2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC) | 978-1-6654-9005-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/VLSI-SoC54400.2022.9939656

Quantum Computing-Assisted Channel Estimation for Massive MIMO mmWave Systems

Evangelos Vlachos and Kostas Blekos
Industrial Systems Institute, ATHENA Research Center, Platani-Rio, Greece
Department of Physics, University of Patras, Rio, Greece
emails: evlachos@athenarc.gr, mplekos@physics.upatras.gr

***Abstract*—Quantum computing (QC)-assisted algorithms promise exponential increase in the computational efficiency, enabling instant solution of large systems of equations. Massive multiple-input multiple-output (MIMO) millimeter-wave (mmWave) systems pose extraordinary demands in computational complexity, due to the usage of huge antenna arrays, massive number of users, and ultra low-latency requirements. This work provides an initial discussion on how QC-based algorithms for solving linear systems of equations could be utilized for assisting the basic operations of the transceivers physical-layer, such as the channel estimation. We identify the connections between the amplitude encoding in the quantum domain and the recovery of the channel information.**

***Index Terms*—Quantum computing-assisted algorithms, physical-layer wireless communications, 6G.**

I. Introduction

The 6G networks are widely projected to provide an increase of $100\times$ in volumetric spectral and energy efficiency (in bps/Hz/m^3) relative to the current 5G networks [1]. Indeed, future wireless networks will have to accommodate massive number of users and devices, all equipped with massive number of antenna arrays. To accomodate the explosive demands, various technologies have been proposed for the design of the transceivers at the physical-layer, such as utilization of the millimeter-wave and the terrahertz spectrum, employment of massive number of antenna arrays, usage of reconfigurable intelligent surfaces, and continuous antenna apertures [2] and 3D antenna arrays. All these technologies pose different challenges for being realized in practice, however they share a commonality, the burden of *very high computational complexity*. To this end, quantum computing can play significant role to harness the potentials of 6G technologies for the design of future networks.

In the last 40 years, many scientific disciplines have converged towards the study and development of quantum algorithms and their experimental realizations. Once quantum computing becomes a commercial reality, it may be used in wireless communications systems in order to speed up specific processes due to its inherited parallelization capabilities. Quantum computing-assisted wireless communications has been first introduced by [3], where a new framework has been introduced that follows from imposing quantum mechanics interpretation of measurements, quantization and consistency. This approach is termed as *quantum computing (QC)-assisted communications* [4].

For certain problems, quantum algorithms can derive exponential speedups over their classical counterparts. In [5], a quantum algorithm is introduced to solve linear systems of equations in time which scales logarithmically in the number of the equations. In [6], an efficient solution for indoor localization in 6G scenarios has been investigated, using the Quantum Euclidean Distance, showcasing that quantum approach outperforms the classical computation if vector size is large. In [7], quantum parallelism is being exploited to for quadratic unconstraned binary optimization to minimize delay in network function virtualization scheduling [8]. In [9], the benefits of quantum computing for mmWave wireless networks for vehicle-to-vehicle communications are highlighted, where the rapidly changing environment sets new challenges for beam-control, data-feed processing, resource, and interference management.

A fundamental function of the wireless transceivers is to perform channel estimation and equalization [10]. However, due to the huge number of dimensions, this becomes very challenging problem in terms of computational complexity. For instance, given that at THz spectrum the terminals accommodate minutarized antenna arrays with hundreds of terminals, this number is multiplied by the hundreds of nodes that operate simultaneously, thus, we end up with a problem of very large dimensions. Moreover, the channel estimation procedure has to be completed within the period where the channel is seen as static. Consequently, at GHz/THz transmissions, the channel estimation has to be completed in nano/pico scale.

In this paper, we discuss on QC-assisted channel estimation for massive multiple-input multple-output (MIMO) millimeter-wave (mmWave) systems. In Section II, we overview the channel model and the fundamental estimation approaches. In Section III, first we provide short QC background information, and a brief description of basic QC algorithm for the solution of linear systems of equations. Afterwards, we identify the connections between the two problems, channel estimation and QC-based solution of a linear system of equations; complexity gains have also been discussed. Section IV makes some conclusions and states some future directions.

Notation: $\mathbf{A}$, $\mathbf{a}$ and a denote a matrix, a vector and a scalar, respectively; $\mathrm{tr}(.)$, $|.|$, $(.)^T$, $(.)^H$ and $\|.\|_F$ denote trace, determinant, transpose, complex conjugate transpose and Frobenius norm, respectively; $\mathcal{E}\{\cdot\}$ denotes the expectation operator.

978-1-6654-9005-4/22/$31.00 ©2022 IEEE

II. MmWave Massive MIMO Communications

We consider a wireless system where the transceivers are equipped with a massive amount of antenna terminals N^2 at both the transmitter (TX) and the receiver (RX), operating at the millimeter-wave (mmWave) spectrum. In order to capitilize on the full gains of massive MIMO, accurate channel state information is of paramount importance for various operations at the physical layer like equalization [11] and beamforming [12]. Depending on the communication system scenario, adopted channel models and properties, the operation of channel estimation may be expected to extract quite different types of channel state information. In the following, we outline three such cases.

