Linear Prediction and Rice Codes Based Two-Stage Method for Lossless Telemetry Data Compression

Mohamed A. Elshafey
Computer Engineering Department
Military Technical College
Cairo, Egypt
E-mail: m.shafey@mtc.edu.eg

***Abstract* – The article describes the two-stage lossless compression method for telemetry data, in which the linear prediction method acts as a decorrelator in the first stage, and Rice codes act as an entropy coder in the second stage. A detailed description of implementing linear prediction is presented and applied in experiments on different telemetry frame structures, in the Inter-Range Instrumentation Group (IRIG) standard 106 format. Telemetry data analysis is performed, and compression efficiency is presented based on estimates of the gain in variance and entropy of the output signal from the decorrelator. Based on the experimental results, conclusions are extracted, and recommendations are suggested.**

Keywords—telemetry date frame; Rice coding; Linear Prediction; IRIG-106; lossless compression.

I. INTRODUCTION

Remote sensed data are widely transmitted in the form of telemetry stream to ground stations for analyzing abnormal situations, recovering bad sections in the telemetry stream and other types of post-processing or real-time data processing. Telemetry data has a huge size. So, it should be compressed at the telemetry source before transmission and/or archived in a compressed form [1]-[6].

This paper presents results of studies conducted with the aim of developing an effective two-stage lossless compression method based on linear prediction and Rice codes for different structures of telemetry streams.

The Rice coding algorithm is widely used for lossless telemetry data compression, but its effectiveness is largely determined by the properties of the encoded information. Therefore, it is not suitable in case of highly deviated data [4] [7].

In telemetry systems, the commutator of the data acquisition subsystem measures values from multiple different sensors and outputs a single stream of pulses, which are then modulated and transmitted to a receiving station, where the decommutator returns the serial digital stream back to a parallel form. One complete acquisition cycle of the commutator generates data words, which represent values of each measured parameter. These data words and a synchronization code, which is needed for the decommutation process, establish the telemetry frame.

In a simple commutator, each data word is sampled once per rotation at a rate compatible with the fastest changing measured parameter. In many cases, rate of change of the measured parameters is different, often by several orders of magnitude. Consequently, it is not preferred to perform sampling for the slow changing parameters per each rotation. These parameters can be sampled in a single data word (i.e., a subframe commutation case), e.g., for three slow changing parameters, the main commutator will take three rotations to sample every parameter at least once. Each rotation of the main commutator produces a minor frame, while these three minor frames together form a major telemetry frame. In each minor frame, an additional synchronization code (usually a counter) is added, so the decommutator can distinguish between subframe commutated parameters [1] [9].

In this paper, the process of implementing linear predictions as a first stage before Rice codes is presented with illustrative figures, implemented, tested and verified with real telemetry data in IRIG-106 standard format. This is a comprehensive and open telemetry standard, developed and maintained by the Telemetry Group of the Range Commanders Council (RCC) [8].

The statistical characteristics of the telemetry tested data are examined, and their effects on the prediction efficiency are investigated, and sequentially on the overall compression efficiency. The effect of telemetry frame structure on the selection of the predictor's order and, even, on the suggested number of predictors is determined.

In this paper, a literature survey on recent telemetry compression methods is presented in Section 2, while in Section 3, a survey is provided on successive processes, which are involved in applying linear prediction on telemetry data for lossless compressing. In Section 4, telemetry data preparation and the tested frame structures are described. Experimental tests and analysis for the effects of telemetry data characteristics and frame structure on compression efficiency are demonstrated in Section 5. Finally, in Section 6, conclusions and future works are suggested.

II. LITERATURE SURVEY

The Rice codes method is considered one of the most powerful entropy coders, which is used for telemetry data compression, e.g., it is recommended by The Consultative Committee for Space Data Systems (CCSDS) to compress telemetry data [4].

Many recent researches are involved in the preprocessing layer before entropy coding in order to improve compression efficiency [10]. In [11], a method based on displaying the original data onto geometric surfaces planes is suggested as a preprocessor layer. Another method based on performing XOR operation on subsequent telemetry frames is suggested for preprocessing data before entropy coding [12] [13]. A Neural Network (NN)-based telemetry classification method is suggested in [14].

