Numerical Analysis for Machine Learning
Professor Edie Miglio

ECG Compressed Sensing with Kronecker Technique & Adaptive Dictionary Learning

Francesco Rosnati

Master's candidate in High Performance Computing Engineering at Politecnico di Milano

September 3, 2024

Abstract

In long-term ECG monitoring, efficient data compression is crucial to reduce memory usage and extend battery life in portable devices. This study explores the use of compressed sensing (CS) techniques to reduce the amount of data collected without sacrificing signal quality. Two dictionary approaches are compared: the widely-used fixed dictionary method with the Discrete Cosine Transform (DCT), and adaptive dictionary learning techniques, specifically the Method of Optimal Directions (MOD) and K-Singular Value Decomposition (KSVD). Additionally, the Kronecker product technique is applied to both fixed and adaptive dictionaries to assess its impact on signal reconstruction quality.

Contents

1 Introduction

In long-term ECG monitoring, where signals are recorded for extended hours, significant challenges arise related to hardware requirements and post-processing times. This can result in higher costs, longer processing times, and difficulties in adapting to portable devices with limited memory and computational power. Portable devices, which are designed for long-term monitoring, need to balance between power and performance, making standard compression methods unsuitable.

This study explores how compressed sensing can help overcome these limitations by acquiring only partial data directly during the signal capture phase. This approach reduces the local memory burden on portable devices, allowing the signal recovery phase to be handled by more powerful systems. The goal is to design software infrastructure that can operate on simple hardware, such as the system described in Izadi et al. (2020) [4].

The investigation focuses on two techniques: adaptive dictionary learning and the Kronecker technique. Adaptive dictionary learning allows the dictionary used in compression to be tailored to the specific signal, while the Kronecker technique enhances reconstruction by optimizing the measurement matrix. The results are compared to a benchmark using fixed dictionaries like the discrete cosine transform (DCT). This will assess when and how adaptive techniques outperform standard methods in terms of compression efficiency and reconstruction quality.

2 Methodology

In this section there will be offered a theoretical review of the topic relevant to the project, together with a focus on some experimentally relevant aspects of the study.

2.1 Sparsity and Compression

Understanding Sparsity

Theoretical Sparsity: A signal $s \in \mathbb{R}^n$ is considered k-sparse if it has exactly k non-zero elements, with $k \ll n$. This means that $n-k$ elements of the signal are exactly zero.

Practical Sparsity: In real-world applications, exact sparsity is rare. Instead, signals are often approximately sparse: only k elements of the sparse representation carry most of the signal's information, and the remaining $n-k$ elements have small, negligible values. Unlike theoretical sparsity, the $n-k$ coefficients are small but not exactly zero. This is a more realistic approach when dealing with ECG signals and other natural signals [4].

Classic Transformation-Based Compression

Most natural signals, such as images and audio, are highly compressible. This compressibility implies that when the signal is expressed in an appropriate basis, only a few components are active, reducing the amount of data that needs to be stored. A compressible signal $x \in \mathbb{R}^n$ can be written as a sparse vector $s \in \mathbb{R}^n$ in a transform basis $\Psi \in \mathbb{C}^{n \times n}$:

$$x = \Psi s$$

where Ψ is a transformation basis such as the Fourier or wavelet basis. Only a few active terms in s are required to reconstruct the original signal x, significantly reducing the data needed for storage or transmission.

Steps of Transformation-Based Compression: The process of transformation-based compression can be broken down into several key steps. First, the signal capture occurs. In this project, the raw signal x, which corresponds to ECG voltage measurements, is fully sensed and stored for further processing.

Next, the signal x is transformed into a sparse domain by finding the sparse vector $s \in \mathbb{R}^n$. This is achieved using a transformation matrix $\Psi \in \mathbb{C}^{n \times n}$, also referred to as a dictionary. Since Ψ is an orthonormal basis, it satisfies the relation $\Psi^H \Psi = I$, where Ψ^H is the Hermitian conjugate (conjugate transpose) of Ψ. The sparse representation s is obtained through the following relation:

$$s = \Psi^H x$$

After the transformation, sparsification takes place. A threshold is applied to the sparse coefficients to retain only the significant values, discarding the negligible ones. This step is essential to achieve compression, as it reduces the amount of data that needs to be processed.

Finally, the remaining significant coefficients and their respective positions are encoded for storage or transmission. The compressed data can then be efficiently stored or sent over communication channels.

While transformation-based compression methods are highly effective, they rely on a thresholding step that introduces non-linearity and additional computational complexity. As an alternative, this project explores compressed sensing (CS) methods, which can overcome some of these challenges and offer different advantages.

For more details on classic compression techniques and the theoretical background, one can refer to Brunton and Kutz's comprehensive book on data-driven science and engineering [2].

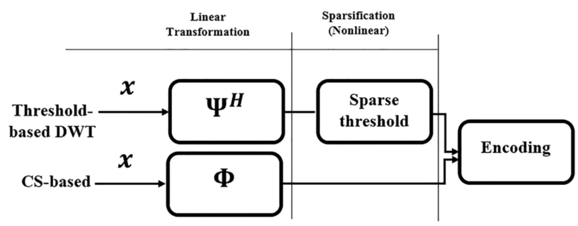


Figure 1: Comparison of methods: classic transformation-based compression vs. compressed sensing.

2.2 Compressed Sensing

Compressed sensing (CS) exploits the sparsity of a signal in a generic basis to achieve full signal reconstruction from surprisingly few measurements. If a signal x is k-sparse in a dictionary Ψ, it becomes possible to collect fewer measurements and still accurately recover the signal by solving for the non-zero elements of its sparse representation s in the transformed coordinate system [2].

Measurement Phase

In the measurement phase, instead of acquiring all n samples, a reduced set of m measurements is obtained by projecting the signal x onto a measurement matrix Φ. This results in a compressed measurement vector y, defined as:

$$y = \Phi x$$

where:

$x \in \mathbb{R}^n$	is the real signal from the sensors,
$y \in \mathbb{R}^m$	is the compressed measurement,
$\Phi \in \mathbb{R}^{m \times n}$	is the measurement matrix with $m \ll n$.

An essential concept in compressed sensing is that in the measurement phase, the sparse representation s is not computed directly. Instead, the measurement matrix Φ is applied to the real signal x, and the matrix Φ typically contains random or structured elements. This ensures that the compressed measurements y retain sufficient information to recover the sparse signal s later.