A. Channel Estimation

In the following, we consider three scenarios: point-to-point, point-to-multi point, and relay-based via intelligent surfaces.

Let us consider point-to-point communication between a single TX and RX. The discrete-time channel is described with the channel matrix $\mathbf{H} \in \mathbb{C}^{N\times N}$ expressed as follows:

$$\mathbf{H}_{\text{p2p}} = \frac{N}{\sqrt{L}} \sum_{l=1}^{L} \alpha_l \mathbf{a}_{\text{RX}}(\phi_l^r, \theta_l^r) \mathbf{a}_{\text{TX}}^H(\phi_l^t, \theta_l^t), \tag{1}$$

where L is the number of propagation paths, α_l is the gain of l-th path, and

$$\mathbf{a}_{\text{TX}}(\phi_l^t, \theta_l^t) = \mathbf{a}_{\text{TX,az}}(\phi_l^t) \otimes \mathbf{a}_{\text{TX,el}}(\theta_l^t), \tag{2}$$

with $\mathbf{a}_{\text{TX,az}}(\phi_l^t) \triangleq \frac{1}{\sqrt{N_T}}[1, e^{j2\pi\frac{d}{\lambda}\phi_l^t}, ..., e^{j(N_T-1)2\pi\frac{d}{\lambda}\phi_l^t}]^T$ is the transmit steering vector following an uniform planar array (UPA) setup with ϕ_l^t the azimuth angle of departure (AoD), and respectively for $\mathbf{a}_{\text{TX,el}}(\theta_l^t)$ with θ_l^t the elevation angle of departure (AoD), d the antenna spacing and λ the wavelength. Similarly, $\mathbf{a}_{\text{RX}}(\phi_l^r, \theta_l^r)$ is the receive array response vector with ϕ_l^r the azimuth angle of arrival (AoA) and θ_l^r the elevation AoA.

For channel estimation, the TX sends a number of training symbols to the RX for each time instance t, represented by the vector $\mathbf{s}_t \in \mathbb{C}^{N\times 1}$. The received signal is given by

$$\mathbf{y}_t = \mathbf{H}\mathbf{s}_t + \mathbf{n}_t, \tag{3}$$

where $\mathbf{n}_t \sim \mathcal{CN}(\mathbf{0}, \sigma_{\text{n}}^2\mathbf{I}_N)$ is the independent and identically distributed (i.i.d.) complex additive white Gaussian noise. Collecting N received vectors $\mathbf{y}_t$, the standard Least-Squares (LS) solution of (3) is expressed as:

$$\min_{\mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{S}\|_F^2, \tag{4}$$

where $\mathbf{Y} \triangleq \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_T \end{bmatrix}^T \in \mathbb{C}^{N\times T}$, and similary for $\mathbf{S} \in \mathbb{C}^{N\times T}$. The computational cost for the LS direct solution with matrix inversion is $\mathcal{O}(N^3)$.

For K users transmitting to the single RX, the channel matrix is given as: $\mathbf{H}_{\text{p2m}} = \sum_{k=1}^{K} \mathbf{H}_k$ The K users sent their orthogonal training symbols, $\mathbf{s}_{k,t} \in \mathbb{C}^{N\times 1}$ to the RX, then the superimposed received vector is:

$$\mathbf{y}_t = \sum_{k=1}^{K} \mathbf{H}_k \mathbf{s}_{k,t} + \mathbf{n}_t = \bar{\mathbf{H}}\bar{\mathbf{s}}_t + \mathbf{n}_t, \tag{5}$$

where $\bar{\mathbf{H}} \triangleq [\mathbf{H}_1 \cdots \mathbf{H}_K] \in \mathbb{C}^{N\times NK}$ is the concatenated channel matrix, and similarly for $\bar{\mathbf{s}}_t \triangleq [\mathbf{s}_{1,t}^T \cdots \mathbf{s}_{K,t}^T]^T \in \mathbb{C}^{NK\times 1}$. In this case, the LS solution requires $\mathcal{O}(N(NK)^2)$ operations; for the scenario of massive users where $K \approx N$ the complexity goes up to $\mathcal{O}(N^4)$.

Moreover, in future, smart environments will be build by smart surfaces that will permit the dynamic reconfiguration of the propagation properties. This extends the problem of channel estimation, since each surface acts as a relay, and thus, two more channel matrices have to be recovered in order to achieve high spectral efficiency ratios.

B. Sparsity and low-rank properties

An alternative to (1) representation for $\mathbf{H}$ is based on the beamspace channel of [13], which is defined as:

$$\mathbf{H} = \mathbf{W}\mathbf{Z}\mathbf{W}^H, \tag{6}$$

where $\mathbf{W} \in \mathbb{C}^{N\times N}$is a unitary matrix based on the Discrete Fourier Transform (DFT), and $\mathbf{Z} \in \mathbb{C}^{N\times N}$ includes the virtual channel gains. Motivated by the low rank property of this matrix, we further assume that $\mathbf{Z}$ contains only few virtual channel gains with high amplitude, i.e., it is a sparse matrix. The sparsity level s of $\mathbf{Z}$ depends on the angular discretization in the beamspace representation given by (6).