 ISBN: 978-1-61208-694-1

Prediction based preprocessor layer are widely suggested and used for telemetry compression. Prediction methods based on NN models are illustrated in [15] [16]. An adaptive prediction layer based on the Normalized Least Mean Square (NLMS) algorithm is proposed in [17]. This method depends on the value of the convergence parameter and the smoothing parameter of the NLMS algorithm. It shows results very close and slightly better than those of the non-adaptive prediction method, however, it may be worse for more complex telemetry frame structures (i.e., telemetry frames which contain sub-commutated parameters). As the total length of the major telemetry frame increases, the suggested prediction order of the NLMS based filter should be increased to match the total frame length for a better compression.

III. LINEAR PREDICTION AND RICE CODES BASED TWO-STAGE COMPRESSION METHOD

The effectiveness of any compression algorithm is affected by the statistical characteristics of the data to be compressed and generally, if the compression algorithm can adapt to these characterises, it will generate better results. So, recent lossless compression algorithms are generally described by a series of two stages as shown in Figure 1:

1) Decorrelation stage, which figures out the correlation in the original data to be encoded.

2) Entropy coding stage, in which entropy coding will be performed on decorrelated data, which in general has lower variance and lower entropy value than of the original ones [18]-[24].

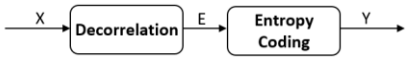

Figure 1. Two stage lossless compression scheme.

A. Rice coding algorithm overview

The efficiency of Rice codes depends on a controllable parameter *k*, which depends on the variance of symbols in the message.

A code word for an encoded number (*v*) by Rice coding algorithm is divided into two parts, i.e., $v=v_i+v_f$, where $v_i = \lfloor v/2^k \rfloor$ and $v_f = v \bmod 2^k$. The coded word is formed by the value of (v_i +1), provided by unary code, and the value of v_f, represented by (*k*) bit binary code.

In this paper, the following formula [23] [25] is used in experiments:

$$k = \log_2(\log_e(2) \times E(|x|)) , \quad (1)$$

where E(|x|) is the average value of symbols of the encoded message.

Rice codes are effective for coding values, which are close to 2^k, and are recommended for the encoding data, whose distribution matches the geometric one.

B. Linear prediction method as a decorrelator

This paper presents a detailed description of applying linear prediction as a decorrelator for reducing correlational dependences in a telemetry stream.

In linear prediction, the value of the sample x_i is predicted by the known values of the previous samples $x_{i-1}, x_{i-2}, \ldots, x_{i-p}$ by the formula:

$$\hat{x}_i = Q\left(\sum_{j=1}^{p} a_j x_{i-j} \right), \quad (2)$$

where $\hat{x}_i$ is the predicted value of sample x_i, Q is a non-linear operator, which denotes level quantization, a_j is the prediction coefficients and p is the linear prediction order. At each step, the prediction error e_i is calculated by:

$$e_i = x_i - \hat{x}_i . \quad (3)$$

To perform a two-stage lossless compression of telemetry data, the following scheme presented in Figure 2 is implemented, where $\hat{a}_i$ represent the quantized values of prediction coefficients a_i.

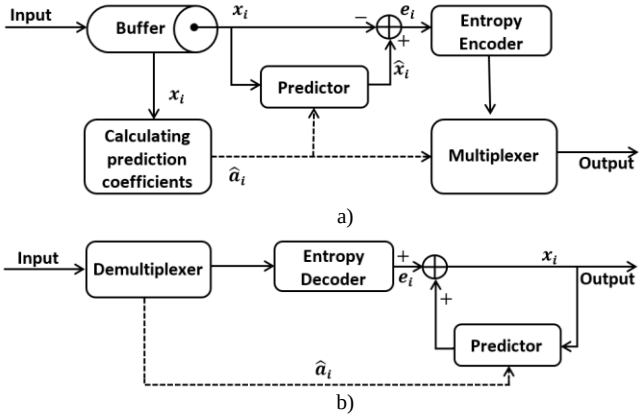

Figure 2. The scheme of two-stage lossless data compression (a) and reconstruction of the original data from the compressed ones (b) [26].

The prediction coefficients are required at the decoder to reconstruct the original signal samples x_i from the prediction error values e_i. So, they are added to the output of the decorrelator.