Although the signal x is not sparse in the time domain, compressed sensing theory leverages the fact that s can be sparsely represented in a transform domain, such as the wavelet or Fourier domain.

Recovery Phase

Given knowledge of s, the sparse representation of x in the dictionary Ψ, it is possible to reconstruct x through the following equation:

$$x = \Psi s$$

The goal of compressed sensing is to find the sparsest vector s that satisfies the relationship:

$$y = \Phi x = \Phi \Psi s$$

where:

$x \in \mathbb{R}^n$	is the real signal,
$y \in \mathbb{R}^m$	is the compressed measurement,
$\Psi \in \mathbb{R}^{n \times n}$	is the dictionary used for sparse representation,
$\Phi \in \mathbb{R}^{m \times n}$	is the measurement matrix,
$s \in \mathbb{R}^n$	is the sparse representation of x in Ψ.

Non-convex Problem

The system of equations is under-determined since there are infinitely many consistent solutions for s. The sparsest solution, which has the fewest non-zero entries, satisfies:

$$\hat{s} = \arg\min_s \|s\|_0 \quad \text{subject to} \quad y = \Phi \Psi s$$

where $\|s\|_0$ represents the ℓ_0-pseudo-norm, which counts the non-zero entries in s. This optimization problem is non-convex, and finding a solution typically requires a combinatorial search, which is intractable for even moderately sized n and k.

Convex Relaxation

Fortunately, under certain conditions on the measurement matrix Φ, it is possible to relax the optimization problem to a convex ℓ_1-minimization:

$$\hat{s} = \arg\min_s \|s\|_1 \quad \text{subject to} \quad y = \Phi \Psi s$$

In the presence of noise, the recovery problem is modified to:

$$\hat{s} = \arg\min_s \|s\|_1 \quad \text{subject to} \quad \|y - \Phi \Psi s\|_2 \leq \epsilon$$

where ϵ is a bound on the noise level. This relaxed problem can be solved more efficiently and is often used in practical applications of compressed sensing.

Key Conditions for Success

There are two critical conditions that must be met for compressed sensing to successfully recover the sparsest solution through ℓ_1-minimization:

Incoherence: Incoherence ensures that the rows of the measurement matrix Φ are not too similar to the columns of the dictionary Ψ. This property ensures that the information about the sparse signal s is spread evenly across the measurements y, allowing accurate recovery from a limited number of measurements.

Recoverability Condition: A k-sparse signal $s \in \mathbb{R}^n$ can be recovered after compressed sensing if the number of measurements m satisfies:

$$m \geq Ck \log\left(\frac{n}{k}\right)$$

where C is a constant that depends on the incoherence between Φ and Ψ. This condition ensures that enough measurements are taken to allow accurate recovery of the sparse signal.

Restricted Isometry Property (RIP)

The Restricted Isometry Property (RIP) is a condition that ensures the measurement matrix $\Phi\Psi$ preserves the geometry of sparse signals. Specifically, for a matrix to satisfy the RIP of order k with constant δ_k, it must hold that:

$$(1-\delta_k)\|x\|_2^2 \leq \|\Phi\Psi x\|_2^2 \leq (1+\delta_k)\|x\|_2^2$$

for all k-sparse vectors x. The constant δ_k should be close to zero to ensure that the matrix approximately preserves the Euclidean length of all k-sparse signals, making it possible to recover the original signal from the measurements.

2.3 Relevant Aspects and Metrics

Work on Signal Block: ECG signals provide continuous data sampling, with record lengths varying from a few minutes to several hours or even days, depending on the purpose of the monitoring. This work focuses on small portable devices that capture signal blocks ranging from 16 to 1024 samples to be compressed. Each signal block is treated as an independent unit for compression, reflecting the constraints and limitations of these portable systems.

Compression Ratio (CR): The compression ratio (CR) is a crucial factor for evaluating different compression methods. It quantifies the amount of compression achieved, defined as:

$$CR(\%) = 100 \times \frac{n-m}{n}$$

where m is the number of compressed samples, and n is the number of original samples. A higher CR indicates a greater level of compression, which is essential for reducing the storage and transmission load in portable devices [4].

Compression Algorithm Complexity: The complexity of the compression algorithm plays a critical role, particularly for limited-resource ECG recorders. The power consumption of these devices is often directly related to the complexity of the system, which is an important consideration for 24-hour ambulatory or remote ECG recorders. Low power consumption is a key requirement for extended monitoring, making it crucial to focus on systems with lower complexity in both the sampling and processing phases.

In this study, a key goal is to demonstrate that a smaller measurement matrix reduces the complexity of the sampling phase, leading to more efficient data acquisition and lower energy consumption, as was shown in the work by Izadi et al. [4].

Processing Speed: In emergency situations, the speed of processing becomes critical. For ambulatory ECG recorders, faster data acquisition and processing allow physicians to access the data sooner, enabling quicker decision-making. Both the acquisition phase and the reconstruction complexity must be considered to ensure the ECG data is usable and available quickly.

In this work, the same fundamental metrics are applied and evaluation aspects as those proposed by Izadi et al. [4].

Assess the Accuracy of Reconstructed Signals

The accuracy of the reconstructed signal in ECG compression algorithms is typically evaluated using two common metrics: the Percentage Root Mean Square Difference (PRD) and Signal-to-Noise Ratio (SNR).

Percentage Root Mean Square Difference (PRD): The PRD measures the difference between the original ECG signal and the reconstructed signal. It is defined as:

$$\text{PRD} = 100 \times \sqrt{\frac{\sum_{i=0}^{N-1}(x(n)-\hat{x}(n))^2}{\sum_{i=0}^{N-1}x(n)^2}}$$

where: - $x(n)$ is the original ECG signal, - $\hat{x}(n)$ is the reconstructed ECG signal, and - N is the length of the signal.

Signal-to-Noise Ratio (SNR): The SNR is another metric used to assess the quality of the reconstructed signal. It is calculated from the PRD using the following formula:

$$\text{SNR} = -20\log_{10}\left(\frac{\text{PRD}}{100}\right)$$

Quality Assessment Based on PRD and SNR

The following table classifies the quality of the reconstructed signal based on PRD and SNR values, as outlined in the work by Izadi et al. [4]:

Quality	PRD Range	SNR Range
Very Good	$0\% < \text{PRD} < 2\%$	$\text{SNR} > 33\,\text{dB}$
Good	$2\% < \text{PRD} < 9\%$	$20\,\text{dB} < \text{SNR} < 33\,\text{dB}$
Undetermined	$\text{PRD} \geq 9\%$	$\text{SNR} \leq 20\,\text{dB}$

Table 1: Quality classification based on PRD and SNR values [4].

