Сравнительный анализ методов K-RLE и LTC сжатия измерительной информации с контролируемыми потерями

К. К. Семенов

Высшая школа компьютерных технологий и информационных систем Санкт-Петербургского политехнического университета Петра Великого

semenov_kk@spbstu.ru

А. Б. Раимжанова

Высшая школа компьютерных технологий и информационных систем Санкт-Петербургского политехнического университета Петра Великого

raimzhanova.adele@gmail.com

Аннотация. **Требования, предъявляемые в метрологии к результатам измерений, накладывают ограничения на возможности их архивирования в случае большого объема получаемой информации с использованием методов сжатия с потерями. Внесение неконтролируемых искажений в данные противоречит условиям сохранности, надежности и достоверности результатов измерений. Указанные обстоятельства привели к появлению нового класса алгоритмов сжатия – методов сжатия с контролируемыми потерями. К данному моменту этот класс алгоритмов представлен в литературе методами K-RLE и LTC. Настоящий доклад посвящен теоретическому анализу данных подходов, а также исследованию их свойств и предельных характеристик – в том числе достижимости ими теоретического предела сжатия с контролируемыми потерями.**

Ключевые слова: сжатие информации, архивирование результатов измерений, информационно-измерительные системы, сжатие с контролируемыми потерями

I. Введение

В области информационно-измерительных систем быстрое развитие электроники привело к многократному росту количества получаемой и обрабатываемой информации, превышающему темпы совершенствования каналов связи и устройств хранения информации для встраиваемых систем. Превалирование объема передаваемых данных над пропускной способностью используемых каналов связи и размером соответствующих хранилищ приводит к неизбежным перегрузкам и переполнениям последних [1]. В связи с этим значительная доля современных информационно-измерительных систем вынуждена использовать различные алгоритмы сжатия передаваемых данных.

Принятая на сегодняшний день классификация делит методы сжатия данных на две большие группы: методы сжатия без потерь (lossless), обеспечивающие взаимно однозначные результаты архивации и разархивирования, и методы сжатия с потерями (lossy) [2–5], позволяющие добиться меньшего размера архива в обмен на частичные искажения данных (тем бóльшими, чем сильнее произведено сжатие). Промежуточное место между ними занимают методы сжатия с контролируемыми потерями, сочетающие отдельные черты как одной группы, так и другой и допускающие метрологический контроль.

Вопрос между степенью сжатия и возникающими потерями при архивировании результатов измерений в настоящее время системно в литературе, посвященной информационно-измерительным системам, не исследован. С одной стороны более высокая степень достигаемого сжатия побуждает широко использовать методы архивации с потерями, но с другой – сложность обеспечения метрологического контроля величины искажений, вносимых в сжимаемые результаты измерений, создает затруднения для их полноценного использования, поскольку предельные характеристики возникающих погрешностей зачастую принципиально не могут быть гарантированы из-за особенностей алгоритмов сжатия с потерями. В настоящее время первое обстоятельство в представленной дилемме превалирует над вторым, что, в конечном счете, приводит к довольно широкому применению методов сжатия с потерями для научных и измерительных данных, исходя из практических и экономических соображений [6].

Компромиссным решением, открывающим существенные возможности для метрологии, могут стать методы сжатия с контролируемыми потерями. Их преимущество состоит в эффективном сочетании высокой степени сжатия, достигаемой за счет допущения потерь, и в возможности при этом оставаться в рамках требований обеспечения единства измерений. Применение алгоритмов сжатия с контролируемыми потерями позволяет «обменять» незначительную часть точности на выигрыш в сжатии, если такой компромисс уместен с экономической точки зрения.

В настоящее время среди методов сжатия с контролируемыми потерями наибольший научный и практический интерес представляют алгоритмы K-RLE [7] и LTC [8], изначально разработанные применительно к беспроводным сетям сенсоров. Авторами данных работ предлагается вносить небольшие контролируемые искажения в измерительные данные с целью их последующего более эффективного сжатия. К настоящему моменту указанные методы являются основными представителями группы методов сжатия временных сигналов с контролируемыми потерями.

