

ПРИМЕНЕНИЕ АЛГОРИТМОВ ПЕРЕМЕЖЕНИЯ ПРИ ПЕРЕДАЧЕ АНАЛОГОВОЙ ИНФОРМАЦИИ В ЦИФРОВОЙ СИСТЕМЕ СВЯЗИ

О. К. Сапрыкин, В. В. Кудинов

Рассматривается разработка математической модели цифровой системы связи с применением алгоритмов перемежения при передаче аналоговой информации. Изучается влияние алгоритмов перемежения при передаче аналоговой информации на качество передаваемой информации в условиях воздействия помех. Предлагается несколько вариантов алгоритмов перемежения и проводится анализ их работы, а также выдаются рекомендации по их применению.

Ключевые слова: информация, цифровая система связи, алгоритм перемежения, перемежитель.

Введение

В современном мире одним из ключевых элементов коммуникации являются цифровые линии связи, которые обеспечивают передачу большого количества информации. С увеличением объема информации и скорости ее передачи возникает потребность в оптимизации и совершенствовании передачи сигналов. Это научная и практическая проблема, которая требует разработки новых методов и алгоритмов обработки сигналов для улучшения помехоустойчивости систем передачи информации. Решение этой проблемы лежит в повышении характеристик систем передачи информации путем модификации уже существующих алгоритмов передачи сигналов [1, 2].

В этой связи не менее важным является групповое перемежение данных, технология, позволяющая эффективно использовать цифровые линии связи. Перемежение является эффективным способом борьбы с пакетными ошибками. Он заключается в разнесении кодовых символов по всей длине последовательности данных. На передающей стороне символы переставляются, а на приемной они восстанавливаются. Такая операция позволяет разделить последовательные ошибочные символы по всей длине кодового слова, что делает их случайно распределенными для декодера.

В данной работе алгоритмы перемежения будут применены к обработке звуковых файлов при их передаче в цифровой системе связи.

Классификация и структура перемежения

Несмотря на то, насколько совершенными являются коды с исправлением ошибок в условиях каналов с замираниями – они не могут гарантировать высокую надежность приема. Это обусловлено резким снижением корректирующих возможностей кодов при появлении в канале пакетов ошибок, вызванных сильными помехами и замираниями. В истории были попытки создания специальных кодов, способных бороться с такими ошибками,

например, использование кодов Хагелбаргера и Файера, но ввиду их высокой вычислительной сложности они не нашли практического применения.

Однако для борьбы с замираниями и возникновением связанных с ними пакетов ошибок был найден неожиданный и простой выход – процедура перемежения. Эта процедура представляет собой перестановку символов кодированной последовательности до модуляции и ее восстановление после демодуляции, что позволяет эффективно разнести соседние символы. При этом процедура не вносит избыточности, изменяя только порядок следования символов или битов [3].

Обобщенная структурная схема канала передачи данных с использованием перемежителей/депермежителей представлена на рис. 1.

Перемежитель – это устройство, которое принимает последовательности элементов в фиксированном алфавите и формирует выходную последовательность того же алфавита, сохраняя все элементы, но изменяя их порядок. В зависимости от организации буфера перемежители подразделяются на два основных класса: блочные, которые будут рассмотрены в данной работе, и сверточные [4].

Блочное устройство в своей самой простой реализации представляет собой двумерный массив, где запись данных осуществляется по столбцам, а чтение – по строкам. При приеме этот процесс осуществляется в обратном порядке, что математически эквивалентно транспонированию матрицы.

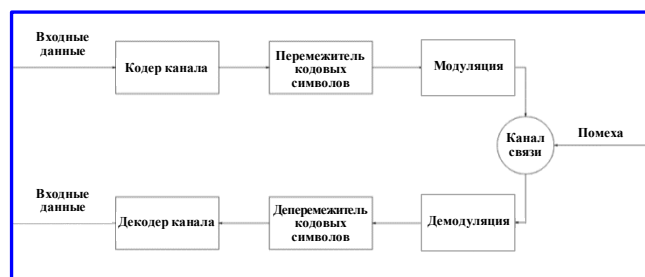


Рис. 1. Обобщенная структурная схема канала передачи данных с использованием перемежителя/депермежителя

Блочный перемежитель сохраняет поступающий блок из K элементов в буфере, после чего производит перестановку каждого блока. Каждая перестановка выполняется независимо от других блоков.

