УДК 004.056.3

# РЕКУРРЕНТНЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ УСЕЧЕННОЙ СВЕРТКИ ПОЛИНОМОВ НАД ПОЛЕМ ГАЛУА И ЕГО АППАРАТНАЯ РЕАЛИЗАЦИЯ

**Рахман П.А.**

*ФГБОУ ВПО «Уфимский государственный нефтяной технический университет», филиал в г. Стерлитамаке, e-mail: pavelar@yandex.ru*

В данной статье рассматривается алгоритм вычисления усеченной свертки полиномов над полем Галуа. Вычисление усеченной свертки полиномов является частью алгоритма декодирования информационного кадра при применении кодов Рида-Соломона. Усеченная свертка полиномов необходима для вычисления значения искаженных символов в информационном кадре при использовании метода Форни. В статье дается определение полинома, усеченной свертки полиномов и формула для ее вычисления над полем Галуа $GF(p^m)$. Также в статье рассматривается алгоритм вычисления значения усеченной свертки полиномов над полем Галуа $GF(2^m)$ при заданном аргументе и предлагаемая автором аппаратная реализация алгоритма.

**Ключевые слова: коды Рида-Соломона, поле Галуа, усеченная свертка, полином**

# ALGORITHM FOR COMPUTING THE TRUNCATED CONVOLUTION OF THE POLYNOMIALS OVER GALOIS FIELD AND ITS HARDWARE IMPLEMENTATION

**Rahman P.A.**

*Ufa State Petroleum Technological University, Sterlitamak branch, e-mail: pavelar@yandex.ru*

This paper deals with recurrent algorithm for computing the truncated convolution of the polynomials over Galois field. Calculation of truncated convolution is a part of the decoding algorithm for information frame, encoded on application of Reed-Solomon codes. Truncated convolution of the polynomials is necessary in the algorithm based on Forney method for computing the error-values of corrupted symbols in information frame. In this paper the definition of polynomial, truncated convolution of the polynomials over Galois field $GF(p^m)$ and calculation formula for truncated convolution are given. Calculation algorithm for value of the truncated convolution at the given argument over Galois field $GF(2^m)$ and hardware implementation of the algorithm offered by author are also overviewed.

**Keywords: Reed-Solomon codes, Galois field, truncated convolution, polynomial**

В современном мире цифровых технологий каналы передачи данных и носители информации остаются далекими от совершенства. Магнитные и оптические носители информации чувствительны к физическим повреждениям, делающим невозможным чтение информации из отдельных участков на поверхности носителя. Кабельные и беспроводные линии передачи информации подвержены воздействию внешних помех, искажающих форму передаваемых сигналов и тем самым делающих невозможным однозначное распознавание информации на стороне приемника.

В настоящее время в системах хранения и передачи данных применяют различные технологии информационного резервирования с применением специальных алгоритмов кодирования на базе корректирующих кодов, в частности, кодов Рида-Соломона [1], которые за счет использования избыточной информации делают возможным исправление искажений. Однако, алгоритмы кодирования и декодирования информации с применением кодов Рида-Соломона достаточно нетривиальны, и для эффективной программной или аппаратной реализации требуется достаточно глубокая математическая проработка с применением алгебры конечных полей Галуа [2]. В частности, алгоритм декодирования состоит из блока вычисления синдрома искажений, полинома локаторов искажений, самих локаторов искажений и значений искажений, и все блоки имеют дело с полиномами, заданными над конечным полем Галуа. Особое место занимает блок вычисления значений искажений на базе метода Форни, использующего усеченную свертку полиномов, заданных над полем Галуа, и здесь требуется особый подход для построения эффективного алгоритма вычисления значения усеченной свертки полиномов при заданном аргументе.

В рамках научных исследований автора в области надежности систем хранения, передачи и обработки данных [3-9], а также в области информационного резервирования [10] возникла научная задача разработки эффективного алгоритма вычисления усеченной свертки полиномов $\Psi(x)$ и $\xi(x)$, заданных над полем Галуа $GF(p^m)$, при заданном аргументе $x$, и аппаратной реализации для поля Галуа $GF(2^m)$.

