(t)$ с точностью до постоянного множителя a равно z , т.е.

$$v(T) = a \cdot z \quad (2.10)$$

Отсюда получаем окончательный результат:

$$v(T) = \int_0^T x(t)h(T-t)dt. \quad (2.11)$$

Нижний предел в интеграле положим равным 0, сигнал начинается при $t=0$. Импульсная характеристика этого фильтра имеет вид:

$$h(t) = a \cdot u(T-t),$$

т.е. представляет собой зеркально отображённый сигнал с запаздыванием на T . Фильтр с такой характеристикой называется *согласованным*.

Общим между коррелятором и согласованным фильтром является равенство с точностью до постоянной выходных напряжений в момент времени

$t = T$. Их различие заключается в следующем. Коррелятор — это устройство с переменными во времени параметрами, а согласованный фильтр — устройство с постоянными параметрами. Следствием этого является то, что согласованный фильтр инвариантен относительно задержки сигнала и его начальной фазы (насколько эти величины изменяются в сигнале на входе фильтра, настолько они изменяются и в сигнале на выходе), а коррелятор не инвариантен.

2.2.2 Построение корреляционного приёмника

На вход приёмника поступает фазово-манипулированный сигнал, моделирование осуществляется формулой:

$$b_i = \sum_{k=i \cdot N}^{(i+1) \cdot N} S_k \cdot L_k \cdot C, \quad (2.12)$$

где i — номер принятого бита;

S — массив, содержащий дискреты манипулированного сигнала;

L — массив, содержащий дискреты ожидаемого сигнала;

C — интервал дискретизации;

N — количество точек, на которые при передаче накладывается один бит данных.

Реализация этой формулы представлена в блок-программах `pd`. Здесь `L1` — массив дискретов сигнала, соответствующего передаче «0», `L2` — массив дискретов сигнала, соответствующего передаче «1». Результатами этих программ являются вектора `sb1` и `sb2`, в которых записаны последовательности битов («1» соответствует ожидаемой частоте, «0» — в обратном случае). Таким образом, в векторе `sb2`, будет содержаться исходная последовательность битов, записанная в векторе `Data`.

$\text{Data} := (0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1)^T$	- передаваемая последовательность битов
$\text{DataLng} := \text{length}(\text{Data})$ $\text{DataLng} = 8$	- длина передаваемой последовательности
$N := 2^7$	- количество точек несущей, на которые накладывается 1 бит данных
$f_{\sin} := 10^3$	- частота несущего сигнала
$W_d := 10^3$	- частота девиации
$T_d := \frac{\pi}{2}$	
$K := N \cdot \text{DataLng}$ $K = 1.024 \times 10^3$	- количество точек, в которых вычисляется сигнал
$k := 0..K - 1$	
$T_{\text{sign}} := \frac{\text{DataLng}}{K}$ $T_{\text{sign}} = 7.813 \times 10^{-3}$	- длительность сигнала
$C := \frac{T_{\text{sign}}}{f_{\sin}}$ $C = 7.813 \times 10^{-6}$	- интервал дискретизации
$A := 1$	
$P_d := 0.3$	- порог срабатывания фазового демодулятора
$F_d := 0.35$	- порог срабатывания частотного демодулятора
$o_k := A \cdot \sin(f_{\sin} \cdot \pi \cdot k \cdot C)$	
$s_k := A \cdot \sin\left(f_{\sin} \cdot \pi \cdot k \cdot C + \text{Data}_{\text{floor}\left(\frac{k}{N}\right)} \cdot \frac{\pi}{2}\right)$	$t := 0, C.. T_{\text{sign}} - C$

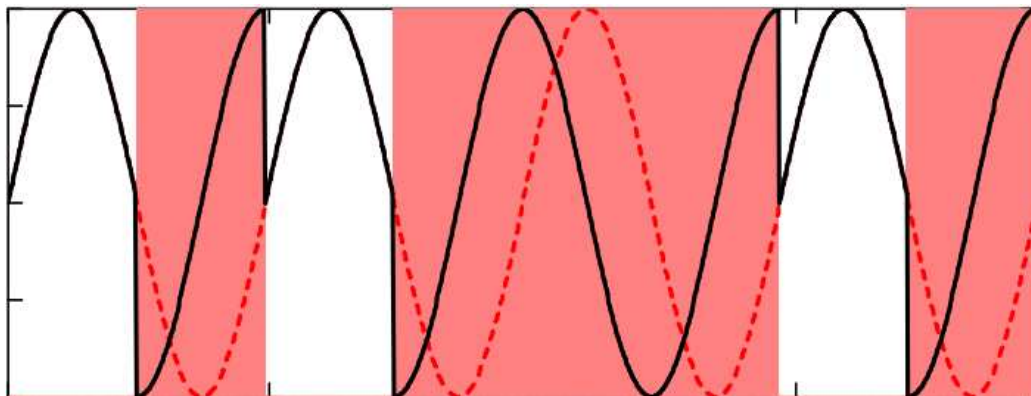


Рисунок 2.7 — Исходный манипулированный сигнал

Реализация корреляционного приёмника:

$$L1(k) := A \cdot \sin(f_{\sin} \cdot \pi \cdot k \cdot C) \quad L2(k) := A \cdot \cos(f_{\sin} \cdot \pi \cdot k \cdot C)$$

$$pd(s, L) := \begin{array}{|l} \text{for } i \in 0.. \frac{K}{N} - 1 \\ \quad t \leftarrow 0 \\ \quad \text{for } k \in N \cdot i.. N \cdot (i + 1) - 1 \\ \quad \quad t \leftarrow t + 1 \text{ if } |s_k - L(k)| > Pd \\ \quad b_{0,i} \leftarrow t \\ \quad b_{1,i} \leftarrow \text{if}(t > 25, 1, 0) \end{array} \quad \begin{array}{l} sb1 := pd(s, L1) = \begin{pmatrix} 0 & 111 & 0 & 111 & 111 & 111 & 0 & 111 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} \\ sb2 := pd(s, L2) = \begin{pmatrix} 111 & 0 & 111 & 0 & 0 & 0 & 111 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Выходной сигнал изображён на рис. 2.8 и рис. 2.9.

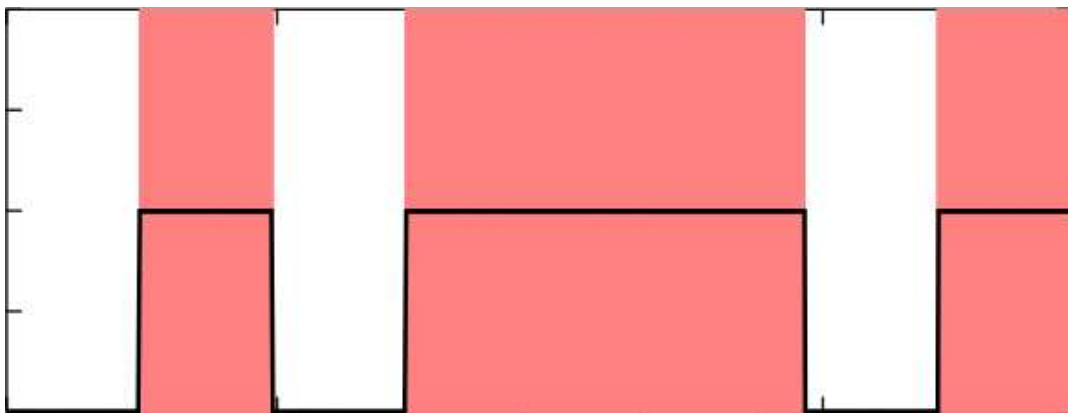


Рисунок 2.8 — Демодулированный сигнал

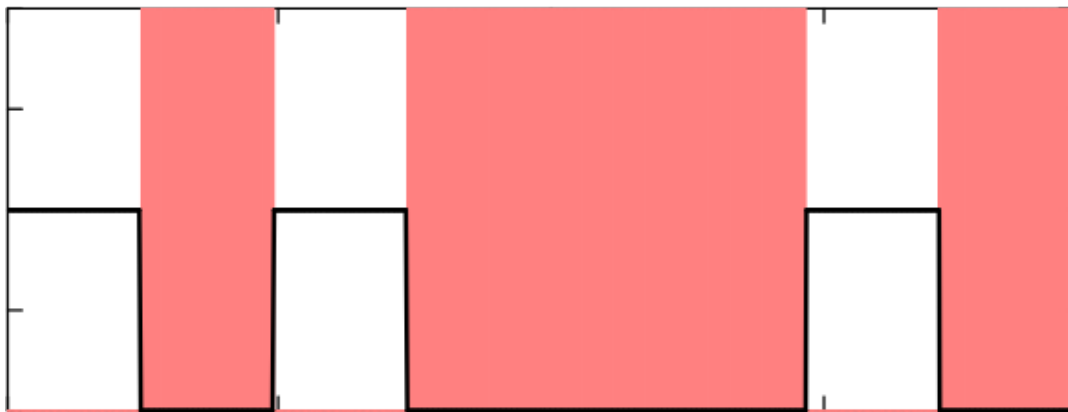


Рисунок 2.9 — Демодулированный сигнал

Проверка сигнала на помехоустойчивость:

$$ss_i := s_i + \text{rnd}(0.5) - \text{rnd}(0.5)$$

$$sb1 := \text{pd}(ss, L1) = \begin{pmatrix} 24 & 106 & 23 & 107 & 111 & 112 & 23 & 114 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} \quad sb2 := \text{pd}(ss, L2) = \begin{pmatrix} 111 & 24 & 109 & 22 & 21 & 22 & 108 & 20 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Исходный сигнал с помехами показан на рис. 2.10, демодулированный сигнал на рис. 2.8 и рис. 2.9 (совпадают с принятыми без помех). Как можно заметить сигнал был принят без ошибок, но в некоторых битах был близок к этому.

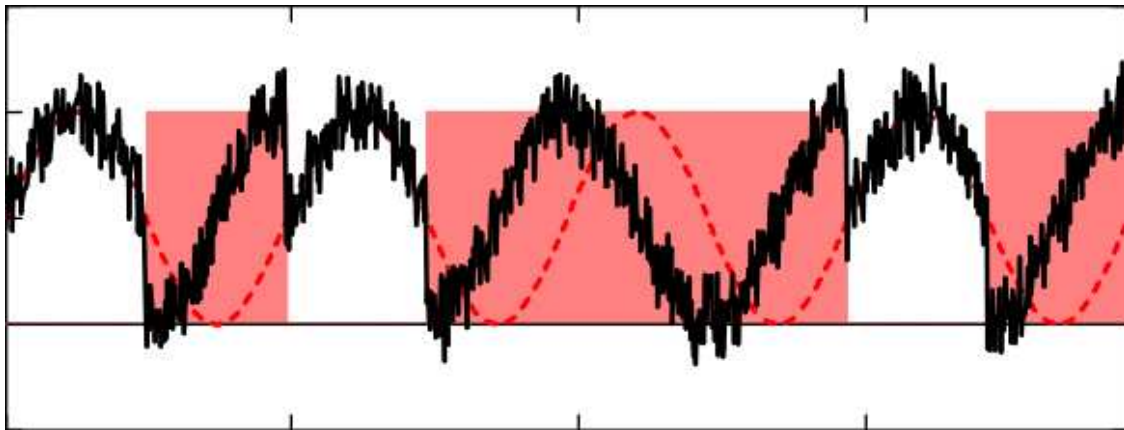


Рисунок 2.10 — Манипулированный исходный сигнал с помехами

2.2.3 Цифровые частотные демодуляторы

Демодуляторы ЧМ сигналов также могут быть реализованы как на цифровых, так и на аналоговых устройствах.

В цифровых частотных демодуляторах реализуется принцип классификации принимаемых сигналов по частоте на основе измерения длительности полупериода (или периода) принимаемого сигнала. На основании измерения длительности полупериода при двоичной модуляции решающее устройство отождествляет принятый полупериод с одним из значений полярности сигнала. Таким образом реальный ЧМ сигнал разбивается на элементарные отрезки сигнала, содержащие полупериод несущего колебания. Определение границ единичных элементов осуществляется с точностью, не превышающей длительность одного элементарного отрезка сигнала. Разновидностью метода измерения длительности полупериода (периода) принимаемого сигнала является метод измерения разности набега фазы каждого

текущего колебания относительно предшествующего периода. Структурная схема цифрового частотного демодулятора приведена на рис. 2.11. Временные диаграммы, поясняющие принцип действия цифрового частотного демодулятора, приведены на рис. 2.12.

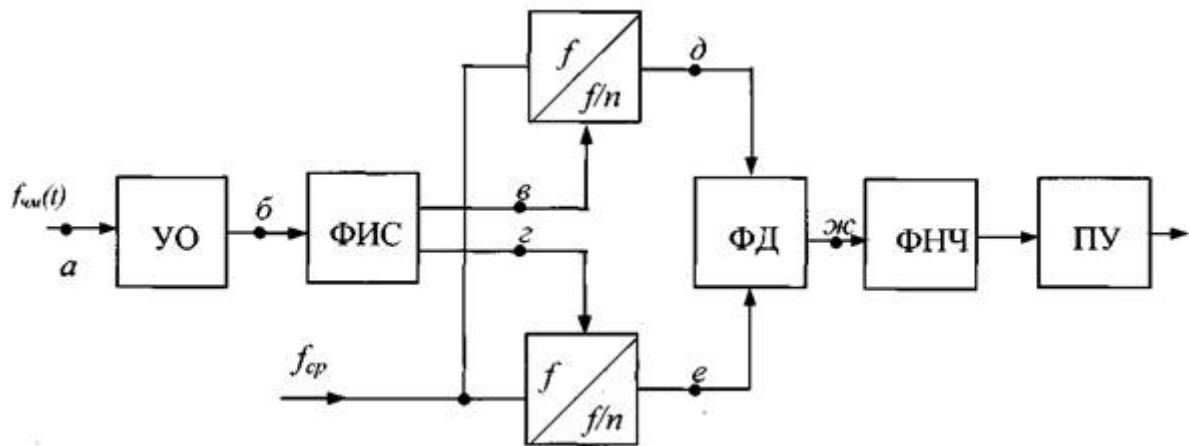


Рисунок 2.11 — Структурная схема цифрового частотного демодулятора

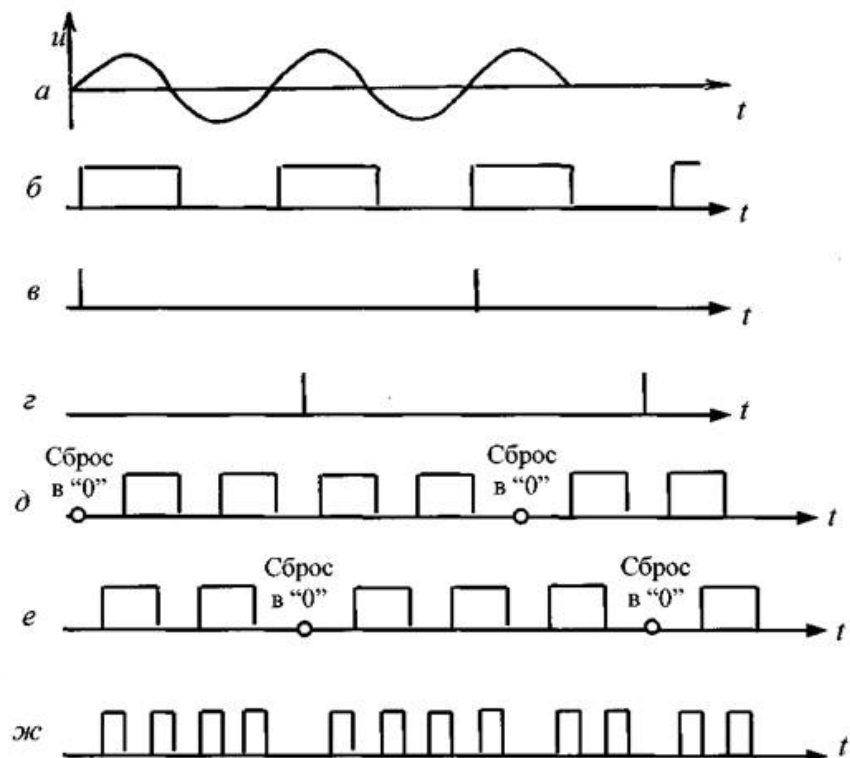


Рисунок 2.12 — Временные диаграммы цифрового частотного демодулятора

Входной сигнал преобразуется усилителем-ограничителем УО в прямоугольные импульсы (б).

В формирователе импульсов сброса ФИС выделяются короткие импульсы, соответствующие каждому периоду входного сигнала из импульсов(б). Короткие импульсы подаются поочерёдно на делители частоты (в,г), устанавливая их в начальное состояние (обозначено точками, д и е). Импульсы на выходе делителей при приёме средней частоты f_{cp} изображены на д и е. В этом случае между сигналами на выходе делителей сдвиг по фазе равен четверти периода, причём знак сдвига фаз меняется после поступления каждого импульса сброса. Сигналы с выходов делителей поступают на вход фазового детектора ФД (выполненного в виде сумматора по mod2), на выходе которого возникает последовательность импульсов (ж), ширина каждого из которых зависит от соотношения фаз сигналов на выходах делителей.

При появлении на входе демодулятора частоты $f_2 > f_{cp}$ последовательность импульсов становится шире, а при появлении на входе демодулятора частоты $f_1 < f_{cp}$ — уже.

2.2.4 Построение частотного приёмника

На вход приёмника поступает частотно-манипулированный сигнал, для того бы подавить помехи, не прибегая к применению частотных фильтров, будем переводить принятый сигнал в импульсы, которые в дальнейшем будем считать. Для того чтобы перевести принятый сигнал в импульсы введём понятие состояние приёмника, которое может принимать 3 значения:

- «0» — приёмник в первоначальном состоянии
- «1» — приёмник принял сигнал больше порога с положительной амплитудой
- «-1» — приёмник принял сигнал больше порога с отрицательной амплитудой.

В начале каждого периода приёмник переводится в состояние «0», переход в любое другое состояние инициирует появление импульса на выходе. Легко заметить что 1 импульс во время периода соответствует принятому «0», а 2 импульса принятой «1».

$$\begin{aligned}
 \text{fd}(s) := & \left| \begin{array}{l} \text{for } i \in 0.. \frac{K}{N} - 1 \\ \quad t \leftarrow 0 \\ \quad \text{state} \leftarrow 0 \\ \quad \text{for } k \in N \cdot i.. N \cdot (i + 1) - 1 \\ \quad \quad \text{newState} \leftarrow \begin{cases} 1 & \text{if } s_k > Fd \\ (-1) & \text{if } s_k < -Fd \\ 0 & \text{otherwise} \end{cases} \\ \quad \quad t \leftarrow t + 1 & \text{ if } \text{newState} \neq 0 \wedge (\text{state} = 0 \vee \text{newState} \neq \text{state}) \\ \quad \quad \text{state} \leftarrow \text{newState} & \text{ if } \text{newState} \neq 0 \\ \quad b_{0,i} \leftarrow t \\ \quad b_{1,i} \leftarrow \text{if}(t > 1, 1, 0) \end{array} \right| \\
 & \text{b}
 \end{aligned}
 \qquad
 \text{sb} := \text{fd}(s) = \begin{pmatrix} 1 & 2 & 1 & 2 & 2 & 2 & 1 & 2 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Исходный входной сигнал и соответствующий ему выходной изображены на рис. 2.13 и рис. 2.14.

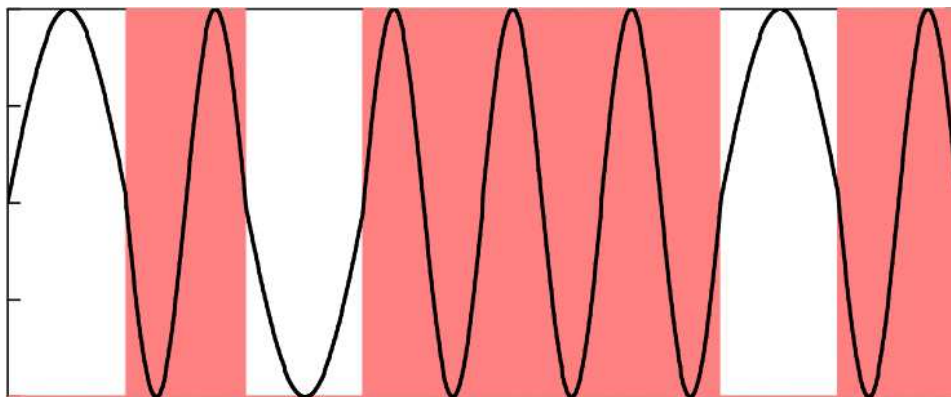


Рисунок 2.13 — Исходный манипулированный сигнал



Рисунок 2.14 — Демодулированный сигнал

Проверка на помехоустойчивость будет проходить образом подобному описанному в пункте 1.2.2:

$$ss_k := s_k + \text{rnd}(0.5) - \text{rnd}(0.5)$$

Манипулированный сигнал с помехами изображён на рис. 15. Демодулированный сигнал изображён на рис. 14 (совпадает со случаем без помех).

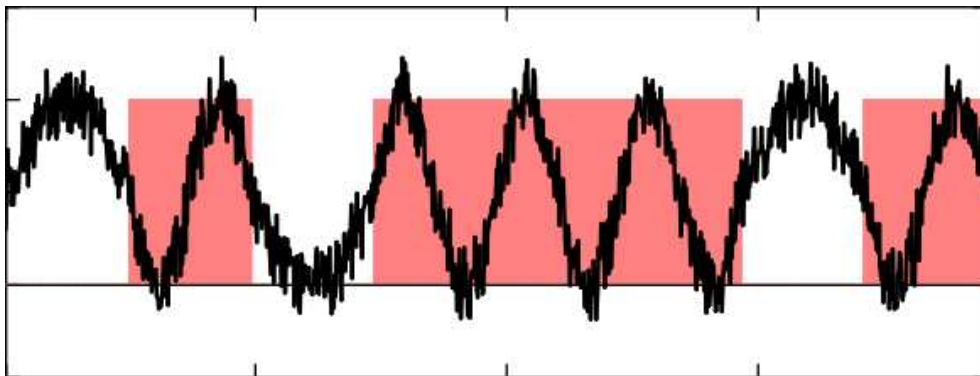


Рисунок 2.15 — Манипулированных сигнал с помехами

2.3 Контрольные вопросы

- Определения модуляции и манипуляции;
- Определения демодуляции и деманипуляции;
- Суть фазовой модуляции, манипуляции и обратных им процессов;
- Суть частотной модуляции, манипуляции и обратных им процессов;

- Суть корреляционного приёмника;
- Суть частотного демодулятора.

2.4 Задание

1. Варианты исходных данных приведены в приложении Б;
2. Составить программу с частотной и фазовой манипуляцией;
3. Исследовать спектры полученных сигналов;
4. Принять манипулированные сигналы для частотной и фазовой манипуляции;
5. Принять манипулированные сигналы с помехами для частотной и фазовой манипуляции;
6. Сделать выводы.
7. Оформить отчет.

2.5 Литература

1. Зюко А.Г., Кловский Д.Д., Назаров М.В., Финк Л.М. Теория передачи сигналов. Москва, «Связь» 1980г.
2. Савчук В.Л. Электронные средства сбора, обработки и отображения информации: Учебное пособие. — Томск: Томский государственный университет систем управления и радиоэлектроники, 2007.
3. Частотно-манипулированные сигналы FSK (frequency shift key) и FSK сигналы с непрерывной фазой CPFSK (continuous phase FSK), <http://www.dsplib.ru/content/fsk/fsk.html>
4. http://ru.wikipedia.org/wiki/Фазовая_модуляция
5. http://ru.wikipedia.org/wiki/Частотная_модуляция

Лабораторная работа №3. Помехозащищающие коды и механизмы избыточности

Обнаружение ошибок в технике связи — действие, направленное на контроль целостности данных при записи/воспроизведении информации или

при её передаче по линиям связи. Исправление ошибок (коррекция ошибок) — процедура восстановления информации после чтения её из устройства хранения или канала связи.

Для обнаружения ошибок используют коды обнаружения ошибок, для исправления — корректирующие коды (коды, исправляющие ошибки, коды с коррекцией ошибок, помехоустойчивые коды).

История кодирования, контролирующего ошибки, началась в 1948 г. публикацией знаменитой статьи Клода Шеннона. Шеннон показал, что с каждым каналом связано измеряемое в битах в секунду и называемое пропускной способностью канала число C . Если требуемая от системы связи скорость передачи информации R (измеряемая в битах в секунду) меньше C , то, используя коды, контролирующие ошибки, для данного канала можно построить такую систему связи, что вероятность ошибки на выходе будет сколь угодно мала. В самом деле, из шенноновской теории информации следует тот важный вывод, что построение слишком хороших каналов является расточительством; экономически выгоднее использовать кодирование. Фактически в работе Шеннона утверждается, что мощность сигнала, шум в канале и полоса частот ограничивают лишь скорость передачи, а не ее точность. Шеннон, однако, не указал, как найти подходящие коды, а лишь доказал их существование. В пятидесятые годы много усилий было потрачено на попытки построения в явном виде классов кодов, позволяющих получить обещанную сколь угодно малую вероятность ошибки, но результаты были скудными. В следующем десятилетии решению этой увлекательной задачи уделялось меньше внимания; вместо этого исследователи кодов предприняли длительную атаку по двум основным направлениям.

Первое направление носило чисто алгебраический характер и преимущественно рассматривало блочные коды. Первые блочные коды были введены в 1950 г., когда Хэмминг описал класс блочных кодов, исправляющих одиночные ошибки. Коды Хэмминга были разочаровывающе слабы по сравнению с обещанными Шенноном гораздо более сильными кодами.

Несмотря на усиленные исследования, до конца пятидесятих годов не было построено лучшего класса кодов. В течение этого периода без какой-либо общей теории были найдены многие коды с малой длиной блока. Основной сдвиг произошёл, когда Боуз и Рой-Чоудхури [1960] и Хоквингем [1959] нашли большой класс кодов, исправляющих кратные ошибки (коды БЧХ), а Рид и Соломон [1960] нашли связанный с кодами БЧХ класс кодов для не двоичных каналов. Хотя эти коды остаются среди наиболее важных классов кодов, общая теория блочных кодов, контролирующая ошибки, с тех пор успешно развивалась.

Открытие кодов БЧХ привело к поиску практических методов построения жестких или мягких реализации кодеров и декодеров. Первый хороший алгоритм был предложен Питерсоном. Впоследствии мощный алгоритм выполнения описанных Питерсоном вычислений был предложен Берлекэмпом и Месси, и их реализация вошла в практику как только стала доступной новая цифровая техника. Второе направление исследований по кодированию носило скорее вероятностный характер. Ранние исследования были связаны с оценками вероятностей ошибки для лучших семейств блочных кодов, несмотря на то, что эти лучшие коды не были известны. С этими исследованиями были связаны попытки понять кодирование и декодирование с вероятностной точки зрения, и эти попытки привели к появлению последовательного декодирования. В последовательном декодировании вводится класс неблочных кодов бесконечной длины, которые можно описать деревом и декодировать с помощью алгоритмов поиска по дереву. Наиболее полезными древовидными кодами являются коды с тонкой структурой, известные под названием сверточных кодов. Эти коды можно генерировать с помощью цепей линейных регистров сдвига, выполняющих операцию свертки информационной последовательности. В конце 50-х годов для сверточных кодов были успешно разработаны алгоритмы последовательного декодирования. Интересно, что наиболее простой алгоритм декодирования — алгоритм Витерби — не был разработан для этих кодов до 1967 г. Применительно к

сверточным кодам умеренной сложности алгоритм Витерби пользуется широкой популярностью, но для более мощных сверточных кодов он не практичен.

В 70-х годах эти два направления исследований опять стали переплетаться. Теорией сверточных кодов занялись алгебраисты, представившие ее в новом свете. В теории блочных кодов за это время удалось приблизиться к кодам, обещанным Шенноном: были предложены две различные схемы кодирования (одна Юстесеном, а другая Гоппой), позволяющие строить семейства кодов, которые одновременно могут иметь очень большую длину блока и очень хорошие характеристики. Обе схемы, однако, имеют практические ограничения. Между тем к началу 80-х годов кодеры и декодеры начали появляться в конструкциях цифровых систем связи и цифровых систем памяти.