Сверточные устройства, как правило, реализуются в виде N регистров сдвига различной длины, в которые данные последовательно записываются с использованием коммутатора. Регистры сдвига обеспечивают задержку данных на определенную величину. Декодер строится аналогично, но с обратным порядком расположения ветвей, и синхронизируется с кодером. В результате пакетная ошибка длительностью менее N символов разбивается на ряд отдельных ошибок, разделенных интервалом в M символов.

Сверточный перемежитель имеет начальное состояние внутри буферного окна и осуществляет перестановку первого блока последовательности, учитывая начальное состояние. Затем он выполняет перестановку с учетом элементов, оставшихся в данном окне от предыдущего блока. Сверточное перемежение требует вдвое меньшего объема памяти, чем блочное.

В рамках данной работы был исследован принцип действия только блочных перемежителей.

Блочные перемежители могут быть либо периодическими, что означает постоянство перестановок с течением времени, либо псевдослучайными, которые осуществляют перестановку информационных (кодовых) символов в соответствии с псевдослучайным законом и может выполняться только над множеством элементов. На практике предпочтение отдается периодическим перемежителям ввиду их более простой технической реализации [4].

Широко распространенными блочными перемежителями являются матричные перемежители. Работу таких перемежителей можно представить как запись и чтение элементов из матриц-накопителей в разных направлениях.

Одним из классических примеров матричных перемежителей является перемежитель вида «строка – столбец». При использовании таких перемежителей передаваемые информационные символы записываются первоначально по строкам или столбцам, а считывание символов производится в обратном порядке (рис. 2).

В реальных каналах связи часто наблюдается группирование ошибок. Если на декодер поступают такие группы ошибок, то в выходном блоке символы останутся ошибочными на протяжении всего блока. При последующем декодировании эти ошибки передаются следующему декодеру, что делает их

исправление затруднительным. В итоге принимается целый блок ошибочной информации, который нельзя будет исправить. В этом случае для качественного декодирования ошибочного блока символов необходим перемежитель. Тогда при возникновении сгруппированных ошибок перемежитель разнесет символы исходной последовательности, и воздействие помехи на сигнал будет незначительным. Для декодера символы, подвергшиеся изменению ввиду действия помехи, окажутся разнесены по разным блокам, и он сможет их исправить (рис. 3).

Однако основным недостатком перемежителей такого типа является их периодичность. В случае если период повторения помех совпадает с периодичностью перемежения, существует вероятность того, что при передаче информации на вход приемника поступит сообщение, полностью состоящее из ошибок.

Псевдослучайные перемежители, в свою очередь, предназначены для устранения регулярных зависимостей в кодированной последовательности. Перестановка элементов случайного перемежителя формируется с использованием датчика случайных чисел.

Результаты проведенных исследований

Исследование работы алгоритмов перемежения происходило на математической модели системы передачи информации, написанной на языке *Octave*. В качестве передаваемого аналогового сигнала был выбран небольшой фрагмент аудиофайла. При считывании программой этот фрагмент представляет собой матрицу, где строки – это соответствующие отсчеты определенной амплитуды, а столбцы – количество каналов, в нашем случае их два. В дальнейшем для каждого из каналов строится вектор [5].

