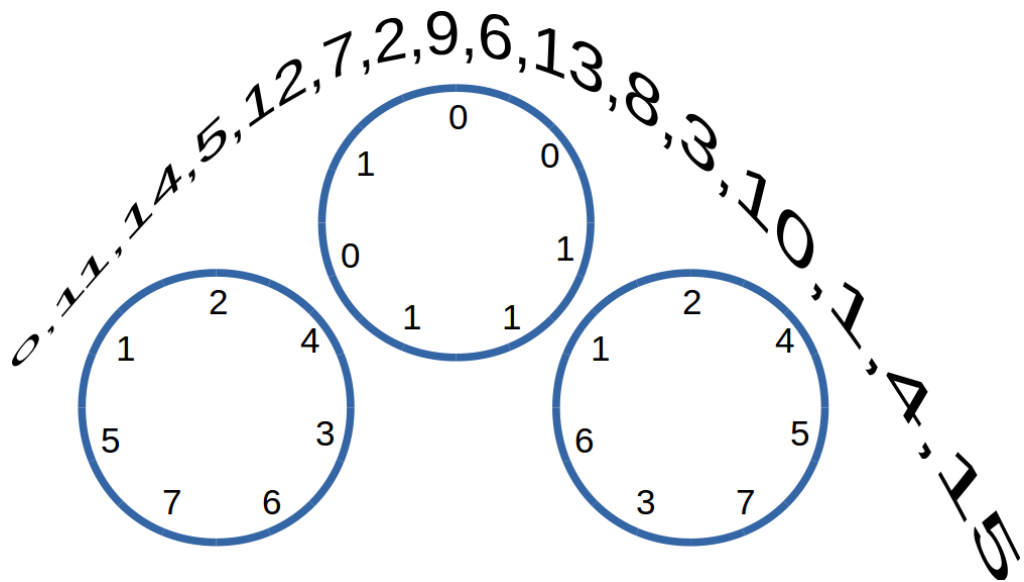


А. С. Бернгардт, А. В. Новиков

ТЕОРИЯ И ТЕХНИКА ПЕРЕДАЧИ ИНФОРМАЦИИ

Методические указания по проведению лабораторных работ
и организации самостоятельной работы студентов



ТОМСК
2015

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ**

Кафедра радиотехнических систем

А. С. Бернгардт, А. В. Новиков

ТЕОРИЯ И ТЕХНИКА ПЕРЕДАЧИ ИНФОРМАЦИИ

**Методические указания по проведению лабораторных работ
и организации самостоятельной работы студентов**

2015

Бернгардт А. С., Новиков А. В.

Теория и техника передачи информации: методические указания по проведению лабораторных работ и организации самостоятельной работы студентов. — Томск: ТУСУР, 2015. — 47 с.

Методические указания по проведению лабораторных работ и организации самостоятельной работы студентов по дисциплинам систем связи для студентов очной формы обучения.

Лабораторный практикум включает две компьютерные лабораторные работы, выполнение которых направлено на освоение методов спектрального представления цифровых сигналов, а также на изучение циклических кодов.

Методические указания содержат описание лабораторных макетов, рекомендуемый порядок проведения исследований, требования к отчету и набор контрольных вопросов.

Для каждой лабораторной работы даны сведения из теории, достаточные для выполнения исследований без обращения к другой учебной литературе.

Одобрено кафедрой радиотехнических систем ТУСУРа в качестве учебного методического пособия для проведения лабораторных работ и организации самостоятельной работы студентов.

© Бернгардт А. С. , Новиков А. В., 2015

© Кафедра радиотехнических систем,
ТУСУР, 2015

СОДЕРЖАНИЕ

Предисловие	4
1 Лабораторная работа № 1. Временное и спектральное представление цифровых сигналов	5
1.1 Введение	5
1.2 Сведения из теории	6
1.2.1 Почему используют гармонические колебания	6
1.2.2 Ряды Фурье для периодических сигналов	8
1.3 Модели сигналов, исследуемых в лабораторной работе.....	11
1.4 Описание лабораторного макета.....	19
1.5 Порядок выполнения работы	20
1.5.1 Самостоятельная теоретическая работа и необходимые расчеты	20
1.5.2 Лабораторное задание	21
1.5.3 Требования к отчету	21
1.5.4 Вопросы для самопроверки	22
1.6 Литература.....	22
2 Лабораторная работа № 2. Циклические коды	23
2.1 Введение	23
2.2 Сведения из теории	27
2.3 Примеры кодирования и декодирования	34
2.4 Описание лабораторного макета.....	37
2.5 Порядок выполнения работы	40
2.5.1 Расчётное задание	40
2.5.2 Анализ результатов выполнения расчётного задания	42
2.5.3 Экспериментальная часть	43
2.6 Источники	46
Приложение А Варианты заданий лабораторной работы № 1	47

ПРЕДИСЛОВИЕ

Большую роль при обучении играет зрительный аппарат человека, а также обратная связь, когда человек изменяет параметр (например, крутит ручку прибора) и следом видит результат этого изменения (например, на экране осциллографа).

При самостоятельном обучении (дома, в библиотеке...) нет возможности передать студенту то «железо», которое установлено в учебных лабораториях, поэтому важная роль отводится лабораторным макетам — программам, запускаемым на персональном компьютере. Однако в мире программ имеется специфическая проблема: лицензирование продукта, поэтому особая роль отводится **open-source** программам (с открытым исходным кодом). Такие программы не требуют лицензий и принадлежат всему **open-source** сообществу.

Компьютерная лабораторная работа № 1 («Спектры») разработана доцентом кафедры РТС А.С. Бернгардтом совместно со студентом группы 121-2 П.Я. Ширяевым. Компьютерная лабораторная работа № 2 («Циклические коды») разработана доцентом кафедры РТС А.В. Новиковым. Общее редактирование представленного варианта цикла лабораторных работ осуществлено доцентом кафедры РТС А.С. Бернгардтом.

Выбор варианта лабораторных работ осуществляется по общим правилам с использованием формулы:

$$V = (N \times K) \operatorname{div} 100,$$

где V — искомый номер варианта,

N — общее количество вариантов,

div — целочисленное деление,

при $V=0$ выбирается максимальный вариант,

K — код варианта.

1 ЛАБОРАТОРНАЯ РАБОТА № 1. ВРЕМЕННОЕ И СПЕКТРАЛЬНОЕ ПРЕДСТАВЛЕНИЕ ЦИФРОВЫХ СИГНАЛОВ

1.1 Введение

Для описания сигналов — переносчиков информации, наряду с временным, очень широко используют спектральное (частотное) представление. Временным является представление сигнала как некоторой действительной функции времени $s(t)$. Спектральным является представление сигнала $s(t)$ в виде суммы гармонических (синусоидальных) колебаний. При этом зависимости амплитуд и фаз гармонических колебаний от частоты называют амплитудным и фазовым спектрами сигнала. На спектральном языке описывают не только сигналы, но и свойства различных радиотехнических устройств, в том числе и устройств передачи информации.

В то же время опыт показывает, что спектральное представление и описание сигналов остается для многих студентов формальным математическим приемом, они могут писать формулы, но не понимают причины и смысл разложения сигнала по гармоническим колебаниям.

При создании данной компьютерной лабораторной работы была поставлена задача максимально просто и наглядно пояснить физический смысл преобразования Фурье, взаимосвязь временного и спектрального представлений сигналов.

Целью лабораторной работы является закрепление имеющихся знаний в виде умений и навыков.

Технические требования к оборудованию при выполнении работы:

- Персональный компьютер с возможностью запуска программного макета в виде исполняемого файла, предназначенного для операционной системы **Windows**;

- Офисный пакет для подготовки отчёта (принимаются файлы с расширением **odt**, **doc** и **docx**).

1.2 Сведения из теории

1.2.1 Почему используют гармонические колебания

Чтобы понять особую роль гармонических колебаний в радиотехнике, рассмотрим некоторые понятия, которые нам еще пригодятся.

Любая радиотехническая система обычно состоит из комбинации *линейных и нелинейных* устройств. *К линейным* устройствам можно отнести *усилители, фильтры, длинные линии и т. д.* *К нелинейным* относят *смесители, модуляторы, ограничители, триггеры, умножители, автоколебательные системы.*

Устройство называется *линейным*, если выполняется *принцип суперпозиции*, который утверждает, что *полный отклик* линейного устройства $y(t)$ на сумму входных сигналов $s(t)$ *равен сумме откликов* $y_k(t)$ на каждый сигнал $s_k(t)$. В противном случае устройство будет *нелинейным*. Таким образом, для линейного устройства при входном воздействии

$$s(t) = \sum_{i=1}^n K_i s_i(t) \quad (1.1)$$

отклик имеет вид

$$y(t) = \sum_{i=1}^n K_i y_i(t), \quad (1.2)$$

где y_i — отклик на входное воздействие $s_i(t)$, K_i — действительное число.

Важно то, что для *линейной* системы параметры самой системы *не зависят* от входного воздействия, а для *нелинейной* системы — *зависят*. Например, коэффициент передачи ограничителя, порогового устройства или нелинейного усилителя зависит от уровня сигнала.

Для нелинейной системы принцип суперпозиции не выполняется, и её отклик содержит различные комбинации откликов.

К чисто линейным устройствам приходят, как правило, в результате *упрощений*, допустимых лишь при определенных условиях. Так, например, усилитель можно отнести к числу линейных устройств лишь в том случае, если входной сигнал располагается в пределах линейного участка в общем случае нелинейной вольтамперной характеристики активного элемента, на котором построен усилитель. Все мы сталкивались с таким явлением, увеличивая громкость звучания акустической системы до появления «хрипов», то есть звуков, которых не было во входном сигнале.

Почему же при описании и исследовании линейных систем используют разложение сигналов на входе и на выходе системы по *гармоническим* колебаниям? Дело в том, что задача определения отклика линейной системы решается наиболее просто, если он представляет собой входное воздействие, смещенное во времени и умноженное на постоянный множитель (константу), то есть $y(t) = K \cdot s(t + \Delta t)$. Временные функции, описывающие такие воздействия, называются *собственными*, а выражение (1.2) для отклика линейной системы приобретает наиболее простой вид

$$y(t) = \sum_{K=1}^n K_i s_i(t + \Delta t_i). \quad (1.3)$$

Оказывается, что для *линейных* систем *собственными* или *инвариантными* функциями являются как раз *гармонические* (синусоидальные) колебания $s(t) = \cos(2\pi ft)$. Следовательно, линейное устройство состоит из элементов, выполняющих некоторые или все из следующих операций: дифференцирование, интегрирование, умножение на постоянную величину, задержка и суммирование результатов указанных операций. При таких преобразованиях форма гармонического колебания не изменяется, изменяется только его амплитуда и сдвиг по времени (начальная фаза).

В первой четверти XIX века французский математик Жан Батист Фурье показал, что любую периодическую функцию с некоторыми огра-

ничениями, указанными ниже, можно представить в виде суммы гармонических колебаний с определенными амплитудами и фазами

$$s(t) = \sum_{k=1}^n A_k \cos(2\pi f_k t + \varphi_k). \quad (1.4)$$

Это позволяет, с учетом принципа инвариантности, анализировать линейные преобразования сигналов в три этапа: 1) представить сигнал в виде комбинации гармонических колебаний (синусоид); 2) определить амплитуды и фазы гармоник после преобразования; 3) просуммировать гармоники для получения результирующего сигнала. В большинстве случаев этот путь является наиболее простым и наглядным.

