**DOI: https://doi.org/10.15276/hait.09.2026.18**
**UDC 004.62.:004.45:629.7.05**

# Improving the information compression algorithm in the unmanned aerial vehicle communication system

**Valerii I. Magro**[1]
ORCID: https://orcid.org/0000-0003-4238-6733; magro.v.i@nmu.one. Scopus Author ID: 6602911357
**Valerii I. Korniienko**[1]
ORCID: https://orcid.org/0000-0002-0800-3359; korniienko.v.i@nmu.one. Scopus Author ID: 56446921900
**Dmytro S. Tymofieiev**[1]
ORCID: https://orcid.org/0000-0002-9718-6678; tymofieiev.d.s@nmu.one. Scopus Author ID: 55437340600
**Iryna M. Udovyk**[1]
ORCID: https://orcid.org/0000-0002-5190-841X; udovyk.i.m@nmu.one. Scopus Author ID: 55998874400
[1] Dnipro University of Technology, 19, Dmytro Yavornytskyi Ave. Dnipro, 49005, Ukraine

## ABSTRACT

The problem of processing large amounts of data transmitted over wireless communication networks with low bandwidth is a **pressing one,** and it is one of the main obstacles to the effective deployment of such networks. The most rational way to solve this problem is to conduct research on improving the compression method, or using a combination of data compression methods, which will reduce the volume and time of data transmission. The **purpose** of the research is to increase the compression of data transmitted between an unmanned aerial vehicle and a control point and reduce the processor time for compressing one packet of the source coding process for a wireless communication channel using the MAVLink protocol in conditions of limited resources of the microcontroller of an unmanned aerial vehicle. The **task** set in the work was to develop an improved lossless data compression algorithm when using it within the Mavlink v2 protocol and evaluate the compression efficiency. The **methods** used were to improve the source stream compression when exchanging data using the MAVLink protocol, which is characterized by small packet volumes. Source coding with lossless compression was carried out by forming bit masks and pairwise compressing them according to their properties. The **scientific novelty** consists in the development of a sequential pair source coding algorithm for a wireless communication channel, which implements stream coding of bytes of the packet structure, which provides lossless information compression. The **practical significance** of the results obtained lies in the possibility of effective application in the development of communication systems for control and information exchange between a control point and an unmanned aerial vehicle using the MAVLink protocol. The **results** of the work are the improvement of the source coding algorithm for effective lossless compression of MAVLink packets in communication systems under conditions of limited resources of an unmanned aerial vehicle, and the establishment of the inefficiency of known algorithms for small MAVLink packets. that the total volume of packets is reduced by fifty-four percent. Comparison of the compression efficiency of the proposed improved source coding algorithm and known compression algorithms showed that the total volume of packets is reduced by seventeen percent. In addition, when using the proposed algorithm, the processor time consumption for compressing one packet is significantly reduced (on average, compression of one packet lasts less than a microsecond), which is significantly less than when using known compression algorithms. The **conclusions** of the work are the proposal of an improved compression algorithm for a wireless communication channel, which provides a higher compression ratio compared to the considered algorithms during streaming serial-pair encoding of bytes of MAVLink protocol packets.

**Keywords:** Streaming information; communication channel; lossless compression; control point; MAVLink protocol; compression algorithm

***For citation:*** **Magro V. I., Korniienko V. I., Tymofieiev D. S., Udovyk I. M. "Improving the information compression algorithm in the unmanned aerial vehicle communication system".** ***Herald of Advanced Information Technology*****. 2026. Vol. 9. No. 3: 275–288. DOI: https://doi.org/10.15276/hait.09.2026.18**

## INTRODUCTION

The transition to digital society is driving significant advances in new technologies across various spheres of human activity [1], [2]. Digital transformation affects all segments of human life [3], [4]. A future society supported by technology can provide a higher standard of living. The transition to a digital society generates large volumes of digital data and requires advanced compression methods. Data compression in various mobile devices helps conserve battery power [5].

Unmanned aerial vehicles (UAVs) are becoming increasingly popular. These vehicles are controlled remotely via a ground control station or autonomously via a programmed task. Over the past decade, the concept of UAV traffic control has become a reality. In particular, the MicroAirVehicle (MAVLink) communication protocol was developed to exchange data between the control point and the UAV, aggregating the necessary data and information for unmanned traffic control. Therefore, when controlling unmanned aerial vehicles, it is necessary to compress the data transmitted to the UAV. In addition, reducing the amount of information transmitted through the channel increases the channel's resistance to destructive influences.

The process of data transmission from the control point to the UAV is carried out using the MAVLink protocol. The transmission algorithms operate in real time based on a microcontroller with limited resources.

When transmitting large amounts of information over communication channels with limited bandwidth, data compression is crucial for increasing channel capacity. On the other hand, transmitting a smaller amount of data over a communication channel requires less processing time of the UAV microcontroller and, accordingly, saves battery power. An unsolved problem is the lack of an effective data compression algorithm in the communication system operating under the MAVLink protocol.

## LITERATURE REVIEW AND PROBLEM STATEMENT

In [6], a comparative analysis of the impact of de-parsing by scaling data on lossless data compression methods is carried out. Different compression algorithms are analyzed according to the efficiency criteria, such as the time required for compression, the time required for decompression, and the compression ratio. These factors are evaluated together with the scaled dataset process. The relationship between scaling and data compression ratio is established. However, the method proposed in this work may reduce the compression ratio compared to a single array when the packet volume is small.

In [7], an efficient lossless compression of records is investigated. An adaptive arithmetic coding approach is proposed that uses an entropy encoder to securely and properly encode data. In this method, the compression ratio can be controlled by changing the mathematical coding methods. This approach allows you to achieve an optimized compression ratio using a statistical method that considers the relative frequencies for each symbol, regardless of the permutation in which the symbols are used in the file. The authors emphasize that a faster version of adaptive arithmetic coding can be used for large data sets, but issues related to the compression of small volumes remain unresolved within the MAVLink protocol.

Some studies emphasize that data compression is a crucial factor in increasing the efficiency of computing in conditions of increasing data volumes [8]. Despite the potential to improve system performance, studies have identified significant problems with existing compression methods, mainly due to their high computational requirements against the background of constantly growing data sizes. The authors of [8] note that one way to increase computing efficiency is to use a more advanced processor equipped with programmable systems on a chip (SoC) and specialized compression accelerators. However, issues remain unresolved regarding the compression of small amounts of data, such as those available in UAVs. All this provides grounds for arguing that it is advisable to conduct a study aimed at overcoming the relevant difficulties by improving the compression algorithm for streaming information in the communication system between the unmanned aerial vehicle and the control point.

