SIGNAL ANALYSIS-SYNTHESIS USING THE QUANTUM FOURIER TRANSFORM

Aradhita Sharma, Glen Uehara, Vivek Narayanaswamy, Leslie Miller, Andreas Spanias
SenSIP Center, School of ECEE, Arizona State University, USA
Email: {ashar314, guehara, vnaray29, lmille42, spanias}@asu.edu

ABSTRACT

This paper presents the development of Quantum Fourier transform (QFT) education tools in the object-oriented Java-DSP (J-DSP) simulation environment. More specifically, QFT and Inverse QFT (IQFT) user-friendly J-DSP functions are developed to expose undergraduate students to quantum computing. These functions provide opportunities to examine QFT resolution, precision (qubits), and the effects of quantum measurement noise. In our study, we also describe a laboratory exercise on QFT-based speech analysis-synthesis which has been deployed in our senior-level DSP class and in our NSF workforce development programs. The software and the laboratory exercise are evaluated using formative and summative assessments.

Index Terms— Java-DSP, quantum Fourier transform, qubits, speech analysis-synthesis, quantum noise.

1. INTRODUCTION

Quantum computing (QC) promises to process data with estimated speeds exceeding 100 million times relative to classical computers [1]. This motivated several recent QC education, training, and research efforts. In fact, presidential committee announcements [2] emphasized the need for training and workforce development in undergraduate and even in K-12 classrooms [3]. In addition, the National Quantum Initiative Act argued on "expanding the number of researchers, educators, and students with training in quantum information science" [2]. Although physics departments offered quantum mechanics courses in undergraduate programs, QC for signal processing is cross-disciplinary and requires workforce training that draws content from engineering [4] and other STEM fields. Several education initiatives have been developed to create QC awareness and began training students in this area [5]. The main challenge is access to quantum computers which is currently limited and expensive. Nevertheless, several companies including Amazon, Google, Honeywell, IBM, Microsoft, and Quantinuum [6] have created quantum simulators that can be accessed by students, researchers, and practitioners. Training undergraduate students to use simulators is difficult to accomplish within semester-long signals and systems-related courses. For this reason, we started creating user-friendly QC tools within a signal-processing computing environment. Previously, we created DSP tools to train students in filter design, linear prediction, speech processing, and machine learning (ML) [7-13].

QC relies on fundamental quantum properties, including *superposition, entanglement,* and *interference* [14,15]. There are currently several university programs that created QC graduate courses [16]. Undergraduate (UG) experiences in QC were also reported in [17-19]. There were also efforts to develop QC experiences in high school honor classes [20]. In addition to university efforts, industry launched programs for quantum coding [21,22]. For example, IBM launched the *Qiskit* Global Summer School to familiarize users with quantum simulators [23].

This paper describes a user-friendly software platform to introduce QC concepts to UG students in Digital Signal Processing (DSP) classes. We describe interactive software functions and web-based exercises for UG students. More specifically, we present software for the QFT and IQFT and we embed these functions in our object-oriented Java-DSP simulation environment [7] at Arizona State University (ASU). The user-friendly J-DSP environment has been used for online laboratories in UG signal processing courses [10]. This freely accessible program enables students to create and run DSP simulations on the web. J-DSP includes several functions that support speech processing and ML [8-13] simulations. In this project, we expand the J-DSP functionality by developing from the ground up, user-friendly QC simulation functions for Fourier transforms. These functions are intended to provide hands-on laboratory experiences within the DSP class. The benefit of J-DSP quantum functions is that the students can obtain experiences on quantum precision, quantum noise, and quantum signal processing in a relatively short time without having to learn an entirely new programming environment.

We focus on signal analysis-synthesis using the QFT and IQFT where students simulate a simple compression scheme based on peak picking [24-26]. Comparisons with FFT and inverse FFT are possible and enable students to observe potential problems with qubit measurement errors and quantum noise. To provide a general background of QC concepts, we developed a module on the basics of QC. This includes qualitative coverage of basic quantum principles, examples of quantum gates and simple quantum circuits for QFT implementation using Qiskit, and an introduction to quantum noise. In addition, the module covers simple QFT simulations using J-DSP. The rest of the paper is organized as follows. Section 2 discusses QC basics and the QFT circuit design. In Sections 3 and 4, we describe new quantum J-DSP functions. Sections 5 and 6 describe the exercise for an UG class and preliminary evaluation results. Section 7 provides our conclusions.