**Полином над полем Галуа**. Пусть задано поле Галуа $GF(p^m)$. Тогда полиномом, заданным над полем Галуа, будем называть функцию:

$$\Psi(x) = \Psi_{k-1}x^{k-1} + \ldots + \Psi_1 x + \Psi_0 = \sum_{i=0}^{k-1} \Psi_i x^i;$$

$$\Psi_i \in GF(p^m); \quad i = 0 \ldots k-1. \qquad (1)$$

Над полиномами можно производить алгебраические операции, также на место переменной $x$ можно подставлять конкретное значение, являющееся элементом поля Галуа $GF(p^m)$, и вычислять значение функции. При пересчете коэффициентов полинома при выполнении операции или вычислении значении функции, строго соблюдаются правила арифметики для поля $GF(p^m)$.

**Сумма полиномов.** Сложение полиномов выполняется путем сложения по правилам арифметики поля Галуа $GF(p^m)$ соответствующих коэффициентов, стоящих перед переменной в одной степени. Отсутствующие в полиномах коэффициенты, считаются равными нулю.

Пусть заданы два полинома $\Psi(x)$ и $\xi(x)$. Тогда, сумма этих полиномов определяется следующим образом:

$$\Psi(x) + \xi(x) = \sum_{q=0}^{\max(k-1,l-1)} (\Psi_q + \xi_q) x^q;$$

$$\Psi_i, \xi_j \in GF(2^m); \deg(\Psi(x)) = k-1;$$

$$\deg(\xi(x)) = l-1. \qquad (2)$$

Пример. Сложим полиномы $\Psi(x) = 49x^2 + 50x + 51$ и $\xi(x) = 19x^2 + 93x + 1$, заданные над полем $GF(2^8)$, заданного с помощью неприводимого многочлена $p(x) = x^8 + x^4 + x^3 + x^3 + 1$. Используя арифметику поля $GF(2^8)$ для сложения коэффициентов полиномов, имеем: $(49 + 19)x^2 + (50 + 93)x + (51 + 1) = 34x^2 + 111x + 50$.

**Произведение полиномов.** Умножение полиномов осуществляется путем вычисления суммы результатов умножения одного полинома на каждый член другого полинома. Соответственно, степень результирующего полинома равна сумме степеней перемножаемых полиномов.

Пусть заданы два полинома $\Psi(x)$ и $\xi(x)$. Тогда, произведение этих полиномов определяется следующим образом:

$$\Psi(x) \cdot \xi(x) = \sum_{i=0}^{k-1} \Psi_i x^i \left( \sum_{j=0}^{l-1} \xi_j x^j \right) =$$

$$= \sum_{q=0}^{k+l-2} x^q \sum_{i+j=q} (\Psi_i \xi_j);$$

$$i = 0 \ldots k-1; \quad j = 0 \ldots l-1;$$

$$\Psi_i, \xi_j \in GF(2^m); \deg(\Psi(x)) = k-1;$$

$$\deg(\xi(x)) = l-1. \qquad (3)$$

Пример. Умножим полиномы $\Psi(x) = 49x^2 + 50x + 51$ и $\xi(x) = 19x^2 + 93x + 1$, заданные над полем $GF(2^8)$, заданного с помощью неприводимого многочлена $p(x) = x^8 + x^4 + x^3 + x^3 + 1$. Используя арифметику поля $GF(2^8)$ для сложения и умножения коэффициентов полиномов, получаем: $\Psi(x) \cdot \xi(x) = (49 \cdot 19)x^4 + (49 \cdot 93 + 50 \cdot 19)x^3 + (49 \cdot 1 + 50 \cdot 93 + 51 \cdot 19)x^2 + (50 \cdot 1 + 51 \cdot 93)x + 51 \cdot 1 = 100x^4 + 218x^3 + 31x^2 + 3x + 51$.

Особо отметим, что если мы будем подразумевать, что любые «отсутствующие» коэффициенты в полиномах, в том числе коэффициенты с индексами больше степени полиномов, равны нулю: $\forall i > k-1$: $\Psi = 0$ и $\forall j > l-1$: $\xi_j = 0$, то мы внутреннее суммирование по двум индексам можем преобразовать суммирование по одному индексу. Для этого мы изменим верхние границы для обоих индексов: $i \le q$ и $j \le q$, при этом мы ничего не теряем, поскольку $i \ge 0$, $j \ge 0$ и $i + j = q$. Случаи, когда $i > q$ (при $k - 1 > q$), или $j > q$ (при $l - 1 > q$), все равно никогда не пройдут условие $i + j = q$, так как $i \ge 0$ и $j \ge 0$. А при $i > k - 1$ (при $q > k - 1$) коэффициенты $\Psi_i = 0$, а при $j > l - 1$ (при $q > l - 1$) коэффициенты $\xi_j = 0$. Тогда в итоге мы получаем следующие три условия: $i + j = q$, $0 \le i \le q$ и $0 \le j \le q$.

