International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-12 Issue-2, December 2022

Development of a Shorted Interleaved Reed-Solomon Codes (siRS) for data downlink in Stratospheric Probes and Nano-Satellites

Eduardo Valadez Campos, J. Eduardo Mendoza Torres

Abstract*: *In the last 10 years, the advancement in the electronics have made possible the development of more sophisticated unmanned vehicles and probes, manufactured using Components-Of-The-Shelf (COTS) available to public, whether be for industrial of scientific purposes. In the matter of aerospace exploration, the CubeSats are the best example of sophisticated devices partially of entirely developed using this commercial COTS. The financial cost of some components is still a point of consideration in the design, but it is not entirely a show-stopper since some low-cost components can be optimized to meet the necessary requirements and increase the reliability. In the case of stratospheric probes and nano-satellites, the Telemetry, Tracking and Command (TT&C) subsystem, in charge of the data uplink/ downlink, can be improved by including an Error Correcting Code (ECC), making possible to use commercial transceivers of low-power are low-cost. In this article we describe an implementation of an interleaved and shortened Reed-Solomon Code, which we called siRS, to improve the image and telemetry downlink in stratospheric probes and nano-satellites, using entirely low-cost commercial RF transceivers and an ARM-based mini-PC.

Keywords*: *Transceiver; Reed-Solomon Codes; Error Correcting Codes; Components-Off-The-Shelf; ARM-based; Galois field; codec; transpose; Forward Correcting Code.

I. INTRODUCTION

The use of Components-Off-The-Shelf (COTS) based solutions has allowed to develop radio-probes, payloads and mini-satellites, with more innovative designs without running a major financial risk, since these take the full advantage of the economy of scale with cheap components rather than the conservative industry where only large agencies could afford them[1][2][3].

In order to choose and possibly increase the reliability of such components, it is necessary to make an assessment of where the final system would be operating and identify the poor performance points. It is not an easy task, since it is required a good planning and to think in all possible scenarios where a risk of failure might present.

Manuscript received on 18 November 2022 | Revised Manuscript received on 25 November 2022 | Manuscript Accepted on 15 December 2022 | Manuscript published on 30 December 2022.

*Correspondence Author (s)

Eduardo Valadez Campos*, Department of Astrophysics, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México. Email: eduardo.valadez@inaoep.mx

J. Eduardo Mendoza Torres, Department of Astrophysics, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México. Email: mend@inaoep.mx

The RF systems based in COTS has been adopted quite well, including navigation systems, transmission data systems for deep-space and remote-sensing applications and modules for communication with Earth in Low-Earth-Orbit (LEO) [1].For the moment we are going to focus ONLY in the development of the TT&C subsystem for stratospheric radio-probes and mini-satellites with components easily available to public. Let's say we are developing a radio-probe that needs to transmit images and environmental data. Usually, for commercial ARM-based mini-PCs, the data gathered from the sensors and cameras, is stored in a compressed format to save memory, for example the images can be stored as a JPEG. Then the data is passed, via UART, i2C or SPI port, to a modular transceiver, which is an electronic device that has both the functions of RF transmitter and receiver. The transceiver transmits the data usually between 1200 to 19200 bps as a RF modulation signal (FSK, GFSK, BPSK, etc.) with a signal power going from 20 mW to 500 mW approx. Depending of the price, some can have a communication protocol like the ZigBee or just a simple Cyclic Redundancy Check (CRC) or simple parity check.

From this description, we can identify 3 main weak points in the communication system.

1) The data compression –Some data formats are susceptible to corruption when a couple bytes in the wrong place present errors.
2) The signal power – Due to low signal power, the Signal-to-Noise (SNR) ratio may be too low when the distance between the sender and the receiver increases.
3) The transfer rate – Most of the transceivers do not transmit above 57600, being 9600 the most common bit rate among CubeSats [6][7]. Also, the target is moving, so the observation window is short and narrow.

For the point 2, the signal power can be compensated by constructing high-gain antennas to increase the SNR. Yagi and circular polarization antennas can be used. As for the other points, the implementation of a Forward Error Correction (FEC) can optimize the communication link.

The FEC are a type of ECC that does not require handshaking between the source and the destination. Introducing a channel coding in a communication system introduces the possibility of approaching the maximum transmission rate theoretically possible [4]. The FEC method is favorable because have a lower consumption of bandwidth and a high-efficiency error correction. This is the reason FEC is suitable for applications in wireless communication [5].