2. QUANTUM CIRCUIT BASICS

2.1. Qubits, quantum states, and quantum gates

Analogous to a binary bit in classical computing, a qubit is the fundamental unit of information in QC. However, unlike a classical bit which can be either be 0 or 1, a quantum bit can exist in a superposition of both $|0\rangle$ and $|1\rangle$ at the same time, expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|\psi\rangle$ is any arbitrary state [27] and $|\alpha|^2$ and $|\beta|^2$ are the probabilities of observing the qubit in $|0\rangle$ and $|1\rangle$ state respectively. Once a quantum state is measured, it collapses and transitions into computational states due to the interference property of the quantum system. Quantum operations are represented using unitary transformations acting on qubits and are called quantum gates which form a quantum circuit. A Hadamard gate (H) is used to create a superposition. A rotation gate (R_k) is used to shift the phase of basis states by a factor of $e^{2\pi i/2^k}$ for the k[th] qubit, where $i = \sqrt{-1}$. The rotation of one qubit can be controlled depending on the control state using a controlled rotation gate (R). A swap gate (SWAP) is used to swap the states of two qubits [27]. The matrix and geometric representations of the gates and the quantum state respectively (Bloch sphere) are shown in Figure 1.

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},\ R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix},\ R = \begin{pmatrix} I & 0 \\ 0 & R_k \end{pmatrix}$$

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

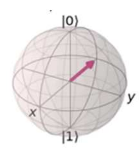

Figure 1: Bloch sphere and matrix representation of gates.

2.2. The Quantum Fourier transform

QFT can be viewed as the quantum realization of the discrete Fourier transform (DFT). The basics of QFT are covered in [27-29] and applications are presented in [29-32]. Consider a state vector $|\psi\rangle = \sum_{n=0}^{N-1} h_n|n\rangle$ which represents the signal, where h_n are the amplitudes associated with N basis states $|n\rangle$. The binary expression of qubits is

$$n = n_0 2^0 + n_1 2^1 + n_2 2^2 + \cdots + n_{m-1} 2^{m-1} \tag{1}$$

where, m is the number of qubits forming an $N = 2^m$ dimensional system. Fractional and integral terms can be written exponentially as

$$e^{\frac{2\pi i n}{2^k}} = e^{2\pi i\left(\frac{n_0}{2^k} + \cdots + \frac{n_{k-1}}{2}\right) + 2\pi i\left(n_k + \cdots + n_{m-1}2^{m-k-1}\right)} \tag{2}$$

The integral part corresponding to $e^{\frac{2\pi i n}{2^k}}$ is equal to 1. Hence, the equation can be written as product of k factors as

$$e^{\frac{2\pi i n}{2^k}} = e^{2\pi i\left(\frac{n_0}{2^k}\right)}.e^{2\pi i\left(\frac{n_1}{2^{k-1}}\right)} \ldots e^{2\pi i\left(\frac{n_{k-1}}{2}\right)} \tag{3}$$

Quantum Fourier transform for $|\psi\rangle$ state can be written as

$$QFT|\psi\rangle \rightarrow \sum_{k=0}^{N-1} b_k\,|k\rangle \tag{4}$$

where,

$$b_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} h_n.\, e^{2\pi i k n/N} \tag{5}$$