Заметим, что оба индекса неотрицательны и ограничены одним и тем же верхним пределом, причем сумма индексов также равна верхнему пределу. Тогда, очевидно, можно избавиться от одного из индексов, попросту выразив его через другой: $j = q - i$, и мы ничего не теряем, поскольку $0 \le q - i \le q$, так как $0 \le i \le q$.

В итоге мы получаем следующую «оптимизированную» формулу произведения:

$$\Psi(x) \cdot \xi(x) = \sum_{q=0}^{k+l-2} x^q \left( \sum_{i=0}^{q} (\Psi_i \, \xi_{q-i}) \right);$$

$$\Psi_i, \xi_j \in GF(2^m); \forall i > \deg(\Psi(x)) = k-1: \Psi_i = 0;$$

$$\forall j > \deg(\xi(x)) = l-1: \xi_j = 0. \qquad (4)$$

**Усеченная свертка полиномов**. Пусть заданы два полинома $\Psi(x)$ и $\xi(x)$. Тогда, усеченной линейной сверткой полиномов будем называть остаток от произведения этих полиномов по модулю полинома $x^r$, иными словами $\Psi(x) \, o \, \xi(x) = (\Psi(x)\xi(x)) \bmod x^r$.

По сути, усеченная линейная свертка – это произведение полиномов, в котором «отброшены» члены, степень переменной $x$ в которых больше либо равна $r$:

$$\Psi(x) \circ \xi(x) = (\Psi(x)\xi(x)) \bmod x^r = \sum_{q=0}^{r-1} x^q \left( \sum_{i=0}^{q} \Psi_i \xi_{q-i} \right); \quad r \ge 2;$$

$$\Psi_i, \xi_j \in GF(p^m); \forall i > \deg(\Psi(x)) = k-1: \Psi_i = 0$$

$$\forall j > \deg(\xi(x)) = l-1: \xi_j = 0. \quad (5)$$

Пример. Найдем усеченную свертку полиномов $\Psi(x) = 49x^2 + 50x + 51$ и $\xi(x) = 19x^2 + 93x + 1$, заданных над полем $GF(2^8)$, по модулю полинома $x^2$. Получаем:

$$\sum_{q=0}^{1} x^q \left( \sum_{i=0}^{q} \Psi_i \xi_{q-i} \right) = \Psi_0\xi_0 + (\Psi_0\xi_1 + \Psi_1\xi_0)x = 51\cdot 1 + (51\cdot 93 + 50\cdot 1)x = 3x + 51.$$

**Вычисление значения усеченной свертки полиномов при заданном аргументе над полем Галуа $GF(p^m)$.** Выше мы определили, что усеченная свертка $\Omega(x)$ двух полиномов $\Psi(x)$ и $\xi(x)$, заданных над полем Галуа $GF(p^m)$, и при заданном $r$, определяется как:

$$\Omega(x) = (\Psi(x)\xi(x)) \bmod x^r = \sum_{q=0}^{r-1} x^q \left( \sum_{i=0}^{q} \Psi_i \xi_{q-i} \right).$$

Нетрудно заметить, что вычисление значения полинома-свертки при заданном аргументе по вышеприведенной формуле в общей сложности требует $r(r+1)/2 \sim r^2/2$ итераций для вычисления двойной суммы. Выполним ряд преобразований формулы:

$$\sum_{q=0}^{r-1} x^q \left( \sum_{i=0}^{q} \Psi_i \xi_{q-i} \right) = \Psi_0\xi_0 + (\Psi_0\xi_1 + \Psi_1\xi_0)x + \ldots + (\Psi_0\xi_{r-1} + \ldots + \Psi_{r-1}\xi_0)x^{r-1} =$$

$$= \Psi_0(\xi_0 + \xi_1 x + \ldots + \xi_{r-1}x^{r-1}) + \Psi_1 x(\xi_0 + \xi_1 x + \ldots + \xi_{r-2}x^{r-2}) + \ldots + \Psi_{r-1}x^{r-1}\xi_0 =$$

$$= \sum_{i=0}^{r-1} \Psi_i x^i \left( \sum_{j=0}^{r-1-i} \xi_j x^j \right).$$

Преобразуем формулу для использования рекуррентной схемы вычисления:

$$\Omega(x) = \left( \ldots(\Psi_{r-1}(\xi_0)x + \Psi_{r-2}(\xi_0 \oplus \xi_1 x))x + \ldots + \Psi_1 \left( \sum_{j=0}^{r-2} \xi_j x^j \right) \right) x + \Psi_0 \left( \sum_{j=0}^{r-1} \xi_j x^j \right).$$