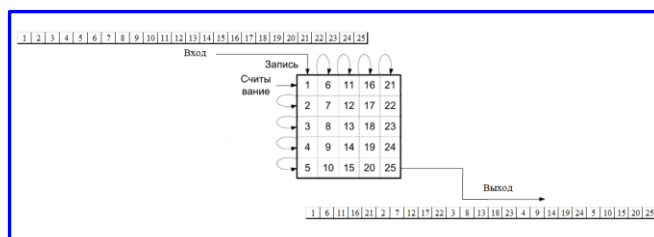


Рис. 2. Пример работы матричного перемежителя

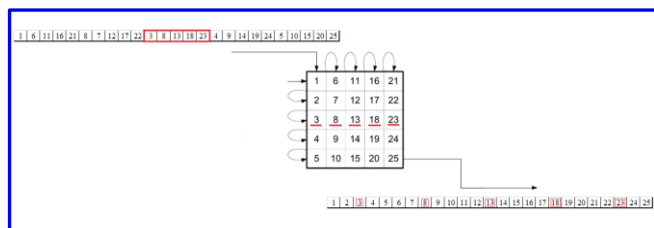


Рис. 3. Пример работы перемежителя при воздействии длительной помехи

Псевдослучайный алгоритм перемежения реализован при помощи встроенной функции *Octave – rand*. Функция *rand* возвращает псевдослучайное число в диапазоне от 0 до какого-то максимального значения. Это число создается алгоритмом, использующим специальное значение – *seed*, которое, обычно, берется из системного времени компьютера, но также его можно задавать вручную. Следовательно, если *seed* всегда будет одним и тем же, то генерируемые числа меняться не будут.

Используя данное свойство функции *rand*, можно получать одну и ту же псевдослучайную последовательность в любой момент времени, что поможет при создании псевдослучайного алгоритма перемежения. В данной работе эта функция была необходима для генерации псевдослучайных чисел, которые в дальнейшем использовались как номера позиций элементов вектора, на которые переставлялись значения из полученного при считывании аудиофайла.

Чтобы восстановить оригинальный порядок следования элементов, выполнялась процедура деперемежения. Для псевдослучайного алгоритма процедура деперемежения состоит в том, чтобы перебрать элементы перемеженного вектора значений амплитуд в обратном порядке. Таким образом, все отсчеты возвращаются на свои места и на выходе деперемежителя появляется исходная последовательность.

Стоит отметить то, что это является именно математической моделью, поэтому подобная реализация данного алгоритма является допустимой.

Матричный (диагональный) алгоритм перемежения работает по принципу, указанному на рис. 4. Во время первой итерации из матрицы берется первый элемент *N1* и записывается в вектор, во второй итерации из матрицы берутся элементы *N6* и *N2* и последовательно записываются в вектор и т.д. После прохождения этим алгоритмом всей матрицы получается вектор перемеженных данных.

Так как в программе при считывании аудиофайла его отсчеты в дальнейшем записываются в вектор, то для реализации матричного алгоритма перемежения необходимо этот вектор преобразовать в матрицу, подобную той, что представлена на рис. 2. Для этого была использована встроенная функция *Octave – reshape*. Данная функция изменяет форму массива путем перестановки существующих элементов, то есть команда *reshape(A,2,4)* преобразует вектор *A* в матрицу размером 2 на 4. Особенностями этой функции является, то, что количество элементов вектора должно совпадать с количеством элементов в создаваемой матрице и то, что в матрицу элементы записываются не по строкам, а по столбцам.

Алгоритм деперемежения работает также, как и алгоритм перемежения, за исключением того, что теперь матрица заполняется элементами из вектора

перемеженных значений, а затем эти элементы последовательно записываются в вектор.

Графики, демонстрирующие работу алгоритмов перемежения без воздействия помехи, изображены на рис. 5–6.

Оценка качества работы алгоритмов происходила при условии пропадания части сигнала (рис. 7–8). Для более подробного анализа работы алгоритмов перемежения исследование проводилось при пропадании на различных по длине участках сигнала. Так как провести численную оценку качества звуковых файлов трудно, для анализа был проведен метод экспертных оценок.

Результатом исследования алгоритмов перемежения является таблица с оценкой результатов их работы (таблица).

Система градации оценок:

5 – отлично, влияние помех не слышно, либо их влияние не значительно;

4 – хорошо, влияние помех заметно, но на точности восприятия информации это не сказывается;

3 – удовлетворительно, помехи сильно влияют на восприятие информации, некоторые слова и фразы невозможно разобрать;

2 – плохо, помехи сильно мешают восприятию информации, разобрать все слова и фразы невозможно;

«–» – исследование не проводилось, ввиду отсутствия необходимости.

