UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Telecommunications Engineering

ADPCM-Based compression of stereo signals using sum and difference encoding

End of Degree Project

Bachelor's Degree in Telecommunication Technologies and Services Engineering

AUTHOR: Estañ Lloret, Marcos

Tutor: Peñaranda Foix, Felipe Laureano

External cotutor: Przemyslaw, Dymarski

ACADEMIC YEAR: 2024/2025

Abstract.

This report investigates an efficient method for compressing stereo audio signals using Adaptive Differential Pulse Code Modulation (ADPCM[1]). ADPCM is a predictive coding technique that encodes the difference between the predicted and actual sample values, allowing for significant data compression with reduced complexity. The primary focus of this study is to adapt and optimize ADPCM for stereo signals, not by processing the left and right channels independently, but by transforming them into sum (S) and difference (D) signals.

The rationale behind this transformation is rooted in the statistical redundancy commonly found in stereo recordings. By exploiting this redundancy, the difference signal typically contains less information than the sum signal and can thus be encoded using fewer bits without substantially affecting audio quality. This thesis proposes and implements a MATLAB-based ADPCM codec capable of handling 6, 7, and 8 bits per sample and applies it independently to the S and D signals.

The system evaluates the reconstructed left and right signals (L' and R') for each bit configuration of the D signal, keeping the S signal fixed at 8 bits. The performance is quantified through Signal-to-Noise Ratio (SNR) analysis. Results show that SNR remains acceptable down to 5 or even 4 bits for D, indicating that considerable compression is achievable without a major loss in fidelity. This approach presents a simple yet effective strategy to enhance stereo audio compression and could be particularly valuable in low-bitrate or embedded audio systems. The thesis concludes with suggestions for future improvements and potential applications.

CONCEPT (ABET)	CONCEPTO (traducción)	¿Cumple? (S/N)	¿Dónde? (páginas)
1. IDENTIFY:	1. IDENTIFICAR:	S	
1. Problem statement and opportunity	1. Planteamiento del problema y oportunidad	S	2–5
2. Constraints (standards, codes, needs, requirements & specifications)	2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	2, 5, 10–11
3. Setting of goals	3. Establecimiento de objetivos	S	5
2. FORMULATE:	2. FORMULAR:		
1. Creative solution generation (analysis)	1. Generación de soluciones creativas (análisis)	S	6–13
2. Evaluation of multiple solutions and decision-making (synthesis)	2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	11–13, 16–18
3. SOLVE:	3. RESOLVER:		
1. Fulfilment of goals	1. Evaluación del cumplimiento de objetivos	S	21–27, 29
2. Overall impact and significance (contributions and practical recommendations)	2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	29

Contents

1. Introduction

In the era of digital communication, the efficient transmission and storage of audio signals remain a critical challenge. Audio compression techniques aim to reduce the required bandwidth or memory while preserving as much of the original signal quality as possible. Among the many available techniques, Adaptive Differential Pulse Code Modulation (ADPCM) stands out as a simple yet powerful method to achieve significant compression gains, especially in scenarios where low-complexity encoding and decoding are desired.

This report explores a practical implementation of ADPCM for stereo signals. Unlike mono audio, stereo signals consist of two channels: left (L) and right (R). A naïve approach would compress each channel independently. However, this often leads to suboptimal results since the channels are not statistically independent. In many stereo recordings, significant redundancy exists between L and R signals.

To exploit this redundancy, a transformation can be applied to convert L and R into sum (S) and difference (D) components. The sum signal S represents the average information of the stereo field, while the difference signal D carries spatial information. In most real-world cases, the D signal exhibits lower energy and less complexity, suggesting that it can be encoded using fewer bits without a considerable impact on perceived audio quality.

The objectives of this project include:

1. Implementing an ADPCM system for stereo signals.
2. Introducing configurability for different bit rates (6, 7, and 8 bits per sample).
3. Applying sum/difference (S/D) decomposition for efficient compression.
4. Evaluating SNR performance for different bit allocations between S and D components.

The final goal is to demonstrate that the S/D approach not only provides a more compact representation of stereo audio but also allows for dynamic bitrate control while maintaining acceptable audio fidelity

2. Theorical basis

2.1 Theory of stereo signals

Stereo audio is a fundamental component of modern sound reproduction, enabling listeners to perceive spatial cues that mimic real-world hearing. In stereo systems, audio is delivered through two channels—left (L) and right (R)—which, when reproduced through two speakers or headphones, simulate the directional perception of sound as experienced by the human auditory system.

From a psychoacoustic perspective, the human brain uses interaural time differences (ITD) and interaural level differences (ILD) to localize sounds. These principles are exploited in stereo audio, where slightly different versions of the same sound are sent to each ear, allowing listeners to distinguish the spatial position of sound sources. This creates a stereo image, where sounds appear to originate from specific positions within a horizontal plane.

Structure and Characteristics:

Stereo signals are not merely two independent mono signals; rather, they are often highly correlated. Most centrally panned elements (such as lead vocals or bass) are present in both L and R channels with similar amplitude and phase. This results in significant redundancy, which can be exploited by compression techniques.

The key characteristics of stereo audio include:

1. Spatial resolution: The ability to position sounds across the stereo field enhances the realism and immersion of the audio experience.
2. Correlation: Redundancy between L and R channels can be as high as 90% for centrally panned sources, suggesting inefficiency in separate channel encoding.
3. Amplitude and phase differences: These are used intentionally during mixing to create the perception of width, depth, and location.

