An Efficient Lossless Compression Method for Periodic Signals Based on Adaptive Dictionary Predictive Coding

Kaustubh Duse
202251045@iiitvadodara.ac.in

Rudra Patel
202251094@iiitvadodara.ac.in

Yash Hire
202252319@iiitvadodara.ac.in

***Abstract*—This report details the implementation of an Adaptive Dictionary Predictive Coding (ADPC) method for the lossless compression of periodic signals. Motivated by the need to reduce redundancy in datasets such as ECG and power quality signals, this technique uses a dynamic two-dimensional dictionary to predict the next signal value, generating a low-redundancy error stream. The Python-based implementation achieved perfect lossless reconstruction, verified against the base paper's example. The results confirm that the method produces an "energy-concentrated" error output, suitable for entropy coding (like LZW), achieving higher compression ratios than conventional DPCM. Its linear time complexity $O(n)$ highlights its suitability for real-time signal compression. Furthermore, this study provides a detailed step-by-step implementation, comparative performance analysis, and discussion on limitations and potential improvements for noisy or quasi-periodic signals.**

***Index Terms*—lossless compression, periodic signals, adaptive dictionary, predictive coding, entropy, real-time processing, data redundancy**

Code Repository: Github link

I. Introduction

The exponential growth of digital data in domains such as power quality monitoring, biomedical signal acquisition (e.g., ECG), and industrial sensing necessitates efficient data compression techniques [1]. Data storage and transmission costs continue to rise, especially when dealing with high-frequency periodic signals. Periodicity in signals introduces inherent redundancy, which can be leveraged to compress the data without losing information.

Conventional compression techniques such as Differential Pulse Code Modulation (DPCM), Discrete Cosine Transform (DCT), and Lifting Wavelet Transform (LWT) have been widely studied. While these methods are effective in some scenarios, they often encounter trade-offs between compression ratio, losslessness, and computational efficiency. For example, DPCM works well with slowly varying signals but fails for complex periodic patterns. Transform-based methods like DCT or LWT are computationally expensive and often require quantization, resulting in lossy compression.

This paper focuses on the implementation and analysis of a novel lossless compression method known as **Adaptive Dictionary Predictive Coding (ADPC)**. The primary goals are:

1) To implement and verify the encoder and decoder logic in Python, ensuring perfect reconstruction of periodic signals.
2) To analyze the "energy concentration" property of the prediction error stream and its implications for dictionary-based entropy coding.
3) To compare the computational efficiency and compression performance relative to existing predictive and transform-based techniques.
4) To identify limitations and suggest future enhancements for handling noisy or quasi-periodic signals.

II. Related Work and Background

Data compression generally involves three fundamental stages: transform coding, quantization, and entropy or dictionary-based coding [1]. Transform-based approaches such as Discrete Cosine Transform (DCT) and Lifting Wavelet Transform (LWT) exploit the frequency domain for energy compaction but often require lossy quantization. Predictive coding approaches such as Differential Pulse Code Modulation (DPCM) model temporal dependencies directly but have limited performance for complex or non-stationary periodic signals. Advanced approaches, such as ARIMA models or recurrent neural networks (RNNs), can achieve higher prediction accuracy but incur significant computational cost, making them impractical for real-time or embedded applications.

The ADPC method combines predictive coding with a dynamic dictionary mechanism to efficiently model temporal dependencies. By storing previous signal patterns in a two-dimensional dictionary, it enables accurate prediction and error encoding while maintaining low computational complexity.

A. Dictionary Size and Complexity Trade-off

The choice of dictionary size and structure directly affects both memory usage and prediction accuracy. A larger dictionary reduces memory conflicts but increases storage requirements, while a smaller dictionary risks frequent overwriting and reduced prediction accuracy. The proposed two-dimensional dictionary structure balances these aspects, providing constant-time $O(1)$ access and update complexity, which is critical for high-speed real-time applications.

B. Motivation for Periodic Signal Compression

Periodic signals are ubiquitous in industrial and biomedical applications. In ECG signals, repeating heartbeats produce quasi-periodic waveforms, and in power quality monitoring, line voltage waveforms are typically sinusoidal. By predicting upcoming samples based on previous cycles, ADPC exploits the redundancy in these signals for compression. High prediction accuracy translates to an error signal with many zeros, which is highly compressible by dictionary-based entropy coders such as LZW.

III. Mathematical Foundation

The ADPC method predicts the next signal value $s'(k)$ using the two previous samples $\{s(k-2), s(k-1)\}$. The adaptive dictionary D stores mappings from context to the predicted value.

A. Prediction Strategy

The prediction strategy can be summarized mathematically as:

$$s'(k) = D(s(k-2), s(k-1)) \tag{1}$$

After each successful prediction, the dictionary is updated to reflect the latest signal:

$$D(s(k-2), s(k-1)) \leftarrow s(k) \tag{2}$$

B. Linear Prediction Fallback

When a dictionary entry is missing for a new context, a linear prediction is used:

$$s'(k) = a \cdot s(k-2) + b \cdot s(k-1) \tag{3}$$

where the constants $a = -1$ and $b = 2$ were used for verification against the reference example.