C. Calculation of predictor coefficient(s)

The optimal values of linear prediction coefficients are obtained by minimizing the Mean Square Error (MSE) of the estimate:

$$e(n) = x(n) - \hat{x}(n) \rightarrow \min (E (e^2(n))) , \quad (4)$$

and based on the principle of orthogonality, thus:

$$E\left[x(n-L) \times e(n) \right] = E\left[x(n-L) \times (x(n) - \hat{x}(n)) \right] = 0 , \quad (5)$$

which can equivalently be written as:

$$\sum_{t=1}^{p} \alpha_t R_{xx}[L-t] = -R_{xx}[L] , \quad (6)$$

where $L=1..p$, $R_{xx}[L]$ is the autocorrelation of x(n), α_t is a vector of length p, $R_{xx}[L-t]$ is a matrix of size $p \times p$ and $R_{xx}[L]$ is a vector of length p. These t equations are Toeplitz

and known as the Yule-Walker equations. A quick and efficient way to solve this matrix is the Levinson-Durbin algorithm, which uses the symmetry of the R_{xx}[L-t] matrix to simplify the calculations from $\{O^3\}$ to $\{O^2\}$ [27].

D. Coding of linear predictor coefficients

Direct quantization of the obtained prediction coefficients has several disadvantages. This operation leads to significant distortions in the frequency domain. Instead, their reflections, using a nonlinear procedure, will be quantized.

Since the values of the prediction coefficients are in the range [–1, + 1], a nonlinear procedure y = arcsin(x), shown in Figure 3, has been implemented [26].

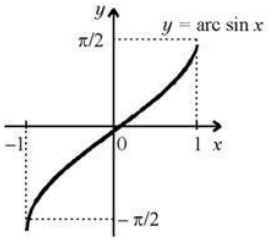

Figure 3. Function diagram of Y=arcsin(X).

The values $\hat{a}_i = \arcsin(a_i)$ are in the range $[-\pi/2, +\pi/2]$. Each value of $\hat{a}_i$ is converted to an 8-bit integer (one bit to represent the sign, and seven bits to represent the output of the arcsin function, which is in the range from 0 to 90). The successive functions, in Figure 4, illustrate the coding process for prediction coefficients for lossless compression.

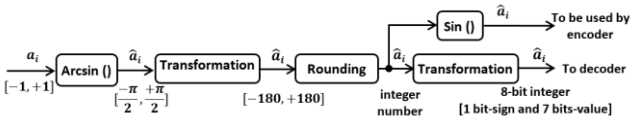

Figure 4. Coding scheme of prediction coefficient using arcsin function.

E. Prediction error transformation

Prediction errors are determined by using (3), which can take both positive and negative values. Therefore, a conversion process is required, which preserves the original range of signal values. The range of error values is equal to the range of values of the input signal to the converter. The error e_i is recalculated into a value δ_i according to the formula:

$$\delta_i = \begin{cases} 2e_i, & 0 \le e_i \le \theta_i, \\ 2|e_i| - 1, & -\theta_i \le e_i < 0, \\ \theta_i + |e_i|, & \text{otherwise}, \end{cases} \quad (7)$$

where $\theta_i = \min(\hat{x}_i - x_{min}, x_{max} - \hat{x}_i)$, x_{min} is the minimum value of the input signal, x_{max} is the maximum value of the input signal. The prediction error is in the range $x_{min} - \hat{x}_i \le e_i \le x_{max} - \hat{x}_i$. The number of bits required to represent errors e_i is the same to represent the signal x_i. If the input signal is represented by positive 8-bit numbers, then $x_{min} = 0$, $x_{max} = 2^8 - 1$.

The inverse transformation is performed according to the following formula:

$$\text{if } \delta_i \le 2\theta_i, \text{ then}$$
$$e_i = \begin{cases} \delta_i/2, & \text{if } \delta_i \text{ even}, \\ -(\delta_i + 1)/2, & \text{if } \delta_i \text{ odd}; \end{cases} \quad (8)$$
$$\text{if } \delta_i > 2\theta_i, \text{ then}$$
$$e_i = \begin{cases} \delta_i - \theta_i, \text{ if } \theta_i = \hat{x}_i - x_{min}, \\ \theta_i - \delta_i, \text{ if } \theta_i = x_{max} - \hat{x}_i. \end{cases}$$

To convert a prediction error to a positive integer value, an error transform block is applied at the output of the decorrelator [28], as shown in Figure 5-a and Figure 5-b at the compressor side and at the decompressor side, respectively.