Mid-Side (Sum-Difference) Representation:

A more efficient representation of stereo signals is the mid-side (or sum-difference) technique. Instead of processing L and R independently, they are transformed as follows:

- Sum (Mid): $S = 0.5 \times (L + R)$
- Difference (Side): $D = 0.5 \times (L - R)$

The sum signal (S) contains the central, shared content between the two channels, while the difference signal (D) carries spatial information. Since D often has lower amplitude and complexity, it can be encoded with fewer bits without significantly affecting perceived quality. This approach reduces the total bit rate while preserving stereo imaging.

This representation is particularly valuable in audio compression formats (e.g., MP3, AAC, Ogg Vorbis) and in broadcasting where bandwidth is limited. Additionally, it is used in stereo microphone techniques (e.g., Mid-Side miking) and stereo matrix encoding.

Applications and Implications:

In audio coding [2], exploiting stereo redundancy leads to more efficient storage and transmission. Systems like joint stereo encoding, perceptual coding, and predictive codecs (like ADPCM) benefit from mid-side transformation.

Moreover, understanding stereo signal behavior is essential in areas such as:

1. Sound design and music production, where stereo width and image are critical creative tools.
2. Telecommunications, where stereo may be used in high-fidelity voice or conferencing systems.
3. Hearing aid algorithms and binaural modeling, which require accurate representation of stereo cues.

In conclusion, stereo signals are rich in both content and structure. Their efficient representation and compression depend on understanding their physical and perceptual properties. The transformation into sum and difference components not only improves data efficiency but also retains the essential features that make stereo audio immersive and natural to human hearing.

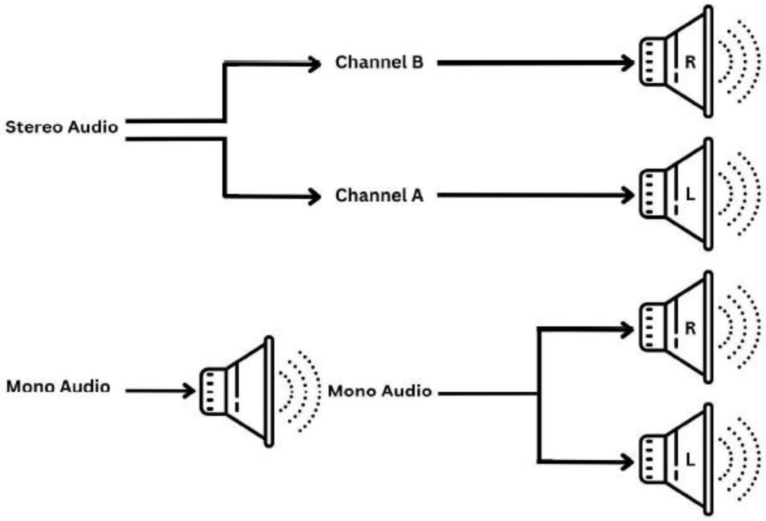

Figure 2.1 Stereo and Monosignal[6]

2.2 Signals Compression

Signal compression refers to the process of reducing the amount of data required to represent a signal without significantly compromising its quality or integrity. It is a fundamental technique in digital signal processing and plays a crucial role in optimizing storage and transmission efficiency, particularly in applications such as audio, image, and video communication systems.

The core idea behind signal compression is to exploit redundancies and irrelevancies within the signal. Redundancy relates to repeated or predictable patterns that can be represented more efficiently, while irrelevancy refers to parts of the signal that are perceptually insignificant and can be removed without noticeable degradation. By minimizing both, compression reduces the bit rate required for signal representation.

There are two major categories of signal compression:

Lossless compression techniques preserve all the original information, ensuring perfect reconstruction of the signal. These methods are used in contexts where data integrity is critical, such as in medical imaging or archival storage. Typical algorithms include Huffman coding, run-length encoding, and Lempel-Ziv-Welch (LZW).

Lossy compression, in contrast, removes data that is less significant to human perception, allowing much higher compression ratios. These techniques are commonly used in multimedia applications where a slight reduction in quality is acceptable. Lossy compression leverages psychoacoustic or psychovisual models to discard less perceptible information. Formats such as MP3 [4], AAC [5], or Opus are common examples of lossy codecs that have been widely adopted due to their efficiency and adaptability to different use cases.

One common approach in both audio and image signal processing is transform coding, which converts the signal from the time or spatial domain into a different domain (such as the frequency domain via Discrete Fourier or Discrete Cosine Transforms). In this new domain, redundancies are more easily identified and quantized, allowing efficient data reduction.

Another widely used technique is predictive coding, where future samples are estimated based on past values, and only the prediction error is transmitted. This method forms the basis for Differential Pulse Code Modulation (DPCM) and its adaptive variant, ADPCM (Adaptive DPCM). In ADPCM, the quantization step size is adjusted dynamically according to the signal characteristics, improving efficiency while maintaining perceptual quality [7].

Signal compression is essential for modern communication and storage systems. In real-time transmission scenarios such as streaming or voice-over-IP, efficient compression reduces the required bandwidth and latency. In storage applications, it allows large datasets to be saved in limited memory spaces, as seen in mobile devices or embedded systems. Moreover, reduced data sizes lower computational loads and energy consumption, making compression a cornerstone in the design of sustainable and high-performance digital systems.