3.1 Цепной код

Цепной код является самым простым из всех известных сверточных кодов, которые относятся к непрерывным или рекуррентным кодам.

В цепном коде каждый проверочный элемент формируется путём сложения двух информационных элементов, отстоящих один от другого на t элементов, по модулю 2:

$$\text{mod}(a_k, a_{k+t}) = b_{k,k+t}, \quad (3.1)$$

где t — шаг сложения (рис. 3.1).

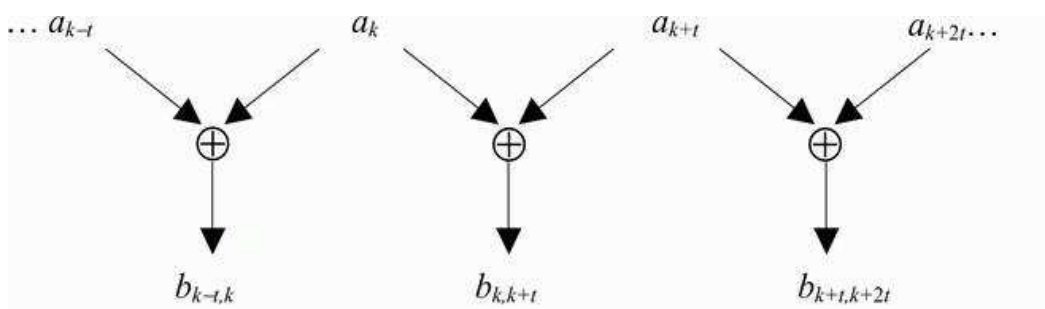


Рисунок 3.1 — Формирование проверочных элементов цепного цикла

Поскольку каждый информационный элемент участвует в формировании двух проверочных элементов, а каждый проверочный элемент формируется по двум информационным, число проверочных элементов, сформированных за время t , будет равно числу информационных элементов, поступивших за то же время на вход кодирующего устройства.

Следовательно, избыточность цепного кода равна $1/2$.

В канал связи передаётся последовательность импульсов, в которой за каждым информационным следует проверочный $a_1, b_2, a_3, b_4, a_5, b_6$ и т. д.

Принцип декодирования принимаемой последовательности единичных элементов и вытекающие из него корректирующие возможности цепного кода становятся очевидными при анализе принципа формирования проверочных элементов.

При декодировании принимаемая последовательность единичных элементов делится на две независимые последовательности: информационную и проверочную. Из информационной последовательности по тому же принципу, что и при кодировании, снова формируется проверочная последовательность.

Единичные элементы вновь полученной проверочной последовательности сравниваются с единичными элементами принятой проверочной последовательности. Если соответствующие сравниваемые элементы совпадают, считается, что ошибки нет, в противном случае должно быть принято решение об ошибке.

Изменение значения информационного элемента, например, a_k (переход 1 в 0 или 0 в 1), приводит к изменению значений двух связанных с ним проверочных элементов $b_{k-t,k}$ и $b_{k,k+t}$. Наличие двух несовпадений проверочных элементов, сдвинутых друг от друга на t информационных элементов, указывает на искажение информационного элемента, общего для обоих проверочных элементов (a_k в рассматриваемом примере). В этом случае значение a_k необходимо инвертировать. Регистрация несовпадения лишь

одного проверочного элемента указывает на ошибочный приём только этого проверочного элемента.

Для исправления ошибочно принятого информационного элемента (например, a_k) необходимо, чтобы предыдущий и последующий информационные элементы этой цепи (a_{k-t} и a_{k+t}), а также соответствующие проверочные элементы ($b_{k-t,k}$ и $b_{k,k+t}$) были приняты правильно.

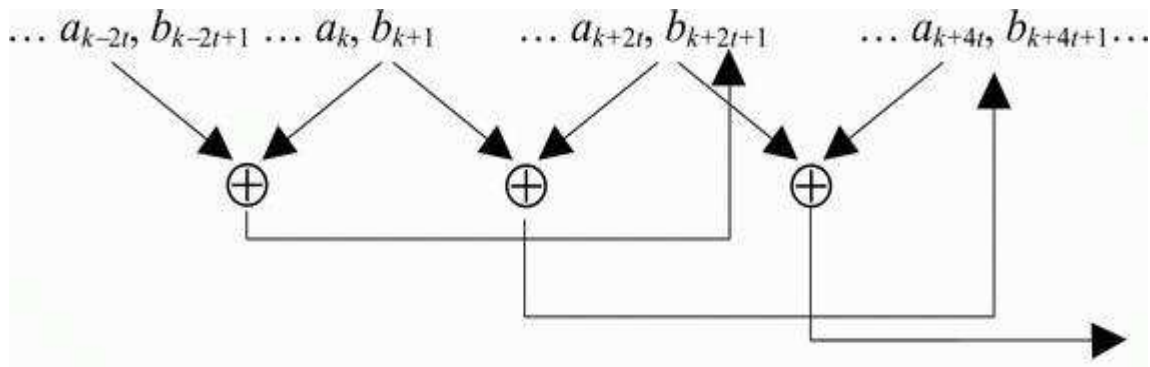


Рисунок 3.2 — Последовательность единичных элементов, передаваемых в канал связи

Это условие выполняется полностью, если длина пакета ошибок не превышает величины $2t$ (t информационных и t проверочных, которые передаются в канал связи с задержкой $2t + 1$).

При этом необходимо, чтобы между последним элементом данного пакета ошибок и первым элементом последующего пакета ошибок было принято подряд не менее $4t$ неискаженных элементов.

Несмотря на свою простоту, цепной код позволяет исправлять одиночные и групповые ошибки.

3.2 Применение цепного кода на практике

Реализация кодирования и декодирования информации будет осуществляться на MathCAD.

Пусть мы имеем последовательность 128 битов информации, сформированной случайным образом.

$$x_i := \text{floor}(\text{rnd}(2)) \quad N := 128 \quad i := 0..N-1 \quad K := 8, \quad (3.2)$$

где x — исходный случайный сигнал,

N — кол-во точек,

K — шаг сложения.

Сформированный сигнал показан на рис. 3.3.

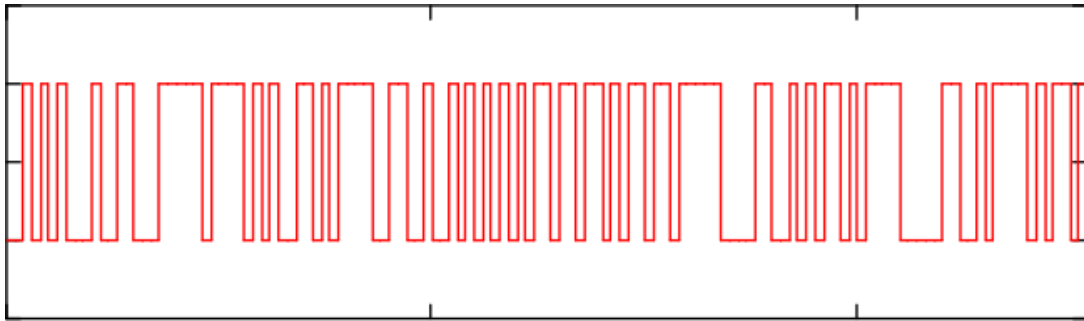


Рисунок 3.3 — Исходный случайный сигнал

Теперь формируется код, в котором после каждой информационной посылки следует проверочная, которая формируется путём сложения по модулю двух информационных посылок, отстоящих одна от другой на шаг сложения k , в данном случае $k=8$. Стоит заметить что подобный сдвиг вызовет появление нескольких лишних бит.

$$\text{code}(x, k) := \begin{array}{l} \text{for } i \in 0.. \text{last}(x) \\ \quad s_{2 \cdot i} \leftarrow x_i \\ \quad s_{2 \cdot i + 1} \leftarrow \begin{array}{l} x_i \oplus x_{i-k} \text{ if } i - k \geq 0 \\ x_i \text{ otherwise} \end{array} \\ \text{for } i \in \text{length}(x) .. \text{length}(x) + k - 1 \\ \quad s_{2 \cdot i + 1} \leftarrow x_{i-k} \end{array} \quad s, \quad (3.3)$$

Сформированный сигнал, содержащий как информационные, так и проверочные посылки показан на рис. 3.4.



Рисунок 3.4 — Сформированный сигнал

Применим над исходным сигналом помехи.

$$\text{noise}(x) := \left\{ \begin{array}{l} \text{MIN} \leftarrow 2 \\ \text{MAX} \leftarrow 5 \\ \text{for } i \in 0.. \text{MIN} + \text{floor}(\text{rnd}(\text{MAX} - \text{MIN} + 1)) - 1 \\ \quad \left\{ \begin{array}{l} r_i \leftarrow \text{floor}(\text{rnd}(\text{length}(x))) \\ x_{r_i} \leftarrow \neg x_{r_i} \end{array} \right. \end{array} \right. x, \quad (3.4)$$

Для наглядности на рис.5 изображена разница между переданным и принятым сигналом, на рис. 3.6 изображена разность между исходными и выделенными из закодированного сигнала с ошибками информационными посылками, на рис. 3.7 разность проверочных посылок сформированных при кодировании и выделенной посылки из сигнала с ошибками.

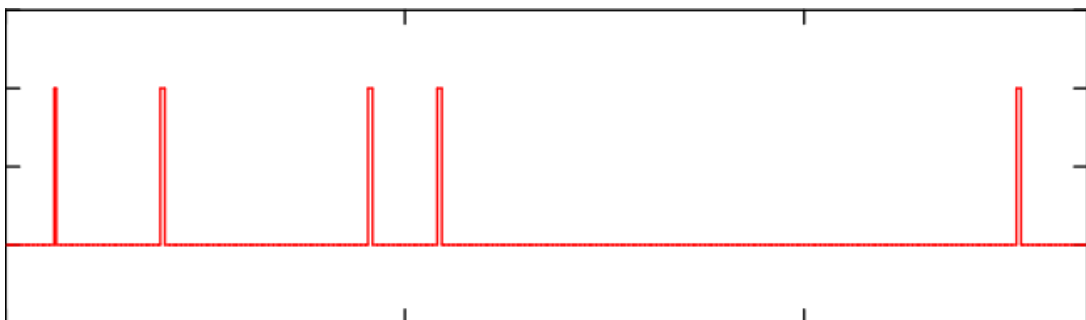


Рисунок 3.5 — Разница между переданным и принятым сигналом

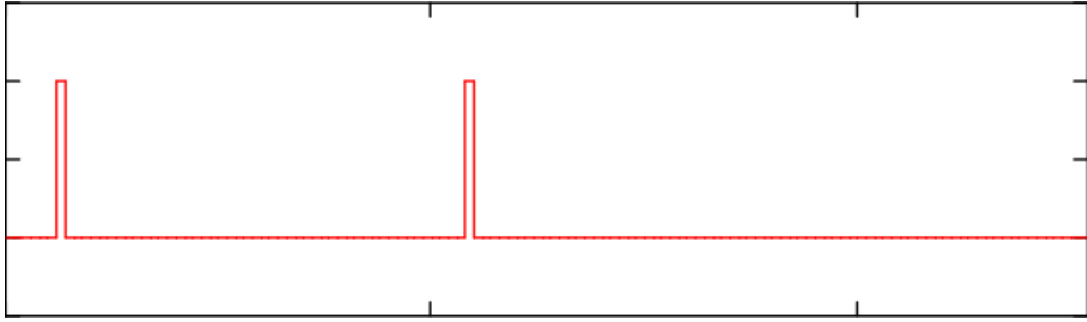


Рисунок 3.6 — Разница между информационными посылками

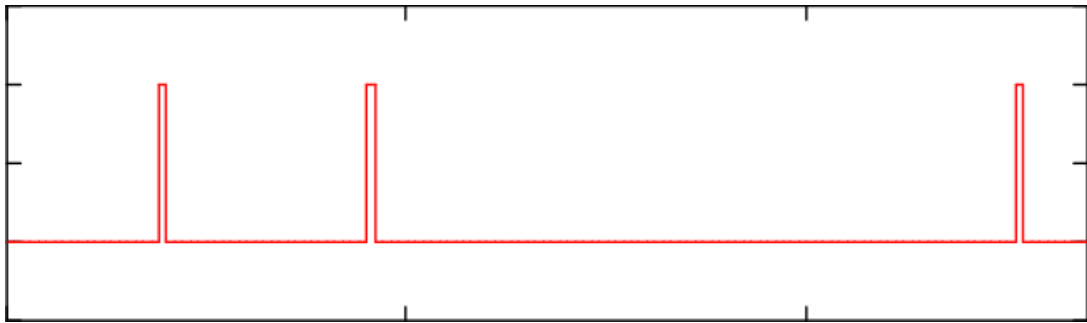


Рисунок 3.7 — Разница между проверочными посылками

При приёме информационные и проверочные посылки разделяются и регистрируются независимо друг от друга. Затем из принятой последовательности информационных посылок формируются контрольные, как при передаче, которая затем сравнивается с принятой и проверочной.

$$\text{errors}(x, c, k) := \begin{array}{l} \text{for } i \in 0 \dots \text{last}(x) \\ \quad \left| \begin{array}{l} c2 \leftarrow \begin{cases} x_i \oplus x_{i-k} & \text{if } i - k \geq 0 \\ x_i & \text{otherwise} \end{cases} \\ s_i \leftarrow |c_i - c2| \end{array} \right. \\ \text{for } i \in \text{length}(x) \dots \text{length}(x) + k - 1 \\ \quad s_i \leftarrow |c_i - x_{i-k}| \\ s \end{array}, \quad (3.5)$$

На рис. 3.8 — разность между контрольной и проверочной посылками. Причём, если ошибка была в информационном разряде, то она выглядит

продублированной, т.е. двойной на расстоянии $k=8$. Если ошибка возникла в проверочных разрядах, то она не дублируется и затем просто игнорируется.

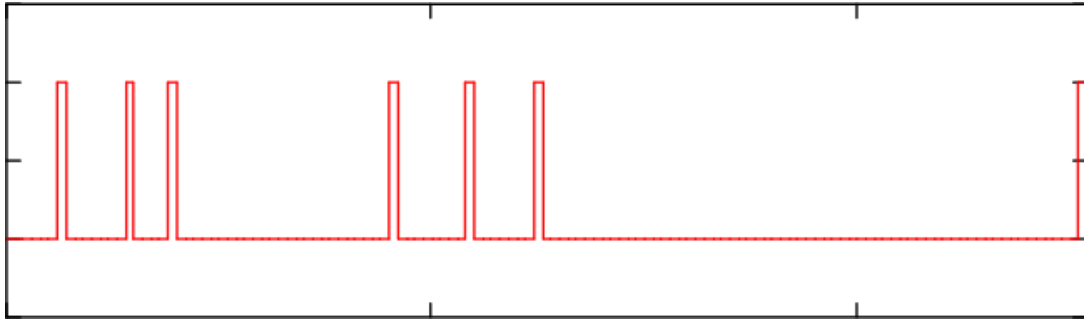


Рисунок 3.8 — Разность между контрольной и проверочной посылкой

Исправляем i -й разряд в принятой информационной последовательности, если у нас в разности, изображённой на рис. 3.8, на i -м и $(i+k)$ -м месте присутствует импульс и т. д.

$$\text{decode}(x, c, k) := \left| \begin{array}{l} e \leftarrow \text{errors}(x, c, k) \\ \text{for } i \in 0.. \text{last}(x) \\ \quad x_i \leftarrow \neg x_i \text{ if } e_i = 1 \wedge e_{i+k} = 1 \\ x \end{array} \right. , \quad (3.6)$$

В итоге получаем первоначальную информационную посылку без ошибок (рис. 3.9) из сигнала принятого с ошибками. Для наглядности на рис. 10 изображена разница между исходной и восстановленной информационной посылкой.

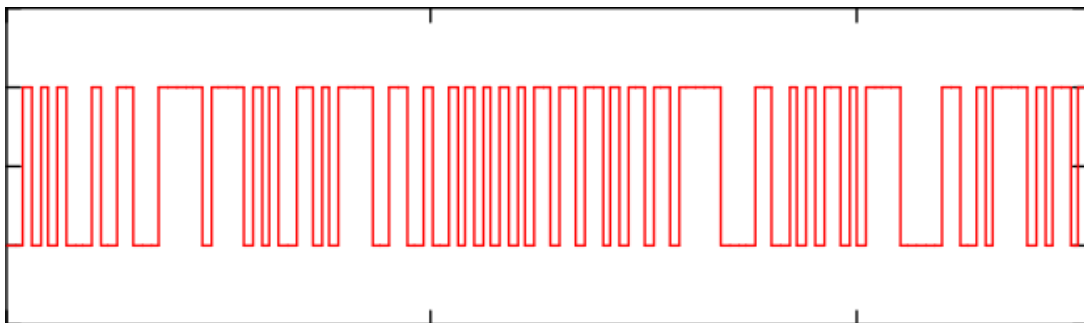


Рисунок 3.9. Восстановленная информационная посылка



Рисунок 3.10. Разница между исходной и восстановленной информационной посылкой

3.3 Контрольные вопросы

- Для чего применяется избыточность;
- Для чего применяется помехоустойчивое кодирование;
- Алгоритмы помехоустойчивого кодирования;
- Вероятность возникновения ошибки в восстановленной информационной посылке, при условии что ровно 2 бита в принимаемом сигнале будут искажены.
- Вероятность возникновения ошибки в восстановленной информационной посылке, при условии что вероятность искажения одного бита (любого) при передаче равна 0.001.

3.4 Задание

1. Повторить все графики приведённые в данном методическом пособии для своего случайного сигнала (8 графиков). Знать алгоритмы всех используемых функций. Уметь сделать выводы по каждому графику.
2. Самостоятельно изучить как минимум 2 других алгоритма помехоустойчивого кодирования.
3. Оформить отчет.

3.5 Литература

1. Помехоустойчивое кодирование. URL: <http://kunegin.narod.ru/ref3/code/index.htm> (дата обращения 11.03.2012)
2. Исследование помехоустойчивости цепного кода. URL: <http://dvo.sut.ru/libr/opds/i285vino/4.htm> (дата обращения 11.03.2012)
3. Р. Морелос-Сарагоса. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. Москва. Техносфера, 2005. — 320 с.

Лабораторная работа №4. Многоканальные системы передачи данных

Любой сигнал занимает определённую полосу частот, существует некоторое время, обладает ограниченной энергией и распространяется в определённой области пространства. В соответствии с этим выделяют четыре вида ресурса канала: частотный, временной, энергетический и пространственный.

Проблема эффективного использования ресурса общего канала особенно обострилась из-за необходимости организации оперативного обмена данными и обеспечения связи с объектами в информационных системах различного назначения в условиях неравномерности и непредсказуемости запросов потребителей во времени. При решении проблемы распределения ресурса общего канала применяются методы мультиплексирования и множественного доступа (multiple access). Понятия «мультиплексирование» и «множественного доступа» сходны тем, что они предполагают распределение ресурса между пользователями. В то же время между ними есть и существенные различия. Так при мультиплексировании ресурс канала связи распределяется через общее оконечное оборудование, формирующие групповой сигнал. При множественном доступе, образуется в результате сложения сигналов пользователей непосредственно в канале (рис. 4.1, где ИС — источник сообщения, ПРД — передатчик, ПРМ — приемник, ПС — получатель сообщения). Множественный доступ характерен для спутниковых каналов, радиоканалов, каналов мобильной связи.

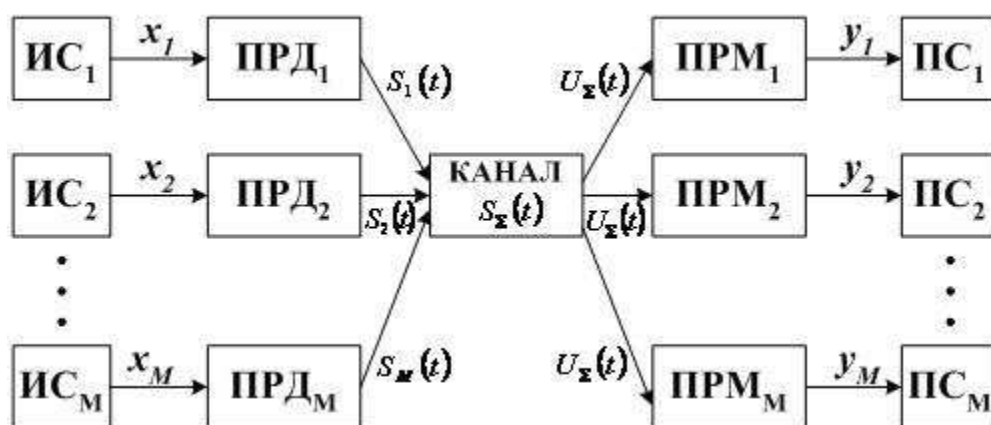


Рисунок 4.1 — Системы передачи со множественным доступом

Принято считать, что мультиплексирование основано на общем аппаратном обеспечении, в то время как множественный доступ (МД) использует определённые процедуры (протоколы), реализуемые с помощью программного обеспечения, хранящегося в памяти каждого терминала.

В большинстве случаев для осуществления операции уплотнения канала источнику сообщений выделяется специальный сигнал, называемый канальным. Промодулированные сообщениями канальные сигналы объединяются, в результате чего образуется групповой сигнал $S_{gp}(t)$. Если операция объединения линейна, то $S_{gp}(t)$ будет линейным групповым сигналом. Он, как правило, образуется линейным суммированием промодулированных канальных сигналов.

В системах так называемого комбинационного уплотнения групповой сигнал формируется посредством определённой логической (нелинейной) обработки, в результате которой каждый элемент сформированного сигнала отображает информацию (комбинацию символов) от всех ИС. Классическим примером такой системы является система двукратного частотного телеграфирования. Для передачи четырёх комбинаций символов двух каналов используется четыре частоты: $f_1 \rightarrow 00$, $f_2 \rightarrow 01$, $f_3 \rightarrow 10$, $f_4 \rightarrow 11$.

Устройство разделения линейного группового сигнала представляет собой набор линейных избирательных цепей, каждая из которых выделяет

только свой канальный сигнал и в идеальном случае совсем не реагирует на другие канальные сигналы. Для осуществления подобного идеального разделения необходимо и достаточно, чтобы промодулированные канальные сигналы составляли ансамбль линейно независимых сигналов. В качестве таких сигналов обычно используют ансамбли ортогональных сигналов.

В классе линейного уплотнения по виду отличительного признака канального сигнала различают временное разделение каналов (ВРК), частотное (ЧРК) и разделение каналов по форме сигналов, называемое кодовым разделением каналов (КРК). Вместо термина «разделение» применяют и термин «уплотнение». При ЧРК полоса частот общего канала Δf разделяется на несколько более узких полос Δf_i , каждая из которых образует канал ИС. При ВРК вся полоса Δf предоставляется поочерёдно через определённые интервалы времени различным источникам для передачи сообщений. При КРК нет деления общего канала между ИС ни по частоте, ни по времени. Канальные сигналы различных ИС, перекрываясь по времени и частоте, остаются ортогональными за счёт различия формы, что и обеспечивает их разделение.

Возможны варианты комбинирования указанных методов. Так, в мобильной связи в качестве метода МД широко используются комбинации ЧРК и ВРК, ВРК и КРК. В первой комбинации каждый частотный канал предоставляется нескольким пользователям на определённые промежутки времени. При второй комбинации в полосе частот Δf формируют каналы с временным разделением, которые предоставляются нескольким пользователям на принципах КРК.

Основные понятия и определения, введённые для многоканальных систем, применимы и для систем МД. К настоящему времени изучено и предложено большое число разнообразных методов МД. Они различаются способом распределения коллективного ресурса канала (фиксированный или динамический), природой процессов принятия решения (централизованные или распределенные), а также степенью адаптации режима доступа к изменяющимся условиям.

Протоколы фиксированного закрепления ресурса канала обеспечивают статическое распределение ресурса канала между пользователями. Наиболее типичными представителями протоколов данного типа являются многостанционный доступ с частотным разделением (FDMA), многостанционный доступ с временным разделением (TDMA), многостанционный доступ с кодовым разделением (CDMA).

Фиксированное закрепление ресурса канала не может обеспечить динамически изменяющиеся требования пользователей сети, т.е. имеет жёсткое управление.

4.1 Принцип частотного объединения и разделения каналов

Практика построения современных телекоммуникационных систем и сетей показывает, что наиболее дорогостоящими звеньями трактов передачи являются линии связи (кабельные, волоконно-оптические, радиосвязи, радиорелейные и др.). Поскольку экономически нецелесообразно использовать дорогостоящую линию связи для передачи информации единственной пары абонентов (от источника к получателю сообщений и обратно при дуплексной связи), то возникает задача построения многоканальных систем передачи, обеспечивающих передачу большого числа сообщений различных источников информации по общей линии связи. Многоканальные системы так же, как и одноканальные, могут быть аналоговыми и цифровыми.

При частотном разделении каналов для передачи данных различных источников сообщений используются определенные поддиапазоны частот. Функциональная схема простейшей системы многоканальной связи с частотным разделением каналов (ЧРК) представлена на рис. 4.2.

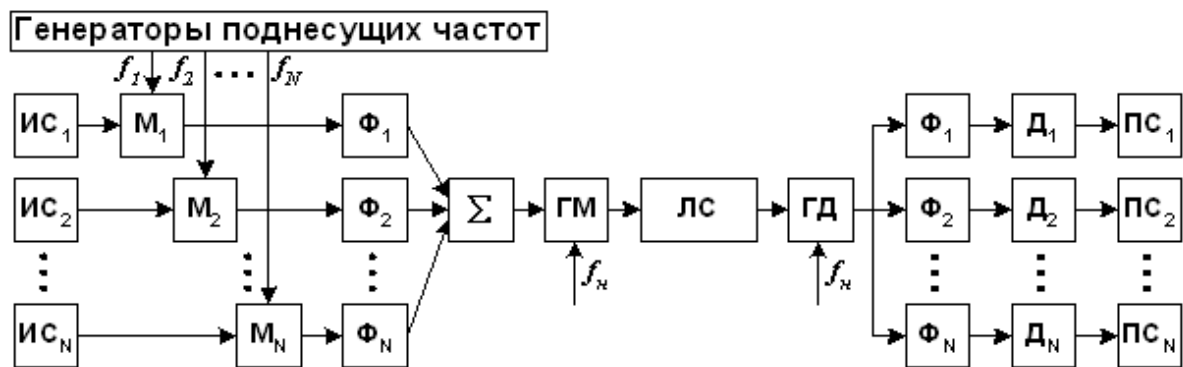


Рисунок 4.2 — Функциональная схема СЭС с ЧРК

Основные этапы образования спектра $A_{\Sigma}(f)$ группового сигнала показаны на рис. 4.3. Пусть в СЭС осуществляется одновременная работа N корреспондентов. В соответствии с передаваемыми сообщениями первичные сигналы от источников сообщений, имеющие энергетические спектры , модулируют поднесущие частоты f_k каждого канала.

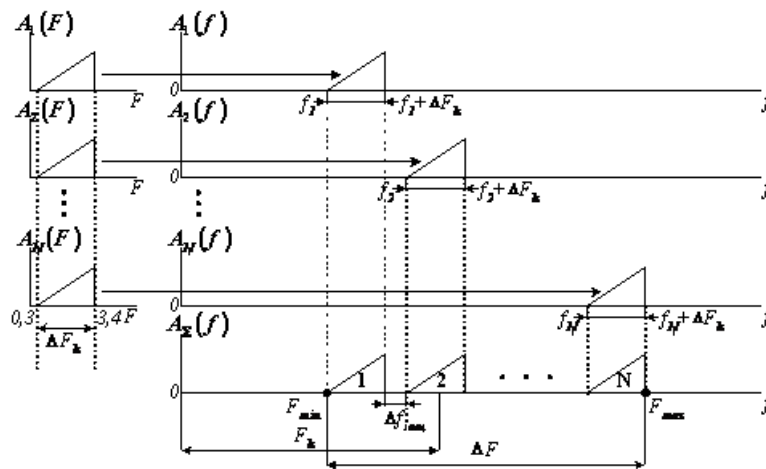


Рисунок 4.3 — Формирование группового спектра при ЧРК

Наиболее распространённый при ЧРК вариант — однополосная амплитудная модуляция. Полученные на выходе полосовых фильтров $\Phi_1, \Phi_2, \dots, \Phi_N$ каналные сигналы суммируются, и их совокупность $A_{\Sigma}(f)$ поступает на групповой модулятор. Здесь спектр $A_{\Sigma}(f)$ с помощью колебания несущей частоты f_n переносится в область частот, отведённую для передачи

данной группы каналов, т. е. групповой сигнал преобразуется в так называемый линейный сигнал, передаваемый по линии связи — кабелю, радио, радиорелейной, спутниковой линии связи. При этом может использоваться любой вид модуляции. На приёмном конце осуществляется вся совокупность обратных преобразований. Групповым демодулятором линейный сигнал преобразуется в групповой, из которого с помощью фильтров выделяются каналные сигналы. С помощью детекторов каналные сигналы преобразуются в ПЭС поступающие к получателям.

