

Quantum Hardware-in-the-Loop for Optimal Power Flow in Renewable-Integrated Power Systems

Zeynab Kaseb, *Student Member IEEE*, Rahul Rane, *Student Member IEEE*,
Aleksandra Lekić, *Senior Member IEEE*, Matthias Möller, Amin Khodaei, *Senior Member IEEE*,
Peter Palensky, *Senior Member IEEE*, and Pedro P. Vergara, *Senior Member IEEE*

Abstract—Quantum computing has emerged as a promising computational paradigm to address unresolved challenges in the modeling and control of modern power systems. However, most existing studies focus on offline simulations, and a practical framework for validating quantum algorithms in real-time operational environments remains lacking. This study proposes a quantum hardware-in-the-loop framework that integrates a real-time digital simulator with quantum and quantum-inspired hardware to solve combinatorial power flow and optimal power flow formulations under dynamic operating conditions. The proposed framework is validated using the IEEE 9-bus test system and a modified version with integrated solar and wind farms. The results confirm successful integration and convergence within a predefined tolerance. The study also identifies key limitations and challenges, such as limited access to quantum and digital annealers, as well as the current scalability limitations, that must be considered in future developments. Nevertheless, the results highlight the potential of quantum computing to significantly enhance the modeling and control of future power systems with high penetration of renewable energy sources.

Index Terms—Adiabatic quantum computing, combinatorial optimization, Ising machines, load flow, quantum annealing, state estimation.

I. INTRODUCTION

The global transition toward sustainable energy has driven a rapid expansion of renewable energy sources (RES), particularly photovoltaic solar panels and wind turbines. However, the inherent intermittency and stochastic behavior of these resources pose significant challenges to power system operation, thus requiring advanced methodologies to ensure stability and reliability [1]. On the other hand, the rising costs and environmental impact of fossil fuels have underscored the urgency of minimizing their use and integrating RES more effectively [2].

To maintain secure and efficient grid operation and assess system conditions, transmission and distribution system operators rely on power flow (PF) analysis [3]. When economic,

stability, or operational objectives must be optimized, this extends to optimal power flow (OPF). Therefore, OPF plays a crucial role in power system operation and planning by determining the most efficient system configuration through the adjustment of control variables, including active and reactive power dispatch, transformer tap positions, and reactive power compensation [4]. Solving OPF problems enables system operators to minimize generation costs, enhance voltage stability, and ensure compliance with operational constraints, thereby improving the overall efficiency and cost-effectiveness of modern power systems [5].

Classical OPF solvers, including linear programming (e.g., [6]), nonlinear programming (e.g., [7]), quadratic programming (e.g., [8]), and Newton-based approaches (e.g., [9]), have been extensively studied and widely adopted in academia and industry [10]. While these solvers have proven effective for traditional power systems, their application to modern RES-integrated systems presents significant challenges. The inherent uncertainty of RES introduces complex, large-scale, and highly nonlinear and non-convex optimization problems, often leading to slow convergence and excessive computational burden [11]. As a result, advanced OPF algorithms are needed to ensure real-time applications, system stability, and optimal operation [12].

Recent research has explored artificial intelligence (AI) algorithms as promising alternatives. These approaches employ data-driven models to enhance scalability, improve convergence speed, and enable real-time decision-making in modern power systems. For example, physics-informed neural networks (e.g., [3], [13]) have been employed to embed underlying equations into AI algorithms, ensuring that solutions remain physically consistent while accelerating PF analysis and OPF. Similarly, graph neural networks (e.g., [14]) have been applied to exploit the underlying topology of power systems, enabling efficient and scalable OPF. Reinforcement learning (e.g., [15]) has also garnered attention for its ability to iteratively learn optimal control policies, thereby adapting to real-time power system dynamics.

Nevertheless, significant challenges remain unresolved for OPF in RES-integrated power systems. AI algorithms, while accelerating OPF, often rely on extensive training datasets and may suffer from generalization issues when encountering unseen grid conditions. In addition, their ability to guarantee feasibility and adherence to power system physics is constrained, necessitating hybrid approaches that still depend on conventional solvers. Furthermore, the non-convex, large-

Zeynab Kaseb, Rahul Rane, Aleksandra Lekić, Peter Palensky, and Pedro P. Vergara are associated with the Intelligent Electrical Power Grids Section, Electrical Sustainable Energy, Delft University of Technology, The Netherlands.

Matthias Möller is associated with the Delft Institute of Applied Mathematics, Delft University of Technology, The Netherlands.

Amin Khodaei is associated with Electrical and Computer Engineering, University of Denver.

Corresponding author: Zeynab Kaseb. Email: Z.Kaseb@tudelft.nl

This study is part of the DATALESs project 482.20.602, which is jointly financed by the Dutch Organization for Scientific Research (NWO), and National Natural Science Foundation of China (NSFC). The RSCAD/RTDS models and communication protocols were created as a part of Horizon Europe project PROSECCO, under grant agreement 101160687.

scale nature of OPF problems continues to pose computational challenges that classical and AI algorithms struggle to address in real-time applications (e.g., [16]).

Quantum computing (QC) is a computational paradigm that exploits the principles of quantum mechanics, such as superposition and entanglement, to solve certain classes of problems more efficiently than classical computers. QC can be broadly categorized into two main paradigms: gate-based quantum computing (GQC) and adiabatic quantum computing (AQC). GQC performs computations through sequences of quantum gates applied in discrete time steps, analogous to logical operations in classical circuits. In contrast, AQC is an analog approach that evolves a quantum system continuously toward the ground state of a problem Hamiltonian. Although AQC is polynomially equivalent to GQC, it is particularly well-suited for solving combinatorial optimization problems that can be formulated as Ising models or quadratic unconstrained binary optimization (QUBO) problems (e.g., [17]).

Recently, GQC has emerged as an alternative approach for OPF, offering the potential for fast computation and improved solution accuracy. Examples are exploring quantum circuit-based methods for OPF [18]–[20], and investigating the quantum-enhanced DC-OPF problem [21]. While GQC has gained significant attention due to advancements in quantum circuit design (e.g., [22]–[24]), its practical application to OPF problems remains challenging, primarily due to the dependence on large-scale, fault-tolerant hardware and the sensitivity to noise, which degrades computational accuracy and limits its effectiveness in solving large-scale OPF problems in the near to mid future (e.g., [25]–[27]).

AQC exhibits better noise resilience and has demonstrated promising results for combinatorial optimization problems, which makes it a promising candidate for OPF. Examples are exploring the application of AQC to the unit commitment problem, which shares structural similarities with OPF [28], and investigating the effectiveness of AQC for higher-order Ising models, which could be used to enhance the efficiency of OPF problems [29]. A comprehensive overview of the theoretical foundations and practical implementations of AQC, highlighting its advantages in solving NP-hard optimization problems, can be found in [30].

Current AQC hardware, however, still faces limitations. For example, Wave’s Advantage™ system (QA)¹ is one of the most mature realizations of AQC hardware, comprising over 5,000 qubits and 35,000 couplers, but suffers from restricted qubit connectivity. This necessitates minor embedding to map problem graphs onto the hardware graph, a process that is NP-hard and can increase resource demands. Furthermore, the limited range of interaction coefficients and local field values can introduce precision loss during problem encoding. An alternative hardware is Fujitsu’s quantum-inspired hardware (QIIO)², which emulates the annealing process using application-specific complementary metal-oxide semiconductor technology and supports massively parallel Markov Chain Monte Carlo updates. QIIO offers handling of up to 100,000

binary variables, organized in units of 8,192 fully connected bits, and operates with 64-bit precision at room temperature, thus providing a practical bridge toward scalable and noise-tolerant hardware.