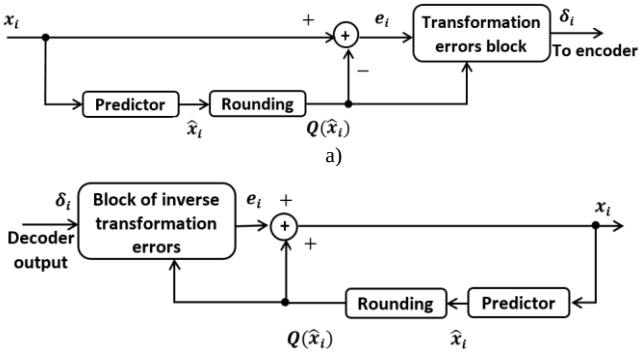

Figure 5. Transformation scheme (a) and inverse transformation scheme (b) of prediction errors.

IV. Preparing Data For Experiments

The telemetry data of automatic control systems have been used in experiments. They are presented in time series of digitized samples of analog sensors, which represent typical telemetry system parameters including temperature, pressure, position data, and so on. These telemetry parameters were obtained in laboratory conditions from real sources. A timed portion of the data samples acquired from nine physical real sensors is shown in Figure 6.

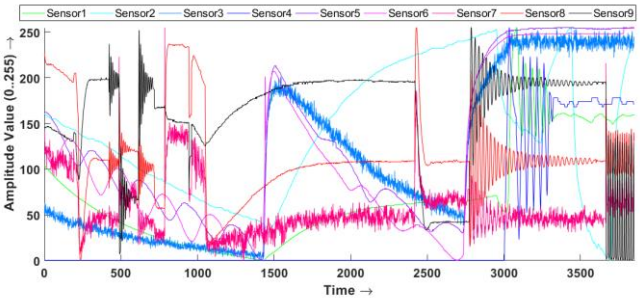

Figure 6. A timed portion of the acquired data samples.

The telemetry data samples used in our experiments are transmitted in frames with a fixed length and a

predetermined fixed internal structure in IRIG-106 standard. In frames, different readings of several sources, i.e., digitized analog sensor readings and readings of digital devices, can be transmitted. Each data source is transmitted in a separate channel of telemetry data recording system in an 8-bit data word in the telemetry frame. The telemetry stream is generated by the telemetry simulation software presented in [29]. The simulator is used to generate different telemetry data streams, based on different telemetry frame structures, in IRIG-106 standard.

Experiments are carried out on two different telemetry frame structures: tf1 and tf2, as shown in Figure 7 and Figure 8, respectively. Telemetry frame structure tf1 consists of 9 channels (DataWords) in one major frame without any subframes, i.e., one level of commutation. An additional piece of information (SYNC_F) of 16 bit (according to IRIG-106 standard) to provide frame synchronization in the telemetry stream, is added.

SYNC_F	DataWord 1	DataWord 2	DataWord 3	DataWord 4	DataWord 5	DataWord 6	DataWord 7	DataWord 8	DataWord 9
SYNC16	Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8	Sensor9

Figure 7. Telemetry frame tf1 without subframes.

The second structure tf2 in Figure 9 is a telemetry frame structure, which contains a subframe of length 4, resulting from a second level of commutation attached to the 4th DataWord. A counter as an additional subframe synchronization (SYNC_SF1) is added to each minor frame.

SYNC_F	SYNC_SF1	DataWord 1	DataWord 2	DataWord 3	DataWord 4	DataWord 5	DataWord 6
SYNC16	C1	Sensor1	Sensor2	Sensor3	Sensor5	Sensor4	Sensor7
SYNC16	C2	Sensor1	Sensor2	Sensor3	Sensor6	Sensor4	Sensor7
SYNC16	C3	Sensor1	Sensor2	Sensor3	Sensor8	Sensor4	Sensor7
SYNC16	C4	Sensor1	Sensor2	Sensor3	Sensor9	Sensor4	Sensor7

Figure 8. Telemetry frame tf2 with a subframe at the 4th DataWord.

