АЛГОРИТМ КАНАЛЬНОГО КОДИРОВАНИЯ ИНФОРМАЦИОННОГО ПОТОКА С ВНЕСЕННОЙ ИЗБЫТОЧНОСТЬЮ ДЛЯ БЕСПРОВОДНЫХ ЦИФРОВЫХ СИСТЕМ ПЕРЕДАЧИ ИНФОРМАЦИИ ПЕРСПЕКТИВНЫХ ПОКОЛЕНИЙ

В.В. Дубровский, *к.ф-м.н., доцент, Белорусская государственная академия связи, v.dubrovsky@bsac.by;*
А.И. Дулькевич, *Белорусская государственная академия связи, a.dulkevich@bsac.by;*
Е.В. Новиков, *к.т.н., доцент, Белорусская государственная академия связи, e.novikov@bsac.by.*

УДК 004.021

Аннотация. В данной статье представлен энергоэффективный подход к кодированию информационного сигнала, основанный на принципах нелинейной динамики. Метод заключается во введении в кодовую структуру избыточной информации, сформированной с использованием метрик Хэмминга. Авторы разработали алгоритмы кодирования и декодирования цифрового потока для устойчивости передачи данных. Результаты численного моделирования подтверждают эффективность предложенного подхода. Также даны практические рекомендации по применению метода в различных цифровых системах связи и обработки данных.

Ключевые слова: избыточность; помехоустойчивость; нелинейный код; расстояние Хэмминга; цифровая связь; декодирование; кодек; информационная безопасность; беспроводной канал.

ALGORITHM OF CHANNEL CODING OF INFORMATION STREAM WITH INTRODUCED REDUNDANCY FOR WIRELESS DIGITAL INFORMATION TRANSMISSION SYSTEMS OF ADVANCED GENERATIONS

V.V. Dubrouski, *Candidate of Physical and Mathematical Sciences, Associate Professor, Belarusian, State Academy of Telecommunications Minsk;*
H.I. Dulkevich, *Belarusian State Academy of Telecommunications Minsk;*
E.V. Novikov, *Candidate of Technical Sciences, Associate Professor, Belarusian State Academy of Telecommunications Minsk.*

Annotation. This paper presents an energy-efficient approach to information signal coding based on the principles of nonlinear dynamics. The method consists in introducing redundant information generated using Hamming metrics into the code structure. The authors have developed algorithms for encoding and decoding of the digital stream, for stability of data transmission. The results of numerical modeling confirm the effectiveness of the proposed approach. Practical recommendations on application of the method in various digital communication and data processing systems are also given.

Keywords: redundancy; noise immunity; nonlinear code; Hamming distance; digital communication; decoding; codec; information security; wireless channel.

Введение

Современные тенденции в совершенствовании методов кодирования цифровых сигналов обозначены тремя основными направлениями:

1) нейросетевые модели и *ML*-методы (machine learning – машинное обучение) кодирования [1, 2]; в эту группу можно отнести и методы на основе генеративно-состязательных сетей (*GAN*) [3];

2) методы полярного кодирования [4, 5] и неортогональные методы множественного доступа (*NOMA*) [6];

3) кодирование методами нелинейной динамики [7-9].

Имеется также ряд гибридных направлений кодирования, связанных с попытками перенести устойчивость биологических объектов в теорию кодирования сигналов или с применением квантовых эффектов. Однако в контексте обсуждения следует разграничивать термин «кодирование» между несколькими самостоятельными областями знаний, что определяет следующие задачи:

- преобразование, модифицирование, адаптация и приведение информации к нужной форме (например, кодирование источника сигнала);
- кодирование в качестве решения задачи обеспечения информационной безопасности (например, криптография и криптоанализ);
- помехоустойчивое кодирование как один из элементов канального кодирования сигналов.