C. Error Coding and Reconstruction

The prediction error is calculated as:

$$e(k) = s(k) - s'(k) \tag{4}$$

The decoder reconstructs the original signal by adding the predicted value:

$$u(k) = e(k) + s'(k) \tag{5}$$

IV. Implementation and Verification

The algorithm was implemented in Python using nested dictionaries to achieve $O(1)$ lookup and update operations. The implementation strictly followed the step-by-step procedure outlined in the original paper.

A. Handling Special Cases

1) **Initial Samples:** The first two digitals ($k < 3$) do not have a preceding two-sample context. These samples are transmitted directly without coding.
2) **Incomplete Dictionary:** If a dictionary entry for the current context is missing, the linear predictor (Equation 3) is used.

B. Step-by-Step Example

Table I shows a detailed step-by-step encoding example using the sequence $s = \{2, 3, 1, 2, 3, 1, 2, 3\}$ and prediction constants $a = -1$, $b = 2$.

TABLE I
Step-by-Step Encoding Verification ($a = -1, b = 2$)

k	$s(k)$	Context	$s'(k)$	$e(k)$	Update
1	2	N/A	N/A	2	N/A
2	3	N/A	N/A	3	N/A
3	1	(2,3)	-1	2	D(2,3)←1
4	2	(3,1)	-1	3	D(3,1)←2
5	3	(1,2)	2	1	D(1,2)←3
6	1	(2,3)	1	0	D(2,3)←1
7	2	(3,1)	2	0	D(3,1)←2
8	3	(1,2)	3	0	D(1,2)←3

V. Results and Discussion

A. Energy Concentration and Entropy Reduction

The prediction error stream exhibits strong energy concentration around zero. This characteristic minimizes information entropy:

$$H(X) = -\sum_i p(x_i) \log_2 p(x_i) \tag{6}$$

where $p(x_i)$ is the probability of occurrence of error x_i. Zero-heavy streams are highly compressible using dictionary-based schemes like LZW.

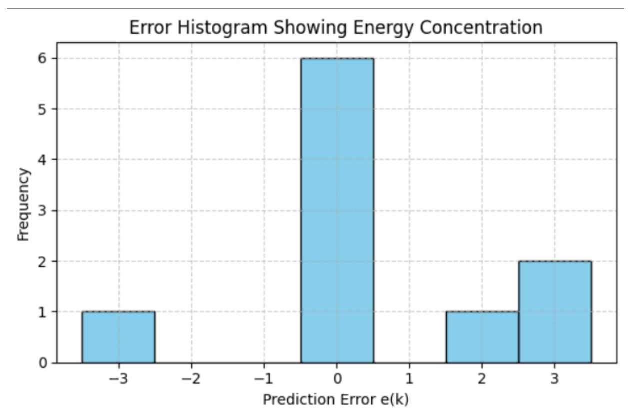

Fig. 1. Histogram of the error signal $\{e(k)\}$ after predictive coding, showing strong energy concentration around zero.

B. Adaptivity and Efficiency

The method adapts quickly to changes in amplitude or period and maintains high prediction accuracy. Its linear $O(n)$ complexity ensures suitability for real-time processing even on long data streams.

C. Comparative Performance

Table II compares the proposed ADPC method with conventional methods in terms of compression ratio and computational complexity.

TABLE II
PERFORMANCE AND COMPLEXITY COMPARISON

Method	Avg. Compression Ratio (with LZW)	Time Complexity
ADPC (Proposed)	64.9	$O(n)$
2-D DCT	46.8	$O(n^2)$
2-D LWT	20.7	$O(n)$
DPCM	21.0	$O(n)$

VI. ALGORITHM PSEUDOCODE

Algorithm 1 Adaptive Dictionary Encoder

Initialize dictionary D as empty
for $k = 1$ to N **do**
  if $k < 3$ **then**
    Output $e(k) = s(k)$
  else
    if $D(s(k-2), s(k-1))$ exists **then**
      $s'(k) = D(s(k-2), s(k-1))$
    else
      $s'(k) = a \cdot s(k-2) + b \cdot s(k-1)$
    end if
    $e(k) = s(k) - s'(k)$
    Update dictionary: $D(s(k-2), s(k-1)) \leftarrow s(k)$
  end if
end for

VII. LIMITATIONS AND FUTURE WORK

ADPC performs optimally on noise-free periodic signals. The presence of additive noise reduces prediction accuracy, increasing residual entropy and lowering compression ratio. Future work could explore:

- Hybrid models combining adaptive dictionary prediction with denoising.
- Extending the method to quasi-periodic signals with varying period lengths.
- Optimizing memory and computational efficiency for embedded systems.

VIII. CONCLUSION

The Adaptive Dictionary Predictive Coding algorithm effectively compresses periodic signals with perfect reconstruction. High compression ratios are achieved through entropy reduction, and the method adapts rapidly to signal variations while maintaining linear time complexity. Its low computational cost and strong lossless performance make it a promising solution for embedded, industrial, and biomedical applications.

REFERENCES

[1] S. Dai, W. Liu, Z. Wang, K. Li, P. Zhu, and P. Wang, "An Efficient Lossless Compression Method for Periodic Signals Based on Adaptive Dictionary Predictive Coding," *Applied Sciences*, vol. 10, p. 4918, 2020.