In parallel, there is a rapid development of sensor networks, in which large amounts of information circulate. In the Internet of Things system, including wireless sensor networks, battery energy saving is a matter of paramount importance [9]. Although electronic devices, as well as software, are constantly developing, battery development technologies have not kept up with this. In addition to the fact that each sensor node usually uses batteries for power, these sensors often cannot be changed or recharged in the usual way. In all sensor node operations, wireless communication is the most energy-intensive operation. Therefore, to extend the sensor node's battery life, it is necessary to limit information transmission over the wireless network. This can be done using data compression. For this, it is important to choose an effective, simple data compression algorithm, since the energy, memory, and computing resources of the sensor node are very limited. A similar problem arises when using UAVs. In [10] it is noted that most often the nodes of wireless sensor networks are in uncontrolled, potentially dangerous environments. Therefore, the battery life, memory capacity, and computing power in these nodes are significantly limited.

This study emphasizes that the collection and transmission of compressed data can significantly increase the energy efficiency of the network and ensure secure data transmission. The approaches to information compression and power consumption reduction described in the mentioned work can be applied to UAVs.

There is a need to develop efficient methods for compressing data circulating in sensor networks by developing a new literal structure [11]. The application of the proposed idea to the communication lines between UAVs and the control point clearly requires additional research. In [12], a high-speed lossless data compression algorithm called SnappyR is proposed for communication between low-power devices. The authors propose a combination of data compression methods that reduces data size without compromising data quality. This proposal improves on the well-known Snappy algorithm by introducing new literal and match token structures that achieve a better compression ratio. An improved compression algorithm is proposed for computing and data storage systems that require high-speed, lossless data compression. The ideas presented in the mentioned work can be applied to further research on UAVs.

The development of intelligent networks (INs) in the energy sector requires integrating digital technologies into facility management and network status monitoring [13]. In [14] it is emphasized that due to the rapid development of intelligent network technologies, huge amounts of system data are being created for the INs. These large amounts of system data are intended for monitoring network status and warning of malfunctions. The large volume of data creates significant challenges for data transmission and storage. In [14], a method of lossless data compression based on the improved Lempel-Ziv-Welch algorithm is proposed. Similar processes can be observed when transmitting data from UAVs, but the issue of using the LZMA method within the MAVLink protocol needs to be studied.

The increasing amount of data circulating in the INs, particularly in the smart meter environment, also requires improvements in data compression [15]. Thus, improvements in lossless data compression technology are becoming an important factor for system data processing. In this paper, several compression methods are investigated for application in the specific case of smart meters. Based on the research presented in this paper, further work is needed on data compression methods for streaming MAVLink packet compression. An integral process of INs is monitoring the dynamics of energy systems [16]. Due to the large volume of this data, communication and data storage systems face serious difficulties. In this paper, several compression methods are investigated to select a lossless data compression method that provides more efficient communication within the INs and improves the storage of this data. This study suggests the need for further research on data compression methods for MAVLink packets.

The widespread use of remote sensing of Earth generates large amounts of telemetry data and requires its compression [17]. It is known that the high-speed data transmission during remote sensing of the Earth in real time depends on the conditions of signal propagation [18], therefore, information compression allows you to reduce the communication session and thereby maintain a stable connection. In [19], it is argued that when transmitting telemetry information, it is advisable to use data compression methods without loss of original information. In this work, the Lempel-Ziv-Markov and GZIP methods are used to compress synthetic telemetry data. These methods improve transmission efficiency, data transfer rates, and compression ratios, but they are not appropriate for transmitting information from UAVs. Onboard satellite hyperspectral images are characterized by a large amount of data collected with decreasing distance to the ground sample, and limited onboard resources to meet relatively stable channel bandwidth requirements. In [20], an efficient hardware architecture for a simple lossless algorithm for compressing hyperspectral data on board a satellite, operating in a sequential band order, is presented. The implementation of the specified approach to compressing hyperspectral data provides a bandwidth of ~3.8 Gbit/s at a clock frequency of 238 MHz with less use of satellite system resources. However, using efficient hardware architecture for a simple lossless algorithm on a UAV has limitations. All this provides grounds for arguing that it is advisable to conduct a study of data compression methods for MAVLink packets.

Encoding the information source for data compression can be performed with or without loss. In [8] it is emphasized that a significant acceleration of compression processes is provided using high-speed equipment. However, installing high-speed equipment on a UAV significantly increases its cost and is not advisable for mass production. Therefore, another approach to increase compression efficiency is to improve known compression algorithms [11].

As is known, telecommunication traffic in packet data systems is considered a heavy-tailed process. There are many complex approaches to traffic forecasting; however, in the simple case of stationary traffic, they may not be needed, and a simple approach such as the Kolmogorov-Wiener filter can be used [21]. The results of this work can be used to forecast stationary telecommunication traffic in packet data systems, including when using the MAVLink protocol. At the same time, the use of both discrete and continuous Kolmogorov-Wiener filters in UAVs can be used for fairly accurate short-term forecasting of a stationary random process [22].

In [23], it is shown that increasing the compression ratio can be achieved using a neural network, allowing a flexible balance between model complexity and compression ratio. At the same time, when applying the specified approach to UAVs, the issue related to connecting the UAV to the neural network remains unresolved. On the other hand, when building wireless sensor networks, it is very important to choose an effective, simple data compression algorithm, since the energy, computing resources, and memory capacity of sensor nodes are very limited [9]. All the above factors are relevant for communication between the UAV and the control point. Therefore, the development of the specified approach for UAVs requires further research. With the development of modern technologies, the amount of data generated, for example, by LCLP (Linear Code Linear Programming) devices is very significant. At the same time, these devices themselves are low-power and cannot perform complex calculations [12]. The same applies to UAVs' information-processing capabilities. On the other hand, data transfer speeds are limited across various segments of the wireless network, including the UAV segment and the control point.

Analysis of literature sources showed that it is necessary to compare the effectiveness of known algorithms also in terms of the amount of processor time required for the compression process. Because in most cases, the UAV does not have a high-performance processor. In this case, the degree of data compression depends on the size of the accumulation buffer, which can affect the amount of compressed information and delay information transmission [24]. When using known algorithms to compress one MAVLink packet, the amount of information transmitted over the wireless channel may increase, since most known compression algorithms are inefficient for working with small amounts of data.

The problem of processing large amounts of data transmitted over low-bandwidth wireless networks is a major obstacle to effective deployment. To solve this problem, it is proposed to conduct research on improving the compression method or on combining data compression methods, thereby reducing data transmission time over the wireless channel between the ground control point and the unmanned aerial vehicle.

## PURPOSE AND OBJECTIVES OF THE STUDY

The purpose of the research is to increase the compression of data transmitted between an unmanned aerial vehicle and a control point and reduce the processor time for compressing one packet of the source encoding process for a wireless communication channel between a ground control point and an unmanned aerial vehicle using the MAVLink protocol under conditions of limited UAV microcontroller resources.

To achieve the purpose, the following tasks were set:

– to analyze the efficiency of data compression by existing compression algorithms Bzip2, Deflate, LZMA when used within the Mavlink v2 protocol in conditions of variable buffer size accumulation;

– to develop an improved lossless data compression algorithm when used within the Mavlink v2 protocol;

– to evaluate the efficiency of compression algorithms: determine the values of compression ratios and analyze the loss of processor time when using the Bzip2, Deflate, LZMA algorithms and the proposed ISEA algorithm when using them within the Mavlink v2 protocol in conditions of limited UAV microcontroller resources.