Despite these promising developments, research on employing AQC for OPF remains in its early stages. Initial efforts, e.g., [31], have demonstrated the feasibility of AQC in power system optimization. However, challenges persist in mapping the full OPF problem onto AQC hardware. Comparative studies suggest that while AQC offers advantages in solving constrained optimization problems efficiently, further advancements in problem formulation, embedding techniques, and quantum hardware capabilities are required for practical deployment in real-time power system operations [29].

While recent theoretical advances in PF analysis and OPF with AQC hardware indicate great potential [31], their validation in real-time simulations remains essential to uncover practical limitations and implementation challenges. Hardware-in-the-loop (HIL) simulation is an effective approach for such validations, as it couples physical hardware, such as quantum or quantum-inspired processors, with real-time power system simulators. This setup provides a controlled yet realistic environment for systematically evaluating algorithmic performance, stability, and responsiveness under dynamic grid conditions, including variable renewable generation and fluctuating loads (e.g., [32], [33]). For example, a quantum-in-the-loop (QIL) framework that integrates GQC with real-time power system simulations is introduced in [34], showing the potential to address complex optimization challenges in modern power systems and to enhance real-time decision-making by processing inputs from numerous distributed sensors and controllers.

This study, therefore, presents a novel quantum hardware-in-the-loop (QHIL) framework for power systems with high RES integration, where the computational capabilities of QA and QIIO are leveraged to solve combinatorial PF and OPF formulations, building on the algorithms presented in [17]. In the proposed framework, quantum and quantum-inspired hardware are interfaced with a real-time digital simulator (RTDS[®]), which accurately models and simulates real-time power system dynamics. This integration also enables the estimation of overhead time associated with QA and QIIO, which is unavoidable with the current hardware and limits potential performance in near- to mid-term applications. As a proof of concept, the framework is validated using the IEEE 9-bus test system and a modified version that incorporates solar and wind farms. The main contributions of this study are as follows:

- Developing a QHIL framework that seamlessly integrates RTDS[®] with AQC hardware to solve combinatorial PF and OPF formulations;
- Developing a Windows-based middleware that enables efficient, reliable, and real-time communication between QIIO and RTDS[®]; and
- Systematically identifying the opportunities and challenges associated with the proposed QHIL framework, based on proof-of-concept simulations of a small-scale

¹<https://www.dwavesys.com>

²<https://www.fujitsu.com/global/services/business-services/digital-annealer>

power system achieving predefined accuracy thresholds for PF and OPF.

II. MATHEMATICAL FORMULATION

A. Power Flow (PF) Analysis

The objective is to compute the bus voltages and power injections that satisfy the power balance equations, as:

$$P_i = P_i^G - P_i^D, \quad \forall i \in \{1, \dots, N\}, \quad (1a)$$

$$Q_i = Q_i^G - Q_i^D, \quad \forall i \in \{1, \dots, N\}, \quad (1b)$$

where P_i and Q_i are net active and reactive power injections; P_i^G and Q_i^G total generated active and reactive power; P_i^D and Q_i^D total demand active and reactive power.

Equation (1) is typically solved iteratively, using numerical methods, such as the NR or Gauss-Seidel. It can also be solved with quantum or quantum-inspired (digital) annealers, which amounts to minimizing the sum of all the terms squared as:

$$H_{\text{obj}}(\mathbf{x}) = \sum_{i=1}^N (P_i - P_i^G + P_i^D)^2 + (Q_i - Q_i^G + Q_i^D)^2, \quad (2)$$

where $\mathbf{x} \in \{0, 1\}$ is vectors of binary variables. To convert the problem into a compatible form with quantum/digital annealers, P_i and Q_i are presented in rectangular coordinates and expanded into:

$$P_i = \sum_{k=1}^N \mu_i G_{ik} \mu_k + \omega_i G_{ik} \omega_k + \omega_i B_{ik} \mu_k - \mu_i B_{ik} \omega_k, \quad (3a)$$

$$Q_i = \sum_{k=1}^N \omega_i G_{ik} \mu_k - \mu_i G_{ik} \omega_k - \mu_i B_{ik} \mu_k - \omega_i B_{ik} \omega_k, \quad (3b)$$

where μ_i and ω_i are the real and imaginary parts of the complex voltage at bus i ; G_{ik} and B_{ik} the conductance and susceptance between buses i and k .

B. Optimal Power Flow (OPF)

The objective is to compute the optimal generator outputs $\{P_i^G, Q_i^G\}$ that minimize generation costs while satisfying all system constraints, i.e., the power balance, generation, and operational limits, formulated as:

$$\min \sum_{k \in \mathbb{G}} f_k(P_k^G), \quad (4)$$

subject to:

$$P_i = P_i^G - P_i^D, \quad (5a)$$

$$Q_i = Q_i^G - Q_i^D, \quad (5b)$$

$$\underline{P}_i^G \leq P_i^G \leq \overline{P}_i^G, \quad (5c)$$

$$\underline{Q}_i^G \leq Q_i^G \leq \overline{Q}_i^G, \quad (5d)$$

$$\underline{V}_i \leq V_i \leq \overline{V}_i, \quad (5e)$$

$$\underline{\delta}_i \leq \delta_i \leq \overline{\delta}_i, \quad (5f)$$

where $f_k(\cdot)$ is the fuel cost function; \mathbb{G} a subset of buses connected to generators $\{1, 2, \dots, N_G\}$; N_G is the total number of generators; \underline{P}_i^G and \overline{P}_i^G the minimum and maximum limits

for the generated active power at bus i ; \underline{Q}_i^G and \overline{Q}_i^G the minimum and maximum limits for the generated reactive power at bus i ; V_i voltage magnitude at bus i ; \underline{V}_i and \overline{V}_i the minimum and maximum limits for the voltage magnitude at bus i ; δ_i voltage phase angle at bus i ; $\underline{\delta}_i$ and $\overline{\delta}_i$ the minimum and maximum limits for the voltage phase angle at bus i .

We restructure the OPF formulation into a problem Hamiltonian, including the objective term, H_{obj} , as defined in (2), along with the inequality constraint terms, H_{const} , and the quality constraint term, H_{cost} , as defined in (6) and (7), respectively.

Inequality constraints (5c) to (5f) are incorporated into the problem Hamiltonian as penalty terms. A typical penalty term for an inequality constraint $g(x) \leq 0$ is $\lambda \max(0, g(x))^2$, where λ is a penalty parameter, resulting in:

$$\begin{aligned} H_{\text{const}}(\mathbf{x}) = & \sum_{i \in \mathbb{G}} \left[\lambda_0 \max(0, P_i^G - \overline{P}_i^G)^2 \right. \\ & + \lambda_1 \max(0, \underline{P}_i^G - P_i^G)^2 \\ & + \lambda_2 \max(0, Q_i^G - \overline{Q}_i^G)^2 \\ & \left. + \lambda_3 \max(0, \underline{Q}_i^G - Q_i^G)^2 \right] \\ & + \sum_{i=1}^N \left[\lambda_4 \max(0, V_i - \overline{V}_i)^2 \right. \\ & + \lambda_5 \max(0, \underline{V}_i - V_i)^2 \\ & + \lambda_6 \max(0, \delta_i - \overline{\delta}_i)^2 \\ & \left. + \lambda_7 \max(0, \underline{\delta}_i - \delta_i)^2 \right]. \end{aligned} \quad (6)$$

Finally, the objective function (4) is incorporated into the problem Hamiltonian as a penalty term. A typical penalty term for an equality constraint $h(x) = 0$ is $\lambda h(x)^2$, where λ is a penalty parameter, resulting in:

$$H_{\text{cost}}(\mathbf{x}) = \lambda_8 \sum_{k \in \mathbb{G}} f_k(P_k^G)^2. \quad (7)$$

Combining the objective and constraints gives:

$$H(\mathbf{x}) = H_{\text{obj}}(\mathbf{x}) + H_{\text{const}}(\mathbf{x}) + H_{\text{cost}}(\mathbf{x}). \quad (8)$$

While (6) and (7) allow for encoding constraints within the problem Hamiltonian, current AQC hardware does not natively support $\max(\cdot)$ operations, nor can they directly enforce inequality conditions. Therefore, we introduce slack binary variables to transform these constraints into an equivalent binary form compatible with existing implementations. Although this transformation increases the problem size, it offers a practical and currently necessary means to represent operational limits within the combinatorial formulation. The trade-off between model fidelity and hardware feasibility is therefore an inherent aspect of current implementations.