C. Equalization and Beamforming

In this subsection, we briefly outline two problems, equalization and beamforming, that are strongly related with channel estimation. Equalization is employed to mitigate inter-symbol interference which occurs in time/frequency selective channels [14], [10]. For the simple case of linear minimum mean square equalization, the problem that has to be solved is stated as:

$$\min_{\mathbf{w}} \mathcal{E}\{\|s_t - \mathbf{w}_t^H \mathbf{y}_t\|_F^2\}, \tag{7}$$

for which the solution is expressed as:

$$\mathbf{w}_t = (\mathcal{E}\{\mathbf{y}_t\mathbf{y}_t^H\})^{-1}\mathcal{E}\{\mathbf{y}_t s_t^*\}, \tag{8}$$

which requires the inversion of a $N \times N$ matrix. Thus, the size of the problem depends on the size of the channel and the equalization approach. In any case, this could pose a hard to solve problem, thus, complexity efficient techniques are essential for achieving high spectral efficiency rates.

For systems with massive antenna arrays, beamforming is used to maximize spectral efficiency. For high rate communications, the digital beamforming usually requires knowledge of the channel and its singular value decomposition (SVD). Specifically, digital beamforming at the TX (i.e., precoder), requires the solution of the problem:

$$\min_{\mathbf{F}} \|\mathbf{F} - \mathbf{Q}\|_F^2 \text{ subject to } \mathbf{F}\mathbf{F}^H = \mathbf{I}, \tag{9}$$

where $\mathbf{Q}$ is the matrix with the N left singular vectors of the channel matrix.

III. Quantum-assisted Channel Estimation

Before proceeding with some basic background on quantum computing, let us first clarify the terminology; so as a *quantum computer* we consider a co-processor which extends the computational capabilities of the conventional computer. Similarly to a graphics processing unit (GPU), which requires specialized programming to take advantage the parallelization, programming the Quantum Processing Unit (QPU) involves writing particular code to take the full advantages of it.

During the last few years various organisations have made quantum resources publicly available. IBM, for example, offers open access to up to a 15-qubit real quantum processor for testing and experimentation on quantum algorithms[1]. At the same time, many actors have contributed to the development of quantum hardware simulators and quantum computing frameworks. One of the most commonly used such frameworks is the python-based *qiskit* package from IBM [15]. Qiskit greatly facilitates the implementation of quantum algorithms and the seamless integration and execution on real or simulated quantum hardware.

A. Background on Quantum Computing

1) Quantum computing basics: In a quantum computer the classical bit is replaced by the *qubit* which is a quantum state of two levels. A qubit can be represented by a two-dimensional vector of complex parameters having norm 1. Using the "ket" $|\cdot\rangle$ to denote qubit variables a qubit can be, therefore, written as $|\mathrm{q}_1\rangle = \begin{bmatrix} a & b \end{bmatrix}^T$ where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. The complex parameters a, b are called amplitudes and are physically unknowable. To "read" the value of a qubit a physical process called *measurement* is performed. The measurement process randomly converts the state of the qubit to one of the two possible "observable" states: either $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$ or $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$ with probability $|a|^2$ and $|b|^2$ respectively.

A series of qubits, referred to as a *quantum register*, is a 2^n-dimensional vector of complex parameters and norm 1. A quantum register of two qubits, for example is

$$|\mathrm{qr}\rangle \equiv |\mathrm{q}_1\mathrm{q}_2\rangle \equiv |\mathrm{q}_1\rangle \otimes |\mathrm{q}_2\rangle = \begin{bmatrix} a & b & c & d \end{bmatrix}^T,$$

where $a, b, c, d \in \mathbb{C}$ and $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. Using a one-hot representation basis (called the "computational basis") the above are written $|\mathrm{qubit}\rangle = a|0\rangle + b|1\rangle$ and $|\mathrm{qr}_2\rangle = a|0\rangle + b|1\rangle + c|2\rangle + d|3\rangle$, where

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \\ \dots \\ 0 \end{bmatrix}, \dots |2^n - 1\rangle = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix}.$$

To manipulate a qubit, or a quantum register in general, "quantum gates" are used. Quantum gates, similarly to qubits, are represented by complex unitary matrices and are realized by specific physical processes.