Данная статья направлена на сравнительный анализ методов сжатия K-RLE и LTC с контролируемыми потерями применительно к информационно-измерительным системам. Для этой цели выполнена оценка эффективности работы этих алгоритмов на реальных измерительных данных и вычислена величина достигаемого коэффициента сжатия, достигнутого за счет внесения в архивируемые данные погрешностей, не выходящих за отведенные для них пределы.

Содержание статьи разбито на следующие разделы: второй раздел посвящен описанию алгоритмов K-RLE и LTC; в третьем разделе представлены результаты сопоставительного исследования указанных методов с точки зрения возможности достижения ими теоретического предела сжатия с контролируемыми потерями; в четвертом разделе изложены результаты анализа полученных результатов.

II. Методы сжатия K-RLE и LTC применительно к информационно-измерительным системам

Алгоритмы K-RLE и LTC активно применяются в большинстве конфигураций беспроводных сетей сенсоров [7, 8]. Изначально к их разработке побуждало то обстоятельство, что узлам таких сетей, как правило, доступен лишь очень ограниченный объем оперативной памяти и при этом велика необходимость в снижении потребляемой мощности в совокупности с увеличением скорости обработки информации. Широко используемые на практике методы сжатия без потерь (LZ77, LZ78, LZW, S-LZW) активно задействуют оперативную память для работы, поэтому их применение не отвечает указанным обстоятельствам. Ниже представлено описание алгоритмов K-RLE и LTC применительно к их использованию в информационно-измерительных системах (впервые на русском языке).

Алгоритм K-RLE

Алгоритм K-RLE [7] является модификацией широко известного классического алгоритма RLE (Run-length encoding). В основе метода K-RLE лежит идея, что если текущий отсчет временного ряда результатов измерений попадает в заранее заданный диапазон, определяемый предшествующими значениями ряда, то хранение данного отсчета является избыточным и он может быть удален. Пусть Δ – величина дополнительного отклонения, которое допустимо внести в значения временного ряда. Тогда, если для текущего отсчета ряда d_i оказывается, что следующие за ним n значений попадают в промежуток [d_i–Δ, d_i+Δ], то все они не хранятся, а для значения d_i указывается, что оно повторяется (n+1) раз подряд – наподобие того, как это происходит в классическом алгоритме RLE. Таким образом, алгоритм K-RLE сводится к следующей последовательности действий.

Входные данные:
di – временной ряд, подлежащий сжатию,
i – индекс, принимающий значения от 1 до N,
N – размер сжимаемого массива отсчетов временного ряда,
Δ – заданная пользователем величина дополнительного отклонения, которое возможно внести в сжимаемые данные.

Выходные данные:
ряд кортежей $\langle c_j, f_j, r_j\rangle$, где
c_j – значения, включенные в архив,
f_j – бинарный индикатор, указывающий на то, что за текущим значением идут значения, которые могут быть распознаны, как равные ему,
n_j – количество подобных повторений,
$r_j = n_j$ при $f_j = 1$ и $r_j = \varnothing$ при $f_j = 0$ – число повторений значения, сохраняемого в архиве (указывается только если индикатор f равен 1),
j – индекс, принимающий значения от 1 до $M \le N$, где M – размер получаемого архива (отложенного также в количестве отсчетов).

1. Здесь и далее присвоить значения в том порядке, как перечислена последовательность операций: i:=1, j:=1, c_j:=d_i, n_j:=0, f_j:=0.
2. Если i+n_j совпадает с N, то выйти из процедуры.
3. Если $d_{i+n_j+1} \notin [d_i - \Delta, d_i + \Delta]$, то i:=i+n_j+1, j:=j+1, c_j:=d_i. В противном случае f_j:=1, n_j:=n_j+1.
4. Вернуться к пункту 2.