We have already developed a stratospheric balloon probe that implements our version of Reed-Solomon code RS(255,223)[2][3].The transceivers are commercial COTS working in the band of 433MHz, with a bandwidth of 1KHz, GFSK modulation and a max baudrate of 9600 bps with a max range of no more than 1000 m in line-of-sight. These transceivers don't have implemented a communication protocol such as Zigbee, hence the low-cost. With the RS algorithm we were able to achieving transmissions of telemetry and images (as JPEG files) at an altitude of 10,000 m.a.s.l. [2]. Making the probe as well as our RS algorithm flight-tested and reliable.

Now, the main goal is to implement a shorten-interleaved Reed-Solomon, that we called siRS. Then run some tests using an unformatted grayscale image and a JPEG image as data to be encoded, pass it though a random Burst Bit Error Channel (BBEC). The Bit-Error Rate (BER) is then measured experimentally by comparing the data before and after. Also, we run a real-world test using an ARM-based OBC with the software developed for the stratospheric probes with the siRS algorithm integrated and again pass the JPEG data through the BBEC and transmit them over RF at a distance of 500 m, using commercial transceivers with a signal power of 20 mW.

II. RELATED WORK

Since the early beginnings of the use of unmanned probes for space exploration, the error correcting codes have been key for the success of the such endeavors. The NASA's 1971 Mariner Mars orbiter mission, launched on May 30 of that year, the main downlink code was a (32,6) biorthogonal Reed-Muller code. The bulk of the data was in the form of digital images of the surface of Mars, for which a decoded error probability of 5×10^{-3} was acceptable for a TV transmission [18]. However, the first use of Reed-Solomon in a deep-space mission were in the both voyager 1 and 2probes, which began in 1977 towards the planes of outer solar system [19], and it has since been a standard in the TT&C subsystems for missions like the Cassini-Huygens [20], the Juno probe, currently orbiting Jupiter, or the Mars Science Laboratory rover.

In our particular case, the need for develop a ECC began when we required to establish a robust RF link between our ground station and a prototype of stratospheric probe, we called Zbezda, to transmit telemetry of environmental sensors, as well as color images taken by a camera [3]. Since the probes were meant to be a platform to test other components in extreme environment conditions, this RF link required to be done using just COTS of low-cost to maintain the budget of the main project within the limits, hence the need to develop our own version of the algorithm of the Reed-Solomon code.

III. REED-SOLOMON CODES

The Reed-Solomon code (RS) is a particularly interesting and useful class of linear block codes. They belong to a class of error-correcting codes that are constructed using polynomials over a Galois field [9] [10]. One of the main characteristics of the RS is that are able to detect and correct multiple soft (erroneous bytes) and hard errors (erasures bytes) [8].

A RS code is specified as RS (n, k) with m-bit symbols. This means that the encoder takes*k*datasymbols of *m*bits each and adds parity symbols to make an symbol codeword. There are*n*-*k*parity symbols of *m*bits each (Fig. 1). A Reed-Solomon decoder can correct up to *t* symbols that contain errors in a codeword, where *2t* = *n* - *k*.

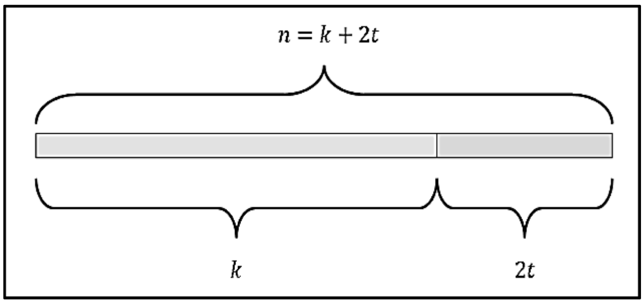


Fig. 1. Reed-Solomon codeword sizes

William A. Geisel from NASA in 1990 and C.K.P. Clarke in 2002 from the BBC both wrote a handbook independently to describe the mathematics behind the implementation of the original Reed-Solomon code [23][24]. Some of the equations are shown to describe briefly the process followed.