4.2 Временное разделение каналов

В настоящее время передача информации в радиорелейных линиях связи осуществляется как методом частотного разделения каналов (ЧРК), так и методом временного разделения каналов (ВРК). Радиорелейные линии с ЧРК-ЧМ обладают сравнительно высокими технико-экономическими показателями, однако они имеют существенный недостаток: трудность осуществления передачи части каналов для группы абонентов, находящихся на промежуточных станциях линии. Каждое ответвление связано с разуплотнением каналов на промежуточных станциях, выделением части каналов для группы абонентов и повторим уплотнением каналов для передачи остальных по линии связи.

Широкое применение нашли радиорелейные линии с ВРК. Их основное достоинство состоит в простоте выделения групп каналов, что весьма важно при создании подвижных радиорелейных станций.

Временное разделение основано на возможности передачи вместо непрерывных сигналов последовательных импульсов (отсчётов). Поскольку при импульсной передаче период следования импульсов обычно намного больше их длительности (импульсы имеют большую скважность), между импульсами одного сигнала остаётся промежуток, на котором можно разместить импульсы от других сигналов. В настоящее время уже реализованы многоканальные системы с временным разделением 12, 15, 30, 120, 480 речевых сигналов.

Радиорелейные линии с ВРК предполагают использование как аналоговых: амплитудно-импульсная модуляция (АИМ), широтно-импульсная модуляция (ШИМ), фазо-импульсная модуляция (ФИМ), так и цифровых: импульсно кодовая модуляция (ИКМ), дельта-модуляция (ДМ) методов импульсной модуляции.

Многоканальные системы с ВРК широко используются для передачи аналоговой и дискретной информации.

Принцип временного объединения каналов удобно пояснить с помощью синхронно вращающихся распределителей на передающей и приёмной стороне рис. 4.4.

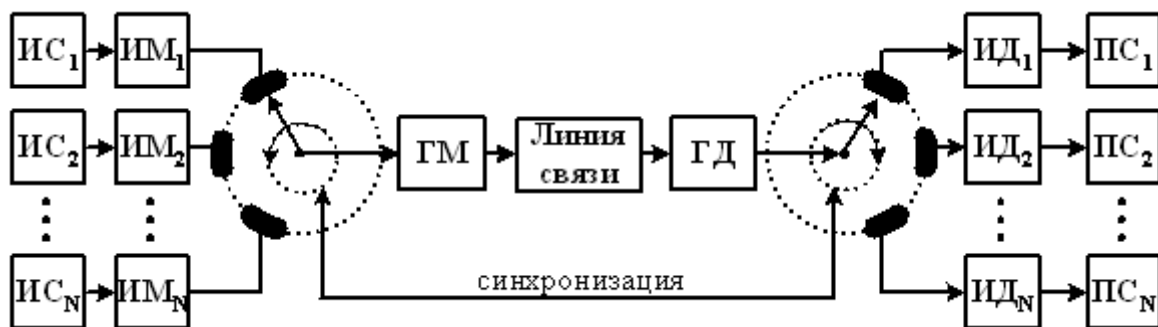


Рисунок 4.4 — Упрощённая блок-схема СЭС с ВРК

Основные этапы образования группового сигнала $S_{\Sigma}(t)$ показаны на рис. 4.5.

приёмной стороне с выхода группового демодулятора (ГД) импульсы группового сигнала $S_{\Sigma}(t)$ поступают на вращающиеся контакты приёмного распределителя для формирования канальных последовательностей $S_1(t)$, $S_2(t)$ и т.д. из которых на выходе импульсных детекторов формируются непрерывные сигналы поступающие к получателям сообщений.

Следует подчеркнуть, что рис. 4.3 служит лишь для иллюстрации идеи временного уплотнения и не отражает современных технических методов коммутации. В действительности аппаратура временного уплотнения обходится без механических распределителей, которые заменены электронными распределителями, выполняющими те же функции (рис. 4.6).

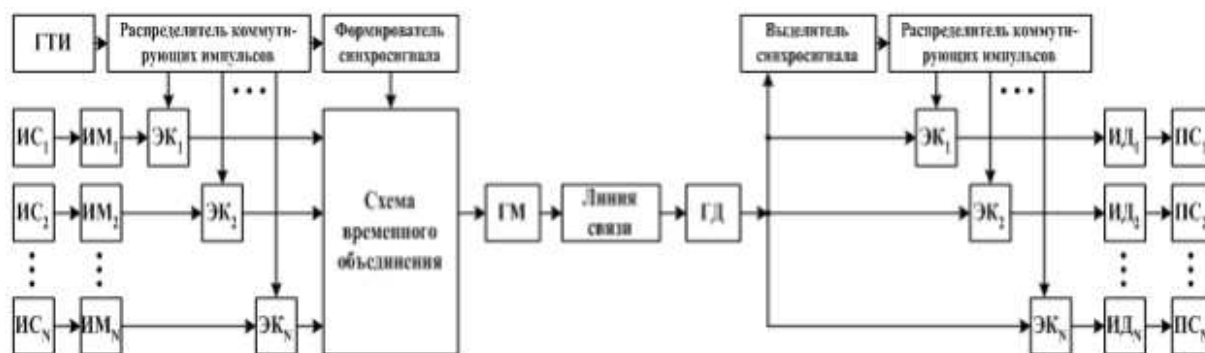


Рисунок 4.6 — Схема многоканальной связи с ВРК

Выходы всех импульсных модуляторов подключены к «своим» электронным ключам, работой которых управляет распределитель коммутирующих импульсов. В свою очередь, распределитель запускается от генератора тактовых импульсов.

Временное разделение сигналов осуществляется устройством, упрощённая структурная схема которого представлена на рис. 4.5. Принятый групповой радиосигнал в групповом демодуляторе преобразуется в групповую импульсную последовательность и поступает одновременно на входы выделителя синхросигнала и каналные электронные коммутаторы.

Процесс временного разделения производится в два этапа. На первом — этапе вхождения системы в синхронизм происходят поиск, обнаружение и

выделение сигналов синхронизации, после чего запускается распределитель канальных коммутирующих импульсов. Распределитель формирует на своих выходах импульсы требуемой длительности и такой очередности, при которой в каждый канальный интервал открывается лишь один электронный коммутатор соответствующего канала.

На втором этапе производится демодуляция каждого канального импульса, после чего сигналы принимаемых каналов подаются к получателям аналоговой информации.

При временном разделении каналов важнейшую роль играет система синхронизации, алгоритм работы которой каждый раз выбирается индивидуально для принятого способа импульсной модуляции, способа временного объединения каналов, структуры сигналов синхронизации и т. д.

4.3 Принцип частотного объединения и разделения каналов на практике

Моделирование будет происходить в пакете MathCAD. Моделирование будет проходить со следующими начальными параметрами:

$$\begin{aligned} s1(t) &:= 1 + \sin(t \cdot \sin(t)) & s2(t) &:= 1 + \cos(t \cdot \sin(t)) & \Delta &:= 0.01 & \Delta T &:= 0.1 & T &:= 6 \\ \omega_1 &:= 20 & \Delta\omega &:= 15 & \omega_2 &:= \omega_1 + 2\Delta\omega \end{aligned} \quad (4.1)$$

где $s1(t), s2(t)$ — исходные сигналы;

Δ — элементарный квант времени;

ΔT — период передачи одного сигнала для ВРК;

T — время моделирования;

ω_1 — частота несущей для первого сигнала

$\Delta\omega$ — безопасный запас частот.

Исходные сигналы изображены на рис. 4.7.

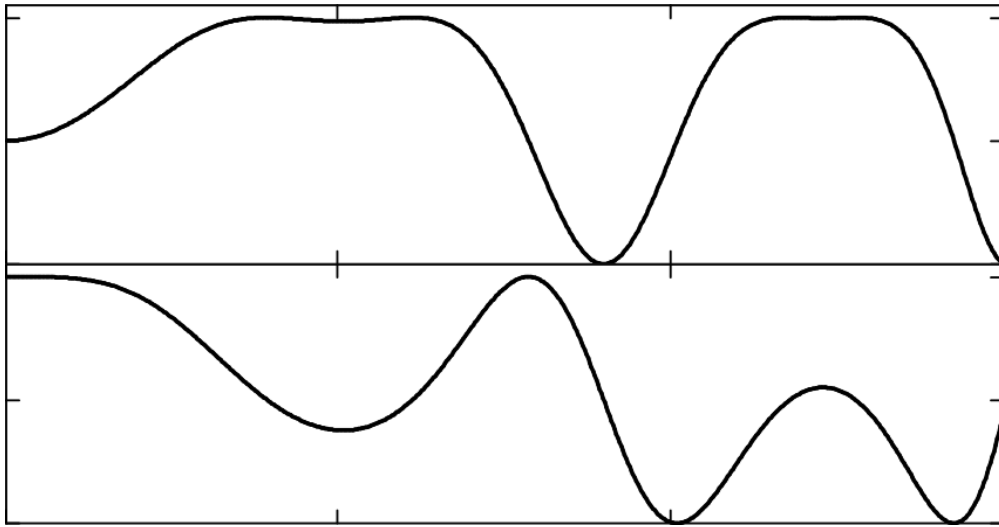


Рисунок 4.7 — Исходные сигналы

Прежде чем передать сигналы, их необходимо модулировать на разные несущие частоты. Модуляция осуществляется с помощью формулы (4.2).

$$\text{modulate}(sd, t, \omega) := sd(t) \cdot \sin(\omega \cdot t) \quad (4.2)$$

где sd — входной сигнал;

t — время;

ω — несущая частота.

Модулированные сигналы изображены на рис. 4.8.

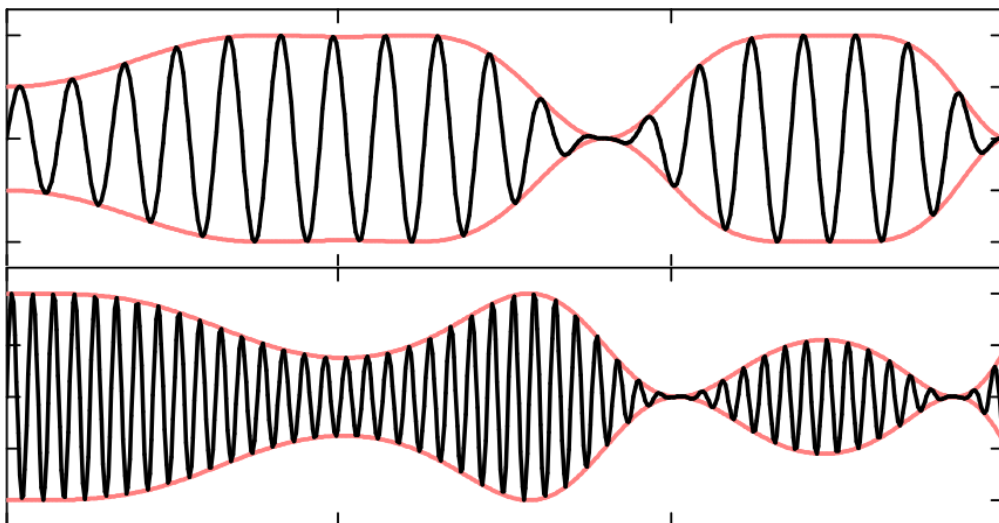


Рисунок 4.8 — Модулированные сигналы

Для наглядности дальнейших действий построим спектры получившихся сигналов. Они изображены на рис. 4.9.

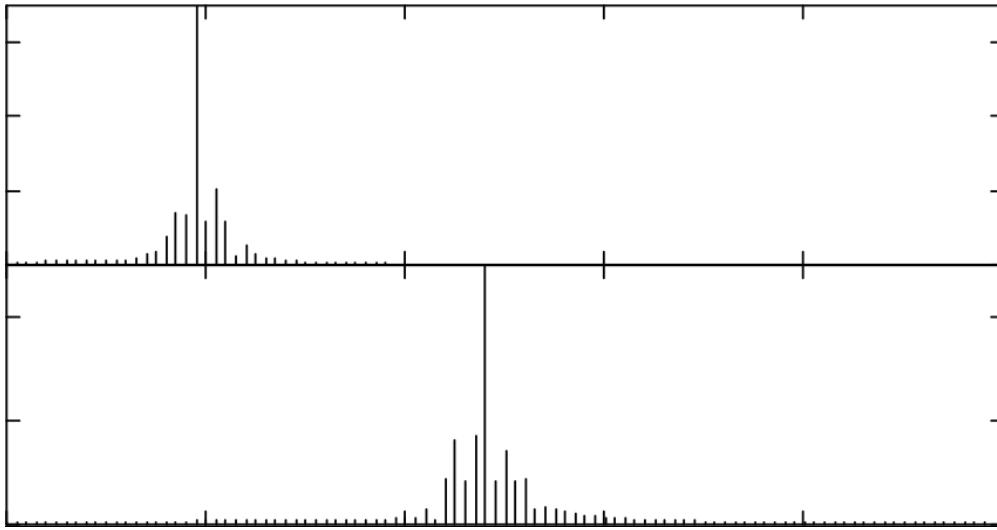


Рисунок 4.9 — Спектры модулированных сигналов

Как можно заметить спектр модулированного сигнала сильно ограничен по ширине, за счёт чего и происходит частотное объединение сигналов. По сути этот процесс представляет линейное сложение 2х сигналов во временной области (и как мы позже увидим, это верно и для частотной области). Объединение сигналов будем проводить с помощью функции (4.3).

$$\text{fcomposition}(t, s) := \begin{array}{|l} R \leftarrow 0 \\ \text{for } i \in 0.. \text{last}(s) \\ \quad | F \leftarrow s_i \\ \quad | R \leftarrow R + F(t) \\ R \end{array} \quad (4.3)$$

где s — массив из функций входных сигналов.

Получившийся сигнал, готовый к передаче изображён на рис. 4.10.

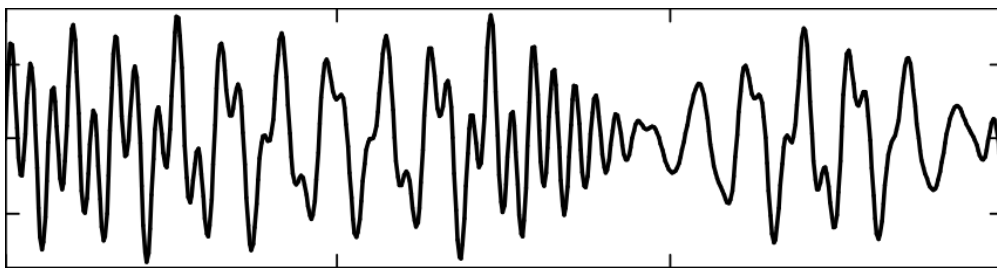


Рисунок 4.10 — Суммарный сигнал

На рис. 4.11 сверху изображён спектр, посчитанный для суммарного сигнала, снизу изображена линейная сумма спектров исходных сигналов. Как можно заметить между ними нет никаких различий.

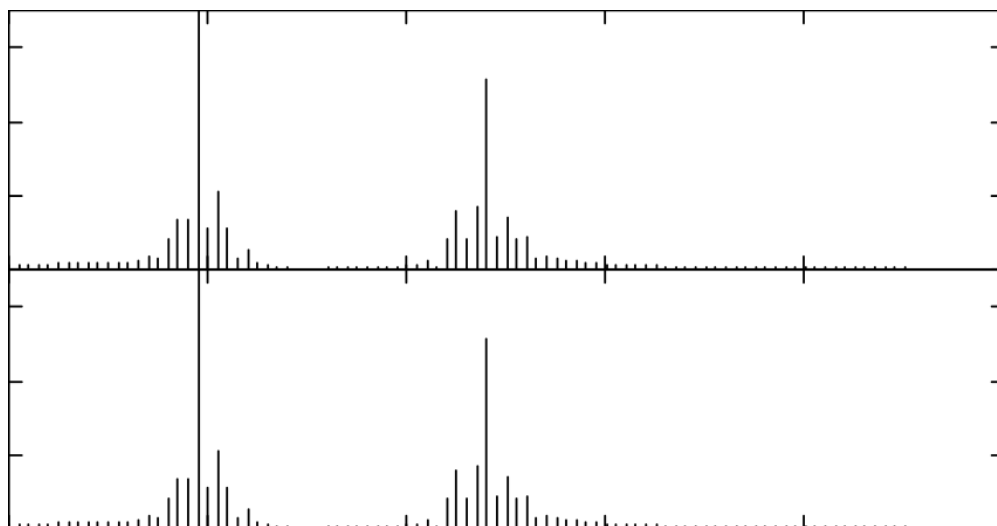


Рисунок 4.11 — Спектр суммарного сигнала

Разделение будем осуществлять с помощью функции (4.4). Она основывается на преобразовании Фурье и состоит из нескольких шагов: дискретизовать сигнал, посчитать его спектр, отфильтровать его часть, и с помощью обратного преобразования Фурье вернуться к сигналу во временной области. В реальности это решается намного проще, с помощью полосно-выделяющего фильтра.

$$\begin{aligned}
 \text{fdecomposition}(\text{sms}, T, \omega, \Delta\omega) := & \begin{array}{l} \text{for } i \in 0 \dots \frac{T}{\Delta} - 1 \\ \quad \text{sm}ds_i \leftarrow \text{sms}(i \cdot \Delta) \\ \quad \text{spectr} \leftarrow \text{cfft}(\text{sm}ds) \\ \quad K \leftarrow \text{last}(\text{spectr}) \\ \quad \text{for } i \in 0 \dots K \\ \quad \quad \left| \begin{array}{l} R_i \leftarrow \text{spectr}_i \text{ if } \left(\left| \frac{2 \cdot \pi \cdot i}{T} - \omega \right| \leq \Delta\omega \right) \vee \left[\left| \frac{2 \cdot \pi \cdot (K + 1 - i)}{T} - \omega \right| \leq \Delta\omega \right] \\ R_i \leftarrow 0 \text{ otherwise} \end{array} \right. \\ \quad R \leftarrow \text{icfft}(R) \\ \quad \text{for } i \in 0 \dots \text{last}(R) \\ \quad \quad R_i \leftarrow \text{Re}(R_i) \end{array}
 \end{aligned} \tag{4.4}$$

где sms — принятый сигнал.

Демодулированные сигналы изображены на рис. 4.12. Для наглядности приведены огибающие, соответствующие исходному сигналу. Как можно заметить существует небольшие отклонения по краям сигналов, что обуславливается погрешностью дискретных преобразований Фурье.

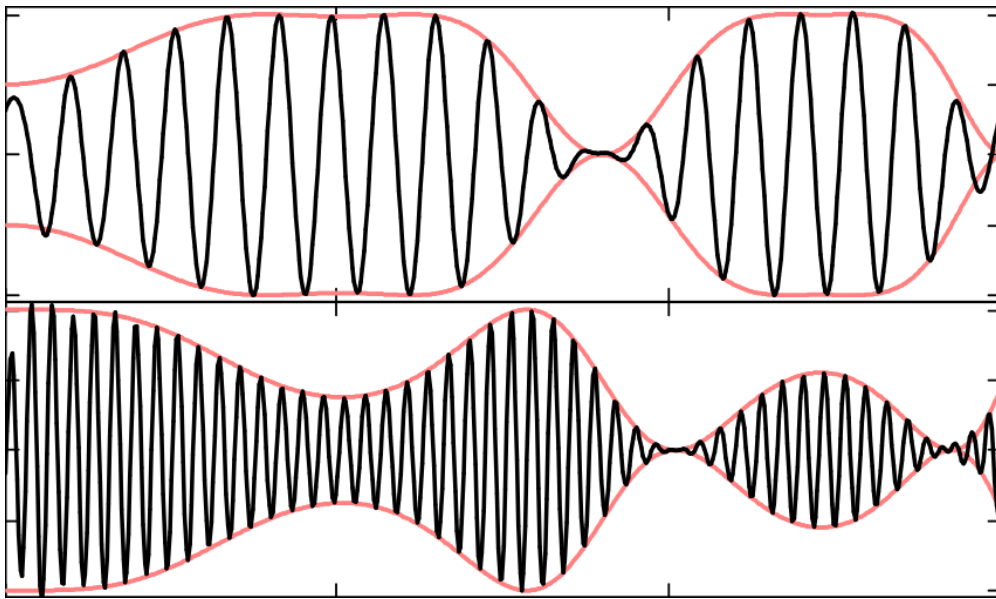


Рисунок 4.12 — Разделённые сигналы

Демодуляция сигнала подразумевает собой построение огибающей сигнала, которое основывается на добавлении ортогонального дополнения сигнала, полученного с помощью преобразования Гилберта (4.5).

$$\text{demodulate}(p) := \left| \begin{array}{l} h \leftarrow \text{hilbert}(p) \\ \text{for } i \in 0.. \frac{T}{\Delta} - 1 \\ \quad R_i \leftarrow \sqrt{(p_i)^2 + (h_i)^2} \\ R \end{array} \right. \quad (4.5)$$

Демодулированные сигналы изображены на рис. 4.13. Для наглядности исходный сигнал изображён на том же графике, менее насыщенным цветом. Как можно заметить принятый сигнал по краям отличается от исходного. В основном это объясняется погрешностями преобразования Фурье, и в реальной жизни для бесконечного сигнала не характерно.

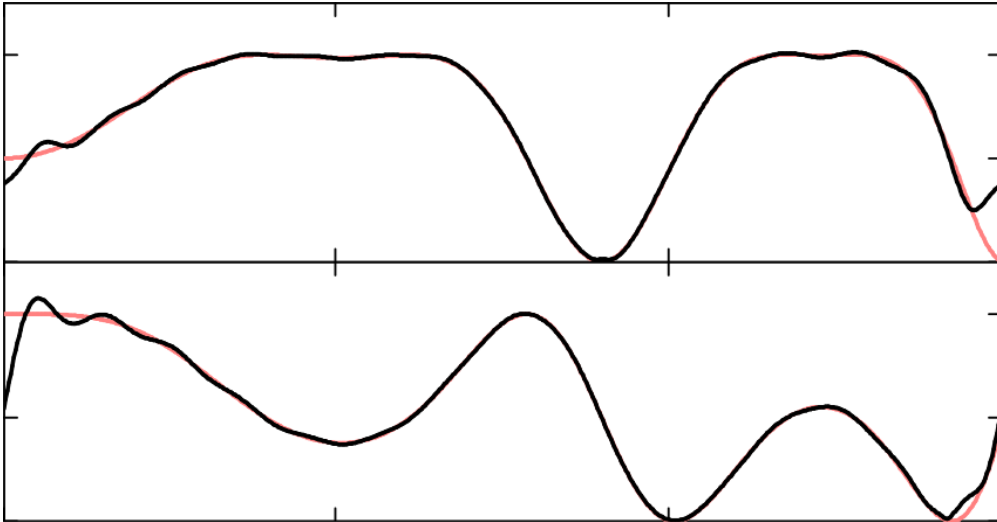


Рисунок 4.13 — Демодулированные сигналы

4.4 Принцип временного объединения и разделения каналов на практике

Для временного объединения каналов не требуется никакой предварительной подготовки. Объединение будет происходить с помощью функции (4.6).

$$\text{tcomposition}(t, \Delta T, s) := \left| \begin{array}{l} F \leftarrow s \\ \quad \text{mod} \left(\text{floor} \left(\frac{t}{\Delta T} \right), \text{length}(s) \right) \\ F(t) \end{array} \right. \quad (4.6)$$

Полученный сигнал изображён на рис. 4.14.

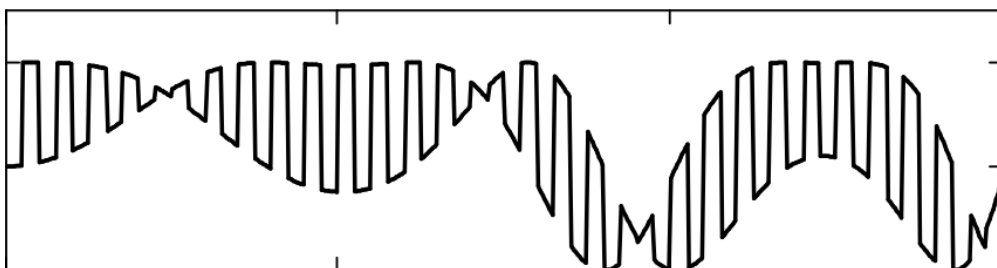


Рисунок 4.14 — Суммарный сигнал

Спектр полученного сигнала изображён на рис. 4.15. Как можно заметить полученный сигнал обладает достаточно широким спектром. При использовании канала передачи с ограниченной способностью передачи высокочастотных сигналов, то сигнал исказится (рис. 4.16).

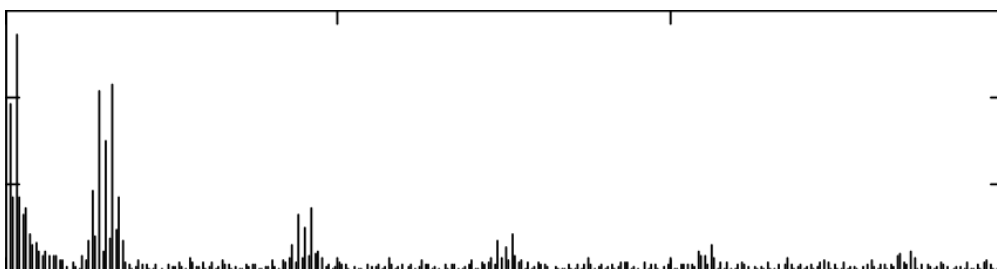


Рисунок 4.15 — Спектр сигнала

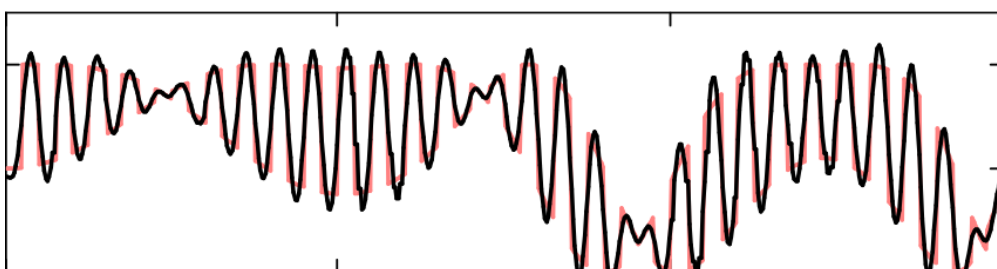


Рисунок 4.16 — Суммарный сигнал, без высокочастотных составляющих

Восстановим исходные сигналы с помощью функции (4.7). Недостающие участки сигналов восстановим с помощью линейной интерполяции.

$$\begin{array}{l}
 \text{tdecomposition}(s, T, \Delta T, c, n) := \left\{ \begin{array}{l}
 k \leftarrow 0 \\
 \text{for } i \in 0.. \frac{T}{\Delta} - 1 \\
 \quad \text{if } \text{mod}\left(\text{floor}\left(\frac{i \cdot \Delta}{\Delta T}\right), c\right) = n \\
 \quad \quad \left\{ \begin{array}{l}
 KX_k \leftarrow i \cdot \Delta \\
 KY_k \leftarrow s(i \cdot \Delta) \\
 k \leftarrow k + 1
 \end{array} \right. \\
 \text{for } i \in 0.. \frac{T}{\Delta} - 1 \\
 \quad R_i \leftarrow \text{interp}(KX, KY, i \cdot \Delta) \\
 R
 \end{array} \right.
 \end{array}
 \quad (4.7)$$

где c — количество объединённых сигналов;

n — номер извлекаемого сигнала.

