Chaos-based compression sensing on wireless sensor network: enabling a low-power and high-performance system

Gian Quoc Anh[1*], Cao Van The[1], Tran Duc Tan[2]

[1]Faculty of Electric-Electronic Engineering, Nam Dinh University of Technology Education, Nam Dinh, 07100, Viet Nam;

[2]Faculty of Electrical and Electronic Engineering, Phenikaa University, Hanoi, Vietnam.

[*]Corresponding author: gianquocanh@gmail.com

Received 18 Apr. 2023; Revised 28 Jun. 2023; Accepted 12 Jul. 2023; Published 25 Oct. 2023.

DOI: https://doi.org/10.54939/1859-1043.j.mst.90.2023.3-10

ABSTRACT

The energy of sensor nodes in operation mode is primarily consumed by the wireless transceivers. Therefore, reducing the transmitted data can lead to significant energy savings. Compressive sensing is a technique that can reproduce an original signal using a smaller number of samples than required by the Nyquist theorem, by exploiting the sparsity of the signal in the represented domain. In Wireless Sensor Networks, compressed sampling is performed at the sensor node, and decompression is performed at the sink node. However, the limited computing and resource constraints in sensor nodes should be taken into consideration when applying the compressed sensing technique. This paper proposes using a non-linear system to generate chaos-based coefficient sequences applied in the sensor nodes of a landslide warning system. The experimental study demonstrated that the sensor node utilizing pseudo-random sampling is faster and less complex in comparison to the sensor node employing random sampling.

Keywords: Compressed sensing; Pseudo-Random; Chaos; WSN; Landslide.

1. INTRODUCTION

Rainfall landslide early warning and monitoring system (EWS) using a wireless sensor network (WSN) is an effective tool for landslide management and risk reduction. Application of WSN in EWS is deployed in hard-to-reach environments where Sensor Nodes (SN) usually rely on battery power. In [1], a battery having a capacity of 6600 mA is utilized, and it can sustain SN to operate in continuous mode for only 7.6 days. This is insufficient to meet the minimum requirement of a rainy season. The duration of wireless transceiver operation is the key factor that contributes to energy consumption in the SN. To save energy, it is necessary to reduce the transmitted data. SNs typically sample signals and transfer them to the sink node without compression. Compressed sensing (CS) is a technique that uses a smaller number of samples than the Nyquist theorem to reproduce an original signal using a nonlinear algorithm. CS compresses data at the sampling stage of the data acquisition process, reducing the amount of data that needs to be handled by the sensor. Compressed sensing is based on the fact that a sparse signal can be restored with a small number of random samples under certain constraints [2, 3]. Sub-Nyquist sampling (sampling fewer samples than the Nyquist theorem) reduces the number of samples to be transmitted and therefore reduces energy consumption in transmitting data [4]. The compressed sensing technique has been applied widely in WSNs [5, 6]. This technique includes two main processes: 1) Sensing at SN, and 2) Nonlinear reconstruction in the sink node. During operation, SNs in the EWS continuously collect a large amount of data from acceleration sensors at different depths to analyze vibrations and calculate displacement to monitor the slope state. The process of sampling the signal is performed at the SNs, while the decompression is carried out at the sink node. The application of CS techniques should consider the limited computing and resource constraints in the SNs.

Random matrices are frequently utilized as measurement matrices in compressed sensing because they can efficiently sample the signal. The authors in [7] used a randomly generated coefficient sequence to encode and reconstruct the original signal. However, generating a random sequence to determine the sampling location in k-space required a high number of calculations or specialized hardware. This costs resources, especially in SNs with low memory and limited processing power. It has been demonstrated that the pseudo-random sampling approach can recover data with a quality comparable to the random sample solution [8, 9]. Experimentation using the Monte Carlo method for evaluation demonstrated the effectiveness of the compressed sensing that uses a pseudo-random sequence compared to the one that uses a random sequence [8].