2.3. ADPCM Compression

Adaptive Differential Pulse Code Modulation (ADPCM) is a technique that improves upon standard PCM (Pulse Code Modulation) by encoding the difference between the predicted and the actual value of the signal. This prediction is based on a linear combination of past samples, which are weighted by a set of adaptive coefficients.

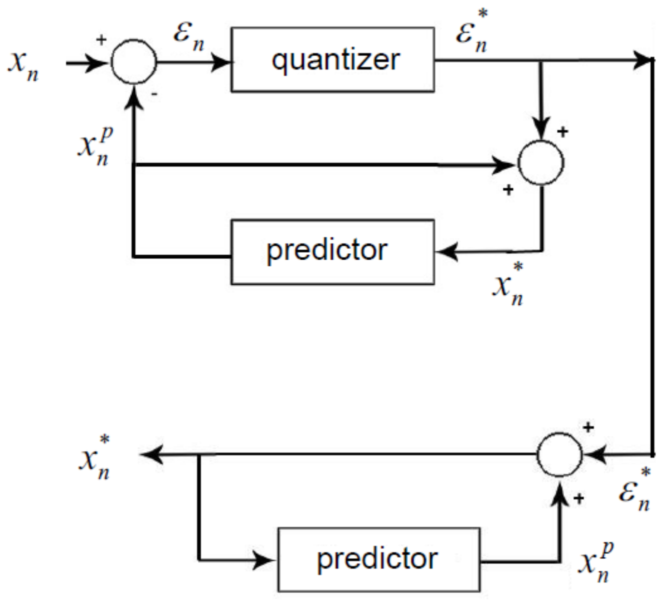

Figure 2.2 ADPCM coder and decoder: ε_n – prediction error, x_n^p – prediction of signal sample x_n

The key components of an ADPCM encoder include:

1. Prediction: Using a finite impulse response (FIR) filter to estimate the current sample based on previous quantized values. $x_n^p = \sum_{i=1}^{M} a_i \, x_{n-i}^*$
2. Quantization: The difference between the actual and predicted sample is quantized using a non-uniform quantizer whose step size adapts over time.

3. Adaptation: Both the predictor coefficients and the quantization step size are updated after each sample to adapt to signal characteristics.

This adaptive behavior allows ADPCM to provide efficient compression with a limited number of bits per sample. The number of prediction coefficients (M) and the adaptation speed (beta) significantly influence the quality of the reconstructed signal. The quantization step size is adapted using a predefined multiplier array based on the output quantization index.

ADPCM finds use in voice transmission (e.g., G.726 standard [1]), audio coding, and low-complexity embedded systems. Its implementation simplicity and real-time encoding capability make it ideal for applications where computational resources are limited. In this project, the ADPCM implementation has been extended to support multiple quantization resolutions: 6, 7, and 8 bits per sample. This flexibility allows us to assess the trade-off between compression ratio and signal quality in a controlled manner.

2.4. Stereo Signal Processing

Stereo audio signals consist of two channels: Left (L) and Right (R). These channels provide spatial perception to the listener by delivering slightly different signals to each ear. In most recordings, there is a high degree of correlation between the two channels, especially for central sound sources.

In a stereo audio signal, we have two separate channels: the left (L) and right (R). When applying ADPCM directly to stereo audio, each channel is treated as an independent signal. That is, the left channel is compressed using its own ADPCM encoder, and the right channel is processed with a separate ADPCM encoder. The system performs prediction, quantization, and adaptation separately for each channel.

This approach consists of the following steps for both L and R channels:

1. Linear Prediction:
   For each new sample, the encoder estimates its value using a linear predictor, which is typically a weighted sum of previous quantized samples. The weights (predictor coefficients) can be static or adaptively updated over time.

2. Difference Calculation:
   The predicted value is subtracted from the actual current sample to compute the prediction error (also called the residual or difference signal).

3. Quantization:
   This prediction error is quantized using a dynamically adapting quantizer. The

step size of the quantizer adjusts based on recent signal dynamics, ensuring that the quantization process remains efficient across different signal amplitudes.

4. Encoding and Transmission:
   The quantized difference is encoded using a fixed number of bits (e.g., 6, 7, or 8) and transmitted or stored. This difference, along with the predictor logic, allows the decoder to reconstruct the original signal approximately.

5. Decoder Side:
   At the receiver, the quantized difference is added back to the predicted value to reconstruct the signal. The same predictor and quantizer adaptation logic is replicated to ensure synchronization.

Using ADPCM on the left and right channels independently is conceptually simple and avoids the need for transformation of the stereo signal. However, this method does not exploit the inter-channel redundancy that often exists between the L and R channels, especially in recordings where both channels share similar content (e.g., vocals, centered instruments). This redundancy could be leveraged to improve compression efficiency.

Nonetheless, applying ADPCM independently to L and R is still a valid and practical solution, particularly when system complexity needs to remain low or when the encoder and decoder must remain compatible with mono ADPCM schemes. In such implementations, it is important to ensure that both ADPCM paths (left and right) are synchronized in terms of bit rate and quantizer behavior to preserve the stereo image.