## MATERIALS AND METHODS

The object of the study is an algorithm for compressing streaming data between an unmanned aerial vehicle (UAV) and a control point, operating within the MAVLink protocol. It is known that the MAVLink communication protocol is an open-source point-to-point network protocol used to transmit telemetry data and control and monitor many small unmanned aerial vehicles. This protocol provides powerful functions not only for monitoring and controlling flights of unmanned systems, but also for their integration into the Internet of Things [25], [26].

The main hypothesis of the study was that the communication channel should be reduced to minimise the amount of information transmitted during the exchange between the control point and the UAV.

The work assumed that the peculiarity of the MAVLink protocol lies in, on the one hand, the degree of information compression and, on the other hand, the speed of information processing. This peculiarity of the MAVLink protocol is due to a small sequence of bytes (up to 280 bytes).

Therefore, the simplifications adopted in the work consist of reducing the length of the input information transmitted through the communication channel.

Resolution of the Cabinet of Ministers of Ukraine No. 835 of October 21, 2015, defines the ZIP, Bzip2, 7z, Gzip algorithms as the main container formats for document exchange. At the same time, the ISO/IEC 21320-1:2019 standard defines the ZIP algorithm as the basic container format for document exchange. The ISO/IEC 21320 document does not use Bzip2 as its primary algorithm; it is based on the ZIP specification, which uses Deflate as its main compression method. A feature of the Deflate algorithm is that data is compressed into separate blocks (up to 64 KB), allowing them to be quickly unpacked without processing the entire file. Regarding the Bzip2 algorithm, it provides higher compression than Deflate but runs more slowly. The LZMA compression algorithm is a lossless compression algorithm that provides a high compression ratio and is used in the 7z archiver. It should be noted that the LZMA algorithm offers a higher compression ratio than the Bzip2 algorithm. In LZMA, compression is slower than in Bzip2, but decompression is very fast.

The paper considers the use of well-known lossless compression algorithms such as Bzip2, Deflate, and LZMA. It was found that they do not work effectively with a small amount of data. And, as a rule, with small amounts of input information, these algorithms expand the amount of information after compression. Because these algorithms work in real time in a microcontroller, where resources are limited, they were compared according to the criterion of optimizing processor time. Such preliminary research enabled us to develop our own data compression algorithm, which effectively reduces the amount of information and consumes a small percentage of the microcontroller's processor time.

The article proposes an encoding algorithm for the MAVLink communication protocol based on the Improved Source Encoding Algorithm (ISEA). This is a lossless compression algorithm.

The proposed algorithm can be generalized by the following formula:

$$V=F(A),$$

where V is the total volume of the packet after compression; A is the total volume of the packet before compression, and F is the transformation function.

The overall development process for the proposed compression algorithm can be divided into five steps (algorithm modifications). After implementing each step, the dependence of the volume of information after compression on the accumulation buffer volume (i.e., the number of MAVLink messages) was checked. If the compression ratio was too low, the next modification (version) of the algorithm was created.

For clarity, we present a description of the first and fifth implementations of the proposed MAVLink packet data compression algorithm.

Let us consider the mathematical and formal formulation of the problem for the first implementation. Let the input MAVLink packet be represented as a sequence of bytes X = (x_1, x_2, \dots, x_N), where x_i ∈ [0, 255], and N is the fixed or dynamic packet length (N ≤ 280).

The algorithm encodes the input array X into an output array of compressed data Y using two classification bit masks:

– sign/boundary mask (M_{sign}): determines whether a number is small (≤ 0x0F);

– subrange mask (M_{range}): determines which of the 15 linear intervals in steps of 16 values the number x_i belongs to if it exceeds 0x0F.

Formally, for each x_i, the range step k_i is defined as: k_i = \lfloor rac{x_i}{16} floor.

The value of the remainder (delta) for packing is calculated as: \Delta_i = x_i \pmod{16}.

Packing of two consecutive deltas \Delta_i and \Delta_{i+1} into a single byte of the output stream is done using a bitwise shift of 4 bits to the left for the first element: y_{packed} = (\Delta_i \ll 4) \lor \Delta_{i+1}.

Let us consider a step-by-step decomposition of the first implementation of the algorithm. The algorithm operates within the framework of clearly defined sequential stages.

1. Initialization: writing the original packet length N into the first two bytes of the output buffer Y.

2. Scanning and forming the first mask: traversing the array X. If x_i ≤ 15, the corresponding bit in the bit mask is set to 1, otherwise to 0. The fully formed mask bytes are written to the output stream.

3. Forming the range mask: for elements where the mask bit is 0, the interval index k_i is calculated. The obtained index is written to the corresponding half-byte (nibble) of the subrange mask.

4. Pairwise compaction: elements are transformed into their four-bit deltas \Delta_i. Adjacent deltas are combined in pairs into bytes using nibble shifting logic and appended to the end of the resulting array.

The program operates on basic low-level data structures without using dynamic memory allocation inside the compression function:

– Static byte arrays (uint8_t): used as linear buffers to minimize processor overhead.

– Pointers (Puint8_t): for fast direct addressing and memory shifting during bitwise packing.

The complexity of the first implementation of the algorithm was evaluated with respect to the length of the input packet N, as shown in Table 1. The linear time complexity O(N) and constant space complexity O(1) make the first implementation of the proposed algorithm extremely fast and suitable for embedded onboard UAV systems with strict real-time constraints.

Let us consider the mathematical and formal formulation of the problem for the fifth implementation of the proposed algorithm. This algorithm consistently combines Run-Length Encoding (RLE) with pairwise packing of low nibbles (small numbers).

Let the input MAVLink packet be represented as a sequence of bytes X = (x_1, x_2, \dots, x_N). The compression is performed in two discrete stages.

Stage I: sequential compression (RLE). The sequence X is mapped to an intermediate sequence V = (v_1, v_2, \dots, v_M). If the element x_i is repeated R times in a row (R ≥ 2), it is encoded by the pair: v_j = x_i, \quad v_{j+1} = R. In this case, the fact of repetition is recorded in the corresponding bit mask of the array.

Stage II: pairwise packing of low nibbles. The resulting array V is analyzed for the presence of adjacent small numbers. The packing possibility criterion for two adjacent bytes is determined by the predicate: P(v_i, v_{i+1}) = (v_i \le 15) \land (v_{i+1} \le 15). If P(v_i, v_{i+1}) = ext{true}, these two bytes are combined into the final byte of the output stream Y according to the formula: y_k = (v_i \ll 4) \lor (v_{i+1} \land 0x0F). To identify the application of this stage, a marker flag (0x80) is added to the beginning of the output packet.

Let us consider a step-by-step decomposition of the fifth implementation of the proposed algorithm.

1. The first stage (RLE compression): iterative scanning of the input data with a repetition counter. If the current byte equals the next, the counter is incremented. The result, together with the mask, is written to a temporary buffer.
2. Analysis of the feasibility of the second stage: the program scans the intermediate buffer and counts the number of pairs that satisfy the condition of a small number (≤ 0x0F), estimating the final data volume.
3. Decision-making (fuse): if the calculated future size is larger than the current one, the second stage is cancelled, and the data is returned in the state after Stage I.
4. Pairwise bit packing: if compression is beneficial, the bytes are shifted bit-wise to the left and right nibbles, respectively, reducing the physical size of the array by half for each pair.