2) Amplitude encoding: There are various ways to encode data on a quantum computer. One of the most used encodings is the *amplitude encoding*. The amplitude encoding uses the amplitudes of the quantum registers to store data. For example, a vector $|v\rangle = \begin{bmatrix} v_0, & v_1, & \dots, & v_{n-1} \end{bmatrix}^T$ would use $\lceil \log_2 N \rceil$ qubits to store the N parameters v_i by setting the amplitudes of the quantum register to $a = \frac{v_0}{\|v\|}, b = \frac{v_1}{\|v\|}, \dots$. The quantum register, therefore, would be

$$|v\rangle = \frac{1}{\|v\|} \sum_{0}^{N-1} v_i |i\rangle .$$

Let us consider a Hermitian matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$ and its eigen-decomposition $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$, where $\mathbf{Q} \in \mathbb{C}^{N \times N}$ is the square matrix the eigenvectors, represented as $|u_i\rangle$ for $i = 1, \dots, N$; and $\mathbf{\Lambda} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with its eigenvalues λ_i, for $i = 1, \dots, N$. Applying the amplitude encoding, matrix $\mathbf{A}$ can be represented as:

$$\mathbf{A} = \sum_i \lambda_i |u_i\rangle \otimes \langle u_i| ,$$

where a "bra" $\langle\cdot|$ vector signifies the Hermitian conjugate of the "ket" vector.

3) HHL algorithm: The Harrow-Hassidim-Lloyd (HHL) algorithm [5] has been proposed for solving systems of linear equations using a quantum computer and provide exponential speedup over the classical methods, such as the direct matrix inversion or the conjugate gradient algorithms.

HHL algorithm finds the vector $\mathbf{x} \in \mathbb{C}^{N \times 1}$ that satisfies

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{10}$$

where vector $\mathbf{b} \in \mathbb{C}^{N \times 1}$ are the measurements, and $\mathbf{x} \in \mathbb{C}^{N \times 1}$ is the unknown. The main idea of the HHL algorithm is to express the vectors $\mathbf{b}$ and $\mathbf{x}$ in terms of the (unknown) eigenvectors of matrix $\mathbf{A}$. The inverse of matrix $\mathbf{A}$ can be written as $\mathbf{A}^{-1} = \sum_i \frac{1}{\lambda_i} |u_i\rangle \otimes \langle u_i|$, vector $\mathbf{b}$, then, can be written in this basis as $\mathbf{b} \equiv |b\rangle = \sum_i \beta_i |u_i\rangle$, and the final solution as

$$\mathbf{b} \equiv |x\rangle = \sum_i \frac{\beta_i}{\lambda_i} |u_i\rangle . \tag{11}$$

In Fig. 1, a circuit schematic for the inputs/outputs of the HHL algorithm for the solution of equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ is shown below. The known vector $\mathbf{b}$ and the matrix $\mathbf{A}$ are used as inputs to the algorithm; after the algorithm is run the unknown vector $\mathbf{x}$ replaces the register where vector $\mathbf{b}$ was initially stored. Apart from the qubits that are used to store the known and unknown vectors $\mathbf{b}$ and $\mathbf{x}$, some additional qubits are needed for the operation of the quantum sub-procedures. For example, an eigenvalue-estimation sub-procedure ("QPE", see below) needs $\mathcal{O}(\log(1/\epsilon))$ additional qubits to estimate the eigenvalues of matrix $\mathbf{A}$ within additive error ϵ. These additional qubits are collectively referred to as the "auxiliary register".

[1]https://www.ibm.com/quantum

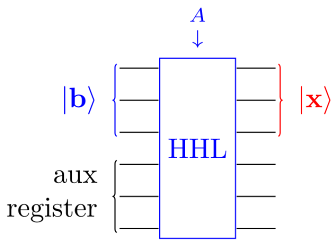

Fig. 1. Circuit schematic for the inputs/outputs of the HHL algorithm.

The core of the HHL algorithm consists of two steps that aim to transform the initial quantum state from $|\mathbf{b}\rangle$ to a state of the form of Equation (11). The first step performs the "Quantum Phase Estimation" (QPE) algorithm which "extracts" the eigenvalues of matrix $\mathbf{A}$ and the second step performs a controlled-rotation that inverts the eigenvalues forming the desired $\frac{\beta_i}{\lambda_i}$ state (Fig. 2). A quantum "controlled" operation is an operation that only affects one of the two states of a "controll" qubit. For example the controlled-U would not apply $\mathbf{U}$ when the control qubit is in state $|0\rangle$: $cU|0\rangle = |0\rangle$ but would apply $\mathbf{U}$ when the control qubit is in state $|1\rangle$: $c\mathbf{U}|1\rangle = \mathbf{U}|1\rangle$, i.e.,

$$c\mathbf{U}\Big(|0\rangle + |1\rangle\Big) = |0\rangle + \mathbf{U}|1\rangle\,. \tag{12}$$

The QPE algorithm efficiently finds an approximation of an eigenvalue of a unitary matrix given access to the respective eigenvector. Formally, it finds an approximation of $\phi \in [0,1)$ given $\mathbf{U}$ and $|\psi\rangle$ such that $\mathbf{U}|\psi\rangle = e^{2j\pi\phi}|\psi\rangle$. The output of the algorithm is a quantum state $|\tilde{\phi}\rangle$ that represents the approximation of ϕ in binary notation[2]. In the case of the HHL the unitary matrix $\mathbf{U}$ encodes matrix $\mathbf{A}$ through the relation $\mathbf{U} = e^{i\mathbf{A}t}$. This conversion—from $\mathbf{A}$ to $\mathbf{U}$—can be done efficiently in a quantum computer under the assumptions mentioned above. Obviously, matrices $\mathbf{U}$ and $\mathbf{A}$ share the same (exponentiated) eigenvalues and the same eigenvectors. Thus, after applying QPE on the state $\sum_i \beta_i |u_i\rangle$ it becomes $\sum_i \beta_i |\tilde{\lambda}_i\rangle |u_i\rangle$.