C. QUBO Representation

To further transform (2) and (8) into a form compatible with AQC hardware, the continuous variables μ_i and ω_i in (3) must be discretized. A straightforward discretization is:

$$\mu_i = \mu_i^0 + \Delta \mu_i (x_{i,0}^{\mu} - x_{i,1}^{\mu}), \quad (9a)$$

$$\omega_i = \omega_i^0 + \Delta\omega_i(x_{i,0}^\omega - x_{i,1}^\omega), \quad (9b)$$

where $x_{i,\{0,1\}}^{\{\mu,\omega\}} \in \{0,1\}$ are binary decision variables whose value decides whether the base values μ_i^0 and ω_i^0 are increased ($x_{i,0}^{[\circ]} = 1 \wedge x_{i,1}^{[\circ]} = 0$), decreased ($x_{i,0}^{[\circ]} = 0 \wedge x_{i,1}^{[\circ]} = 1$), or kept at their current value ($x_{i,0}^{[\circ]} = 0 \wedge x_{i,1}^{[\circ]} = 0$ or $x_{i,0}^{[\circ]} = 1 \wedge x_{i,1}^{[\circ]} = 1$). It should be noted that $[\circ]$ is a placeholder notation for μ and ω , respectively. That is, μ_i^0 can increase ($x_{i,0}^\mu = 1 \wedge x_{i,1}^\mu = 0$) while ω_i^0 decreases ($x_{i,0}^\omega = 0 \wedge x_{i,1}^\omega = 1$).

Working out all the terms at (2) and (8) yields a fourth-order polynomial for the binary variables. Detailed information on the expanded formulations is available in [17]. The solvers are QA and QIIO. While QIIO can effectively handle higher-order terms, QA can handle up to quadratic terms. Therefore, higher-order terms are reduced by introducing auxiliary variables of the form $z_{ij} = x_i x_j$ and replacing triplet interactions by:

$$x_i x_j x_k = \min_{z_{ij}} \{z_{ij} x_k + \lambda P(x_i, x_j; z_{ij})\}, \quad \lambda > 0, \quad (10)$$

with:

$$P(x_i, x_j; z_{ij}) = x_i x_j - 2(x_i + x_j)z_{ij} + 3z_{ij}, \quad (11)$$

Similarly, expressions with four binary variables can be replaced by:

$$x_i x_j x_k x_l = \min_{z_{ij}, z_{kl}} \{\lambda P(x_i, x_j; z_{ij}) + \lambda P(x_k, x_l; z_{kl})\}. \quad (12)$$

The discretized formulation can then be expressed as a QUBO formulation, which is minimized using quantum or quantum-inspired hardware. A QUBO formulation is defined by a symmetric, real-valued matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$, where n is the total number of decision variables, including slack and auxiliary variables. The corresponding binary optimization problem is formulated as:

$$\min_{\mathbf{x} \in \{0,1\}^n} f_{\mathbf{Q}}(\mathbf{x}), \quad (13)$$

with the quadratic objective function given by:

$$f_{\mathbf{Q}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{Q} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j. \quad (14)$$

The Python package PyQUBO is used to develop the QUBO based on (2) for PF and (8) for OPF when the solver is QA to reduce higher-order terms.

III. ALGORITHM IMPLEMENTATION

A detailed description of the designed algorithm for PF analysis can be found in [17]. Algorithm 1 is designed to solve the combinatorial OPF formulation (8) using quantum and digital annealers. In lines 1-3, \mathbf{P}^D , \mathbf{Q}^D , and $\mathbf{Y} = G_{ik} + jB_{ik}$ are defined based on the given operating scenario. In lines 4-7, μ^0 , ω^0 , $\Delta\mu$, and $\Delta\omega$ are initialized by the user. The values of \mathbf{P} and \mathbf{Q} are then calculated in lines 8 and 9 using μ^0 and ω^0 , where P_1 and Q_1 associated with the *slack* bus are excluded. In line 10, the problem Hamiltonian $H(\cdot)$ is

Algorithm 1 Adiabatic quantum algorithm to solve the combinatorial OPF formulation.

- 1: Initialize $\mathbf{P}^D = [P_1^D, P_2^D, \dots, P_{N-N_G-1}^D]$
 - 2: Initialize $\mathbf{Q}^D = [Q_1^D, Q_2^D, \dots, Q_{N-N_G-1}^D]$
 - 3: Initialize $\mathbf{Y} = \{(G_{ik} + jB_{ik}) : i, k = 1, 2, \dots, N\}$
 - 4: $\Delta\mu \leftarrow 1 \times 10^{-2}$
 - 5: $\Delta\omega \leftarrow 1 \times 10^{-3}$
 - 6: $\mu^0 = [\mu_1^0, \mu_2^0, \dots, \mu_N^0] \leftarrow 1$
 - 7: $\omega^0 = [\omega_1^0, \omega_2^0, \dots, \omega_N^0] \leftarrow 0$
 - 8: Calculate \mathbf{P} for PQ buses using (3a)
 - 9: Calculate \mathbf{Q} for PQ buses using (3b)
 - 10: Calculate residual using $H(\mathbf{x})$ in (8)
 - 11: $\epsilon \leftarrow 1 \times 10^{-2}$
 - 12: $it \leftarrow 0$
 - 13: **while** $H_{obj}(\mathbf{x}) > \epsilon$ and $it < it_{\max}$ **do**
 - 14: Solve QUBO and Update \mathbf{x}
 - 15: Update μ^0 and ω^0 using (9)
 - 16: Calculate \mathbf{P} for PQ buses using (3a)
 - 17: Calculate \mathbf{Q} for PQ buses using (3b)
 - 18: Calculate residual using $H(\mathbf{x})$ in (8)
 - 19: Update μ^0 and ω^0
 - 20: Update $\Delta\mu$, $\Delta\omega$
 - 21: $it \leftarrow it + 1$
 - 22: **end while**
-

evaluated with $\mathbf{x} = \{0\}^{4n}$ to propose a candidate solution with the initial values for μ^0 , ω^0 , $\Delta\mu$, and $\Delta\omega$. In line 11, the residual threshold $\epsilon = \frac{MW^2 + MVAR^2}{2}$ is defined by the user. In line 12, the iteration counter it is set to zero.

The iterative loop starts in line 13. In the first iteration, the initial candidate solution $\mathbf{x} = \{0\}^{4n}$ is used to update μ^0 and ω^0 in line 15, followed by calculating \mathbf{P} and \mathbf{Q} in lines 16 and 17. From the second iteration on, after solving the QUBO problem in line 14 using QA or QIIO, the resulting bitstring $\mathbf{x} = [x_{0,0}^\mu, \dots, x_{n,1}^\omega] \in \{0,1\}^{4n}$ is obtained and used to update μ^0 and ω^0 in line 15 according to (9). Note that the auxiliary variables are neglected. For all solvers, the number of readouts is 2,000. When the calculated residual using $H(\mathbf{x})$ in line 18 falls below ϵ , μ^0 , ω^0 are considered as the final solution. If not, they are updated in line 19.