Кодирование информационного сигнала методами нелинейной динамики предоставляет широкие возможности для совмещения информационной безопасности цифрового потока, передаваемого по открытым каналам связи, и помехоустойчивости. Алгоритм, представленный в работе [1], гибок в управлении характеристиками кодера и эффективно исправляет одиночные ошибки. Устойчивость в условиях действия помех в указанном случае достигается за счет внесения избыточной информации методами нелинейной динамики, что существенно усложняет декодирование сигнально-кодовой конструкции (СКК) на стороне приема и обработки. Исследования различных вариантов формирования проверочных блоков в СКК, с последующей их обработкой и декодированием показало, что эффективным с точки зрения вычислительных затрат может оказаться вариант инжектирования блока избыточной информации, формируемого линейными методами на основе метрик Хэмминга. Одно из существенных отличий представляемого алгоритма канального кодирования от [1] состоит в том, что:

- вычислительно алгоритм несколько упрощается, что призвано снизить энергопотребление кодека;
- часть энергетических затрат на вычисления возлагается на кодер, т. е. передающую сторону, и, соответственно, снижает затраты декодера.

Алгоритм нелинейного формирования помехоустойчивого сигнала с вносимой избыточностью

Общий замысел нелинейного кодирования с внесенной избыточностью на основе метрик Хэмминга состоит в подборе такого проверочного символа v_k, который в смысле Хэмминга находился бы максимально близко к закодированному значению, но максимально далеко к потенциально искаженным значениям [11]. Исходим из того, что в блоке поражается один бит информации.

Аналитически алгоритм канального кодирования состоит в следующем:

1. На k-м такте формируется кодированный отсчет:

$$h_k(i) = f_i(v_{k-1};\ h_{k-1}),\ i = x_k, \tag{1}$$

где: $h_k(i)$ и h_{k-1} – текущий и предыдущий кодированный отсчет соответственно; $f_i(\cdot)$ – одна из $N = 2^Q$ кодирующих функций; Q бит – длина информационного блока; v_{k-1} – избыточный отсчет в предыдущем блоке; $x_k \in [1; N]$ – значение информационного сигнала к началу k-го такта работы кодера, определяющее индекс i порождающей функции.

2. Определяем сигнальный вектор кодера:

$$\boldsymbol{S}_{h,\,k} = f_i(v_{k-1};\ h_{k-1}),\ \forall i = \overline{1,\ N}. \tag{2}$$

Физически $\boldsymbol{S}_{h,\,k}$ – множество всех возможных значений сигнального отсчета h_k, включающее в себя кодированное значение $h_k(i)$.

3. Формируем образ потенциальных одиночных ошибок Ξ_k:

$$\boldsymbol{\Xi}_k = BitChange(h_k,\ n),\ \forall n = \overline{1,\ Q}, \tag{3}$$

где: n – номер бита в двоичном представлении h_k.

4. Для всех $\boldsymbol{\Xi}_k$, принадлежащих $\boldsymbol{S}_{h,\,k}$, определяем субблок избыточной информации v_k согласно условию:

$$v_k \Leftarrow \{dist[\boldsymbol{\Xi}_k,\ \ v_k] = max, \tag{4}$$

где: $dist[a,\ b]$ – расстояние Хэмминга между кодовыми словами a и b.

Для практически применимого случая, когда большинство значений из образа ошибок (3) принадлежат множеству (2), проверочный субблок равен значению $h_k(i)$, подвергнутому битовой инверсии, что в арифметическом представлении соответствует смене знака значения и вычитании из него единицы:

$$v_k = BitInverse[h_k(i)] = -h_k(i) - 1, \tag{5}$$

где: операция $BitInverse(\cdot)$ – побитовая инверсия аргумента.

Таким образом, каждый блок $[h_k; v_k]$ кодированной информации состоит из двух подблоков (субблоков), первый из которых сигнальный (1), второй – проверочный (4), (5). Проверочный субблок обеспечивает исправление неоднозначностей декодирования в случае их наличия, а также идентификацию местоположения ошибки, что в последующем позволит скорректировать поток и извлечь информационный отсчет. Индекс i в при v_k не указан, поскольку значение зависит от состояния информационного отсчета косвенно и его величина в различные моменты времени может быть различной, исходя из общего условия (4).