To ensure two-stage processing, the algorithm uses a more complex memory management model.

– Dynamic memory (heap): using the `malloc()` function to create an intermediate buffer `buffer2` inside the encoding function. At the end of processing, `free()` is necessarily called.

– Control variables and flags: special logical flags (`true`/`false`) and bit markers to mark the decompression phases.

The complexity of the fifth implementation of the algorithm was estimated with respect to the length of the input packet N, as shown in Table 2. The fifth implementation of the proposed algorithm provides higher mathematical compression efficiency (especially in the presence of a series of zeros in MAVLink), but has a higher spatial complexity O(N) due to dynamic memory allocation.

## RESULTS OF THE STUDY ON INCREASING THE EFFICIENCY OF THE SOURCE CODING PROCESS FOR A WIRELESS COMMUNICATION CHANNEL

### Study of the efficiency of existing compression algorithms when used within the MAVLink protocol

The objective of the research in this section was to analyze the efficiency of data compression by existing compression algorithms, Bzip2, Deflate, and LZMA, when used within the MAVLink v2 protocol under conditions of variable accumulation buffer size.

To solve this problem, a test file with a set of Mink v2 messages of 151 KB was created. The methodology of this study was that the Bzip2, Deflate, and LZMA algorithms encoded each message separately, “simulating real-time operation.” After that, the compression ratio was calculated. In this case, an “accumulation buffer” with different volumes was considered; that is, the dependence of the compression ratio on the size of the input information was studied. When these algorithms did not effectively compress a small amount of input data, the size of the “accumulation buffer” was changed, respectively, by 5, 10, 20, or 40 MAVLink messages.

The Bzip2 algorithm was first investigated. Table 3 shows the dependence of the change in the total size of the information after compression (compression ratio) of data by the Bzip2 algorithm on the size of the “accumulation buffer”.

The term “compression ratio” shows how many times the original file has decreased:

$$S=V_0/V_{rat},$$

where $V_0$ is the initial packet size; Vrat is the packet size after compression.

For clarity, in this table, the size of the information after compression is compared with the size of the input information without the compression procedure (presented as percentages).

*Table 1.* **Estimated computational complexity (O-notation) of the first implementation of the proposed algorithm**

| Difficulty type | Difficulty class | Mathematical justification |
|---|---|---|
| Time | $O(N)$ | The algorithm performs 3 consecutive linear passes over the data array of length N (counting mask 1, counting mask 2, packing deltas). There are no nested loops. Total time T(N) = c_1 \cdot N + c_2 \cdot N + c_3 \cdot N = O(N) |
| Space | $O(N)$ | The algorithm uses a fixed number of local variable counters and masks. The amount of additional memory does not depend on the change of N, since the work is done directly in dedicated external buffers |

***Source:*** **compiled by the authors**

*Table 2.* **Estimated computational complexity (O-notation) of the fifth implementation of the proposed algorithm**

| Difficulty type | Difficulty class | Mathematical justification |
|---|---|---|
| Time | $O(N)$ | The algorithm contains several sequential passes (RLE, size analysis, and final packing). The nesting of the loops is limited and does not depend on the square of the data, since the inner RLE loop simply shifts the main index forward. Total: T(N) = O(N) |
| Space | $O(N)$ | Unlike the first version, this algorithm requires allocating an additional dynamic buffer whose size is directly proportional to the length of the input packet N. That is, S(N) = O(N) |

***Source:*** **compiled by the authors**

*Table 3.* **Dependence of the compression efficiency of the Bzip2 algorithm on the volume of the “accumulation buffer"**

| Indicator | Number of MAVLink messages in buffer | | | | |
|---|---|---|---|---|---|
| | 1 | 5 | 10 | 20 | 40 |
| The volume of information after compression, KB | 142 | 117 | 103 | 91 | 83 |
| Data compression ratio | 1.063 | 1.290 | 1.466 | 1.659 | 1.819 |

***Source:*** **compiled by the authors**

*Table 4.* **Dependence of the compression efficiency of the Deflate algorithm on the volume of the “accumulation buffer”**

| Indicator | Number of MAVLink messages in buffer | | | | |
|---|---|---|---|---|---|
| | 1 | 5 | 10 | 20 | 40 |
| The volume of information after compression, KB | 110 | 86 | 81 | 77 | 71 |
| Data compression ratio | 1.372 | 1.755 | 1.864 | 1.961 | 2.126 |

***Source:*** **compiled by the authors**

*Table 5.* **Dependence of the compression efficiency of the LZMA algorithm on the volume of the “accumulation buffer”**

| Indicator | Number of MAVLink messages in buffer | | | | |
|---|---|---|---|---|---|
| | 1 | 5 | 10 | 20 | 40 |
| The volume of information after compression, KB | 105 | 84 | 77 | 71 | 65 |
| Data compression ratio | 1.438 | 1.797 | 1.961 | 2.126 | 2.323 |

***Source:*** **compiled by the authors**

The Deflate algorithm was tested similarly to the Bzip2 algorithm. The work with a separate MAVlink package and with the size of the “accumulation buffer” of 5, 10, 20, 40 packages was evaluated (Table 4). The results of testing the LZMA algorithm are presented in Table 5.

Testing of the LZMA algorithm was carried out similarly to testing of the Bzip2 and Deflate algorithms.

The conducted research showed that the data compression ratio of the Bzip2 algorithm is in the range from 0.612 to 1.819, while according to the data of work [25] the compression ratio depending on the data type for this algorithm is in the range from 1.320 to 19.208; the Deflate algorithm has a compression ratio in the range from 1.372 to 2.126, while according to the data of work [27] the compression ratio depending on the data type for this algorithm is in the range from 4.958 to 14.911; the LZMA algorithm has a compression ratio in the range from 1.438 to 2.323, while according to the data of work [27] the compression ratio depending on the data type for this algorithm is in the range from 1.639 to 12.198.

Studies conducted on the effectiveness of the Bzip2, Deflate, and LZMA compression algorithms when used within the Mavlink v2 protocol showed that the obtained compression ratios are not sufficiently high. This necessitates the development of an improved compression algorithm.

## Development of an improved information compression algorithm

During the study, five modifications of the improved information compression algorithm (ISEA – Improved Source Encoding Algorithm) were developed, each applied to the MAVLink v2 test set of messages, totaling 151 KB. In Table 6, the obtained compression ratios for each modification of the improved source encoding algorithm (ISEA) are compared.

The coefficients of the sequence X1 are obtained:

0x00 – 0x1F;
0x01 – 0x2F;
0x02 – 0x3F;
0x03 – 0x4F;
0x04 – 0x5F;
0x05 – 0x6F;
0x06 – 0x7F;
0x07 – 0x8F;
0x08 – 0x9F;
0x09 – 0xAF;
0x0A – 0xBF;
0x0B – 0xCF;
0x0C – 0xDF;
0x0D – 0xEF;
0x0E – 0xFF.