In line 20, step sizes are adaptively updated to enhance convergence behavior. Specifically, $\Delta\mu$ and $\Delta\omega$ are defined as functions of it , such that larger step sizes are applied during the initial iterations to accelerate convergence, while smaller step sizes are used as the algorithm approaches the solution. Furthermore, $\Delta\mu$ and $\Delta\omega$ are updated per bus considering the history of changes in μ^0 and ω^0 from the previous and second previous iterations. This approach enables independent adjustment of μ_i and ω_i in different directions. The loop continues until convergence or the maximum iteration count it_{\max} are reached.

IV. QUANTUM HARDWARE-IN-THE-LOOP FRAMEWORK

A. Computational and Information Flow

Fig. 1 depicts the computational and information flow within the QHIL framework for OPF. Different colors and

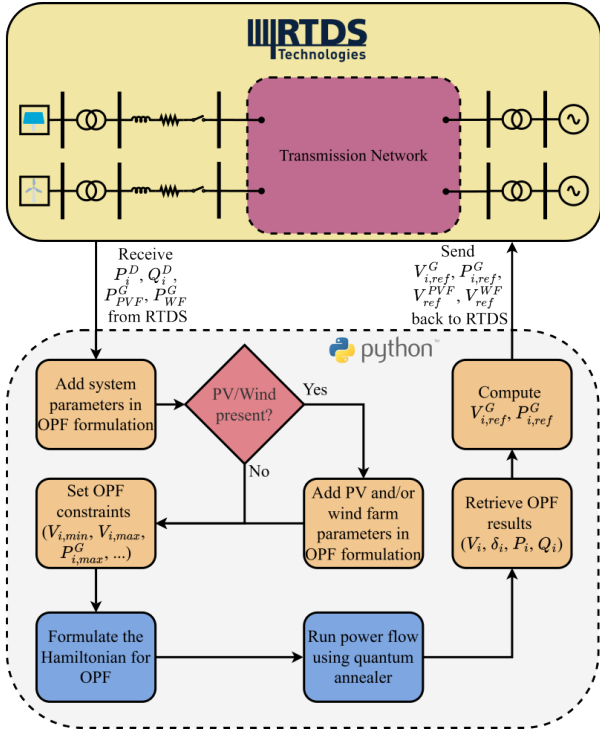


Fig. 1. Computational and information flow within the QHIL framework for OPF calculations. The OPF formulation receives system parameters, real-time data, P_i^D , Q_i^D , P_i^{PVF} , P_i^{WFF} , and operational constraints. The OPF is solved using a problem Hamiltonian on quantum/digital annealers. The results are compared to the NR solver in Pandapower. The computed generator set points $V_{i,ref}^G$ and $P_{i,ref}^G$ are then employed in the real-time power system simulation.

line styles are used to distinguish components and improve visual clarity. In particular, the yellow box represents the RTDS[®] hardware, while the dashed gray box corresponds to the Python environment. Solid box outlines denote the overall system architecture, whereas dashed boxes indicate internal processes, with arrows passing through them to show step-by-step operations and data flows within the system. The problem formulation is initially given essential power system parameters, such as the number of buses N , system connectivity, and transmission line resistance, inductance, and capacitance. In addition, real-time data are obtained from the simulator, including demand power values P_i^D and Q_i^D , as well as RES power generation outputs P_i^{PVF} and P_i^{WFF} . If solar and/or wind farms are integrated into the power system, the corresponding parameters are also added. For OPF, relevant constraints, including bus voltage limits \underline{V}_i and \overline{V}_i , as well as generation limits \underline{P}_i^G , \overline{P}_i^G , \underline{Q}_i^G , and \overline{Q}_i^G , are provided to ensure that the power system functions within acceptable operating limits.

The OPF problem is then solved using the proposed algorithm in Section III, executed on QA and QHIO. The results are subsequently collected from the annealers and can be used for modeling and control purposes, e.g., to calculate the required generator voltage $V_{i,ref}^G$ and active power $P_{i,ref}^G$ set points based on the approach used for modeling the generator's excitor and governor in the real-time power system simulation.

The required generator voltage $V_{i,ref}^G$ set point is given by:

$$V_{i,ref}^G = \frac{E_{f,i,pu}^G}{K_a} + V_{i,pu}^G \quad (15a)$$

$$\bar{E}_{f,i,pu}^G = \bar{V}_{i,pu}^G + jx_d \bar{I}_{i,pu}^G \quad (15b)$$

$$\bar{I}_{i,pu}^G = \frac{S_{i,pu}^*}{\bar{V}_{i,pu}^{G*}} \quad (15c)$$

$$S_{i,pu} = \frac{P_i^G + jQ_i^G}{S_{base}} \quad (15d)$$

where K_a is the excitor gain; x_d the d -axis reactance of the generator; $E_{f,i,pu}^G$ the field voltage of the generator in per-unit; $\bar{I}_{i,pu}^G$ the current flowing through the generator in per-unit; $\bar{V}_{i,pu}^G$ the generator terminal voltage; $S_{i,pu}$ the apparent power supplied by the generator in per-unit; S_{base} the base MVA value.

Similarly, the generator's active power $P_{i,ref}^G$ set point can be computed as:

$$P_{i,ref}^G = R \frac{P_i^G}{S_{base}} \quad (16)$$

where R is the governor droop coefficient.

Finally, the estimated generator and RES converter set points $V_{i,ref}^G$, $P_{i,ref}^G$, $V_{i,ref}^{PVF}$, and $V_{i,ref}^{WFF}$ are returned to the simulator as input for system operation.

B. Hardware Configuration

Fig. 2 shows the hardware configuration of the QHIL framework, where the dashed gray box highlights all equipment modeled in RTDS[®] and the associated connections, showing the internal operations of the real-time simulation framework. Power system simulations are carried out on the RTDS[®] NovaCor racks, and the response can be monitored using the RSCAD interface³. To provide real-time connectivity with the simulator, an external communication GTNETx2 card is used. When configured with the card's SKT protocol, it allows the simulator and external devices to communicate via the transmission control protocol (TCP). This TCP communication further allows the simulator to export system parameters, such as power demand values P_i^D and Q_i^D , as well as RES power generation P_i^{PVF} and P_i^{WFF} for OPF. Here, P_i^{PVF} and P_i^{WFF} are the active power generated by the solar and wind farms, respectively. Subsequently, the computed generator and RES converter set points $V_{i,ref}^G$, $P_{i,ref}^G$, $V_{i,ref}^{PVF}$, and $V_{i,ref}^{WFF}$ are sent back to the simulator in real-time.

RTDS[®] can be directly interfaced with QA. However, a direct connection with QHIO is currently not feasible, primarily due to the lack of proximity between the hardware and the simulation environment, and secondly, because vendor-specific VPN network access is required. Therefore, a Windows-based middleware is developed to facilitate communication between QHIO and RTDS[®].