Структурная схема кодера представлена на рис. 1. Исходными данными, обеспечивающими корректную работу кодера, являются следующие величины:

1. M, Q – разрядность счета и длина информационного блока соответственно.

2. Величина $N = 2^Q$, определяющая количество нелинейных порождающих код функций (НПФ).

3. h_1, v_1 – начальное состояние кодера; оно должно быть известно как на приемной, так и на передающей стороне;

4. Вид НПФ: кусочно-линейные, квадратичные, кубические полиномы.

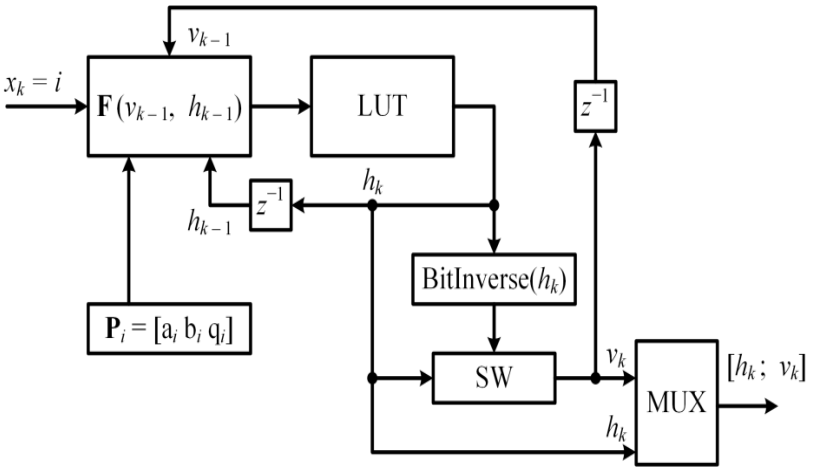

Рисунок 1

На рис. 1 представлен алгоритм кодирования информационного потока, где: *LUT* – *LookupTable*, таблица соответствия, на входе которой формируется множество возможных значений кодирующей функции, а на выходе наблюдается значение h_k; *BitInverse* (h_k) – функция, реализующая инверсию бит входного значения-аргумента; *SW* – коммутатор, осуществляющий передачу на выход одно из двух входных значений; *MUX* – мультиплексор.

На вход кодера поступает последовательность информационных символов x_i, управляющая выходным состоянием блока *LUT*, на входе которого действует N значений, определенных в поле Галуа $\left\{2^M\right\}$:

$$h_k(i) = \boldsymbol{F}(v_{k-1},\ h_{k-1}) = \boldsymbol{P} \times \begin{bmatrix} v_{k-1} \\ h_{k-1} \\ 1 \end{bmatrix}, \quad i = x_k, \tag{6}$$

где: x_k – информационное значение на k-м такте.

Аргументами системы порождающей функции ***F(·)*** являются два предыдущие состояния системы: кодированное значение h_{k-1} и избыточный отсчет v_{k-1}. Матрица коэффициентов ***P*** системы порождающих функций хорошо описана в (11) и определяется общим выражением:

$$\mathbf{P} = \begin{bmatrix} a_0 & b_0 & q_0 \\ & \cdots & \\ a_{N-1} & b_{N-1} & q_{N-1} \end{bmatrix} \tag{7}$$

где: a_i, b_i, q_i – произвольные коэффициенты, задающие характер поведения нелинейной системы.

Извлечение информационного сигнала нелинейным декодером

В отличие от алгоритмов, представленных в [10], кодер (4) позволяет вычислительно упростить алгоритм декодирования, поскольку дополнительный субблок v_k не является результатом вычисления согласно выражению (6), а определен через битовую инверсию h_k в случае отсутствия неоднозначностей или передается непосредственно без изменений, если наблюдается не более двух неоднозначностей. При этом последующие отсчеты h_{k+1}, h_{v+1} зависят от кодированного значения h_k и искусственно синтезированного значения v_k.