Пусть s = 64 – размер памяти, занимаемый каждым из отсчетов сжимаемого временного ряда (в битах). Тогда размер исходного массива составит $R_0 = N \cdot s$ бит, а размер архива, полученного по приведенной процедуре, не превысит $R_1 = M \cdot (s+1) + (N-M) \cdot \lceil \log_2(N-M) \rceil$ бит, где один бит добавляется к каждому сохраненному значению на представление в памяти значения индикатора f наличия повторных значений, идущих за текущим значением, а размер памяти, необходимый для сохранения числа повторений n не может быть больше $\lceil \log_2(N-M) \rceil$, где $\lceil \cdot \rceil$ – это результат округления числа до целого в большую сторону. Коэффициент сжатия CR (compression ratio) оказывается равна

$$CR_{K-RLE} = \frac{R_0}{R_1} = \frac{N/M}{1+\frac{1}{s}\cdot\left(\frac{N}{M}-1\right)\cdot(\lceil \log_2(N-M) \rceil+1)}. \qquad (1)$$

Эффективность сжатия оказывается тем выше, чем больше близких друг к другу значений оказывается в сжимаемом временном ряду. Все значения, после передачи восстановленные из архива, с гарантией не будут отличаться от исходных на величину, большую Δ.

Процедура восстановления сжатой последовательности из архива, полученного с применением алгоритма K-RLE, осуществляется по следующей последовательности действий (при тех же обозначениях, что были использованы выше).

1. Здесь и далее присвоить значения в том порядке, как перечислена последовательность операций: i:=1, j:=1.
2. Если j равно M+1, то выйти из процедуры.
3. Если f_j равно 0, то d_i := c_j, i:=i+1, j:=j+1. В противном случае при всех q, пробегающих значения 0, 1, …, (n_j–1), выполнить присвоения d_{i+q} := c_j, затем присвоить i:=i+n_j, j:=j+1.
4. Вернуться к пункту 2.

Алгоритм LTC

В основе алгоритма LTC [8] лежит поиск линейных трендов во временном ряду, подлежащем сжатию.

Пусть, как и выше, d_i – это значения отсчетов временного сигнала, подлежащего сжатию, Δ – величина дополнительного отклонения, которое допустимо внести в значения временного ряда. Последовательность точек (i, d_i) заменяется последовательностью значений (j_k, c_k), такой, что все значения d_i оказываются отстоящими от кусочно-линейной функции, чьими узлами выступают указанные точки (j_k, c_k), не более чем на Δ. Как и раньше, индекс i последовательно принимает все значения от 1 до N. Индекс k же пробегает значения от 1 до M, значения j_k совпадают с теми значениями индекса i (позициями в сжимаемом одномерном массиве, позициями узлов кусочно-линейной функции), на которых должны быть размещены значения c_k при разархивировании. Пример построения последовательности значений c_k иллюстрирует Рис. 1 [8].

Принцип определения значений c_k, представляющих собой архив исходной последовательности, в оригинальной статье [8] заключается в следующем. Для текущего значения d_i строится промежуток $[d_{i+1}–\Delta, d_{i+1}+\Delta]$ и две прямые $l = l(k)$ (миноранта) и $u = u(k)$ (мажоранта), которые соединяют точку (i, d_i) с точками $(i+1, d_{i+1}+\Delta)$ и $(i+1, d_{i+1}–\Delta)$ соответственно (см. Рис. 1а), аргумент k принимает значения из области возможных значений индекса i. Далее рассматривают идущее следом значение d_{i+2}. Если данное значение попадает в область между l и u, то выполняют их переопределение: прямую u опускают вниз так, чтобы она проходила через ту точку из $(i+1, d_{i+1}+\Delta)$ и $(i+2, d_{i+2}+\Delta)$, что оказалась лежащей более низко; прямую l поднимают вверх так, чтобы она проходила через ту точку из $(i+1, d_{i+1}–\Delta)$ и $(i+2, d_{i+2}–\Delta)$, что оказалась лежащей более высоко (см. Рис. 1б). Для следующих значений $d_{i+3}, d_{i+4}, \ldots, d_{i+q+1}$ сжимаемой последовательности поступают аналогичным образом до тех пор, пока следующее рассматриваемое значение d_{i+q+1} не окажется лежащим между l и u. В таком случае в архив вносят значения $(j_k, c_k)=(i, d_i)$ и $(j_{k+1}, c_{k+1}) = (i+q, (l(i+q)+u(i+q))/2)$ (где c_{k+1} совпадает с серединой промежутка между прямыми l и u при значении индекса, равном $i+q$), что и создает эффект сжатия информации, и повторяют описываемую процедуру уже с точки (j_{k+1}, c_{k+1}) в качестве отправной.