A. Galois field

A Galois-field or finite-field is a set of elements in which we can do addition, subtraction, multiplication, and division without leaving the set. The operation of addition and multiplication must satisfy the commutative, associative, and distributive laws. The number of elements in a field is called the order of the field and, according to Galois, in order for a filed to be finite, the numbers of elements are$p^m$where $p$is a prime number and m is a positive integer (Eq. 1)

$$GF(P^m) = \{0,1, \dots, P^{m-1}\} \quad (1)$$

We are going to focus in the binary field $GF(2)$ and the extension field $GF(2^m)$ (Eq. 2 and 3), where addition is EXCLUSIVE OR (XOR) and multiplication is AND. Since the only invertible element is 1, division is the identity function [21].

$$GF(2) = \{0, 1\} \quad (2)$$

$$GF(2^m) = \{0,1, \alpha, \alpha^2, \alpha^3, \dots, \alpha^{2^m-2}\} \quad (3)$$

Since we are working with 8-bit symbols, m = 8.

$$GF(2^8) = \{0,1, \alpha, \alpha^2, \alpha^3, \dots, \alpha^{254}\} \quad (4)$$

B. Primitive polynomial and field symbols

Polynomials over the binary field $GF(2^m)$ are any polynomials with binary coefficients. Each of these polynomials, denoted as $f(x)$, is simply the product of its irreducible factors [21].

An irreducible polynomial $f(x)$ of degree m is said to be primitive $p(x)$ if the smallest positive integer n for which $f(x)$ divides $x^n + 1$ is $n = 2^m - 1$.

Retrieval Number:100.1/ijeat.B39221212222
DOI: 10.35940/ijeat.B3922.1212222
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The primitive polynomials are polynomials that generate the elements contained in a finite field which in turn are needed to define the RS Codes. The primitive polynomial determined for a $GF(2^8)$used for our algorithms are shown in the table 1

Table 1. Primitive polynomials P(x) of the Galois field GF (256). The P(x) marked in yellow is the one used for our algorithm

$x^8+x^4+x^3+x^2+1$
$x^8+x^5+x^3+x+1$
$x^8+x^5+x^3+x^2+1$

The field symbols or primitive elements α^ithat conform $GF(2^m)$ are determined using the modulo method (Eq. 6)

$$\alpha^i = \alpha(x) \text{mod } p(x) \quad (5)$$

C. Generator polynomial

A RS codeword is generated using the generator polynomial, it is defined by the size of the parity-bytes and composed of the minimal polynomial [11].

Once the size of the parity-bytes $(2t)$ is chosen, that is powers of $x \in GF(2^m)$, it is proceeded to compute all their minimal polynomials in order to produce the generator polynomial. For each consecutive power of α, the minimal polynomial will be $x - \alpha^i$ and the generator polynomial is constructed using the next expression:

$$g(x) = \prod_{i=FCR}^{FCR+2t-1}(x-\alpha^i) \quad (6)$$

Where FCR is the power of the first consecutive root in $g(x)$. The proposed value 2t = 18 and the first consecutive root FCR = 0, therefore:

$$g(x) = \prod_{i=0}^{17}(x-\alpha^i) \quad (7)$$

$$g(x) = (x-1)(x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{17}) \quad (8)$$

D. Encoding

Now with the generator polynomial, the parity-check polynomial can be defined by using the next expression [22]:

$$parity(x) = x^{n-k}M(x) \;\; mod \;\; g(x) \quad (9)$$

And then we can define the codeword as:

$$c(x) = x^{n-k}M(x) + parity(x)$$

$$c(x) = M_{k-1}x^{n-1} + \cdots + M_0x^{n-k} + r_{n-k-1}x^{n-k-1} + \cdots + r_0 \quad (10)$$

E. Decoding

We define the received data as r(x) and is the sum of the codeword c(x) with added errors E(x)

$$r(x) = c(x) + E(x) \quad (11)$$

The decoding process takes a quite large process that the RS encoding. The next steps are the standard methods to do so [23]:

1) Calculate the syndrome components from the received word. A syndrome vector tells you which equations are not satisfied in a received codeword. An all zero syndrome tells you that all parity check equations are satisfied [23][24]. The i-th syndrome is defined in equation 13. Where the error locators are $X_j = \alpha_j^i$ and the error values are defined as $Y_j = E_{i_j}$ (Eq. 14)

$$S_i = c(\alpha^i) + E(\alpha^i) \quad (12)$$

$$S_i = \sum_{j=1}^{v} E_{i_j}(\alpha^i) = \sum_{j=1}^{v} Y_jX_j^i \quad (13)$$

2) Calculate the error-locator polynomial from the syndrome components [23]. In order to solve the equation system given by equation 14, it is precise to determine the Error-locator polynomial, which has got roots defined as $X_1^{-1}, X_2^{-1}, \dots, X_v^{-1}$.