Поскольку на стороне приема и обработки полезного сигнала априорно не известен факт наличия или отсутствия в декодируемом блоке ошибки, а в случае ее выявления не известна позиция, то алгоритм декодирования проверяет на справедливость три гипотезы.

Гипотеза 1: «ошибка в блоке отсутствует».

Проверка гипотезы реализуется через проверочную таблицу N значений, где $N = 2^Q$ – количество порождающих функций в уравнении (6).

$$\boldsymbol{S}_{h,\,k} = \boldsymbol{F}(v_{k-1},\ h_{k-1}) = f_i[v_{k-1},\ h_{k-1}],\quad \forall i = \overline{1,\ 2^Q}, \tag{8}$$

где: $\boldsymbol{S}_{h,\,k}$ – вектор декодирования субблока h_k соответственно; $i = \overline{1, 2^Q}$ – номер функции в системе НПФ, определяющий состояние информационного отсчета; значения h_{k-1} фиксированы и точно известны на k-м такте работы декодера.

Декодер извлекает информационное значение согласно выражению, непосредственно следующему из (4) или (5) при рассмотрении упрощенного варианта кодирования:

$$x_k^* = i \Leftarrow \begin{cases} \boldsymbol{S}_{h,\,k}(i) = BitInverse[v_k]; \\ \boldsymbol{S}_{h,\,k}(i) = v_k, \end{cases} \tag{9}$$

где: x_k^* – оценочное значение информационного сигнала на k-м такте работы.

Декодер определяет, при каком i одно из вычисленных согласно (8) значений совпадает с проверочным v_k. Если такой индекс существует, то делается вывод, что ошибки нет и декодер переходит к обработке следующего блока. На выход декодера поступает оценочное значение информационного отсчета $x_k^* = i$ и выносится решение о подтверждении гипотезы.

В случае, когда равенства (9) ни при каких индексах не соблюдаются или одновременно соблюдаются оба равенства, гипотеза опровергается, и декодер делает вывод, что в блоке $[h_k;\ v_k]$ присутствует ошибка, и переходит к проверке второй гипотезы.

Гипотеза 2: «ошибка присутствует в блоке h».

На k-м такте работы декодера формируем образ ошибок $\Xi_h(k)$ – таблицу размером $3 \times N$ – следующим образом:

$$\Xi_k\,(k):\begin{cases} h_{k,sup}\,(i) = BitChange[h_k],\ \ \forall i = \overline{1,\ 2^Q}; \\ i_{h,sup} = \{S_{h,k}\,(i) = h_{k,sup}\,(i)\},\ \forall i = \overline{1,\ 2^Q}; \\ d_{ham}(i) = D\left[h_{k,sup}\,(i), v_k\right],\ \ \forall i = \overline{1,\ 2^Q}; \end{cases} \tag{10}$$

где: функция *BitChange* поочередно инвертирует каждый бит аргумента в двоичном представлении; $h_{k,\,\sup}(i)$ – вектор из N значений, которые определяют предполагаемые состояния субблока h_k; $i_{h,sup}$ – вектор индексов совпадений между значениями, вычисленных согласно первому уравнению системы (8), и значениями $h_{h,sup}$ для каждого i; $d_{ham}(i)$ – вектор расстояний Хэмминга между вычисленными согласно (5) значениями и значением v_k для каждого i; $D[\alpha, \beta]$ – функция вычисления расстояния Хэмминга между аргументами α и β в двоичном представлении.