В следующем параграфе приведены минимально необходимые сведения из теории рядов Фурье [1, с. 34].

1.2.2 Ряды Фурье для периодических сигналов

Сигнал будет периодическим, если выполняется условие

$$s(t) = s(t + nT), \quad (2.1)$$

где T — постоянная величина, называемая периодом, а n — любое положительное или отрицательное целое число.

Если сигнал $s(t)$ с периодом T имеет на этом периоде не более конечного числа точек разрыва и абсолютно интегрируем, то есть

$$\int_{-T}^{+T} |f(t)| dt < \infty,$$

тогда его можно представить рядом Фурье по системе ортогональных тригонометрических функций кратных частот (гармоник)

$$s(t) = a_0 + \sum_{k=1}^{\infty} A_k \cos(2\pi f_k t + \varphi_k), \text{ где } f_k = k \frac{1}{T}. \quad (2.2)$$

Здесь $f_1 = 1/T$ — частота *первой гармоники*, или *основная частота*. Частоту измеряют в герцах, это число периодов в секунду. *Основная частота* f_1 совпадает с частотой повторения сигнала. Частоты всех составляющих кратны основной частоте. Совокупность амплитуд $\{A_k\}$, фаз $\{\varphi_k\}$

и частот $\{f_k\}$ суммируемых гармоник называют соответственно *спектром амплитуд*, *спектром фаз* и *спектром частот*.

Последнее выражение может быть выражено в другой, часто используемой форме, через косинусную a_i и синусную b_i (*квадратурные*) составляющие гармонического колебания

$$s(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos(2\pi f_k t) + b_k \sin(2\pi f_k t)), \quad (2.3)$$

причем амплитуды, фазы *гармоник* и амплитуды *квадратурных* составляющих связаны соотношениями

$$a_k = A_k \cos \varphi_k, \quad b_k = A_k \sin \varphi_k, \quad (2.4)$$

$$A_k = \sqrt{a_k^2 + b_k^2}, \quad \operatorname{tg} \varphi_k = -\frac{b_k}{a_k}. \quad (2.5)$$

Коэффициенты a_k и b_k определяются по формулам:

$$a_k = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \cos(2\pi f_k t) dt, \quad k = 1, 2, \dots \quad (2.6)$$

$$b_k = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \sin(2\pi f_k t) dt, \quad i = 1, 2, \dots \quad (2.7)$$

Величина a_0 определяет среднее значение сигнала за период, называется постоянной составляющей и вычисляется по формуле:

$$a_0 = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) dt. \quad (2.8)$$

Обратите внимание, что для *четных функций* $s(t) = s(-t)$ все $b_k = 0$ и ряд Фурье содержит только косинусы, следовательно,

$$s(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(2\pi f_k t). \quad (2.9)$$

$$A_k = |a_k|, \quad \varphi_k = \begin{cases} 0, & a_k \geq 0, \\ \pi, & a_k < 0. \end{cases} \quad (2.10)$$

Если сигнал $s(t) = -s(-t)$ — *нечётная функция*, все $a_k = 0$, ряд Фурье содержит только синусы, следовательно,

$$s(t) = a_0 + \sum_{k=1}^{\infty} b_k \sin(2\pi f_k t), \quad (2.11)$$

$$A_k = |b_k|, \quad \varphi_k = \begin{cases} \frac{\pi}{2}, & b_k \geq 0, \\ -\frac{\pi}{2}, & b_k < 0. \end{cases} \quad (2.12)$$

Сигнал $s(t)$ общего вида всегда можно представить в виде суммы чётной и нечётной составляющих, то есть $s(t) = s_{\text{чет}}(t) + s_{\text{неч}}(t)$.

Отметим некоторые важные и полезные свойства ряда Фурье:

1) Изменение величины сигнала

Если $s_1(t) = c \cdot s(t)$, $c > 0$, тогда $A_{1k} = cA_k$, $a_{1k} = ca_k$, $b_{1k} = cb_k$, $\varphi_{1k} = \varphi_k$.

При изменении величины сигнала в c раз спектр амплитуд также изменяется в c раз, спектр фаз не изменяется.

2) Умножение сигнала на (-1)

Если $s_1(t) = -s(t)$, тогда $A_{1k} = A_k$, $a_{1k} = -a_k$, $b_{1k} = -b_k$, $\varphi_{1k} = \varphi_k + \pi$.

Следовательно, квадратурные составляющие изменяют свой знак на противоположный. При этом все фазы изменяются на величину π (180°).

3) Сдвиг сигнала во времени на величину Δt

Используя (1.4), получим

$$s(t + \Delta t) = \sum_{i=1}^n A_k \cos(2\pi f_k (t + \Delta t) + \varphi_k) = \sum_{i=1}^n A_k \cos(2\pi f_k t + \varphi_k + \psi_k), \quad (2.13)$$

$$\text{где } \psi_k = 2\pi \Delta t f_k = k \frac{\Delta t}{T} 2\pi.$$

Видим, что спектр амплитуд не изменяется, но в спектре фаз появляется дополнительный фазовый сдвиг, пропорциональный номеру гармоники и величине временного сдвига.

4) Линейность преобразования Фурье

Для сигнала $s_p(t) = c_1 s_1(t) + c_2 s_2(t)$

$$a_{pk} = c_1 a_{1k} + c_2 a_{2k}, \quad b_{pk} = c_1 b_{1k} + c_2 b_{2k}. \quad (2.14)$$

То есть линейное преобразование сигнала приводит к аналогичному преобразованию амплитуд квадратурных составляющих.

Важно то, что разложение в ряд Фурье позволяет исследовать спектральные свойства не только периодических, но и исходных непериодических сигналов. Действительно, исходный непериодический сигнал $s(t)$ можно получить из периодического $s_n(t)$ в результате предельного перехода при $T \rightarrow \infty$, при этом спектр становится непрерывным, но его форма (структура) сохраняется.

1.3 Модели сигналов, исследуемых в лабораторной работе

Базовой моделью цифрового сигнала является последовательность прямоугольных импульсов.

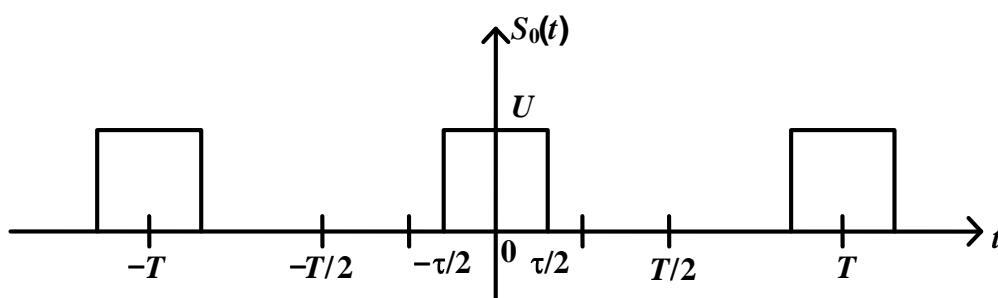


Рис. 1.1 — Периодическая последовательность прямоугольных импульсов

Аналитическое выражение для одного периода прямоугольного импульса на симметричном отрезке $[-T/2, T/2]$ имеет вид

$$s_0(t) = \begin{cases} u, & |t| \leq \frac{\tau}{2}, \\ 0, & \frac{\tau}{2} < |t| \leq \frac{T}{2}, \end{cases} \quad (3.1)$$

где u — амплитуда, τ — длительность импульса.

Это — *чётная функция*, поэтому $b_k = 0$, а постоянная составляющая и косинусная составляющая принимают значения

$$a_{00} = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) dt, = \frac{u\tau}{T} = \frac{u}{g}, \quad (3.2)$$

где $g = \frac{T}{\tau}$ — скважность периодической последовательности импульсов.

$$a_{0k} = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \cos(2\pi f_k t) dt = \frac{2u\tau}{T} \frac{\sin \pi f_k \tau}{\pi f_k \tau}, \quad f_k \tau = k \frac{\tau}{T} = \frac{k}{g}. \quad (3.3)$$

Обратите внимание, что в работе принята следующая нумерация коэффициентов ряда Фурье: первый индекс определяет номер исследуемого сигнала, а второй — номер гармоники. Кроме того, мы будем использовать нормированные на амплитуду первой гармоники A_{01} значения коэффициентов ряда Фурье. Например, обозначение A_{nk} означает нормированное значение амплитуды k -й гармоники n -го сигнала, φ_{nk} — соответствующее значение фазы.

$$a_{00}^i = \frac{\pi\tau}{2T} \frac{1}{\sin \pi \frac{\tau}{T}}, \quad (3.4)$$

$$a_{0k}^i = \frac{a_k}{a_1} = \frac{1}{k} \frac{\sin \pi k \frac{\tau}{T}}{\sin \pi \frac{\tau}{T}}. \quad (3.5)$$

Сигнал 1. Длительность импульса $\tau = T/2$, скважность $g = 2$.

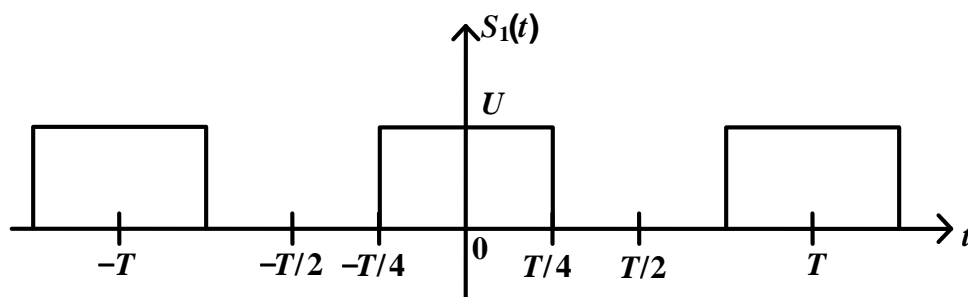


Рис. 1.2 — График сигнала $s_1(t)$

Запишем аналитическое выражение

$$s_1(t) = \begin{cases} u, & |t| \leq \frac{T}{4}, \\ 0, & \frac{T}{4} < |t| \leq \frac{T}{2}. \end{cases}$$

В данном случае

$$\sin \pi \frac{\tau}{T} = \sin \frac{\pi}{2} = 1,$$

и коэффициенты ряда Фурье принимают следующие значения:

$$a_{10}^H = \frac{\pi}{4} = 0,785, \quad a_{1k}^H = \frac{1}{k} \sin \frac{k\pi}{2} = \begin{cases} (-1)^{m-1}, & k = 2m - 1, \\ 0, & k = 2m, m = 1, 2, \dots \end{cases}$$

$$A_{1k} = |a_{1k}^H|, \quad \varphi_{1k} = \begin{cases} \pi, & k = 4m - 1, \quad m = 1, 2, \dots, \\ 0, & \text{другие } k. \end{cases}$$

При выполнении лабораторной работы фазу удобнее задавать в градусах, $\varphi^\circ = \frac{\varphi}{\pi} 180^\circ$.