This paper proposes using a nonlinear system to generate pseudo-random coefficient sequences for compressed sensing. A pseudo-random number generator (PRNG) uses an algorithm to generate numbers with a uniform distribution but in a deterministic manner. Instead of storing the whole sensing matrix, we simply require to save the matrix seeds, such as the initial value, chaos parameters, sampled start location, and sample step. With the improved pseudo-random code, the decoder can easily reproduce the coefficients if the parameters of the code generator are known. The experiment results indicate that using a pseudo-random sequence in the compressed sensing system enables the reproduction of the same signal as with a random sequence, with reduced execution time on the device. The remainder of this paper is structured as follows. In item 2, we provide an overview of compressed sensing and random number generator algorithms. Following this, we present the experiment in item 3. We report the experiment result and discussion in item 4. Lastly, we conclude in item 5.

2. THEORETICAL FOUNDATIONS

2.1. Compressed Sensing

CS relies on the sparsity of the signal in the represented domain. Sparsity is related to the nature of the signal when it is represented on a suitable basis. In real-world applications, $\boldsymbol{x}$ is typically non-sparse, but it can often be represented in a sparse form in an alternative domain. In this form, it only contains a few important elements, whereas the majority of the elements are zero or almost zero. The compressed sensing technique is applied by sending a small amount of randomly sampled coefficients corresponding to the signal to the central station. The original signal $\boldsymbol{x}$ is compressed into a shorter signal $\boldsymbol{y}$ with a compression ratio depending on the sparsity of the signal $\boldsymbol{x}$. The number of measurements taken to reconstruct a signal is determined by the signal's sparsity, rather than its bandwidth.

$$y = \Phi x \tag{1}$$

where the measurement matrix Φ ($\Phi \in R^{M\times N}$) is related to the compressing process, while $\boldsymbol{y}$ represents a one-dimensional measurement with length M. If $\boldsymbol{x}$ is a sparse signal, the length of $\boldsymbol{y}$ is much shorter than the N values required to satisfy the Nyquist sampling theorem. In WSN, this transformation is done by SNs. To accurately retrieve the original signal, the measurement matrix Φ must fulfill the Restricted Isometry Property [10]. Finding a suitable measurement matrix that enables a sparse representation of the vibration is not evident. Typically, selecting such a measurement matrix is done through trials consider both the characteristics of the signal under analysis and the desired level of compression. The effectiveness of CS is determined by both the degree of sparseness in the signal and the algorithm employed for reconstruction.

The central station receives the compressed data and uses a nonlinear algorithm such as Orthogonal Matching Pursuit (OMP) to reconstruct the original data [9]. Specifically, the logic map is converted into a sequence with Gaussian behavior defined by the equation [11]:

$$q_L(n+1) = \rho q_L(n)\big(1 - q_L(n)\big) \tag{2}$$

In which ρ is a control parameter; n=0, 1,... , L-1; $q_L(0)$ is the initial condition. A small change in $q_L(0)$ will lead to a large change in the value of $q_L(n)$. The original signal can be reconstructed using the l1 regularized least-squares approach [12]. The problem to be solved is

$$\begin{gathered} \arg\min_{x}\{\|F_u x - y\|_2^2 + \lambda\|\Phi x\|_1\} \\ Subject\ to\ \|F_u x - y\|_2 < \varepsilon \end{gathered} \tag{3}$$

In which λ is the adjustment constant and F_u is the Fourier operator being sampled. Advanced signal processing techniques are used to perform compressed sampling, such as L1-norm optimization or non-linear optimization to recover the original signal from compressed samples by utilizing the sparsity property. These algorithms also minimize the amount of data collected and stored while preserving the important information of the signal. Therefore, assuming a signal has sparse properties and utilizing advanced signal processing techniques, compressed sampling can effectively be performed on signals obtained from sensors.

2.2. TRNG and PRNG Algorithm

Compressed sensing allows for the exact recovery of a signal from a random sample of the signal. In the sensing step, we use a measurement matrix, in which random matrices are the most commonly used, to obtain a sparse signal. Random sampling is related to the inconsistency property in CS, to reproduce the original signal with high accuracy [3]. The true random number generator (TRNG) program generates truly random numbers. The circuit is created utilizing a microcontroller that employs an ADC module to read the noise at the microcontroller pin [13]. The steps generate true random numbers, as shown in figure 1.