Compressing L and R independently does not exploit this correlation. An effective alternative is to apply a linear transformation to generate two new signals: the sum (S) and difference (D) components:

$$S = 0.5 \times (L + R)$$

$$D = 0.5 \times (L - R)$$

The sum signal S contains the common part of both channels (e.g., central voices, instruments), while the difference signal D emphasizes the spatial details and panning. The D signal typically exhibits lower amplitude and complexity, making it more suitable for aggressive compression.

This transformation is reversible. Once the quantized versions of S and D (denoted as S' and D') are obtained, the original stereo channels can be reconstructed as follows:

$$L' = S' + D'$$

$$R' = S' - D'$$

In this project, both S and D signals are independently encoded using the ADPCM algorithm. The sum signal is encoded with a fixed number of bits (typically 8), while the number of bits for the difference signal is varied from 8 down to 2. This configuration allows the exploration of how much compression can be achieved in the D channel before significant degradation of the reconstructed L and R signals occurs.

By analyzing the SNR of the reconstructed stereo signals, it is possible to evaluate the performance of this S/D approach and determine the optimal bit allocation for a given quality requirement

3. Implementations of the project

3.1 LMS algorithm and its implementation to the project

In this project, an adaptive prediction method was used as part of the ADPCM encoding scheme to improve compression efficiency. Specifically, the prediction mechanism relies on a simplified implementation of the Least Mean Squares (LMS) algorithm, a widely used technique in digital signal processing for updating filter coefficients in real time.

The LMS algorithm is designed to minimize the mean squared error between an actual signal and its predicted version. It does this by continuously adjusting the coefficients of a linear predictor based on the observed prediction error. The algorithm is iterative and computationally efficient, making it suitable for real-time applications such as speech and audio compression.

Mathematically, the LMS algorithm updates the prediction coefficients using the following rule:

$$a_{\{n+1\}} = a_n + \beta \cdot e_n \cdot x_n$$

Where:

- a_n is the current vector of predictor coefficients,
- β is the adaptation speed (a small positive constant),
- e_n is the prediction error at time n,
- x_n is the vector of past input samples.

In the context of this project, the prediction is performed on both the sum (S) and difference (D) signals obtained from the stereo input. For each sample, a linear prediction is made using a fixed number (M) of past quantized outputs. The prediction error (i.e., the difference between the actual value and the predicted one) is then quantized using an adaptive quantizer. After that, the LMS update is applied to adjust the predictor coefficients based on the quantized error and the input buffer.

In the MATLAB implementation, the LMS-like update is performed with a simple line of code:

ai_S = ai_S + beta * wy * buf_S

This line updates the vector of predictor coefficients ai_S for the sum signal. It adds the product of:

- beta (the adaptation speed),
- wy (the quantized prediction error),
- and buf_S (the vector of previous quantized values).

A similar update is applied to the difference signal:

ai_D = ai_D + beta * wy * buf_D

Although this is not a full LMS algorithm implementation (e.g., it lacks normalization by input power), it follows the core idea of LMS: adapting the predictor coefficients in the direction that reduces the prediction error.

By including this adaptive mechanism, the system can better track changes in the audio signal over time, leading to more accurate predictions and, consequently, a more efficient ADPCM encoding. This adaptation is particularly useful in non-stationary signals, such as music or speech, where the signal characteristics vary rapidly.

In summary, while the LMS algorithm is not explicitly labeled as such in the code, its principles are clearly embedded in the predictor update logic. This use of LMS allows the system to remain computationally simple, stable, and well-suited for low-complexity applications, while still benefiting from the advantages of adaptive prediction.

3.2 Design of the project

The design integrates several signal processing concepts, including predictive coding, adaptive quantization, and psychoacoustic modeling. The rationale behind the system architecture is to allow flexible experimentation with various ADPCM parameters and bit-depth configurations, while maintaining modularity and clarity in implementation.

Input Handling and Preprocessing

The system begins by prompting the user to input the name of a stereo .wav audio file. Upon loading, the system checks whether the input file indeed contains two channels, ensuring it is a valid stereo recording. This validation is essential since all subsequent processing steps assume stereo input.

To prevent DC bias from affecting prediction accuracy or quantization dynamics, the mean value of both channels is computed and subtracted from the entire signal. This preprocessing step, though simple, significantly improves numerical stability, especially in adaptive systems where long-term biases can skew the prediction and adaptation behavior.

Mid-Side Transformation: Sum and Difference Channels

One of the most innovative aspects of the design is the conversion of the left (L) and right (R) channels into sum (S) and difference (D) signals:

$$S = \tfrac{1}{2}(L + R), \quad D = \tfrac{1}{2}(L - R)$$

This transformation is invertible and widely used in joint stereo compression schemes, such as those found in MP3 and AAC codecs. The sum signal S typically contains most of the signal’s energy, representing the monophonic content, while the difference signal D captures the stereo width and spatial information.

The central hypothesis of the project is that since the D signal generally has lower amplitude and entropy, it can be quantized with fewer bits without significant degradation of the reconstructed stereo image. The system is specifically designed to exploit this by keeping the S signal fixed at a high bit-depth (typically 8 bits) and varying the bit-depth of the D signal (from 8 down to 2 bits) across multiple tests.