These metrics will be used throughout this study to evaluate the accuracy and quality of the reconstructed ECG signals.

2.4 Measurement Matrix

ECG devices provide continuous data sampling for consecutive hours. For instance, the MIT-BIH Arrhythmia Database offers records for each patient for approximately 30 hours, sampled at 360 samples per second. This results in each record containing about 650,000 samples.

A compressed sensing (CS) approach allows for storing only a fraction of this data by immediately computing compressed measurements. Although this study does not delve into hardware specifics, the work of Izadi, Shahri, and Ahani (2020) [4] provides a possible hardware implementation.

In this study, the focus is on working with groups of consecutive samples, known as *signal blocks*, within a record, rather than compressing the entire signal at once.

Compressing Blocks of Samples within a Signal: In compressed sensing, the signal x is compressed using:

$$y = \Phi x$$

where:

$x \in \mathbb{R}^n$	is the original signal from the sensors,
$y \in \mathbb{R}^m$	is the compressed measurement, and
$\Phi \in \mathbb{R}^{m \times n}$	is the measurement matrix, with $m \ll n$.

For practical purposes, the signal x is divided into blocks. The block $x_{\text{block}} \in \mathbb{R}^d$ is the group of consecutive samples, where d represents the block size. The compression for each block is given by:

$$y_{\text{block}} = \Phi_{p,d} \cdot x_{\text{block}}$$

where $\Phi_{p,d} \in \mathbb{R}^{p \times d}$, with $p \ll d$, reduces d original samples to p compressed samples for each block. The final compressed measurement y for the whole signal is obtained by concatenating the compressed blocks.

Ensuring Restricted Isometry Property (RIP)

The Restricted Isometry Property (RIP) is crucial for ensuring that compressed sensing can accurately recover sparse signals from a reduced number of measurements. However, verifying whether a specific matrix $A = \Phi\Psi$ satisfies RIP is computationally infeasible for large matrices because it would require checking the property across all possible sparse vectors.
To address this, generating the measurement matrix Φ randomly, as in this study, increases the likelihood that the matrix $A = \Phi\Psi$ satisfies RIP, without the need for direct verification. This inherent randomness provides a strong theoretical basis for the effectiveness of compressed sensing [4].

Measurement Matrices Used in the Project: In this project, both deterministic and random measurement matrices are employed. Specifically:

- A **Deterministic Binary Block Diagonal (DBBD)** matrix is used as a deterministic matrix,
- Additionally, **random binary** and **Gaussian random matrices** are utilized for comparison.

These different types of matrices allow the study to investigate how deterministic and random matrices impact compression performance and recovery accuracy.

2.5 Dictionary

In signal processing, the choice of dictionary plays a pivotal role in compressing and reconstructing signals efficiently. In the context of ECG signal compression, two broad categories of dictionaries are commonly used: fixed dictionaries and adaptive dictionary learning techniques. Fixed dictionaries, such as the Discrete Cosine Transform (DCT), offer a standardized basis for sparsifying signals but are limited by their inability to adapt to the specific characteristics of the data. In contrast, adaptive dictionary learning provides a flexible, data-driven approach to sparsity that can significantly enhance the quality of the reconstructed signal, especially when applied to stable signals like long-term ECG recordings.

Fixed Dictionaries: Fixed dictionaries, like the DCT, have been extensively used as standard tools in signal compression due to their simplicity and efficiency. These methods transform the signal into a sparse domain, where most of the signal's energy is concentrated in a few coefficients. However, fixed dictionaries are not tailored to the specific characteristics of the signal, potentially limiting their ability to achieve optimal sparsity.

In this work, the DCT is used as a benchmark to evaluate the improvements offered by adaptive dictionary learning.

Further details on DCT are provided in the appendix.

Adaptive Dictionary Learning: Adaptive dictionary learning techniques, unlike fixed dictionaries, are designed to learn a basis or dictionary from the data itself, optimizing it for sparsity. This approach allows for better signal representation, particularly when the characteristics of the signal are consistent, as is the case with long-term ECG recordings. By learning the dictionary directly from the signal, adaptive methods can provide a much sparser representation compared to fixed dictionaries like DCT. The concept of learning a sparse code from natural signals was first introduced by Olshausen and Field (1996) [6], where they demonstrated that adaptive dictionaries could capture key structural features in signals like natural images. This principle has since been extended to other domains, including biomedical signals such as ECG.

In this study, two main adaptive dictionary learning algorithms are employed: *Method of Optimal Directions (MOD)* and *K-Singular Value Decomposition (K-SVD)*.

Both MOD and K-SVD offer significant advantages over fixed dictionaries by generating a dictionary that is more closely aligned with the characteristics of the ECG signal. This tailored approach compensates for the reduction in the size of the measurement matrix, ensuring that the quality of the reconstructed signal remains high, even as the number of measurements is reduced.

Method of Optimal Directions (MOD)

The *Method of Optimal Directions (MOD)*, proposed by Engan et al. (1999) [3], is an iterative dictionary learning algorithm that seeks to minimize the reconstruction error for a given set of sparse coefficients. At each iteration, MOD updates the dictionary by solving a least-squares problem, ensuring that the dictionary adapts to the data to provide the best possible sparse representation.

In MOD, the signal x is decomposed as:

$$x = \Psi s$$

where:

$$x \in \mathbb{R}^n \quad \text{is the signal,}$$
$$\Psi \in \mathbb{R}^{n \times k} \quad \text{is the dictionary,}$$
$$s \in \mathbb{R}^k \quad \text{is the sparse coefficient vector.}$$

MOD iteratively updates the dictionary Ψ to minimize the reconstruction error for each signal x.

The process alternates between two primary steps:

- **Approximation/Selection**: Given an initial set of vectors, approximate the target data using a subset of these vectors.
- **Update/Adjustment**: Adjust the vectors to minimize the residual error based on the current approximations.

In each iteration, MOD improves the accuracy of approximations by optimally adjusting the vectors to reduce the error, continuing until the stopping criteria (e.g., convergence or a set number of iterations) is satisfied.