Для проверки правильности работы алгоритмов были построены попарные гистограммы для исходного аудиофайла с перемеженным аудиофайлом и для перемеженного аудиофайла с пропаданием с деперемеженным (рис. 9–10).

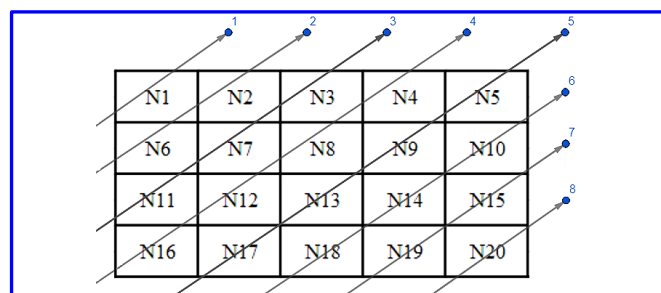


Рис. 4. Принцип работы диагонального алгоритма перемежения

Таблица

Оценка работы алгоритмов перемежения при пропадании участка сигнала

Оценка слышимости звука		
% пропадания сигнала	Тип перемежения	
	Случайное	Диагональное
10	5	5
33	5	4
50	4	3
80	3	2
90	3	–
96	2	–

Исходя из полученных данных, приведенных в таблице, можно сделать вывод, что при 10% и 33% длительности пропадания сигнала от всей его длины оба алгоритма перемежения справляются с тем, чтобы информацию можно было распознать. Однако при 50% длительности помехи диагональный алгоритм перемежения с обработкой сигнала справляется хуже, по сравнению со случайным алгоритмом.

При 80% длительности помехи информацию можно распознать только с использованием случайного алгоритма перемежения, в то время как матричный алгоритм уже не позволяет этого сделать. Это происходит из-за того, что диагональный алгоритм перемежения имеет некую периодичность в отличие от случайного. Интересным оказалось то, что даже при длительности помехи 90% использование