Разделённые сигналы изображены на рис. 4.17. Как можно убедиться почти идеально совпадают с исходными сигналами.

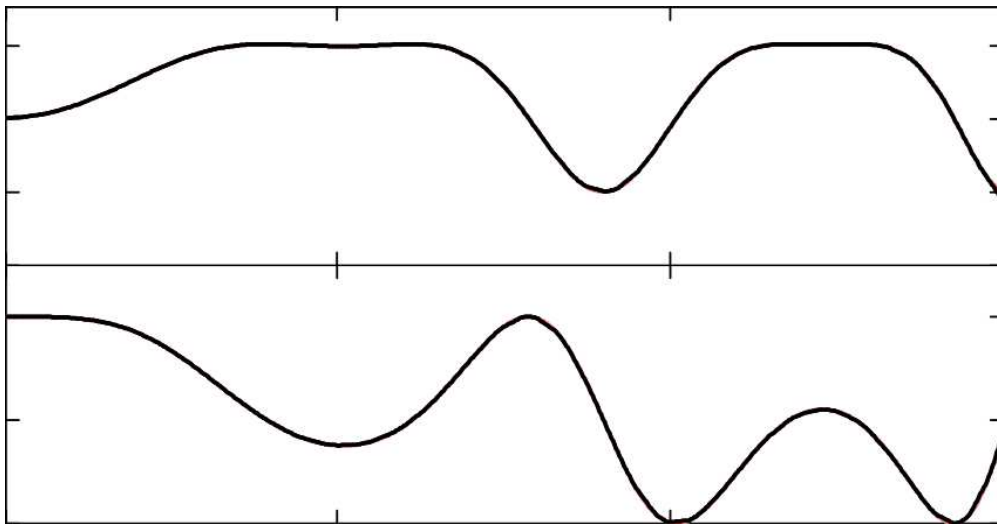


Рисунок 4.17 — Принятые сигналы

Так же стоит смоделировать сигналы, полученные разделением искажённого сигнала. Они изображены на рис. 4.18. Как можно убедиться полученные сигналы сильно искажены. Добиться более точного варианта

можно воспользовавшись полосно-выделяющим фильтром низких частот (рис. 4.19), но даже с ним сигнал отличается от исходного.

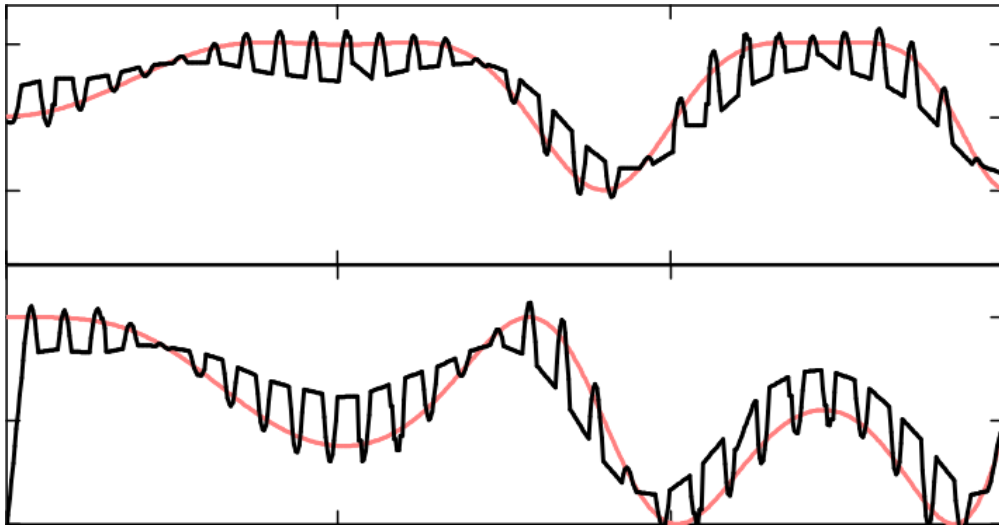


Рисунок 4.18 — Принятые сигналы

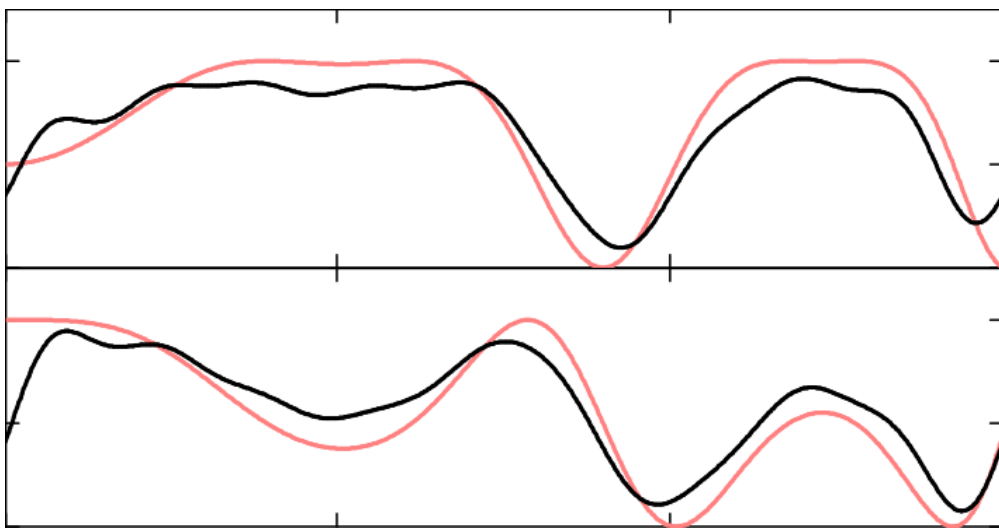


Рисунок 4.19 — Принятые сигналы, пропущенные через фильтр низких частот

4.5 Контрольные вопросы

- В каких случаях используются многоканальные системы связи?
- Какие существуют принципы объединения и разделения каналов?
- Поясните принцип работы систем с ЧРК.
- Поясните принцип работы системы с ВРК.
- Как должна выбираться длительность цикла в системах с ВРК?

- Какие способы объединения и разделения каналов нашли наибольшее распространение в системах связи?
- В чем состоит основное различие между многоканальными системами и системами множественного доступа?

4.6 Задание

1. Варианты исходных данных представлены в приложении В;
2. Повторить все графики, приведённые в данном методическом пособии для своего варианта (22 графика).
3. Изучить алгоритмы всех используемых функций.
4. Сделать выводы по каждому графику.
5. Оформить отчет.

4.7 Литература

1. Теория электрической связи: учебное пособие / К.К. Васильев, В.А. Глушков, А.В. Дормидонтов, А.Г. Нестеренко; под общ. ред. К.К. Васильева. — Ульяновск: УлГТУ, 2008. — 452 с.
2. Основы теории многоканальной передачи сообщений. URL: <http://kunegin.narod.ru/ref/lec/613.htm> (дата обращения 16.03.2012)
3. Огибающая сигнала. URL: <http://www.labviewportal.eu/viewtopic.php?f=81&t=1709> (дата обращения 16.03.2012)
4. Преобразование Гильберта. Аналитический сигнал. URL: <http://www.dsplib.ru/content/hilbert/hilbert.html> (дата обращения 19.03.2012)

Лабораторная работа №5. Линейное предсказание при анализе речевого сигнала

Линейное предсказание является одним из наиболее эффективных методов анализа речевого сигнала. Этот метод становится наиболее часто используемым при оценке основных параметров речевого сигнала, например,

период основного тона, спектр, функция площади речевого тракта, а также при сокращённом представлении речи с целью ее низкоскоростной передачи и экономного хранения. Основной принцип метода линейного предсказания состоит в том, что текущий отсчёт речевого сигнала можно аппроксимировать линейной комбинацией предшествующих отсчётов. Коэффициенты предсказания при этом определяются однозначно минимизацией среднего квадрата разности между отсчётами речевого сигнала и их предсказанными значениями (на конечном интервале). Коэффициенты предсказания — это весовые коэффициенты, используемые в линейной комбинации.

5.1 Модель предсказания

Модель, наиболее удобная для решения задач линейного предсказания представлена на рис. 5.1. В этом случае общий спектр, обусловленный излучением, речевым трактом и возбуждением, описывается с помощью линейной системы с переменными параметрами и передаточной функцией:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (5.1)$$

Эта система возбуждается импульсной последовательностью для вокализованных звуков речи и шумом для невокализованных. Таким образом, модель имеет следующие параметры: классификатор локализованных и нелокализованных звуков, период основного тона для вокализованных сегментов, коэффициент усиления G и коэффициенты цифрового фильтра. Все эти параметры во времени изменяются медленно.

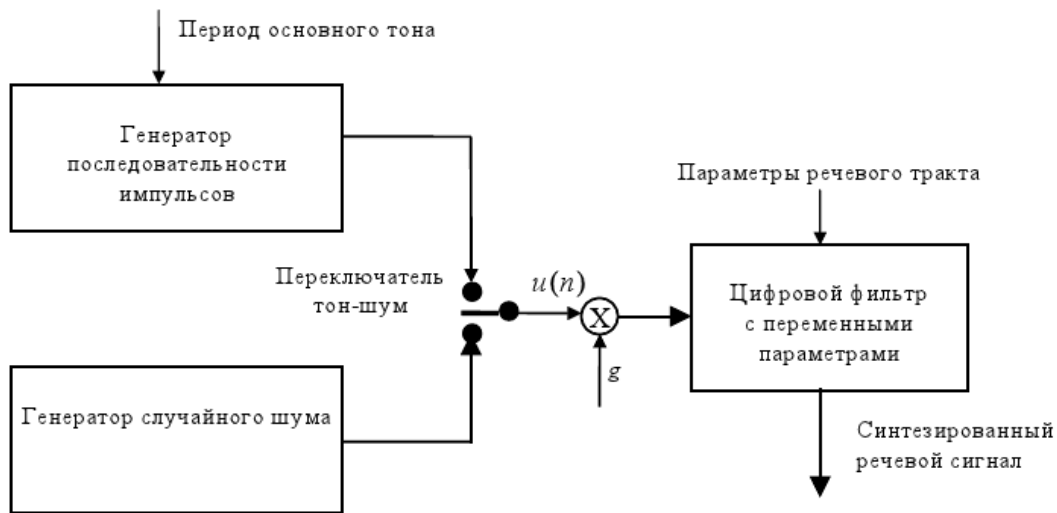


Рисунок 5.1 — Структурная схема упрощённой модели речеобразования

Для системы рис. 5.1 отсчёт речевого сигнала $s(n)$ связан с сигналом возбуждения $u(n)$ простым разностным уравнением:

$$s(n) = \sum_{k=1}^p a_k s(n-k) + Gu(n) \quad (5.2)$$

Линейный предсказатель с коэффициентами a_k определяется как система, на выходе которой имеем:

$$s(n) = \sum_{k=1}^p a_k s(n-k) + Gu(n) \quad (5.3)$$

Такие системы использовались для уменьшения дисперсии погрешности предсказания. Системная функция предсказателя представляет собой полином вида:

$$P(z) = \sum_{k=1}^p \alpha_k z^{-k} \quad (5.4)$$

Погрешность предсказания определяется как:

$$e(n) = s(n) - \bar{s}(n) = s(n) - \sum_{k=1}^p \alpha_k s(n-k) \quad (5.5)$$

Из (5.5) видно, что погрешность предсказания представляет собой сигнал на выходе системы с передаточной функцией:

$$A(z) = 1 - \sum_{k=1}^p \alpha_k z^k \quad (5.6)$$

Сравнение (5.2) и (5.5) показывает, что если сигнал точно удовлетворяет модели (5.2) и , то $e(n) = Gu(n)$. Таким образом, фильтр погрешности предсказания $A(z)$ является обратным фильтром для системы $H(z)$, соответствующей уравнению (5.1), т.е.

$$H(z) = \frac{G}{A(z)} \quad (5.7)$$

Основная задача анализа на основе линейного предсказания заключается в непосредственном определении параметров оп речевому сигналу с целью получения хороших оценок его спектральных свойств путём использования уравнения (5.7). Основным подходом является определение параметров предсказания таким образом, чтобы минимизировать дисперсию погрешности на коротком сегменте сигнала. При этом предполагается, что полученные параметры являются параметрами системной функции $H(z)$ в модели речеобразования.

Пусть , тогда $e(n) = Gu(n)$. Для вокализованной речи это означает, что $e(n)$ будет весьма мало почти все время. Поэтому в данном случае минимизация погрешности предсказания позволит получить требуемые коэффициенты.

Если сигнал формируется системой (5.2) с постоянными во времени параметрами, которая возбуждается либо единичным импульсом, либо белым шумом, то коэффициенты предсказания, найденные по критерию минимизации

среднего квадратичного значения погрешности (в каждый момент времени), совпадают с коэффициентами в (5.2).

Подобная минимизация приводит к линейной системе уравнений, решение которых приводит к получению параметров предсказания. Полученные параметры составляют основу для точного описания сигнала.

Кратковременная энергия погрешности предсказания:

$$E_n = \sum_m e_n^2(m) = \quad (5.8)$$

$$= \sum_m (s_n(m) - \bar{s}_n(m))^2 = \quad (5.9)$$

$$= \sum_m \left[s_n(m) - \sum_{k=1}^p \alpha_k s_n(m-k) \right]^2 \quad (5.10)$$

где $s_n(m)$ — сегмент речевого сигнала, выбранный в окрестности отсчёта n , т.е.

$$s_n(m) = s(m+n) \quad (5.11)$$

Параметры a_k можно получить, минимизируя E_n в (5.10) путём вычисления, $i = 1, 2, \dots, p$, что приводит к системе уравнений:

$$\sum_m s_n(m-i) s_n(m) = \sum_{k=1}^p \alpha_k \sum_m s_n(m-i) s_n(m-k), 1 \leq i \leq p \quad (5.12)$$

где $1 \leq i \leq p$ — значение $1 \leq i \leq p$, минимизирующее $1 \leq i \leq p$.

Если ввести определение:

$$\phi_n(i, k) = \sum_m s_n(m-i) s_n(m-k) \quad (5.13)$$

тогда (5.12) можно записать в более компактном виде:

$$\sum_{k=1}^p \alpha_k \phi_n(i, k) = \phi_n(i, 0), \quad 1 \leq i \leq p \quad (5.14)$$

Эта система из p уравнений с p неизвестными может быть решена достаточно эффективным способом для получения неизвестных коэффициентов предсказания, минимизирующих средний квадрат погрешности предсказания на сегменте $s_n(m)$. Используя (5.10) и (5.12), покажем, что средняя квадратическая погрешность предсказания имеет вид:

$$E_n = \sum_m s_n^2(m) - \sum_{k=1}^p \alpha_k \sum_m s_n(m) s_n(m-k) \quad (5.15)$$

и, используя (5.14), можно выразить E_n в виде:

$$E_n = \phi_n(0, 0) - \sum_{k=1}^p \alpha_k \phi_n(0, k) \quad (5.16)$$

Таким образом, общая погрешность предсказания состоит из двух слагаемых, одно из которых является постоянным, а другое — зависит от коэффициентов предсказания.

Для решения системы уравнений относительно коэффициентов предсказания следует вычислить величины $\phi_n(i, k)$, $1 \leq i \leq p$ и $0 \leq k \leq p$. Только после этого можно переходить к решению (5.14) и получению оценок α_k .

5.2 Автокорреляционный метод

Один из способов определения пределов в (5.8)-(5.10) и (5.12) основан на предположении, что сигнал равен нулю вне интервала $0 \leq m \leq N-1$. Это удобно записать в виде:

$$s_n(m) = s(m+n)\omega(m) \quad (5.17)$$

где $\omega(m)$ — окно конечной длительности, равною нулю вне интервала.

Значение этого предположения при решении вопроса о пределах суммирования в выражении для E_n можно рассматривать на примере соотношения (5.5). Очевидно, что если $s_n(m)$ отличен от нуля только на интервале $0 \leq m \leq N-1$, то соответствующая погрешность предсказания $e_n(m)$ для предсказания порядка p будет отлична от нуля на интервале $0 \leq m \leq N-1+p$. В этом случае E_n имеет вид:

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \quad (5.18)$$

Пределы при вычислении $\phi_n(i, k)$ в (5.13) совпадают с пределами (5.18). Но, поскольку $s_n(m)$ равно нулю вне интервала $0 \leq m \leq N-1$, легко показать, что:

$$\phi_n(i, k) = \sum_{m=0}^{N+p-1} s_n(m-i) s_n(m-k), \quad 1 \leq i \leq p, 0 \leq k \leq p \quad (5.19a)$$

Можно выразить в виде:

$$\phi_n(i, k) = \sum_{m=0}^{N-1-(i-k)} s_n(m) s_n(m+i-k), \quad 1 \leq i \leq p, 0 \leq k \leq p \quad (5.19b)$$

Легко видеть, что в данном случае $\phi_n(i, k)$ совпадает с кратковременной автокорреляционной функцией сигнала, вычисленной для $(i-k)$. Это означает, что:

$$\phi_n(i, k) = R_n(i-k) \quad (5.20)$$

где

$$R_n(k) = \sum_{m=0}^{N-1-k} s_n(m) s_n(m+k) \quad (5.21)$$

Поскольку $R_n(k)$ — чётная функция, то:

$$\phi_n(i, k) = R_n(|i - k|), i = 1, 2, \dots, p, k = 1, 2, \dots, p \quad (5.22)$$

Таким образом, (5.14) можно представить в виде:

$$\sum_{k=1}^p \alpha_k R_n(|i - k|) = R_n(i), 1 \leq i \leq p \quad (5.23)$$

Аналогично минимальный средний квадрат погрешности предсказания:

$$E_n = R_n(0) - \sum_{k=1}^p \alpha_k R_n(k) \quad (5.24)$$

Систему уравнений (5.23) можно записать в матричной форме:

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \cdots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \cdots & R_n(0) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \vdots \\ R_n(p) \end{bmatrix} \quad (5.25)$$

Матрица размером $p \times p$ является симметричной и такой, что элементы на любой диагонали равны между собой.

5.3 Ковариационный метод

Другой основной подход к определению сегмента речевого сигнала и пределов суммирования заключается в том, что фиксируется интервал, на котором вычисляется средний квадрат погрешности, и рассматривается влияние этого обстоятельства на вычисление $\phi_n(i, k)$, другими словами, если определить:

$$E_n = \sum_{m=0}^{N-1} e_n^2(m) \quad (5.26)$$

то $\phi_n(i, k)$ выражается формулой:

$$\phi_n(i,k) = \sum_{m=0}^{N-1} s_n(m) s_n(m-k), \quad 1 \leq i \leq p, 0 \leq k \leq p \quad (5.27)$$

Изменив индексы суммирования, (5.27) можно выразить в идее:

$$\phi_n(i,k) = \sum_{m=-i}^{N-i-1} s_n(m) s_n(m+i-k), \quad 1 \leq i \leq p, 0 \leq k \leq p \quad (5.28a)$$

или

$$\phi_n(i,k) = \sum_{m=-k}^{N-k-1} s_n(m) s_n(m+k-i), \quad 1 \leq i \leq p, 0 \leq k \leq p \quad (5.28b)$$

В (28) используется значение сигнала $s_n(m)$ вне интервала $0 \leq m \leq N-1$. Действительно, для вычисления $\phi_n(i,k)$ для всех требуемых значений i и k необходимо использовать значение $s_n(m)$ на интервале $-p \leq m \leq N-1$. Для того, чтобы это не противоречило пределам суммирования в (5.26), в данном случае используются необходимые значения сигнала без ограничения последовательности отсчётов окном конечной длительности, уменьшающимся к концам интервала, как это имело место в автокорреляционном методе. Таким образом, здесь используются отсчёты и вне интервала $0 \leq m \leq N-1$. Данный метод похож на вычисление модифицированной автокорреляционной функции, это приводит не к автокорреляционной, а к взаимокорреляционной функции между двумя очень сходными, но не одинаковыми сегментами речевого сигнала конечной длительности. Хотя различие между (5.28) и (5.19) сводится к небольшим вычислительным подробностям, система уравнений:

$$\sum_{k=1}^p \alpha_k \phi_n(i,k) = \phi_n(i,0), \quad i = 1, 2, \dots, p \quad (5.29a)$$

Обладает свойствами, которые значительно влияют на метод решения и свойства получаемого оптимального предсказателя. В матричной форме система уравнений имеет вид:

$$\begin{bmatrix} \phi_n(1,1) & \phi_n(1,2) & \phi_n(1,3) & \cdots & \phi_n(1,p) \\ \phi_n(2,1) & \phi_n(2,2) & \phi_n(2,2) & \cdots & \phi_n(2,p) \\ \phi_n(3,1) & \phi_n(3,2) & \phi_n(3,3) & \cdots & \phi_n(3,p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_n(p,1) & \phi_n(p,2) & \phi_n(p,2) & \cdots & \phi_n(p,p) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \phi_n(1,0) \\ \phi_n(2,0) \\ \phi_n(3,0) \\ \vdots \\ \phi_n(p,0) \end{bmatrix} \quad (5.29б)$$

В этом случае, поскольку $\phi_n(i,k)=\phi_n(k,i)$, матрица размером $p \times p$ является квазикорреляционной симметричной. Действительно, можно сказать, что диагональные элементы связаны соотношением:

$$\phi_n(i+1, k+1) = \phi_n(i, k) + s_n(-i-1)s_n(-k-1) = s_n(N-1-i)s_n(N-1-k) \quad (5.30)$$

Метод анализа, основанный на вычислении $\phi_n(i, k)$, называется ковариационным методом, поскольку матрица обладает свойствами ковариационной матрицы.

5.4 Вычисление коэффициента усиления модели

Коэффициент усиления G можно определить путём согласования энергии сигнала и линейно-предсказанных отсчётов. Постоянную G можно включить в сигнал возбуждения и ошибку предсказания (5.2) и (5.3). Сигнал возбуждения можно представить в виде:

$$Gu(n) = s(n) - \sum_{k=1}^p \alpha_k s(n-k) \quad (5.31a)$$

при этом погрешность предсказания будет представлена в виде:

$$e(n) = s(n) - \sum_{k=1}^p \alpha_k s(n-k) \quad (5.31б)$$

В случае, когда $a_k = \alpha_k$ т.е. коэффициенты предсказания совпадают с коэффициентами модели,

$$e(n) = Gu(n) \quad (5.32)$$

т.е. входной сигнал модели пропорционален погрешности предсказания с коэффициентом пропорциональности G .

Поскольку (5.32) является приближенным (т.е. справедливо лишь при равенстве параметров модели и предсказателя), в общем случае определить G непосредственно по погрешности предсказания невозможно. Целесообразно допустить, что энергия погрешности предсказания равна энергии сигнала возбуждения, т.е.

$$G^2 \sum_{m=0}^{N-1} u^2(m) = \sum_{m=0}^{N-1} e^2(m) = E_n \quad (5.33)$$

В этом случае для определения G по каким-либо параметрам, например по α_k и коэффициентам корреляции, необходимы некоторые предположения относительно $u(n)$. Для вокализованной речи естественно предположить, что $u(n) = \delta(n)$, т.е. возбуждение представляет единичный отсчёт в нулевой момент времени. Для этого необходимо, чтобы порядок предсказателя был достаточным для описания как передаточной функции речевого тракта, так и эффекта возбуждения. Для невокализованных сегментов целесообразно предположить, что $u(n)$ представляет собой белый шум с нулевым средним и единичной дисперсией.

Определим G на основе соотношения (5.33). Для вокализованных сегментов на входе имеется сигнал $G\delta(n)$. Если обозначить сигнал на выходе для этого случая через $h(n)$, получим соотношение:

$$h(n) = \sum_{k=1}^p \alpha_k h(n-k) + G\delta(n) \quad (5.34)$$

Можно показать, что автокорреляционная функция для $h(n)$, определяемая как

$$h(n) = \sum_{k=1}^p \alpha_k h(n-k) + G\delta(n) \quad (5.35)$$

удовлетворяет соотношениям:

$$\bar{R}(m) = \sum_{k=1}^p \alpha_k \bar{R}(m-k), m=1, 2, \dots, p \quad (5.36a)$$

и

$$\bar{R}(0) = \sum_{k=0}^p \alpha_k \bar{R}(k) + G^2 \quad (5.36b)$$

поскольку (5.36) совпадает с (5.32), следовательно,

$$\bar{R}(m) = R_n(m), 1 \leq m \leq p \quad (5.37)$$

учитывая, что полные энергии сигнала $[R(0)]$ и импульсной реакции $[R(0)]$ должны быть равны, можно использовать (5.24), (5.33) и (5.34), чтобы получить

$$G^2 = R_n(0) - \sum_{k=1}^p \alpha_k (R(k)) = E_n \quad (5.38)$$

Интересно, что требование равенства и (5.37) приводит к тому, что первые $p+1$ коэффициентов автокорреляции импульсной характеристики модели и сигнала должны совпадать.

В случае невокализированной речи корреляционная функция определяется статистическим усреднением. Предполагается, что сигнал возбуждения — белый шум с нулевым единичным и единичной дисперсией, т.е.

$$E[u(n)u(n-m)] = \delta(m) \quad (5.39)$$

Если возбудить систему случайным процессом $Gu(n)$ и обозначить процесс на выходе через $g(n)$, то

$$g(n) = \sum_{k=1}^p \alpha_k g(n-k) + Gu(n) \quad (5.40)$$

Если теперь $R(m)$ будет обозначать автокорреляционную функцию $g(n)$, то

$$\begin{aligned} \bar{R}(m) = E[g(n)g(n-m)] &= \sum_{k=1}^p \alpha_k E[g(n-k)g(n-m)] + \\ &+ E[Gu(n)g(n-m)] = \sum_{k=1}^p \alpha_k \bar{R}(m-k) \end{aligned} \quad (5.41)$$

Поскольку $E[g(n)g(n-m)] = 0, m > 0$, вследствие некоррелированности предшествующим $u(m)$. Для $m = 0$ получаем:

$$\bar{R}(0) = \sum_{k=1}^p \alpha_k \bar{R}(k) + GE[u(n)g(n)] = \sum_{k=1}^p \alpha_k \bar{R}(k) + G^2 \quad (5.42)$$

где $E[u(n)g(n)] = E[u(n)(Gu(n) + \text{члены, предшествующие } n)] = G^2$.

Поскольку энергия отклика на $Gu(n)$ должна совпадать с энергией сигнала, получаем:

$$\bar{R}(m) = R_n(m), 0 \leq m \leq p \quad (5.43)$$

или

$$G^2 = R_n(0) - \sum_{k=1}^p \alpha_k R_n(k) \quad (5.44)$$

т.е. имеем то же самое, что и в случае импульсного возбуждения.

5.5 Математическое моделирование

Исследования будем проводить над речевым сигналом «А». Для начала необходимо записать звук «А». Для этого подключим к компьютеру микрофон и откроем программу для аудиозаписи (для примера используется Audacity). Внешний вид программы приведен на рис. 5.2.