$$\Lambda(x) = \prod_{j=1}^{v}(1-xX_j)$$

$$\Lambda(x) = 1 + \Lambda_1x + \cdots + \Lambda_{v-1}x^{v-1} + \Lambda_vx^v \quad (14)$$

Substitute the equation of the syndromes (Eq. 14) in the equation 15 and we get the next linear system:

$$\Lambda(x) = 1 + \Lambda_1S_{i+v-1} + \cdots + \Lambda_vS_i \quad (15)$$

To solve such system, there are several algorithms to do so, the most popular are the Berlekamp-Massey algorithm [12][13] and the Peterson–Gorenstein–Zierler algorithm [14]; and to find the roots of $\Lambda(x)$ of the error locator polynomial it can be used the Chien search algorithm [15].

3) Calculate the error values from the syndrome components and the error-locator X_k numbers. This can be achieved by using the Forney algorithm [11][23][24].

$$E_{i_j} = X_j\frac{\Omega(x_j^{-1})}{\Lambda'(x_j^{-1})} \quad (16)$$

Where the polynomial $\Lambda'(x)$ is defined has:

$$\Lambda'(x) = \sum_j^t \Lambda_jx^{j-1} \quad (17)$$

And the polynomial Ω(x), called the error-evaluator polynomial, is computed as follows:

$$\Omega(x) = S(x)\Lambda(x)\text{mod}x^{2t-1} \quad (18)$$

4) Calculate the decoded code word from the received word. By having located the symbols containing errors, identified by X_j and calculated the values Y_j of those errors, the errors can be corrected by adding the error polynomial E(x) o the received polynomial R(x) [23].

$$r(x) = c(x) + E(x)$$
$$c(x) = r(x) - E(x)$$

And in Galois algebra, (-) is equal to (+), therefore:

$$c(x) = r(x) + E(x) \quad (19)$$

IV. THE SHORTENING OF RS|

Mathematically speaking, the generation of the codeword for a regular RS and our algorithm for generate the siRS is the same; both use the same Galois Field $GF(2^8)$, and the same primitive polynomials. The shortening consists in encoding an array of length = k but the useful symbols have a length < k, the rest are blanks bytes with a determined value, in our case just zeros [18].

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

This allows it to be encoded as a codeword of length n but with a useful information of $n_s < n$.

These blank bytes are stripped away from the codeword before to be send by the transceiver of before entering to the interleaver block. For the decoding, these blank bytes are inserted into the codeword for a proper decoding and, once again, stripped away after finishing the process to reveal the "original" information sequence, that is if the quantity of errors don't pass the value of t [18].

V. INTERLEAVER

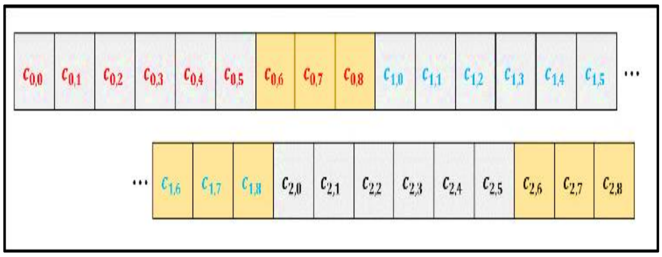
It is well stablished that the maximum number of errors/erasures that RS is able to detect is 2t and the number of errors/erasures being able to correct are t. So, in a data array, concatenating several codewords lined-up, we can pass it through a Bit Error Channel (BEC) with a salt-pepper noise model and perhaps the number of erroneous symbols doesn't pass the value of t.

However, in practice, the noise usually come as a burst of bit errors, so instead the data array passes through a Burst Bit Error Channel (BBEC), and the probability to the number of errors surpass the value of t increases drastically.