Тогда, в первом приближении имеем следующую итерационную схему вычисления значения полинома-свертки $\Omega(x)$ при заданном аргументе $x^*$:

$$\begin{cases} \Omega^{(s)}(x^*) = (x^*)(\Omega^{(s-1)}(x^*)) + \Psi_{r-s} \left( \sum_{j=0}^{s-1} (x^*)^j \xi_j \right); \\ \Omega^{(0)}(x^*) = 0; \quad s = 1\ldots r; \quad r \geq 0. \end{cases} \quad (6)$$

Заметим, что хотя и итерационная процедура вычисляет $\Omega(x^*)$ за $r$ итераций, но на каждой итерации, требуется вычисление суммы $\sum_{j=0}^{s-1} (x^*)^j \xi_j$, количество слагаемых в которой с каждой итерацией $s = 1\ldots r$ только растет. Однако заметим, что количество слагаемых в сумме в точности совпадает с номером итерации, и можно не заново пересчитывать сумму на каждой итерации $s$, а лишь добавлять очередное слагаемое к сумме, вычисленной на предыдущей итерации $s$ 1. Обозначив $\sum_{j=0}^{s-1} (x^*)^j \xi_j$, имеем рекуррентное соотношение: $\xi^{(s)}(x^*) = \xi^{(s-1)}(x^*) + (x^*)^{s-1}\xi_{s-1}$, причем $\xi^{(0)}(x^*) = 0$. Очевидно, можно совместить обе итерационные схемы, на каждой итерации вычисляя сначала очередную $\xi^{(s)}(x^*)$, используя $\xi^{(s-1)}(x^*)$, а затем и $\Omega^{(s)}(x^*)$, используя $\Omega^{(s-1)}(x^*)$ и $\xi^{(s)}(x^*)$. Наконец, отметим, что для вычисления $(x^*)^s$ также необязательно на каждой итерации возводить аргумент $x^*$ в степень $s$, можно использовать соотношение $(x^*)^s = (x^*)^{s-1}x^*$, причем $(x^*)^0 = 1$. Тогда, имеем «оптимизированную» итерационную схему вычисления значения полинома-свертки $\Omega(x)$ при заданном аргументе $x^*$:

$$\begin{cases} \xi^{(s)}(x^*) = \xi^{(s-1)}(x^*) + (x^*)^{s-1}\xi_{s-1}; \\ \Omega^{(s)}(x^*) = (x^*)(\Omega^{(s-1)}(x^*)) + \Psi_{r-s}\xi^{(s)}(x^*); \\ (x^*)^s = (x^*)^{s-1}x^*; \\ \Omega^{(0)}(x^*) = 0; \quad \xi^{(0)}(x^*) = 0; \quad (x^*)^0 = 1; \quad s = 1\ldots r; \quad r \geq 0. \end{cases} \quad (7)$$

Пример. Вычислим значение свертки полиномов $\Psi(x) = 49x^2 + 50x + 51$ и $\xi(x) = 19x^2 + 93x + 1$, заданных над полем $GF(2^8)$, по модулю полинома $x^2$ (т.е. $r = 2$) при заданном аргументе $x^* = 64$, используя итерационную схему.

Имеем следующие начальные значения: $\Omega^{(0)}(x^*) = 0$; $\xi^{(0)}(x^*) = 0$; $(x^*)^0 = 1$.