Данная процедура может быть представлена следующим кратким псевдокодом.

Входные данные:

d_i – временной ряд, подлежащий сжатию,
i – индекс, принимающий значения от 1 до N,
N – размер сжимаемого массива отсчетов временного ряда,
Δ – заданная пользователем величина дополнительного отклонения, которое возможно внести в сжимаемые данные.

Выходные данные:

ряд кортежей $\langle j_k, c_k \rangle$, где
j_k – позиции в сжимаемом массиве, на которые при разархивации должны быть помещены значения c_k,
k – индекс, принимающий значения от 1 до $M \le N$, где M – размер получаемого архива (отложенного также в количестве отсчетов).

1. Здесь и далее присвоить значения в том порядке, как перечислена последовательность операций: i:=1, k:=1, q:=1 и j_1:=1, c_1:=d_1.
2. Если значение $(i+q–1)$ совпадает с N, то j_k:=N, c_k:=0,5·$(l(i)+u(i))$ и выйти из процедуры.
3. Определить две прямые
$$l(t) = R_l \cdot (t - i) + c_i,$$
$$u(t) = R_u \cdot (t - i) + c_i.$$
где $R_l = \frac{\max_{t=i+1,\ldots,i+q} d_t - \Delta - c_i}{\operatorname{argmax}_{t=i+1,\ldots,i+q} d_t - i}$, $R_u = \frac{\min_{t=i+1,\ldots,i+q} d_t + \Delta - c_i}{\operatorname{argmin}_{t=i+1,\ldots,i+q} d_t - i}$.
4. Если $l(i+q+1) \le d_{i+q+1} \le u(i+q+1)$, то q:=q+1. В противном случае i:=$i+q$, j_k:=i, c_k:=0,5·$(l(i)+u(i))$, q:=1, k:=k+1.
5. Вернуться к пункту 2.

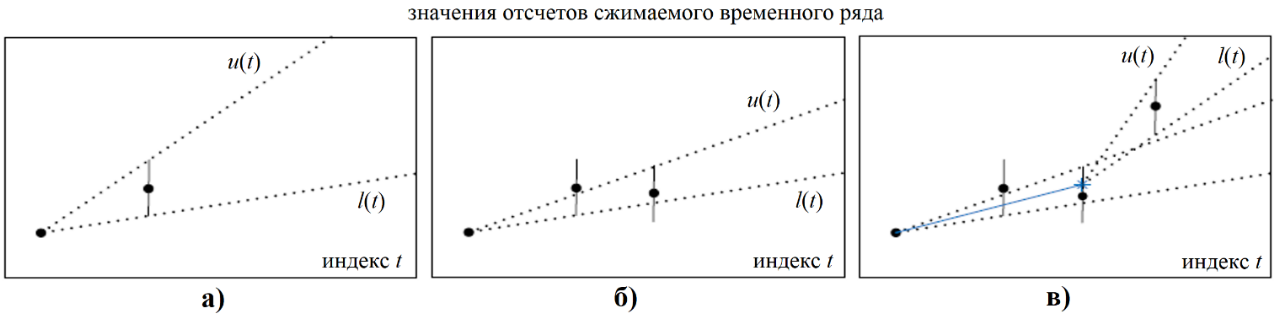

Рис. 1. Иллюстрация работы алгоритма LTC: (а) первый шаг – определение прямых l и u; (б) второй шаг – уточнение прямых l и u для следующей точки сжимаемой последовательности; (в) последний шаг – точка остановки и формирование значений, сохраняемых в архив [8]

Важно отметить, что пункт 3 в представленной процедуре может быть исполнен без необходимости каждый раз при увеличении значения q решать задачу определения максимума и минимума: при увеличении q (то есть при добавлении следующей точки в рассмотрение линейного тренда) достаточно лишь сравнить ее значение со значениями максимума и минимума, определенных на предыдущем шаге. Подобная запись с операторами оптимизации в пункте 3 использована для лаконичности представления алгоритма.