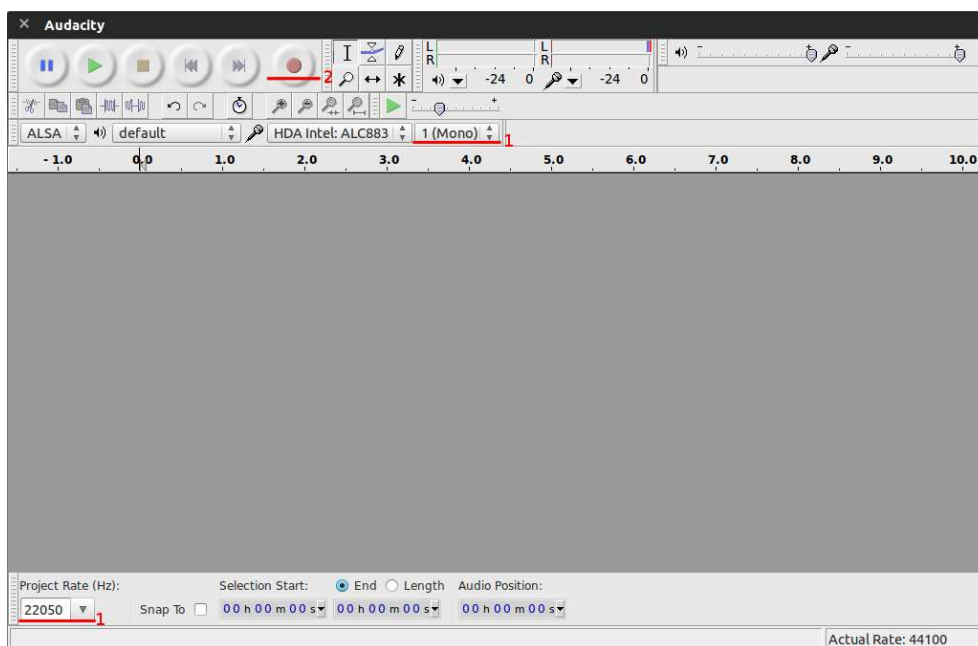


Рисунок 5.2 — Внешний вид программы Audacity

Запись звука надо проводить с параметрами указанными на рис. 5.2 (под цифрой 1, частота дискретизации 22050 Гц, 1 канал). Затем начните говорить непрерывно букву «А» и щелкните на кнопку 2. Записать надо около 4-6 секунд, стараясь сделать звук как можно более монотонным.

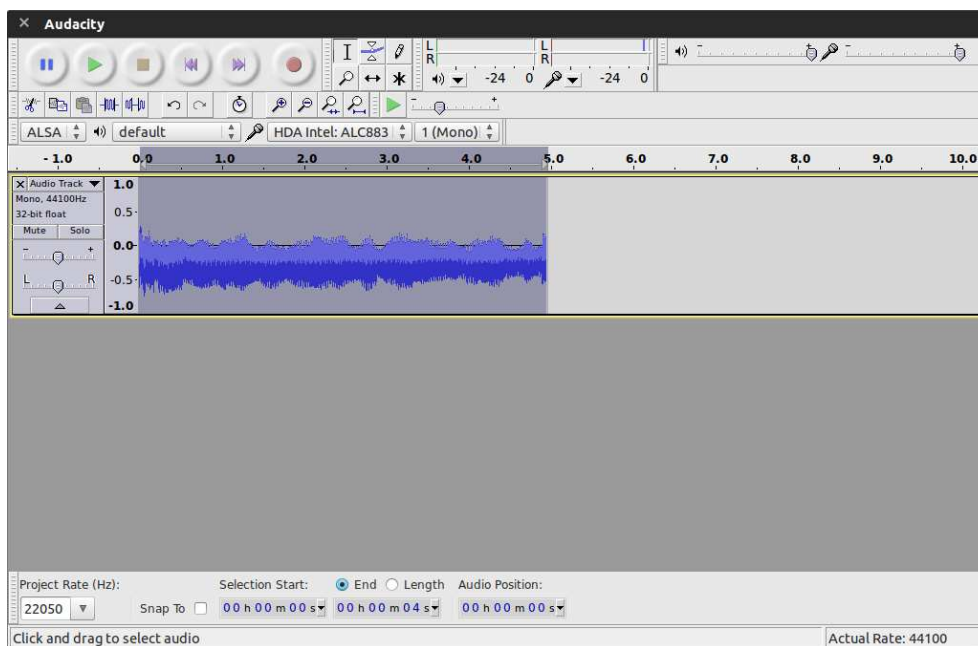


Рисунок 5.3 — Записанный звук

После записи звука (рис. 5.3) надо сделать 2 вещи. Во-первых, необходимо найти во сколько отсчётов укладывается 1 период «А». Во-вторых, его необходимо сохранить в формат WAV PCM signed 16 bit (рис. 5.5).

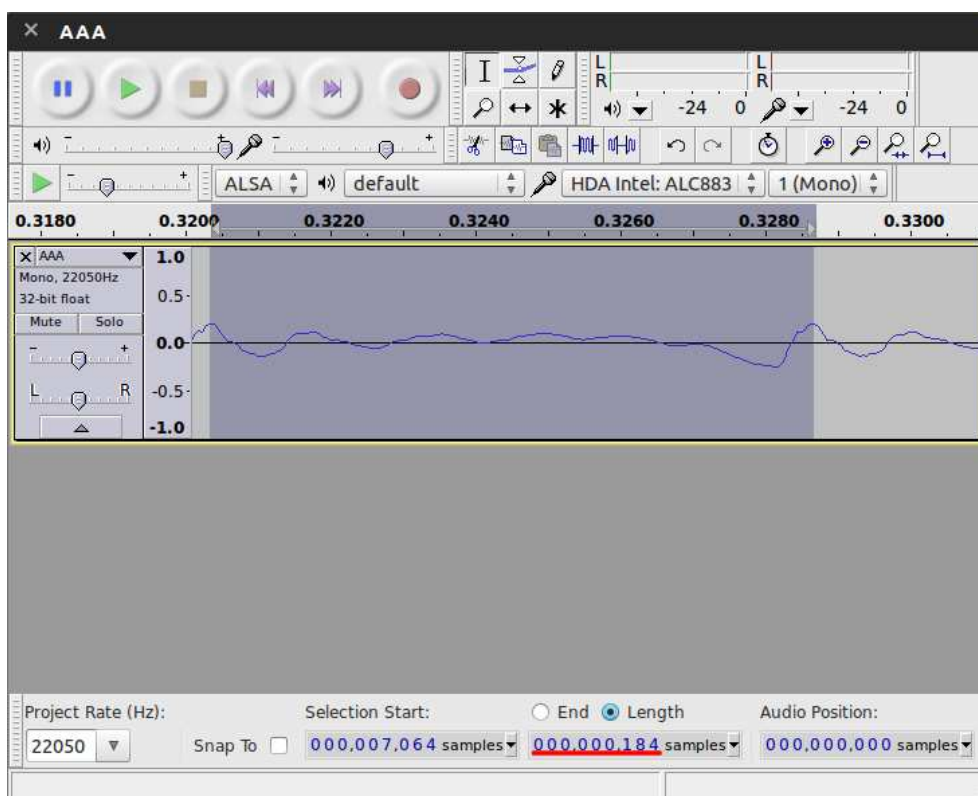


Рисунок 5.4 — Определение кол-ва отсчетов для одного периода

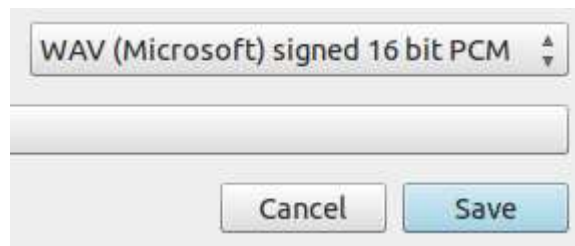


Рисунок 5.5 — Параметры сохранения

Следующим шагом будет запуск программы Analysys/Synthesys (Arm.exe). Откроем в нем наш записанный звук («Файл — Открыть звуковой файл *.wav»). На экране отобразится окно с АХ звука (рис. 5.6). При работе с данной программой следует избегать длинных имён директорий, и наличия пробелов в них.

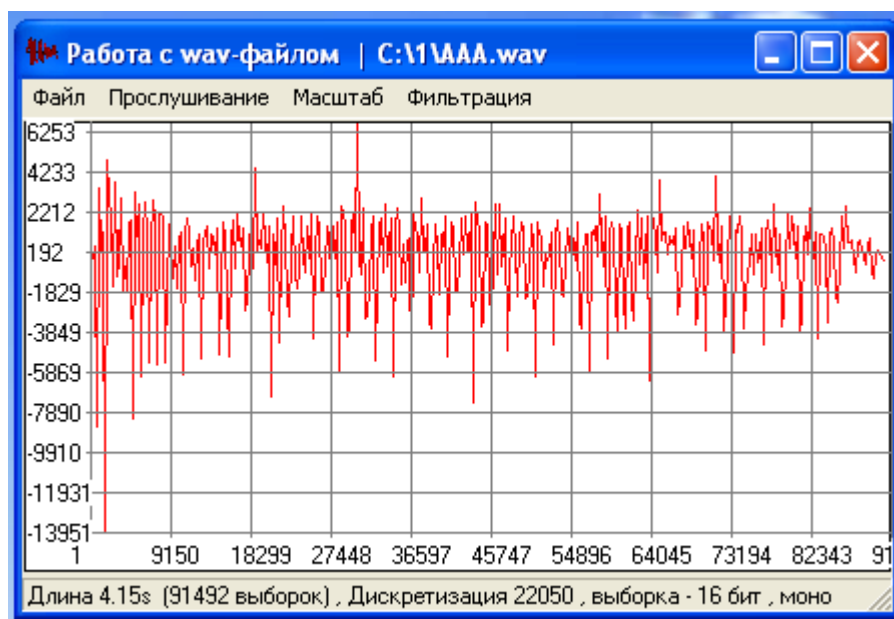


Рисунок 5.6 — АХ характеристика звука

Следующим шагом будет генерация полосового фильтра. Для этого надо в окне «Анализ речевых сигналов» выбрать «Фильтры-Синтез полосовых фильтров». После установить настройки указанные на рис. 5.7. и нажать кнопку «Ok».

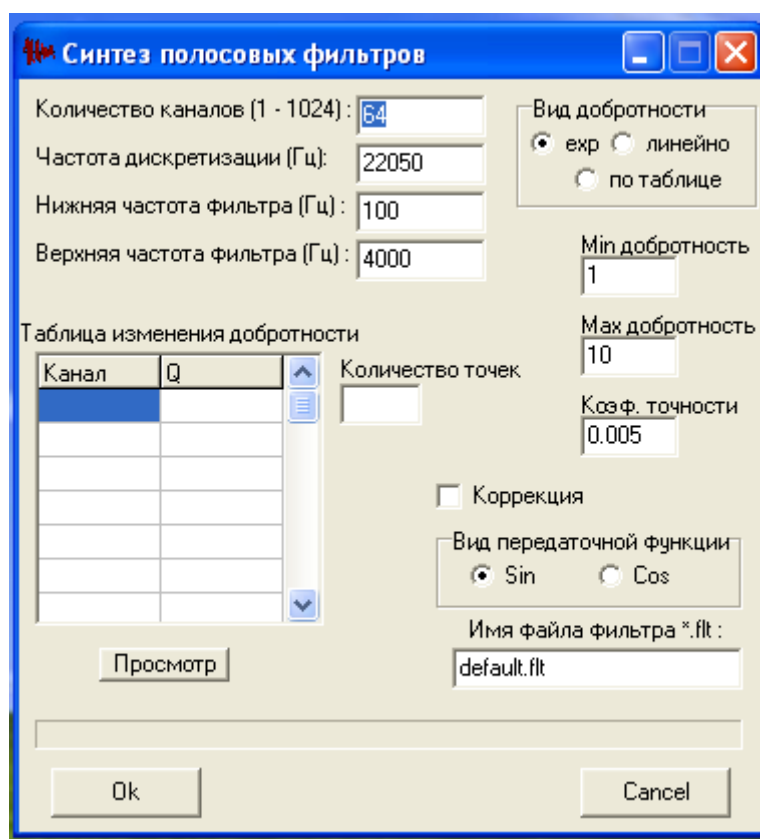


Рисунок 5.7 — Настройки полосового фильтра

После генерации фильтра необходимо профильтровать сигнал. Для этого в окне «Работа с wav-файлом» выбираем пункт «Фильтрация — Фильтровать сигнал», выбираем наш файл и фильтр, вводим выходное имя файла и щелкаем «Ок».

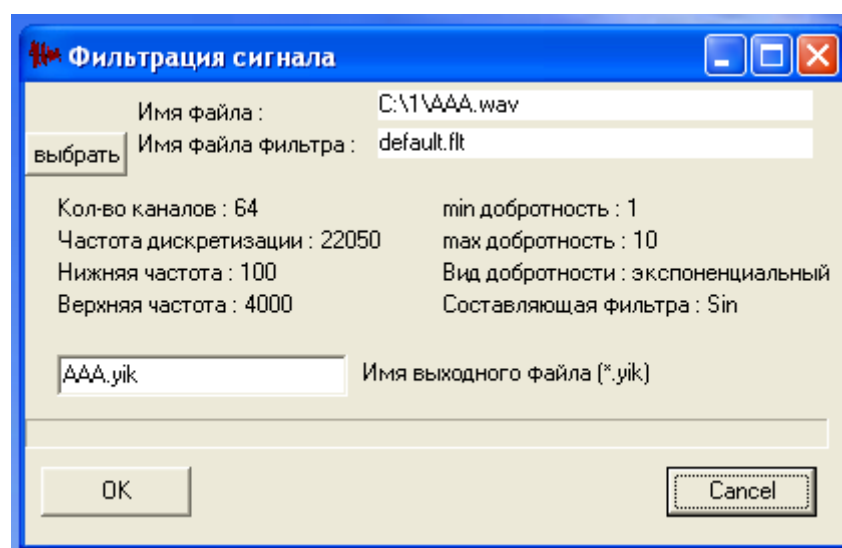


Рисунок 5.8 — Настройки фильтрации сигнала

После завершения фильтрации необходимо открыть получившийся результат с помощью окна «Анализ речевых сигналов», «Файл — Открыть обработанные файлы (*.yik, *.y2, *.dp)». Появится окно подобное окну на рис. 5.9.

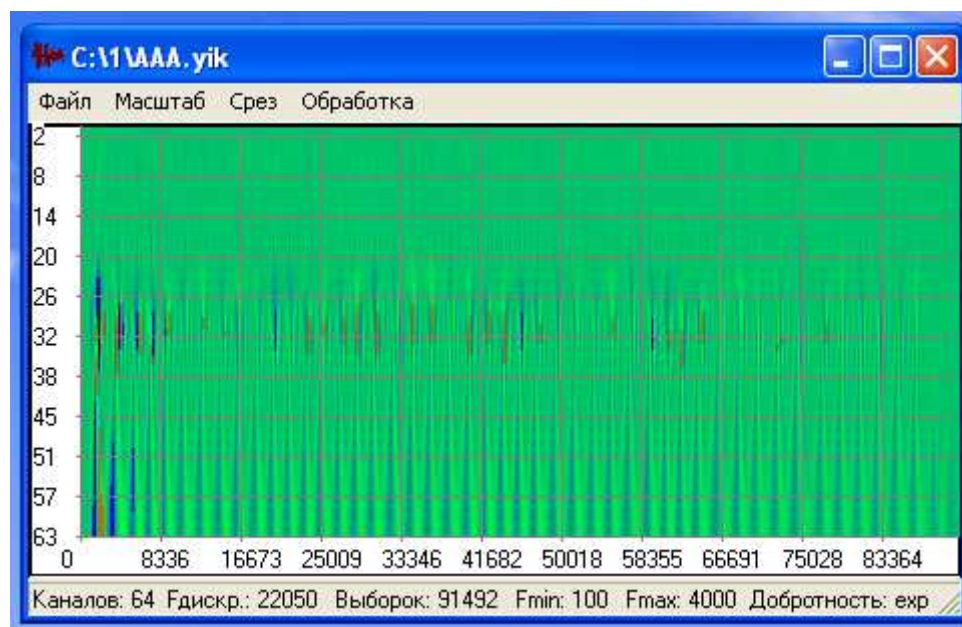


Рисунок 5.9 — Спектральная картина звука

Перед изучением стоит разобраться и поэкспериментировать с самой программой, например с масштабированием и переключением вида «Отображать частоту/канал» (в меню «Масштаб»). Из этого, например, можно узнать что за 32 каналом стоит частота в 614 Гц, и именно она соответствует звуку «А» приведённому в примере (рис. 5.10 и рис. 5.11).

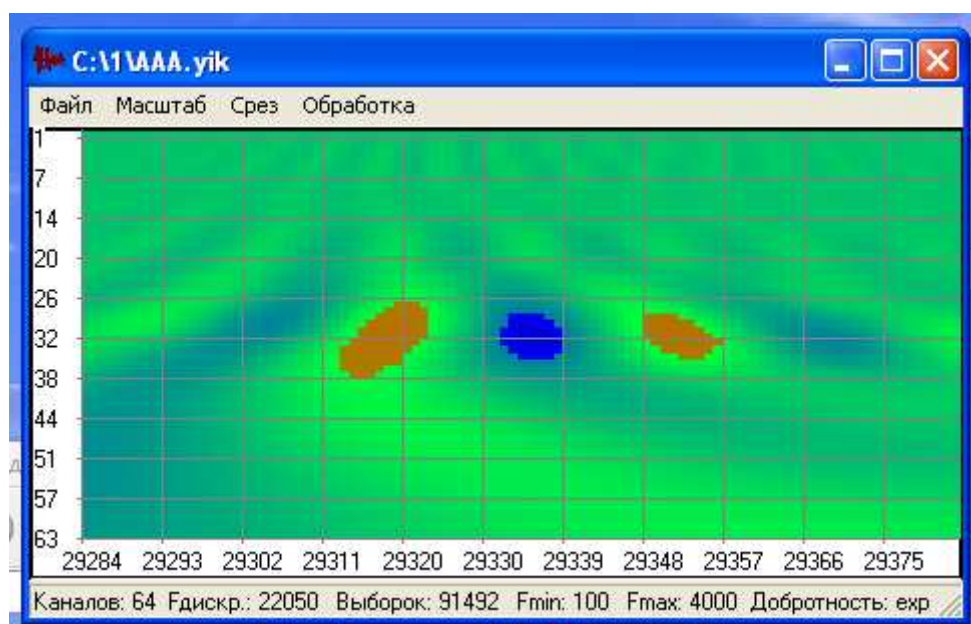


Рисунок 5.10 — Спектральная картина звука

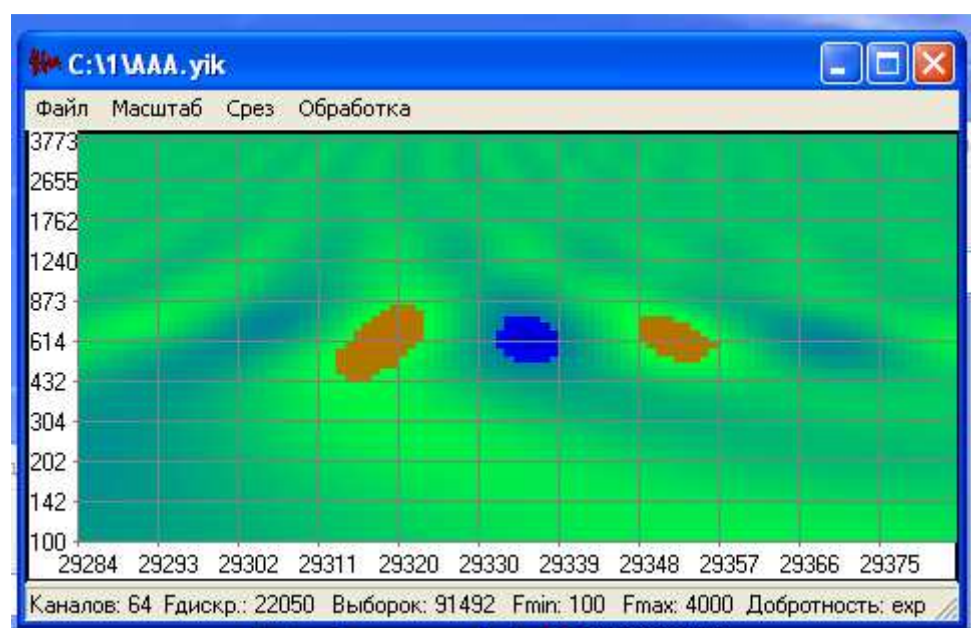


Рисунок 5.11 — Спектральная картина звука

Теперь для анализа нам потребуется взять выборки сигнала. Для этого щёлкнем «Обработка — Спектральный анализ». Откроется окно, изображённое на рис. 5.12.

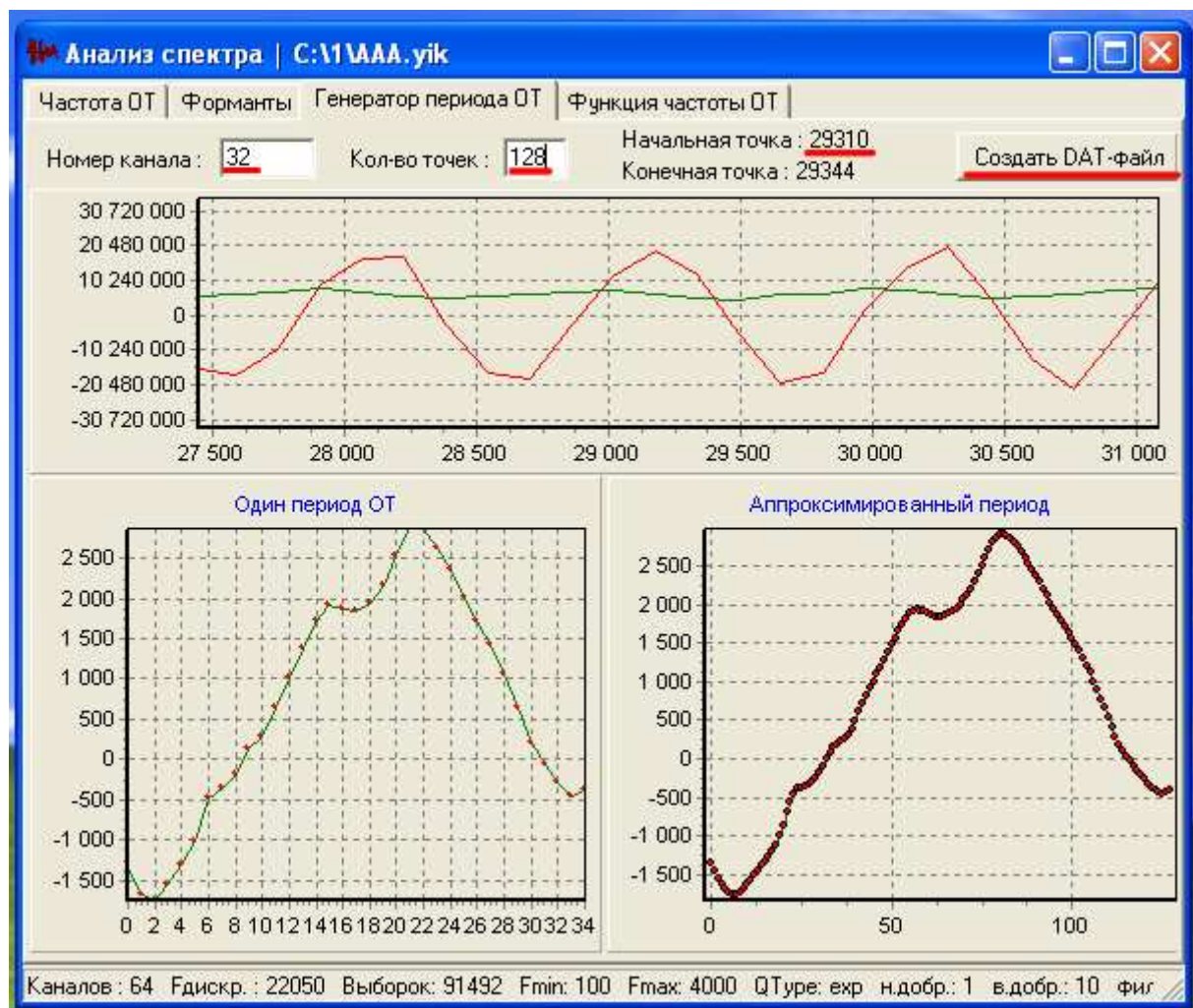


Рисунок 5.12 — Анализ спектра

В нем следует кол-во точек — 128, затем выбрать номер канала, соответствующий середине коричневой пятна (рис. 5.11), и начальную точку таким образом чтобы центр пятна попадал в центр выборки (рис. 5.11). После чего щёлкнуть «Создать DAT-файл» и сохранить его под именем «a1.dat». Прodelать эту операцию ещё 4 раза, и сохранить их под именами «a2.dat», «a3.dat», «a4.dat», «a5.dat». На этой работе с программой Analysys/Synthesys заканчивается, и дальнейшая работа будет проходить в MathCad.

Считываем файлы в Mathcad.

$$a := \begin{cases} \text{for } i \in 1..5 \\ R_i \leftarrow \text{READPRN}(\text{concat}("a", \text{num2str}(i), ".dat"))^T \\ R \end{cases}$$

И строим их графики:

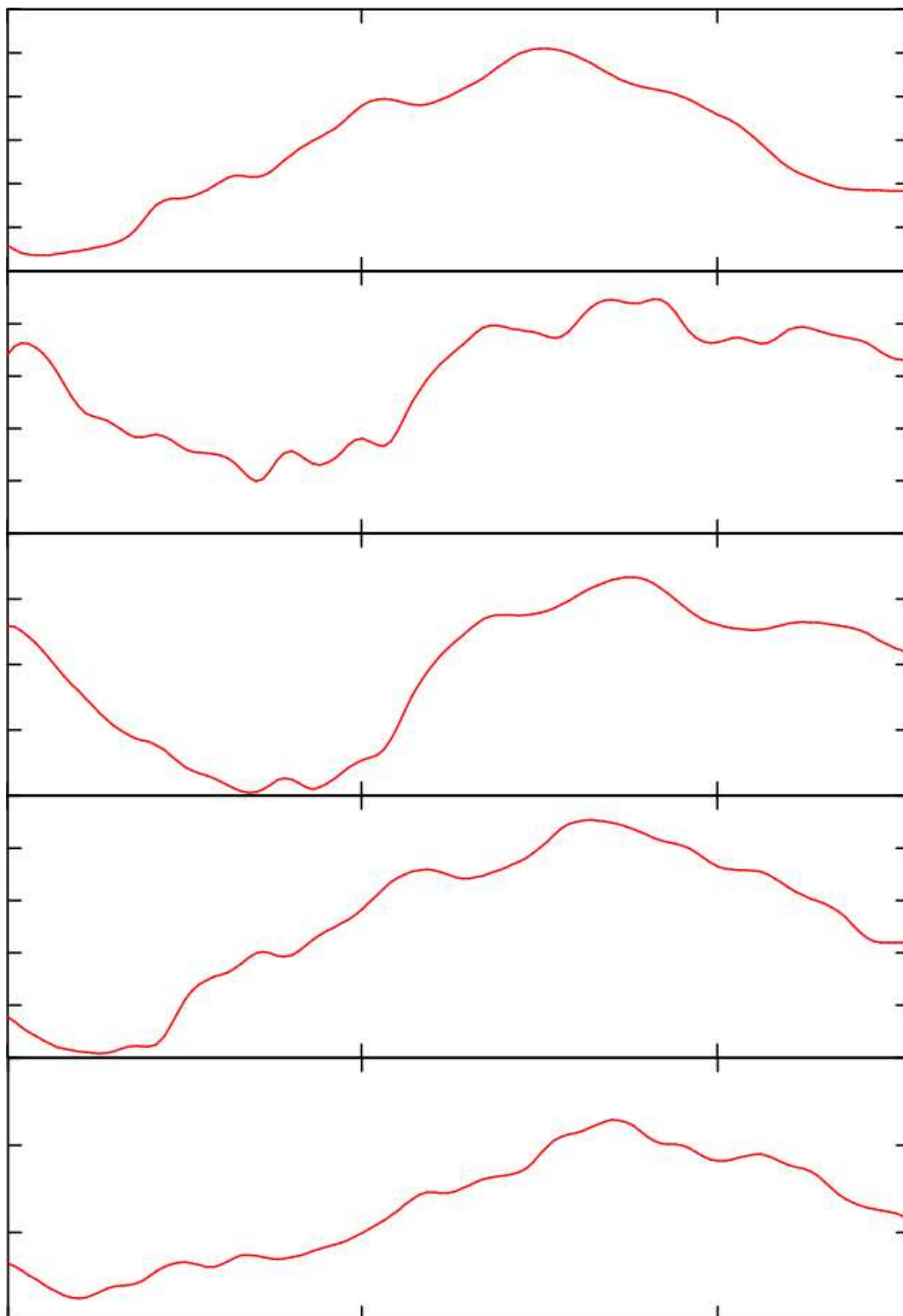


Рисунок 5.13 — Графики выборов буквы "А"

Посчитаем матрицу ковариации (рис. 5.14).