The highest data compression ratio was achieved with the fifth modification of the ISEA algorithm within the Mavlink v2 protocol. This modification of the ISEA algorithm was used in further studies.

## Research on CPU time consumption when using different compression methods in conditions of limited microcontroller resources

The task of the study in this section was to evaluate the efficiency of compression algorithms: to determine the values of the compression ratio and to analyze the loss of processor time when using the Bzip2, Deflate, LZMA algorithms and the proposed ISEA algorithm when using them within the Mavlink v2 protocol in conditions of limited UAV microcontroller resources.

During the study, the effectiveness of both the developed compression algorithm and the Bzip2, Deflate, and LZMA algorithms was tested not only by the degree of packet compression, but also by the criterion of algorithm speed (processor time spent on the compression procedure).

The evaluation was carried out on an x86 processor with a core clock frequency of 3.9 GHz by two indicators:

– the average value of the processor time spent (in microseconds) for packet compression;

– the maximum value of the processor time spent (in microseconds) for packet compression.

The proposed sequential-pair compression algorithm ISEA (fifth modification) was compared with the known algorithms Bzip2, Deflate, LZMA [27], according to efficiency criteria, such as compression ratio and algorithm speed. Table 7 shows the results of comparing the compression ratios per MAVLink packet for the proposed improved algorithm of the fifth modification (ISEA) and for known compression algorithms.

Table 8 shows the average and maximum times required to compress a single packet for different compression algorithms.

Comparison of the proposed fifth modification of the developed ISEA compression algorithm with the Bzip2, Deflate, and LZMA algorithms according to two criteria: compression ratio and algorithm speed (value of processor time spent) showed the advantage of the proposed ISEA compression algorithm. Analysis of the second efficiency criterion, namely the algorithm's speed, shows that a significant reduction in the average time spent by the processor when compressing one packet with the

ISEA algorithm helps preserve the UAV's battery charge and, accordingly, increases the UAV's flight time.

## DISCUSSION OF THE RESULTS

According to Table 3, the Bzip2 algorithm is ineffective with small amounts of data. If this algorithm is used to compress a single MAVLink packet, the final size of the information entering the communication channel will decrease slightly, and the compression ratio will be 1.06. If you use an “accumulation buffer” of 40 packets, the compression ratio will be 1.819. However, with such a size of the “accumulation buffer” of 40 packets, the compression and accumulation process in the buffer will lead to delays in information transmission. Therefore, Bzip2 is not effective for use in the system under study.

From Table 4, the Deflate algorithm is more efficient than the Bzip2 algorithm. But it still requires the creation of an accumulation buffer. When working with a separate MAVLink package, the compression ratio will be 1.372.

From Table 5, the LZMA algorithm is more efficient than the Deflate algorithm. But it still requires the creation of an accumulation buffer for more efficient compression.

The first modification of the proposed algorithm was to reduce the value of one byte to four bits and keep two bytes in one.

When forming a new compressed packet, the first two bytes record the initial size of the incoming packet. Next, a bit mask is formed for all bytes of the packet, with each byte weighted by X/8. Each bit stores information about whether the byte is <= 0x0F. If not, then four bits are written to the second bit mask (in the next field), which contains the coefficient from which four bits of the current block must be subtracted to obtain the full eight-bit value.

Thus, the ISEA compression algorithm in the first modification consists of the following:

1) information (sequence of bytes X) is read from the source;

2) the size of the source information X is determined;

3) the initial size of the compressed packet is formed from 2 bytes of the sequence X;

4) for each byte of the sequence X, if it is greater than 0x0F, a byte with a coefficient from the sequence X1 is formed; otherwise, we go to the next byte;

5) after considering all bytes from the sequence X, each pair of bytes is written 4 bits into one common byte.

Compression is achieved by storing a 2-byte value in a single byte and using small values in the packet (<= 0x0F). In the worst case, the packet size will be 2 + (X / 8) + (X / 2) + (X / 2) [bytes], where X is the size of the input packet.

*Table 6.* **The value of the degree of information compression for five modifications of the ISEA algorithm**

| Indicator | ISEA algorithm modification number | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| The volume of information after compression, KB | 97 | 91 | 90 | 86 | 83 |
| Data compression ratio | 1.556 | 1.659 | 1.677 | 1.755 | 1.819 |

***Source:*** **compiled by the authors**

*Table 7.* **Results of comparing the efficiency of the improved source coding algorithm with other algorithms when compressing a single MAVLink packet**

| Indicator | Compression algorithm name | | | |
|---|---|---|---|---|
| | Bzip2 | Deflate | LZMA | ISEA |
| The volume of information after compression, KB | 142 | 110 | 105 | 83 |
| Data compression ratio | 1.063 | 1.372 | 1.438 | 1.819 |

***Source:*** **compiled by the authors**

*Table 8.* **Dependence of the processor time spent on the packet compression on the type of compression algorithm**

| Indicator | Compression algorithm name | | | |
|---|---|---|---|---|
| | Bzip2 | Deflate | LZMA | ISEA |
| Average CPU time spent, µs | 14 | 11 | 150 | 0.6 |
| Maximum CPU time spent, µs» | 56 | 64 | 321 | 4 |

***Source:*** **compiled by the authors**

The algorithm in the first modification without an additional accumulation buffer reduces the total amount of information to 97 KB (Table 6). But it does not respond very much to the size of the input packet. However, this is the only algorithm that compresses individual MAVLink packets. Therefore, there was a need to improve (simplify) the first compression modification.

Instead of preserving the size of the input packet to calculate the bit masks in the compressed packet, the second modification of the algorithm uses the following packet structure: 1 mask byte that indicates which of the next 8 bytes were combined. If it is not possible to concatenate the bytes, then the value is not written because it will not result in compression (since we still must remember 4 bits).

Then the ISEA compression algorithm in the second modification is as follows:

1) the compressed information according to the first modification of the algorithm (sequence of bytes X) is read, and the size of X is determined;

2) for every 8 bytes of the sequence X, a mask byte is formed, the bits of which take the value 1 if the current byte is greater than 0x0F and - 0 otherwise;

3) after considering all bytes from the sequence X for each set of 8 bytes, pairs of bytes in which the mask bits are 0 are combined by 4 bits into one common byte (as in the first modification).

The size of the original packet, in the worst case, will be (X/8) + X. That is, 2 bytes of the sequence have been removed (according to the rule, 1 byte describes the next 8 bytes), and the packet is decoded without loss.

The results of evaluating the effectiveness of the second modification of the compression algorithm showed that it reduces the total amount of information to 91 KB.

In the third modification to the ISEA compression algorithm, it is proposed to increase compression by searching for a larger number of byte pairs with values <= 0x0F.