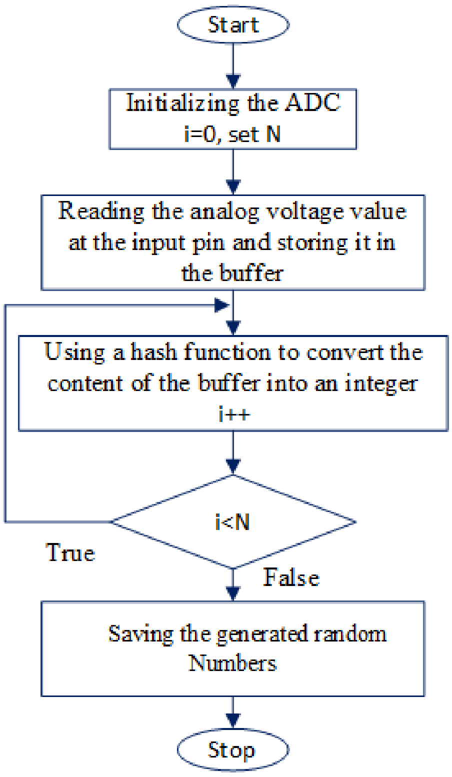


Figure 1. *The steps generate true random numbers.*

The PRNG program generates numbers with uniform distribution but are actually generated by a deterministic algorithm. To save energy and limit memory resource usage, the PRNG algorithm using the Linear Congruential Generator (LCG) method is employed. LCG is a simple and fast algorithm, widely used in many applications due to its ease of implementation and relatively good randomness. The simple steps in generating pseudo-random numbers using LCG are as follows [14]:

1. Establish the constants a, c, and M (where M is a positive integer, 0 < a < m, 0 ≤ b < m).
2. Set x as the initial value (0 ≤ x < M).
3. Implement the loop:

$$x_{n+1} = (a * x_n + c) \, mod \, M \tag{4}$$

Store the value of x_{n+1}

4. Return the random sequence.

To implement the LCG algorithm, the values of ***a***, ***c***, and ***M*** are selected based on specific mathematical properties to ensure that the generated sequence is pseudo-random [15]. The values of ***a*** and ***c*** must satisfy certain conditions to ensure that the generated sequence has good random properties, such as long period, uniform distribution, and good statistical properties.

3. SOLUTION AND EXPERIMENTS

Our main goal is to examine the potential of compressed sensing in WSN energy-efficient sensing. In this study, we evaluate the performance of compressed sensing with a pseudo-random sequence. The execution time compared between generating pseudo-random and random sequences on sensor nodes. The same pseudo-random number generator is implemented by sensor nodes in WSNs to produce a measurement matrix at each node. The SNs take random sub-Nyquist measurements of the signal and transmit them back to a central station. Then, the central station reconstructs the signal using OMP. The reconstructed signal is compared to the signal that was measured by the SNs using traditional uniform Nyquist sampling. We experimented with a value of N = 128 for the number of accelerator data points needed for the reconstruction process. In each experiment, a fixed N value was used to reconstruct the data with different compression ratios.

The effectiveness of the reconstruction system and the impact of the compression ratio were analyzed using a parameter called relative reconstruction error (e):

$$e = 100\% \frac{1}{L} \sum_{i=1}^{L} \frac{|x_i - \hat{x}_i|}{|\hat{x}_i|} \tag{5}$$

where L is the total amount of data used for the calculation, x represents the original data and $\hat{\boldsymbol{x}}$ represents the reconstructed data.

The SN uses a Waspmote PRO V1.2 main board, an XBee-PRO ZB wireless communication module, a temperature sensor, and accelerometers (figure 1). The XBee-PRO ZB module, which uses Zigbee protocol, is applied in WSN because it has low power consumption. Each SN is powered by a 3.7 V, 6600 mAh rechargeable lithium battery. To function as routers, SNs use a solar battery to supplement the power provided by the lithium battery.

Accelerometer sensors are used to determine slope deformation, including vibration, distance, and displacement speed at the monitored location. MEMS accelerometers have a good frequency response in low-frequency regions, which makes them suitable for replacing geophones in tilt measurement. The ADXL335 accelerometer is mounted in series to measure displacement at different depths to determine the depth of the slip surface. The tilt angle is determined on the basis of the gravity vector and its projection on the sensor axes. The EWS consists of SNs located on the slope (figure 2). We have published a number of articles that report the experiments and data in [16, 17].