V. EXPERIMENTAL TESTS AND ANALYSIS

The efficiency of the decorrelation stage can be evaluated by two parameters:

- Prediction gain in variance Prediction_Gain, which is the relation between the variance σ_X^2 of the original data (X) and the variance σ_E^2 of the prediction error (E):

$$\text{Prediction\_Gain} = \sigma_X^2 / \sigma_E^2 . \quad (9)$$

- The entropy H (in bits / symbol):

$$H = -\sum_{i=1}^{Z} P_i \times \log_2(P_i) , \quad (10)$$

where Z is the number of symbols in encoded data and P_i is the probability of each symbol i.

For evaluating the efficiency of the compression algorithm, the compression ratio (R) is checked:

$$R = S_{input} / S_{output} , \quad (11)$$

where S_{input} is the size of the original data (X), and S_{output} is the size of compressed data at the system output (output of entropy coder).

Probability distribution of telemetry data samples is shown in Figure 9. According to this histogram, the values are distributed throughout the scale. The histogram has several pronounced peaks typical of telemetry information, in which the monitored parameters for a long time retain values close to constant, and deviate slightly from them under the influence of noise. It has entropy value $H_x = 7.45$ (bits/symbol) and variance $\sigma_X^2 = 6085.77$.

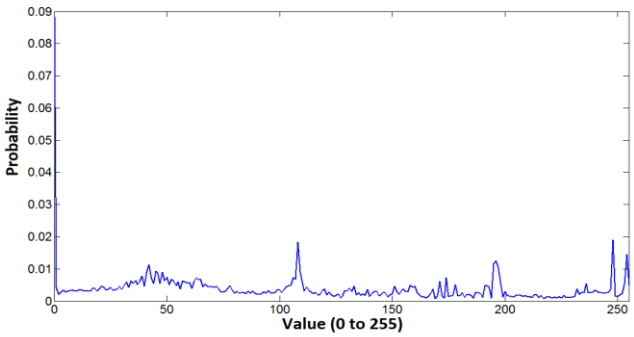

Figure 9. Probability distribution of data samples in telemetry stream.

Applying Rice codes only to the original data (X) leads to a compression ratio R = 1.03 at parameter *k* = 7, which means nearly no compression.

The linear prediction method is applied on the original data (X), in the first telemetry frame structure tf1. Experimental dependencies of linear prediction efficiency, using (9) and (10), on different prediction orders are shown in Table I.

TABLE I. LINEAR PREDICTION EFFICIENCY AT DIFFERENT PREDICTION ORDERS APPLIED ON TELEMETRY FRAME TF1

Prediction Efficiency	Predictor				
	Order 1	*Order 2*	*Order 8*	*Order 9*	*Order 18*
σ_E^2	5599.01	5565.08	4394.12	78.32	82.42
Prediction_Gain	1.08	1.09	1.38	77.70	73.83
H_E	7.86	7.78	7.71	3.06	3.79

The best results in Table I, i.e., the highest value of prediction gain and the lowest of entropy, are obtained using the 9th order of linear prediction. The probability distribution of the prediction errors of the 9th order predictor is shown in Figure 10, in which the graph has an expressed peak at zero.

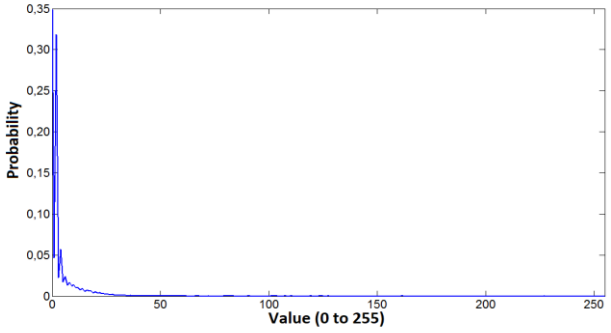

Figure 10. Probability distribution of prediction errors of 9th order predictor applied on telemetry frame tf1.

Applying Rice codes as a second stage, after a 9th order linear predictor at a first stage, leads to a compression ratio value R = 2.13 at parameter k = 3.