The controlled-rotation step "converts" the eigenvalues from their state-binary representation to real valued amplitudes, inversely proportional to the respective eigenvalues. That is, this step uses one more qubit through which a transformation of the following form is performed:

$$|\lambda\rangle|0\rangle \xrightarrow{\text{rotation}} |\lambda\rangle\left(\sqrt{1-\frac{C^2}{\lambda^2}}\,|0\rangle + \frac{C}{\lambda}|1\rangle\right), \tag{13}$$

where C is a normalization constant that should be $|C| < \min \lambda_i$ [5]. Applying this to the state after the QPE algorithm it becomes

$$\sum_i \beta_i |\tilde{\lambda}_i\rangle |u_i\rangle \left(\sqrt{1-\frac{C^2}{\lambda^2}}\,|0\rangle + \frac{C}{\lambda}|1\rangle\right), \tag{14}$$

[2]e.g. $|\tilde{\phi}\rangle = |010\rangle \rightarrow \phi \approx \frac{0}{2} + \frac{1}{4} + \frac{0}{8}$

where now there is the solution encoded in the $|1\rangle$ of the auxiliary qubit. The next, final, steps of the algorithm transform this to a "readable" solution. By performing the QPE algorithm in reverse the $|\tilde{\lambda}\rangle_i$ of Equation 14 become $|0\rangle$. Then, by measuring until we find the auxiliary qubit at state $|1\rangle$ we finally have the initial quantum register at the state

$$\sum_i C' \frac{\beta_i}{\lambda_i}|0\rangle|u_i\rangle = C'|0\rangle|x\rangle \tag{15}$$

with C' being the appropriate normalization constant.

Classically, the runtime of algorithms that solve N systems of linear equations is $\mathcal{O}(cN)$, where c encodes other relevant parameters of the system like the sparsity and the condition number of matrix $\mathbf{A}$ and the required precision of the solution. The conjugate gradient descent method, for example, has $c = sk\log(1/\epsilon)$ with s, k, ϵ being the sparsity, condition number and precision respectively. Quantum mechanically, using the HHL algorithm, the runtime is exponentially reduced to $\mathcal{O}(c'\log N)$ where c', again, encodes the other relevant parameters[3]. The initial proposal [5] for HHL requires that $\mathbf{A}$ is Hermitian, sparse and well-conditioned. Since then, many variations of the algorithm have been proposed that bypass some of these restrictions or are more efficient under different assumptions etc [16],[17].

The HHL algorithm can also be easily extended to solve the matrix equation $\mathbf{AX} = \mathbf{B}$, where $\mathbf{X}, \mathbf{B} \in \mathbb{C}^{N\times N}$, instead of solving for a system of linear equations. Even by naively applying N times the HHL algorithm the quantum mechanical runtime is $\mathcal{O}(c'N\log N)$, outperforming the best classical algorithms with runtime $\mathcal{O}(N^{2+\alpha})$ where α is a constant > 0.3. Different algorithms have also been proposed for more general or specific versions of inverting a matrix or solving a matrix equation, e.g. [18], [19], [20]

B. Connections between the two worlds

Let us now discuss on how the fundamental operations of the physical-layer mmWave massive MIMO communications, such as channel estimation, equalization and beamforming, could benefit from QPUs in terms of significant computational speed ups. Specifically, for speeding up the conventional channel estimation operation, QPU could be employed to quickly recover the channel matrix, or some of its basic properties, such as the second order statistics, or its eigenvalues.

1) Input-output formulation: Ignoring the AWGN for this discussion, the input/output formulation of a mmWave massive MIMO system is given by:

$$\mathbf{Y} = \mathbf{F}_{\text{RX}}\,\mathbf{H}\mathbf{F}_{\text{TX}}\mathbf{S}, \tag{16}$$

where $\mathbf{Y} \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{C}^{N\times N}$ and $\mathbf{H} \triangleq [\mathbf{h}_1, \ldots, \mathbf{h}_N] \in \mathbb{C}^{N\times N}$, while $\mathbf{F}_{\text{TX}}, \mathbf{F}_{\text{RX}}$ are the beamforming matrices at the TX and RX, respectively. To align (16) with the HHL system model in (10), equation (16) can be written equivalently as:

$$\bar{\mathbf{y}} = ((\mathbf{F}_{\text{TX}}\mathbf{S})^T \otimes \mathbf{F}_{\text{RX}})\bar{\mathbf{h}}, \tag{17}$$