	0	1	2	3	4
0	$2.325 \cdot 10^6$	$2.456 \cdot 10^6$	$2.519 \cdot 10^6$	$2.52 \cdot 10^6$	$2.48 \cdot 10^6$
1	$2.456 \cdot 10^6$	$2.612 \cdot 10^6$	$2.688 \cdot 10^6$	$2.692 \cdot 10^6$	$2.647 \cdot 10^6$
2	$2.519 \cdot 10^6$	$2.688 \cdot 10^6$	$2.773 \cdot 10^6$	$2.783 \cdot 10^6$	$2.741 \cdot 10^6$
3	$2.52 \cdot 10^6$	$2.692 \cdot 10^6$	$2.783 \cdot 10^6$	$2.801 \cdot 10^6$	$2.77 \cdot 10^6$
4	$2.48 \cdot 10^6$	$2.647 \cdot 10^6$	$2.741 \cdot 10^6$	$2.77 \cdot 10^6$	$2.755 \cdot 10^6$
5	$2.411 \cdot 10^6$	$2.568 \cdot 10^6$	$2.664 \cdot 10^6$	$2.705 \cdot 10^6$	$2.708 \cdot 10^6$
6	$2.308 \cdot 10^6$	$2.448 \cdot 10^6$	$2.543 \cdot 10^6$	$2.596 \cdot 10^6$	$2.62 \cdot 10^6$
7	$2.18 \cdot 10^6$	$2.298 \cdot 10^6$	$2.389 \cdot 10^6$	$2.455 \cdot 10^6$	$2.503 \cdot 10^6$
8	$2.038 \cdot 10^6$	$2.129 \cdot 10^6$	$2.214 \cdot 10^6$	$2.293 \cdot 10^6$	$2.365 \cdot 10^6$
9	$1.893 \cdot 10^6$	$1.957 \cdot 10^6$	$2.036 \cdot 10^6$	$2.126 \cdot 10^6$	$2.222 \cdot 10^6$
10	$1.755 \cdot 10^6$	$1.797 \cdot 10^6$	$1.87 \cdot 10^6$	$1.971 \cdot 10^6$	$2.087 \cdot 10^6$
11	$1.637 \cdot 10^6$	$1.663 \cdot 10^6$	$1.734 \cdot 10^6$	$1.844 \cdot 10^6$	$1.978 \cdot 10^6$
12	$1.539 \cdot 10^6$	$1.558 \cdot 10^6$	$1.63 \cdot 10^6$	$1.749 \cdot 10^6$	$1.896 \cdot 10^6$
13	$1.448 \cdot 10^6$	$1.464 \cdot 10^6$	$1.538 \cdot 10^6$	$1.665 \cdot 10^6$	$1.823 \cdot 10^6$
14	$1.351 \cdot 10^6$	$1.362 \cdot 10^6$	$1.438 \cdot 10^6$	$1.571 \cdot 10^6$	$1.739 \cdot 10^6$
15	$1.235 \cdot 10^6$	$1.236 \cdot 10^6$	$1.31 \cdot 10^6$	$1.45 \cdot 10^6$...

Рисунок 5.14 — Матрица ковариации

И получаем наши выходные сигналы, основанные на линейном предсказании (рис. 5.15).

$$su_n := cov \cdot a_n$$

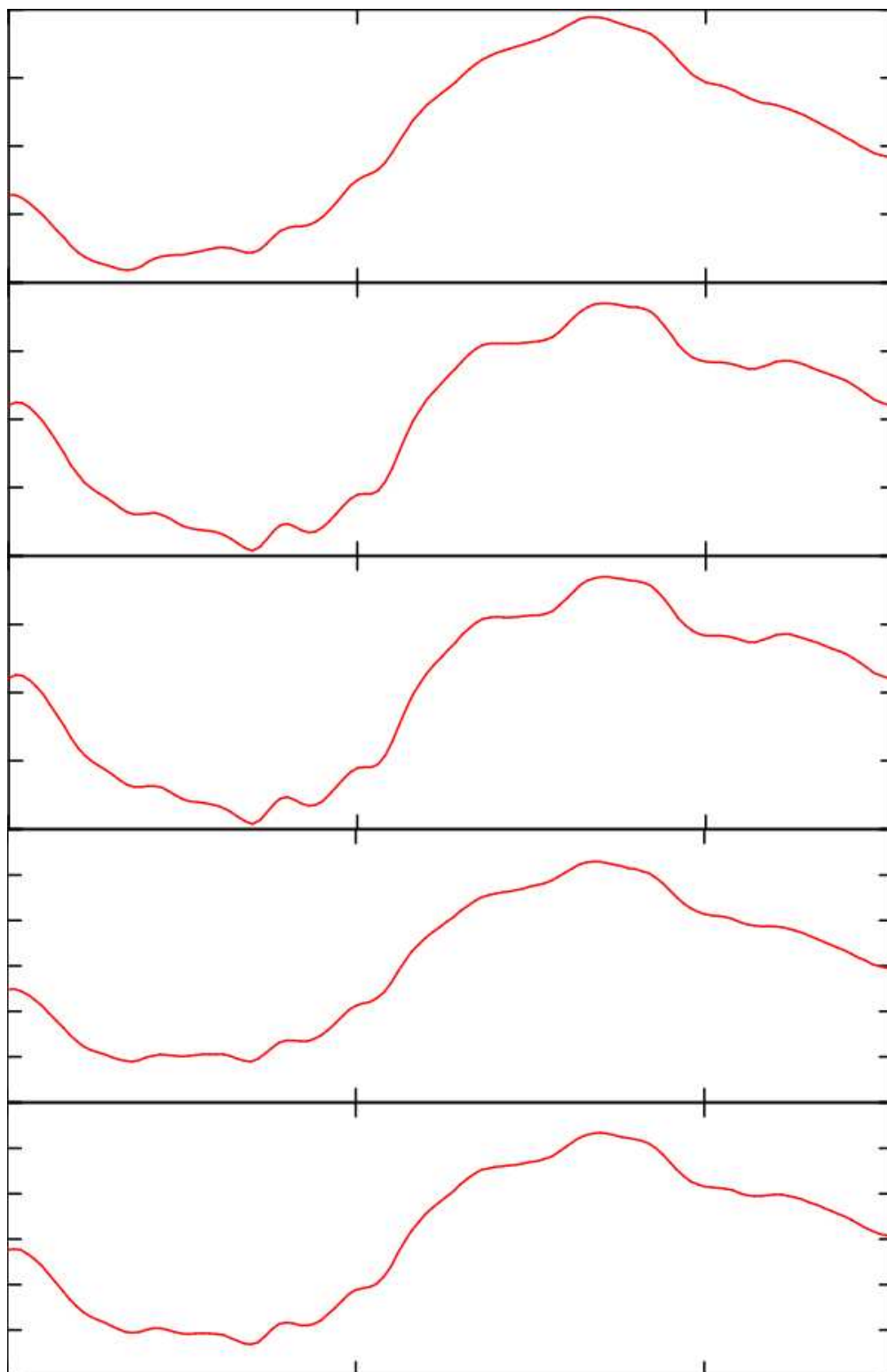


Рисунок 5.15 — Выходной сигнал

Последней задачей будет собрать последовательность из полученных выборок и записать это в файл. Однако эта задача немного оттягивается тем обстоятельством что сигнал необходимо масштабировать по вертикали (из-за преобразований) и по горизонтали (из за проецировании выборки на 128 точек).

Для масштабирования по вертикали возьмём минимум и максимум выборки, а для масштабирования по горизонтали воспользуемся значением полученным из рис. 5.4.

$$sz := \left\lceil \frac{\max(su_1) - \min(su_1)}{2} \right\rceil$$

$$sl(x) := \frac{\text{linterp}\left(lu, su_1, \frac{\text{mod}(x, p) \cdot 128}{p}\right)}{sz} \cdot 2^{14}$$

$$wav_x := sl(x)$$

$$\text{WRITEWAV}("out.wav", 22050, 16) := wav$$

В результате появится файл «out.wav», при проигрывании которого вы должны услышать свой голос, хоть и излишне монотонный, но узнаваемый.

5.6 Задание

1. Самостоятельно выполнить действия, описанные в разделе 5.5;
2. Продемонстрировать результат преподавателю;
3. Подготовить отчет;

5.7 Литература

1. Цифровая обработка речевых сигналов: Рабинер Л.Р., Шафер Р.В. –М.: Радио и связь, 1981г.
2. Системы анализа и синтеза речевого сигнала с линейным предсказанием: Кортаев Г.А. –Зарубежная радиоэлектроника, 1976г.
3. Линейное предсказание речи: Маквел Дж.Д., Грей А.Х. –М.: Связь, 1980г.

Лабораторная работа №6. Введение в IP-сети

Целью работы является изучение работы IP-сетей, протоколов IP, ICMP, и нескольких стандартных утилит работы с сетью.

6.1 Протокол IP

Internet Protocol или IP (англ. internet protocol — межсетевой протокол) — маршрутизируемый сетевой протокол, протокол сетевого уровня семейства TCP/IP.

Протокол IP используется для негарантированной доставки данных, разделяемых на так называемые пакеты от одного узла сети к другому. Это означает, что на уровне этого протокола (третий уровень сетевой модели OSI) не даётся гарантий надёжной доставки пакета до адресата. В частности, пакеты могут прийти не в том порядке, в котором были отправлены, продублироваться (когда приходят две копии одного пакета; в реальности это бывает крайне редко), оказаться повреждёнными (обычно повреждённые пакеты уничтожаются) или не прибыть вовсе. Гарантию безошибочной доставки пакетов дают протоколы более высокого (транспортного уровня) сетевой модели OSI — например, TCP — которые используют IP в качестве транспорта.

6.1.1 Протокол IP версии 4

В современной сети Интернет используется IP четвёртой версии, также известный как IPv4. В протоколе IP этой версии каждому узлу сети ставится в соответствие IP-адрес длиной 4 октета (4 байта). При этом компьютеры в подсетях объединяются общими начальными битами адреса. Количество этих бит, общее для данной подсети, называется маской подсети (ранее использовалось деление пространства адресов по классам — А, В, С; класс сети определялся диапазоном значений старшего октета и определял число адресуемых узлов в данной сети, сейчас используется бесклассовая адресация).

6.1.2 Протокол IP версии 6

В настоящее время вводится в эксплуатацию шестая версия протокола — IPv6, которая позволяет адресовать значительно большее количество узлов, чем IPv4. Эта версия отличается повышенной разрядностью адреса, встроенной возможностью шифрования и некоторыми другими особенностями. Переход с

IPv4 на IPv6 связан с трудоёмкой работой операторов связи и производителей программного обеспечения и не может быть выполнен одномоментно. На середину 2010 года в Интернете присутствовало более 3000 сетей, работающих по протоколу IPv6. Для сравнения, на то же время в адресном пространстве IPv4 присутствовало более 320 тысяч сетей, но в IPv6 сети гораздо более крупные, нежели в IPv4.

6.2 IP-Пакет

IP-пакет — форматированный блок информации, передаваемый по вычислительной сети. Соединения вычислительных сетей, которые не поддерживают пакеты, такие как традиционные соединения типа «точка-точка» в телекоммуникациях, просто передают данные в виде последовательности байтов, символов или битов. При использовании пакетного форматирования сеть может передавать длинные сообщения более надёжно и эффективно.

6.2.1 IP-Пакет версии 4 (IPv4)

0								1								2								3											
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7				
Версия				IHL				Тип обслуживания								Длина пакета																			
Идентификатор																Флаги				Смещение фрагмента															
Время жизни (TTL)								Протокол								Контрольная сумма заголовка																			
IP-адрес отправителя (32 бита)																																			
IP-адрес получателя (32 бита)																																			
Параметры (до 320 бит)																Данные (до 65535 байт минус заголовок)																			

- Версия — для IPv4 значение поля должно быть равно 4.
- IHL — длина заголовка IP-пакета в 32-битных словах (dword). Именно это поле указывает на начало блока данных в пакете. Минимальное корректное значение для этого поля равно 5.
- Идентификатор — значение, назначаемое отправителем пакета и предназначенное для определения корректной последовательности

фрагментов при сборке датаграммы. Для фрагментированного пакета все фрагменты имеют одинаковый идентификатор.

- 3 бита флагов. Первый бит должен быть всегда равен нулю, второй бит DF (don't fragment) определяет возможность фрагментации пакета и третий бит MF (more fragments) показывает, не является ли этот пакет последним в цепочке пакетов.
- Смещение фрагмента — значение, определяющее позицию фрагмента в потоке данных.
- Протокол — идентификатор интернет-протокола следующего уровня (см. IANA protocol numbers и RFC 1700). В IPv6 называется «Next Header».
- Контрольная сумма заголовка — вычисляется с использованием операций сложения 16-разрядных слов заголовка по модулю 1. Сама контрольная сумма является дополнением по модулю один полученного результата сложения.

6.2.2 IP-Пакет версии 6 (IPv6)

Позиция в октетах		0				1				2				3																			
	Позиция в битах	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Версия		Класс трафика				Метка потока																									
4	32	Длина полезной нагрузки												След. заголовок				Число переходов															
8	64	IP-адрес отправителя																															
12	96																																
16	128																																
20	160																																
24	192	IP-адрес получателя																															
28	224																																
32	256																																
36	288																																

- Версия — для IPv6 значение поля должно быть равно 6.
- Класс трафика — определяет приоритет трафика (QoS, класс обслуживания).
- Метка потока — уникальное число, одинаковое для однородного потока пакетов.

- Длина полезной нагрузки — длина данных (заголовок IP-пакета не учитывается).
- Следующий заголовок — задаёт тип расширенного заголовка (англ. IPv6 extension), который идёт следующим. В последнем расширенном заголовке поле Next header задаёт тип транспортного протокола (TCP, UDP и т. д.) и определяет следующий инкапсулированный уровень.
- Число переходов — максимальное число маршрутизаторов, которые может пройти пакет. При прохождении маршрутизатора это значение уменьшается на единицу и по достижении нуля пакет отбрасывается.

6.3 Протокол ICMP

ICMP (англ. Internet Control Message Protocol — межсетевой протокол управляющих сообщений) — сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или хост, или маршрутизатор не отвечают. Также на ICMP возлагаются некоторые сервисные функции.

6.3.1 Технические подробности

Протокол ICMP описан в RFC 792 (с дополнениями в RFC 950) и является стандартом Интернета (входит в стандарт STD 5 вместе с IP). Хотя формально ICMP использует IP (ICMP-пакеты инкапсулируются в IP пакеты), он является неотъемлемой частью IP и обязателен при реализации стека TCP/IP. Текущая версия ICMP для IPv4 называется ICMPv4. В IPv6 существует аналогичный протокол ICMPv6.

ICMP-сообщение строится из IP-пакетов, сгенерировавших ICMP-ответ. IP инкапсулирует соответствующее ICMP-сообщение с новым заголовком IP (чтобы отправить ICMP-сообщение обратно отправителю) и передает полученные пакеты дальше.

Например, каждая машина (такая, как маршрутизатор), которая перенаправляет IP-пакеты, уменьшает Time to live (TTL) поля заголовка IP на единицу, если TTL достигает 0, ICMP-сообщение о превышении TTL отправляется на источник пакета.

Каждое ICMP-сообщение инкапсулируется непосредственно в пределах одного IP-пакета, и, таким образом, как и UDP, ICMP является ненадежным (надежным является TCP).

ICMP основан на протоколе IP. Его цели отличны от целей транспортных протоколов, таких как TCP и UDP: он, как правило, не используется для передачи и приема данных между конечными системами. ICMP не используется непосредственно в приложениях пользователей сети (исключение составляют инструменты Ping и Traceroute).

6.3.2 Использование ICMP-сообщений

ICMP-сообщения (тип 12) генерируются при нахождении ошибок в заголовке IP-пакета (за исключением самих ICMP-пакетов, дабы не привести к бесконечно растущему потоку ICMP-сообщений об ICMP-сообщениях).

ICMP-сообщения (тип 3) генерируются маршрутизатором при отсутствии маршрута к адресату.

Утилита Ping, служащая для проверки возможности доставки IP-пакетов использует ICMP-сообщения с типом 8 (эхо-запрос) и 0 (эхо-ответ).

Утилита Traceroute, отображающая путь следования IP-пакетов, использует ICMP-сообщения с типом 11.

ICMP-сообщения с типом 5 используются маршрутизаторами для обновления записей в таблице маршрутизации отправителя.

ICMP-сообщения с типом 4 используются получателем (или маршрутизатором) для управления скоростью отправки сообщений отправителем.

Полный перечень типов ICMP сообщений вы можете найти в приложении Д.

6.3.3 Формат пакета ICMP

Формат пакета ICMP			
Бит	0—7	8—15	16—31
0	Тип	Код	Контрольная сумма
32	Содержание сообщения (зависит от значений полей «Код» и «Тип»)		

6.4 Утилита Ping

Ping — утилита для проверки соединений в сетях на основе TCP/IP. Буквальный перевод слова «ping» с английского на русский «свист», «визг», «стук», «гудение».

Она отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT, от англ. Round Trip Time) позволяет определять двусторонние задержки (RTT) по маршруту и частоту потери пакетов, то есть косвенно определять загруженность на каналах передачи данных и промежуточных устройствах.

Также пингом иногда ошибочно называют время, затраченное на передачу пакета информации в компьютерных сетях от клиента к серверу и обратно от сервера к клиенту. Это время называется лагом (англ. отставание; задержка, запаздывание) или собственно задержкой и измеряется в миллисекундах. Лаг связан со скоростью соединения и загруженностью каналов на всём протяжении от клиента к серверу.

Полное отсутствие ICMP-ответов может также означать, что удалённый узел (или какой-либо из промежуточных маршрутизаторов) блокирует ICMP Echo-Reply или игнорирует ICMP Echo-Request.

Программа ping является одним из основных диагностических средств в сетях TCP/IP и входит в поставку всех современных сетевых операционных систем. Функциональность ping также реализована в некоторых встроенных ОС маршрутизаторов, доступ к результатам выполнения ping для таких устройств по протоколу SNMP определяется RFC 2925 (Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations).

Так как для отправки ICMP-пакетов требуется создавать raw-сокеты, для выполнения программы ping в unix-системах необходимы права суперпользователя. Чтобы обычные пользователи могли использовать ping в правах доступа файла /bin/ping устанавливают SUID-бит.

6.4.1 Этимология

Название происходит от английского названия звука, издаваемого сонаром при получении отраженного от исследуемого объекта импульса.

Также есть несколько альтернативных толкований:

1. PING— акроним «Packet InterNet Grouper (Groper)».
2. Ping — часть названия игры пинг-понг. Это толкование подразумевает, что компьютеры обмениваются сигналами аналогично тому, как игроки в пинг-понг отбивают друг другу мяч.

6.4.2 История

Программа была написана Майком Мууссом (англ. Mike Muuss), учёным Исследовательской Лаборатории Баллистики (англ. Ballistics Research Lab) США в декабре 1983 года.

Практическое использование

- Можно узнать IP-адрес по доменному имени.
- Можно узнать, работает ли сервер. Например, системный администратор может узнать завис ли только веб-сервер или на сервере глобальные проблемы.
- Можно узнать, есть ли связь с сервером. Например, проблемы с настройкой DNS серверов на машине можно узнать, задав в ping сначала доменное имя, а потом IP-адрес.
- Так же можно узнать качество канала, посмотрев сколько ответов не пришло. Это часто используется игроками в сетевые игры, потому что качество связи для них очень важно.

6.5 Утилита Traceroute

Traceroute — это служебная компьютерная программа, предназначенная для определения маршрутов следования данных в сетях TCP/IP. Traceroute основана на протоколе ICMP.

Программа traceroute выполняет отправку данных указанному узлу сети, при этом отображая сведения о всех промежуточных маршрутизаторах, через которые прошли данные на пути к целевому узлу. В случае проблем при доставке данных до какого-либо узла программа позволяет определить, на каком именно участке сети возникли неполадки. Здесь хочется отметить, что программа работает только в направлении от источника пакетов и является весьма грубым инструментом для выявления неполадок в сети. В силу особенностей работы протоколов маршрутизации в сети Интернет, обратные маршруты часто не совпадают с прямыми, причем это справедливо для всех промежуточных узлов в трейсе. Поэтому, ICMP ответ от каждого промежуточного узла может идти своим собственным маршрутом, затеряться или прийти с большой задержкой, хотя в реальности с пакетами которые адресованы конечному узлу этого не происходит. Кроме того, на промежуточных маршрутизаторах часто стоит ограничение числа ответов ICMP в единицу времени, что приводит к появлению ложных потерь.

Traceroute входит в поставку большинства современных сетевых операционных систем. В системах Microsoft Windows эта программа носит название tracert, а в системах *nix, GNU/Linux, Cisco IOS и Mac OS — traceroute.

6.5.1 Принцип работы traceroute

Для определения промежуточных маршрутизаторов traceroute отправляет серию (обычно три) пакетов данных целевому узлу, при этом каждый раз увеличивая на 1 значение поля TTL («время жизни»). Это поле обычно указывает максимальное количество маршрутизаторов, которое может быть пройдено пакетом. Первая серия пакетов отправляется с TTL, равным 1, и

поэтому первый же маршрутизатор возвращает обратно сообщение ICMP, указывающее на невозможность доставки данных. Traceroute фиксирует адрес маршрутизатора, а также время между отправкой пакета и получением ответа (эти сведения выводятся на монитор компьютера). Затем traceroute повторяет отправку серии пакетов, но уже с TTL, равным 2, что позволяет первому маршрутизатору пропустить их дальше.

Процесс повторяется до тех пор, пока при определённом значении TTL пакет не достигнет целевого узла. При получении ответа от этого узла процесс трассировки считается завершённым.

На конечном хосте IP-дейтаграмма с TTL = 1 не отбрасывается и не вызывает ICMP-сообщения типа срок истёк, а должна быть отдана приложению. Достижение пункта назначения определяется следующим образом: отсылаемые traceroute дейтаграммы содержат UDP-пакет с таким номером UDP-порта адресата (превышающим 30 000), что он заведомо не используется на адресуемом хосте. В пункте назначения UDP-модуль, получая подобные дейтаграммы, возвращает ICMP-сообщения об ошибке «порт недоступен». Таким образом, чтобы узнать о завершении работы, программе traceroute достаточно обнаружить, что поступило ICMP-сообщение об ошибке этого типа.

6.6 Утилита WHOIS

WHOIS (от англ. who is — «кто такой?») — сетевой протокол прикладного уровня, базирующийся на протоколе TCP (порт 43). Основное применение — получение регистрационных данных о владельцах доменных имен, IP-адресов и автономных систем.

Протокол подразумевает архитектуру «клиент-сервер» и используется для доступа к публичным серверам баз данных (БД) регистраторов IP-адресов и регистраторов доменных имён. Текущая версия этого протокола описана в RFC 3912. Чаще всего WHOIS-клиенты реализованы в виде консольных программ. Однако, поскольку для многих пользователей командная строка недоступна или

неудобна, на основе консольных клиентов обычно создаются веб-формы, доступные пользователям на многих сайтах в Интернете. Кроме того, существуют WHOIS-клиенты и с графическим интерфейсом.

Изначально целью появления системы WHOIS на свет было дать возможность системным администраторам искать контактную информацию других администраторов IP-адресов или доменных имен (аналогично «Белым страницам», «white pages»).

У WHOIS есть аналог — стандарт протокола RWhois.

6.6.1 Централизованная и распределённая модели

Базы данных, имеющие WHOIS-интерфейс, бывают централизованными и распределёнными.

В первом случае один WHOIS-сервер содержит полную БД и отвечает на запросы, касающиеся всех регистраторов. По такой схеме построен WHOIS-сервер для домена .org и .ru.

Во втором случае центральный WHOIS-сервер не содержит полную БД и лишь перенаправляет пользователя на WHOIS-сервер соответствующего регистратора. По такой схеме работает WHOIS для домена .com. Когда WHOIS-клиент «умеет» распознавать такое перенаправление, он сам запрашивает нужный периферийный WHOIS-сервер, в противном случае пользователю приходится делать это вручную. В протоколе WHOIS не предусмотрено различие централизованной и распределённой моделей.

Конкретная реализация хранилища регистрационных записей зависит от регистратора. Некоторые домены верхнего уровня, включая .com и .net, используют распределённую архитектуру, что дает регистраторам возможность включать в записи произвольную информацию о своих клиентах. Другие реестры доменов, например .org, хранят данные централизованно.

6.6.2 Интернационализация протокола

Изначально протокол WHOIS был создан без учета поддержки национальных языков (с кодировкой символов ASCII) и сам по себе стандарт никак не регламентирует кодировку запросов клиента и ответа сервера. Для избежания этой неопределенности, в частности, в случае использования доменных имен и зон на национальном языке в настоящее время используется специальный алгоритм кодирования punycode.

6.7 Утилита Wireshark

Wireshark (ранее — **Ethereal**) — программа-анализатор трафика для компьютерных сетей Ethernet и некоторых других. Имеет графический пользовательский интерфейс. В июне 2006 года проект был переименован в Wireshark из-за проблем с торговой маркой[2].

Функциональность, которую предоставляет Wireshark, очень схожа с возможностями программы tcpdump, однако Wireshark имеет графический пользовательский интерфейс и гораздо больше возможностей по сортировке и фильтрации информации. Программа позволяет пользователю просматривать весь проходящий по сети трафик в режиме реального времени, переводя сетевую карту в неразборчивый режим (англ. promiscuous mode).

Программа распространяется под свободной лицензией GNU GPL и использует для формирования графического интерфейса кроссплатформенную библиотеку GTK+. Существуют версии для большинства типов UNIX, в том числе Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Mac OS X, а также для Windows.

Wireshark — это приложение, которое «знает» структуру самых различных сетевых протоколов, и поэтому позволяет разобрать сетевой пакет, отображая значение каждого поля протокола любого уровня. Поскольку для захвата пакетов используется pcap, существует возможность захвата данных только из тех сетей, которые поддерживаются этой библиотекой. Тем не менее, Wireshark умеет работать с множеством форматов входных данных,

соответственно, можно открывать файлы данных, захваченных другими программами, что расширяет возможности захвата.

Внешний вид программы изображён на рис. 6.1.

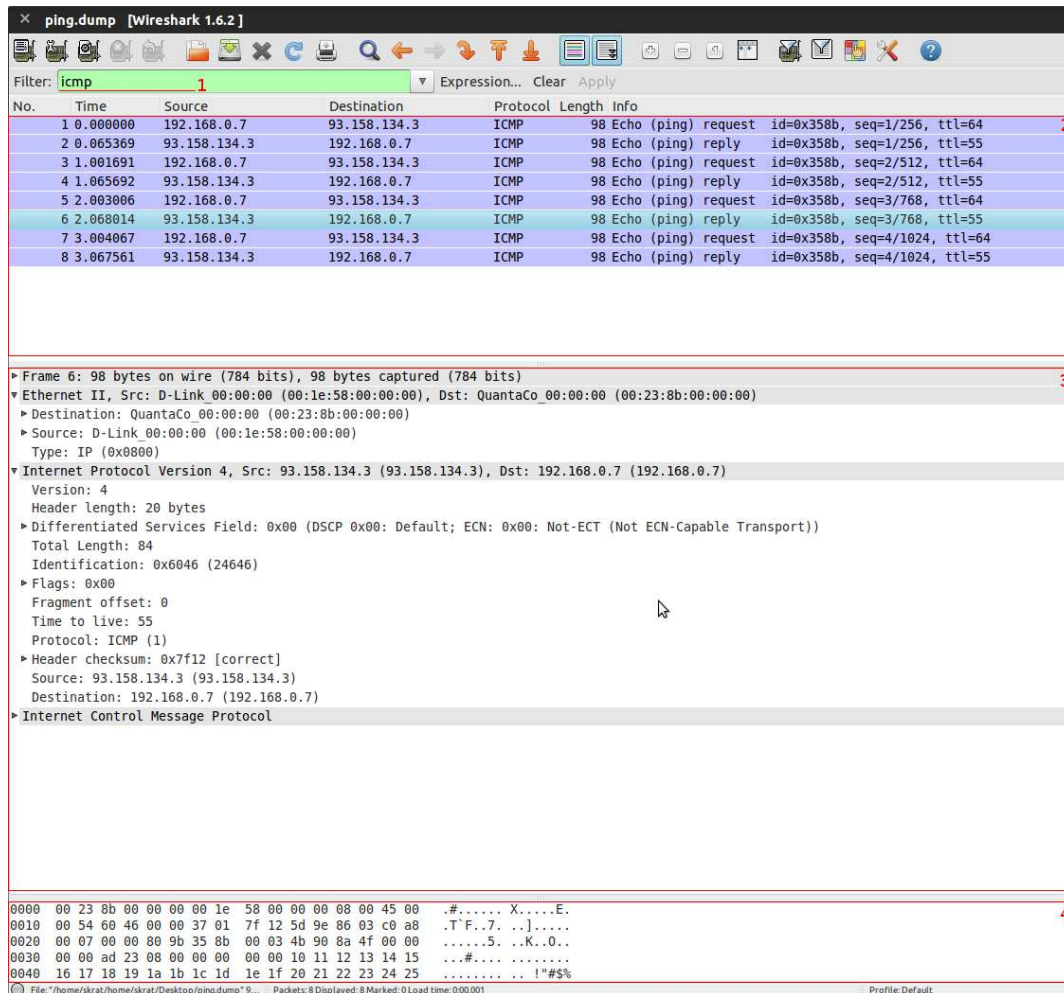


Рисунок 6.1 — Внешний вид программы Wireshark

Для того чтобы сделать лабораторную работу о программе Wireshark, надо знать совсем немного, а именно 4 вещи, обозначенные красной рамкой:

- Фильтр для отображаемых пакетов;
- Список отфильтрованных пакетов;
- Содержимое пакета в разобранном виде;
- Содержимое пакета в сыром виде.