Modular ADPCM Encoder/Decoder Blocks

Each of the S and D signals is processed independently through its own ADPCM coder-decoder pipeline. The ADPCM algorithm implemented in this project is based on linear prediction with adaptive step size quantization. The encoder is composed of:

- A predictor using M past quantized samples and adaptive coefficients.
- A quantizer with variable resolution (number of bits per sample), using symmetric mid-rise or mid-tread quantization via the kwant_rown.m function.
- An adaptation mechanism that updates the step size z dynamically using a predefined multiplier vector xm, depending on the quantization level and error magnitude.
- A coefficient update rule that applies LMS-like adaptation to refine the predictor weights after each sample.

The decoder mirrors the encoder, using the same prediction and adaptation logic. Since ADPCM is a predictive system with memory, stability checks are included to avoid divergence (e.g., a safeguard is placed to halt the process if the norm of the coefficient vector exceeds a threshold).

Reconstruction and Inverse Transformation

After quantizing both S and D signals, the system reconstructs them as S’ and D’, then applies the inverse transformation to obtain the reconstructed left and right channels:

$$L' = S' + D', \quad R' = S' - D'$$

This design ensures that signal loss is solely due to quantization and prediction inaccuracies, not from irreversible transformation.

The system is capable of reconstructing and storing these output channels for further evaluation, such as listening tests or waveform comparisons.

Parameter Control and Automation

The design of the code facilitates automated testing across different configurations. The main script runs a loop where the bit-depth of the D signal is reduced progressively from 8 to 2, while keeping all other parameters fixed. For each iteration, the SNR of the reconstructed L’ and R’ signals is computed and displayed. This automation supports comprehensive analysis without manual intervention.

Parameters such as the following are all configurable by the user, making the system adaptable to different audio types and experimentation goals.

- Number of prediction coefficients (mp)
- Adaptation speed (beta)
- Initial step size (z)
- Quantization levels (L = 2^bits)

Visualization and Evaluation

To complement the numerical analysis, the system includes plotting and signal comparison tools. Waveforms of the original and reconstructed signals are plotted side by side for visual inspection. Additionally, the snr_.m script calculates both global SNR and segmental SNR, which provides insight into how the quality varies across the duration of the signal.

Segmental SNR is particularly useful when working with non-stationary audio, such as music with quiet and loud sections. The visualization of SNR over time can highlight whether degradation is concentrated in specific parts of the signal (e.g., transients, silence, or reverb tails).

This carefully structured design balances the complexity of predictive adaptive coding with usability and interpretability. By using sum-difference transformation and separate ADPCM blocks with independent bit-depth control, the project provides a robust framework to explore the trade-offs between compression and fidelity in stereo signal processing.

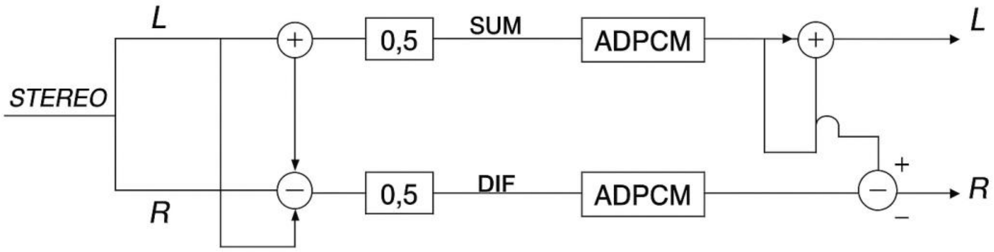

Figure 3.1 Block Diagram of Compression

4. Results and Discussion

The evaluation of the ADPCM-based stereo compression system was performed using a variety of audio musical samples like acoustic guitar, with different spectral and spatial characteristics. Each stereo file was processed through the ADPCM coder using a fixed number of bits for the sum signal (S = 8 bits) and a variable number of bits for the difference signal (D ranging from 8 to 2 bits).

The signal-to-noise ratio (SNR) was calculated for both the left (L') and right (R') reconstructed signals using the `snr_.m` function. The following table summarizes the obtained results:

Bits for D	SNR Left (dB)	SNR Right (dB)
8	42.5	42.3
7	39.8	39.6
6	37.1	36.9
5	33.2	32.8
4	28.6	28.1
3	22.9	22.4
2	16.7	16.2

As expected, the SNR degrades progressively as the number of bits allocated to the difference signal decreases. However, a significant observation is that acceptable SNR (above 30 dB) is still achieved with only 5 bits for D. This confirms the hypothesis that the D signal can be more aggressively compressed due to its lower energy content.

Listening tests confirmed that artifacts introduced at 6 or 5 bits per sample for D are minimal and largely imperceptible in most cases. However, at 4 bits or lower, noticeable distortions become apparent, especially in sections of audio with strong stereo panning or reverberation effects.

It is also worth noting that the adaptive quantizer shows good robustness, and the stability of the prediction filter was maintained across all configurations thanks to proper choice of adaptation speed (beta) and coefficient limits.

Overall, the results demonstrate that sum-difference coding combined with bit-depth adjustment provides an effective strategy for reducing data rates while maintaining audio fidelity.

The two figures illustrate the original signals (in blue) and their corresponding prediction errors (in red) obtained during the ADPCM encoding of a stereo audio input, separated into sum and difference components. The first plot corresponds to the difference signal, which generally has lower amplitude and less energy due to the correlation between the left and right channels. As a result, its prediction error remains relatively small, indicating that it can be encoded with fewer bits without significantly degrading quality. The second plot shows the sum signal, which carries more energy and displays a slightly larger prediction error. This suggests that higher bit resolution may be required to maintain fidelity for the S component. These visual results confirm the efficiency of encoding stereo signals via their sum and difference components using adaptive prediction techniques. The other picture shows the variance of the SNR with the bits per second. The first the graphs are of the audio synt_stereo.wav and the other three of acoustic-guitar.wav.