The dependences of the compression ratio, (resulting from linear prediction based decorrelation and Rice codes entropy coding), on the order of the linear predictor are shown in Figure 11. We notice that the compression ratio increases dramatically when the order of the linear predictor becomes equal to the count of data words in the repeating structure of the information flow. In this case, this count is equal to the frame length of the main commutator (i.e., order = 9).

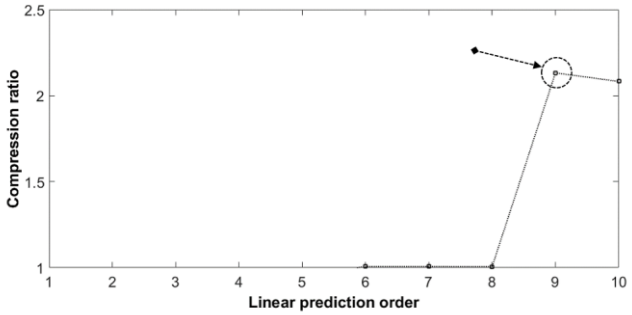

Figure 11. Experimentally obtained dependences of the compression ratio on the order of the predictior for telemetry frame tf1.

Experiments are carried out on the second telemetry frame structure tf2. Based on results obtained, shown in Figure 12, the compression ratio increases explicitly when the predictor order becomes equal to the count of words contained in the four frames of the main commutator (i.e., Best compression ratio R = 1.65 with a linear prediction of order 24).

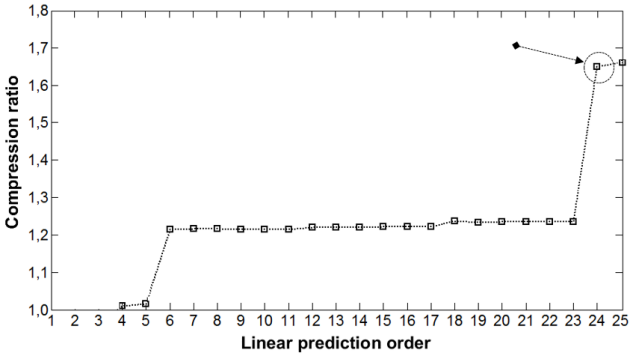

Figure 12. Experimentally obtained dependences of the compression ratio on the order of the predictior for telemetry frame tf2.

From the obtained results, it can be seen that, between readings of the same parameter, located in adjacent frames, there is a strong correlational dependency, and there are usually no such dependencies between adjacent words in the telemetry stream. On the other hand, increasing the order of the predictor to identify these dependencies leads to more complex calculations. In general, the use of the linear prediction method for data decorrelation in a telemetry stream is justified in the case of a small frame length and a simple one level of commutation. If these conditions are not met, then this method of decorrelation is not very efficient or requires significant computational resources.

From the above and based on the fact that the output frames from the main commutator have a fixed length and a constant predetermined internal structure, a linear prediction based decorrelation method is presented. In this method, a decorrelation stage is applied for each telemetry parameter channel, as described in Figure 13. For a given telemetry frame structure in Figure 13-a, a linear prediction is applied on each parameter channel, as shown in Figure 13-b.

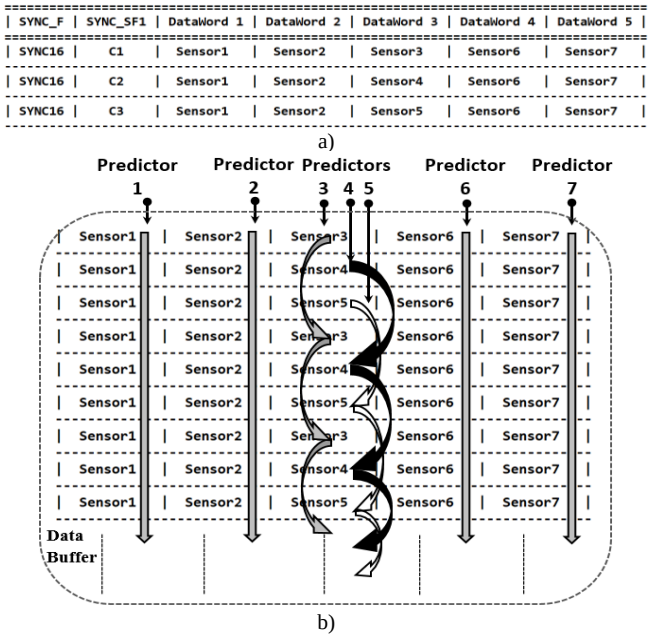

Figure 13. Description of linear prediction of each parameter's channel.