[3]In some implementations, for example, $c' = s^2k^2/\epsilon$

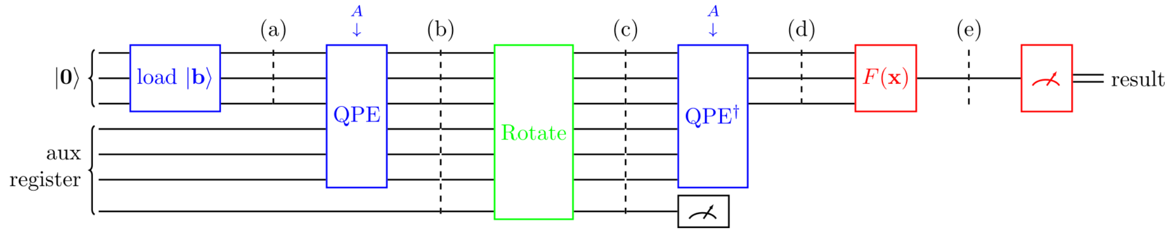

Fig. 2. The steps of the HHL algorithm for the solution of an equation $A\mathbf{x} = \mathbf{b}$. (a) load vector $\mathbf{b}$ to a quantum register (i.e. convert classical to quantum data) (b) perform the Quantum Phase Estimation algorithm to form $|\tilde{\lambda}\rangle$ (c) apply a controlled-rotation to form $\frac{1}{\tilde{\lambda}}|\tilde{\lambda}\rangle$ (d) perform the inverse Quantum Phase Estimation algorithm to disentangle the output register from the auxiliary register; the output register know contains the solution vector $\mathbf{x}$, (e) apply a function that outputs a desired metric of the vector (see Subsection III-C).

where $\bar{\mathbf{y}} \triangleq \text{vec}(\mathbf{Y}) \in \mathbb{C}^{N^2 \times 1}$ and $\bar{\mathbf{h}} \triangleq \text{vec}(\mathbf{H}) \in \mathbb{C}^{N^2 \times 1}$,

Considering that $\mathbf{A} \equiv (\mathbf{F}_{\text{TX}}\mathbf{S})^T \otimes \mathbf{F}_{\text{RX}}$, $\mathbf{b} \equiv \bar{\mathbf{y}}$, and $\mathbf{x} \equiv \bar{\mathbf{h}}$, the problem (17) can be treated naively by collecting N consecutive measurements/linear equations and solve them independently casting HHL algorithm N times, which will need $\mathcal{O}(c' N \log N)$ complexity. Alternatively, HHL could be employed for solving all the N^2 systems using, as already mentioned, more sophisticated variations of HHL, such as [20], which requires only $\mathcal{O}(c' 2 \log N)$ complexity.

2) Sparsity and low-rank properties: The system in (17) does not take into account the inherited sparsity and low-rank properties for the case of mmWave massive MIMO systems. Recall that, the unknown channel matrix can be decomposed in the beamspace, using the unitary DFT matrices $\mathbf{W} \in \mathcal{F}^{N \times N}$ as:

$$\mathbf{H} = \mathbf{W}\mathbf{Z}\mathbf{W}^H \Rightarrow \bar{\mathbf{h}} = (\mathbf{W}^* \otimes \mathbf{W})\bar{\mathbf{z}}, \tag{18}$$

where $\bar{\mathbf{z}} \triangleq \text{vec}(\mathbf{Z}) \in \mathbb{C}^{N^2 \times 1}$ is a sparse/compressible vector with $\|\mathbf{z}\|_1 \leq s$. Thus, plugging (18) into (17) we have:

$$\bar{\mathbf{y}} = ((\mathbf{F}_{\text{TX}}\mathbf{S})^T \otimes \mathbf{F}_{\text{RX}})(\mathbf{W}^* \otimes \mathbf{W})\bar{\mathbf{z}} = (\mathbf{F}_{\text{TX}}^T\mathbf{S}^T\mathbf{W}^* \otimes \mathbf{F}_{\text{RX}}\mathbf{W})\bar{\mathbf{z}}. \tag{19}$$

Classically, sparsity gives a linear contribution to the complexity of solving (19). The complexity of the quantum algorithm presented here, on the other hand, has a quadratic dependence on sparsity. This can be improved, though, and variations have been proposed recently that also perform almost linearly in regards to the sparsity s [17].

3) Hermitian property: To enable the representation of the measurement matrix $\mathbf{A}$ in a quantum computer, it has to be Hermitian, i.e., $\mathbf{A} = \mathbf{A}^H$. In general, the system matrix of (19), defined as $\mathbf{A} \triangleq \mathbf{F}_{\text{TX}}^T\mathbf{S}^T\mathbf{W}^* \otimes \mathbf{F}_{\text{RX}}\mathbf{W}$, is not Hermitian. For this case, an augmented Hermitian matrix can be defined and used in its place, i.e.,

$$\mathbf{A}' \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^H & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{2N \times 2N}, \text{ and } \mathbf{b}' \triangleq \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \in \mathbb{C}^{2N \times 1},$$

which increases the required qubits of the QPU by one, while the complexity of the simulation environment (e.g., qiskit) doubles up.

