Data Compression based on Discrete Cosine Transform

Discrete Cosine Transform (DCT) is a widely used transformation technique that can transform time-series data or an image to its frequency domain and have a compact representation of the signal. After DCT, a small fraction of the data points contains most of the signal information and the rest of the data points in the frequency domain have very small values which can be removed without causing significant distortion in the reconstructed signal. Therefore, a sensor node only needs to send the data points with large values and achieve the goal of data compression.

DCT-II transform of a length L signal x is given as,

$$y_k = \sqrt{\frac{2-\delta(k)}{L}} \sum_{n=0}^{L-1} x_n \cos\left(\frac{\pi}{L}(n+\frac{1}{2})k\right), \tag{1}$$

where y_k is the k^{th} DCT coefficient of the signal, $\mathbf{x}$, for a particular k. The function $\delta(k)$ is given as $\delta(k) = \begin{cases} 1 & k = 0, \\ 0 & elsewhere. \end{cases}$

DCT transformation can also be rewritten in the matrix form as,

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \tag{2}$$

where $\mathbf{H} \in \mathcal{R}^{L\times L}$ is a DCT matrix.

The signal reconstruction process is as follows

$$\mathbf{x} = \mathbf{H}^{-1}\mathbf{y}, \tag{3}$$

where $\mathbf{H}^{-1} \in \mathcal{R}^{L\times L}$ is the inverse DCT matrix.

The data that we received from this assignment can be plotted as:

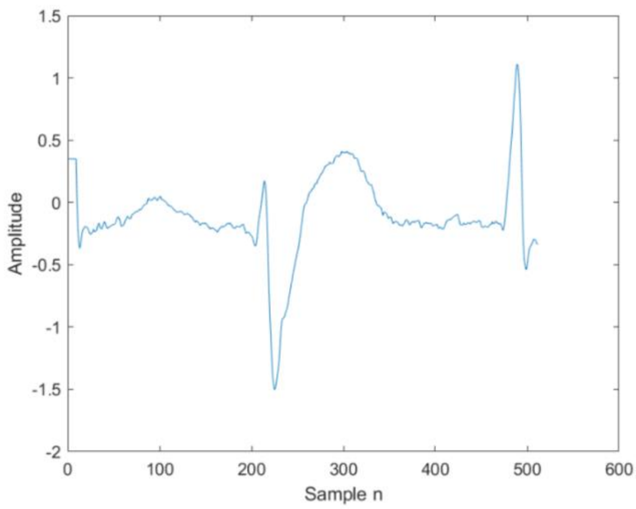


Fig 1: Original signal

Study findings

The study part of the assignment is to implement the DCT matrix, and run it on the TelosB mote, to test the processing time. We will try to see the difference in processing time, when we try different values for L, and see the difference in reconstructed signal quality, when trying different values for M. For results in the study, see the report.

To reconstruct the signal, we used an existing implementation in MATLAB.

Two different cases have been studied to see the difference in L's and M's.

Case 1L:

N = 512 & L = 8

Processing time in the mote = 2 seconds

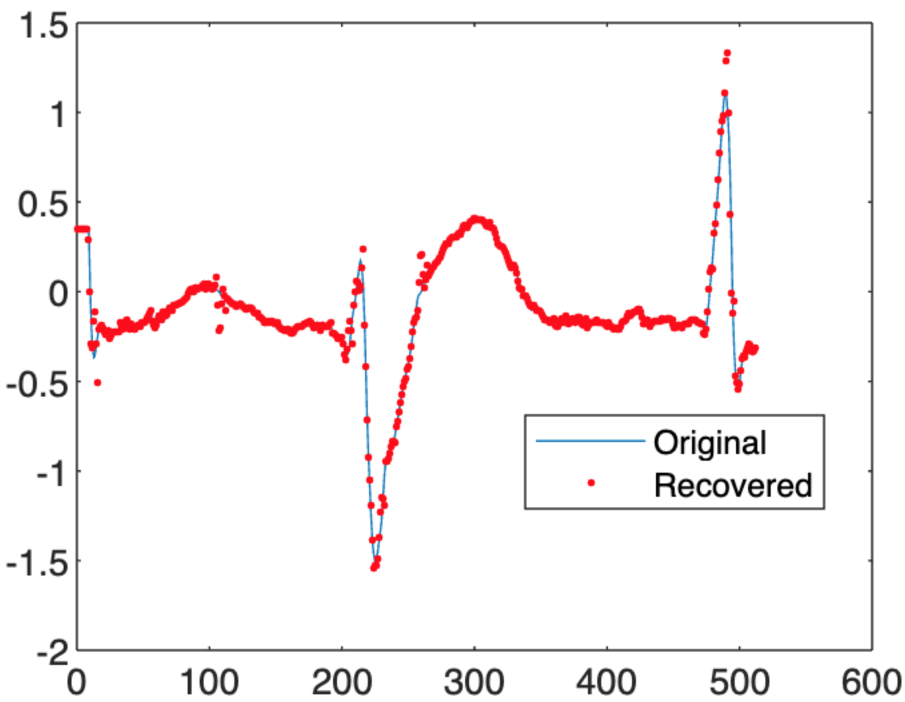
Case 2L:

N = 512 & L = 16

Processing time in the mote = 4 seconds

Case 1M:

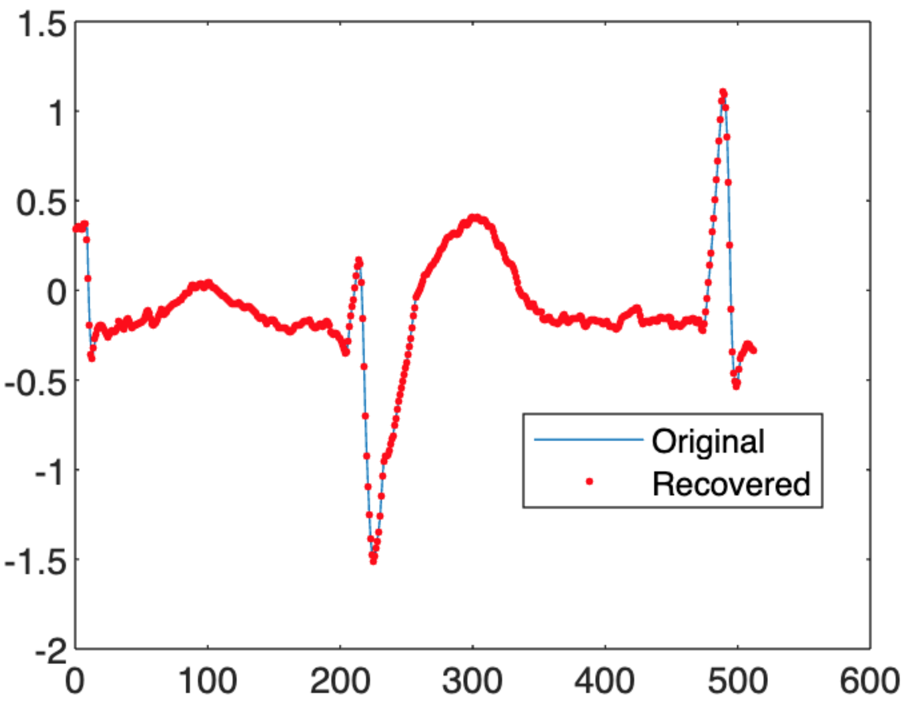
N = 512 & L = 8 & M = 4



Plots look very nice, very close to the original

Case 2M:

N = 512 & L = 16 & M = 8



Plot looks even better, almost no points differ from the original.

Conclusion on study

L has an impact on the processing time. The higher L, the higher process time. In our example the processing time took twice as long for the mote to encode the signal.

M has an impact on the precision of the reconstructed signal quality. In our example the higher the M, the more precise the plotting becomes.