Let the algorithm calculate all the pairs of bytes before compression. These pairs will be divided into groups of coefficients, from which it is necessary to subtract the values of the bytes. The total coefficient of the system (from 0x00 to 0xFF) is determined by the largest number of pairs of bytes from this group. That is, if in the sequence of pairs of bytes, the values of which come from 0x00 to 0x0F are less than the pairs of bytes with values from 0x5F to 0x6F, then the total coefficient will be 0x6F. All bytes will be subtracted from this coefficient to get a larger number of pairs that can be compressed. And the first byte of the output sequence stores the index of the coefficient (from 0x00 to 0x0F). In the third modification of the algorithm, the output length of the compressed packet is also calculated, and if it is neither less than nor greater than the input sequence, then the output MAVLink packet is transmitted, which starts with 0xFD.

It was found that the total amount of information when using the third modification decreased to 90 KB (Table 6).

To increase the compression level, the fourth modification uses a bit mask to indicate that this byte has a linear repetition (the next byte specifies the number of repetitions). That is, a certain idea of the PPM (Prediction by Partial Matching) algorithm is implemented: an adaptive statistical algorithm for lossless data compression based on contextual modelling and prediction, in the PPMd version.

In this case, there is no point in processing the first byte: it will always remain as 0xFD, i.e. the MAVLink packet header. In addition, the rule is checked: if, after compression, the packet has not decreased or, on the contrary, increased its size, then the packet is transmitted without compression. Since the header is not processed, the first byte of the bit mask will not exceed the value 0x7F. Thus, if the first byte is 0xFD when decoding, then this is the original MAVLink packet.

It was found that for the fourth modification of the algorithm (which includes the previous three modifications), the total amount of information decreased to 86 KB.

The fifth modification combines all previous modifications of the algorithm. The first modification of the algorithm eliminates consecutive repetitions, and the second compresses newly formed pairs. There is also a requirement to check the packet size. At the output, we have the first byte either 0xFD (indicating the original MAVLink packet), a mask with a value from 0x80 to 0xBF (indicating pairwise compression), or a mask with a value up to 0x80 (indicating a compressed sequence).

It was found that for the fifth modification of the algorithm (which includes the previous three modifications), the total amount of information decreased to 83 KB (compression ratio 1.819) (Table 6).

Unlike [25], which reports an average compression ratio of 3.825 for the Deflate algorithm, in the conducted study, when compressing one MAVLink packet with this algorithm, the compression ratio was 1.372 (Table 7).

For the LZMA algorithm, the study obtained a compression ratio of 1.438, whereas the average compression ratio reported in [25] is 5.88.

According to the data of work [25], the compression ratio for the Bzip2 algorithm is 5.975, and in the conducted study, when compressing one MAVLink packet with the Bzip2 algorithm, the compression ratio was 0.616 (Table 7). That is, applying the efficient Bzip2 compression algorithm to small amounts of data is not effective.

Instead, the proposed compression algorithm for compressing a single MAVLink packet yields a compression ratio of 1.8193. This becomes possible due to the use of the fifth modification of the proposed algorithm.

The study found that the improved source-encoding algorithm requires significantly less processor time for compression than the known algorithms Bzip2, Deflate, and LZMA (Table 8).

The proposed ISEA algorithm effectively compresses small amounts of data. In contrast, the LZMA algorithm compresses data effectively but requires the creation of additional accumulation buffers and consumes a lot of computing resources. Among the considered compression algorithms, Bzip2 showed the best compression results (in terms of CPU time). It was determined that the known compression algorithms increase the volume by 5-10% after compression for relatively small MAVLink packets in real-time mode. They begin to work effectively if an accumulation buffer is created for MAVLink messages. However, creating an accumulation buffer will lead to signal delays in the channel and a decrease in communication efficiency.

The proposed ISEA algorithm improves transmission efficiency and enhances the channel’s immunity to destructive influences. When using the proposed ISEA algorithm, the total amount of transmitted information is reduced by 17 %.

The scope of the proposed compression algorithm is limited to its use in information exchange using the MAVLink protocol. And its efficiency relative to known algorithms is due to the small amount of data it compresses. The computational complexity of the algorithm's basic operations is O(N), making it suitable for real-time systems.

The disadvantage of the proposed compression algorithm is its difficulty in scaling to other protocols. This may require significant research into the mathematical improvement of this algorithm for these specific cases.

## CONCLUSIONS

1. It was determined that the considered compression algorithms, Bzip2, Deflate, and LZMA, do not provide the proper degree of compression of MAVLink packets in conditions of variable accumulation buffer volume.

2. An ISEA compression algorithm is proposed for a wireless communication channel between a control point and a UAV, which provides a higher compression ratio compared to the considered algorithms during streaming serial-pairwise encoding of bytes of MAVLink protocol packets.

3. The efficiency criteria of lossless compression methods were evaluated: the compression ratio values were determined, and the speed of algorithms (processor time consumption) was analyzed when using the Bzip2, Deflate, LZMA algorithms and the proposed ISEA algorithm within the Mavlink v2 protocol in conditions of limited UAV microcontroller resources. It was found that the proposed ISEA algorithm provides an average compression time of less than 1 microsecond per packet, while the LZMA algorithm consumes about 150 microseconds.

Further research should focus on developing encryption and coding algorithms for the communication channel between the control point and the UAV to improve noise immunity.

## REFERENCES

1. Holenko, M. Yu. “Adaptive super-resolution integration to enhance object detection on low-quality unmanned aerial vehicle imagery”. *Herald of Advanced Information Technology*. 2025; 8 (2): 164–178, https://www.scopus.com/pages/publications/105024953273. DOI: https://doi.org/10.15276/hait.08.2025.10.

2. Shunin, O. Y. & Maslov, O. V. “Information model of remote sensing of soil”. *Herald of Advanced Information Technology*. 2025; 8 (3): 352–365, https://www.scopus.com/pages/publications/105024773539. DOI: https://doi.org/10.15276/hait.08.2025.23.

3. Kozlov, M. S., Malakhov, E. V. & Shvets, Y. O. “Comparative analysis of chaotic and deterministic methods to territory coverage by drone swarms”. *Herald of Advanced Information Technology*. 2025; 8 (4): 447–463, https://www.scopus.com/pages/publications/105026617701.
DOI: https://doi.org/10.15276/hait.08.2025.28.

4. Kryvenko, S., Lukin, V., Bataeva, E., Krylova, O. & Kryvenko, L. “Comparison of visually lossless compression of dental images by different codes based on harps metric”. *Advanced Information Systems*.

2025; 9 (3): 83–90, https://www.scopus.com/pages/publications/105012851297. DOI: https://doi.org/10.20998/2522-9052.2025.3.10.

5. Vadrevu, S. & Manikandan, M. S. “A new quality-aware quality-control data Compression framework for power reduction in IoT and smartphone PPG monitoring devices”. *IEEE Sensors Letters.* 2019; 3 (7): 6001404, https://www.scopus.com/pages/publications/85082634043. DOI: https://doi.org/10.1109/LSENS.2019.2920849.

6. Erkuş, E. C. & Bursali, A. “A comparative analysis of the lossless data compression methods for Unsparsed Tabular Data”. *ICECET Proceedings Volumes*. 2024. p. 1–6, https://www.scopus.com/pages/publications/85207468979. DOI: https://doi.org/10.1109/ICECET61485.2024.10698440.