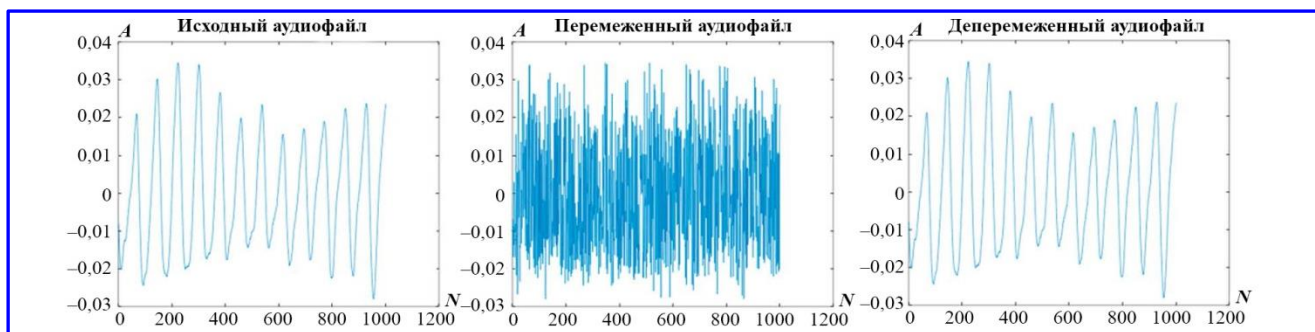


Рис. 5. Графики сигнала при работе случайного алгоритма перемежения

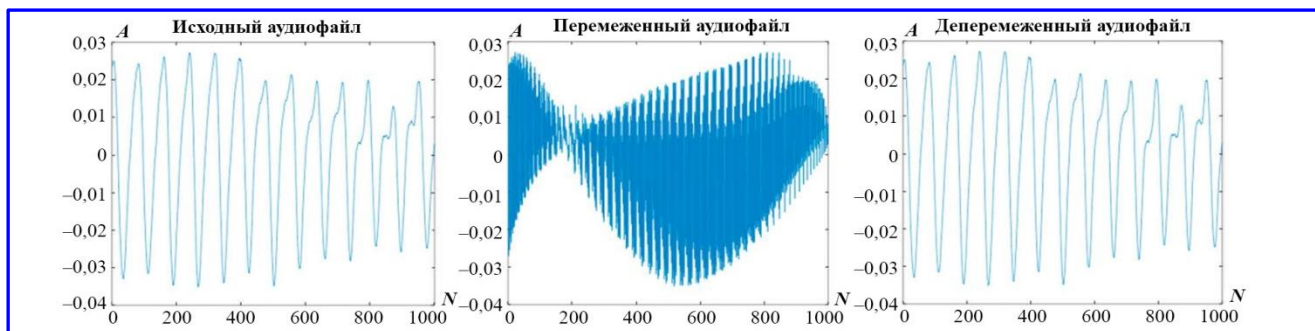


Рис. 6. Графики сигнала при работе диагонального алгоритма перемежения

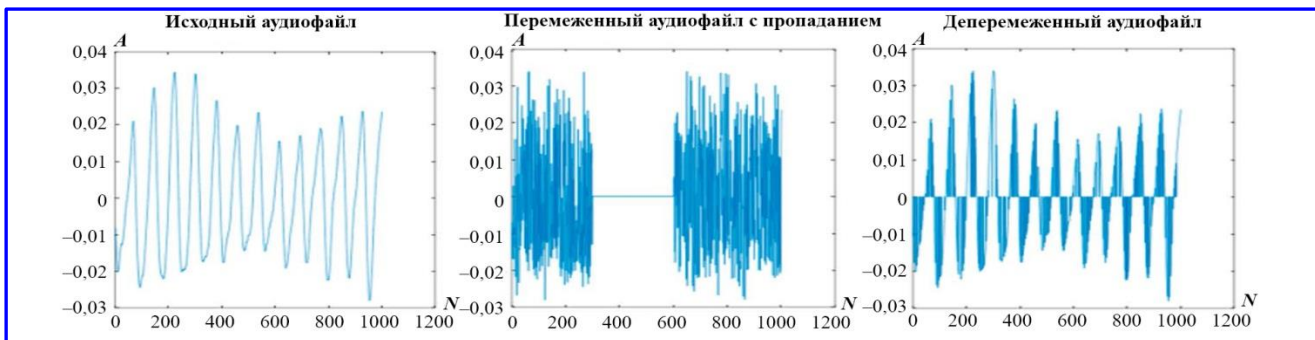


Рис. 7. Графики сигнала при работе случайного алгоритма перемежения и длительности помехи 50%

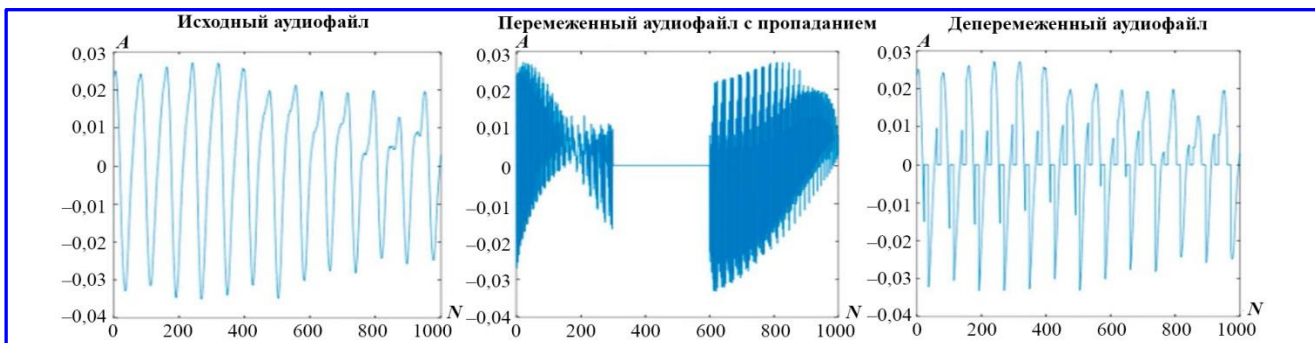


Рис. 8. Графики сигнала при работе диагонального алгоритма перемежения и длительности помехи 50%