Вектор $i_{h,sup}$ при определенных i содержит пустые множества, поскольку совпадений между предполагаемым значением $h_{h,sup}(i)$ и вектором декодирования $\boldsymbol{S}_{h,k}$ не наблюдается. Отметим, что третье выражение в фигурных скобках (10) при условии безошибочности принятого значения v_k (что это соответствует условию гипотезы) даст либо нулевое, либо максимальное возможное значение M для определенного непустого значения $i_{h,\,\sup}$, что позволяет определить справедливость гипотезы согласно выражению:

$$x_k^* = i \Leftarrow \begin{cases} d_{\text{ham}}(i) = M \cup d_{\text{ham}}(i) = 0; \\ h_{k,\,\sup}(i) \notin \varnothing. \end{cases} \tag{11}$$

Алгоритм проверки гипотезы (11) следует читать так: если оценочное значение $h_{k,\,\sup}(i)$ не является пустым множеством, осуществляется поиск тех i, при которых метрика Хэмминга либо максимальна – M, либо минимальна. В ином случае гипотеза не подтверждается, и алгоритм переходит к проверке следующей гипотезы [12].

Гипотеза 3: «ошибка присутствует в блоке v».

Для рассматриваемого случая логика проверки гипотез существенно упрощается: по h либо точно, либо приблизительно – в случае наличия неоднозначностей – определяются возможные индексы функций, то есть информационное состояние $x_k^* = i$. Затем i либо точно подтверждается, либо уточняется по предполагаемому значению v через образ ошибок $\Xi_v(k)$:

$$\Xi_v(k):\ \left\{ v_{k,\,\sup}(i) = \text{BitChange}\left[v_k\right],\ \ \forall i = \overline{1, 2^Q}. \right. \tag{12}$$

Аналитически алгоритм проверки гипотезы формализуется следующим образом:

$$x_k^* = i \Leftarrow \begin{cases} \mathbf{S}_{h,\,k}(i) = h_k; \\ h_k = \mathrm{BitInverse}\left[v_{k,\,\sup}\right] \cup h_k = v_{k,\,\sup}. \end{cases} \tag{13}$$

Алгоритм (13) определяет индекс совпадения с элементом проверочной либо значения $v_{k,\,\sup}(i)$, либо его же, но побитно инвертированного.

Структурная схема алгоритма декодирования на основе проверки гипотез представлена на рис. 2 [13].