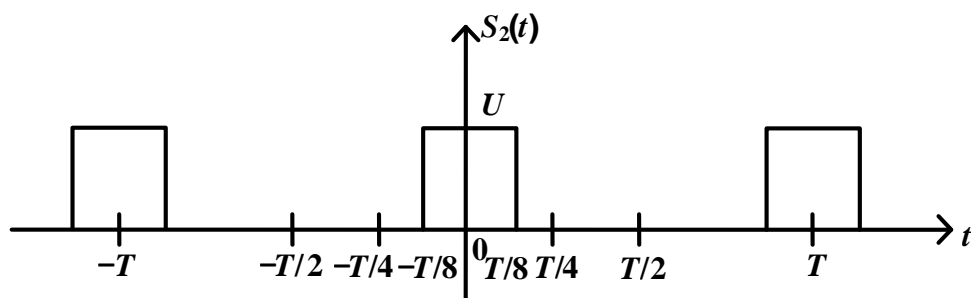
В таблице 1 приведены значения коэффициентов ряда Фурье для сигнала $s_1(t)$.

Таблица 1

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a_k	1		-0.33	0	0.20		-0.14	0	0	0	-0.09	0	0.08	0	-0.06	0
A_{1k}	1	0	0.33	0	0.20	0	0.14	0	0.11	0	0.09	0	0.08	0	0.06	0
φ_{1k}°	0	0	180	0	0	0	180	0	0	0	180	0	0	0	180	0

Сигнал 2. Длительность импульса $\tau = T/4$, скважность $g = 4$.

$$s_2(t) = \begin{cases} u, & |t| \leq \frac{T}{8}, \\ 0, & \frac{T}{8} < |t| \leq \frac{T}{2}. \end{cases}$$

Рис. 1.3 — График сигнала $s_2(t)$

В данном случае

$$\sin \pi \frac{\tau}{T} = \sin \frac{\pi}{4} = \frac{1}{\sqrt{2}},$$

и коэффициенты ряда Фурье принимают следующие значения:

$$a_{20}^i = \frac{\pi}{8} \frac{1}{\sin \frac{\pi}{4}} = \frac{\pi}{8} \sqrt{2} = 0.55, \quad a_{2k}^i = \frac{1}{k} \frac{\sin \frac{k\pi}{4}}{\sin \frac{\pi}{4}} = \frac{\sqrt{2}}{k} \sin \frac{k\pi}{4},$$

$$A_{2k} = |a_{2k}^i|, \quad \varphi_{2k} = \begin{cases} \pi, & a_{2k}^i \leq 0, \\ 0, & a_{2k}^i > 0. \end{cases}$$

В таблице 2 приведены значения коэффициентов ряда Фурье для сигнала $s_2(t)$.

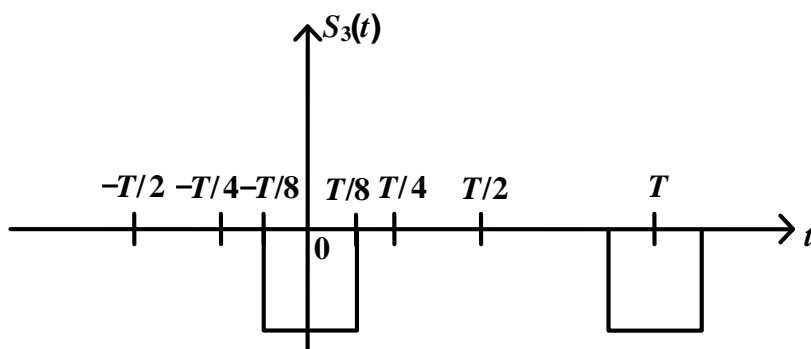
Таблица 2

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a_k	1	0.71	0.33	0	-0.20	-0.24	-0.14	0	0.11	0.14	0.09	0	-0.08	-0.1	-0.06	0
A_{2k}	1	0.71	0.33	0	0.20	0.24	0.14	0	0.11	0	0.09	0	0.08	0.1	0.06	0
φ_{2k}°	0	0	0	0	180	180	180	0	0	0	0	0	180	180	180	0

Сигнал 3. Сигнал $s_3(t) = (-1)s_2(t)$.

Длительность импульса $\tau = T/4$, скважность $g = 4$.

$$s_3(t) = \begin{cases} -u, & |t| \leq \frac{T}{8}, \\ 0, & \frac{T}{8} < |t| \leq \frac{T}{2}. \end{cases}$$

Рис. 1.4 — График сигнала $s_3(t)$

В этом случае $a_{30}^i = a_{20}^i = -0.55$, $A_{3k} = A_{2k}$, $\varphi_{3k} = \varphi_{2k} + \pi$.

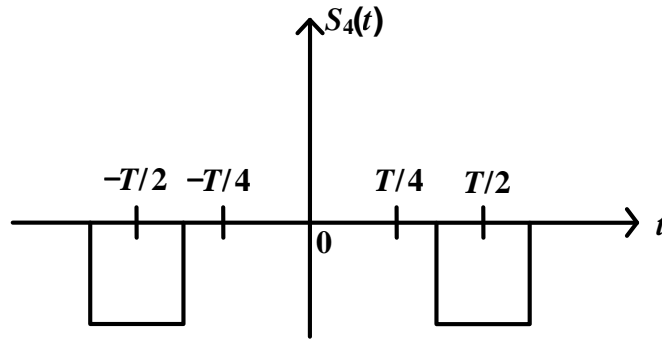
В таблице 3 приведены значения коэффициентов ряда Фурье сигнала $s_3(t)$.

Таблица 3

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a_k	-1	-0.71	-0.33	0	0.20	0.24	0.14	0	-0.11	-0.14	-0.09	0	0.08	0.1	0.06	0
A_{3k}	1	0.71	0.33	0	0.20	0.24	0.14	0	0.11	0	0.09	0	0.08	0.1	0.06	0
φ_{3k}°	180	180	180	0	0	0	0	0	180	180	180	0	0	0	0	0

Сигнал 4. Сдвиг сигнала $s_3(t)$ на величину $\Delta t = T/2$, $s_4(t) = s_3(t + T/2)$.

$$s_4(t) = \begin{cases} 0, & |t| \leq \frac{3T}{8}, \\ -u, & \frac{3T}{8} < |t| \leq \frac{T}{2}. \end{cases}$$

Рис. 1.5 — График сигнала $s_4(t)$

Используя выражение (2.13), получим

$$a_{40}^i = a_{30}^i = -0.55, \quad A_{4k} = A_{3k}, \quad \varphi_{4k} = \varphi_{3k} + k\pi.$$

В таблице 4 приведены значения коэффициентов ряда Фурье сигнала $s_4(t)$.

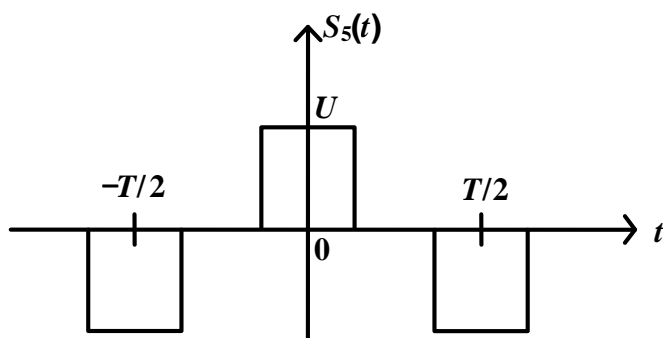
Таблица 4

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a_k	1	-0.71	0.33	0	-0.20	0.24	-0.14	0	0.11	-0.14	0.09	0	-0.08	0.1	-0.06	0
A_{4k}	1	0.71	0.33	0	0.20	0.24	0.14	0	0.11	0	0.09	0	0.08	0.1	0.06	0
φ_{4k}°	0	180	0	0	180	0	180	0	0	180	0	0	180	0	180	0

Сигнал 5. Сумма сигналов

$$s_5(t) = \frac{1}{2} \left[s_2(t) - s_2\left(t + \frac{T}{2}\right) \right] = \frac{1}{2} s_2(t) + s_4(t) .$$

$$s_5(t) = \begin{cases} \frac{u}{2}, & |t| \leq \frac{T}{8}, \\ 0, & \frac{T}{8} < |t| < \frac{3T}{8}, \\ -\frac{u}{2}, & \frac{3T}{8} \leq |t| \leq \frac{T}{2}. \end{cases}$$

Рис. 1.6 — График сигнала $s_5(t)$

Используя выражение (2.1), таблицу 2 и таблицу 4, получим

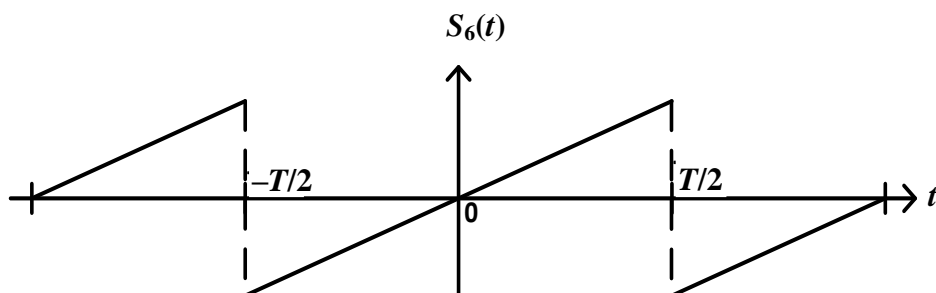
$$a_{50}^i = 0, \quad A_{5k} = \begin{cases} A_{4k}, & k = 2m - 1, \\ 0, & k = 2m, \quad m = 1, 2, \dots \end{cases}$$

В таблице 5 приведены значения коэффициентов ряда Фурье сигнала $s_5(t)$.

Таблица 5

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A_{5k}	1	0	0.33	0	0.20	0	0.14	0	0.11	0	0.09	0	0.08	0	0.06	0
φ_{5k}°	0	0	0	0	180	0	180	0	0	0	0	0	180	0	180	0

Сигнал 6. $s_6(t) = ut, \quad -\frac{T}{2} \leq t \leq \frac{T}{2}$.

Рис. 1.7 — График сигнала $s_6(t)$

Это — нечётная функция, коэффициенты ряда Фурье принимают следующие значения

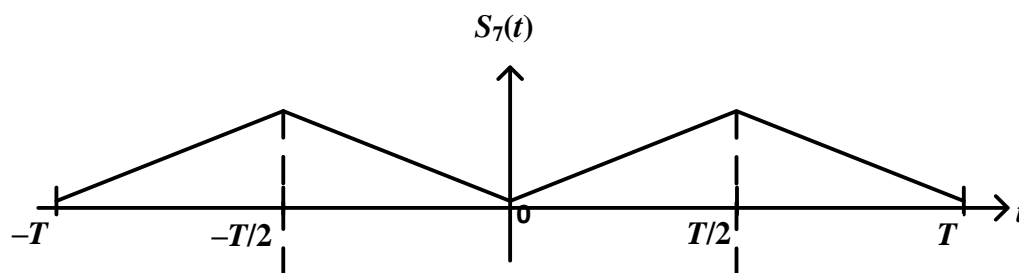
$$a_{60}^i = 0, a_{6k}^i = 0, b_{6k}^i = \frac{1}{k}(-1)^{2k+1}, A_{6k}^i = |b_{6k}^i|, \varphi_{6k}^i = \frac{\pi}{2} + k\pi, \quad k = 1, 2, \dots$$

В таблице 6 приведены значения коэффициентов ряда Фурье сигнала $s_6(t)$.

Таблица 6

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A_{6k}	1	0.5	0.33	0.25	0.20	0.17	0.14	0.12	0.11	0.1	0.09	0.08	0.08	0.07	0.06	0.06
φ_{6k}°	-90	90	-90	90	-90	90	-90	90	-90	90	-90	90	-90	90	-90	90

Сигнал 7. $s_7(t) = u|t|, -\frac{T}{2} \leq t \leq \frac{T}{2}$.