The placement of the central station is determined based on survey results and numerical model analysis. The central station is placed in a safe location near the slope to ensure regular operation in the landslide case. In warning mode, when slope stability is low, the priority of the system is to monitor data and measure vibration. As a result, the start/stop cycle at the SN is shortened to one minute. The data recorded from the SNs in EWS is mainly low-frequency data. At each SN, acceleration is continuously sampled over a period of time. Next, the data is transformed from the time domain to the frequency domain using the Fourier transform, pseudo-random sampling, encapsulating, and transmitted wirelessly to the sink node via the XBee Pro ZB module. The sink node receives data from the sensor and reconstructs the original signal.

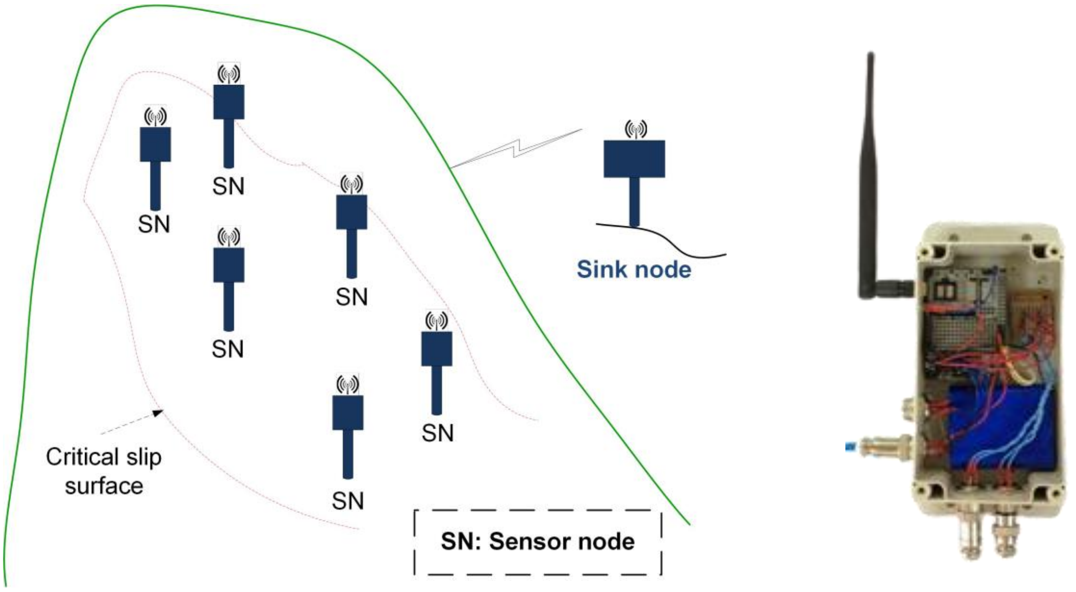


Figure 2. *Distribution of SNs on slopes and sensor node image**.***

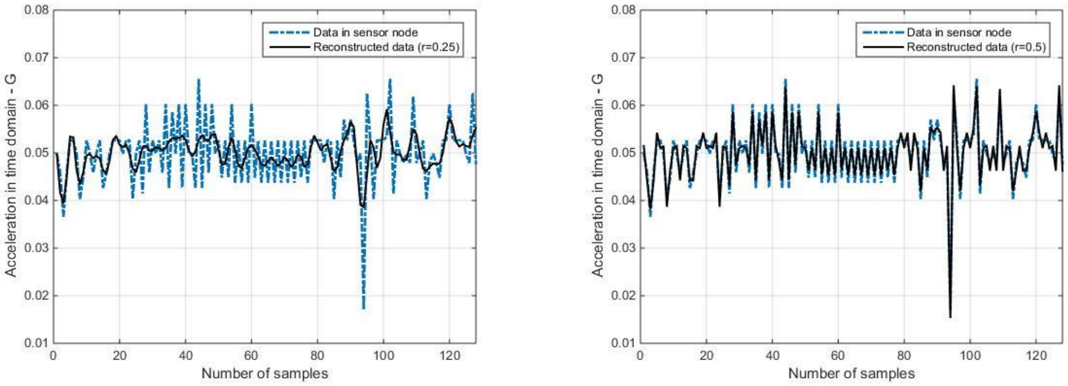
The experiment was conducted on the Waspmote Pro v1.2 control board, which utilizes the Atmega1281 microcontroller (frequency 14.7456 MHz, SRAM 8 KB, Flash 128 KB). The board does not support specialized hardware for generating random values. The program was written in C++ and compiled using the Waspmote Pro IDE v06.31. Each algorithm was evaluated as a function. The execution time of the function depends on many factors, including the processing speed of the microcontroller and the length of the sequence to be generated. To measure the execution time of the function, it needs to be executed multiple times, and the average execution time must be calculated. Specifically, in the experiment, the function was called repeatedly 1000 times to calculate the average execution time. The Waspmote has an integrated real-time clock with millisecond accuracy, which is used to evaluate the start and end times of the execution process.