Algorithm 1 Method of Optimal Directions (MOD)

Initialization:
Initialize a set of vectors $F = \{f_1, f_2, \ldots, f_K\}$
Set maximum iterations `max_iter`
Set stopping criterion (e.g., minimum change in error)
MOD Iterative Process:
for $i = 1$ to `max_iter` **do**
  Approximation (Vector Selection):
  for each target vector x_l **do**
    Approximate x_l using a subset of vectors from F
    Calculate coefficients $w_l(j)$ for the selected vectors
  end for
  Compute Residuals:
  for each target vector x_l **do**
    $\texttt{residual}_l = x_l - \sum_j w_l(j) \cdot f_j$
  end for
  Update Vectors:
  for each vector f_j in F **do**
    Adjust f_j to minimize the sum of residuals:
    $f_j = f_j + \Delta f_j$ ▷ Optimal adjustment
  end for
  Check Stopping Criterion:
  if stopping criterion is met **then**
    break
  end if
end for
Return the optimized set of vectors F

K-Singular Value Decomposition (K-SVD)

The *K-Singular Value Decomposition (K-SVD)*, introduced by Aharon et al. (2006) [1], is an adaptive dictionary learning algorithm that iteratively updates the dictionary by optimizing both the sparse representation coefficients and the dictionary atoms. Unlike the Method of Optimal Directions (MOD), K-SVD updates each dictionary atom individually through Singular Value Decomposition (SVD), thus optimizing the overall sparse representation of the signal.

At each iteration, K-SVD solves the following optimization problem:

$$\min_{\Psi, s} \|x - \Psi s\|_2^2$$

where:

$$x \in \mathbb{R}^n \quad \text{is the signal,}$$
$$\Psi \in \mathbb{R}^{n \times k} \quad \text{is the dictionary,}$$
$$s \in \mathbb{R}^k \quad \text{is the sparse coefficient vector.}$$

The goal is to minimize the reconstruction error while ensuring that the sparse coefficients remain sparse.

The objective is to solve the following optimization problem for a set of training signals $\{x_i\}_{i=1}^N \in \mathbb{R}^n$:

$$\min_{D, \Gamma} \sum_{i=1}^{N} \|x_i - D\gamma_i\|_2^2 \quad \text{subject to} \quad \|\gamma_i\|_0 \leq T,$$

where $\|\gamma_i\|_0 \leq T$ is the sparsity constraint, meaning each signal x_i is represented by at most T non-zero coefficients.

The K-SVD algorithm alternates between two primary steps:

- **Sparse Coding Stage:** In this stage, the dictionary D is fixed, and the sparse coefficients γ_i are computed for each signal x_i by solving the following optimization problem:

  $$\min_{\gamma_i} \|x_i - D\gamma_i\|_2^2 \quad \text{subject to} \quad \|\gamma_i\|_0 \leq T.$$

  Sparse coding algorithms such as Orthogonal Matching Pursuit (OMP) are often used to efficiently approximate the sparse coefficients.

- **Dictionary Update Stage:** After sparse coding, each dictionary atom d_k is updated individually. For the signals using atom d_k, the residual error matrix E_k is computed as:

  $$E_k = X_{\omega_k} - \sum_{j \neq k} d_j \gamma_j,$$

  where X_{ω_k} contains the signals that use d_k and γ_j are the corresponding coefficients for other atoms. The rank-1 approximation of E_k is computed using SVD:

  $$E_k = U \Sigma V^T,$$

  where U, Σ, and V are the components of the SVD. The dictionary atom d_k is updated as the first column of U, and the corresponding sparse coefficients are updated as the first column of V^T, scaled by the largest singular value σ_1.

The K-SVD algorithm alternates between these two steps, refining the dictionary to better represent the data with sparse coefficients. When constrained to select only one dictionary atom per signal, K-SVD reduces to the classical K-means algorithm.

Efficiency Enhancements:

- Atoms that are used infrequently are replaced by random signals from the training set to avoid local minima.
- Atoms with high inner products (i.e., similar atoms) are pruned to reduce redundancy and improve efficiency.

Algorithm 2 K-SVD Algorithm

```
Input: Training signals X = {x_1, x_2, ..., x_N}, sparsity level T
Initialize: Dictionary D, maximum iterations max_iter, stopping criterion
for i = 1 to max_iter do
    Sparse Coding:
    for each signal x_i do
        Compute sparse coefficients γ_i by solving
            min_{γ_i} ||x_i − D γ_i||_2^2   subject to   ||γ_i||_0 ≤ T
    end for
    Dictionary Update:
    for each dictionary atom d_k do
        Compute the residual matrix E_k = X_{ω_k} − Σ_{j≠k} d_j γ_j
        Perform SVD on E_k: E_k = U Σ V^T
        Update d_k as the first column of U, and update the coefficients as the first column of V^T scaled by the largest singular value σ_1
    end for
    Check Stopping Criterion:
    if stopping criterion is met then
        break
    end if
end for
Output: Optimized dictionary D, sparse coefficients γ_i
```

2.6 Recovery Method: SL0

Compressed sensing (CS) seeks to recover a sparse signal by solving an underdetermined system of equations. A common approach is to minimize the ℓ_1-norm as an approximation to the ℓ_0-minimization problem, which is generally NP-hard. However, solving ℓ_1-minimization directly can be computationally expensive, especially in the presence of noise.

The problem is typically framed as:

$$\hat{s} = \arg_s \min \|s\|_1 \quad \text{subject to} \quad \|y - \Phi\Psi s\|_2 \leq \epsilon$$

where ϵ is a bound on the noise level, Φ is the measurement matrix, Ψ is the dictionary, and y represents the compressed measurements. [5]

Smoothed ℓ_0 (SL0) Algorithm The Smoothed ℓ_0 (SL0) algorithm offers a faster alternative to solving the ℓ_1-minimization problem by directly approximating the ℓ_0-norm using a smooth function. The ℓ_0-norm counts the number of non-zero elements in a vector, but solving an ℓ_0-minimization problem is computationally prohibitive. SL0 addresses this challenge by approximating the ℓ_0-norm with a smooth function that can be minimized efficiently.