From equation (4) and (5), we obtain the QFT for $|\psi\rangle$ state as

$$QFT|\psi\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} h_n.\, e^{2\pi i k n/N}\,|k\rangle \tag{6}$$

where, $\frac{1}{\sqrt{N}}$ is the normalizing factor. Considering $|n\rangle$ state of the state vector $|\psi\rangle$ and calculating its QFT gives us

$$QFT|n\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i k n/N}\,|k\rangle \tag{7}$$

where, $|n\rangle = |n_{m-1} \ldots n_1 n_0\rangle$ and $|k\rangle = |k_{m-1} \ldots k_1 k_0\rangle$ is their binary representation. The above equation is similar to DFT with the opposite sign of the phase exponent. Writing the equation as tensor products of qubit's basis states [27] to represent QFT in a compact manner as

$$QFT|n\rangle \rightarrow \frac{1}{\sqrt{2^m}}\left(|0\rangle + e^{2\pi i n_0}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i n_0 n_1}|1\rangle\right) \ldots$$
$$\ldots \otimes \left(|0\rangle + e^{2\pi i n_0 n_1 n_2 \ldots n_{m-1}}|1\rangle\right) \tag{8}$$

This tensor product state is obtained using the controlled rotation of the basis states in superposition. Basis states can be expressed as a tensor product that enables unitary transformation implementations for creating a quantum circuit. The inverse QFT (IQFT) of a state $|k\rangle$ is given by

$$IQFT|k\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-2\pi i k n/N}\,|n\rangle \tag{9}$$

2.3. QFT Circuit Design

For implementing the QFT, a circuit is constructed with the combination of a single-qubit H gate, a two-qubit R gate represented with purple lines between qubits as U(angle of rotation), and a SWAP gate represented as a blue line. The gates are separated with black measurement blocks using a barrier (gray line) at the end of the circuit (Figure 2).

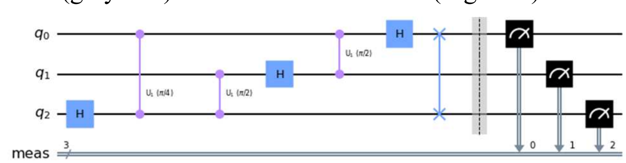

Figure 2: QFT circuit for 3 qubits after measurement.

It is seen that a total of $\frac{m(m+1)}{2}$ H and R gates, and at most $\frac{m}{2}$ SWAP gates are required for the QFT circuit. Hence, the complexity is $O(m^2)$ where m is the number of qubits. An exponential speedup is observed in the QFT due to quantum parallelism achieved by the entanglement and superposition property [33,34]. The 3-qubit IQFT circuit can be developed by modifying the QFT [35] (Figure 3).

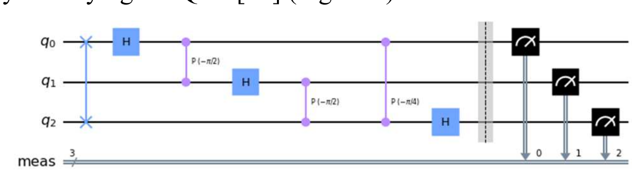

Figure 3: IQFT circuit for 3 qubits after measurement.

2.4. Quantum Noise Models

Quantum computers are susceptible to noise from various sources such as external environmental interactions, crosstalk caused by the neighboring qubits when excited by lasers, and quantum implementation errors. Other sources of interference are reported in [36]. To consider how the algorithm will behave in real devices, quantum error models are introduced to perform noisy simulations. We use three types of quantum errors which represent the effect of decoherence on qubits [37,38]. They are namely,

- Amplitude damping channel occurs as an effect of energy dissipation from the quantum system.
- Phase damping channel represents the loss of partial information from quantum relative phases [27].
- Measurement error occurs during the final state and measures the output that differs by 1 bit, or 2 bits [37].

3. DESIGNING THE J-DSP FUNCTIONS FOR QFT

The initial version of J-DSP was deployed in the early 2000s to provide remote online learning laboratory experiences [8]. Since then, several new functions have been added. The new HTML5 J-DSP (2017) embeds complex operations [12] including the proposed QFT functions (2022).

3.1. Quantum Computing J-DSP Simulation Functions

We developed an initial QC suite of blocks in J-DSP for quantum signal analysis and synthesis which are described in this section. This suite will be enriched with additional quantum functions later to accommodate other applications, such as machine learning.