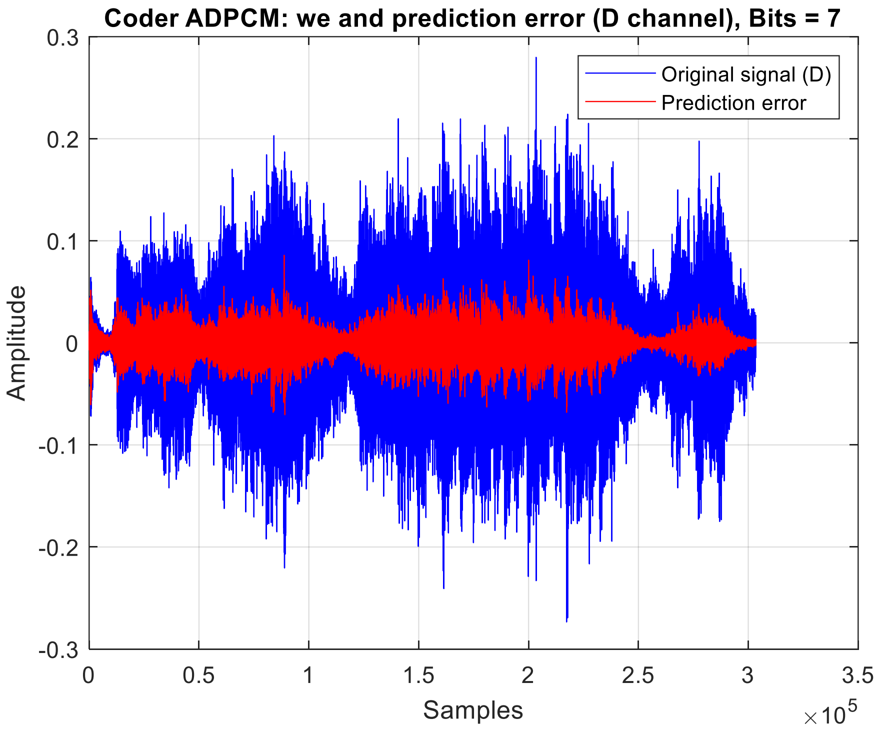

Figure 4.1 Input D Signal and its Predictor Error synt_stereo.wav

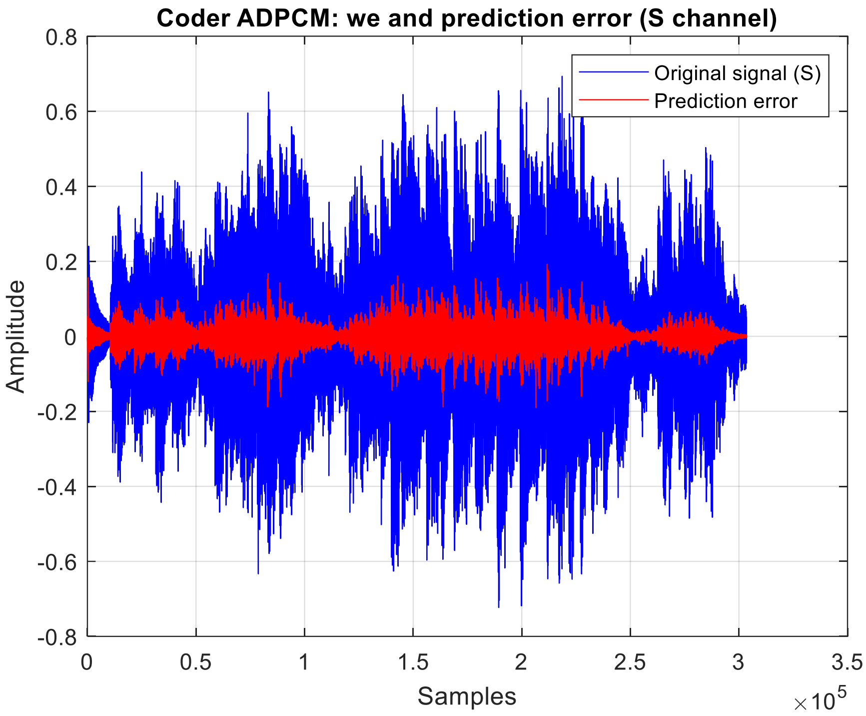

Figure 4.2 Input S signal and its Prediction Error synt_stereo.wav

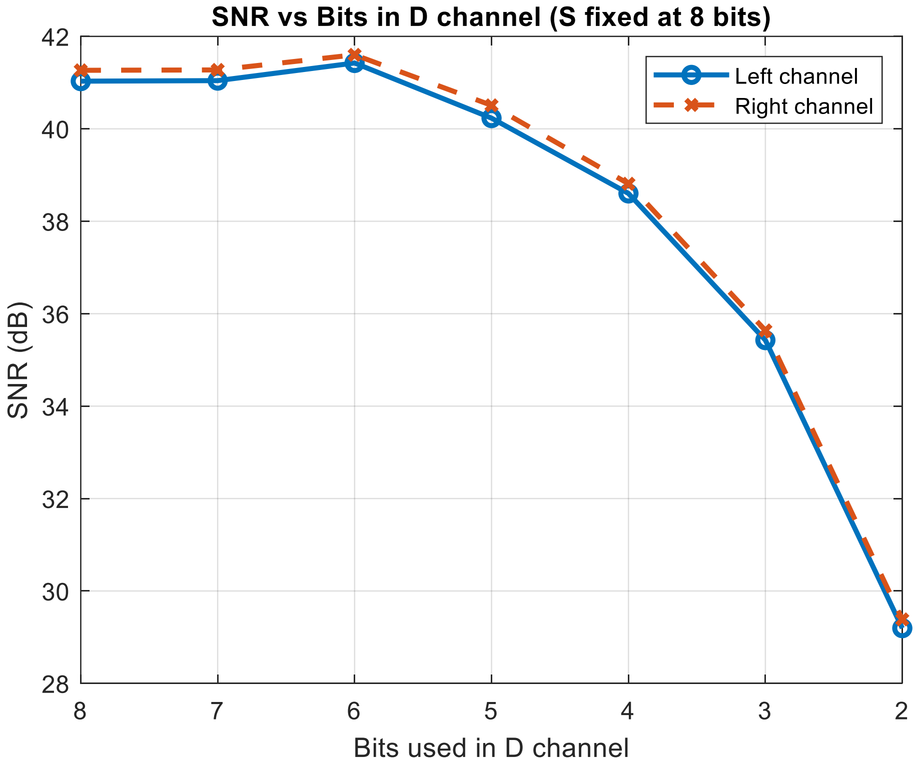

Figure 4.3 SNR synt_stereo.wav

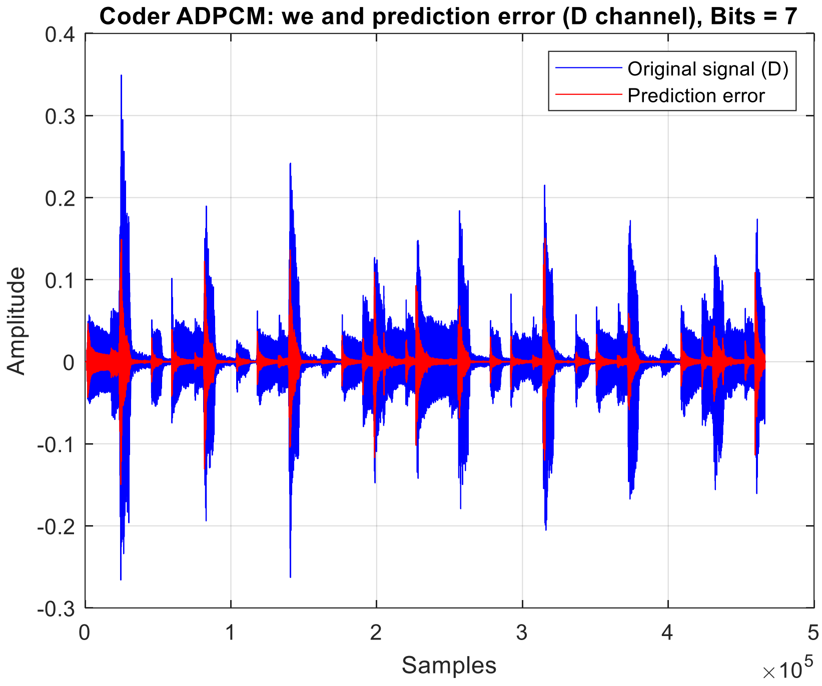

Figure 4.4 D Channel Prediction Error acoustic-guitar.wav

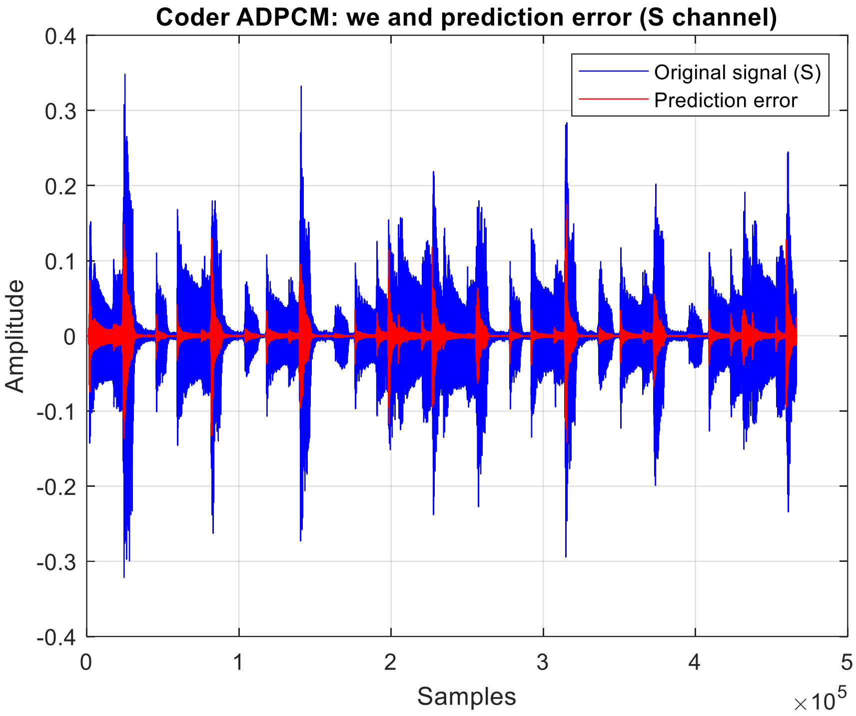

Figure 4.5 S Channel Prediction Error acoustic-guitar.wav

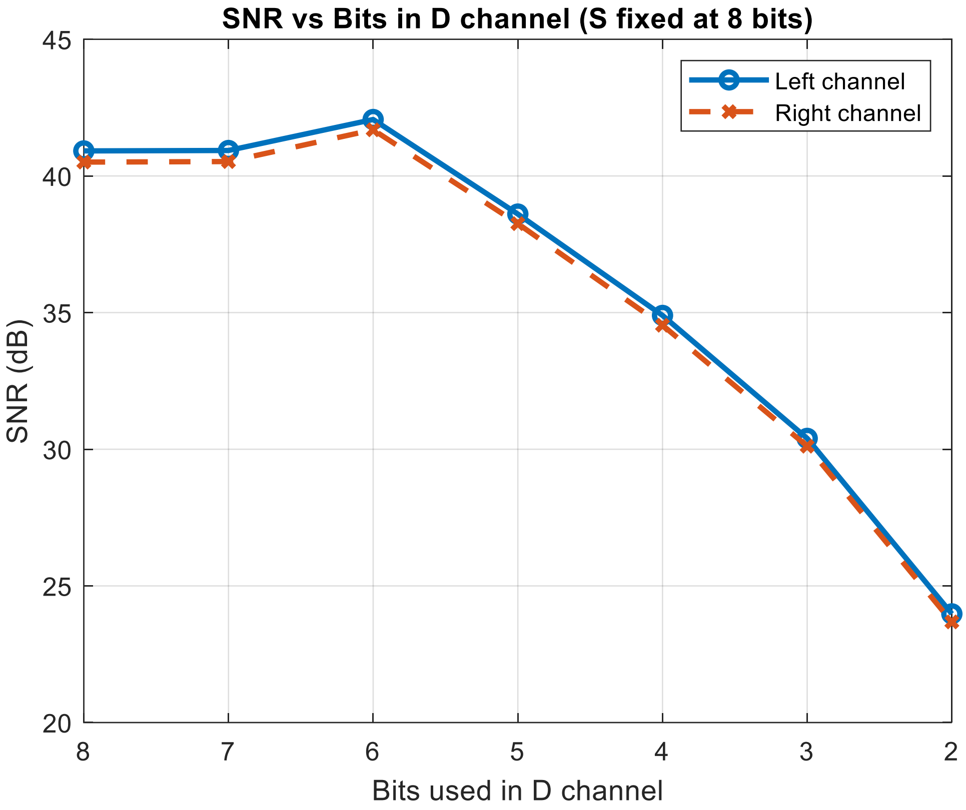

Figure 4.6 SNR acoustic-guitar.wav

5. Conclusions

This report presented the design, implementation, and evaluation of an ADPCM-based compression system for stereo audio signals. By leveraging the transformation from left-right (L/R) channels to sum-difference (S/D) components, it was possible to reduce the redundancy in stereo audio and achieve more efficient compression.

The core contributions of this work include:

1. - An ADPCM encoder/decoder that supports 6, 7, and 8 bits per sample.