However, there are cases where the resulting linear systems of equations involve a measurement matrix that is inherently Hermitian. In particular, the matrix $\mathbf{S}$ with the training symbols, and the beamforming matrices $\mathbf{F}_{\text{TX}}$, $\mathbf{F}_{\text{RX}}$ could be properly designed, e.g., by requiring $\mathbf{F}_{\text{TX}} = \mathbf{F}_{\text{TX}}^H$, $\mathbf{F}_{\text{RX}} = \mathbf{F}_{\text{RX}}^H$,

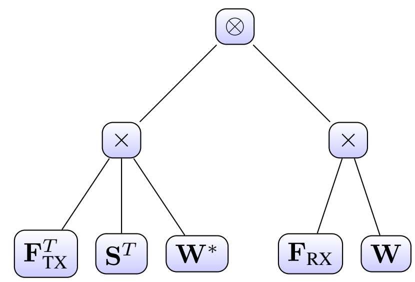

Fig. 3. Syntax tree paradigm for QPU-based composition of $\mathbf{A} \triangleq \mathbf{F}_{\text{TX}}^T\mathbf{S}^T\mathbf{W}^* \otimes \mathbf{F}_{\text{RX}}\mathbf{W}$

and $\mathbf{S} = \mathbf{S}^H$ then based on the property of the Kronecker product $(\mathbf{Q}_1^H \otimes \mathbf{Q}_2^H) = (\mathbf{Q}_1 \otimes \mathbf{Q}_2)^H$, we have that $\mathbf{A} = \mathbf{A}^H$. Recall that, the beamforming matrices are usually DFT-based, and thus, Hermitian matrices. Moreover, in other cases related to channel estimation problem, such as the linear MMSE equalization, the involved matrices may be Hermitian by default, e.g., matrix $\mathbf{R} \triangleq \mathcal{E}\{\mathbf{y}_t\mathbf{y}_t^H\}$ in (8) is Hermitian.

4) Matrix functions: The matrix $\mathbf{A}$ that we provide as an input to the QPU can be composed by several other matrices, i.e.,

$$\mathbf{A} \triangleq \mathbf{F}_{\text{TX}}^T\mathbf{S}^T\mathbf{W}^* \otimes \mathbf{F}_{\text{RX}}\mathbf{W}. \tag{20}$$

However, as the matrices' dimensions increase, it may become inefficient to derive the composition with the conventional resources. Alternatively, the matrix function $\mathbf{A}$ can be derived in the QPU, providing the composing matrices as inputs. An example of matrix embeddings have been proposed in [19], which allows the estimation of matrix functions corresponding to arbitrary syntax trees, as shown in Fig. 3 for (20).

C. Channel recovery from the quantum realm

It must be stressed, though, that the quantum versions of these algorithms *do not give access* to the full solution, e.g., the actual values of the solution. **The solution is stored, usually through the amplitude encoding, in a quantum state, the parameters of which are physically unknowable.** Therefore, we can distinguish two approaches for channel recovery:

1) Retrieve the physical values statistically: To extract the parameters of the quantum state one needs to statistically deduce their values by repeatedly running steps (a)-(e) in Fig. 2. This procedure has to be performed $\mathcal{O}(K)$ times in order to extract K parameters, while in this case, function $F(\mathbf{x})$ is the identity function. Although higher in complexity, this approach does not diminish the value of using them, as the logarithmic "head start" leaves a lot of room for improved performance of quantum versions of derived calculations on the solution. For instance, exploitation of the channel sparsity in $\bar{\mathbf{z}}$ may speed up the convergence of the physical values and reduce the required number of HHL employments.

2) Compute a function of the channel physical values: In their initial proposal [5] also suggested that, instead of calculating $\mathbf{A}$, $F(\mathbf{A})\,|b\rangle$ might be computed for any computable F. Similarly, using the technique presented in this work, instead of calculating matrix $\mathbf{H}$, $F(\mathbf{H})$ could be computed for various useful functions F. Such functions might include the mean value of some observable M by calculating $\langle x|M|x\rangle$ ($\int x^H Mx$), the absolute average $\frac{1}{N^2}|\sum \mathbf{H}_{i,j}|$, the singular value estimation, the verification of matrix products, etc [21], [22]. Specifically, estimation of the channel covariance matrix, $\mathcal{E}\{\mathbf{H}\mathbf{H}^H\}$, could be provided more efficiently than the recovery of the physical-values of $\mathbf{H}$. Afterwards, a covariance-based channel recovery technique could be applied, e.g., [23], [24], [25].

IV. Conclusions and Future Work

This work serves as a springboard for the development of quantum computing-assisted algorithms, focused on the physical-layer of the next-generation wireless systems. This is motivated by the massive increase of the computation requirements of the next-generation systems, where a massive amount of data have to be processed online. In this paper, we have identified some of the basic connections between the channel estimation problem and solution of a linear system of equations in a QPU. We provide a figure of merit for computational gains of QPU over the conventional approach. Also, we set two basic directions for exploiting the amplitude encoding for channel recovery. One obvious next step for the work presented here is the testing of the quantum algorithms on real data and on real or simulated quantum hardware using qiskit. Error mitigation, occuring in the quantum and communication domains will be investigated in subsequent works.