Additional experiments, based on the proposed method, are applied on telemetry frame tf2. It should be noted that the use of first order predictors to predict values of each channel is quite effective.

In experiments, applying 9 linear predictors, each of order 1, for each parameter channel in telemetry frame tf2 leads to a better prediction efficiency than the other obtained from applying a single predictor, as shown in Table II.

TABLE II. LINEAR PREDICTION EFFICIENCY OF DIFFERENT PREDICTORS APPLIED ON TELEMETRY FRAME TF2

Prediction Efficiency	Single linear predictor of order 24 for the stream	First order linear predictor for each parameter channel
Prediction_Gain	28.31	40.84
H_E	4.57	3.34

Applying single first order predictor for each channel, in a two stage Rice based compression method for telemetry frame tf2 achieves compression ratio R=1.96.

 ISBN: 978-1-61208-694-1

VI. Conclusion And Future Work

Applying linear prediction method as a decorrelator is sufficient enough to transform the probability distribution model of the encoded data samples into a model which is more efficient when implementing entropy coding.

For telemetry major frames without subframes, a single linear predictor of order equal to the length of the major frame (i.e., total number of data words) is an efficient decorrelator, while for telemetry major frames with subframes, applying first order linear predictor for each telemetry parameter channel in the major frame is more effective.

Applying linear prediction for each channel transfers the problem of decorrelation into a parallel one, which can be implemented on Graphics Processing Units (GPUs) for high performance parallel computations.

A preprocessor stage of noise removal from measured telemetry signals can lead to a better efficiency in the subsequent stage (i.e., prediction based decorrelation stage).

Recent deep learning algorithms based on Convolutional Neural Networks (CNN) are promising to provide better and worthy prediction efficiency. As for applying CNNs on onboard systems, a recent method of deep compression should be visited [30].

References

[1] F. Carden, R. Jedlicka, and R. Henry, Telemetry Systems Engineering, MA, Norwood: Artech House, 2002.

[2] J. P. Arcangeli, M. Crochemore, J. N. Hourcastagnou, and J. E. Pin, “Compression for an Effective Management of Telemetry Data,” NASA Technical Documents: nasa_techdoc_19940019483, pp. 823-830. 1993.

[3] J. A. Martínez-Heras, D. Evans, and R. Timm, “Housekeeping Telemetry Compression: When, how and why bother,” First International Conference on Advances in Satellite and Space Communications, 2009, pp. 35-40, doi:10.1109/SPACOMM.2009.30.

[4] Lossless Data Compression. Recommendation for Space Data System Standard, Recommended Standard CCSDS 121.0-b-2, Blue Book, May 2012. Available online: https:// public.ccsds.org / publications / BlueBooks.aspx [Retrieved: Feb., 2019].

[5] P. Staudinger et al., “Lossless Compression for Archiving Satellite Telemetry Data,” IEEE Aerospace Conference Proceedings, 2000, vol. 2, pp. 299-304.

[6] D. A. Maluf, P. B. Tran, and D. Tran, “Effective Data Representation and Compression in Ground Data Systems,” IEEE Aerospace Conference, 2008. pp. 1-7.

[7] R. F. Rice, Some Practical Universal Noiseless Coding Techniques - Parts I-III, JPL Tech. Reps. JPL-79-22 (1979), JPL-83-17 (1983), and JPL-91-3 (1991).

[8] Telemetry Standards, IRIG Standard 106-13. Secretariat Range Commanders Council US Army White Sand Missile Range. New Mexico (USA), 2013. Available on-line: http: // www.irig106.org / docs / 106-13 / [Retrieved: Feb., 2019].

[9] S. Horan, Introduction to PCM Telemetering Systems. 2nd Edition. USA: CRC Press, 2002.

[10] A. V. Levenets, “The Basic Principles and Methods of the System Approach to Compression of Telemetry Data,” Journal of Physics: Conference Series, vol. 944, no. 1, IOP Publishing, 2018.