7. Sudhamsu, G., Singh, S. & Gautam, C. K. “Efficient lossless data compression using adaptive arithmetic coding”. *ICCCNT Volumes*. 2024; 1–6, https://www.scopus.com/pages/publications/85212957531. DOI: https://doi.org/10.1109/ICCCNT61001.2024.10725862.

8. Li, Y., Kashyap, A., Chen, W., Guo, Y. & Lu, X. “Accelerating lossy and lossless compression on emerging BlueField DPU Architectures”. *IPDPS Proceedings Volumes*. 2024. p. 373–385. https://www.scopus.com/pages/publications/85198902620.
DOI: https://doi.org/10.1109/IPDPS57955.2024.00040.

9. Le, T. L. & Vo, M.-H. “Lossless data compression algorithm to save energy in wireless sensor network”. *GTSD Proceedings Volumes*. 2018; 597–600, https://www.scopus.com/pages/publications/85061474826. DOI: https://doi.org/10.1109/GTSD.2018.8595614.

10. Sudheer, B. N. & Sujatha, K. “A brief survey on data aggregation and data compression models using blockchain model in wireless sensor network”. *ICIDCA Proceedings Volumes*. 2023. p. 406–413, https://www.scopus.com/pages/publications/85159115181.
DOI: https://doi.org/10.1109/ICIDCA56705.2023.10100009.

11. Chen, R. & Xu, L. “SnappyR: A new high-speed lossless data compression algorithm”. *DCC Proceedings Volumes*. 2023. p. 334–336. DOI: https://doi.org/10.1109/DCC55655.2023.00052.

12. Hanumanthaiah, A., Gopinath, A., Arun, C., Hariharan, B. & Murugan, R. “Comparison of lossless data compression techniques in low-cost low-power (LCLP) IoT systems”. *ISED Proceedings Volumes*. 2019. p. 1–5, https://www.scopus.com/pages/publications/85085731569.
DOI: https://doi.org/10.1109/ISED48680.2019.9096229.

13. Gusiev, O. & Magro, V. “Microgrid monitoring method for predicting energy anomalies”. *Scientific and Applied Journal Renewable Energy*. 2025; 80 (1): 22–28. DOI: https://doi.org/10.36296/1819-8058.2025.1(80).22-28.

14. Wu, Y., Xu, S., Xi, C., Nie, P., Jiang, W. & Hashimoto, S. “A dynamic and parallel two-stage lossless data compression method for Smart Grid”. *IEEE Access*. 2023; 11: 143475–143485. https://www.scopus.com/pages/publications/85180311830.
DOI: https://doi.org/10.1109/ACCESS.2023.3343436.

15. Jeong, H., Seo, G. & Hwang, E. “Lossless data compression with bit-back coding on massive smart meter data”. *Big Data Proceedings Volumes*. 2022. p. 6667–6669, https://www.scopus.com/pages/publications/85147907925. DOI: https://doi.org/10.1109/BigData55660.2022.10020726.

16. Chen, C., Wang, W., Yin, H., Zhan, L. & Liu, Y. “Real-time lossless compression for ultrahigh-density synchrophasor and point-on-wave data”. *IEEE Transactions on Industrial Electronics*. 2022; 69 (2): 2012–2021, https://www.scopus.com/pages/publications/85100928829.
DOI: https://doi.org/10.1109/TIE.2021.3057034.

17. Magro, V. I. & Panfilov, O. G. “Research of the features digital formation in satellite communication lines”. *Radio Electronics Computer Science, Control*. 2024; 68 (1): 28–40. DOI: https://doi.org/10.15588/1607-3274-2024-1-3.

18. Magro V. I. & Plaksin S. V. “Enhancement of communication quality in the X-band downlink channel for LEO earth observation satellites”. *Science and Innovation.* 2025; 21 (5): 76–88, https://www.scopus.com/pages/publications/105019955561. DOI: https://doi.org/10.15407/scine21.05.076.

19. Gummidela, S. C., Bhupani, S. M. K., Ikkurthi, G. S., Bandaru, T. L. & Vankayalapati, H. D. “Lossless telemetric data compression using Lempel-Ziv-Markov-algorithm”. *NEleX Proceedings Volumes*. 2023. p. 1–6, https://www.scopus.com/pages/publications/85186517063.
DOI: https://doi.org/10.1109/NEleX59773.2023.10421382.

20. Joshi, V. & Rani, S. J. “An efficient FPGA implementation of a simple lossless algorithm (SLA) for on-board satellite hyperspectral data compression”. *ISCAS Proceedings Volumes*. 2024. p. 1–6, https://www.scopus.com/pages/publications/85198517811.
DOI: https://doi.org/10.1109/ISCAS58744.2024.10558039.

21.Gorev, V., Gusev, A., Korniienko, V. & Shedlovska, Y. “On the use of the Kolmogorov–Wiener filter for heavy-tail process prediction”. *Journal of Cyber Security and Mobility*. 2023; 12 (03): 315–338, https://www.scopus.com/pages/publications/85168554337. DOI: https://doi.org/10.13052/jcsm2245-1439.123.4.

22. Gorev, V. N., Gusev, A. Y. & Korniienko, V. I. “Kolmogorov-Wiener filter for continuous traffic prediction in the GFSD model”. *Radio Electronics Computer Science, Control*. 2022; 31 (3): 315–338. DOI: https://doi.org/10.15588/1607-3274-2022-3-3.

23. Qin, L. & Sun, J. “Model compression for data compression: neural network based lossless compressor made practical”. *DCC Proceedings Volumes*. 2023. p. 52–61, https://www.scopus.com/pages/publications/85160778780.
DOI: https://doi.org/10.1109/DCC55655.2023.00013.

24. Gusev, O. Yu., Magro, V. I. & Nikolska, O. I. “Teletraffic forecasting in media service systems”. *Radio Electronics Computer Science, Control*. 2023; 67 (4): 7–15. DOI: https://doi.org/10.15588/1607-3274-2023-4-1.

25. Košuda, M., Lipovsky, P., Szöke, Z., Fil'ko, M., Novotňak, J. & Heško, F. “MAVLink messaging protocol as potential candidate for the UTM communication”. *NTSP Proceedings Volumes.* 2020. p. 1–7. https://www.scopus.com/pages/publications/85096533033.
DOI: https://doi.org/10.1109/NTSP49686.2020.9229550.

24. Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J. & Robinson, J. E. “Unmanned aircraft system traffic management (UTM) concept of operations”. *American Institute of Aeronautics and Astronautics.* – Available from: https://ntrs.nasa.gov/api/citations/20190000370/downloads/20190000370.pdf. – [Accessed: 10 February, 2026].

26. Gupta, A., Bansal, A. & Khanduja, V. “Modern lossless compression techniques: Review, comparison and analysis”. *ICECCT Proceedings Volumes.* 2017. p. 1083–1090, https://www.scopus.com/pages/publications/85040548144. DOI: https://doi.org/10.1109/ICECCT.2017.8117850.