Рис. 1.8 — График сигнала $s_7(t)$

Это — чётная функция, $b_k = 0$, коэффициенты ряда Фурье принимают следующие значения:

$$a_{70}^i = \frac{\pi^2}{8} = 1.23, \quad b_{7k}^i = \begin{cases} -\frac{1}{k^2}, & k = 2m-1, \\ 0, & k = 2m, \end{cases} \quad A_{7k}^i = |b_{7k}^i| = \frac{1}{k^2}, \quad \varphi_{7k}^i = \pi.$$

В таблице 7 приведены значения коэффициентов ряда Фурье сигнала $s_7(t)$.

Таблица 7

k	1	2	3	4	5	6	7	8	9	10
A_{7k}	1	0	0.11	0	0.04	0	0.02	0	0.01	0
φ_{7k}°	180	0	180	0	180	0	180	0	180	0

Сигнал 8. Это сигнал в виде функции Котельникова

$$s_8(t) = \frac{\sin(\pi f_{\max} t)}{\pi f_{\max} t}, \quad -\frac{T}{2} \leq t \leq \frac{T}{2}.$$

Здесь $f_{\max} = k_{\max} f_1$, k_{\max} — число суммируемых гармоник.

Это — чётная функция, $b_k = 0$, коэффициенты ряда Фурье принимают следующие значения:

$$a_{80}^i = 0, \quad a_{8k}^i = \begin{cases} 1, & k \leq k_{\max}, \\ 0, & k > k_{\max} \end{cases}, \quad A_{8k}^i = |b_{8k}^i| = 1, \quad \varphi_{8k}^i = 0.$$

1.4 Описание лабораторного макета

Виртуальный лабораторный макет выполнен в среде программирования *Delphi* и представляет собой исполняемый файл **Спектр.exe**, находящийся в папке **./windows** и предназначенный для запуска в операционной системе **Windows**.

В макете реализована возможность изучения взаимосвязи между временными и спектральными характеристиками сигналов.

Временные реализации исследуемых сигналов (график в верхней части макета) формируются суммированием гармоник кратных частот. Амплитуды гармоник задаются с помощью ряда вертикальных ползунков, значения фаз в градусах задаются в окнах, находящихся под ними. Ряд горизонтальных ползунков справа позволяет изменять постоянную составляющую, масштаб графика временной реализации и число периодов на интервале наблюдения. Значение постоянной составляющей можно задать в окне рядом с ползунком. Флажок в окне «Показать Спектр» позволяет получить графики спектров амплитуд и фаз в привычной форме. Кнопки «Сохранить» и «Показать» позволяют сохранить необходимые графики и использовать их для сравнения или написания отчета.

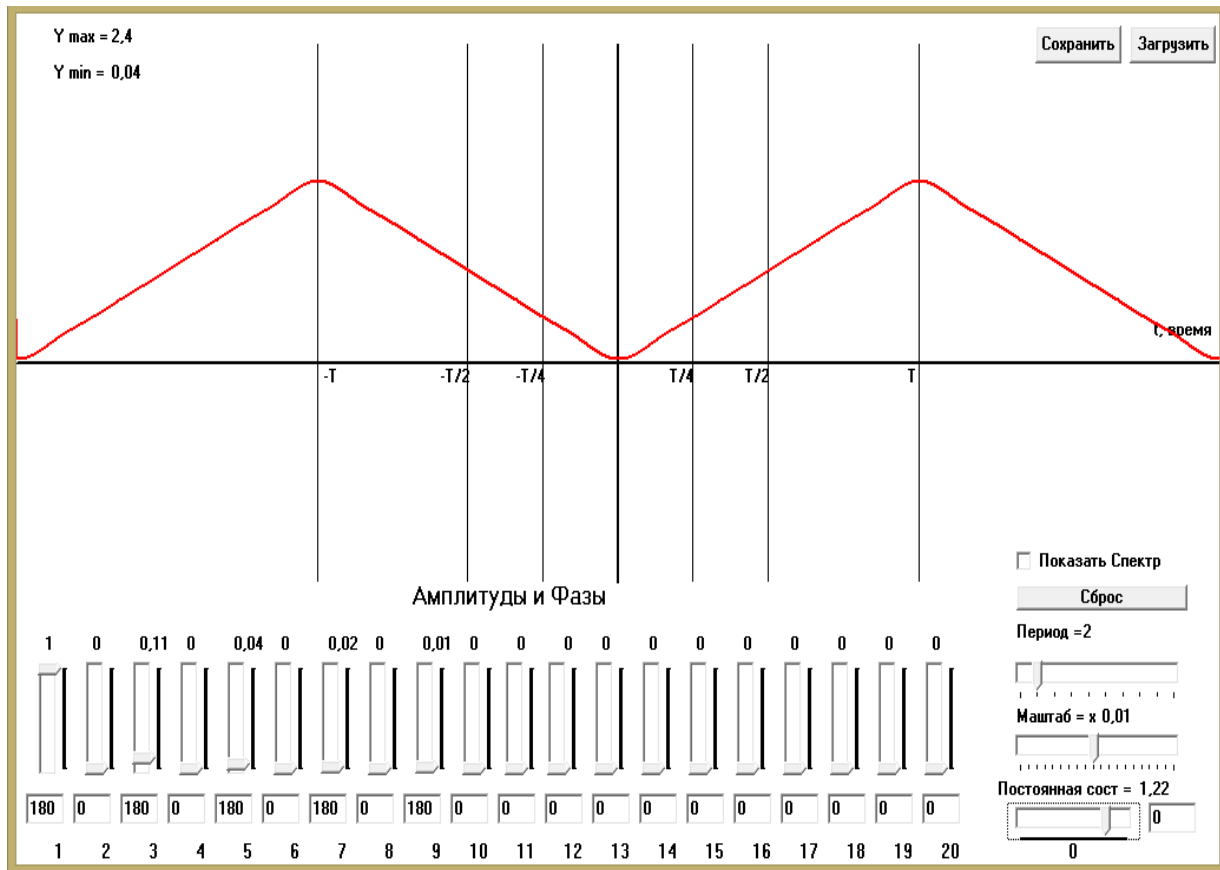


Рис. 1.9 — Окно программного лабораторного макета
для изучения спектров цифровых сигналов

1.5 Порядок выполнения работы

1.5.1 Самостоятельная теоретическая работа и необходимые расчеты

1) Проработать теоретический материал и рекомендованную литературу по теме лабораторной работы.

2) Изучить модели сигналов, исследуемых в лабораторной работе, рассмотренные в разделе 3.

3) Для выбранного варианта лабораторной работы (варианты заданий представлены в **Приложении А**), т.е. варианта сигнала $s_k(t)$ и величины временного сдвига Δt , произвести расчет спектра амплитуд и фаз, результаты расчетов оформить в виде таблицы.

1.5.2 Лабораторное задание

При выполнении работы необходимо:

1) Исследовать зависимости формы спектра периодической последовательности с постоянным периодом повторения от длительности импульса на примере сигналов $s_1(t)$, $s_2(t)$, $s_8(t)$. Для последнего сигнала длительность определять по координате первого нуля временной реализации. Зафиксировать и представить наиболее характерные временные и спектральные зависимости.

2) Исследовать зависимость формы сигнала от числа суммируемых гармоник на примере сигналов $s_2(t)$ и сигнала Вашего варианта. Зафиксировать и представить наиболее характерные временные и спектральные зависимости.

3) На примере сигналов $s_2(t) - s_5(t)$ проследить изменения амплитудного и фазового спектров при умножении сигналов на число, положительное или отрицательное, сдвиге во времени и суммировании. Зафиксировать и представить наиболее характерные временные и спектральные зависимости.

4) Сравнить временные реализации и соответствующие им спектры амплитуд и фаз сигналов $s_2(t)$, $s_6(t)$, $s_7(t)$ и $s_8(t)$. Объяснить различия в скорости убывания спектра амплитуд. Зафиксировать и представить наиболее характерные временные и спектральные зависимости.

1.5.3 Требования к отчету

Отчёт должен содержать:

- Титульный лист.
- Расчётное задание.
- Анализ расчётного задания.
- Экспериментальную часть.

- Выводы.
- Ответы на вопросы для самопроверки.

1.5.4 Вопросы для самопроверки

- 1) Какими свойствами обладают спектры периодических сигналов?
- 2) Как влияет изменение периода повторения при постоянной длительности импульса на спектр периодической последовательности прямоугольных видеоимпульсов?
- 3) Как отразится на спектре периодического сигнала изменение положения начала отсчета времени?
- 4) Как влияет изменение длительности импульса при постоянном периоде повторения на спектр периодической последовательности прямоугольных видеоимпульсов?
- 5) Как изменится спектр периодического сигнала, если период повторения устремить в бесконечность?
- 6) Как связаны между собой длительности импульса и ширина спектра?
- 7) Какое влияние оказывает уменьшение числа суммируемых гармоник на положение и крутизну фронта прямоугольного импульса?
- 8) Какое влияние оказывает изменение знака сигнала на спектр амплитуд?
- 9) Какое влияние оказывает изменение знака сигнала на спектр фаз?

1.6 Литература

1. Каратаева Н.А. Радиотехнические цепи и сигналы: учеб. пособие / Н.А. Каратаева. — Томск: Томский межвузовский центр дистанционного образования, 2012. — Ч.1. — 260 с.
2. Акулиничев Ю. П. Теория и техника передачи информации : учеб. пособие / Ю.П. Акулиничев, А.С. Бернгардт. — Томск: Эль Контент, 2012. — 210 с.

2 ЛАБОРАТОРНАЯ РАБОТА № 2. ЦИКЛИЧЕСКИЕ КОДЫ

2.1 Введение

Тема работы относится к разделу **кодирование канала**.

В современных цифровых системах передачи информации кодирование канала, как правило, используется совместно с **кодированием источника**, несмотря на то, что они как «лебедь, рак и щука» тянут в разные стороны. Канальное кодирование направлено на получение выигрыша в помехоустойчивости, что достигается **искусственной избыточностью**, которая снижает скорость передачи информации; кодирование источника направлено на уменьшение **естественной избыточности**, что даёт плюс к скорости передачи информации и минус к помехоустойчивости. Примерами кодирования источника являются, например, универсальные архиваторы файлов **zip** или **rar**, звуковые кодеки типа **mp3** или **flac**, видеокодеки типа **mp4** или **HuffYUV** и тому подобное.

Естественная избыточность, содержащаяся практически в любой информации, в общем случае не может быть использована для защиты от помех, так как для технического устройства она носит случайный характер. Например, сумму в 12000 рублей можно записать **многими, заранее не предсказуемыми** способами:

12000 руб.

Двенадцать тысяч рублей 00 копеек

12 тыс. руб. 00 к.

и так далее, и тому подобное. Поэтому кодированием источника эту избыточность уменьшают, после чего добавляют искусственную избыточность для обеспечения требуемой помехоустойчивости. Естественной избыточностью может пользоваться только обученный языку человек.

Яркий пример искусственной избыточности — контрольная сумма некоторого файла, вычисляемая по известному алгоритму. Это, например,

циклические суммы **CRC16**, **CRC32**, хэш-суммы **MD5**, **SHA-1**, **SHA-2**, **SHA-3**. Контрольная сумма — это проверочные символы, по которым можно оценить целостность записанного, например, на оптический диск файла.