The inputs required the OPF problem are sent via Python scripts to the proposed algorithms in Section III. The Python script is executed within a Windows Subsystem for Linux (WSL) environment, which runs a virtual Ubuntu instance,

³<https://www.rtds.com/technology/graphical-user-interface>

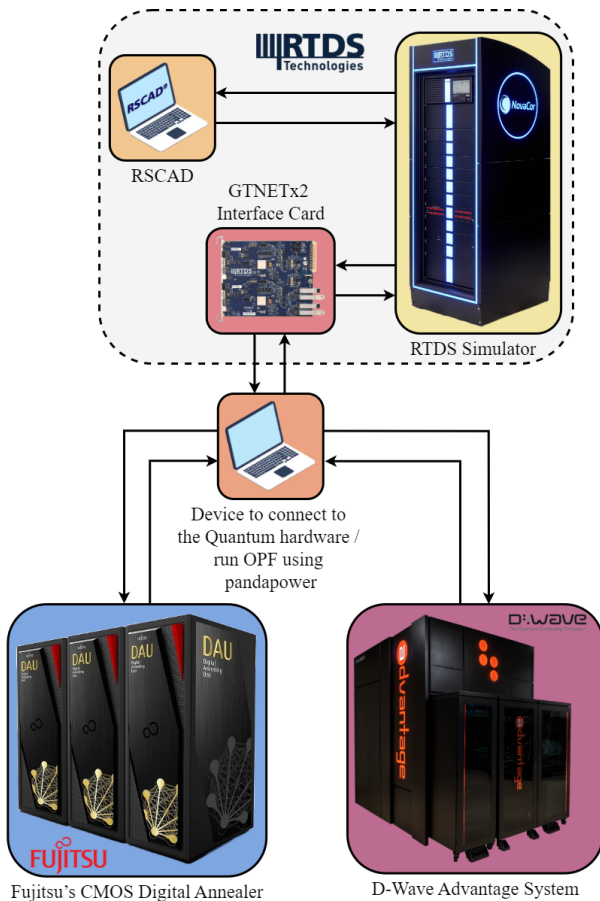


Fig. 2. Hardware configuration of the proposed QHIL framework for real-time power system simulations using RSCAD/RTDS[®]. A GTNETx2 card with SKT protocol enables TCP communication to exchange system parameters and RES generation data with the OPF algorithm running on the quantum hardware or local device, which then returns generator and converter set points to the simulator in real time.

where the required Python packages (e.g., D-Wave SDK, PyQUBO, DADK) are installed. Azure VPN Client is used to establish a connection with QIIO. However, this step is not necessary for QA. After obtaining solutions for OPF, the corresponding generator set points are then transmitted back to the simulation loop.

V. RESULTS

The proposed QHIL framework is validated by performing PF analysis and OPF for the IEEE 9-bus test system and a modified version that incorporates solar and wind farms, resulting in a 13-bus test system. Power system simulations are performed on the RTDS[®] NovaCor rack, and AQC experiments are performed on QA and QIIO. The compiler used for the former is PyQUBO, and for the latter is DADK. The tolerance threshold of $\epsilon = 1 \times 10^{-2}$ is chosen for the error bound of PF and OPF due to our limited hardware access and the computational expenses associated with QA and QIIO. As observed, the residuals obtained at termination remain close to the prescribed threshold, indicating that the iterative process is approaching convergence and would reach tighter tolerances if additional iterations were permitted. Therefore,

the imposed iteration cap reflects high computational cost and limited access to available quantum(-inspired) computing resources, rather than the algorithm's theoretical convergence behavior or scalability.

A. Computational Performance Assessment

The computational setup for solving PF and OPF combinatorial formulations using QA and QIIO is summarized in Table I. Key computational performance metrics include the number of variables in the QUBO formulation, compilation time, iteration time, and the total number of iterations required for convergence. Overall, QIIO demonstrates significantly faster compilation and iteration times compared to QA. For example, in the PF problem for the IEEE 9-bus test system, the compilation time drops from 4.35 seconds with QA to 0.27 seconds with QIIO, while the iteration time reduces from 238.48 seconds for QA to 0.93 seconds for QIIO. The OPF problem for the 13-bus test system exhibits higher computational complexity, involving 635 variables and a longer compilation time of 41.25 seconds.

The computational times reported in Table I include the computation overhead associated with QA and QIIO, which is unavoidable with current hardware. The processing time of QA and QIIO is also compared with that of the NR solver. For the 9-bus test system, NR achieves an average processing time of 0.049 seconds for PF and 0.397 seconds for OPF, measured over 10,000 runs. For 2,000 readouts/samples per iteration, the average anneal time per iteration for PF is 0.04 seconds using QA and 0.671 seconds using QIIO. The corresponding time for OPF on the 9-bus test system using QIIO is 0.981 seconds. Note that with NR, the OPF processing time is roughly eight times that of PF, while with QIIO, this ratio is less than two, which suggests improved scalability potential as hardware continues to advance. It should be emphasized, however, that the processing times reported for QA and QIIO are subject to variability due to several factors, such as hardware noise, specialized interfaces, and VPN connections. Consequently, accurately isolating the pure processing time remains challenging at present.

B. Error Assessment

Table II shows the differences in V_i , δ_i , P_i , and Q_i for PF and OPF using QA and QIIO relative to the results obtained by NR. These differences are quantified as the mean and standard deviation of the absolute discrepancies between the reference NR solution and the QA or QIIO solutions across all buses. Accordingly, the final deviations for V_i and δ_i remain within the predefined residual threshold (10^{-2}), which indicates the feasibility and reliability of the proposed algorithms.

To examine the achieved errors more closely, the PF result for the 9-bus test system obtained using QA and QIIO is compared against that of NR in Fig. 3. The top row shows the computed values of V_i , δ_i , P_i , and Q_i , while the bottom row shows the absolute errors of QA, QIIO, relative to NR for each corresponding value. As observed, QA accurately computes both V_i and δ_i , yielding P_i and Q_i values that closely match those of NR. QIIO also produces results within the predefined

TABLE I
PERFORMANCE COMPARISON OF QA AND QIIO FOR THE PF AND OPF PROBLEMS IN TERMS OF VARIABLE COUNT, COMPILATION TIME, AND ITERATION TIME ACROSS VARIOUS TEST SYSTEMS.

Test System	Problem	Annealer	Compiler	Var. No.	Compile Time (s)	It. Time (s)	It. No.
9-bus	PF	QA	PyQUBO	666	4.35	238.48	152
9-bus	PF	QIIO	DADK	249	0.27	0.93	119
9-bus	OPF	QIIO	DADK	331	0.42	1.81	200
13-bus	OPF	QIIO	DADK	635	41.25	42.37	200

TABLE II
DEVIATION ANALYSIS OF VALUES FOR V_i , δ_i , P_i , AND Q_i OBTAINED BY QA AND QIIO FOR THE PF AND OPF PROBLEMS.

Test System	Problem	Annealer	$ \Delta V_i $ (p.u.)	$\sigma_{ \Delta V_i }$ (p.u.)	$ \Delta \delta_i $ (deg.)	$\sigma_{ \Delta \delta_i }$ (deg.)	$ \Delta P_i $ (MW)	$\sigma_{ \Delta P_i }$ (MW)	$ \Delta Q_i $ (MVAR)	$\sigma_{ \Delta Q_i }$ (MVAR)
9-bus	PF	QA	2.53×10^{-4}	2.41×10^{-4}	2.13×10^{-3}	1.37×10^{-3}	0.03	0.04	0.18	0.29
9-bus	PF	QIIO	1.29×10^{-3}	1.64×10^{-3}	1.88×10^{-2}	2.15×10^{-2}	0.47	0.76	0.86	1.85
9-bus	OPF	QIIO	1.24×10^{-3}	1.75×10^{-3}	1.37×10^{-2}	9.95×10^{-3}	0.13	0.23	1.82	3.25
13-bus	OPF	QIIO	1.57×10^{-3}	9.83×10^{-4}	1.52×10^{-2}	2.20×10^{-2}	0.06	0.16	1.13	2.84