[11] A. V. Levenets, I. V. Bogachev, and E. U. Chye, “Telemetry Data Compression Algorithms Based on Operation of Displaying onto Geometric Surfaces,” In Control and Communications (SIBCON), 2017 International Siberian Conference on, pp. 1-6, IEEE, 2017.

[12] M. A. Elshafey, “A Two-Stage Rice-Based Lossless Compression Method for Telemetry Data,” Journal of Engineering Science and Military Technologies, vol. 2, no. 3, pp. 147-152, doi:10.21608/ejmtc.2018.20314.

[13] D. J. Evans and A. Donati, “The ESA POCKET+ Housekeeping Telemetry Compression Algorithm: Why Make Spacecraft Operations Harder than it already is?” In 2018 SpaceOps Conference, 2018.

[14] X. Shi, Y. Shen, Y. Wang, and L. Bai, “Differential-Clustering Compression Algorithm for Real-Time Aerospace Telemetry Data,” IEEE Access, vol. 6, pp. 57425-57433, 2018, doi:10.1109/ACCESS.2018.2872778

[15] R. Logeswaran and C. Eswaran, “Neural network based lossless coding schemes for telemetry data,” In Geoscience and Remote Sensing Symposium, 1999, IGARSS'99 Proceedings, IEEE 1999 International, vol. 4, pp. 2057-2059.

[16] R. Logeswaran, ”Fast Two-Stage Lempel-Ziv Lossless Numeric Telemetry Data Compression Using a Neural Network Predictor,” Journal of Universal Computer Science, vol. 10, no. 9, pp. 1199-1211, 2004.

[17] M. A. Elshafey and I. M. Sidyakin, “Lossless Compression of Telemetry Information using Adaptive Linear Prediction," Science and Education Bauman MSTU, vol. 14, no. 4, pp. 354-363, Apr. 2014, doi:10.7463/0414.0707364.

[18] N. Fout and K. L. Ma, “An Adaptive Prediction-Based Approach to Lossless Compression of Floating-Point Volume Data,” IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 12, pp. 2295-2304, 2012.

[19] Y. Ren, X. Liu, W. Xu, and W. Zhang, “Multi-Channel Data Structure and Real-Time Compression Algorithm Research”, IEEE Aerospace and Electronic Systems Magazine, vol. 24, no. 11, 2009, pp. 28-35, doi:10.1109/MAES.2009.5344179.

[20] K. Sayood, Lossless Compression Handbook, New York: Academic Press, 2003.

[21] P. D. Johnson, G. A. Harris, and D. C. Hankerson, Introduction to Information Theory and Data Compression, Second Edition. Chapman and Hall CRC, 2003.

[22] G. Mandyam, N. Magotra, and S. D. Stearns, “Lossless Waveform Compression,” In Industrial Electronics Handbook, IEEE/CRC press, 1997.

[23] M. Hans and R.W. Schafer, “Lossless Compression of Digital Audio”, IEEE Signal Processing Magazine, 2001, vol. 18, no. 4, pp. 21-32.

[24] T. Robinson, SHORTEN: Simple Lossless And Near-Lossless Waveform Compression, Technical Report. Cambridge University, Engineering Department, CUED/F-INFENG/TR.156, 1994.

[25] J. H. Kim, Lossless Wideband Audio Compression: Prediction and Transform, PhD dissertation, Institute for Communication Sciences, Technical University Berlin, Germany, 2004.

[26] T. Liebchen, “MPEG-4 Lossless Coding for High-Definition Audio," in Audio Engineering Society Convention 115. Audio Engineering Society, 2003.

[27] M. Ohsmann, “Fast Transforms of Toeplitz Matrices,” Linear Algebra and its Applications, vol. 231, pp. 181-192, 1995.

[28] Lossless Data Compression. Report Concerning Space Data System Standard. Informational Report CCSDS 120.0-G-3 Green Book. April 2013. Available online: https: // public.ccsds.org / publications / GreenBooks.aspx [Retrieved: Feb., 2019].

[29] M. A. Elshafey and I. M. Sidyakin, “Simulation of Telemetry Information Transmission over a Noisy Channel,” Engineering Herald, Electronic Scientific and Technical Journal, Russia, 2014, no. 1, pp. 38-51.

[30] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” International Conference on Learning Representations (ICLR), 2016, arXiv preprint arXiv:1510.00149.

 ISBN: 978-1-61208-694-1