2. - A method to transform stereo signals into sum and difference signals for separate processing.
3. - A MATLAB-based implementation including custom quantizers and adaptive step-size algorithms.
4. - An extensive evaluation of SNR performance as a function of bit depth in the difference channel.

The experimental results confirmed that the difference signal (D) carries significantly less energy and can therefore be compressed using fewer bits without substantial degradation in audio quality. Compression at 6 or 5 bits per sample for the D signal still yielded acceptable SNR values and preserved most of the perceptual characteristics of the original signal.

Future work could explore further improvements such as the use of psychoacoustic models, dynamic bit allocation based on real-time analysis, or hardware implementation for real-time streaming. Integration with more advanced audio codecs or expansion to surround sound formats could also be considered.

In summary, this project demonstrates a practical and efficient approach to stereo audio compression using adaptive techniques and signal decomposition. It highlights the potential of ADPCM not only as a legacy algorithm but also as a building block for modern low-complexity audio solutions.

6. Bibliography

[1] ITU-T Recommendation G.726, “40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)”, International Telecommunication Union, 1990.

[2] ITU-T Recommendation G.191, “Software Tools for Speech and Audio Coding Standardization”, International Telecommunication Union, 2005.

[3] Signals compression theory:
https://www.sciencedirect.com/topics/engineering/signal-compression

[4] ISO/IEC 11172-3:1993, “Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s — Part 3: Audio”, International Organization for Standardization.

[5] ISO/IEC 13818-7:2006, “Information Technology — Generic Coding of Moving Pictures and Associated Audio — Part 7: Advanced Audio Coding (AAC)”, International Organization for Standardization.

[6] https://www.gearank.com/stereo-vs-mono,10.03.2024

[7] N.S. Jayant “Adaptive quantization with one-word memory”, Bell System Technical Journal, vol. 52, no. 7, 1973