SL0 Algorithm Steps

The SL0 algorithm works through a series of steps to progressively approximate the ℓ_0-norm:

1. **Initialization:** The algorithm starts with an initial estimate of the sparse coefficients, often obtained through a least-squares solution.
2. **Smoothing and Gradient Descent:** A smooth approximation of the ℓ_0-norm is applied, and gradient descent is used to minimize this function. This step promotes sparsity in the solution.
3. **Progressive Narrowing:** The smooth approximation is progressively made less smooth over iterations, which brings the solution closer to that of the true ℓ_0-minimization. As the smoothing parameter decreases, the solution becomes increasingly sparse.
4. **Stopping Criterion:** The algorithm halts when the desired level of sparsity is achieved or when the smooth approximation sufficiently converges to the ℓ_0-norm.

SL0 avoids the high computational cost of ℓ_1-minimization while still finding sparse solutions. It also accommodates noise by relaxing the equality constraint, similar to the standard compressed sensing problem:

$$\hat{s} = \arg_s \min \|s\|_0 \quad \text{subject to} \quad \|y - \Phi\Psi s\|_2 \leq \epsilon$$

Though SL0 is an approximation of the ℓ_0-norm, it has been shown to provide efficient and accurate solutions in practice. It is particularly useful in situations where noise tolerance is required and computational resources are limited, offering a faster alternative to ℓ_1-based methods. [5]

2.7 Kronecker Technique

The Kronecker technique leverages the Kronecker product[1] to reduce the dimensionality of the problem and improve computational efficiency. This approach can be used in various ways to handle multiple measurement vectors and achieve compressed sensing recovery.

In the following, two different methods to apply Kronecker Technique are presented:

Method A

In *Method A*, the Kronecker product is applied to the product of the measurement matrix Φ and the dictionary

[1] See Appendix for Kron Product

Ψ. The process begins by first multiplying the measurement matrix and the dictionary:

$$\hat{\Phi}_{p\times n} = \Phi_{p\times n}\Psi_{n\times n}$$

Once the product $\Phi\Psi$ is obtained, the Kronecker product is applied across multiple signal blocks, concatenating the measurements for all segments. The overall measurement matrix is then expressed as:

$$\hat{\Phi}_{tp\times tn} = I_{t\times t} \otimes (\Phi_{p\times n}\Psi_{n\times n}),$$

where t is the Kronecker factor representing the number of segments considered together, $\Phi_{p\times n}$ is the measurement matrix, and $\Psi_{n\times n}$ is the dictionary used to sparsify the signal.

The sparse coefficient vector $s_{tn\times 1}$ for the entire signal is computed as:

$$s_{tn\times 1} = F(y_{tp\times 1}, \hat{\Phi}_{tp\times tn}),$$

where $s_{tn\times 1}$ is the concatenation of the sparse coefficients for each segment.

By applying the Kronecker product to $\Phi\Psi$, Method A enables the processing of multiple signal blocks as one larger signal, ensuring that the overall reconstruction maintains coherence while leveraging the efficiency of the Kronecker product for compressed sensing. [7]

Method B

In *Method B*, the Kronecker product is applied only to the measurement matrix Φ, while the sparsifying dictionary Ψ is generated directly at the larger dimension without needing to apply the Kronecker product. The measurement matrix is defined as:

$$\hat{\Phi}_{tp\times tn} = I_{t\times t} \otimes \Phi_{p\times n}$$

The sparsifying dictionary $\Psi_{tn\times tn}$ is generated directly at the larger dimension, bypassing the need for a Kronecker product. The full matrix representation of the combined process is:

$$\hat{A}_{tp\times tn} = \begin{bmatrix} \Phi_{p\times n} & 0 & \cdots & 0 \\ 0 & \Phi_{p\times n} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Phi_{p\times n} \end{bmatrix} \cdot \Psi_{tn\times tn}$$

This method adjusts the sparsifying dictionary at a larger dimension without modifying it after applying the Kronecker product. Instead, the projection matrix Φ is modified, and the resulting dictionary $\Psi_{tn\times tn}$ accommodates the entire set of signal blocks.

By applying the Kronecker product only to Φ, Method B ensures flexibility in selecting the sparsifying basis while maintaining lower mutual coherence. This approach can result in improved signal recovery quality, as represented by the mutual coherence inequality:

$$\mu(\Phi, \Psi_l) < \mu(\Phi, \Psi_s),$$

where $\mu(\Phi, \Psi_l)$ and $\mu(\Phi, \Psi_s)$ represent the mutual coherence between the measurement matrix Φ and the large and small sparsifying bases, respectively. [7]

3 Results

This section presents the experimental results, comparing the performance of different compression methods under two distinct measurement matrix scenarios: a deterministic Block Diagonal Binary (DBBD) matrix and a random unscaled binary matrix. The effectiveness of using both fixed and adaptive dictionaries for ECG signal compression and reconstruction is analyzed, with a focus on key metrics such as Signal-to-Noise Ratio (SNR).

3.1 DBBD Measurements Analysis

The purpose of this study is to evaluate whether adaptive dictionary learning methods—MOD and KSVD—offer significant improvements over the fixed dictionary (DCT) when applied with a DBBD measurement matrix. Additionally, the study examines whether incorporating the Kronecker technique enhances performance in both DBBD-based compression and adaptive dictionary learning approaches.

Performance Overview The following table summarizes the key results from three different methods (DCT, MOD, and KSVD) using a DBBD measurement matrix. The results also include cases where the Kronecker technique was applied to each method:

Method	Average SNR	Min SNR	Max SNR
DCT	45.16 dB	45.16 dB	45.16 dB
MOD	43.56 dB	43.56 dB	43.56 dB
KSVD	41.69 dB	39.90 dB	43.47 dB
DCT-KRON	47.53 dB	47.53 dB	47.53 dB
MOD-KRON	45.29 dB	45.29 dB	45.29 dB
KSVD-KRON	43.05 dB	38.93 dB	45.15 dB

Table 2: SNR analysis of different methods with DBBD measurement matrices.

DCT Performance Analysis The DCT method consistently achieves an average SNR of 45.16 dB, attributed to the deterministic nature of the DCT and DBBD combination. This high SNR demonstrates the effectiveness of fixed dictionaries for sparse signal recovery in ECG compression when used with a DBBD measurement matrix.

Applying the Kronecker technique (DBBD-DCT-KRON) increases the average SNR to 47.53 dB, underscoring the benefits of the Kronecker product in enhancing compression efficiency without significant loss of information, even when using a fixed dictionary.