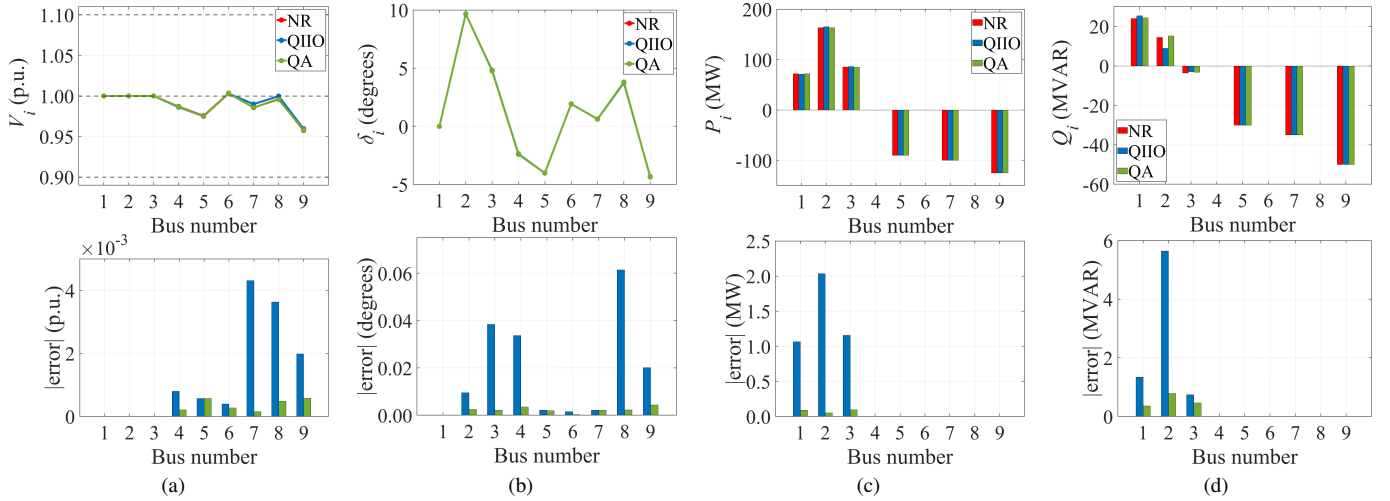


Fig. 3. PF results for the IEEE 9-bus test system using QA and QIIO, compared to NR. The top row shows the computed values for (a) V_i , (b) δ_i , (c) P_i , and (d) Q_i . The bottom row shows the absolute errors between QA, QIIO, and NR for each corresponding parameter.

residual threshold for V_i and δ_i , as shown in Figs. 3a and 3b. However, QA achieves higher accuracy in P_i and Q_i by up to 90%, as shown in Figs. 3c and 3d. These results indicate that while both QA and QIIO satisfy the overall convergence and accuracy criteria based on average error across all buses, QA demonstrates superior accuracy per bus.

The experiments are extended to include OPF for the 9-bus and 13-bus test systems for which the results using QIIO are presented in Figs. 4 and 5, respectively. QA is not included as the problem graph could not be embedded onto the available hardware graph. Furthermore, due to limited access to QIIO, simulations are terminated after $it_{\max} = 200$ iterations, leading to residuals slightly more than the predefined residual threshold, see Tables I and II. Nevertheless, the V_i and δ_i values computed by QIIO exhibit agreement with NR. For the 9-bus test system, the mean deviation in P_i between QIIO and NR is 2.20×10^{-2} MW, while in Q_i , the mean deviation is 1.13 MVAR, as shown in Figs. 4c and 4d. For the 13-bus test system, the corresponding deviations are 6.92×10^{-2} MW and 1.15 MVAR, respectively, as shown in Figs. 5c and 5d.

The maximum error in \mathbf{P} for PF is 0.1% using QA and 2% using QIIO for the 9-bus test system. For OPF, QA results are not included due to hardware embedding limitations. Using QIIO, the maximum error in \mathbf{P} is 0.6% for both the 9-bus and 13-bus test systems. The corresponding maximum errors in \mathbf{Q} are 9.2% and 9.7% for the 9-bus and 13-bus test systems, respectively. The relatively higher deviations observed in \mathbf{Q} can be attributed to several factors. First, the IEEE 9-bus system exhibits a low R/X ratio (see Table V), which is characteristic of transmission networks. In such systems, \mathbf{Q} calculations are more sensitive to numerical approximations, and it is well known that classical PF solutions may exhibit larger discrepancies in \mathbf{Q} compared to \mathbf{P} (e.g., [35]). Second, the higher deviations are primarily observed at generator buses, where differences in modeling assumptions between the NR formulation and the RTDS model, such as generator limits or shunt compensation, can lead to slight voltage mismatches. Due to the strong coupling between voltage magnitude and reactive power, even small deviations in \mathbf{V} can lead to amplified errors in computed \mathbf{Q} .

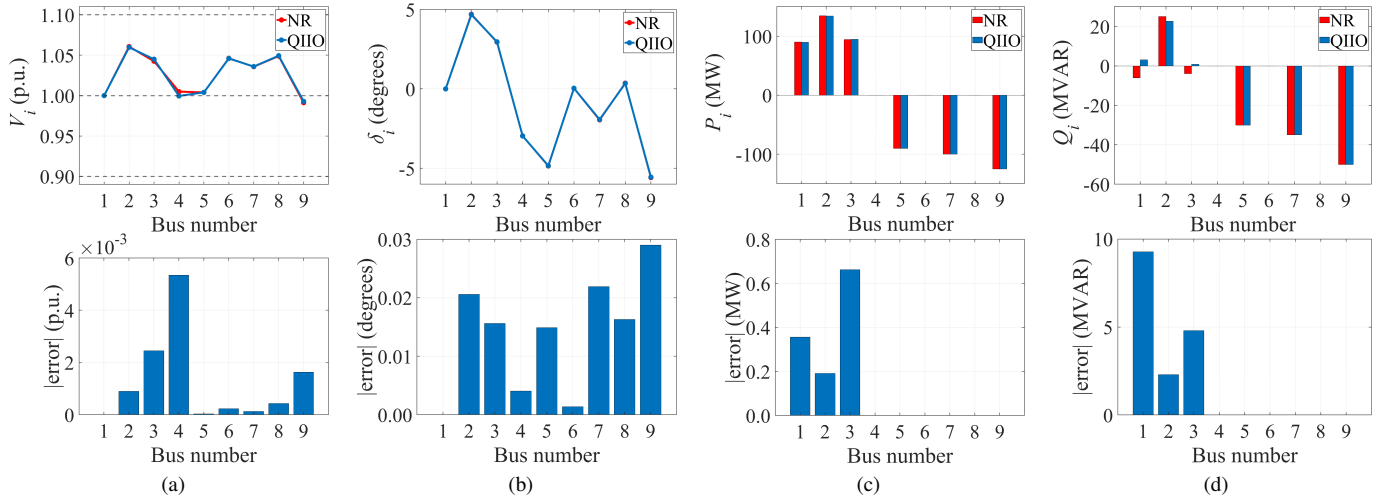


Fig. 4. OPF results for the IEEE 9-bus test system using QIIO, compared to NR. The top row shows the computed values for (a) V_i , (b) δ_i , (c) P_i , and (d) Q_i . The bottom row shows the absolute errors between QIIO and NR for each corresponding parameter.