Кодер источника — это «чёрный ящик» для экономной упаковки (сжатия) информации. Здесь уместно провести аналогию с конструктором **LEGO**, когда кодер разбирает массивную конструкцию и компактно упаковывает мелкие детали в коробку, прилагая рядом схему сборки. Декодер, соответственно, служит для распаковки архива и сборки информации (рис. 1).

Кодер канала — «чёрный ящик» для упаковки информации в «оболочку», которая предназначена для защиты от помех, неизбежно присутствующих в любых каналах связи, а в особенности — в радиоканалах. Декодер канала распаковывает эту «оболочку», по возможности делает «косметический ремонт» принятой информации, и затем передаёт её следующему устройству в цепочке преобразования информации, а «оболочка» за ненужностью отбрасывается.

Кодер и декодер принято называть одним словом — **кодек**.

Предполагается, что студент знает основную суть процессов модуляции-демодуляции цифровых сигналов (символы или цифры не могут передаваться по каналу связи без соответствующих сигналов), и понимает, что для упрощения эти блоки на рис. 1 не показаны.

В данной лабораторной работе изучаются возможности циклического кода Хэмминга и алгоритмы кодирования и декодирования.

Для выполнения данной работы понадобятся:

1. Персональный компьютер с возможностью запуска программного макета в виде исполняемого файла, предназначенного для операционной системы **Windows**.

2. Офисный пакет для подготовки отчёта (принимаются файлы с расширением **odt**, **doc** и **docx**).

Методические требования к выполнению работы:

Если в тексте встречается выделенное жирным шрифтом сказуемое (**показать, заполнить...**), то процесс выполнения соответствующего задания должен быть отражён в отчёте по ходу работы.

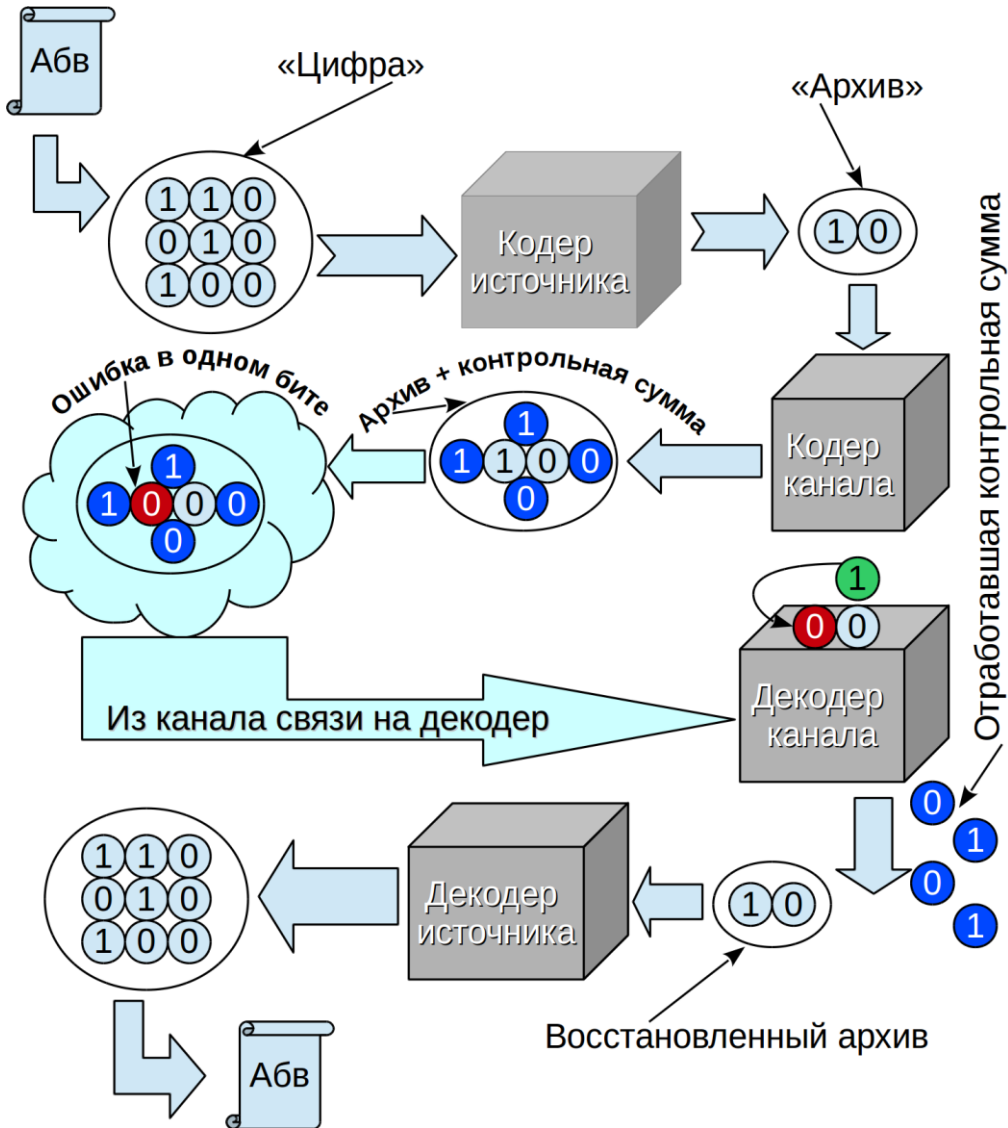
Работу, по возможности, выполнять последовательно, так как задания связаны между собой. Например, ответ на текущий вопрос предполагает знание ответа на некоторые предыдущие. Теорию рекомендуется читать полностью, проверяя «на бумажке с карандашом» выводы формул.

Отчёт должен содержать:

- Титульный лист.
- Ход работы.
- Ответы на вопросы.
- Выводы (резюме).

Теорию, скопированную из данного пособия, приводить не следует. Исключение составляют мысли, изложенные непосредственно тем, кто выполнил данную работу.

Свиток «Абв» — аналоговая информация (текст, речь, видео, изображение...).
 Овалы с нулями и единицами — пакеты с цифровой информацией.
 Стрелки указывают направление передачи информации от отправителя к получателю.

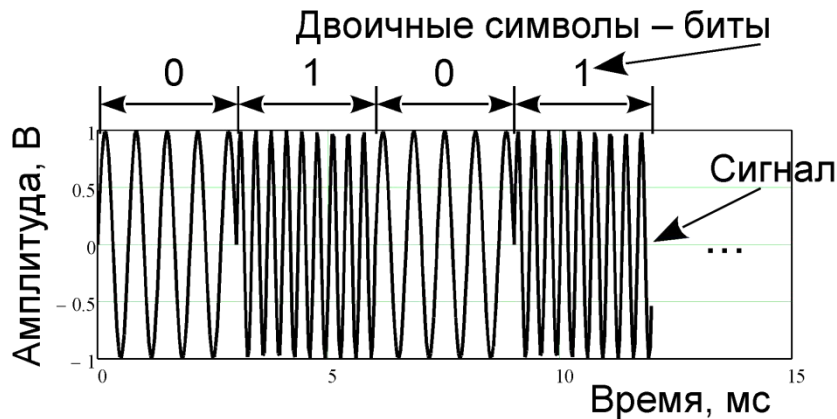


Информация с помощью аналого-цифрового преобразователя переходит в цифровую форму, сжимается кодером источника, а затем каналным кодером вносится искусственная избыточность, после чего упакованная информация передаётся по каналу связи с помехами. Канальный декодер анализирует принятую информацию на предмет наличия ошибок, по возможности исправляет их и передаёт информацию декодеру источника, который её распаковывает. И, наконец, исходный цифровой поток поступает на цифро-аналоговый преобразователь для отображения информации, годной для получателя.

Рис. 1 — Роль двух видов кодирования и их место в цифровых системах передачи информации

2.2 Сведения из теории

Циклические коды, относящиеся к классу *помехоустойчивых кодов*, предназначены для борьбы с помехами в канале связи, которые могут приводить к ошибкам при демодуляции принимаемого сигнала (при угадывании символа по форме принятого сигнала).



Использование помехоустойчивых кодов принято называть *кодированием канала*. Основное свойство данных кодов — наличие избыточной информации, введённой в канал связи по заранее известным правилам, проверка которых на приёмной стороне позволяет обнаружить и даже исправить некоторые ошибки.

Самый понятный способ кодирования-декодирования — табличный, когда составляется таблица, состоящая из двух колонок, в одной из которых перебираются всевозможные информационные слова, а в другой записываются соответствующие разрешённые (табл. А).

Таблица А — Пример табличного помехоустойчивого кодирования

Информационное слово	Разрешённое кодовое слово
00	010101
01	110011
10	111000
11	111111

При этом нет никаких ограничений на правило кодирования, кроме требования **однозначности декодирования**: можно кодировать «на глаз», можно использовать формулы и т.д. В качестве примера табличного кодирования можно привести код **4В/5В**, используемый на физическом уровне сетевой технологии **Ethernet 100Мбит/с (100Base-TX)** для самосинхронизации. Данный код гарантирует не более трёх нулевых битов подряд в выходном потоке. Кодовая таблица содержит всего 16 строк, что не накладывает серьёзных ограничений на использование памяти кодера/декодера. Избыточность кода **4В/5В** проявляется в том, что на входе кодера четыре бита, а на выходе — пять, то есть $2^4 = 16$ слов кодируются пятью битами, в то время как минимальное число битов равно четырём.

Если оценить количество требуемых строк таблицы, например, для кода с информационными словами из ста **двоичных** символов, то получается 2^{100} или около десяти в тридцатой степени! Очевидно, что никакое электронное устройство не обеспечит такой объём памяти с приемлемым быстродействием поиска кодовых слов в таблице. Однако коды таких размеров, и даже больших (около 40000 битов), успешно используются в реальных системах связи. Яркий пример тому — цифровое телевидение [1], а причиной успеха является выделение класса **линейных кодов**.

Основная особенность линейных кодов — линейность операций кодирования (вычисления проверочных символов) и декодирования (анализа принятых символов). Вместо кодовой таблицы используется матрица из k строк, которая позволяет сгенерировать 2^k **двоичных** кодовых слов. Разница в объёмах — налицо, и она тем больше, чем больше длина кода. Ограничение на правило кодирования у линейных кодов — требование линейной независимости строк порождающей матрицы, так как разрешённые кодовые слова получаются с помощью всевозможных сумм строк этой матрицы. Однозначность декодирования никто не отменял, и эта одно-

значность табличных кодов вылилась в линейную независимость строк порождающей матрицы.

Стремясь к компактности кодеков, на этом не останавливаются, и среди линейных кодов выделяют класс **циклических**, основная особенность которых состоит в том, что **циклические перестановки** разрешённых кодовых слов дают разрешённые кодовые слова. При этом используется не матричный, а полиномиальный способ кодирования, который требует задания всего одного вектора, состоящего из коэффициентов некоторого полинома. Цифровые устройства, реализующие данный способ, — обычные цифровые фильтры, состоящие из регистра сдвига, набора сумматоров и умножителей. Причём для двоичных кодов умножители заменяются ключами с состояниями «включено/выключено».

По сути, для циклических кодов достаточно знать лишь одну строку матрицы, с помощью которой осуществляется кодирование, а оставшиеся строки выражаются через циклические перестановки исходной.