4. RESULTS AND DISCUSSION

3.1. Results

Figure 3 compares the reconstructed and original data from the accelerometer sensor for a compression ratio of $r = 0.25$ and $r = 0.5$. It can be observed that there is a significant error between the actual and acquired data for a compression ratio value $r = 0.25$.

Figure 4 illustrates the impact of compression ratio on relative error. It can be observed that the error is higher for compression ratios ranging from 0.25 to 0.5. However, if the compression ratio is greater than or equal to 0.55, the error rapidly decreases. These results suggest that k-space data is sufficient for reconstructing the original data. In this paper, the compressed sensing technique was applied by sending a small amount of randomly sampled Fourier coefficients of the signal to the central station. By exploiting the natural properties of the data, as seen in figure 3, or by observing the amplitude spectrum of the data, it can be observed that there always exists a one-dimensional component (zero frequency) with a definite value.



a) r = 0.25 b) r = 0.5

Figure 3. *Data from accelerometer with a) r = 0.25; b) r=0.5.*

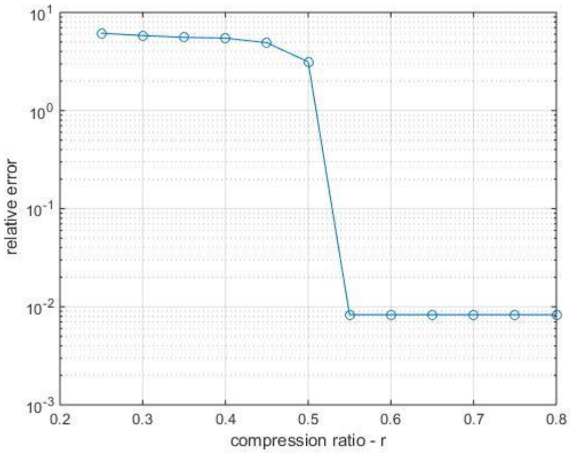


Figure 4. *Effect of compression ratio on relative error.*

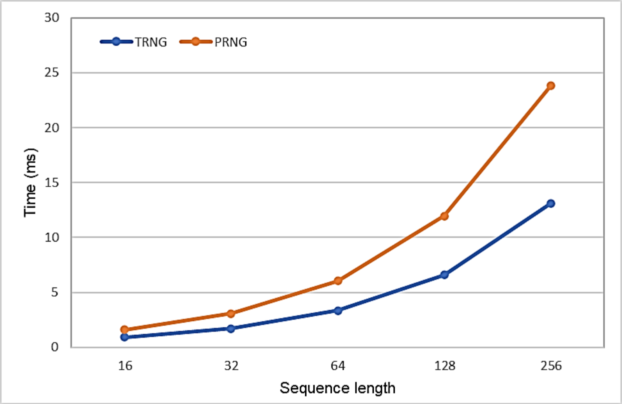


Figure 5. *Compare the execution time of pseudo-random and random sequence generation algorithms.*

In this study, the authors conducted experiments to compare the performance of pseudo-random sequences and truly random sequences generated by a microcontroller. The results of the time measurement for generating an 8-bit integer sequence are shown in figure 5. The data showed that the execution time varies depending on the length of the generated random sequence. For a sequence of length 16 elements, the corresponding execution times for TRNG and PRNG are 0.92 milliseconds and 0.68 milliseconds, respectively. As the sequence length increases to 256 elements, the corresponding times for generating TRNG and PRNG sequences are 13.01 milliseconds and 10.73 milliseconds. In all experiments, the pseudo-random sequence generation method had a shorter execution time for sequences of any length.