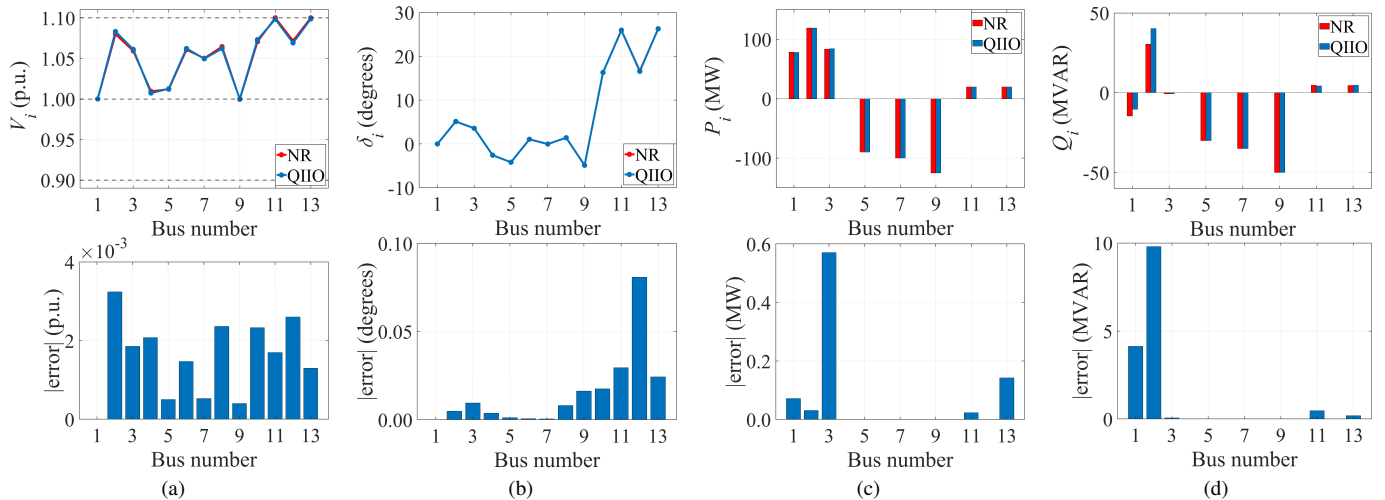


Fig. 5. OPF results for the IEEE 13-bus test system with integrated RES using QIIO, compared to NR. The top row shows the computed values for (a) V_i , (b) δ_i , (c) P_i , and (d) Q_i . The bottom row shows the absolute errors between QIIO and NR for each corresponding parameter.

VI. DISCUSSION

Combinatorial PF and OPF formulations enable compatibility with AQC hardware and a discrete modeling approach that ensures convergence. However, discretization introduces approximations that can reduce numerical accuracy, as shown in Table II. The quadratic nature of current AQC hardware necessitates reducing higher-order terms via auxiliary variables, and slack binary variables are also needed to encode hard constraints, both of which increase the dimensionality. In our previous study [17], we conducted extensive simulations on small-scale test systems, including ill-conditioned cases, to demonstrate the robustness and accuracy of the proposed algorithms. The present study introduces a novel QHIL framework for power systems with high RES integration, leveraging the computational capabilities of QA and QIIO to address combinatorial PF and OPF formulations. Therefore, comprehensive sensitivity analyses under varying load conditions or high RES uncertainty are beyond the scope of this study and are left

for future investigation. In addition, in transmission systems, decoupled PF methods may provide more accurate estimates of reactive power (e.g., [36]). Therefore, future research can investigate the performance of different classical solvers to identify the most suitable approaches for both transmission and distribution systems.

In the algorithms originally proposed in [17], the step sizes are not fixed but are adaptively updated. Specifically, larger step sizes are employed during the initial iterations to accelerate convergence when the obtained values are far from the solution, whereas smaller step sizes are used as the algorithm approaches the solution to improve stability and accuracy. This adaptive strategy is governed by the residual norm, in which the step size is reduced when the residual falls below predefined thresholds, indicating proximity to the solution and the need for improved numerical stability. This strategy is consistent with standard practices in gradient-based methods, among others, that start with a large step size and

reduce until a sufficient decrease condition is met (e.g., [37]). In addition, the step sizes are updated per bus, enhancing numerical stability, mitigating cumulative discretization errors, and reducing the number of hardware calls. For example, oscillating the value obtained at a given bus across successive iterations indicates that the step size is too large and is accordingly reduced to ensure convergence.

Despite current hardware limitations, early-stage investigations are essential to assess the potential of quantum and quantum-inspired technologies for power system applications, as has already been explored in other domains such as computational fluid dynamics (e.g., [38]). Therefore, this study focuses on methodological development and feasibility rather than large-scale deployment, which is not yet practically attainable with existing quantum hardware. Ongoing hardware advancements, particularly those enabling many-body interactions, higher qubit counts, and improved connectivity, are expected to alleviate problem size issues. Yet, we have explored several strategies to mitigate these challenges within current hardware limitations. In related studies [17], we demonstrate that reformulating the problem using spin variables $\{\pm 1\}$ instead of binary variables $\{0, 1\}$ can reduce the number of decision variables and couplers needed. In addition, a partitioned formulation is proposed that solves subsets of the problem and decreases the problem size per iteration without compromising accuracy. However, developing problem-specific schemes for more efficient reduction of higher-order terms remains an important direction for future study.

VII. CONCLUSION

This study is motivated by the increasing complexity of renewable-integrated power systems and the emerging potential of AQC to address large-scale nonlinear optimization problems. We develop a QHIL framework to bridge the gap between real-time power system modelling and control and AQC hardware. We also develop a middleware that ensures efficient real-time communication between QIIO and RTDS[®]. The framework is validated using the IEEE 9-bus test system and a modified version incorporating solar and wind farms. The algorithms developed in [17], are used to solve the combinatorial PF and OPF formulations. Based on the observations, lessons learned, and results, we demonstrate the feasibility of the proposed QHIL framework and identify practical challenges associated with current and near- to mid-future hardware capabilities.

The tolerance threshold of 10^{-2} and $it_{\max} = 200$ are selected as stopping criteria for PF and OPF due to limited hardware access and the computational cost associated with QA and QIIO. Within this relatively relaxed tolerance, the results show good agreement with those obtained using the NR solver in Pandapower. Furthermore, we observe that QIIO exhibits better scalability in terms of computational processing time compared to the NR solver. As the problem size increases, the growth rate of the computational processing time of QIIO remains lower than that of NR.

While the average and per bus deviations in \mathbf{P} remain below 1% for all experiments, the corresponding deviations

in \mathbf{Q} are comparatively higher. This behavior is consistent with established findings showing that voltages are generally more sensitive to variations in \mathbf{Q} than in \mathbf{P} (e.g., [35]). Consequently, small deviations in \mathbf{V} can lead to amplified errors in computed \mathbf{Q} . Furthermore, we observe that while QA and QIIO satisfy the overall accuracy criteria based on average error across all buses, QA demonstrates superior accuracy per bus. In this perspective, a bus-specific residual, rather than relying on average deviations, can potentially enhance accuracy per bus.

For the current AQC hardware, which is not yet as readily accessible or scalable as mature classical hardware, computational overhead remains a key limitation for real-time applications. Nevertheless, the information flow within the proposed QHIL framework operates effectively and demonstrates reliable communication between components. That is, once fault-tolerant quantum hardware becomes available, the framework is expected to respond efficiently to dynamic changes in system topology or operating conditions in real time. This capability arises from the algorithm's iterative structure and warm-starting strategy. When system conditions change, the solution from the previous run is used as the initial point for the next run, which enables rapid re-convergence. Note that this dynamic adaptability is independent of the residual threshold, which only determines the stopping criterion of each individual run.

Furthermore, current QA is constrained by limited qubit counts, sparse connectivity, and restricted precision in coupling coefficients, which collectively hinder their application to large-scale power systems. These challenges are expected to diminish as quantum hardware matures. In contrast, QIIO can handle up to 100,000 decision variables and therefore does not currently face the same scalability limitations. Another key observation is that middleware design and data exchange efficiency are highly platform-specific, that is, different hardware platforms require specific middleware implementations even when solving the same PF and OPF formulations. This highlights the importance of developing standardized, adaptable middleware for future real-time applications across various hardware platforms.

APPENDIX

The IEEE 9-bus test system is a simplified representation of a power grid, with nine buses, three generators, three transformers, and six transmission lines, as depicted in Fig. 6. Three of the buses are categorized as *PV* buses, indicating that they are equipped with generators that provide active power and regulate bus voltage by controlling reactive power. These generators are connected to buses 1, 2, and 3. The remaining six buses are *PQ* buses, which may or may not be connected to a load. Specifically, loads are connected to buses 5, 7, and 9. Three step-up transformers connect the generators to the transmission network, and six transmission lines interconnect the buses.