Далее, на итерации $s = 1$ выполняем следующие вычисления:

$$\begin{cases} \xi^{(1)}(x^*) = \xi^{(0)}(x^*) + (x^*)^0 \xi_0 = 0 + 1 \cdot 1 = 1; \\ \Omega^{(1)}(x^*) = (x^*)(\Omega^{(0)}(x^*)) + \Psi_1 \xi^{(1)}(x^*) = 64 \cdot 0 + 50 \cdot 1 = 50; \\ (x^*)^1 = (x^*)^0 x^* = 64. \end{cases}$$

Наконец, на итерации $s = 2$ (последняя итерация) получаем:

$$\begin{cases} \xi^{(2)}(x^*) = \xi^{(1)}(x^*) + (x^*)^1 \xi_1 = 1 + 64 \cdot 93 = 223; \\ \Omega^{(2)}(x^*) = (x^*)(\Omega^{(1)}(x^*)) + \Psi_0 \xi^{(2)}(x^*) = 64 \cdot 50 + 51 \cdot 223 = 243; \\ (x^*)^2 = (x^*)^1 x^* = 205. \end{cases}$$

Тогда, искомое значение свертки полиномов равно $\Omega^{(2)}(x^*) = 243$.

Для проверки полученного значения, подставим заданный аргумент $x^* = 64$ в полином-свертку $(\Psi(x)\xi(x)) \bmod x^2 = 3x + 51$, полученный в рассмотренном выше примере для полиномов $\Psi(x) = 49x^2 + 50x + 51$ и $\xi(x) = 19x^2 + 93x + 1$. Имеем: $3 \cdot 64 + 51 = 243$. Таким образом, очевидно, результат вычисления по рекуррентной схеме верен.

**Аппаратная реализация для поля Галуа $GF(2^m)$**. Теперь рассмотрим аппаратную реализацию вычисления значения полинома-свертки $\Omega(x)$ при заданном аргументе $x^*$ над полем Галуа $GF(2^m)$. Аппаратная реализацию можно построить при помощи трех $m$-битных регистров, а также двух сумматоров и четырех умножителей элементов поля Галуа $GF(2^m)$. Аппаратная реализация быстрых сумматоров и умножителей элементов поля Галуа $GF(2^m)$ хорошо освещены в литературе [1, 2].

Ниже на рисунке представлена предлагаемая автором функциональная схема вычислителя значения полинома-свертки $\Omega(x)$ при заданном аргументе $x^*$.

Регистр $RG1$ используется для хранения вычисленного на предыдущей итерации степени аргумента $(x^*)^{s-1}$, регистр $RG2$, соответственно, $\xi^{(s-1)}(x^*)$, наконец, в регистре $RG3$ хранится $\Omega^{(s-1)}(x^*)$. Перед началом работы по сигналу сброса, поступающего на вход *Reset*, регистр $RG1$ устанавливается в значение (0…01), соответствующее элементу «1» поля $GF(2^m)$, а регистры $RG2$ и $RG3$ сбрасываются в нулевое значение.

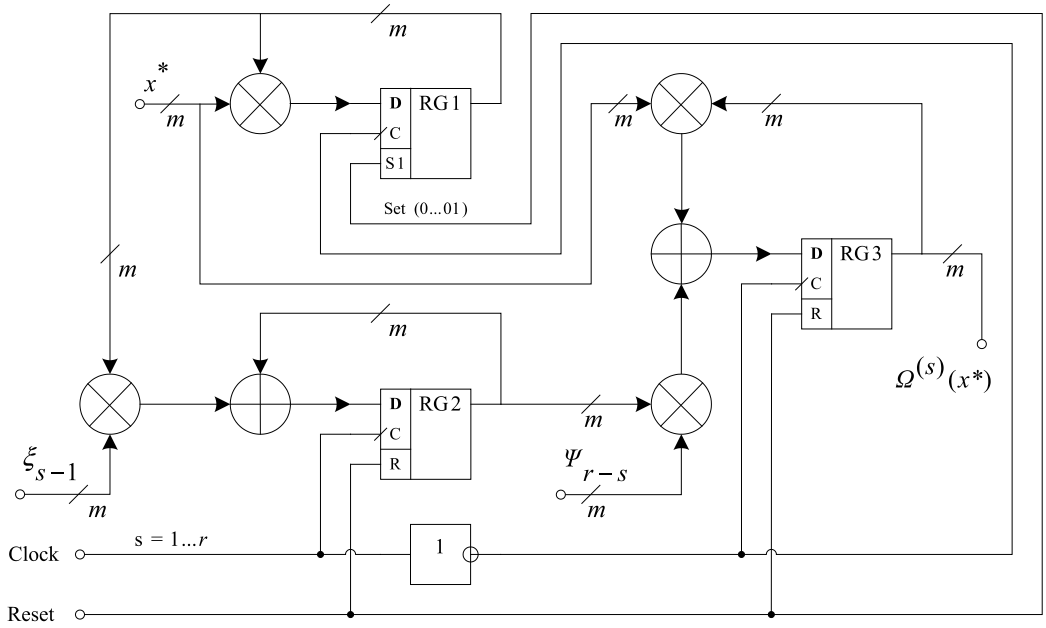
Далее на вход *Clock* поступают тактовые сигналы в течение итераций $s = 1 \ldots r$. Особо отметим, что вход синхронизации ($C$) регистра $RG2$ подключен к *Clock* напрямую, и, соответственно, регистр принимает информацию со своего информационного входа ($D$) по фронту тактового сигнала. Что же касается регистров $RG1$ и $RG2$ то они принимают информации по спаду тактового сигнала, так как и входы синхронизации подключены к *Clock* через инвертор.