MOD Performance Analysis In the case of the MOD algorithm, the average SNR drops slightly to 43.56 dB, which suggests that while adaptive dictionary learning has the potential to outperform fixed dictionaries, it may not always be effective with a DBBD measurement matrix. This could be due to the deterministic structure of DBBD not fully exploiting the flexibility offered by MOD.

However, when the Kronecker technique is applied (DBBD-MOD-KRON), the average SNR rises to 45.29 dB. While still not surpassing the DCT's performance, the Kronecker technique does improve the effectiveness of the MOD method. This result suggests that for applications where adaptive dictionary learning methods are preferred, the Kronecker technique may be essential for reaching comparable or better performance than traditional fixed dictionaries like DCT.

KSVD Performance Analysis KSVD shows the lowest average SNR at 41.69 dB, with significant variation between trials (Min SNR of 39.90 dB and Max SNR of 43.47 dB). This variability is likely due to the non-deterministic nature of KSVD, where the dictionary atoms are iteratively updated, sometimes leading to suboptimal sparse representations when used with a DBBD measurement matrix.

When the Kronecker technique is applied (DBBD-KSVD-KRON), the average SNR improves to 43.05 dB. Although the Kronecker technique boosts the performance, it still does not outperform the DCT-based approach. This result suggests that while KSVD benefits from the added structure imposed by the Kronecker product, it may not be the best-suited adaptive dictionary learning method when paired with a DBBD measurement matrix.

Impact of the Kronecker Technique The Kronecker technique consistently improves the SNR for all methods, with the most notable improvements observed in the DCT-based approach. Specifically, DBBD-DCT-KRON achieves the highest overall SNR (47.53 dB), clearly demonstrating the benefit of using the Kronecker technique with fixed dictionaries.

For adaptive dictionary learning methods, the Kronecker technique still provides a noticeable boost. The SNR improvements for both MOD (45.29 dB) and KSVD (43.05 dB) with the Kronecker technique suggest that it helps mitigate some of the limitations imposed by the DBBD measurement matrix. However, neither method is able to surpass the DCT-based approach, even with the Kronecker product applied.

Conclusion The results indicate that while adaptive dictionary learning methods such as MOD and KSVD have the potential to outperform fixed dictionaries in some scenarios, they do not show a significant improvement over DCT when paired with a deterministic DBBD measurement matrix.

The Kronecker technique plays an important role in enhancing performance for both fixed and adaptive dictionaries, with the greatest benefit observed in the DCT-based approach. However, even with the Kronecker product, neither MOD nor KSVD were able to outperform the DCT method. Thus, for applications involving a DBBD measurement matrix, DCT remains the most effective option, particularly when combined with the Kronecker technique.

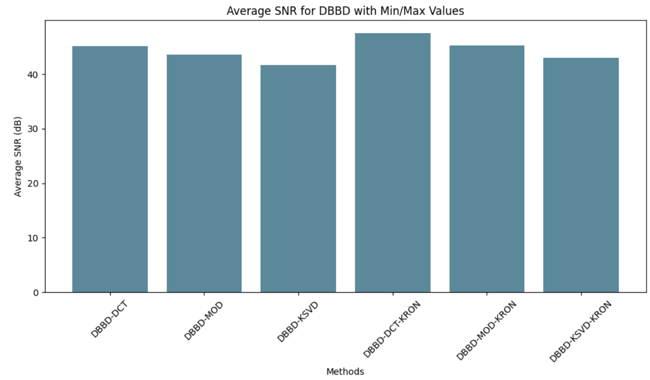


Figure 2: This graph visually represents the comparison between the different methods (DCT, MOD, KSVD) and the effect of applying the Kronecker technique, when a DBBD measurement matrix is used.

3.2 Random "Unscaled Binary" Measurements Analysis

This section analyzes the performance of three primary methods (DCT, MOD, KSVD) when paired with a random "unscaled binary" measurement matrix. The impact of the Kronecker technique is also examined for each method to assess whether it significantly enhances signal reconstruction performance.

Performance Overview The following table summarizes the SNR results for each method, with and without the Kronecker technique:

Method	Average SNR	Min SNR	Max SNR
DCT	16.71 dB	-1.58 dB	34.47 dB
MOD	35.30 dB	21.48 dB	40.40 dB
KSVD	32.54 dB	17.40 dB	40.08 dB
DCT-KRON	17.62 dB	-1.44 dB	33.19 dB
MOD-KRON	38.67 dB	31.73 dB	45.04 dB
KSVD-KRON	37.65 dB	31.39 dB	43.07 dB

Table 3: SNR analysis of different methods with unscaled binary random measurement matrices.

DCT Performance Analysis With the random unscaled binary measurement matrix, DCT shows a considerably lower average SNR (16.71 dB) than in the DBBD case, with a wide variation in performance across trials (Min SNR of -1.58 dB, Max SNR of 34.47 dB). The random nature of the unscaled binary matrix seems to be less compatible with fixed dictionaries like DCT, resulting in poor performance and instability in the signal reconstruction.

Applying the Kronecker technique (DCT-KRON) improves the average SNR slightly to 17.62 dB, but the range of SNR values remains large, and negative SNR values (-1.44 dB) indicate that the Kronecker technique cannot fully compensate for the challenges posed by the unscaled binary measurement matrix when used with DCT.

MOD Performance Analysis The MOD method exhibits a much higher average SNR of 35.30 dB, indicating that adaptive dictionary learning methods are better suited for random measurement matrices compared to DCT. The improved performance reflects MOD's ability to learn a more effective sparse representation, even in a random measurement matrix scenario.

When the Kronecker technique is applied (MOD-KRON), the average SNR rises to 38.67 dB, with an increased minimum SNR of 31.73 dB and a maximum of 45.04 dB. The more consistent performance of MOD-KRON across trials suggests that the Kronecker technique significantly enhances the reconstruction quality when paired with adaptive dictionary learning methods.

KSVD Performance Analysis KSVD performs reasonably well, with an average SNR of 32.54 dB, showing similar behavior to MOD but with slightly less stability (Min SNR of 17.40 dB and Max SNR of 40.08 dB). The variation in SNR highlights the non-deterministic nature of KSVD, where the iterative learning process may result in some variability in the dictionary's sparse representation ability.