По сути дела, алгоритм LTC обобщает метод K-RLE: если последний предлагает аппроксимировать сжимаемую последовательность значений кусочнопостоянной функцией с минимальным количеством узлов, то первый – кусочно-линейной также с минимальным количеством узлов при одной и той же метрике для величины вносимых искажений.

Обозначим как M количество заполненных значений в последовательности c_j. Пусть, как и выше, $s = 64$ – размер памяти, занимаемый каждым из отсчетов сжимаемого временного ряда (в битах). Тогда размер исходного массива по-прежнему составит $R_0 = N \cdot s$ бит, а размер архива, полученного по алгоритму LTC, не превысит $R_1 = M \cdot s + M \cdot \lceil \log_2 N \rceil$ бит, где второе слагаемое указывает на необходимость хранения индексов значений, помещенных в архив для восстановления исходных данных. Коэффициент сжатия *CR* (compression ratio) оказывается равна

$$CR_{LTE} = \frac{R_0}{R_1} = \frac{N/M}{1+\frac{1}{s}\lceil \log_2 N \rceil}. \qquad (2)$$

Как и в случае K-RLE, все значения, после передачи восстановленные из архива, с гарантией не будут отличаться от исходных на величину, большую Δ.

Процедура восстановления сжатой последовательности из архива, полученного с применением алгоритма LTC, осуществляется по следующей последовательности действий (при тех же обозначениях, что были использованы выше).

1. Здесь и далее присвоить значения в том порядке, как перечислена последовательность операций: i:=2, k:=2, d_1:=c_1.
2. Если k равно M+1, то выйти из процедуры.
3. Если разность (j_{k-1}–j_k) равна 1, то d_i:=c_k, i:=i+1, k:=k+1. В противном случае определить прямую

   $$m(t) = \frac{c_k - c_{k-1}}{j_k - j_{k-1}} \cdot (t - j_{k-1}) + c_{k-1},$$

   и для всех целых значений q от (j_{k-1}+1) до (j_k–1) выполнить присвоения d_q:=$m(q)$, затем присвоить i:=j_k, d_i:=c_k, k:=k+1.
4. Вернуться к пункту 2.

III. Сравнительное исследование

Представленные алгоритмы K-RLE и LTC были сопоставлены на предмет достигаемого ими коэффициента сжатия на модельных временных рядах результатов измерений при разных значениях величины вносимой дополнительной погрешности Δ. Также для построения соответствующих выводов были построены следующие две границы возможного сжатия: миноранта CR_l для сжатия с контролируемыми потерями и мажоранта CR_u для сжатия без потерь. Данные зависимости были определены так. CR_l – это тот коэффициент сжатия, который будет обеспечен, если возможность внесения дополнительной погрешности просто приводит к отбрасыванию становящихся незначимыми разрядов в значениях сжимаемой последовательности (наиболее тривиальный способ сжатия с контролируемыми потерями):

$$CR_l \approx \frac{\max_i |d_i|}{\max_i |d_i| - \Delta}. \qquad (3)$$

Максимальный коэффициент сжатия для методов без потерь определяется согласно теореме Шеннона:

$$CR_u \approx \frac{s \cdot N}{s \cdot 2^H + H \cdot N}, \qquad (4)$$

где H – энтропия сжимаемой последовательности (в битах). Вид формулы определяется необходимостью передачи словаря до собственно сжатия.