3.1.1 The Quantum Fourier Transform Block

A QFT block is included in the J-DSP software which takes signal values as input and returns the QFT coefficients as the output. The input signal is scaled using a vector norm such that the sum of squared amplitudes equals 1, and the signal is represented as a quantum state. The QFT of this quantum state is evaluated according to equation (6) as

$$QFT|\psi\rangle = p_0|n_0\rangle + p_1|n_1\rangle + \cdots + p_{N-1}|n_{N-1}\rangle \quad (10)$$

where p_k is the wavefunction value, and ${p_k}^2$ corresponds to the probabilistic values [36] associated with the quantum basis, with $\sum_{k=0}^{N-1} {p_k}^2 = 1$. A model of possible outcomes is constructed by utilizing the probability distribution using Monte Carlo simulation to replicate the quantum measurement [40]. The final QFT coefficients are represented in Euler's form including the probability values and the phase information obtained from the wavefunction.

$$QFT_k = \sqrt{{p_k}^2} \, . \tan^{-1}\left(\frac{Im(p_k)}{Re(p_k)}\right) \quad (11)$$

These coefficients are the output of the QFT block.

3.1.2 The Inverse Quantum Fourier Transform Block

A function block called IQFT is a part of the J-DSP QC architecture which takes the QFT coefficients as input and scales them using the vector norm to be represented as a quantum state. The IQFT of the quantum state is evaluated by equation (9). Quantum measurement is performed using Monte Carlo simulation and the IQFT output is calculated similarly to the QFT. The two blocks are shown in Figure 4.

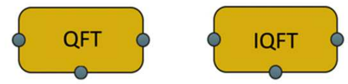

Figure 4:The QFT and IQFT blocks for computing up to 10 qubits.

3.1.3 Normalization Transform Block

Since the quantum measurement results in probabilistic values, there is a need to normalize these values to the desired range [41]. Therefore, the QFT and IQFT coefficients acquired from their respective blocks need to be normalized using the normalization block (Figure 5). The normalization factors for QFT and IQFT are

$$QFT\ Factor = \sqrt{2^n} * vector\ norm \quad (12)$$

$$IQFT\ Factor = vector\ norm / \sqrt{2^n} \quad (13)$$

3.1.4 Quantum Noise Block

A quantum noise block is included in J-DSP QC blocks to demonstrate the effects of noise in QC (Figure 5), which ultimately affects the quantum states. We introduce amplitude damping, phase damping, and measurement error. The types and percentage of noise to be introduced in the quantum circuit can be selected using this block.

- To simulate amplitude damping, we introduced randomness in the wave functions of the quantum state.
- To simulate phase damping, we introduced random phase rotations $\pm e^{-\theta}$ in the quantum state.
- To simulate measurement error, we randomized probability outcomes corresponding to basis state differing by 1 or 2 bits [37] using Monte Carlo simulation.

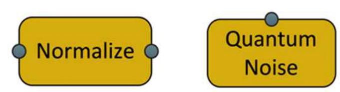

Figure 5: Normalize and Quantum Noise block in J-DSP.

3.2. Signal Analysis-Synthesis using the QFT and IQFT

We can analyze the signal [24] in the quantum domain by decomposing it into its quantum components obtained through the QFT function. Quantum noise can be added to observe the effects on the QFT algorithm. Similarly, the IQFT is used to synthesize the signal and the time domain values are computed. The signal analysis-synthesis block diagram without peak-picking is shown in Figure 6.

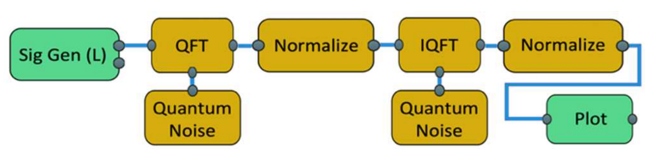

Figure 6: J-DSP Signal analysis-synthesis with the QFT and IQFT.