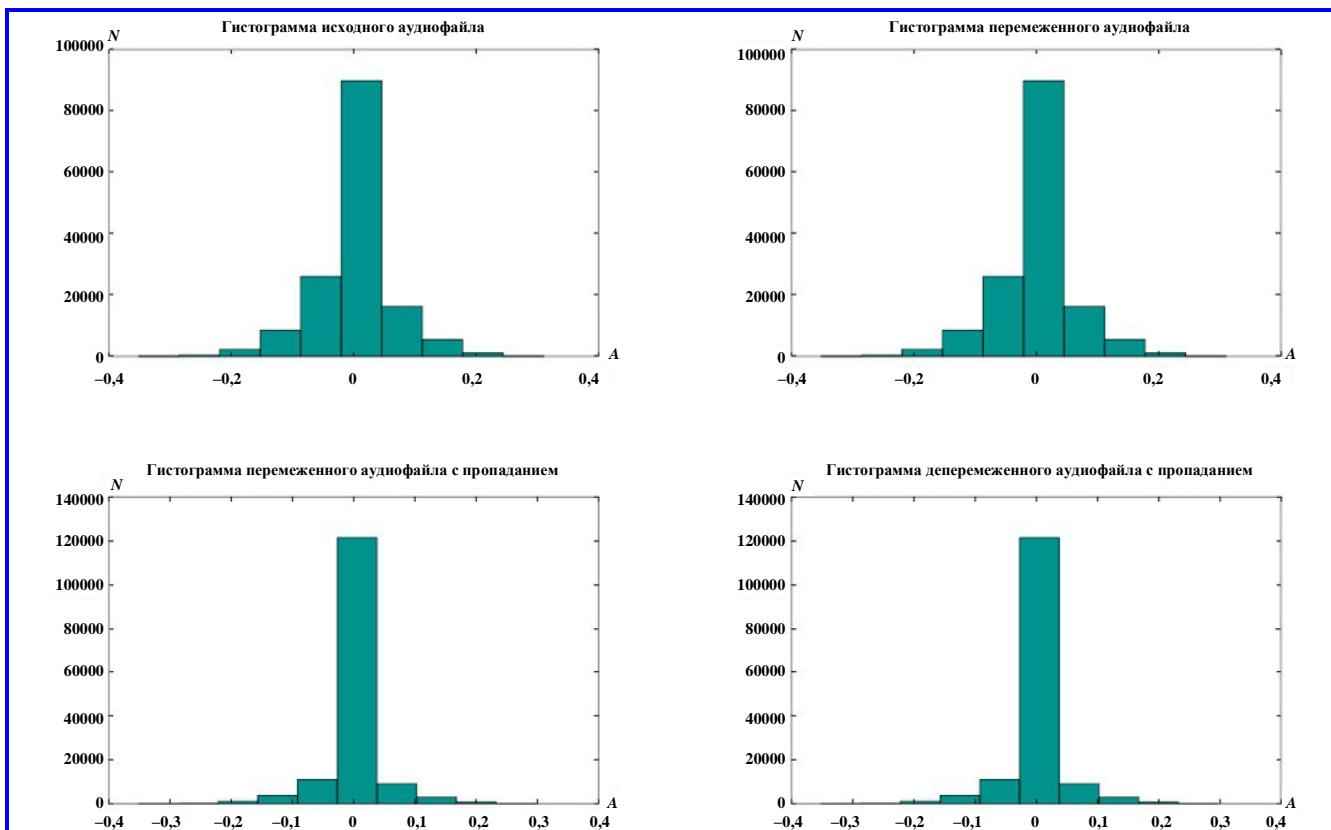


Рис. 9. Гистограммы аудиофайла при работе случайного алгоритма перемежения с длительностью помехи 50%