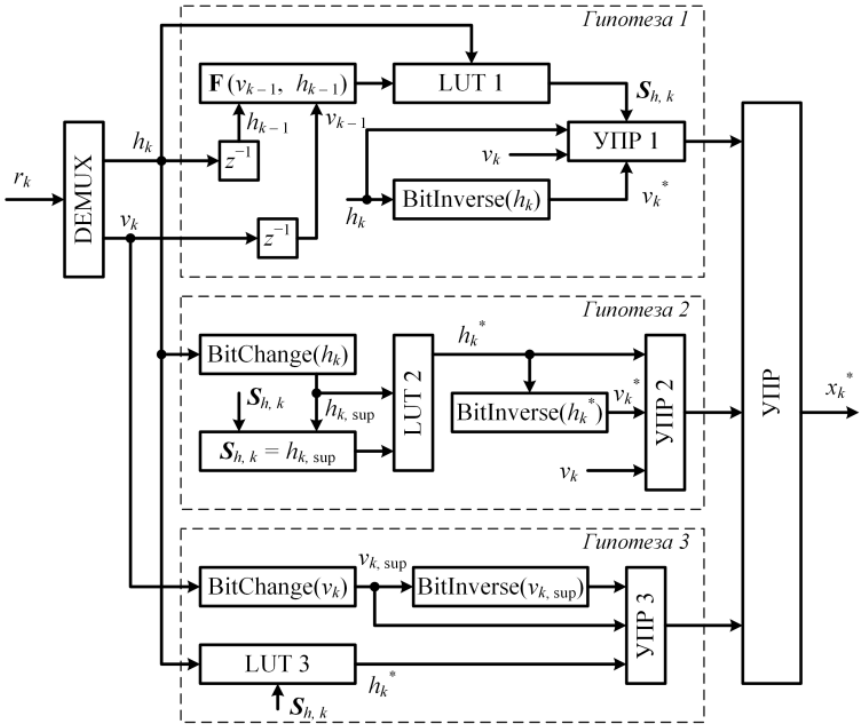

Рисунок 2

В таблице соответствия *LUT*1 формируется вектор вычисленных значений $\boldsymbol{S}_{h,k}$, т.е., таблица соответствия, позволяющая декодеру корректно сопоставить принятое значение h_k с одним из возможных значений. *LUT*1 обеспечивает реализацию выражения (8) [14]. Устройство принятия решения УПР 1 в явном виде сопоставляет согласно алгоритму (9) четыре состояния, два из которых получены из канала связи, два – вычислены на стороне приема и обработки: h_k, v_k, $\mathrm{h}_\mathrm{k}^* \Leftarrow h_k = \boldsymbol{S}_{h,\,k}(i)$ и v_k^*.

Таблица соответствия *LUT*2 содержит множество предполагаемых значений $h_{k,\,\sup}$, а также индексы совпадений $\boldsymbol{S}_{h,\,k}(i) = h_{k,\,sup}$, что позволяет определить оценочное значение h_k^* и подтвердить или опровергнуть гипотезу 2, имея точно принятый избыточный отсчет v_k. Устройство принятия решения УПР 2 дает либо пустое множество, если гипотеза опровергнута, либо значение декодированного отсчета x_k [15, 16].

Таблица соответствия $LUT3$ содержит единственную строку $\boldsymbol{S}_{h,k}$ для всех возможных индексов $i=\overline{1,2^Q}$, благодаря чему происходит точная или приблизительная оценка состояния h_k. Устройство принятия решения УПР2 определяет справедливость гипотезы и дает на выходе оценочное значение x_k.

Оконечное устройство принятия решения УПР определяет, на каком из трех входов действует непустое множество и выдает его выход канального декодера.

Заключение

1. Представленный в работе алгоритм декодирования является самосинхронизирующимся: в случае возникновения ошибки кратность больше единицы алгоритм теряет декодирующую способность, однако восстанавливает ее через два такта после того, как в блоках $[h_k;\ v_k]$ начинают наблюдаться одиночные ошибки.

2. Эффективность кодера возрастает, когда система НПФ формирует векторы с множеством совпадающих чисел, кроме основного. В этом случае меньше значений, к которым надо подбирать проверочный блок (4) – больше отбрасываемых значений.

3. Проверочный блок чаще всего – это инвертированный с вычетом единицы основной блок. В случае выполнения п. 2 могут быть случайные значения. Если в (4) четное число значений, то самим алгоритмом вносится случайность, что существенно усложняет структуру фазовых переходов нелинейной системы и не позволяет третьим лицам проводить анализ НПФ.

4. Важное отличие предлагаемого алгоритма кодирования-декодирования от известных состоит в том, что проверочный отсчет участвует в вычислении последовательности. Его влияние на структуру фазовых переходов СКК приводит к еще большему усложнению нелинейного характера результирующего сигнала. Однако это накладывает и определенные ограничения на структуру декодера, поскольку необходимо не просто использовать проверочный субблок v_k для исправления и идентификации возникающей ошибки в отсчете h_k, но и корректировать значение v_k.

5. При практической реализации декодера с учетом проверки гипотез удобнее ввести три специальных флага – 1, 2, 3. Если флаг равен 1 (гипотеза 1), то позиция ошибки не играет роли, поскольку в данном случае декодер проверяет лишь наличие/отсутствие в блоке $[h_k;\ v_k]$ ошибки. Если ее нет, то выводится декодированное значение x_k в отсутствие ошибок; если информационный индекс – пустое множество, то ошибка есть в одном из субблоков h_k или v_k, и необходима проверка гипотез.