4. QFT AND DFT PERFORMANCE WITH NOISE

In QFT and IQFT computation, quantum noise affects the spectral output. Because of quantum noise, the QFT spectrum components are not as precise as those obtained using the DFT. To compare QFT and DFT numerical output, we evaluate the magnitude difference between QFT and DFT coefficients. Comparative spectral estimation simulations were run with sinusoidal and speech segments. For this simulation, a quantum noise model with 20% amplitude damping was introduced which affected the values of the quantum basis states [27]. After we obtained the spectra for 256 frame-size speech segments, we compared the DFT and QFT components. Table 1 provides the percentage magnitude difference as a function of the number of qubits for the QFT. The error is calculated by

$$Error(\%) = \frac{avg(\,abs(QFT) - abs(DFT))}{avg(\,abs(DFT))} * 100 \quad (14)$$

We observe that as the number of qubits increases, the error increases due to quantum noise. We note that in our sinusoidal signal simulations, we did not observe a change in the frequency estimate – only in the magnitude and phase.

Table 1: Percentage Magnitude Difference between QFT and DFT.

DFT Resolution	16	32	64	128	256	512	1024
QFT qubits	4	5	6	7	8	9	10
Error (%)	4.91	6.64	7.67	8.33	10.01	15.45	22.99

5. QFT EXERCISE DESIGNED USING J-DSP

An exercise is developed to introduce the concept of QC and compare the behavior of the QFT and classical DFT for a signal analysis-synthesis task in J-DSP.

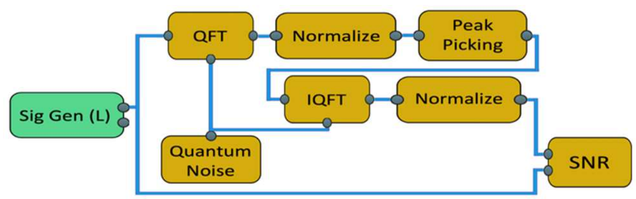

Figure 7: QFT-based simulation including quantum noise effects.

The simulation uses spectral peak-picking. Peak-picking associates with Parseval's theorem. We note that perceptual audio analysis was used in several audio compression applications [25,26]. J-DSP provides options to change the parameters of QFT signal analysis-synthesis and quantum noise. The input (speech) signal can be generated from a signal generator block, which is given as an input to the QFT function to analyze the quantum components of the signal in presence of quantum noise. The QFT output is normalized to be in the frequency domain range. For signal compression, the first L or highest L components (peaks) are selected where L depends on the desired compression ratio; the rest of the components are set to zero. This modified input is passed on to the IQFT function (with quantum noise) to reconstruct the signal. Normalization is performed to scale the reconstructed signal. At the end, the Signal-to-Noise Ratio (SNR) is calculated to compare the original and the reconstructed quantum-based signal as shown in Figure 7. The result is compared with the signal compression approach using classical DFT-based computation. Selecting the highest components of the frequency domain maximizes the SNR. As expected, the SNR increases as the number of selected components (L) is increased. It is also noticed that the SNR for quantum computation is less than that of the classical FFT. This difference increases with an increasing number of qubits because of quantum noise effects. Through this exercise, the students get exposed to: QC and QFT/ IQFT, peak-picking processes, transform-based signal compression, and quantum noise effects.

6. PRELIMINARY EVALUATION

A preliminary evaluation was implemented in the of Fall 2022. Pre- and post-quizzes were designed to determine if the participating students gained knowledge and confidence after completing the exercise. The quizzes included questions on QFT signal analysis-synthesis which covered QFT peak picking, qubit and quantum noise basics. This preliminary evaluation is used to determine the impact of introducing QC in an UG class and the impact on student knowledge of QFT basics. This evaluation (Figure 8) showed that students gained knowledge on introductory QC concepts and QFT complexity (Q.2. and Q.3.), quantum noise effects (Q.1.), the effect of the number of qubits (Q.4.), and the effect of selecting QFT components (Q.5.) in signal analysis-synthesis. The average improvement observed in the post-quiz is around 46%.

