Quantum Speech Representation- QPAM

Rohan Biswas

January 21, 2025

Introduction

We know that the state of a system with n qubits is described by a vector in a 2^n-dimensional complex vector space, represented as:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle \tag{1}$$

Here:

- $|i\rangle$ are the computational basis states ($|0\rangle, |1\rangle, |10\rangle, |11\rangle$, etc.).
- $c_i \in \mathbb{C}$ are complex coefficients such that:

$$\sum_{i=0}^{2^n-1} |c_i|^2 = 1 \tag{2}$$

A quantum register is used in audio representation to create a superposition of all possible time indices for the audio samples. This register, called the time register, represents each time position as a quantum state. Each state acts as an index, marking a specific point in time, similar to a classical register. The corresponding audio samples are encoded in these states using different methods, one of which we are going to be discussing here.

Quantum Speech representations can be broadly divided into two parts : Probabilistic (Coefficient based), and Deterministic (State based).

Coefficient-Based Representation

The coefficient-based quantum audio representation is probabilistic in nature. In this representation, the amplitude coefficients of the quantum state are used to encode the audio information. The audio signal $x(t)$ is discretized into N samples, represented as $\{x_0, x_1, \ldots, x_{N-1}\}$. The corresponding quantum state can be expressed same as that shown in equation (1), but here

- $|i\rangle$: The computational basis state corresponding to the i-th time index.
- $c_i \in \mathbb{C}$: The amplitude coefficient encoding the i-th audio sample x_i.

For real-valued audio signals, the coefficients c_i are defined as:

$$c_i = \frac{x_i}{\sqrt{\sum_{j=0}^{N-1} x_j^2}}$$

State based Representation

In the state-based representation, the audio samples are directly encoded as separate quantum registers. While quantum mechanics underpins this representation, the encoding itself does not rely on the probabilistic nature of the quantum amplitudes. Instead, it uses the states explicitly to represent time indices and sample values. The probabilistic aspect only arises during measurement, as with any quantum system, but it is not inherent to how the information is represented.

Let $x(t)$ again be discretized into N samples. The quantum state however will now be expressed as :

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle |x_i\rangle \tag{3}$$

Here:

- $|i\rangle$: The computational basis state of the time register, encoding the time index i.
- $|x_i\rangle$: An additional register encoding the audio sample value x_i.

This representation uses multiple quantum systems, allowing the encoding of both time and amplitude information in distinct subsystems. For binary encoding of the sample values x_i, the state $|x_i\rangle$ can be further decomposed into :

$$|x_i\rangle = |b_{i,0}\rangle \otimes |b_{i,1}\rangle \otimes \cdots \otimes |b_{i,m-1}\rangle$$

where $b_{i,j}$ are the binary bits of the quantized sample value x_i, and m is the number of qubits used for the amplitude register.

Quantum Probability Amplitude Modulation

It is a type of Coefficient based representation. The amplitude information is encoded as probability amplitudes of each quantum state, and the quantum audio of size N with n or ($\lceil log \rceil$) qubits is shown by equation 1. But let's just assume that the c_i be represented by α_i cause we need convenience to know that its indeed the probability "amplitude" of audio.

Theoretically, how do we actually do it ?

- Step 1 : **Mapping digital amplitudes for QPAM**. Now, the digital audio amplitide $\in [-1, 1]$ and sum $\neq 1$. So we try to normalize.

  $$\alpha_i = \frac{1}{\sqrt{g}} \sqrt{\frac{(a_i + 1)}{2}} \ ; \ g = \sum_k \frac{(a_k + 1)}{2}$$

  Here, a_i are the audio amplitudes, and α_i are the probability amplitudes.
  Note that α_i has values between 0 and 1.

- Step 2 : **QPAM Preparation**. Now that we have the probability amplitudes we initialize qubits to a desired superposition, and we map the set of qubits initialized in $|0000 \cdots\rangle$ state into an arbitrary state.

- Step 3 : **QPAM retrieval**. Measurement is required. Retrieving audio requires that we prepare many identical quantum versions of audio and make statistical analysis of result of all measurements. Analysis will give us a histogram from which when considering how the system was prepared in first place, it indicates that histogram itself is scaled and shifted version of digital audio.

  We get back reconstructed audio using :

$$a_i = 2g|\alpha_i|^2 - 1$$

  Note that $|\alpha|^2 \neq \alpha^2$ since $p_i = |\alpha|^2$ of $|i\rangle$. So

$$a_i = 2g\bar{p_i} - 1$$

  where g for output would be :

$$g = \sum_k \bar{p_k}$$

What I actually did ?

I followed these steps in my implementation of QPAM :

1. Import an audio, and normalized it to range [-1,1].
2. Mapped the digital amplitudes obtained to qubits for QPAM.
3. Initialized into an arbitrary state and transpiled. Note : The more is the number of counts, the better will be the result.
4. Reconstructed the audio amplitudes using the step 3 mentioned above.

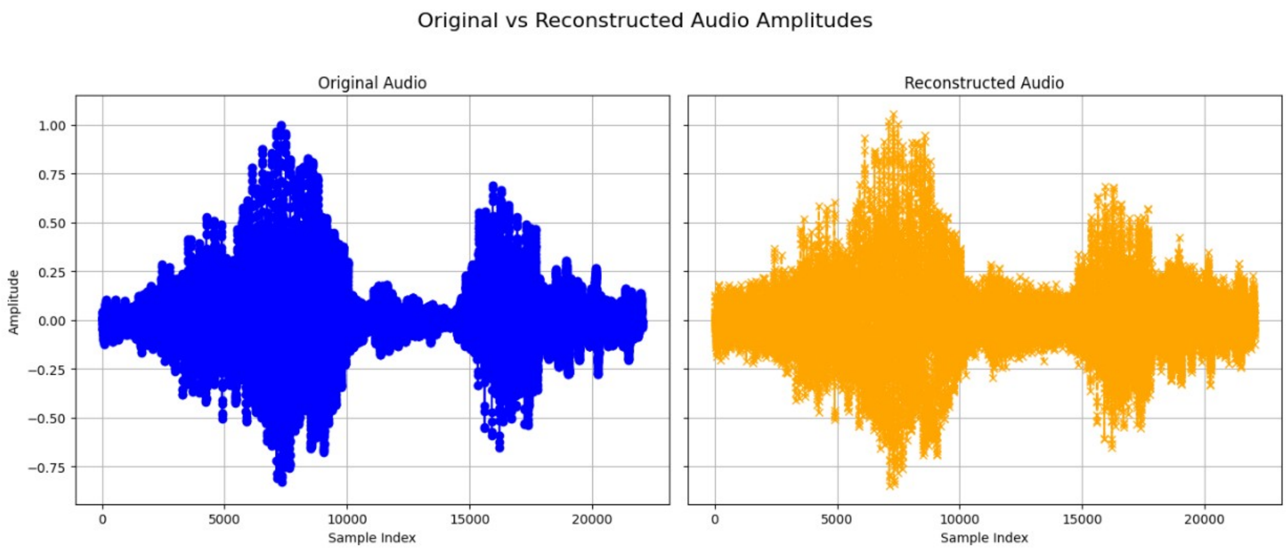

Figure 1: The audio had the message "Happy". As seen the Original and Reconstructed versions are almost similar with little fluctuations in the reconstructed one because of measurement noise, interpolation or other issues.

5. Did something Extra. Retrieved the audio and played it. I used Savitzky-Golay filter for noise reduction.

You may find my code in the GitHub Repository : https://github.com/RohanBiswas67/QPAM