Такая «двухтактная» синхронизации регистров позволяет на каждом такте $s = 1 \ldots r$ сначала вычислять и записывать в регистр $RG2$ значение $\xi^{(s)}(x^*) = \xi^{(s-1)}(x^*) + (x^*)^{s-1}\xi_{s-1}$, и только потом вычислять и записывать $\Omega^{(s)}(x^*) = (x^*)(\Omega^{(s-1)}(x^*)) + \Psi_{r-s}(\xi^{(s)}(x^*))$ в регистр $RG1$, и одновременно с этим также вычислять и записывать очередную степень аргумента $(x^*)^s = (x^*)^{s-1} x^*$ в регистр $RG3$, и таким образом, соблюдать правильную «очередность» вычислений на каждом такте.

В результате за $r$ тактов схема вычисляет искомое значение полинома-свертки $\Omega(x) = (\Psi(x)\xi(x)) \bmod (x^r)$ при аргументе $x^*$.

## Заключение

Таким образом, в рамках данной статьи рассматривается алгоритм вычисления усеченной свертки полиномов над полем Галуа. В статье приведено определение полинома, усеченной свертки полиномов и формула для ее вычисления над полем Галуа $GF(p^m)$. Также в статье рассматривается алгоритм вычисления значения усеченной свертки полиномов над полем Галуа $GF(2^m)$ при заданном аргументе и предлагаемая автором аппаратная реализация алгоритма.




*Функциональная схема последовательного вычислителя значения полинома-свертки* $\Omega(x) = (\Psi(x)\xi(x)) \bmod (x^r)$ *при заданном аргументе* $x^*$

Полученные результаты были использованы автором для разработки обучающей программы и лабораторных стендов для изучения студентами технических специальностей технологии кодирования информации при применении кодов Рида-Соломона.

**Список литературы**

1. Todd K. Moon. Error correcting coding: mathematical methods and algorithms. – Hoboken, New Jersey: John Wiley & Sons Inc., 2005.

2. Рахман П.А., Григорьева Т.В. Кодирование информации с применением кодов Рида-Соломона. – Уфа: Изд-во УГНТУ, 2015.

3. Рахман П.А., Шарипов М.И. Модель надежности двухузлового кластера приложений высокой готовности в системах управления предприятием // Экономика и менеджмент систем управления. – 2015. – Т. 17. № 3. – С. 85–102.

4. Рахман П.А., Шарипов М.И. Модели надежности каскадных дисковых массивов в системах управления предприятием // Экономика и менеджмент систем управления. – 2015. – Т. 17. № 3.1. – С. 155–168.

5. Рахман П.А. Коэффициент готовности трехуровневых локальных сетей передачи данных // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–3. – С. 463–466.

6. Рахман П.А. Показатели надежности восстанавливаемых систем с заданным порогом аварийного отключения // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–3. – С. 467–470.

7. Рахман П.А. Среднее время до потери данных двухдискового массива // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–4. – С. 603–607.

8. Рахман П.А. Коэффициент готовности системы обработки данных с основным и резервным узлами // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 9–4. – С. 608–611.

9. Рахман П.А. Модель надежности мажоритарной вычислительной системы на базе элементов с тремя состояниями // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 10–1. – С. 33–37.

10. Рахман П.А. Алгоритм выбора кратности исправляемых искажений для кодирования информации с применением кодов Рида-Соломона // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 10–2. – С. 208–212.