Естественно, что у циклических кодов есть как матричный способ записи процесса кодирования, так и табличный, то есть данные коды — это ограниченный по количеству (не всякий линейный код будет циклическим), но при этом весьма интересный и не совсем простой для изучения класс помехоустойчивых кодов.

В данной работе изучаются **два** двоичных циклических кода Хэмминга (7, 4) с генераторными полиномами

$$\begin{aligned} g_1(x) &= x^3 + x + 1, \\ g_2(x) &= x^3 + x^2 + 1. \end{aligned} \tag{1}$$

Параметр (7, 4) означает то, что на вход кодирующего устройства подаются четыре двоичных символа (бита), а с выхода считываются семь символов, в которых содержатся четыре входных и три проверочных символа (трёхбитовая контрольная сумма). Избыточность кода равна $3/7$.

Пусть соответствие произвольной кодовой комбинации коэффициентам некоторого полинома строится по правилу «младший символ справа»

$$s_6 s_5 s_4 s_3 s_2 s_1 s_0 \rightarrow s(x) = s_6 \cdot x^6 + s_5 \cdot x^5 + s_4 \cdot x^4 + s_3 \cdot x^3 + s_2 \cdot x^2 + s_1 \cdot x + s_0. \quad (2)$$

Тогда, например, двум генераторным полиномам (1) будут соответствовать два кодовых слова

$$\begin{aligned} g_1 &= 0001011, \\ g_2 &= 0001101. \end{aligned} \quad (3)$$

Циклические коды являются **блочными**, то есть все кодовые слова имеют постоянную длину, а так как мы рассматриваем пример кода (7, 4), то длина входных слов равна четырём, выходных — семи.

Для двоичных кодов применяется правило сложения элементов кодовых слов (и, естественно, коэффициентов полинома) «по модулю два»

$$\begin{array}{ll} 0+0=0 & 1+0=1 \\ 0+1=1 & 1+1=0, \end{array} \quad (4)$$

которое сформировано на основе обычного сложения, но так, чтобы результат сложения никогда не выходил за пределы 0 и 1 (чётные числа равносильны нулю, нечётные — единице).

Циклические коды являются блочными (n, k) -кодами, поэтому **степени** полиномов должны быть меньше n , из-за чего их берут по модулю n

$$x^n = x^0 = 1.$$

Например, рассмотрим некоторый полином для $n = 7$

$$s(x) = x^4 + x + 1, \quad (5)$$

которому по правилу (2) соответствует кодовое слово

$$0, 0, 1, 0, 0, 1, 1.$$

Если полином (5) умножить на x^4 , то получится новый полином

$$s(x) \cdot x^4 = x^8 + x^5 + x^4, \quad (6)$$

что соответствует **обычному** (нециклическому) сдвигу элементов кодового слова на четыре разряда (бита) влево

0,0,1,0,0,1,1,0,0,0,0 .

Однако это даст кодовое слово длиной одиннадцать элементов (11 бит), что выходит за пределы блока из семи битов. Поэтому для получения циклической перестановки выпадающий крайний слева бит необходимо ставить на первую позицию справа, то есть так, как это показано на рис. 2.

В результате циклического сдвига исходного слова на четыре бита влево (против часовой стрелки, если рассматривать кольцо) получается искомого, которое соответствует полиному (6) после приведения степеней «по модулю семь» (в этом случае $x^7 = 1$, $x^8 = x$), поэтому

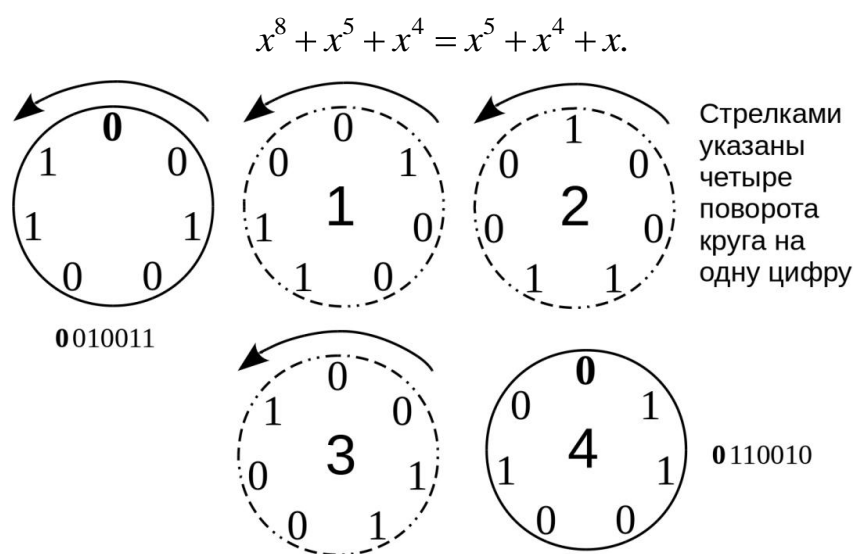


Рис. 2 — Циклический сдвиг кодового слова из семи битов на четыре бита против часовой стрелки

Результат сдвига считывается аналогично, начиная с отметки «12 часов» по часовой стрелке: 0110010. Это и есть циклическая перестановка кодового слова 0010011 на четыре разряда влево при длине блока (регистра памяти) равной семь.

Правило кодирования для циклических кодов простое: входное кодовое слово преобразуется в выходное с помощью умножения входного полинома $a(x)$ на генераторный

$$s(x) = a(x) \cdot g(x). \quad (7)$$

При этом степень генераторного полинома нет смысла брать больше, чем $r = n - k$, так как степень входного полинома по определению не больше $k - 1$, что после перемножения даст степень полинома (7), меньшую n . Коэффициенты при крайних степенях генераторного (порождающего) полинома не должны быть равны нулю, иначе код снизит свою помехоустойчивость. Для двоичных кодов единственным ненулевым элементом является единица, поэтому всегда

$$g_0 = g_r = 1.$$

При способе кодирования (7) нельзя дать гарантию того, что в блоке на выходе кодера можно будет выделить два последовательных блока: **блок информационных** символов плюс **блок проверочных** символов. Помимо перестановок информационных символов, могут наблюдаться их функциональные преобразования (сложения по модулю два)¹.

Зачастую бывает удобно кодировать и декодировать с разделением на два последовательных блока: **блок информационных** и **блок проверочных** символов. Для этого информационные символы сдвигают на r позиций влево. Этому, как мы уже знаем, соответствует умножение $a(x)$ на x^r . Тем самым резервируется r нулевых позиций под проверочные симво-

¹ Например, выходное кодовое слово кода (7, 4) с $g(x) = x^3 + x + 1$ для любого входного слова a_3, a_2, a_1, a_0 равно $a_3, a_2, a_1 + a_3, a_0 + a_2 + a_3, a_1 + a_2, a_0 + a_1, a_0$, откуда следует, что на первых двух позициях слева стоят два информационных символа, на крайней правой позиции — один информационный. Информационный символ a_1 скрыт среди третьего, пятого или шестого символа выходного слова, и неясно, где три проверочных, а где скрытый информационный символ.

лы, которые ещё предстоит вычислить. Добавление вычисленных символов даёт гарантию того, что итоговое кодовое слово будет разрешённым.

Чтобы некоторое кодовое слово было разрешённым, из правила кодирования (7) следует, что ему соответствующий полином должен делиться без остатка на генераторный полином (функция нахождения остатка обозначается как \mathbf{mod}^2 , полином остаток — как $res(x)$),

$$res(x) = s(x) \bmod g(x) = 0. \quad (8)$$

Поэтому дальше находят остаток от деления «сдвинутого» полинома

$$\left[a(x) \cdot x^r \right] \bmod g(x). \quad (9)$$

По определению степень полинома остатка всегда **меньше** степени делителя. Чтобы полином $a(x)x^r$ делился без остатка, необходимо вычесть из него найденный остаток (9), но для двоичных кодов вычитание эквивалентно сложению «по модулю два»

$$1 - 1 = 1 + 1 = 0.$$

В итоге выходной полином будет иметь вид

$$s(x) = a(x) \cdot x^r + \left[a(x) \cdot x^r \right] \bmod g(x). \quad (10)$$

Правило (10) называют *кодированием в систематической форме*. В практической части данной работы исследуется именно оно. Правило (7) — *кодирование в несистематической форме*.

Процесс декодирования — это:

А) **вычисление остатка** от деления полинома, соответствующего принятому кодовому слову, на генераторный полином;

В) **определение номеров ошибочных символов**, которые в случае двоичных кодов заменяются на обратные.

² Например, часовая стрелка использует арифметику по модулю 12, минутная — по модулю 60.

Пункт В является необязательным, так как декодер может работать лишь в режиме обнаружения ошибок.

Процесс декодирования можно выразить формулами:

$$\begin{aligned}v(x) &= s(x) + e(x), \\res(x) &= v(x) \bmod g(x) = e(x) \bmod g(x),\end{aligned}\tag{11}$$

где $e(x)$ — полином, соответствующий **вектору ошибок**.

Для анализа типов ошибок вводят такое понятие как кратность ошибки q , которая для двоичных кодов численно равна количеству единиц в векторе ошибок e , или, что одно и то же, его весу $q = w(e)$.

Из (11) следует, что остаток от деления определяется лишь вектором ошибки и не зависит от того, какое кодовое слово передавалось.

Если остаток от деления равен нулю, то ошибки либо нет, либо она такой кратности, что не обнаруживается декодером. Если остаток не равен нулю, то ошибка есть, и в некоторых случаях может быть исправлена.

Подытоживая предыдущий абзац, можно сказать, что декодирование — это анализ принятых символов и вынесение определённого решения из заранее составленного набора решений.

2.3 Примеры кодирования и декодирования

Выберем генераторный полином кода Хэмминга (7, 4)

$$g_2(x) = x^3 + x^2 + 1.$$

Зададим входное (информационное) слово 1,1,1,0, которому соответствует полином

$$a(x) = x^3 + x^2 + x.$$

Рассмотрим оба способа кодирования (7) и (10), а также пример декодирования по правилу (11).

Кодирование в систематической форме

Число проверочных символов $r = n - k = 7 - 4 = 3$. Умножим поли-

$\begin{array}{r} \oplus x^6 + x^5 + x^4 \\ x^6 + x^5 + x^3 \\ \hline x^4 + x^3 \\ \oplus x^4 + x^3 + x \\ \hline x \end{array}$	$\begin{array}{r} x^3 + x^2 + 1 \\ \hline x^3 + x \end{array}$	<p>ном $a(x)$ на x^3 и найдём остаток от деления методом деления «в столбик». Получим остаток $res(x) = x$ и выходной полином</p> $s(x) = x^6 + x^5 + x^4 + x,$
--	--	--

что соответствует кодовому слову $s = 1,1,1,0,0,1,0$, где проверочные символы выделены жирным шрифтом.

При делении в столбик следует учесть, что

$$x^m + x^m = x^m \cdot 1 + 1 = x^m \cdot 0 = 0.$$

Кодирование в несистематической форме

$$\begin{aligned} s(x) &= a(x)g(x) = x^3 + x^2 + x \cdot x^3 + x^2 + 1 = \\ &= x^6 + x^5 + x^3 + x^5 + x^4 + x^2 + x^4 + x^3 + x = x^6 + x^2 + x, \end{aligned}$$

или $s = 1,0,0,0,1,1,0$. В данном случае нельзя напрямую (выделением блоков) отделить информационные и проверочные символы. **Показать**, что первый слева и два последних символа в выходном слове (выделены жирным шрифтом) — это информационные символы a_3 , a_1 и a_0 соответственно.