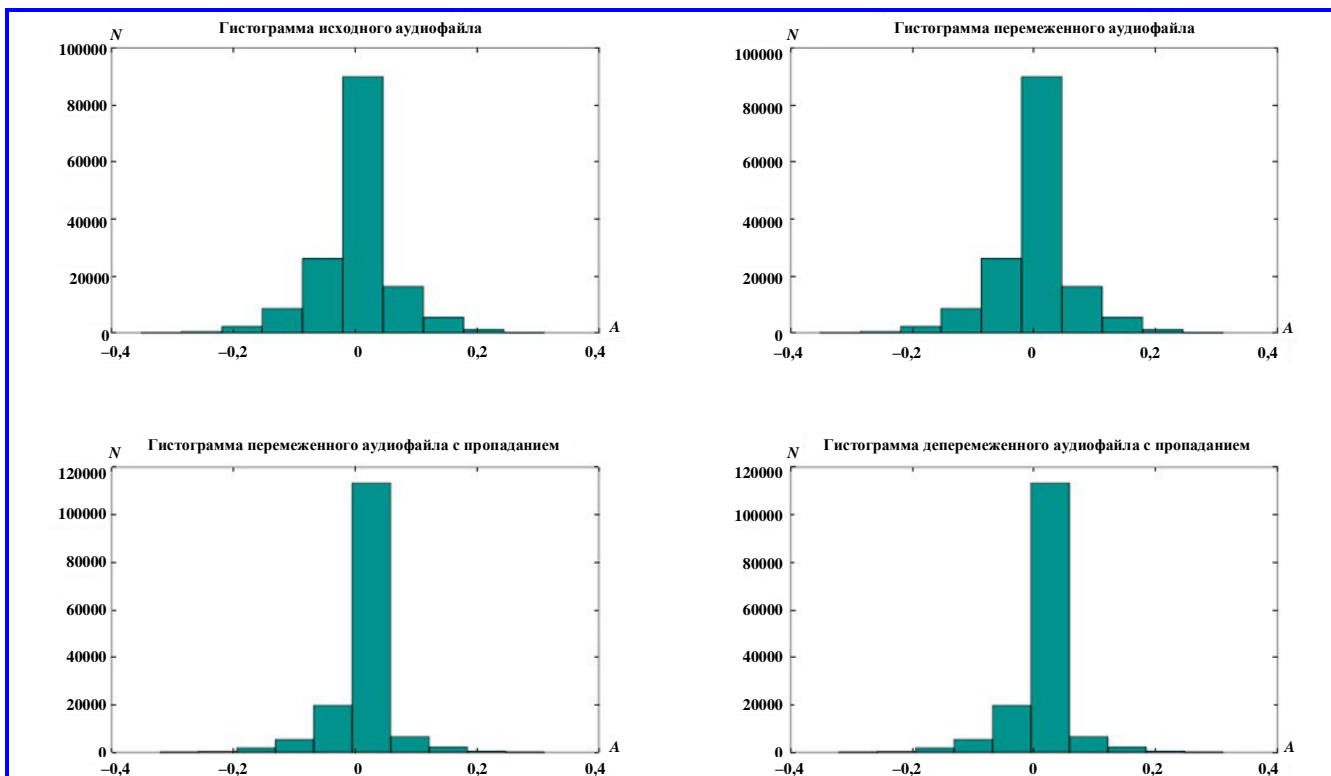


Рис. 10. Гистограммы аудиофайла при работе диагонального алгоритма перемежения с длительностью помехи 50%

случайного алгоритма перемежения позволяет отчетливо разобрать передаваемую информацию. Но при длительности помехи 96% относительно длительности сигнала случайный алгоритм оказывается не способен обработать сигнал так, чтобы информацию можно было отчетливо воспринять.

Выводы

В результате проведенного исследования были протестированы два разных метода перемежения информации – случайный и диагональный, и каждый из них показал свою эффективность. Из этих методов следует выделить случайный алгоритм перемежения, так как он при одинаковых условиях проведения экспериментов показывает лучший результат, чем диагональный алгоритм перемежения. Отсюда можно сделать вывод, что при наличии в канале передачи информации большого количества помех, следует применять случайный алгоритм перемежения.

Подводя итог, можно утверждать, что использование методов перемежения данных при передаче сигнала по цифровым линиям связи является достаточно эффективным методом борьбы с помехами и их использование в сочетании с другими методами

защиты и кодирования информации позволяет повысить надежность и качество системы связи в целом.