Выполнить сравнение изучаемых методов удобно на сигналах, для которых сжатие без потерь не приводит к выигрышу. К таким сигналам относятся в первую очередь шумоподобные сигналы. По этой причине в качестве моделей реальных временных рядов результатов измерений, подлежащих сжатию, были взяты:

1) массив значений типа double случайной величины, равномерно распределенной в пределах от 0 до 1;

2) массив значений типа double случайной величины, распределенной по нормальному закону с математическим ожиданием, равным 0,5, и среднеквадратическим отклонением, равным 0,25, усеченному по границам интервала [0, 1].

Размер каждого массива выбирался равным 10^5 значений, чтобы нивелировать эффекты от конечного размера сжимаемой последовательности.

Величина допустимой вносимой абсолютной погрешности Δ для значений сжимаемых последовательностей задавалась от 0,02 до 1,00 с шагом 0,02. В результате применения методов K-RLE и LTC были получены зависимости коэффициента сжатия $CR_{K\text{-}RLE}(\Delta)$ и $CR_{LTC}(\Delta)$ от Δ, представленные на рис. 2 и 3.

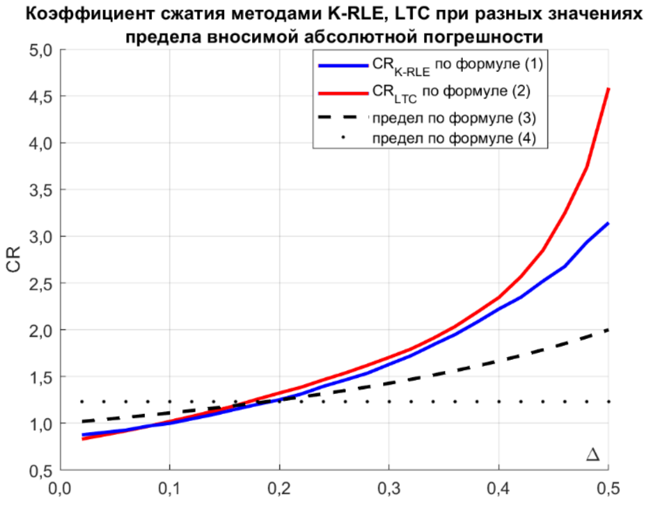

Рис. 2. Зависимость величины коэффициента сжатия CR от величины вносимых искажений Δ для алгоритмов K-RLE и LTC для равномерно распределенного шумоподобного сжимаемого временного ряда

Как видно из рис. 2–3, сжатие с контролируемыми потерями превосходит потенциально достижимые результаты сжатия без потерь. Алгоритмы K-RLE и LTC, начиная с некоторого значения Δ, сильно улучшают результаты сжатия, которые можно получить тривиальным отбрасыванием незначимых цифр по соотношению (3). Видно, что для шумоподобных сигналов алгоритм LTC показывает лучшие результаты, чем K-RLE, чего следовало ожидать ввиду возможности учета последним только трендов в виде кусочно-постоянных функций, а первым – в виде кусочно-линейных функций, что расширяет его аппроксимирующие возможности.

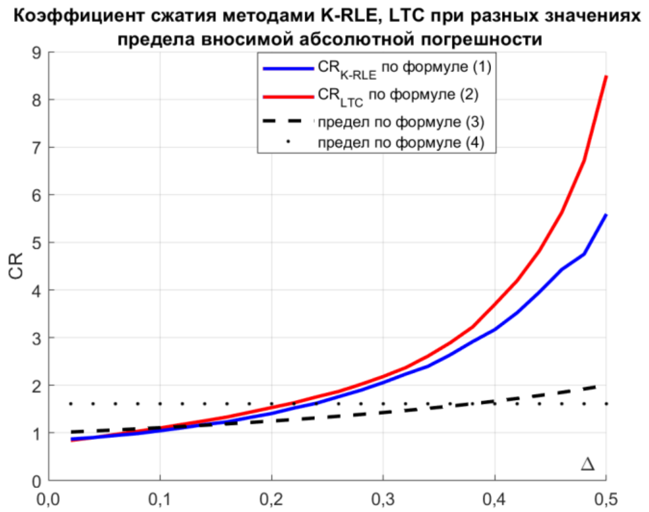

Рис. 3. Зависимость величины коэффициента сжатия CR от величины вносимых искажений Δ для алгоритмов K-RLE и LTC для нормально распределенного шумоподобного сжимаемого временного ряда, усеченного по уровню ±2·СКО

Также может быть сделан вывод, что чем сильнее распределение значений временного ряда, подлежащего сжатию, отличается от равномерного (наиболее энтропийного распределения при наличии ограничений на величину принимаемых значений), тем сильнее достигаемый выигрыш. Данное обстоятельство объясняется тем, что в сжимаемой последовательности в таком случае возникает больше участков, элементы которых отличаются друг от друга не более чем на Δ.