Декодирование

Разделим «в столбик» полином $x^6 + x^5 + x^4 + x$, соответствующий

$\begin{array}{r} \oplus x^6 + x^5 + x^4 + x \\ x^6 + x^5 + x^3 \\ \hline x^4 + x^3 + x \\ \oplus x^4 + x^3 + x \\ \hline 0 \end{array}$	$\begin{array}{r} x^3 + x^2 + 1 \\ \hline x^3 + x \end{array}$	<p>найденному разрешённому кодовому слову систематического кода, на выбранный генераторный полином.</p> <p>Получили нулевой остаток, что говорит о том, что принятое кодовое</p>
--	--	--

слово является разрешённым, то есть оно содержится в таблице кодирования. Таблица из-за больших размеров в явном виде не составляется.

Вводя однократную ошибку, например, во второй символ слева, то есть задавая вектор и полином ошибок в виде

$$e = 0100000 \rightarrow e(x) = x^5,$$

получим полином

$$v(x) = s(x) + e(x) = x^6 + x^5 + x^4 + x + x^5 = x^6 + x^4 + x,$$

деление которого даст ненулевой остаток, который можно вычислить, раз-

$\begin{array}{r} \oplus x^5 \\ \underline{x^5 + x^4 + x^2} \\ \oplus x^4 + x^2 \\ \underline{x^4 + x^3 + x} \\ \oplus x^3 + x^2 + x \\ \underline{x^3 + x^2 + 1} \\ x + 1 \end{array}$	$\begin{array}{r} x^3 + x^2 + 1 \\ \hline x^2 + x + 1 \end{array}$	<p>делив $e(x)$ на генераторный полином.</p> <p>Остаток от деления равен $x + 1$ или $0,1,1$ в двоичном представлении (младший бит, как всегда, справа).</p>
---	--	---

Вводя разные **однократные** ошибки (только в первый, только во второй ... только в седьмой символы), можно убедиться, что **все остатки получаются разными**. Этот факт доказывает стопроцентное исправление рассматриваемым кодом всех однократных ошибок.

2.4 Описание лабораторного макета

Лабораторный макет — исполняемый файл **cyclic**, лежащий в папке **./windows** и предназначенный для запуска в операционных системах **Windows**. Программа написана на C++ с использованием открытых библиотек **Qt 4.8** [2] (рис. 3).

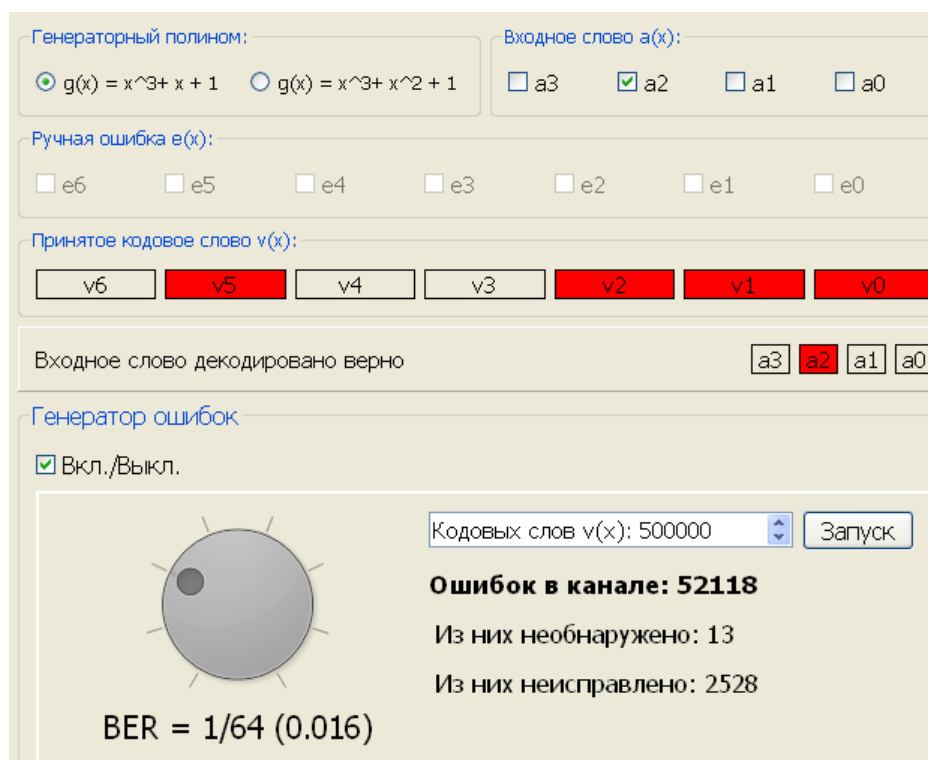


Рис. 3 — Программный макет «Циклический код Хэмминга (7, 4)»

В макете реализовано кодирование и декодирование в систематической форме.

Генераторный полином $g(x)$ выбирается радиокнопками; входное слово $a(x)$ и ручная ошибка $e(x)$ задаются флажками (бит «1» — флажок отмечен, бит «0» — снят). Автоматически вычисляется принятое кодовое слово (красный цвет — бит «1», цвет фона окна — бит «0») с учётом ручной ошибки, которое затем декодируется, а результат выводится на панель, расположенную ниже панели **Принятое кодовое слово $v(x)$** .

На рис. 3 результат декодирования дан в виде текста **Входное слово декодировано верно** и четырёх информационных символов ($a_3 a_2 a_1 a_0$), совпадающих с символами входного слова.

Панель **Генератор ошибок** предназначена для генерации случайных ошибок в автоматическом режиме и их последующем подсчёте. Используется модель **двоичного симметричного канала связи с независимыми ошибками** (рис. 5). **Вероятность битовой ошибки** в канале выставляется ручкой **BER**. Количество генерируемых кодовых слов задаётся в поле ввода **Кодовых слов $v(x)$: 500000**.

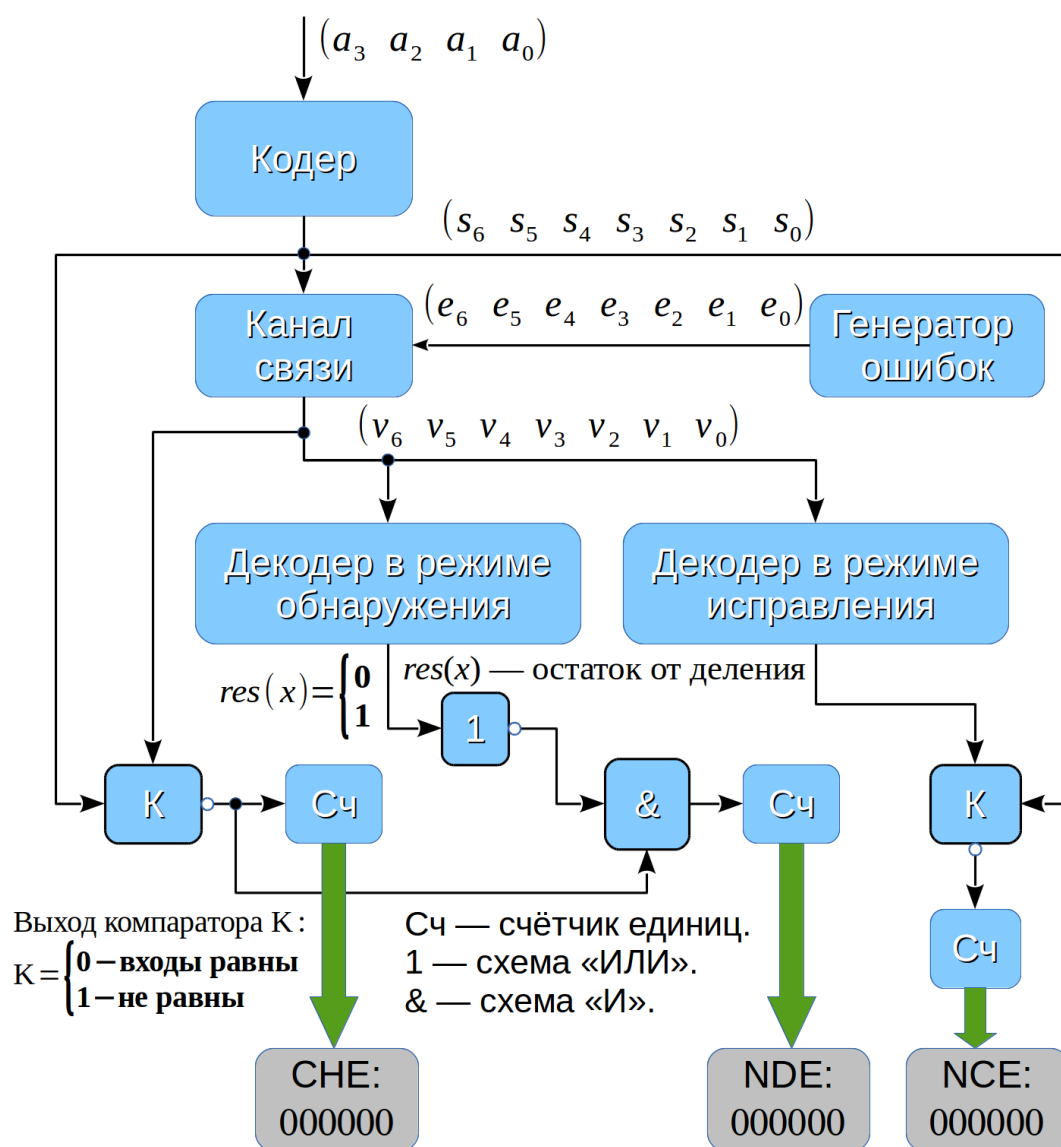
Запуск статистических испытаний осуществляется соответствующей кнопкой, после чего в текстовые поля выводятся показания трёх счётчиков (рис. 3, рис. 4). При включении генератора ошибок ручная ошибка автоматически выключается.

Декодер может работать в двух режимах (рис. 4):

- Обнаружение ошибок;
- Исправление ошибок.

Рассматриваемый код Хэмминга (7, 4) способен **гарантированно** обнаружить все ошибки вплоть до кратности «два» включительно, а **гарантированно** исправить — лишь однократные (детали этого выяснятся после выполнения пунктов 2.5.1–2.5.2).

Если декодер работает в режиме обнаружения, то он, после обработки принятого слова, выносит одно из двух возможных решений: либо ошибка обнаружена, либо — нет. Это можно сделать благодаря сравнению остатка от деления с нулём: он либо отличен от нуля, либо — равен. Программа подсчитывает (счётчик **NDE, No Detected Error**, рис. 4) количество кодовых слов с пропущенными ошибками. Не будут обнаружены такие ошибки, которым соответствуют полиномы $e(x)$, **делящиеся без остатка** на генераторный.



SHE — число ошибочных кодовых слов на выходе канала связи (на входе декодера).
 NDE — число необнаруженных ошибок на выходе декодера.
 NCE — число неисправленных ошибок на выходе декодера.

Рис. 4 — Блок-схема работы лабораторного макета
 «Циклический код Хэмминга (7, 4)»

Если же декодер работает в режиме исправления, то он вычисляет остаток от деления и по номеру остатка корректирует один бит из семи принятых. Если кратность ошибки больше единицы, то коррекция только навредит, зато все однократные ошибки будут исправлены. Вводя в программе ошибку вручную, в этом нужно убедиться. Программа подсчитывает (счёт-

чик **NCE, No Corrected Error**, рис. 4) количество неисправленных кодовых слов, то есть все варианты с кратностями ошибки выше единицы.