3.2. Discussion

This paper improved the creation of the chaotic sequence by selecting the position corresponding to zero frequency in the chaotic sequence. If the index position created by the chaotic sequence also selects the same position, the program will automatically increase an additional index to ensure the required number of coefficients. The advantage of improvement assurance in the chaotic system always has a corresponding index with one-dimensional frequency component sampling. This operation ensures the restoration of the components in the sampled signal and shortens the chaotic sequence by one sample compared to the work [8].

However, there are some limitations to using compressed sampling. When the amount of data is reduced, the amount of information that can be extracted from the data is also reduced. This can lead to a decrease in the accuracy of applications that use the signal. In addition, reducing the amount of data may make it more difficult to restore the original signal from the compressed samples, especially in cases where the compressed samples are not sufficient to represent the entire original signal (in this paper, the selected compression ratio is 0.55). Therefore, compressed sampling is an effective method to reduce energy consumption in SN, but careful consideration is also required for the limitations and difficulties that may arise when using this method.

PRNGs are found to be less complex, easier to implement, and quicker than TRNGs. However, PRNGs are not as safe and do not generate truly random numbers. On the other hand, TRNGs are more complex and slower, but are commonly used in applications that require high security.

4. CONCLUSIONS

The present study assesses the efficacy of pseudo-random sequences in sub-Nyquist sampling and reconstructing the original signal on microcontrollers. The experimental outcomes indicate that, by employing an appropriate compression ratio, a minimally erroneous reproduction of the original signal can be achieved. The paper compares and evaluates two different types of random number generators: Pseudo-Random Number Generators (PRNG) and True Random Number Generators (TRNG). In this paper, the PRNG initialization sequences are chaotic and based on the work of [8]. Using a pseudo-random sequence results in better performance than a random sequence to compress the signal on the microcontroller.

REFERENCES

[1]. C. D. Nguyen, T. D. Tran, N. D. Tran, H. T. Huynh, and D. T. Nguyen, *"Flexible and Efficient Wireless Sensor Networks for Detecting Rainfall-Induced Landslides,"* International Journal of Distributed Sensor Networks, vol. 11, no. 11, p. 13, (2015).

[2]. C. Zhao, B. Tang, Y. Huang, and L. J. I. T. o. I. I. Deng, *"Edge Collaborative Compressed Sensing in Wireless Sensor Networks for Mechanical Vibration Monitoring,"* (2022).

[3]. E. J. Candès and M. B. Wakin, *"An introduction to compressive sampling,"* IEEE signal processing magazine, vol. 25, no. 2, pp. 21-30, (2008).

[4]. V. Abolghasemi and M. H. J. I. S. L. Anisi, *"Compressive sensing for remote flood monitoring,"* vol. 5, no. 4, pp. 1-4, (2021).

[5]. D.-g. Zhang, T. Zhang, J. Zhang, Y. Dong, and X.-d. Zhang, *"A kind of effective data aggregating method based on compressive sensing for wireless sensor network,"* EURASIP Journal on Wireless Communications Networking, vol. 2018, no. 1, pp. 1-15, (2018).

[6]. Q. Wang, D. Lin, P. Yang, and Z. J. I. S. J. Zhang, *"An energy-efficient compressive sensing-based clustering routing protocol for WSNs,"* vol. 19, no. 10, pp. 3950-3960, (2019).

[7]. D.-C. Nguyen, D.-T. Tran, and D.-N. Tran, *"Application of compressed sensing in effective power consumption of WSN for landslide scenario,"* in Asia Pacific Conference on Multimedia and Broadcasting (APMediaCast), Bali, Indonesia, pp. 111-115, (2015).