Если в формуле (4) использовать значение H = H(Δ) энтропии, вычисленной в предположении равенства значений, разность между которыми не превосходит величины Δ, то получим, что существует довольно большой запас по улучшению эффективности методов сжатия с потерями. Методы K-RLE и LTC являются лишь первыми в череде возможных подходов, причем далеко не оптимальными. Действительно, для рассмотренных равномерно и нормально распределенных сжимаемых последовательностей большой длины N>>1 формула (4) оказывается хорошо приближаемой при 0,0<Δ<0,5 прямой вида $CR_u(\Delta) \approx 88 \cdot \Delta + 9{,}3$.

IV. Заключение

В настоящей работе впервые на русском языке представлено описание основных существующих на данный момент методов сжатия с контролируемыми потерями – K-RLE и LTC. Наиболее кратким образом изложены их основные идеи и псевдокод соответствующих алгоритмов.

Выполнены численные исследования и теоретические построения, показавшие, что

1) методы K-RLE и LTC для шумоподобных сигналов сильно превосходят предел сжатия, существующий для методов сжатия без потерь;

2) методы K-RLE и LTC сильно превосходят по коэффициенту сжатия тривиальный алгоритм, заключающийся в отбрасывании тех цифр, что оказываются при внесении дополнительной погрешности незначимыми;

3) метод LTC превосходит K-RLE по коэффициенту сжатия на шумоподобных временных сигналах измерительной информации;

4) методы K-RLE и LTC довольно далеко отстоят от теоретического предела сжатия с контролируемыми потерями, что указывает на возможности дальнейшего совершенствования методов в данной области.

Список литературы

[1] I.C. Paschalidis and S. Vassilaras, “On the estimation of buffer overflow probabilities from measurements,” IEEE Transactions on Information Theory, vol. 47(1), pp. 178-191, 2001.

[2] J. Uthayakumar, T. Vengattaraman and P. Dhavachelvan, “A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications,” Journal of King Saud University – Computer and Information Sciences, vol. 33(2), pp. 119-140, 2018.

[3] C. Chen, L. Zhang and R.L.K. Tiong, “A new lossy compression algorithm for wireless sensor networks using Bayesian predictive coding,” Wireless Networks, vol. 26(8), pp. 5981-5995, 2020.

[4] Бурый А.С., Лобан А.В., Ловцов Д.А. Модели сжатия массивов измерительной информации в автоматизированной системе управления // Автоматика и телемеханика. 1998. № 5. С. 3–26.

[5] Раимжанова А.Б., Семенов К.К. Классификация и статистика применения методов сжатия измерительной информации // Международная конференция по мягким вычислениям и измерениям. 2024. Т. 1. С. 427-431.

[6] F. Cappello, S. Di, S. Li, X. Liang, A.M. Gok, D. Tao, C.H. Yoon, X.C. Wu, Y. Alexeev and F.T. Chong, “Use cases of lossy compression for floating-point data in scientific data sets,” The International Journal of High Performance Computing Applications, vol. 33(6), pp. 1201-1220, 2019.

[7] E.P. Capo-Chichi, H. Guyennet and J.M. Friedt, “K-RLE: a new data compression algorithm for wireless sensor network,” In 2009 Third international conference on sensor technologies and applications, pp. 502-507, 2009.

[8] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow and D. Estrin, “Lightweight temporal compression of microclimate datasets,” In 29th annual IEEE international conference on local computer networks, pp. 1-9, 2004.