Если флаг равен 1 (гипотеза 1) и x_k – непустое множество, то позиция указывает на ошибку в h_k. Если флаг равен 2 (гипотеза 2) и x_k – непустое множество, то позиция указывает на ошибку в v_k. В случае, когда при проверке той или иной гипотезы $x_k \in \varnothing$, декодер выносит решение, что гипотеза ошибочна.

6. Перед программированием конкретного устройства требуется проверка системы НПФ на отсутствие аномальных срабатываний на ошибки и точного определения гипотез. Проверка должна осуществляться на длительных интервалах наблюдения, поскольку даже для 8-битных устройства и $Q=8$ количество

комбинаций равно $256^{3\times256} = 2^{6144}$, что намного порядков превышает максимальную длину ключа в стандарте асимметричного шифрования *RSA*.

7. Фундаментальный смысл предложенного в статье алгоритма канального кодирования состоит в совмещении задачи обеспечения информационной безопасности сигнала и его помехоустойчивости.

Литература

1. Prabavathi V. Lamport Blum Shub Signcryptive Extreme Learning Machine for Secure Transmission of Digital Images // Procedia Computer Science, 2025. – V. 215. – P. 120-130.
2. Kishore M.K. Secure lightweight digital twin (DT) technology for seamless wireless communication in vehicular ad hoc network// Computers and Electrical Engineering, 2025.
3. Virupakshappa A. Secure communication over wireless sensor network using image steganography with generative adversarial networks // Measurement: Sensors, 2022.
4. Zheng, Y. Highly reliable DHR-based polar compilation code communication method // Computer Networks, 2024.
5. An Z.A secure scheme based on polar-coded and 3D mixed domain scrambling in W-band MMW-RoF system // Optics Communications, 2024.
6. Liang, H. User scheduling for non-orthogonal multiple access-enabled 5G aeronautical mobile airport communications system // Physical Communication, 2024.
7. Chen Q. Physical layer encryption based on digital chaos in THz wireless communication // Optical Fiber Technology, 2025.
8. Wang J. Neural network and Sparse identification of Nonlinear Dynamics Integrated Algorithm for Digital Twin identification // IFAC-PapersOnLine, 2023.
9. Zhao M. Fully distributed self-triggered secure consensus for nonlinear multiagent systems with sequential communication link scaling attacks // Applied Mathematics and Computation, 2025.
10. Дубровский В.В., Лакизо П.Ю. Нелинейный алгоритм помехоустойчивого кодирования сигнала на основе вставки синтетических корректирующих блоков // Проблемы инфокоммуникаций, 2025. – № 1 (21). – С. 10-19.
11. Дубровский В.В., Дулькевич А.И., Попова М. С. Цифровой адаптивный алгоритм кодирования информации методами нелинейной динамики для перспективных систем подвижной цифровой связи // Проблемы инфокоммуникаций, 2024. – № 1 (19). – С. 12-24.
12. Дубровский В.В., Половеня С.И. Алгоритмы защиты информации в системах телекоммуникаций методами нелинейной динамики // Вестник связи, 2016. – № 1(135).
13. Дубровский В.В., Дулькевич А.И. Алгоритм каскадного кодирования информационного потока для защищенных систем передачи цифровых данных // Проблемы инфокоммуникаций, 2024. – №2 (20). – С.17-25
14. Листопад Н.И., Козел В.М., Дубровский В.В., Горбачев К.Л. Теоретические основы цифровой радиосвязи. Учебник, гриф МО РБ Мн.: БГУИР, 2012. – 522 с.
15. Зеневич АО., Новиков Е.В., Матковская Т.А. Способ выявления утечки данных из волоконно-оптических линий связи с использованием ответвителя-прищепки // Проблемы инфокоммуникаций, 2022. – № 1 (15). – С.5-12
16. Гулаков И.Р., Зеневич А.О., Матковская Т.А., Новиков Е.В. Оценка возможности формирования канала утечки информации из оптического волокна тепловым воздействием // Труды учебных заведений связи, 2024. – гл. 10, № 6. – С. 19-25.