[8]. N. Linh-Trung, D. Van Phong, Z. M. Hussain, H. T. Huynh, V. L. Morgan, and J. C. Gore, *"Compressed sensing using chaos filters,"* in Australasian Telecommunication Networks and Applications Conference, pp. 219-223: IEEE, (2008).

[9]. J. A. Tropp, M. B. Wakin, M. F. Duarte, D. Baron, and R. G. Baraniuk, *"Random filters for compressive sampling and reconstruction,"* in 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 3, pp. III-III: IEEE, (2006).

[10]. E. J. J. C. r. m. Candes, *"The restricted isometry property and its implications for compressed sensing,"* vol. 346, no. 9-10, pp. 589-592, (2008).

[11]. J. C. Sprott and J. C. Sprott, *"Chaos and time-series analysis. Oxford University Press"*, (2003).

[12]. S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *"An interior-point method for large-scale l1-regularized least squares,"* IEEE journal of selected topics in signal processing, vol. 1, no. 4, pp. 606-617, (2007).

[13]. S. Matsuoka, S. Ichikawa, and N. Fujieda, *"A true random number generator that utilizes thermal noise in a programmable system‐on‐chip (PSoC),"* International Journal of Circuit Theory and Applications, vol. 49, no. 10, pp. 3354-3367, (2021).

[14]. P. L'ecuyer, *"Tables of linear congruential generators of different sizes and good lattice structure,"* Mathematics of Computation, vol. 68, no. 225, pp. 249-260, (1999).

[15]. D. V. Origines, A. M. Sison, and R. P. Medina, *"A Novel Pseudo-random number generator algorithm based on entropy source epoch timestamp,"* in 2019 International Conference on Information and Communications Technology (ICOIACT), pp. 50-55: IEEE, (2019).

[16]. Q. A. Gian, D.-T. Tran, D. C. Nguyen, V. H. Nhu, and D. Tien Bui, *"Design and implementation of site-specific rainfall-induced landslide early warning and monitoring system: a case study at Nam Dan landslide (Vietnam),"* Geomatics, Natural Hazards and Risk, vol. 8, no. 2, pp. 1978-1996, (2017).

[17]. G. Q. Anh, T. D. Tan, N. D. Chinh, and B. T. Dieu, *"Flexible Configuration of Wireless Sensor Network for Monitoring of Rainfall-Induced Landslide,"* Indonesian Journal of Electrical Engineering and Computer Science, vol. 12, no. 3, pp. 1030 -1036, (2018).

TÓM TẮT

Nén dữ liệu thông qua sử dụng chuỗi hỗn loạn trong mạng cảm biến không dây: tối ưu hóa hiệu quả năng lượng và hiệu suất

Năng lượng tiêu thụ chủ yếu của các nút cảm biến trong mạng không dây được sử dụng bởi bộ truyền nhận. Do đó, giảm lượng dữ liệu truyền có thể dẫn đến tiết kiệm năng lượng đáng kể. Kỹ thuật lấy mẫu nén (Compressed sensing - CS) có thể tái tạo tín hiệu gốc bằng một số mẫu nhỏ hơn so với yêu cầu của định lý Nyquist bằng cách khai thác tính thưa của tín hiệu trong miền biểu diễn. Tuy nhiên, việc áp dụng kỹ thuật nén dữ liệu cần phải xem xét đến hạn chế về khả năng tính toán và tài nguyên trong các nút cảm biến. Bài báo này đề xuất sử dụng hệ thống phi tuyến để tạo ra chuỗi hệ số hỗn loạn (chaos-based coefficient sequences) được áp dụng trong hệ thống cảnh báo lở đất. Thực nghiệm cho thấy nút cảm biến sử dụng mẫu giả ngẫu nhiên với chuỗi hệ số hỗn loạn có thể nhanh chóng hơn và ít phức tạp hơn so với sử dụng mẫu ngẫu nhiên. Do đó, phương pháp đề xuất có thể giúp tối ưu hóa hiệu suất và tiết kiệm năng lượng trong mạng cảm biến không dây.

Từ khoá: Lấy mẫu nén; Chuỗi giả ngẫu nhiên; Chuỗi hỗn loạn; Mạng cảm biến không dây; Trượt lở đất.