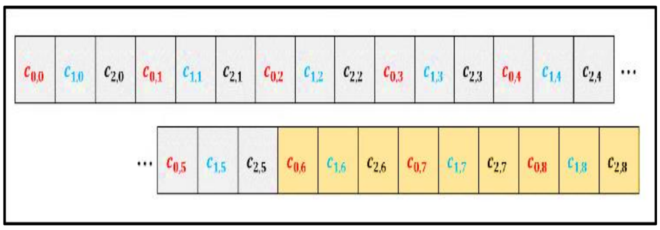
One way to optimize the capability to correct errors is to "interleave" them along the data array. By doing so, instead of having for example 10 errors/erasures in a single codeword, we get 1 error/erasures per 10 codewords. To implement the interleaved to RS, we create a matrix $C_{i \times n}$, where i is called the interleaved depth and *n* the length of the codeword (do not confuse with the "i" described in the subsection 2.1.). Every term of $c(x)$ (Eq. 11) is an element of the matrix.

$$C_{i\times n} = \begin{bmatrix} \mathrm{M}_{k-1}x_0^{n-1} & \cdots & M_0 x_0^{n-k} & \mathrm{r}_{n-k-1}x_0^{n-k-1} & \cdots & r_0 \\ \mathrm{M}_{k-1}x_1^{n-1} & \cdots & M_0 x_1^{n-k} & \mathrm{r}_{n-k-1}x_1^{n-k-1} & \cdots & r_0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{M}_{k-1}x_{i-1}^{n-1} & \cdots & M_0 x_{i-1}^{n-k} & \mathrm{r}_{n-k-1}x_{i-1}^{n-k-1} & \cdots & r_0 \end{bmatrix} \quad (20)$$

Then we take the transpose of $C_{i\times n}$ and convert to 1-D byte array.



a) linear concatenation



b) Interleaved with i = 3

Fig. 2. Graphic example of 3 codewords of data rearranged as 1-D array

VI. THE ERROR CHANNEL

The data is passed through a random BBEC, with 2 types of noise: salt-pepper noise (SPN) and random-valued impulse noise (RVIN); all in software using Python. This kind of noises are introduced in the digital data through transmission and acquisition [16]. A significant characteristic both noises is that only fractions of the bytes are degraded.

In the case of SPN, every byte corrupted acquire the minimum or the maximum possible value for a byte, that is 0 or 255. On the other hand, as its name implies, the bytes corrupted by an RVIN event acquire a random value between 0 to 255.The error position along the byte array follows a binomial distribution model (Eq. 22), included by the Numpy module [17].

$$P(N) = \binom{n}{N} p^N (1-p)^{n-N} \quad (21)$$

Since the goal is to use the siRS on compressed image data, like the JPEG format, sowe consider it is not necessary to run the tests with another error distributions, that is because when we deal with compressed data, a single erroneous byte can corrupt the entire data [2].

VII. EXPERIMENTAL TESTS

A. Error channel simulations

In order to estimate the performance of the ECC quantitative, we perform a series of error-adding simulations to a test data, where the expected amount of noise goes from low to high. The test data is an unformatted grayscale (value = 127) image (Fig. 3), with a size of 320x240, encoded using both the siRS(126, 18, i=3), and a regular RS(255, 223). In every test, it is calculated the Bit-Error Rate (BER) and the Signal-to-Noise Ratio (SNR).

$$BER = \frac{number\ of\ erroneous\ bits}{Total\ bits} \quad (22)$$

It is important to point out that both, the siRS and the regular RS, can be used to intent to protect from random errors to any data in any format, however, the use of a grayscale image as target is for visualization purposes as well as to metrics the SNR in an already known consistent digital signal.

Fig. 3. To the left, unformatted image with a value of 127 in all the pixels. To the right the test image for the RF transmission test using the siRS algorithm

Retrieval Number:100.1/ijeat.B39221212222
DOI: 10.35940/ijeat.B3922.1212222
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

B. RF transmission using cots

For the point-to-point transmission test from the ARM-based mini-PC to a ground station computer, the data format to use is an image compressed in a JPEG file; an image of the Large Millimeter Telescope "Alfonso Serrano" located at the Volcán Sierra Negra (Fig. 3).

Once again, the data is going to be encoded using both the siRS(126, 18, i=3), and a RS(255, 223) and passed through the BBEC for both SPN and RVIN noises. This test focuses on the algorithm under real-world conditions plus the errors added by the BBEC, so it is only going to be shown the resulting JPEG images received and decoded. It is expected more errors caused by the RF noise, such noise between the transceivers is not measured. The figure 4 and 5 show the program of the sender, which it is executed by the RAM-bases mini-PC and receiver in the groundstation computer.