Applying the Kronecker technique (KSVD-KRON) improves the average SNR to 37.65 dB, with the minimum SNR increasing to 31.39 dB and the maximum reaching 43.07 dB. Similar to MOD, the Kronecker technique stabilizes and boosts the performance of KSVD, making it a viable option when paired with random measurement matrices.

Impact of the Kronecker Technique The Kronecker technique demonstrates a consistent improvement in SNR across all methods. For DCT, while the improvement is modest, the average SNR still benefits slightly from the Kronecker product. For the adaptive dictionary learning methods (MOD and KSVD), the Kronecker technique delivers a more substantial boost in performance, with MOD-KRON achieving the highest overall SNR of 38.67 dB.

The improvement in SNR when applying the Kronecker technique to MOD and KSVD suggests that this approach is particularly useful in random measurement matrix scenarios, helping to stabilize the reconstruction performance and reduce variability.

Conclusion In contrast to the DBBD case, adaptive dictionary learning methods (MOD and KSVD) significantly outperform the fixed DCT dictionary when using a random unscaled binary measurement matrix. The Kronecker technique further enhances the performance of both MOD and KSVD, pushing their SNR values close to 40 dB.

While DCT still benefits from the Kronecker product, its performance lags behind the adaptive methods, indicating that in scenarios involving random measurement matrices, adaptive dictionary learning particularly when combined with the Kronecker technique—offers a clear advantage in terms of signal reconstruction quality.

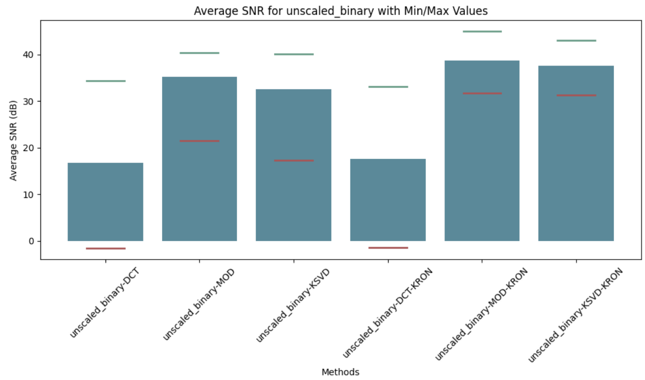
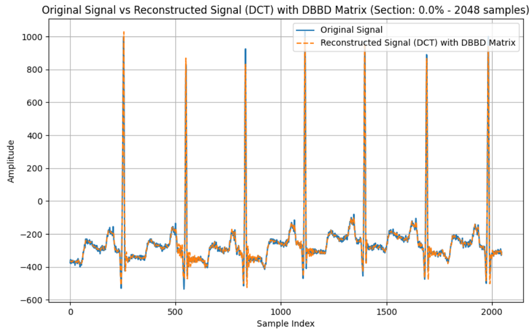
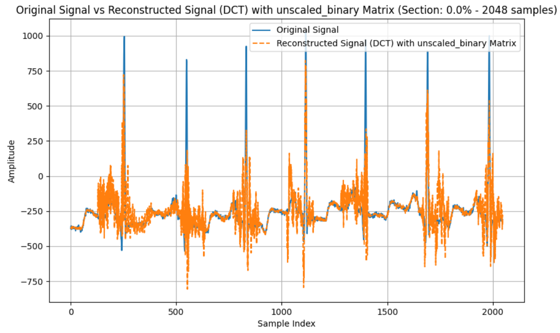


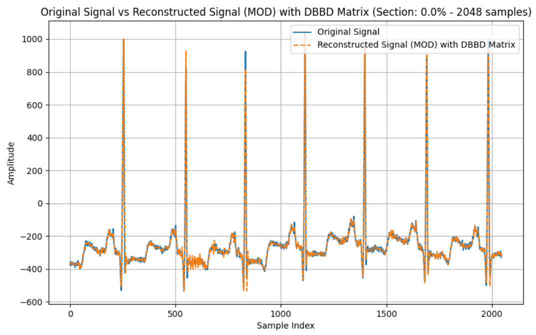
Figure 3: This graph visually represents the comparison between the different methods (DCT, MOD, KSVD) and the effect of applying the Kronecker technique when an unscaled binary random measurement matrix is used.



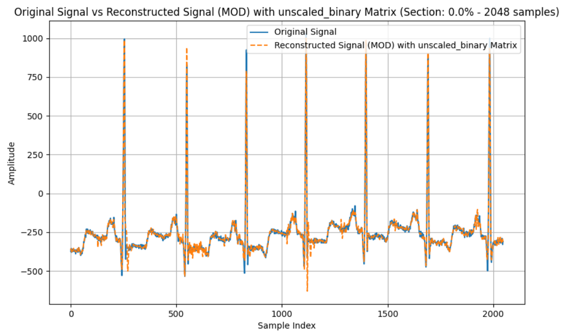
(a) DBBD - DCT



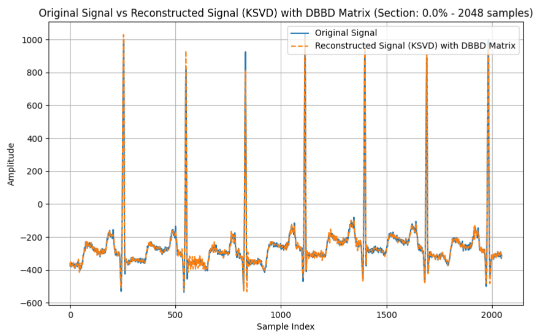
(b) Binary - DCT



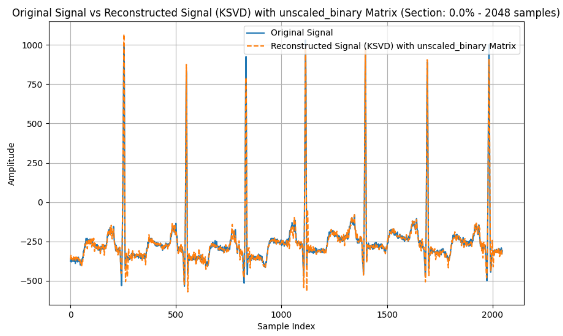
(c) DBBD - MOD



(d) Binary - MOD



(e) DBBD - KSVD



(f) Binary - KSVD

Figure 4: Comparison of signal reconstruction methods using the Kronecker technique. All signals are recovered from record 100 of the MIT-BIH Arrhythmia Database, illustrating the performance of different dictionary learning methods under consistent conditions.