6.8 Контрольные вопросы

- Каковы могут быть причины потери ICMP пакетов?
- Почему до некоторых узлов пакеты могут идти разными маршрутами?
- Совпадает ли маршрут пакета запроса с маршрутом пакета ответа?
- Почему некоторые узлы не отвечают на пакеты, посланные утилитой traceroute, хотя отвечают на пакеты утилиты ping?
- Почему некоторые узлы недоступны, если они являются получателями пакетов утилиты traceroute, хотя корректно отвечает на транзитные пакеты?
- Сколько допустимых IP адресов находится в сети с префиксом /27?
- Сколько сетей с префиксом /27 можно разместить в сети класса B?
- Почему для некоторых IP адресов в результате whois запроса возвращается несколько сетей?

6.9 Задание

1. Из списка узлов, перечисленных в приложении Г выберите не менее 5-ти узлов для исследования. Так же вы можете предложить свои варианты узлов сети для исследования;
2. Используя ICMP запрос типа 8 вычислите IP-адрес узла сети, минимальное, среднее и максимальное время между отправкой запроса и получением ответа для списка узлов, процент успешной доставки пакетов. Для каждого узла необходимо провести не менее 10 экспериментов, в каждом измерении не менее 50-ти запросов;
3. Используя поле времени жизни IP пакета постройте карту маршрутов до каждого выбранного узла сети. После этого необходимо построить маршруты до каждого узла на карте маршрутов. Построение маршрутов до каждого узла необходимо проводить не менее 5-ти раз. Для построения карты маршрутов рекомендуется использовать программу TrueTraceRoute;
4. Постройте карту маршрутов сети;

5. Для каждого узла на карте узнайте к какой сети он принадлежит. Обведите рамкой узлы, принадлежащие одной сети. Если whois запрос в программе TrueTraceRoute не работает, значит настройки подключения к интернет неверны (возможно используется проху-сервер). В этом случае используйте
http://www.db.ripe.net/whois?form_type=simple&full_query_string=&;
6. Разобраться с последовательностью пакетов и их содержимым посылаемым при использовании утилит ping и traceroute;

6.10 Литература

1. <http://ru.wikipedia.org/wiki/Ping>
2. <http://ru.wikipedia.org/wiki/Traceroute>
3. <http://ru.wikipedia.org/wiki/IP>
4. <http://ru.wikipedia.org/wiki/Wireshark>

Лабораторная работа №7. Система доменных имён

Целью работы является изучение принципов работы протокола DNS.

7.1 Описание протокола DNS

DNS (Domain Name System — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись).

Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу (с административной точки зрения — другой организации или человеку), что

позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

Начиная с 2010 года, в систему DNS внедряются средства проверки целостности передаваемых данных, называемые DNS Security Extensions (DNSSEC). Передаваемые данные не шифруются, но их достоверность проверяется криптографическими способами. Внедряемый стандарт DANE обеспечивает передачу средствами DNS достоверной криптографической информации (сертификатов), используемых для установления безопасных и защищённых соединений транспортного и прикладного уровней

7.1.1 Ключевые характеристики DNS

DNS обладает следующими характеристиками:

- Распределённость администрирования. Ответственность за разные части иерархической структуры несут разные люди или организации.
- Распределённость хранения информации. Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности и (возможно) адреса корневых DNS-серверов.
- Кеширование информации. Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.
- Иерархическая структура, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.
- Резервирование. За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

7.1.2 Ключевые понятия DNS:

- Домен (domain — область) — узел в дереве имён, вместе со всеми подчинёнными ему узлами (если таковые имеются), то есть именованная ветвь или поддереву в дереве имен.

Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается слева направо от младших доменов к доменам высшего уровня (в порядке повышения значимости), корневым доменом всей системы является точка ('.'), ниже идут домены первого уровня (географические или тематические), затем — домены второго уровня, третьего и т. д. (например, для адреса kcup.tusur.ru домен первого уровня — ru, второго tusur, третьего kcup). На практике точку в конце имени часто опускают, но она бывает важна в случаях разделения между относительными доменами и FQDN (англ. Fully Qualified Domain Name, полностью определённое имя домена).

- Поддомен (subdomain) — подчинённый домен (например, tusur.ru — поддомен домена ru, а kcup.tusur.ru — домена tusur.ru).

Теоретически такое деление может достигать глубины 127 уровней, а каждая метка может содержать до 63 символов, пока общая длина вместе с точками не достигнет 254 символов. Но на практике регистраторы доменных имён используют более строгие ограничения. Например, если у вас есть домен вида mydomain.ru, вы можете создать для него различные поддомены вида mysite1.mydomain.ru, mysite2.mydomain.ru и т. д.

- Ресурсная запись — единица хранения и передачи информации в DNS.

Каждая ресурсная запись имеет имя (то есть привязана к определенному Доменному имени, узлу в дереве имен), тип и поле данных, формат и содержание которого зависит от типа.

- Зона — часть дерева доменных имен (включая ресурсные записи), размещаемая как единое целое на некотором сервере доменных имен (DNS-сервере, см. ниже), а чаще — одновременно на нескольких серверах.

Целью выделения части дерева в отдельную зону является передача ответственности (см. ниже) за соответствующий домен другому лицу или организации. Это называется делегированием (см. ниже). Как связанная часть дерева, зона внутри тоже представляет собой дерево. Если рассматривать пространство имен DNS как структуру из зон, а не отдельных узлов/имен, тоже получается дерево; оправданно говорить о родительских и дочерних зонах, о старших и подчиненных. На практике, большинство зон 0-го и 1-го уровня ('.', ru, com, ...) состоят из единственного узла, которому непосредственно подчиняются дочерние зоны. В больших корпоративных доменах (2-го и более уровней) иногда встречается образование дополнительных подчиненных уровней без выделения их в дочерние зоны.

- Делегирование — операция передачи ответственности за часть дерева доменных имен другому лицу или организации.

За счет делегирования в DNS обеспечивается распределенность администрирования и хранения. Технически делегирование выражается в выделении этой части дерева в отдельную зону, и размещении этой зоны на DNS-сервере (см. ниже), управляемом этим лицом или организацией. При этом в родительскую зону включаются «склеивающие» ресурсные записи (NS и A), содержащие указатели на DNS-сервера дочерней зоны, а вся остальная информация, относящаяся к дочерней зоне, хранится уже на DNS-серверах дочерней зоны.

- DNS-сервер — специализированное ПО для обслуживания DNS, а также компьютер, на котором это ПО выполняется. DNS-сервер может

быть ответственным за некоторые зоны и/или может перенаправлять запросы вышестоящим серверам.

- DNS-клиент — специализированная библиотека (или программа) для работы с DNS. В ряде случаев DNS-сервер выступает в роли DNS-клиента.

Авторитетность (authoritative) — признак размещения зоны на DNS-сервере.

Ответы DNS-сервера могут быть двух типов: авторитетные (когда сервер заявляет, что сам отвечает за зону) и неавторитетные (англ. Non-authoritative), когда сервер обрабатывает запрос, и возвращает ответ других серверов. В некоторых случаях вместо передачи запроса дальше DNS-сервер может вернуть уже известное ему (по запросам ранее) значение (режим кеширования).

DNS-запрос (англ. DNS query) — запрос от клиента (или сервера) серверу. Запрос может быть рекурсивным или нерекурсивным.

Система DNS содержит иерархию DNS-серверов, соответствующую иерархии зон. Каждая зона поддерживается как минимум одним авторитетным сервером DNS (от англ. authoritative — авторитетный), на котором расположена информация о домене.

Имя и IP-адрес не тождественны — один IP-адрес может иметь множество имён, что позволяет поддерживать на одном компьютере множество веб-сайтов (это называется виртуальный хостинг). Обратное тоже справедливо — одному имени может быть сопоставлено множество IP-адресов: это позволяет создавать балансировку нагрузки.

Для повышения устойчивости системы используется множество серверов, содержащих идентичную информацию, а в протоколе есть средства, позволяющие поддерживать синхронность информации, расположенной на разных серверах. Существует 13 корневых серверов, их адреса практически не изменяются

Протокол DNS использует для работы TCP- или UDP-порт 53 для ответов на запросы. Традиционно запросы и ответы отправляются в виде одной UDP датаграммы. TCP используется для AXFR-запросов.

7.1.3 Типы запросов в DNS

Термином Рекурсия в DNS обозначают алгоритм поведения DNS-сервера, при котором сервер выполняет от имени клиента полный поиск нужной информации во всей системе DNS, при необходимости обращаясь к другим DNS-серверам.

DNS-запрос может быть рекурсивным — требующим полного поиска, — и нерекурсивным (или итеративным) — не требующим полного поиска.

Аналогично, DNS-сервер может быть рекурсивным (умеющим выполнять полный поиск) и нерекурсивным (не умеющим выполнять полный поиск). Некоторые программы DNS-серверов, например, BIND, можно сконфигурировать так, чтобы запросы одних клиентов выполнялись рекурсивно, а запросы других — нерекурсивно.

При ответе на нерекурсивный запрос, а также — при неумении или запрете выполнять рекурсивные запросы, — DNS-сервер либо возвращает данные о зоне, за которую он ответствен, либо возвращает адреса серверов, которые обладают большим объёмом информации о запрошенной зоне, чем отвечающий сервер, чаще всего — адреса корневых серверов.

В случае рекурсивного запроса DNS-сервер опрашивает серверы (в порядке убывания уровня зон в имени), пока не найдёт ответ или не обнаружит, что домен не существует. (На практике поиск начинается с наиболее близких к искомому DNS-серверов, если информация о них есть в кэше и не устарела, сервер может не запрашивать другие DNS-серверы.)

Рассмотрим на примере работу всей системы.

Предположим, мы набрали в браузере адрес `ksup.tusur.ru`. Браузер спрашивает у сервера DNS: «какой IP-адрес у `ksup.tusur.ru`»? Однако, сервер DNS может ничего не знать не только о запрошенном имени, но даже обо всём

домене tusur.ru. В этом случае сервер обращается к корневому серверу — например, 82.117.162.140. Этот сервер сообщает — «У меня нет информации о данном адресе, но я знаю, что 204.74.112.1 является ответственным за зону ru.» Тогда сервер DNS направляет свой запрос к 204.74.112.1, но тот отвечает «У меня нет информации о данном сервере, но я знаю, что 212.192.120.60 является ответственным за зону tusur.ru.» Наконец, тот же запрос отправляется к третьему DNS-серверу и получает ответ — IP-адрес, который и передаётся клиенту — браузеру.

В данном случае при разрешении имени, то есть в процессе поиска IP по имени:

браузер отправил известному ему DNS-серверу рекурсивный запрос — в ответ на такой тип запроса сервер обязан вернуть «готовый результат», то есть IP-адрес, либо пустой ответ и код ошибки NXDOMAIN;

DNS-сервер, получивший запрос от браузера, последовательно отправлял нерекурсивные запросы, на которые получал от других DNS-серверов ответы, пока не получил ответ от сервера, ответственного за запрошенную зону;

остальные упоминавшиеся DNS-серверы обрабатывали запросы нерекурсивно (и, скорее всего, не стали бы обрабатывать запросы рекурсивно, даже если бы такое требование стояло в запросе).

Иногда допускается, чтобы запрошенный сервер передавал рекурсивный запрос «вышестоящему» DNS-серверу и дожидался готового ответа.

При рекурсивной обработке запросов все ответы проходят через DNS-сервер, и он получает возможность кэшировать их. Повторный запрос на те же имена обычно не идет дальше кэша сервера, обращения к другим серверам не происходит вообще. Допустимое время хранения ответов в кэше приходит вместе с ответами (поле TTL ресурсной записи).

Рекурсивные запросы требуют больше ресурсов от сервера (и создают больше трафика), так что обычно принимаются от «известных» владельцу сервера узлов (например, провайдер предоставляет возможность делать

рекурсивные запросы только своим клиентам, в корпоративной сети рекурсивные запросы принимаются только из локального сегмента). Нерекурсивные запросы обычно принимаются ото всех узлов сети (и содержательный ответ даётся только на запросы о зоне, которая размещена на узле, на DNS-запрос о других зонах обычно возвращаются адреса других серверов).

7.1.4 Записи DNS

Записи DNS, или Ресурсные записи — единицы хранения и передачи информации в DNS. Каждая ресурсная запись состоит из следующих полей:

- имя (NAME) — доменное имя, к которому привязана или которому «принадлежит» данная ресурсная запись,
- TTL (Time To Live) — допустимое время хранения данной ресурсной записи в кэше неответственного DNS-сервера,
- тип (TYPE) ресурсной записи — определяет формат и назначение данной ресурсной записи,
- класс (CLASS) ресурсной записи; теоретически считается, что DNS может использоваться не только с TCP/IP, но и с другими типами сетей, код в поле класс определяет тип сети,
- длина поля данных (RDLEN),
- поле данных (RDATA), формат и содержание которого зависит от типа записи.

Наиболее важные типы DNS-записей:

- Запись SOA (Start of Authority) или начальная запись зоны указывает, на каком сервере хранится эталонная информация о данном домене, содержит контактную информацию лица, ответственного за данную зону, тайминги (параметры времени) кеширования зонной информации и взаимодействия DNS-серверов.

- Запись A (address record) или запись адреса связывает имя хоста с адресом IP. Например, запрос A-записи на имя kcup.tusur.ru вернет его IP адрес — 88.204.72.214
- Запись AAAA (IPv6 address record) связывает имя хоста с адресом протокола IPv6. Например, запрос AAAA-записи на имя google.com вернет его IPv6 адрес — 2404:6800:4002:800::1000
- Запись CNAME (canonical name record) или каноническая запись имени (псевдоним) используется для перенаправления на другое имя
- Запись MX (Mail eXchange) или почтовый обменник указывает сервер(ы) обмена почтой для данного домена.
- Запись NS (name server) указывает на список DNS-серверов, ответственный за хранение ресурсных записей для данного домена.
- Запись PTR (pointer) или запись указателя связывает IP хоста с его каноническим именем. Запрос в домене in-addr.arpa на IP хоста в reverse форме вернёт имя (FQDN) данного хоста (см. Обратный DNS-запрос). Например, (на момент написания), для IP адреса 192.0.34.164: запрос записи PTR 164.34.0.192.in-addr.arpa вернет его каноническое имя referrals.icann.org. В целях уменьшения объёма нежелательной корреспонденции (спама) многие серверы-получатели электронной почты могут проверять наличие PTR записи для хоста, с которого происходит отправка. В этом случае PTR запись для IP адреса должна соответствовать имени отправляющего почтового сервера, которым он представляется в процессе SMTP сессии.
- SRV-запись (server selection) указывает на серверы для сервисов, используется, в частности, для Jabber и Active Directory.

7.1.5 Балансировка нагрузки

Балансировка нагрузки может быть использована для расширения возможностей фермы серверов, состоящей более чем из одного сервера. Она также может позволить продолжать работу даже в условиях, когда несколько

серверов вышли из строя. Благодаря этому растёт отказоустойчивость, за счёт установки устройств в кластере. Кроме этого, балансировка нагрузки используется, когда один сервер физически не может справиться с обработкой всех поступающих запросов.

В протоколе DNS балансировка нагрузки достигается наличием нескольких серверов имен, ответственных за каждую зону. Кроме того, многие регистраторы доменных имен требуют, чтобы сервера имен, ответственные за одну зону располагались в разных IP сетях.

Кроме этого, сами DNS серверы могут балансировать нагрузку на сервисы обслуживаемой доменной зоны. Самым простым способом балансировки нагрузки при помощи DNS является алгоритм Round robin DNS

7.1.5.1 Round robin DNS

Round robin DNS — один из методов распределения нагрузки, или отказоустойчивости за счёт избыточности количества серверов, с помощью управления ответами DNS-сервера в соответствии с некой статистической моделью. Обычно применяется к таким Интернет-протоколам, как Веб-серверы, FTP-серверы.

В простейшем случае Round robin DNS работает, отвечая на запросы не только одним IP-адресом, а списком из нескольких адресов серверов, предоставляющих идентичный сервис. Порядок, в котором возвращаются IP-адреса из списка, основан на алгоритме round-robin. С каждым ответом последовательность ip-адресов меняется. Как правило, простые клиенты пытаются устанавливать соединения с первым адресом из списка, таким образом разным клиентам будут выданы адреса разных серверов, что распределит общую нагрузку между серверами.

Не существует стандартной процедуры для определения того, какие адреса будут использоваться запрашивающим приложением — некоторые серверы пытаются изменить порядок списка, уделяя приоритетное внимание численно более «близким» сетям. Некоторые настольные клиенты пытаются

получить альтернативные адреса после того, как не удалось установить соединение в течение 30-45 секунд.

Круговая система DNS часто используется для распределения нагрузки территориально-распределенных веб-серверов. Например, у компании есть один домен и три идентичных веб-сайта, расположенных на трех серверах с тремя разными адресами. Когда один пользователь получает доступ к главной странице, он будет направлен на первый адрес IP. Второй пользователь, обращающийся к главной странице, будет отправлен на следующий адрес IP, а третий пользователь будет отправлен на третий адрес IP. В каждом случае, когда IP-адрес выдается, он отправляется в конец списка. Четвертый пользователь, следовательно, будет отправлен вновь на первый адрес IP, и так далее.

Хотя Round robin DNS (RR DNS) легко реализовать, всё же этот алгоритм имеет несколько проблематичных недостатков, связанных с кэшированием записи в иерархии RR DNS самого себя, а также с кэшированием на стороне клиента, выданного адреса и его повторного использования, сочетание которых трудно управляемо. RR DNS не опирается на доступность услуг. К примеру, если сервис на одном из адресов недоступен, RR DNS будет продолжать раздавать этот адрес и клиенты будут по-прежнему пытаться добраться до неработоспособных услуг.

Кроме того, оно не может быть лучшим выбором для балансировки нагрузки на самого себя, поскольку он лишь заменяет порядок адресов каждый раз, когда имя сервера запрашивается. Не существует учёта соответствия IP-адреса пользователя и его географического расположения, времени выполнения, нагрузки на сервер, перегрузки сети и т.д. Круговая система DNS нагрузки лучше всего подходит для услуг с большим количеством равномерно распределенных соединений с серверами эквивалентной мощности. В противном случае он просто делает распределение нагрузки.

7.1.5.2 Географическое распределение нагрузки.

Данный метод используется, когда возможностей RR DNS уже недостаточно. Конечно, можно увеличивать количество серверов, обслуживающих какой-либо сервис, причем весьма существенно. Но рано, или поздно мы упремся в длину ответа DNS и перечень всех серверов просто не влезет в ответ.

Так что, рано или поздно, приходится прибегать к географическому распределению нагрузки.

Суть данного метода заключается в том, что в разные диапазоны IP сетей отправляются разные ответы, содержащие разные адреса серверов. В сочетании с RR DNS можно организовать пулы серверов, ответственные за обслуживание клиентов из разных IP сетей, или даже географических зон, объединив IP сети в такие зоны. Например, пул серверов, обслуживающих пользователей России, пользователей западного побережья США, восточного побережья США, Латинской Америки, и т. д.

7.2 Утилита Wireshark

С программой Wireshark вы уже должны были ознакомиться в лабораторной работе №6. Информации необходимой для данной лабораторной работы, по сравнению с предыдущей будет совсем немного. Для того чтобы отфильтровать DNS в поле фильтра (рис. 7.1) необходимо ввести *udp.port == 53* (порт соответствующий протоколу DNS), либо ещё проще просто ввести *dns* (фильтровать все DNS пакеты). Также для удобства, Wireshark содержит встроенные диссекторы для анализа DNS трафика, результат их работы можно увидеть на рис. 7.2.

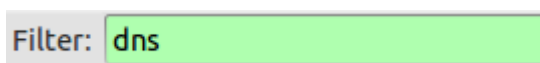


Рисунок 7.1 — Настройка фильтра

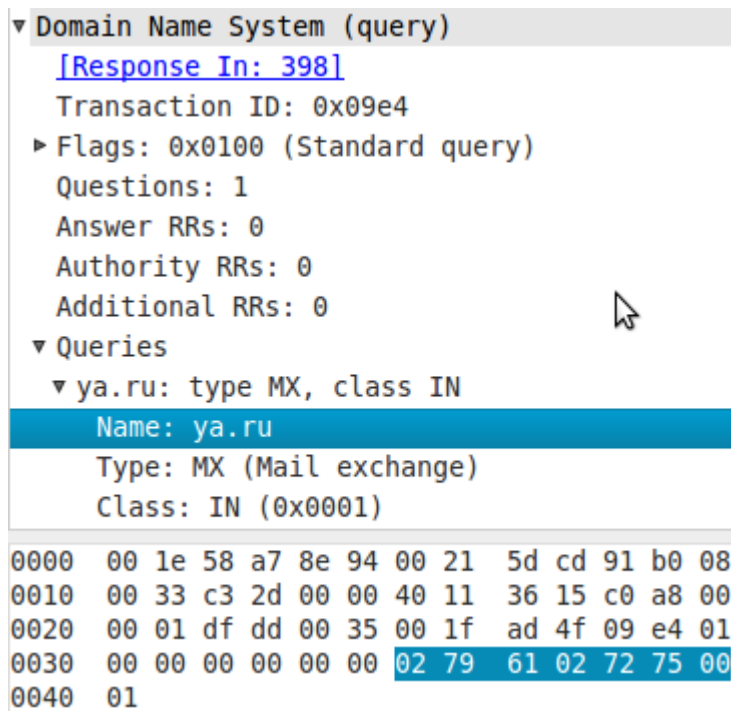


Рисунок 7.2 — Разобранный пакет

7.3 Утилита Nslookup

Существует два режима использования средства **nslookup.exe**: интерактивный и «не интерактивный». «Не интерактивный» режим используется, если ответ может быть получен в виде одного набора данных. Ниже представлен синтаксис, используемый при запуске средства **nslookup** в «не интерактивном» режиме.

```
nslookup [-параметры] [узел] [сервер]
```

Чтобы запустить средство **nslookup.exe** в интерактивном режим, выполните в командной строке команду **nslookup** (пример см. ниже).

```
C:\> nslookup
Default Server:  nameserver1.domain.com
Address:  10.0.0.1
>
```

7.3.1 Поиск данных различных типов

Чтобы найти в адресном пространстве домена данные различных типов, выполните команды **set type** или **set q[querytype]**. Типы могут быть **A**, **CNAME**,

MX, **NS**, **PTR** и другие. Например, чтобы получить список почтовых обменников, выполните следующие команды.

```
>set q=mx
```

Для «не интерактивного» режима это можно сделать при помощи опции **-q=mx**

7.3.2 Получение данных с удалённого сервера имён напрямую

Чтобы направить запрос удалённому серверу имён напрямую, задайте имя нужного сервера с помощью команды **server** или **lserver**. Команда **lserver** определяет адрес сервера на который следует отправлять запросы, используя локальный сервер; команда **server** использует для получения этого адреса текущий сервер по умолчанию.

```
>server 8.8.8.8
```

Для «не интерактивного» режима это можно сделать дописав адрес DNS сервера в конце команды

7.4 Утилита **dig**

dig — это утилита, стандартная для Unix, поэтому работает только в «не интерактивном» режиме.

Синтаксис запуска:

```
dig [@server] [-b address] [-c class] [-f filename] [-k filename]
[-p port#] [-q name] [-t type] [-x addr] [-y [hmac:]name:key] [-4]
[-6] [name] [type] [class] [queryopt...]
```

В общем-то, пользоваться ею проще, чем **nslookup**, несмотря на такой страшный синтаксис. Например, для того, чтобы выполнить поиск серверов обмена почты достаточно набрать:

```
dig gmail.com MX
```

А для того чтобы запросить эту информацию не у сервера по умолчанию, достаточно указать его после символа **@**:

```
dig gmail.com @8.8.8.8
```

7.5 Контрольные вопросы

- Почему и для чего для некоторых зон предусмотрены множественные ip адреса?
- Как ведёт себя клиент, в случае, описанном в п.1?
- Почему некоторые DNS серверы отвечают на одинаковые запросы разными ответами?
- Какой тип DNS сервера обслуживает вашу сеть.

7.6 Задание

Первая часть лабораторной работы выполняется с локальной машины, не с терминального сервера! Отчёт должен содержать несколько разобранных DNS пакетов, с полями сопоставленными с сырыми данными (в hex-виде).

1. Запустить WireShark и настроить для захвата DNS пакетов
2. Открыть в браузере yaplakal.com
3. Дождаться загрузки
4. Переписать сколько и каких DNS запросов было сделано локальным DNS-сервером.
5. Очистить лог WireShark
6. Выполнить запрос разыменования доменного имени gmail.com.
7. Расшифровать содержимое ответного пакета.
8. Подключиться к учебному unix серверу, выполнить пункт 6 с помощью утилиты dig.
9. Сравнить результаты, полученные в пунктах 6 и 8.
10. Выполнить запрос разыменования доменного имени google.com и vk.com к следующим DNS серверам:
DNS сервер по умолчанию;
ns.runnet.ru
8.8.8.8
8.8.4.4
4.2.2.2
198.153.192.1
Сравните результаты. Сделать выводы
11. Выполнить пункт 10 после истечения времени жизни локальной копии квитанции. Сравнить результаты. Сделать выводы
12. Сравнить результаты пунктов 10 и 11. Сделать выводы.

7.7 Литература

1. <http://ru.wikipedia.org/wiki/DNS>
2. <http://ru.wikipedia.org/wiki/Nslookup>
3. <http://ru.wikipedia.org/wiki/Dig>

Лабораторная работа №8. Протокол доставки электронной почты

Цель данной лабораторной работы —изучить принципы прикладных протоколов сетевого взаимодействия на примере протокола SMTP и принцип работы почтовых программ.

8.1 Описание протокола SMTP

SMTP (англ. Simple Mail Transfer Protocol — простой протокол передачи почты) — это сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP. В настоящее время под «протоколом SMTP», как правило, подразумевают ESMTP (англ. *Extended SMTP*) и его расширения.

SMTP используется для отправки почты от пользователей к серверам и между серверами для дальнейшей пересылки к получателю. Для приёма почты почтовый клиент должен использовать протоколы POP3 или IMAP.

Чтобы доставить сообщение до адресата, необходимо переслать его почтовому серверу домена, в котором находится адресат. Для этого обычно используется запись типа MX (англ. Mail eXchange — обмен почтой) системы DNS. Если MX запись отсутствует, то для тех же целей может быть использована запись типа A. Некоторые современные реализации SMTP-серверов (например, Exim) для определения сервера, обслуживающего почту в домене адресата, также могут задействовать SRV-запись (RFC 2782).

