Применение теоремы Котельникова.

Цель работы: применение теоремы Котельникова при кодировании информации

Методические указания.

При этом как следует из названия, символы некоторого первичного алфавита (например, русского) кодируются комбинациями символов двоичного алфавита (т.е. *0* и *1*), причем, длина кодов и, соответственно, длительность передачи отдельного кода, могут различаться. Длительности элементарных сигналов при этом одинаковы *($\tau_0=\tau_1=\tau$)*. За счет чего можно оптимизировать кодирование в этом случае? Очевидно, суммарная длительность сообщения будет меньше, если применить следующий подход: тем буквам первичного алфавита, которые встречаются *чаще*, присвоить более *короткие* по длительности коды, а тем, относительная частота которых меньше – коды более длинные. Но длительность кода – величина дискретная, она *кратна* длительности сигнала τпередающего один символ двоичного алфавита. Следовательно, коды букв, вероятность появления которых в сообщении выше, следует строить из возможно меньшего числа элементарных сигналов. Построим кодовую таблицу для букв русского алфавита, Очевидно, возможны различные варианты двоичного кодирования, однако, не все они будут пригодны для практического использования – важно, чтобы закодированное сообщение могло быть *однозначно декодировано*, т.е. чтобы в последовательности *0* и *1*, которая представляет собой многобуквенное кодированное сообщение, всегда можно было бы различить обозначения отдельных букв. Проще всего этого достичь, если коды будут разграничены *разделителем* – некоторой постоянной комбинацией двоичных знаков. Условимся, что разделителем отдельных кодов букв будет последовательность *00* (признак конца знака), а разделителем слов – *000* (признак конца слова – пробел). Довольно очевидными оказываются следующие правила построения кодов:

- код признака конца знака может быть включен в код буквы, поскольку не существует отдельно (т.е. кода всех букв будут заканчиваться *00*);
- коды букв не должны содержать двух и более нулей подряд в середине (иначе они будут восприниматься как конец знака);
- код буквы (кроме пробела) всегда должен начинаться с *1*;
- разделителю слов (*000*) всегда предшествует признак конца знака; при этом реализуется последовательность *00000* (т.е. если в конце кода встречается комбинация *…000* или *…0000*, они не воспринимаются как разделитель слов); следовательно, коды букв могут оканчиваться на *0* или *00* (до признака конца знака).

Длительность передачи каждого отдельного кода t_i, очевидно, может быть найдена следующим образом: $t_i = k_i \bullet \tau$, где k_i – количество элементарных сигналов (бит) в коде символа *i*. В соответствии с приведенными выше правилами получаем следующую таблицу кодов:

Таблица 1.

Буква	Код	$p_i \cdot 10^3$	k_i	Буква	Код	$p_i \cdot 10^3$	k_i
пробел	*000*	174	3	я	*1011000*	18	7
о	*100*	90	3	ы	*1011100*	16	7
е	*1000*	72	4	з	*1101000*	16	7
а	*1100*	62	4	ь,ъ	*1101100*	14	7
и	*10000*	62	5	б	*1110000*	14	7
т	*10100*	53	5	г	*1110100*	13	7
н	*11000*	53	5	ч	*1111000*	12	7
с	*11100*	45	5	й	*1111100*	10	7
р	*101000*	40	6	х	*10101000*	9	8
в	*101100*	38	6	ж	*10101100*	7	8
л	*110000*	35	6	ю	*10110000*	6	8
к	*110100*	28	6	ш	*10110100*	6	8
м	*111000*	26	6	ц	*10111000*	4	8
д	*111100*	25	6	щ	*10111100*	3	8
п	*1010000*	23	7	э	*11010000*	3	8
у	*1010100*	21	7	ф	*11010100*	2	8

Теперь по формуле можно найти среднюю длину кода $K^{(2)}$ для данного способа кодирования:

$$K^{(2)} = \sum_{i=1}^{32} p_i \cdot k_i = 4{,}964$$

Поскольку для русского языка $I_1^{(r)} = 4{,}356$ *бит*, избыточность данного кода, составляет:

$Q^{(r)} = 1 - 4{,}356/4{,}964 \approx 0{,}122;$

это означает, что при данном способе кодирования будет передаваться приблизительно на 12% больше информации, чем содержит исходное сообщение. Аналогичные вычисления для английского языка дают значение $K^{(2)} = 4{,}716$, что при $I_1^{(e)} = 4{,}036$ *бит* приводят к избыточности кода $Q^{(e)} = 0{,}144$.

Рассмотрев один из вариантов двоичного неравномерного кодирования, попробуем найти ответы на следующие вопросы: возможно ли такое кодирование без использования разделителя знаков? Существует ли наиболее оптимальный способ неравномерного двоичного кодирования?

Суть первой проблемы состоит в нахождении такого варианта кодирования сообщения, при котором последующее выделение из него каждого отдельного знака (т.е. декодирование) оказывается однозначным без специальных указателей разделения знаков. Наиболее простыми и употребимыми кодами такого типа являются так называемые *префиксные коды*, которые удовлетворяют следующему условию (*условию Фано*):

Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с началом (префиксом [1]) какого-либо иного более длинного кода.

Например, если имеется код *110*, то уже не могут использоваться коды *1*, *11*, *1101*, *110101* и пр. Если условие Фано выполняется, то при прочтении (расшифровке) закодированного сообщения путем сопоставления со списком кодов всегда можно точно указать, где заканчивается один код и начинается другой.

Задание

Пусть имеется следующая таблица префиксных кодов:

а	**л**	**м**	**р**	**у**	**ы**
10	***010***	***00***	***11***	***0110***	***0111***

Требуется декодировать сообщение: *00100010000111010101110000110*

Декодирование производится циклическим повторением следующих действий:

1. отрезать от текущего сообщения крайний левый символ, присоединить к рабочему кодовому слову;
2. сравнить рабочее кодовое слово с кодовой таблицей; если совпадения нет, перейти к (1);
3. декодировать рабочее кодовое слово, очистить его;
4. проверить, имеются ли еще знаки в сообщении; если «да», перейти к (1).

Применение данного алгоритма дает:

Шаг	Рабочее слово	Текущее сообщение	Распо-знанный знак	Декодиро-ванное сообщение
0	пусто	***0***0100010000111010101110000110	–	–
1	*0*	***0***1000100001110101011100001 10	нет	–
2	***00***	***1***0001000011101010111000011 0	***м***	*м*
3	*1*	***0***0010000111010101110000110	нет	*м*
4	***10***	***0***0100001110101011100001 10	***а***	*ма*
5	*0*	***0***1000011101010111000011 0	нет	*ма*
6	***00***	***1***0000111010101110000110	***м***	*мам*
…				

Доведя процедуру до конца, получим сообщение: «*мама мыла раму*».

Таким образом, использование префиксного кодирования позволяет делать сообщение более коротким, поскольку нет необходимости передавать разделители знаков. Однако, условие Фано не устанавливает способа формирования префиксного кода и, в частности, наилучшего из возможных.

Контрольные вопросы:

1. Формулировка теоремы Котельникова
2. Что называется разделителем?
3. Какие коды называются префиксными кодами?
4. В чем заключается условие Фано?
5. Когда может быть декодированнеравномерный код?