27. Zhao, X. & Li, B. “Implementation of the LZMA compression algorithm on FPGA”. *EDSSC Proceedings Volumes.* 2017. p. 1–2, https://www.scopus.com/pages/publications/85043505873. DOI: https://doi.org/10.1109/EDSSC.2017.8126506.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest regarding this study, including financial, personal, authorship, or other interests that could influence the research and the results presented in this article

Received 14.04.2026
Received after revision 12.06.2026
Accepted 18.06.2026

**DOI: https://doi.org/10.15276/hait.09.2026.18**
**UDC 004.62.:004.45:629.7.05**

# Удосконалення алгоритму стиснення інформації в системі зв’язку з безпілотним літальним апаратом

**Магро Валерій Іванович**[1]
ORCID: https://orcid.org/0000-0003-4238-6733; magro.v.i@nmu.one. Scopus Author ID: 6602911357
**Корнієнко Валерій Іванович**[1]
ORCID: https://orcid.org/0000-0002-0800-3359; korniienko.v.i@nmu.one. Scopus Author ID: 56446921900
**Тимофєєв Дмитро Сергійович**[1]
ORCID: https://orcid.org/0000-0002-9718-6678; tymofieiev.d.s@nmu.one. Scopus Author ID: 55437340600
**Удовик Ірина Михайлівна**[1]
ORCID: https://orcid.org/0000-0002-5190-841X; udovyk.i.m@nmu.one. Scopus Author ID: 55998874400
[1] Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19. Дніпро, 49005, Україна

## АНОТАЦІЯ

**Актуальною** є проблема обробки великих обсягів даних, що передаються в бездротових мережах зв'язку з низькою пропускною здатністю є однією з основних перешкод для ефективного розгортання таких мереж. Найбільш раціональним шляхом розв’язку даної проблеми є проведення дослідження щодо вдосконалення методу стиснення, або використання комбінації методів стиснення даних, що дозволить зменшити обсяг та час передачі даних. **Метою дослідження** є збільшення стиснення даних що передаються між безпілотним літальним апаратом і пунктом керування та зменшення процесорного часу на стиснення одного пакету процесу кодування джерела для бездротового каналу зв’язку з використанням протоколу MAVLink в умовах обмежених ресурсів мікроконтролера безпілотного літального апарата. **Завдання**, що було поставлено в роботі полягало в розробці вдосконаленого алгоритма стиснення даних без втрат при його використанні в рамках протоколу Mavlink v2 та оцінці ефективності стиснення. **Методи**, які використовувались полягали у вдосконаленні потокового стиснення джерела при обміні даними з використанням протоколу MAVLink, що характеризується малими обсягами пакетів. Кодування джерела із стисненням без втрат здійснювалось шляхом формування побітних масок та попарного їх ущільнення відповідно до їх властивостей. **Наукова новизна** полягає розробці алгоритма послідовного парного кодування джерела для бездротового каналу зв'язку, який реалізує потокове кодування байтів структури пакетів, що забезпечує стиснення інформації без втрат. **Практична значимість** отриманих результатів полягає у можливості ефективного застосовані при розробці систем зв’язку для керування та обміні інформації між пунктом керування та безпілотним літальним апаратом за протоколом MAVLink. **Результтами** роботи є вдосконаленя алгоритму кодування джерела для ефективного стиснення без втрат пакетів MAVLink в системах зв’язку в умовах обмежених ресурсів безпілотного літального апарату та встановлення неефективності відомих алгоритмів для малих пакетів MAVLink. Порівняння ефективності стиснення запропонованого вдосконаленого алгоритму кодування джерела та відомих алгоритмів стиснення показало, що загальний обсяг пакетів зменшується на п’ятдесят чотири відсотки. Крім того, при використанні запропонованого алгоритму суттєво зменшуються витрати процесорного часу на стиснення одного пакету (в середньому стиснення одного пакету триває менше мікросекунди), що значно менше, ніж при використанні відомих алгоритмів стиснення. **Висновками** роботи є пропозиція вдосконаленого алгоритму стиснення для бездротового каналу зв'язку, який забезпечує більший ступінь стиснення у порівнянні з розглянутими алгоритмами під час потокового послідовно-парного кодування байтів пакетів протоколу MAVLink.

**Ключові слова:** потокова інформація; канал зв’язку; стиснення без втрат; пункт керування; протокол MAVLink; алгоритм стиснення

## ABOUT THE AUTHORS

**Valerii I. Magro** - PhD, Associate Professor, Information Security and Telecommunications Department. Dnipro University of Technology, 19, Dmytro Yavornytskyi Ave. Dnipro, 49005, Ukraine
ORCID: https://orcid.org/0000-0003-4238-6733; magro.v.i@nmu.one. Scopus Author ID: 6602911357
***Research field:*** telecommunications, radio engineering, radio physics, cybersecurity

**Магро Валерій Іванович** - кандидат фізико-математичних наук, доцент, професор кафедри Безпеки інформації та телекомунікацій, Національний технічний університет «Дніпровська політехніка», пр. Дмитра Яворницького 19. Дніпро, 49005, Україна

**Valerii I. Korniienko** - Doctor of Engineering Sciences, Professor, Head of Information Security and Telecommunications Department. Dnipro University of Technology, 19, Dmytro Yavornytskyi Ave. Dnipro, 49005, Ukraine
ORCID: https://orcid.org/0000-0002-0800-3359; korniienko.v.i@nmu.one. Scopus Author ID: 56446921900
***Research field:*** telecommunications, automation, cybersecurity

**Корнієнко Валерій Іванович** - доктор технічних наук, професор, завідувач кафедри Безпеки інформації та телекомунікацій, Національний технічний університет «Дніпровська політехніка», пр. Дмитра Яворницького, 19. Дніпро, 49005, Україна

**Dmytro S. Tymofieiev** - Senior Lecturer, Information Security and Telecommunications Department. Dnipro University of Technology, 19, Dmytro Yavornytskyi Ave. Dnipro, 49005, Ukraine
ORCID: https://orcid.org/0000-0002-9718-6678; tymofieiev.d.s@nmu.one. Scopus Author ID: 55437340600
***Research field:*** information technology, cybersecurity, information security management

**Тимофєєв Дмитро Сергійович** - старший викладач кафедри Безпеки інформації та телекомунікацій. Національний технічний університет «Дніпровська політехніка», пр. Дмитра Яворницького, 19. Дніпро, 49005, Україна

**Iryna M. Udovyk** - PhD, Associate Professor, Dean of Information Technologies Faculty. Dnipro University of Technology, 19, Dmytro Yavornytskyi Ave. Dnipro, 49005, Ukraine
ORCID: https://orcid.org/0000-0002-5190-841X; udovyk.i.m@nmu.one. Scopus Author ID: 55998874400
***Research field:*** information technology, computer science, software engineering, cybersecurity

**Удовик Ірина Михайлівна** - кандидат технічних наук, доцент, декан факультету Інформаційних технологій. Національний технічний університет «Дніпровська політехніка», пр. Дмитра Яворницького, 19. Дніпро, 49005, Україна