Счётчик **CHE (Channel Error**, рис. 4) выдаёт количество кодовых слов, пришедших на вход декодера с ошибкой (хотя бы в одном бите из семи).

2.5 Порядок выполнения работы

Для экономии времени проделать три пункта по порядку:

1. Выполнить расчётное задание.
2. Ответить на вопросы, по возможности соблюдая порядок 1, 2, 3...
3. Выполнить экспериментальную часть.

2.5.1 Расчётное задание

Найти, путём деления в столбик (пункт 2.3), все разрешённые кодовые слова для полиномов $g_1(x)$ и $g_2(x)$ **систематического** кода. Свести результаты в таблицы В и С. Вычислить веса w (количество единиц) всех кодовых слов.

Таблица В — Разрешённые кодовые комбинации систематического циклического кода $(7, 4)$ с полиномом $g_1(x) = x^3 + x + 1$

№	a_3	a_2	a_1	a_0	r_2	r_1	r_0	w
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
3	0	0	1	1				
4	0	1	0	0				
5	0	1	0	1				
6	0	1	1	0				
7	0	1	1	1				
8	1	0	0	0				

9	1	0	0	1				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

Таблица С — Разрешённые кодовые комбинации систематического циклического кода (7, 4) с полиномом $g_2(x) = x^3 + x^2 + 1$

№	a_3	a_2	a_1	a_0	r_2	r_1	r_0	w
0	0	0	0	0				
1	0	0	0	1				
...				
15	1	1	1	1				

Проверить найденные кодовые слова с помощью программы **cyclic**. На выбор из таблиц В и С взять по одному разрешённому полиному $s(x)$ и разделить их на соответствующие генераторные полиномы. Убедиться, что остатки от деления равны нулю, вручную подтвердив правильность кодирования.

В таблицу D выписать полиномы-остатки от деления целых степеней x^i на полиномы $g_1(x)$ и $g_2(x)$.

Таблица D — Остатки от деления целых степеней на генераторные полиномы

Генераторный полином $g_1(x)$		Генераторный полином $g_2(x)$	
Степень x	Остаток	Степень x	Остаток
x^0		x^0	
x^1		x^1	
x^2		x^2	
x^3		x^3	
x^4		x^4	
x^5		x^5	
x^6		x^6	

После **правильного** выполнения расчётного задания ответить на вопросы, подготовив себя к практической части лабораторной работы. Ответы **привести** в отчёте по ходу работы.

2.5.2 Анализ результатов выполнения расчётного задания

На каждый вопрос дать два ответа: для первого кода и второго.

1. Есть ли среди разрешённых кодовые слова с весом пять (то есть состоящие из пяти единиц)? С весом шесть? Семь?

2. Определить веса, по которым сразу определяются запрещённые кодовые слова. Например, вес **один**, так как если в слове только одна единица, то из кодовой таблицы следует, что оно однозначно запрещённое. Проверить веса 0, 1, 2, 3, 4, 5, 6, 7.

3. Какие веса может иметь разрешённое кодовое слово?

4. Обязано ли произвольное слово из семи бит с весом четыре быть разрешённым? С весом ноль (то есть все семь битов — нули)?

5. Сколько в принципе существует разных двоичных слов из семи битов с весом четыре? Сколько подобных слов среди разрешённых?

6. Если к любому разрешённому кодовому слову прибавить любое, имеющее вес два (то есть ввести двукратную ошибку), то обнаружится ли эта ошибка? Всегда? Анализ сделать на основании остатков от деления.

7. Если ввести трёхкратную ошибку, обнаружится ли она? Всегда? Если не всегда, то каков процент того, что ошибка не будет обнаружена?

8. Справится ли декодер с двукратной ошибкой, если он работает в режиме исправления? С трёх-, четырёх-, пяти-, шести-, семикратной? Для анализа использовать таблицу D.

2.5.3 Экспериментальная часть

В программном лабораторном макете предусмотрены два режима ввода ошибки в кодовые слова кода Хэмминга (7, 4):

1. Режим ручной ошибки, когда оператор отмечает/снимает отметки на панели Ручная ошибка $e(x)$ (рис. 3). Если флажок отмечен, то соответствующий бит в принятом кодовом слове инвертируется («0» на «1», «1» на «0»).
2. Режим генерации ошибок, когда оператор задаёт количество испытаний, и после нажатия кнопки **Запуск** программа автоматически случайным образом будет вводить ошибки в неизменное разрешённое кодовое слово.

Режим ручной ошибки

Запустив программу и выбрав полином и входное кодовое слово в соответствии с Вашим вариантом³, вручную ввести следующие ошибки:

- всевозможные однократные (их семь);

³ Если номер Вашего варианта — чётное число, необходимо выбрать полином $g_2(x)$. В противном случае — полином $g_1(x)$. Входное слово берётся из заполненной таблицы пункта 2.5.1 по номеру её строки, который должен совпадать с Вашим вариантом по правилу «вариант 00» — «номер строки № 0» ... «15 — № 15», «16 — № 0» и так далее.

- две трёхкратные, но так, что первую обнаруживает, а вторую — нет;
- две четырёхкратные, одну из которых обнаруживает, другую — нет;
- все шестикратные (их семь);
- семикратную (она одна).

Результаты декодирования **привести** в отчёте (копируя результаты из программы в виде таблицы битов «Вход: 1010, Выход: 1011»).

Почему одну выбранную трёхкратную ошибку обнаруживает, другую — нет? Четырёхкратную? **Ответить** на два данных вопроса.

Режим генерации ошибок

В данной работе рассматривается модель двоичного симметричного канала связи с независимыми ошибками (рис. 5), возникающими в каждом бите с вероятностью p (вероятность битовой ошибки, BER, *Bit Error Rate*).

Предварительно сделав расчётное задание (пункт 2.5.1) и ответив на вопросы (пункт 2.5.2), для двух рассматриваемых кодов (7, 4) **определить** в виде формул три вероятности:

- вероятность P_1 того, что кодовое слово длиной семь бит, поступающее на вход декодера из канала связи, **будет содержать ошибку**;
- Вероятность P_2 того, что содержащаяся в принятом кодовом слове ошибка **не будет обнаружена**;
- Вероятность P_3 того, что имеющаяся ошибка **не будет исправлена**.

При выводе трёх формул использовать формулу Бернулли для вероятности $P(q)$ количества успехов q при проведении n независимых статистических испытаний

$$P(q) = C_n^q p^q (1-p)^{n-q},$$

где p — вероятность успеха (для нас «успех» — это битовая ошибка);

n — длина кодового слова ($n = 7$);

$$C_n^q = \frac{n!}{q! (n-q)!} \text{ — число сочетаний из } n \text{ по } q.$$

Например, если $q = 2$, то это означает, что в кодовом слове из семи бит произошла ошибка в двух любых битах. Число разных двукратных ошибок в нашем случае составляет $C_7^2 = 21$. Помимо факториальной формулы, числа сочетаний легко могут быть вычислены по треугольнику Паскаля.

Также при выводе трёх формул **учесть** результаты ответов на вопросы из пункта 2.5.2.

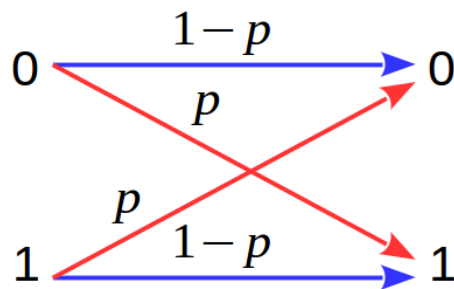


Рис. 5 — Модель двоичного симметричного канала связи с независимыми ошибками

Вычислить по найденным формулам численные значения трёх вероятностей P_1, P_2, P_3 для набора BER

$$p = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256} \right).$$

Занести результаты в таблицу.

С помощью программы **csulic**, включив генератор ошибок, провести статистический эксперимент по обнаружению и исправлению ошибок **для обоих** рассматриваемых кодов. Результаты измерений **занести** в таблицу. Входное слово можно задавать любым (кстати, почему?). Число испытаний выбрать из диапазона 2'000'000–5'000'000 и **не изменять** на всём протяжении измерений.

Оценить по показаниям трёх счётчиков три вероятности P_1, P_2, P_3 и **сравнить** их с вычисленными теоретическими значениями P_1, P_2, P_3 .

В отчёте все сделанные выводы собрать воедино (сделать резюме).

Ответить на вопросы:

1. Чем отличаются два рассмотренных кода?
2. Какой из них более помехоустойчив и, если это так, то почему?

2.6 Источники

1. Шахнович И. DVB-T2 — новый стандарт цифрового телевизионного вещания [Электронный ресурс]// Электроника: наука, технология, бизнес, 6/2009. — URL http://www.electronics.ru/files/article_pdf/0/article_258_849.pdf , режим доступа: открытый, дата обращения: 25.06.2015.

2. Qt | Cross-platform application & UI development framework [Электронный ресурс]. — URL <http://www.qt.io/>, режим доступа: открытый, дата обращения: 25.06.2015.

ПРИЛОЖЕНИЕ А Варианты заданий лабораторной работы № 1

Вариант	$S(t)$	Δt	Вариант	$S(t)$	Δt	Вариант	$S(t)$	Δt
01	$S_1(t)$	$-T/6$	17	$S_3(t)$	$T/4$	33	$S_6(t)$	$-T/3$
02	$S_1(t)$	$-T/4$	18	$S_3(t)$	$T/6$	34	$S_6(t)$	$T/3$
03	$S_1(t)$	$-T/3$	19	$S_4(t)$	$-T/6$	35	$S_6(t)$	$T/4$
04	$S_1(t)$	$T/3$	20	$S_4(t)$	$-T/4$	36	$S_6(t)$	$T/6$
05	$S_1(t)$	$T/4$	21	$S_4(t)$	$-T/3$	37	$S_7(t)$	$-T/6$
06	$S_1(t)$	$T/6$	22	$S_4(t)$	$T/3$	38	$S_7(t)$	$-T/4$
07	$S_2(t)$	$-T/6$	23	$S_4(t)$	$T/4$	39	$S_7(t)$	$-T/3$
08	$S_2(t)$	$-T/4$	24	$S_4(t)$	$T/6$	40	$S_7(t)$	$T/3$
09	$S_2(t)$	$-T/3$	25	$S_5(t)$	$-T/6$	41	$S_7(t)$	$T/4$
10	$S_2(t)$	$T/3$	26	$S_5(t)$	$-T/4$	42	$S_7(t)$	$T/6$
11	$S_2(t)$	$T/4$	27	$S_5(t)$	$-T/3$	43	$S_8(t)$	$-T/6$
12	$S_2(t)$	$T/6$	28	$S_5(t)$	$T/3$	44	$S_8(t)$	$-T/4$
13	$S_3(t)$	$-T/6$	29	$S_5(t)$	$T/4$	45	$S_8(t)$	$-T/3$
14	$S_3(t)$	$-T/4$	30	$S_5(t)$	$T/6$	46	$S_8(t)$	$T/3$
15	$S_3(t)$	$-T/3$	31	$S_6(t)$	$-T/6$	47	$S_8(t)$	$T/4$
16	$S_3(t)$	$T/3$	32	$S_6(t)$	$-T/4$	48	$S_8(t)$	$T/6$