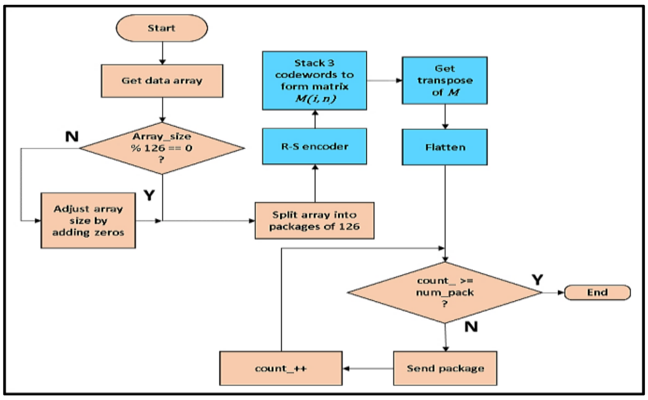


Fig. 4. Flowchart of the forward sender function. The blue blocks are the siRS section

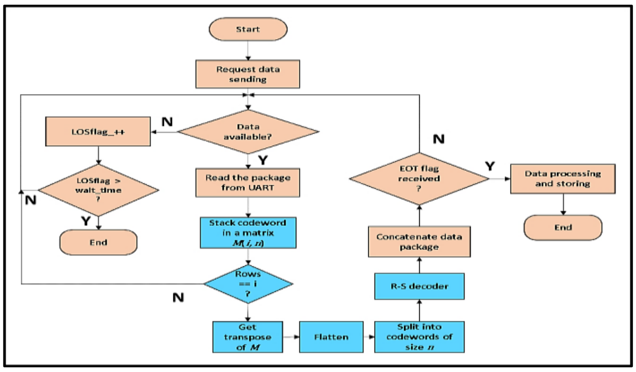
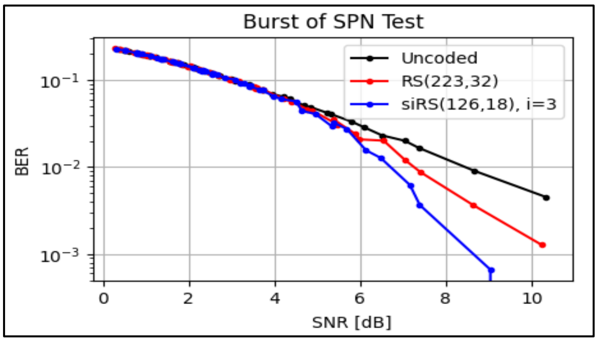


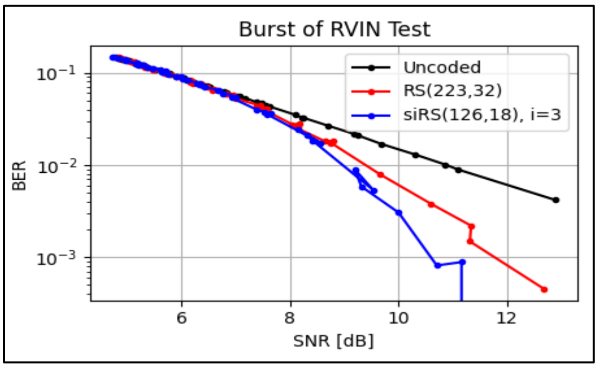
Fig. 5. Flowchart of the forward receiver function. The blue blocks are the siRS section

VIII. RESULTS

The siRS algorithm has been able to correct the errors added by the BBEC with the SPN as well RVIN noise model. As shown in the table2, we can see the grayscale image before decoding and after decoding. In the table 3 it is shown a series of data transmissions of the JPEG image. As we can see, not all the images could be recovered at 100% but there are not entirely lost. In the Figure 6, it is shown a graphics of the performance of the siRS(126,18, i=3) and the RS(255, 223) during a series of test where the SNR goes from low to high. What those graphics are showing is that conforming the SNR increases (there is more noise than actual signal), so the erroneous bits, the uncoded data (marked as the black dots and line in the graphics) shows exactly this behavior. After the decoding, the BER decreases at the same level of SNR, meaning that part of the erroneous bytes encountered have been corrected.