Широкое распространение SMTP получил в начале 1980-х годов. До него использовался протокол UUCP, который требовал от отправителя знания полного маршрута до получателя и явного указания этого маршрута в адресе получателя, либо наличия прямого коммутируемого или постоянного соединения между компьютерами отправителя и получателя.

Протокол был разработан для передачи только текста в кодировке ASCII, кроме того, первые спецификации требовали обнуления старшего бита каждого передаваемого байта. Это не даёт возможности отсылать текст на национальных языках (например, кириллице), а также отправлять двоичные файлы (такие как изображения, видеофайлы, программы или архивы). Для снятия этого ограничения был разработан стандарт MIME, который описывает способ преобразования двоичных файлов в текстовые. В настоящее время большинство серверов поддерживают 8BITMIME, позволяющий отправлять двоичные файлы так же просто, как текст.

Сервер SMTP — это конечный автомат с внутренним состоянием. Клиент передает на сервер строку команда<пробел>параметры<перевод строки>. Сервер отвечает на каждую команду строкой, содержащей код ответа и текстовое сообщение, отделенное пробелом. Код ответа — число от 100 до 999, представленное в виде строки, трактуемый следующим образом:

- 2XX — команда успешно выполнена
- 3XX — ожидаются дополнительные данные от клиента
- 4XX — временная ошибка, клиент должен произвести следующую попытку через некоторое время
- 5XX — неустраняемая ошибка

Текстовая часть ответа носит справочный характер и предназначена для человека, а не программы.

ESMTP — расширяемый протокол, в отличие от SMTP. При установлении соединения сервер объявляет о наборе поддерживаемых расширений (в качестве ответа на команду EHLO). Соответствующие расширения могут быть использованы клиентом при работе. Необходимо помнить, что если сессия начинается с команды HELO (используемой в «классическом» SMTP, RFC 821), то список расширений выводиться не будет.

8.2 Пример отправки письма клиентом

8.2.1 Подключаемся к Unix-серверу;

Для этого используем программу putty. Параметры подключения из учебных аудиторий кафедры КСУП:

- Host Name: x.kcup.tusur.ru;
- Port: 22;

Параметры подключения не из учебных аудиторий кафедры КСУП:

- Host Name: kcup.tusur.ru;
- Port: 2222;

8.2.2 Выясняем адрес SMTP сервера;

Воспользуемся утилитой dig. Эта утилита предназначена для просмотра записей DNS. Например, выполните команду `dig kcup.tusur.ru MX`

Результат этой команды представлен ниже:

```
; <<>> DiG 9.4.2 <<>> kcup.tusur.ru MX
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18401
;; flags:  qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 2,
ADDITIONAL: 3

;; QUESTION SECTION:
;kcup.tusur.ru.                IN      MX

;; ANSWER SECTION:
kcup.tusur.ru.                86400   IN      MX      5 kcup.tusur.ru.
kcup.tusur.ru.                86400   IN      MX      5 mx1.tusur.ru.
kcup.tusur.ru.                86400   IN      MX
esafe.keva.tusur.ru.
;; AUTHORITY SECTION:
kcup.tusur.ru.                86400   IN      NS      ns.tusur.ru.
```

```
kcup.tusur.ru.      86400    IN      NS      kup.tusur.ru.
```

```
;; ADDITIONAL SECTION:
```

```
esafe.keva.tusur.ru. 71071    IN      A      79.136.240.4
kcup.tusur.ru.      86400    IN      A      88.204.72.214
ns.tusur.ru.        1857     IN      A      212.192.120.60
```

```
;; Query time: 3 msec
```

```
;; SERVER: 192.168.0.1#53(192.168.0.1)
```

```
;; WHEN: Tue May 4 20:08:10 2010
```

```
;; MSG SIZE rcvd: 173
```

Здесь мы видим в секции ANSWER SECTION, что принимать e-mail в зоне kup.tusur.ru могут 3 сервера: kup.tusur.ru (он сам может это сделать), mx1.tusur.ru (на случай, если нет связи с кафедрой или корпусом, как только связь появится, почта будет доставлена), и esafe.keva.tusur.ru (здесь располагается Анти-спам фильтр). Заметим так же, что перед их именами есть цифры, указывающие приоритет доставки. Самым приоритетным является esafe.keva.tusur.ru.

В секции ADDITIONAL SECTION мы можем узнать IP адреса этих серверов

8.2.3 Подключаемся к SMTP серверу

В качестве сервера выберем наиболее сервер с именем kup.tusur.ru. Его IP адрес 88.204.72.214. Для подключения наберём команду telnet 88.204.72.214 25. где 25 номер порта, на котором ожидает запросы почтовый сервер. Этот порт является стандартным для протокола SMTP.

```
[user@X ~]$ telnet 88.204.72.214 25
```

```
Trying 88.204.72.214...
```

```
Connected to kup.SUR.net.ru.
```

```
Escape character is '^]'.
```

```
220 kup.tusur.ru ESMTP Exim 4.69 Sat, 08 May 2010 13:09:31 +0700
```

Здесь после того как мы подключились к почтовому серверу, сервер уведомляет нас, что для завершения сессии необходимо нажать клавиши Ctrl+]. Далее следует код 220, означающий что всё в порядке, имя сервера, заявляется

поддержка расширенного протокола ESMTP (Extended SMTP), название почтового сервера и его версия, текущие дата и время на сервере. Данная строка приветствия на других серверах может очень сильно отличаться. Далее следует приглашение сервера к вводу.

8.2.4 Передаём серверу команду HELO и наш IP адрес

Теперь нужно передать команду HELO и указать свой IP адрес или доменное имя. Вводим следующую команду:

```
[user@X ~]$ HELO x.kcup.tusur.ru
```

Получаем ответ:

```
250 kcup.tusur.ru Hello x.kcup.tusur.ru [192.168.0.7]
```

Здесь получаем удачный ответ с кодом 250, далее следует имя сервера, приветствие, имя клиента и его IP-адрес.

8.2.5 Указываем отправителя

Далее нужно передать команду MAIL FROM: и указать адрес электронной почты в угловых скобках "<" и ">", который будет указываться, как адрес для ответа. В кавычках указывается имя отправителя. Можно не указывать имя пользователя в кавычках.

```
MAIL FROM: "I am a Student" <student@mail.ru>
```

Получаем ответ:

```
250 OK
```

8.2.6 Указывает получателя

Далее указываем адрес получателя после команды RCPT TO:. Формат аналогичен предыдущей команде:

```
RCPT TO: "Dear Teacher" <teacher @kcup.tusur.ru>
```

```
250 Accepted
```

8.2.7 Передаём заголовок письма

Для начала передачи письма необходимо выполнить команду DATA.

```
DATA
```

354 Enter message, ending with "." on a line by itself

Ответ означает, что для завершения отправки письма следует ввести строку, содержащую только символ точка ".".

Пишем заголовок письма:

```
From: "I am a Student" <student@mail.ru>
To: "Dear Teacher" <teacher @kcup.tusur.ru>
Subject: Это моё первое письмо без использования почтовых
клиентов :-)
```

Здесь в поле From: указывается адрес отправителя, который будет указан в почтовом клиенте в поле От кого пришло письмо. Формат аналогичен команде MAIL FROM:, но в общем случае, эти адреса могут не совпадать.

В поле to: указывается получатель. Как и в поле From: адрес может и не совпадать с полем RCPT TO:.

В поле subject: указывается тема письма, если это поле не указать, то оно будет содержать "(без темы)".

Далее могут следовать различные поля, название и содержание которых зависит от почтовых клиентов, отправляющих их, серверов ретрансляторов (SMTP relay), через которые проходят письма от почтовых клиентов к почтовым серверам и собственно почтовых серверов.

Заголовок должен заканчиваться пустой строкой, но для большинства современных серверов это требование не принципиально.

8.2.8 Передаём тело письма;

Далее мы просто передаём тело письма:

```
Здравствуй, уважаемый преподаватель!
Я успешно доставил своё первое электронное письмо без
использования почтовых программ.
```

Буду рад, если мои старания будут соответственно оценены.

С Уважением, Ваш студент, ФИО, № группы.

.

Как говорилось ранее, тело письма должно заканчиваться строкой, содержащей только символ точка ".". После ввода такой строки сервер вернёт нам:

```
250 OK id=10AeDT-000LNB-C6
```

ЭТО ОЗНАЧАЕТ, ЧТО ПИСЬМО ПРИНЯТО К ДОСТАВКЕ, ДАЛЕЕ СЛЕДУЕТ ЕГО УНИКАЛЬНЫЙ ИДЕНТИФИКАТОР.

8.2.9 Отключаемся от SMTP сервера.

Для этого нам необходимо выполнить команду QUIT

```
QUIT
```

После этой команды следует ответ сервера:

```
221 kcup.tusur.ru closing connection
Connection closed by foreign host.
[user@X ~]$
```

Это означает, что сессия с почтовым сервером закрыта.

8.2.10Послесловие

Если вы точно знаете адрес почтового сервера вы можете выполнить 3-9 в консоли ОС Windows. Для этого вам нужно нажать кнопку Пуск, выполнить, ввести cmd.

8.3 Задание

1. Отправить письмо на электронный адрес преподавателя со своего адреса электронной почты. Для этого узнайте адрес преподавателя, отправьте письмо в качестве адреса отправителя и адреса для ответа укажите свой почтовый адрес. В теме укажите Ваши группу, фамилию, имя и отчество;
1 Отправить письмо на электронный адрес преподавателя с несуществующего адреса электронной почты. Сделайте тоже, что и в пункте 4.1 , только в поле отправителя укажите электронный адрес вида:
your_login@kcup.tusur.ru;

2 Отправить письмо на электронный адрес преподавателя с несуществующего адреса электронной почты, содержащие СПАМ. Большинство СПАМ фильтров фильтруют письма, содержащие следующие фразы:

- This is no SPAM!;
- This is No Joke!;
- You won 1 MILLION DOLLARS!;
- Enlarge your mail!;
- И т.д.

3 Отправить письмо на электронный адрес преподавателя с несуществующего адреса электронной почты, содержащие СПАМ через СПАМ-фильтр. Адрес сервера, на котором работает СПАМ фильтр можно найти в разделе 8.2.2. Попробуйте отправить письмо, что и в пункте 4.3.

4 Отправить письмо на свой электронный адрес. Продемонстрировать преподавателю. Сделайте тоже, что и в пункте 1, только на свой, не кафедральный почтовый ящик. Преподавателю необходимо представить перечень команд, приведший к доставке письма и показать содержимое письма в вашем почтовом ящике.

5 Подготовьте отчет.

Приложение А. Варианты исходных данных для лабораторной работы №1.

Вариант, №	Время процесса дискретизации T_c , с	Элементарный квант времени Δ , с	Период дискретизации T , с	Входной сигнал	Длительность импульса tn , с
1	5	0.001	1/6	$s(t) = \sin(t^{\sin(t)})$	1/24
2	1.5	0.001	1/4	$s(t) = \cos(t^{\sin(t)})$	1/8
3	4	0.001	1/5	$s(t) = \frac{1 - \sin t(t)}{2}$	1/12
4	10	0.001	1/2	$s(t) = \frac{1 + \sin(t)}{2 + \sin(t)}$	1/4
5	6	0.001	1/3	$s(t) = \sin(t) $	1/4
6	5	0.001	1/6	$s(t) = \cos(t^{\sin(t)/t})$	1/20
7	6	0.001	1/4	$s(t) = t \cdot \sin(t) $	1/7
8	7	0.001	1/5	$s(t) = \frac{\sin^2(t)}{t}$	1/5
9	8	0.001	1/1	$s(t) = \sin^4(t) + \cos^2(t)$	1/8
10	7	0.001	1/3	$s(t) = \sin(t) \cdot \Phi(\sin(t))$	1/12
11	6	0.001	1/6	$s(t) = t$	1/12
12	5	0.001	1/4	$s(t) = \frac{1 + \sin(t^2)}{2}$	1/21
13	4	0.001	1/5	$s(t) = \sin(t \cdot \sin(t^2)) $	1/13

Вариант, №	Время процесса дискретизации T_c , с	Элементарный квант времени Δ , с	Период дискретизации T , с	Входной сигнал	Длительность импульса t_n , с
14	3	0.001	1/2	$s(t) = \frac{\Phi(\cos(4t)) + \Phi(\sin(4t))}{2}$	1/3
15	4	0.001	1/3	$s(t) = \sin(t^{\sin(t)})$	1/7
16	1.5	0.001	1/6	$s(t) = \cos(t^{\sin(t)})$	1/8
17	6	0.001	1/4	$s(t) = \frac{1 - \sin t(t)}{2}$	1/4
18	11	0.001	1/5	$s(t) = \frac{1 + \sin(t)}{2 + \sin(t)}$	1/18
19	8	0.001	1/2	$s(t) = \sin(t) $	1/26
20	7	0.001	1/3	$s(t) = \cos(t^{\sin(t)/t})$	1/7
21	5	0.001	1/6	$s(t) = t \cdot \sin(t) $	1/32
22	7	0.001	1/4	$s(t) = \frac{\sin^2(t)}{t}$	1/26
23	7	0.001	1/2	$s(t) = t \cdot (\sin^4(3t) + \cos^2(3t))$	1/13
24	10	0.001	1/2	$s(t) = \sin(t) \cdot \Phi(\sin(t))$	1/40
25	3	0.001	1/3	$s(t) = t^2$	1/3
26	5	0.001	1/5	$s(t) = \frac{1 + \sin(t^2)}{2}$	1/34
27	6	0.001	1/4	$s(t) = \sin(t \cdot \sin(t^2)) $	1/4
28	7	0.001	1/2	$s(t) = \frac{\Phi(\cos(2t)) + \Phi(\sin(2t))}{2}$	1/23

Приложение Б. Варианты исходных данных для лабораторной работы №2.

Вариант №	Порог определения фазового приёмника Pd , с	Порог срабатывания частотного приёмника Fd , с	Входной сигнал
1	0.3	0.45	0 1 0 1 0 1 0 0
2	0.37	0.41	0 1 0 0 1 1 0 0
3	0.39	0.37	0 1 1 1 1 1 0 1
4	0.36	0.39	1 1 1 0 1 0 1 0
5	0.32	0.42	1 1 0 0 1 1 0 0
6	0.38	0.44	1 0 0 1 0 1 0 1
7	0.41	0.40	0 1 0 1 0 1 1 0
8	0.31	0.37	1 1 0 1 1 0 1 0
9	0.34	0.36	1 1 0 1 1 0 1 1
10	0.38	0.39	1 1 1 0 0 0 0 1
11	0.33	0.40	1 0 0 1 0 1 0 1
12	0.35	0.38	1 0 1 0 1 1 1 1
13	0.32	0.36	0 1 0 0 0 1 1 0
14	0.4	0.35	0 1 0 1 1 1 1 0
15	0.3	0.45	1 0 1 1 0 1 1 1
16	0.37	0.41	1 1 0 1 0 0 0 1

Вариант №	Порог определения фазового приёмника Pd , с	Порог срабатывания частотного приёмника Fd , с	Входной сигнал
17	0.39	0.37	0 0 1 1 0 0 1 1
18	0.36	0.39	1 1 0 0 1 1 0 1
19	0.32	0.42	0 0 1 1 1 0 1 1
20	0.38	0.44	1 0 0 1 1 0 1 1
21	0.41	0.40	0 0 1 0 0 0 1 1
22	0.31	0.37	0 1 0 0 0 0 1 1
23	0.34	0.36	1 0 0 0 0 1 1 0
24	0.38	0.39	1 0 1 1 0 0 0 1
25	0.33	0.40	0 1 0 1 0 1 0 0
26	0.35	0.38	0 1 1 1 0 1 0 1
27	0.32	0.36	1 0 0 1 1 0 0 0
28	0.4	0.35	0 1 0 1 0 0 0 1

Приложение В. Варианты исходных данных для лабораторной работы №4.

Вариант №	Продолжительность сигнала T , с	Частота несущей для первого сигнала ω_1 , Гц	Безопасный запас частот $\Delta\omega$, Гц	Период передачи одного сигнала для ВРК ΔT , с	Входной сигнал 1	Входной сигнал 2
1	6	40	15	0.1	$s(t) = \sin(t^{\sin(t)})$	$s(t) = 1 + \cos(t^{\sin(t)})$
2	6	35	15	0.15	$s(t) = 1 + \cos(t^{\sin(t)})$	$s(t) = \frac{1 - \sin(t)}{2}$
3	4	30	15	0.12	$s(t) = \frac{1 - \sin(t)}{2}$	$s(t) = \frac{1 + \sin(t)}{2 + \sin(t)}$
4	6	20	15	0.14	$s(t) = \frac{1 + \sin(t)}{2 + \sin(t)}$	$s(t) = \sin(t) $
5	5	25	25	0.13	$s(t) = \sin(t) $	$s(t) = \cos(t^{\sin(t)/t})$
6	5	25	25	0.16	$s(t) = \cos(t^{\sin(t)/t})$	$s(t) = t \cdot \sin(t) $
7	6	20	15	0.2	$s(t) = t \cdot \sin(t) $	$s(t) = \frac{\sin^2(t)}{t}$
8	7	15	20	0.19	$s(t) = \frac{\sin^2(t)}{t}$	$s(t) = \sin^4(t) + \cos^2(t)$
9	7	15	15	0.17	$s(t) = \sin^4(t) + \cos^2(t)$	$s(t) = \sin(t) \cdot \Phi(\sin(t))$
10	6	20	25	0.18	$s(t) = \sin(t) \cdot \Phi(\sin(t))$	$s(t) = t$

Ва- риант №	Продолжительность сигнала T , с	Частота несущей для перво- го сигнала ω_1 , Гц	Безопасный запас частот $\Delta\omega$, Гц	Период пе- редачи од- ного сигнала для ВРК ΔT , с	Входной сигнал 1	Входной сигнал 2
11	5	25	20	0.1	$s(t) = t$	$s(t) = \frac{1 + \sin(t^2)}{2}$
12	4	30	20	0.15	$s(t) = \frac{1 + \sin(t^2)}{2}$	$s(t) = \sin(\sin(t^2)) $
13	3	35	25	0.12	$s(t) = \sin(\sin(t^2)) $	$s(t) = \frac{\Phi(\cos(4t)) + \Phi(\sin(4t))}{2}$
14	4	30	25	0.14	$s(t) = \frac{\Phi(\cos(4t)) + \Phi(\sin(4t))}{2}$	$s(t) = \sin(t^{\sin(t)})$
15	6	40	20	0.13	$s(t) = \sin(t^{\sin(t)})$	$s(t) = 1 + \cos(t^{\sin(t)})$
16	6	40	20	0.16	$s(t) = 1 + \cos(t^{\sin(t)})$	$s(t) = \frac{1 - \sin(t)}{2}$
17	6	25	15	0.2	$s(t) = \frac{1 - \sin(t)}{2}$	$s(t) = \frac{1 + \sin(t)}{2 + \sin(t)}$
18	8	15	20	0.19	$s(t) = \frac{1 + \sin(t)}{2 + \sin(t)}$	$s(t) = \sin(t) $
19	7	15	20	0.17	$s(t) = \sin(t) $	$s(t) = \cos(t^{\sin(t)/t})$
20	5	25	15	0.18	$s(t) = \cos(t^{\sin(t)/t})$	$s(t) = t \cdot \sin(t) $
21	5	25	15	0.1	$s(t) = t \cdot \sin(t) $	$s(t) = \frac{\sin^2(t)}{t}$

Ва- риант №	Продолжительность сигнала T , с	Частота несущей для перво- го сигнала ω_1 , Гц	Безопасный запас частот $\Delta\omega$, Гц	Период пе- редачи од- ного сигнала для ВРК ΔT , с	Входной сигнал 1	Входной сигнал 2
22	7	15	20	0.15	$s(t) = \frac{\sin^2(t)}{t}$	$s(t) = t \cdot (\sin^4(3t) + \cos^2(3t))$
23	7	15	20	0.12	$s(t) = t \cdot (\sin^4(3t) + \cos^2(3t))$	$s(t) = \sin(t) \cdot \Phi(\sin(t))$
24	3	35	30	0.14	$s(t) = \sin(t) \cdot \Phi(\sin(t))$	$s(t) = t^2$
25	3	35	25	0.13	$s(t) = t^2$	$s(t) = \frac{1 + \sin(t^2)}{2}$
26	5	25	20	0.16	$s(t) = \frac{1 + \sin(t^2)}{2}$	$s(t) = \sin(\sin(t^2)) $
27	6	25	15	0.2	$s(t) = \sin(\sin(t^2)) $	$s(t) = \frac{\Phi(\cos(2t)) + \Phi(\sin(2t))}{2}$
28	1.5	40	25	0.19	$s(t) = \frac{\Phi(\cos(2t)) + \Phi(\sin(2t))}{2}$	$s(t) = \sin(t^{\sin(t)})$

Приложение Г. Варианты исходных данных для лабораторной работы №6. Список адресов для исследования

Вариант №	DNS имена				
	1	2	3	4	5
1	google.com	fasebook.com	vkontakte.ru	vk.com	google.ru
2	google.com	fasebook.com	vkontakte.ru	vk.com	gmail.com

Вариант №	DNS имена				
	1	2	3	4	5
3	gmail.com	google.ru	google.com	vk.com	vkontakte.ru
4	gmail.com	google.ru	google.com	fasebook.com	vk.com
5	google.com	fasebook.com	vkontakte.ru	google.ru	gmail.com
6	google.com	fasebook.com	vkontakte.ru	youtube.com	vk.com
7	google.ru	google.com	fasebook.com	vk.com	youtube.com
8	google.ru	google.com	fasebook.com	youtube.com	vkontakte.ru
9	google.com	fasebook.com	vkontakte.ru	gmail.com	youtube.com
10	youtube.com	gmail.com	google.ru	vkontakte.ru	vk.com
11	youtube.com	gmail.com	google.ru	vk.com	fasebook.com
12	youtube.com	gmail.com	google.ru	fasebook.com	vkontakte.ru
13	gmail.com	google.ru	google.com	vk.com	youtube.com
14	youtube.com	gmail.com	google.ru	google.com	vkontakte.ru
15	google.ru	google.com	fasebook.com	gmail.com	youtube.com
16	google.com	fasebook.com	vkontakte.ru	twitter.com	vk.com
17	google.ru	google.com	fasebook.com	vk.com	twitter.com
18	google.ru	google.com	fasebook.com	twitter.com	vkontakte.ru
19	google.com	fasebook.com	vkontakte.ru	gmail.com	twitter.com
20	google.com	fasebook.com	vkontakte.ru	gmail.com	twitter.com
21	gmail.com	google.ru	google.com	twitter.com	vk.com
22	gmail.com	google.ru	google.com	vkontakte.ru	twitter.com
23	gmail.com	google.ru	google.com	fasebook.com	twitter.com
24	google.com	fasebook.com	vkontakte.ru	youtube.com	twitter.com
25	google.ru	google.com	fasebook.com	twitter.com	youtube.com
26	twitter.com	youtube.com	gmail.com	vkontakte.ru	vk.com
27	twitter.com	youtube.com	gmail.com	fasebook.com	vk.com
28	youtube.com	gmail.com	google.ru	twitter.com	vk.com

Приложение Д. Типы пакетов ICMP (полный список)

Тип	Код	Сообщение	Данные (длина, бит)
0	0	Эхо-ответ	Идентификатор (16) Номер последовательности (16) Данные (переменная)
1, 2		Зарезервировано	
3		Адресат недоступен	Не используется (32) Заголовок IP, Начало исходной дейтаграммы (64)
	0	Сеть недостижима	
	1	Узел недостижим	
	2	Протокол недостижим	
	3	Порт недостижим	
	4	Необходима фрагментация, но установлен флаг ее запрета (DF)	
	5	Неверный маршрут от источника	
	6	Сеть назначения неизвестна	
	7	Узел назначения неизвестен	
	8	Узел источник изолирован	
	9	Сеть административно запрещена	
	10	Узел административно запрещён	
	11	Сеть недоступна для ToS	

Тип	Код	Сообщение	Данные (длина, бит)
	12	Узел недоступен для ToS	
	13	Коммуникации административно запрещены	
	14	Нарушение порядка предпочтения узлов	
	15	Активно отсечение порядка предпочтения	
4	0	Сдерживание источника (отключение источника при переполнении очереди)	
5	Перенаправление		Адрес маршрутизатора (32) Заголовок IP, Начало исходной дейтаграммы (64)
	0	Перенаправление пакетов в сеть	
	1	Перенаправление пакетов к узлу	
	2	Перенаправление для каждого типа обслуживания (ToS)	
	3	Перенаправление пакета к узлу для каждого типа обслуживания	
6	0	Альтернативный адрес узла	
7		Зарезервировано	
8	0	Эхо-запрос	Идентификатор (16) Номер последовательности (16) Данные (переменная)

Тип	Код	Сообщение	Данные (длина, бит)
9	0	Объявление маршрутизатора	Количество адресов (8) Размер элемента (8) Срок действия (16) Адрес[1] (32) Предпочтительность[1] (32) ... Адрес[N] (32) Предпочтительность[N] (32)
10	0	Запрос маршрутизатора	Не используется (32)
11	Время жизни дейтаграммы истекло		Не используется (32) Заголовок IP, Начало исходной дейтаграммы (64)
	0	Время жизни пакета (TTL) истекло при транспортировке	
	1	Время жизни пакета истекло при сборке фрагментов	
12	Неверный параметр (проблема с параметрами дейтаграммы: ошибка в IP-заголовке или отсутствует необходимая опция)		Указатель (8) Не используется (24) Заголовок IP, Начало исходной дейтаграммы (64)
	0	Указатель говорит об ошибке	
	1	Отсутствует требуемая опция	
	2	Некорректная длина	
13	0	Запрос метки времени	Идентификатор (16)

Тип	Код	Сообщение	Данные (длина, бит)
14	0	Ответ с меткой времени	Номер последовательности (16) Начальное время (32) Время приёма (32) Время отправки (32)
15	0	Информационный запрос	Идентификатор (16)
16	0	Информационный ответ	Номер последовательности (16)
17	0	Запрос адресной маски	Идентификатор (16)
18	0	Отклик на запрос адресной маски	Номер последовательности (16) Маска (32)
19		Зарезервировано (для обеспечения безопасности)	
20— 29		Зарезервировано (для экспериментов на устойчивость к ошибкам)	
30	Трассировка маршрута		Идентификатор (16)
	0	Исходящий пакет успешно отправлен	Не используется (16)
	1	Путь для исходящего пакета не найден, пакет уничтожен	Количество хопов исходящего пакета (16) Количество хопов возвращающегося пакета (16) Скорость линии связи (32) MTU линии связи (32)
31	Ошибка преобразования датаграммы		Указатель (32)
	0	Неизвестная или неуказанная ошибка	
	1	Невозможно конвертировать опцию	
	2	Неизвестная обязательная опция	
	3	Неподдерживаемая обязательная опция	

Тип	Код	Сообщение	Данные (длина, бит)
	4	Неподдеживаемый транспортный протокол	
	5	Превышена полная длина	
	6	Превышена длина заголовка IP	
	7	Номер транспортного протокола больше 255	
	8	Номер порта вне допустимого диапазона	
	9	Превышена длина заголовка транспортного протокола	
	10	Переход через границу 32 бит и установлен бит ASK	
	11	Неизвестная обязательная опция транспортного протокола	
32		Перенаправление для мобильного узла	
33		IPv6 Where-Are-You (где вы находитесь)	
34		IPv6 I-Am-Here (я здесь)	
35		Запрос перенаправления для мобильного узла	
36		Отклик на запрос перенаправления для мобильного узла	
37		Запрос доменного имени	
38		Ответ на запрос доменного имени	
39		SKIP	
40	Photuris		
	0	Зарезервировано	
	1	Неизвестный индекс параметров безопасности	

Тип	Код	Сообщение	Данные (длина, бит)
	2	Параметры безопасности верны, но произошла ошибка аутентификации	
	3	Параметры безопасности верны, но произошел сбой при расшифровке	
	4	Требуется проверка подлинности	
	5	Требуется авторизация	
41– 255		Зарезервировано	