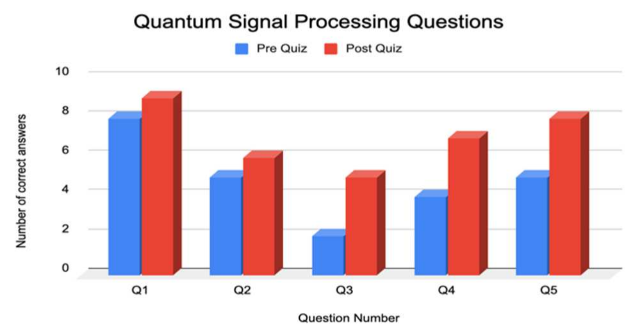

Figure 8: Graph showing the number of students who answered correctly by question number for pre- and post-quiz assessment.

7. CONCLUSION

In this paper, we presented a study on the introduction of QC and QFT concepts in an undergraduate DSP class using user-friendly visual programming software. New interactive J-DSP functions were designed for QFT-based signal analysis-synthesis. The software enabled students to experiment with QFTs, vary the number of qubits, and assess the effects of quantum noise. A J-DSP signal compression exercise was developed, and students experimented with QFT peak-picking tasks. Signal reconstruction was evaluated using the SNR function. The effects of quantum noise were also examined. A preliminary evaluation was deployed to assess the software, the exercise, and the overall knowledge gained by the students. Assessments were implemented in our 2023 classes and in our NSF workforce development programs.

8. ACKNOWLEDGEMENT

This study was supported in part by NSF awards 2215998, 1659871, 1854273, and 1953745.

9. REFERENCES

[1] Madsen, L.S et al., Laudenbach, F., Askarani, M.F., Rortais, F., Vincent, T., Bulmer, J.F., Miatto, F.M., Neuhaus, L., Helt, L.G., Collins, M.J. and Lita, A.E., 2022. Quantum computational advantage with a programmable photonic processor. *Nature*, *606*(7912), pp.75-81.

[2] The White House, Press Release, "Readout of the NQCO director's conversation with directors of nqi centers and institutes," September 17, 2021.

[3] K12, "Empowering K12 Students through online learning," https://www.k12.com/about-k12.html,

[4] Asfaw, A. et al. "Building a quantum engineering undergraduate program," *IEEE Trans. on Educ.* (2022): 220-242.

[5] Kaur, M., and A. Venegas-Gomez. "Defining the quantum workforce landscape: a review of global quantum education initiatives." *Optical Engineering* 61, (2022): 081806-081806.

[6] Deshpande, Advait. "Assessing the quantum-computing landscape." *Communications of the ACM* 65, no. 10 (2022): 57-65. /

[7] http://jdsp.engineering.asu.edu/jdsp.html

[8] Spanias, Andreas, S. Urban, A. Constantinou, M.Tampi, A. Clausen, X. Zhang, J. Foutz, and G. Stylianou, "Development and Evaluation of a Web- Based Signal and Speech Processing Laboratory for Distance Learning," *IEEE ICASSP-2000*, p. 3534, June 2000.

[9] A. Spanias, Digital Signal Processing; An Interactive Approach – 2^{nd} Edition , *Lulu Press Morrisville,* May 2014.

[10] A. Spanias and V. Atti, "Interactive On-line Undergraduate Laboratories Using J-DSP," *IEEE Trans. on Education,* no. 4, pp. 735-749, Nov. 2005.

[11] U. S. Shanthamallu, S. Rao, A. Dixit, V. S. Nrayanaswamy, J. Fan and A. Spanias, "Introducing Machine Learning in Undergraduate DSP Classes," *IEEE ICASSP 2019,* Brighton, 2019, pp. 7655-7659.

[12] A. Dixit, S. Katoch, P. Spanias, M. Banavar, H. Song and A. Spanias, "Development of signal processing online labs using HTML5 and mobile platforms," *IEEE FIE 2017, Oct. 2017*.

[13] Atti, V. and A. Spanias. "On-line simulation modules for teaching speech and audio compression techniques, " *IEEE FIE 2003*. Vol, 1, Denver, Oct. 2003.

[14] A. Khrennikov, Roots of quantum computing supremacy: superposition, entanglement, or complementarity?. *Eur. Phys. J. Spec. Top.* **230,** 1053–1057 (2021).