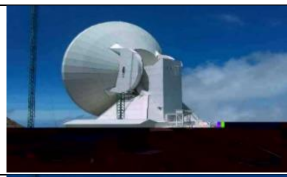
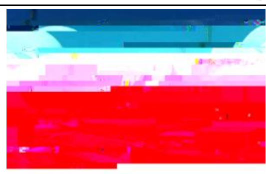
a) series of tests with SPN



b) series of tests with RVIN

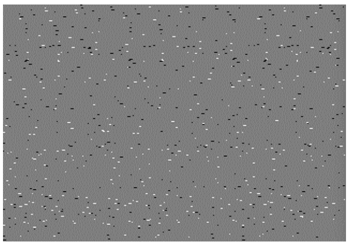
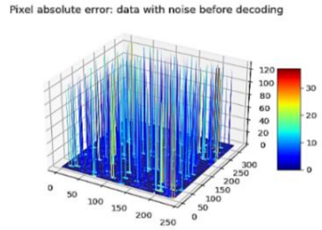
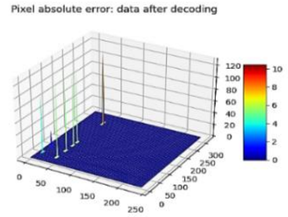
Fig. 6. Performance of the siRS and RS. We observe that the siRS have a lower BER at the same level of SNR than the regular RS.

Table 2. Series of JPEG images received from the RF link

Before	After
	
	
	
	

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table 3. Test data before and after the siRS decoding along with plots showing the absolute error

	SNR	Test data	Absolute error graphics
Before	9.017 dB		
After	18.637 dB		

IX. CONCLUSIONS

We have been able to develop an algorithm of shortened interleaved Reed-Solomon code that can be used in ARM-based OBC as well as ported to Windows OS. We testes the performance by comparing the BER vs SNR with a regular RS (255, 223) that have being previously tested in a previous balloon probe flight. Also, it was tested with JPEG images in RF transmissions using entirely low-cost COTS transceivers and determined that it is capable to find and correct errors in data with format. To improve the performance of the siRS codec, it is required further analysis of the time execution. Another future test involves the use of the siRS algorithm to protect files stored in the internal memory of the ARM-based mini-PCs, since one of the main problems associated to the use of COTS is the limited radiation tolerance [1].