Acknowledgement

In this paper, Evangelos Vlachos has received funding from the European Union's Horizon 2020 innovation action programme ENERMAN under grant agreement No 958478.

References

[1] Syed Junaid Nawaz, Shree Krishna Sharma, Shurjeel Wyne, Mohammad N. Patwary, and Md. Asaduzzaman, "Quantum machine learning for 6g communication networks: State-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46317–46350, 2019.

[2] Ziwei Wan, Zhen Gao, Feifei Gao, Marco Di Renzo, and Mohamed-Slim Alouini, "Terahertz massive mimo with holographic reconfigurable intelligent surfaces," *IEEE Transactions on Communications*, vol. 69, no. 7, pp. 4732–4750, 2021.

[3] Yonina C. Eldar and Alan V. Oppenheim, "Quantum signal processing," *IEEE Signal Process. Mag.*, vol. 19, no. 6, pp. 12–32, 2002.

[4] Panagiotis Botsinis, Dimitrios Alanis, Zunaira Babar, Hung Viet Nguyen, Daryus Chandra, Soon Xin Ng, and Lajos Hanzo, "Quantum search algorithms for wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1209–1242, 2019.

[5] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, oct 2009.

[6] Helen Urgelles, Pablo Picazo-Martínez, and Jose F. Monserrat, "Application of quantum computing to accurate positioning in 6g indoor scenarios," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 643–647.

[7] Wenlu Xuan, Zhongqi Zhao, Lei Fan, and Zhu Han, "Minimizing delay in network function visualization with quantum computing," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, 2021, pp. 108–116.

[8] Behnam Ojaghi, Ferran Adelantado, Angelos Antonopoulos, and Christos Verikoukis, "Impact of network densification on joint slicing and functional splitting in 5g," *IEEE Communications Magazine*, vol. 60, no. 7, pp. 30–35, 2022.

[9] Karthik Kakaraparty, Edgard Muñoz-Coreas, and Ifana Mahbub, "The future of mm-wave wireless communication systems for unmanned aircraft vehicles in the era of artificial intelligence and quantum computing," in *2021 IEEE MetroCon*, 2021, pp. 1–8.

[10] Evangelos Vlachos, Aris S. Lalos, and Kostas Berberidis, "Low-Complexity OSIC Equalization for OFDM-Based Vehicular Communications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3765–3776, 2017.

[11] Evangelos Vlachos, Aris S. Lalos, Kostas Berberidis, and John Thompson, "Adaptive Windowing for ICI Mitigation in Vehicular Communications," *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 974–977, 2018.

[12] George C. Alexandropoulos, Evangelos Vlachos, and John Thompson, "Wideband channel tracking for millimeter wave massive mimo systems with hybrid beamforming reception," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 8698–8702.

[13] A M Sayeed, "Deconstructing multiantenna fading channels," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2563–2579, Oct. 2002.

[14] Evangelos Vlachos, Aris S. Lalos, and Kostas Berberidis, "Stochastic Gradient Pursuit for Adaptive Equalization of Sparse Multipath Channels," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 413–423, 2012.

[15] MD SAJID ANIS et. al, "Qiskit: An Open-source Framework for Quantum Computing," 2021.

[16] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma, "Exponential improvement in precision for simulating sparse Hamiltonians," in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, May 2014, STOC '14, pp. 283–292, Association for Computing Machinery.

[17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma, "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision," *SIAM Journal on Computing*, vol. 46, no. 6, pp. 1920–1950, Jan. 2017.

[18] Liming Zhao, Zhikuan Zhao, Patrick Rebentrost, and Joseph Fitzsimons, "Compiling basic linear algebra subroutines for quantum computers," *Quantum Machine Intelligence*, vol. 3, no. 2, pp. 21, June 2021.

[19] Wentao Qi, Alexandr I. Zenchuk, Asutosh Kumar, and Junde Wu, "Quantum algorithms for matrix operations and linear systems of equations," Mar. 2022.

[20] Li Xu, Xiao-qi Liu, Jin-min Liang, Jing Wang, Ming Li, and Shu-qian Shen, "Quantum algorithm for solving matrix equations of the form AX," *Laser Physics Letters*, vol. 19, no. 5, pp. 055202, Mar. 2022.

[21] Changpeng Shao, "Quantum Algorithms to Matrix Multiplication," July 2018.

[22] Harry Buhrman and Robert Spalek, "Quantum Verification of Matrix Products," July 2005.

[23] Dheeraj Prasanna and Chandra R. Murthy, "MmWave Channel Estimation via Compressive Covariance Estimation: Role of Sparsity and Intra-Vector Correlation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2356–2370, 2021.

[24] Jun Fang, Xingjian Li, Hongbin Li, and Feifei Gao, "Low-rank covariance-assisted downlink training and channel estimation for fdd massive mimo systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1935–1947, 2017.

[25] Shuang Qiu, David Gesbert, Da Chen, and Tao Jiang, "A covariance-based hybrid channel feedback in fdd massive mimo systems," *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8365–8377, 2019.