[15] G. S. Uehara, A. Spanias, and W. Clark. "Quantum information processing algorithms with emphasis on machine learning," *IEEE IISA* 2021, July 2021.

[16] Friedrichs, L. Carl, et al. "An analysis of quantum computing courses worldwide," *Proceedings of Seidenberg Student Faculty Research Day* (2021): 7.

[17] De Jesus, et al. "Quantum Computing: an undergraduate approach using Qiskit, " *arXiv:2101.11388* (2021).

[18] Pearson, Brett J., and David P. Jackson. "A hands-on introduction to single photos and quantum mechanics for undergraduates," *American J. of Physics* 78.5, 471-484 (2010)

[19] Perron, Justin K., et al. "Quantum undergraduate education and scientific training," *arXiv:2109.13850* (2021).

[20] L. Westfall and A. Leider, "Teaching Quantum Computing," in *Proceedings of FTC 2018*, Cham, pp. 63-80, 2019.

[21] "Qubitbyqubit.org", https://www.qubitbyqubit.org/

[22] IBM.com, "Quantum Computing Education Must Reach a Diversity of Students," https://www.ibm.com/blogs/research/2020/07/future-quantum-workforce/

[23] Qiskit.org, "Qiskit Global Summer School 2022," https://qiskit.org/events/summer-school/

[24] R. McAulay, and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. on ASSP*, *34*(4), 744-754, 1996.

[25] A. Spanias, "Speech coding: A tutorial review." Proceedings of the IEEE 82, no. 10 (1994): 1541-1582

[26] T. Quatieri, and R. McAulay, "Audio signal processing based on sinusoidal analysis/synthesis," *Applications of DSP to audio and acoustics*. Springer, Boston, MA, 2002. 343-416.

[27] Nielsen, M.A. & Chuang, I.L., 2011. Quantum Computation and Quantum Information: 10^{th} Anniversary Edition, Cambridge University Press.

[28] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *IEEE FOCS*, 1994.

[29] Kitaev, Alexei Y.. "Quantum measurements and the Abelian Stabilizer Problem," *Electron. Colloquium Comput. Complex.* TR96 (1995).

[30] R. Cleve and J. Watrous, "Fast parallel circuits for the quantum Fourier transform," *IEEE FOCS*, 2000.

[31] A. Barenco, A. Ekert, K. A. Suominen & P. Törmä, "Approximate quantum Fourier transform and decoherence," *Physical Review A*, (1996).

[32] S. Caraiman and V. Manta. "Quantum image filtering in the frequency domain," *Advances in Electrical and Computer Engineering* 13.3 (2013): 77-85.

[33] S. Zhou et al, "Quantum Fourier transform in computational basis," *Quantum Inf. Process*., (2017), vol. 16, no. 3.

[34] Musk, Damian R. "A comparison of quantum and traditional fourier transform computations," *Computing in Science & Engineering* 22.6 (2020): 103-110.

[35] Camps, Daan, Roel Van Beeumen, and Chao Yang. "Quantum Fourier transform revisited." *Numerical Linear Algebra with Applications* 28.1 (2021): e2331.

[36] Shor, Peter W. "Algorithms for quantum computation: discrete logarithms and factoring." *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994.

[37] Saki, Abdullah Ash, Mahabubul Alam, and Swaroop Ghosh. "Study of decoherence in quantum computers: A circuit-design perspective." *arXiv preprint arXiv:1904.04323* (2019).

[38] Harper, Robin, Steven T. Flammia, and Joel J. Wallman. "Efficient learning of quantum noise." *Nature Physics* 16.12 (2020): 1184-1188.

[39] R. Bayontaweepanya, "Effect of depolarizing noise on entangled photons," *Journal of Physics: Conference Series*. Vol. 1144. No. 1. IOP Publishing, 2018.

[40] S. Raychaudhuri, "Introduction to Monte Carlo simulation," 2008 *Winter Simulation Conference*, 2008, pp. 91-100.

[41] D. McMahon, *Quantum computing explained*. John Wiley & Sons, 2007.