REFERENCES

1. Lovascio, A., D'Orazio, A. and Centonze, V., 2020. Characterization of a COTS-Based RF Receiver for Cubesat Applications. Sensors, 20(3), p.776. [CrossRef]
2. Valadez-Campos, E., Mendoza-Torres, E. and DeRoa-Campoy, A., 2021. Low-cost testing platform for high-altitude balloon flights. Educative Clues (Pistas Educativas), 42(136), p.21. Available at: <http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/view/2425/0> [Accessed 28 October 2022].
3. Valadez-Campos, E., Mendoza-Torres, J.E., 2021. Implementation of Reed-Solomon Codes for Stratospheric Balloon Probes and Nano-Satellites. International Journal of Engineering Research and Technology, 10(2), p.512.
4. Sklar, B. and Harris, F., 2004. The ABCs of linear block codes. IEEE Signal Processing Magazine, 21(4), pp.14-35. [CrossRef]
5. Phat Nguyen Huu and Vinh Tran-Quang†, Takumi Miyoshi., 2012. Multi-hop Reed-Solomon encoding scheme for image transmission on wireless sensor networks. Fourth International Conference on Communications and Electronics (ICCE). [CrossRef]
6. GOMX-3 (GomSpace Express-3). eoPortal. (2022). Retrieved 28 October 2022, from https://www.eoportal.org/satellite-missions/gomx-3#rf-communications.
7. NanoCom AX100 Transceiver - Transceiver | SatCatalog. Satcatalog.com. (2022). Retrieved 12 September 2022, from https://www.satcatalog.com/component/nanocom-ax100-transceiver/
8. Jeon, S., Kumar, B., Hwang, E., & Cheng, M. (2010). Evaluation of Error-Correcting Codes for Radiation-Tolerant Memory. Interplanetary Network Progress Report, 42-181, pp 1-18. Retrieved 5 September 2022, from https://ipnpr.jpl.nasa.gov/progress_report/42-181/181A.pdf.
9. N. Ambramson., Information Theory and Coding, 1963, McGraw-Hill, USA.
10. E.R. Berlekamp Algebraic Coding Theory, 1968, McGraw-Hill, New York (USA).
11. Hamming, R., 1986. Coding and information theory. Englewood Cliffs, N.J.: Prentice-Hall.
12. R.T. Moenck. Practical fast polynomial multiplication. In Proc. 3rd ACM Symposium on Symbolic and algebraic computation, pages 136–148. ACM, 1976. [CrossRef]
13. Blahut R. E., 2003, Algebraic Codes for Data Transmission, Cambridge University Press. [CrossRef]
14. Peterson, W., 1960. Encoding and error-correction procedures for the Bose-Chaudhuri codes. IEEE Transactions on Information Theory, 6(4), pp.459-470. [CrossRef]
15. Chien, R., 1964. Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes. *IEEE Transactions on Information Theory*, 10(4), pp.357-363. [CrossRef]
16. Banerjee, S., Bandyopadhyay, A., Mukherjee, A., Das, A., & Bag, R. (2017). Random Valued Impulse Noise Removal Using Region Based Detection Approach. Engineering, Technology &Amp; Applied Science Research, 7(6), 2288–2292. https://doi.org/10.48084/etasr.1609 [CrossRef]
17. numpy.random.binomial — NumPy v1.23 Manual. (n.d.). Retrieved November 1, 2022, from https://numpy.org/doc/stable/reference/random/generated/numpy.random.binomial.html
18. McEliece, R.J., & Swanson, L. (1994). Reed-Solomon Codes and the Exploration of the Solar System. From https://trs.jpl.nasa.gov/bitstream/handle/2014/34531/94-0881.pdf?sequence=1.
19. Urban, M. (1987). Voyager Image Data Compression and Block Encoding. International Telemetering Conference Proceedings, (23), 137-162. Retrieved 6 September 2022, from http://hdl.handle.net/10150/615317.
20. National Aeronautics and Space Administration. (2020). TM synchronization and channel coding — summary of concept and rationale (1st ed., pp. ch 7, p11). Retrieved 6 September 2022, from https://public.ccsds.org/Pubs/130x1g3.pdf.
21. Ryan, W.E., Lin, S. (2009). Channel Codes: Classical and Modern (Illustrated). Cambridge University Press. pp. 38-68. [CrossRef]
22. Reed, I. S., & Solomon, G. (1960). Polynomial Codes Over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics, 8(2), 300–304. https://doi.org/10.1137/0108018. [CrossRef]
23. William A. Geisel. (1990). Tutorial On Reed-Solomon Error Correcting Coding [Dataset]. National Aeronautics and Space Administration (NASA). https://ntrs.nasa.gov/citations/19900019023.
24. C.K.P Clarke. (2002). Reed-Solomon Error Correction. In BBC Research & Development (WHP-031). British Broadcasting Corporation. https://downloads.bbc.co.uk/rd/pubs/whp/whp-pdf-files/WHP031.pdf

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

AUTHORS PROFILE

Eduardo Valadez-Campos Currently doing the Masters of Space Science and Technology at the Instituto Nacional de Astrofisica, Óptica y ElectrÓnica (INAOE). He obtained his bachelor degree in Electronics with the thesis titled: "Simulation of the movement of a ground vehicle for programmed paths applying fuzzy control", Benemérita Universidad Autónoma de Puebla (BUAP). Since 2014, he is currently working with Dr. Eduardo Mendoza in the development of astronomical instrumentation and aerospace technology with several applications at the INAOE. He has worked in the development of software for instrumentation and control, the prototype of miniaturized telescope in the NIR and MIR wavelengths as a payload in Cubesats, the prototype of stratospheric probes for component testing and space weather studies and the prototype of telecommunication a tracking subsystem using COTS.

J. Eduardo Mendoza-Torres Researcher in the department of Astrophysics at the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). Obtain his Ph.D. in Astrophysics and Radio-Astronomy with the tesis titled "Study of solar flares in microwavelengths with observations carried out at RATAN-600 radiotelescope, Large Radiotelescope at Pulkovo and Small Baseline Radio Interferometer", Special Astrophysical Observatory of the Academy of Sciences, Russia. Among other things, he is in charge of several research projects regarding the development of payloads for mini-satellites and cubesats, as well as the miniaturization of astronomical instrumentation.

Retrieval Number:100.1/ijeat.B39221212222
DOI: 10.35940/ijeat.B3922.1212222
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)