Литература

1. Лукин, А. Введение в цифровую обработку сигналов (математические основы) / Алексей Лукин. – Москва : Лаборатория Компьютерной графики и мультимедиа, МГУ, 2007. – PDF-файл. – 54 с. – URL : <https://audio.rightmark.org/lukin/dspcourse/dspcourse.pdf>. – Текст : электронный.
2. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение / Б. Скляр ; пер. с англ. Е. Г. Грозы [и др.] ; под редакцией А. В. Назаренко. – 2-е изд., испр. – Москва : Вильямс, 2003. – 1104 с.
3. Астретсов, Д. В. Общая теория связи. Лабораторный цикл : учебно-методическое пособие / Д. В. Астретсов, М. П. Трухин. – Екатеринбург : УрФУ, 2012. – 71 с.
4. Баринов, А. Ю. Перемежение в канальном кодировании : свойства, структура, специфика применения / А. Ю. Баринов. – DOI : 10.30898/1684-1719.2019.1.13. – Текст : электронный // Журнал радиоэлектроники. – 2019. – № 1. – С. 1–32.
5. Рабинер, Л. Р., Шафер, Р. В. Цифровая обработка речевых сигналов / Л. Р. Рабинер, Р. В. Шафер ; под ред. М. В. Назарова и Ю. Н. Прохорова ; пер. с англ. – Москва : Радио и связь, 1981. – 496 с., ил.

Поступила в редакцию 07.07.2024

Олег Константинович Сапрыкин, инженер, e-mail: sfsdc.01@mail.ru, т. +7 (903) 293-30-14.
(Акционерное общество «Научно-исследовательский институт точных приборов»).

Владислав Владимирович Кудинов, инженер-конструктор 1 категории, e-mail: vlad_kud77@mail.ru, т. +7 (977) 937-56-40.
(Акционерное общество «Особое конструкторское бюро Московского энергетического института»).

APPLICATION OF INTERLEAVING ALGORITHMS IN THE TRANSMISSION OF ANALOG INFORMATION IN A DIGITAL COMMUNICATION SYSTEM

O. K. Saprykin, V. V. Kudinov

The goal of the work is to develop a mathematical model of a digital communication system using interleaving algorithms when transmitting analog information. Study how they affect the quality of transmitted information under conditions of interference. Propose several options for interleaving algorithms and analyze their work, as well as provide recommendations for their use.

Key words: information, digital communication system, interleaving algorithm, interleaver.

References

1. Lukin, A. Introduction to digital signal processing (mathematical foundations) / Aleksey Lukin. – Moscow : Laboratory of Computer Graphics and Multimedia, Moscow State University, 2007. – PDF-file. – 54 p. – URL : <https://audio.rightmark.org/lukin/dspcourse/dspcourse.pdf> (date of access : 23.07.2024). – Text: electronic.
2. Sklyar, B. Digital communication. Theoretical foundations and practical application / B. Sklyar ; translated from English by E. G. Grozy [et al.]; edited by A. V. Nazarenko. – 2nd ed., cor. – Moscow : Williams, 2003. – 1104 p.
3. Astretsov, D. V. General theory of communication. Laboratory cycle : an educational and methodical manual / D. V. Astretsov, M. P. Trukhin. – Yekaterinburg : UrFU, 2012. – 71 p.
4. Barinov, A. Y. Interleaving in channel coding : properties, structure, specifics of application / A. Y. Barinov. – DOI : 10.30898/1684-1719.2019.1.13. – Text : electronic // Journal of Radioelectronics. – 2019. – № 1. – P. 1–32.
5. Rabiner, L. R. Digital processing of speech signals / L. R. Rabiner, R. V. Shafer ; edited by M. V. Nazarov and Y. N. Prokhorov. – Trans. from English. – Moscow : Radio and Communications, 1981. – 496 p., ill.

Oleg Konstantinovich Saprykin, engineer, e-mail: sfsdc.01@mail.ru, t. +7 (903) 293-30-14.
(Joint Stock Company «Scientific Research Institute of Precision Instruments»).

Vladislav Vladimirovich Kudinov, design engineer of the 1st category, e-mail: vlad_kud77@mail.ru, t. +7 (977) 937-56-40.
(Joint Stock Company «Special Design Bureau of the Moscow Power Engineering Institute»).