4 Conclusions

This study demonstrates that compressed sensing offers a viable solution for reducing the memory and computational load on portable ECG devices, especially when real-time long-term monitoring is required. Our investigation reveals that different combinations of measurement matrices and dictionaries impact both the compression efficiency and signal reconstruction quality.

When **deterministic Block Diagonal Binary (DBBD) measurement matrices** are used, **fixed dictionaries** like DCT exhibit superior performance in terms of Signal-to-Noise Ratio (SNR), even when compared to adaptive dictionary learning methods such as MOD and KSVD. This result suggests that, in deterministic scenarios, the simplicity and stability of fixed dictionaries are sufficient, and the added complexity of adaptive methods may not be justified. However, the **Kronecker technique** provides a notable improvement in performance across all methods, demonstrating that it can further enhance compression without sacrificing signal quality.

In contrast, when **random unscaled binary measurement matrices** are employed, adaptive dictionary learning methods like MOD and KSVD significantly outperform fixed dictionaries such as DCT. This is likely due to the flexibility of adaptive methods in adjusting to the variability of random matrices, which fixed dictionaries cannot fully exploit. The **Kronecker technique** also proves effective in this scenario, improving the reconstruction quality for both MOD and KSVD. Particularly, MOD combined with the Kronecker technique consistently achieves the highest SNR, suggesting that adaptive dictionaries, when coupled with structured techniques like the Kronecker product, can handle randomness in the measurement process more effectively.

Future Research Directions:

To build on this work, future studies should focus on designing optimized measurement matrices that are inherently well-matched with adaptive dictionaries. Further exploration of the interplay between measurement matrix coherence and dictionary sparsity could also yield new strategies for improving compression efficiency. Additionally, as compressed sensing continues to evolve, efforts should aim to balance performance and complexity, ensuring that these advancements remain compatible with the limited resources of portable ECG devices.

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[2] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[3] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 5, pages 2443–2446. IEEE, 1999.

[4] Vahi Izadi, Pouria Karimi Shahri, and Hamed Ahani. A compressed-sensing-based compressor for ecg. *Biomedical engineering letters*, 10(2):299–307, 2020.

[5] Hosein Mohimani, Massoud Babaie-Zadeh, and Christian Jutten. A fast approach for overcomplete sparse decomposition based on smoothed ℓ_0 norm. *IEEE Transactions on Signal Processing*, 57(1):289–301, 2008.

[6] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[7] H Zanddizari, S Rajan, and Houman Zarrabi. Increasing the quality of reconstructed signal in compressive sensing utilizing kronecker technique. *Biomedical engineering letters*, 8(2):239–247, 2018.

Appendix

A Kronecker Product

If B is a $p \times q$ matrix and A is an $m \times n$ matrix, then the *Kronecker product* of B and A is a $pm \times qn$ matrix, defined as:

$$B_{p\times q} \otimes A_{m\times n} = \begin{bmatrix} b_{11}A & b_{12}A & \cdots & b_{1q}A \\ b_{21}A & b_{22}A & \cdots & b_{2q}A \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1}A & b_{p2}A & \cdots & b_{pq}A \end{bmatrix}$$

For a special case, when B is an identity matrix $I_{p\times p}$—a square matrix with ones on the main diagonal and zeros elsewhere—the Kronecker product is:

$$I_{p\times p} \otimes A_{m\times n} = \begin{bmatrix} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A \end{bmatrix}$$

Restricted Isometry Property (RIP) and Kronecker Product

Orthonormal matrices such as the identity matrix have an RIP (Restricted Isometry Property) constant $\delta_K = 0$. Duarte et al. have proven that the RIP constant for the new measurement matrix formed via the Kronecker product obeys the following inequality:

$$\delta_K(A_1 \otimes A_2 \otimes \cdots \otimes A_I) \leq \prod_{i=1}^{I}(1 + \delta_K(A_i)) - 1$$

where $\delta_K(A_i)$ is the RIP constant of matrix A_i.

A smaller RIP constant indicates a more efficient matrix for sensing. In the equation above, the output RIP constant does not exceed that of the original matrix. While the Kronecker product cannot be guaranteed to improve the RIP constant, it is confirmed that it will not negatively impact the RIP constant of the resulting matrix. The simplified inequality for the specific case of the identity matrix I and a general matrix A is as follows:

$$\delta_K(I \otimes A) \leq (1 + \delta_K(I))(1 + \delta_K(A)) - 1$$
$$\delta_K(I \otimes A) = \delta_K(A)$$

Thus, the RIP constant of the Kronecker product does not increase beyond the RIP constant of the original matrix.

B Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) expresses a signal as a sum of cosine functions oscillating at different frequencies, concentrating most of the signal's energy in a few low-frequency components, which makes it highly efficient for compression.

The most commonly used variant is DCT-II. For a sequence of N real numbers $x[n]$, where $n = 0, 1, \ldots, N-1$, the DCT-II is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \quad \text{for} \quad k = 0, 1, \ldots, N-1$$

The inverse DCT-II (IDCT) reconstructs the signal from the frequency-domain coefficients as follows:

$$x[n] = \frac{1}{N}\left(\frac{X[0]}{2} + \sum_{k=1}^{N-1} X[k] \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]\right) \quad \text{for} \quad n = 0, 1$$

The DCT has several key properties that make it particularly suitable for compression:

- **Energy Compaction**: Most of the signal's energy is concentrated in a few low-frequency components, which makes it possible to discard or coarsely quantize the higher-frequency components without significant quality loss.
- **Orthogonality**: The cosine basis functions used in the DCT are orthogonal, meaning that the different frequency components do not overlap.
- **Real-Valued Output**: For real-valued input signals, the DCT produces real-valued output, which simplifies computations.

The DCT can be represented in matrix form. Let $\mathbf{x}$ be the input signal, a column vector of length N. The DCT of this signal can be written as:

$$\mathbf{X} = \mathbf{\Psi}\mathbf{x}$$

where $\mathbf{\Psi}$ is the $N \times N$ DCT matrix with elements defined as:

$$\mathbf{\Psi}[k, n] = \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \quad \text{for} \quad k, n = 0, 1, \ldots, N-1$$

The matrix $\mathbf{\Psi}$ is orthogonal, satisfying $\mathbf{\Psi}^\top\mathbf{\Psi} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. This ensures that the DCT and IDCT operations perfectly invert each other, preserving the original signal's energy.