A 1 MW solar farm and a 2.5 MW wind farm have been modeled and connected to bus 7 to integrate RES, see Fig. 6. These farms, Generators 4 and 5 in Table III, are then scaled

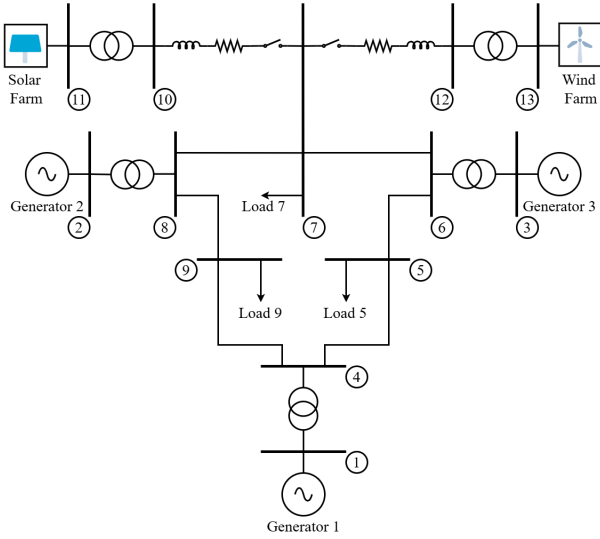


Fig. 6. The IEEE 9-bus test system representation with integrated RES. The system includes nine buses, three generators at buses 1, 2, and 3 (PV buses), and loads at buses 5, 7, and 9 (PQ buses). A 1 MW solar farm and a 2.5 MW wind farm are connected to bus 7 which are scaled to 20 MW each. Both RES units are connected using step-up transformers and RL branches, maintaining a constant short circuit ratio. Voltage controllers at the RES buses allow them to function as PV buses in PF and OPF formulations.

up to 20 MW each using the sub-step scaling option available when modeling in RTDS[®]. Each RES is connected to the power system via a step-up transformer and a series RL branch, ensuring that the farms' short-circuit ratio remains constant as the power output varies. A switch is also included to allow for the connection and disconnection of the RES when simulating various operational conditions. Since these RES units are equipped with voltage controllers in the grid-side converters at their respective connecting buses, they can be effectively modeled as PV buses in the PF and OPF formulations.

The power output of each RES is measured at the grid-side converter, corresponding to the point of common coupling (PCC), such that the measured active power represents the net power injected into the grid. Therefore, converter losses are inherently excluded from the measured values. Consequently, the maximum active power reported in Table III (19.8 MW) corresponds to the actual power delivered to the grid, which is slightly lower than the nominal 20 MW rating. The cost parameters of these RES units are set lower than those of conventional generators, reflecting their reduced operational and maintenance costs while providing a realistic economic representation in the OPF simulations. Tables III to VI respectively summarize the test system's generator, load, line, and cost parameters, considering base values of 345 kV and 100 MVA.

ACKNOWLEDGMENT

The authors extend their sincere gratitude to Jülich Supercomputing Centre for providing computing time on the D-Wave Advantage[™] System JUPSI through the Jülich Unified Infrastructure for Quantum computing (JUNIQU). The research received support from the Center of Excellence RAISE,

TABLE III
GENERATOR PARAMETERS FOR THE IEEE 9-BUS TEST SYSTEM WITH INTEGRATED RES, INCLUDING GENERATOR LOCATIONS, CAPACITIES, AND OPERATIONAL LIMITS.

Sr. No.	Bus i	P_i^G (MW)	V_i (p.u.)	δ_i (deg.)	P_i^G (MW)	P_i^G (MW)	Q_i^G (MVAR)	Q_i^G (MVAR)
1	1	-	1	0	250	10	300	-300
2	2	163	1	-	300	10	300	-300
3	3	85	1	-	270	10	300	-300
4	11	19.8	1	-	19.8	19.8	20	-20
5	13	19.8	1	-	19.8	19.8	20	-20

TABLE IV
LOAD PARAMETERS FOR THE IEEE 9-BUS TEST SYSTEM WITH INTEGRATED RES, CONTAINING THE ACTIVE AND REACTIVE POWER DEMANDS AT EACH LOAD-CONNECTED BUS.

Sr. No.	Bus i	P_i^D (MW)	Q_i^D (MVAR)
1	5	90	30
2	7	100	35
3	9	125	50

TABLE V
TRANSMISSION LINE PARAMETERS FOR THE IEEE 9-BUS TEST SYSTEM WITH INTEGRATED RES, SPECIFYING THE RESISTANCE (r_{ij}), INDUCTANCE (x_{ij}), AND SUSCEPTANCE (b_{ij}) VALUES OF EACH LINE CONNECTING THE BUSES BASED ON A 345 kV, 100 MVA BASE.

Sr. No.	From Bus i	To Bus j	r_{ij} (p.u.)	x_{ij} (p.u.)	b_{ij} (p.u.)
1	1	4	0	0.0576	0
2	4	5	0.017	0.092	0.158
3	5	6	0.039	0.17	0.358
4	3	6	0	0.0586	0
5	6	7	0.0119	0.1008	0.209
6	7	8	0.0085	0.072	0.149
7	8	2	0	0.0625	0
8	8	9	0.032	0.161	0.306
9	9	4	0.01	0.085	0.176
10	7	10	0.2379	1.6189	0
11	10	11	0.01	1	0
12	7	12	0.2583	1.6552	0
13	12	13	0.01	1	0

TABLE VI
GENERATOR COST PARAMETERS FOR THE IEEE 9-BUS TEST SYSTEM WITH INTEGRATED RES, SPECIFYING THE COST COEFFICIENTS USED IN THE OPF CALCULATIONS. THE COST FUNCTION IS MODELED AS A POLYNOMIAL $C_0 + P_i^G C_1 + P_i^{G2} C_2$.

Generator	C_0 (euros)	C_1 (euros/MW)	C_2 (euros/MW ²)
1	150	5	0.11
2	600	1.2	0.085
3	335	1	0.1225
4	8	0.3	0.0005
5	10	0.5	0.001

which receives funding from the European Union's Horizon 2020–Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement number 951733. The authors also would like to thank Fujitsu Technology Solutions for providing access to the QIIO software⁴ and to Markus Kirsch and Matthieu Parizy for their support and custom extensions of the DADK Python package. Furthermore, the authors acknowledge the support of lab technician

⁴<https://en-portal.research.global.fujitsu.com/kozuchi>

Remko Koornneef for his assistance in setting up the RTDS[®] laboratory and facilitating hardware communication and integration.

REFERENCES

- [1] M. S. Alam, F. S. Al-Ismael, A. Salem, and M. A. Abido, "High-level penetration of renewable energy sources into grid utility: Challenges and solutions," *IEEE access*, vol. 8, pp. 190 277–190 299, 2020.
- [2] H. van Asselt, P. Fragkos, L. Peterson, and K. Fragkiadakis, "The environmental and economic effects of international cooperation on restricting fossil fuel supply," *International Environmental Agreements: Politics, Law and Economics*, vol. 24, no. 1, pp. 141–166, 2024.
- [3] Z. Kaseb, S. Orfanoudakis, P. P. Vergara, and P. Palensky, "Physics-informed neural network with adaptive activation for power flow," *International Journal of Electrical Power & Energy Systems*, vol. 174, p. 111525, 1 2026.
- [4] M. Huneault and F. D. Galiana, "A survey of the optimal power flow literature," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 762–770, 1991.
- [5] N. Mbuli and W. Ngaha, "A survey of big bang big crunch optimisation in power systems," *Renewable and Sustainable Energy Reviews*, vol. 155, p. 111848, 2022.
- [6] S. Mhanna and P. Mancarella, "An exact sequential linear programming algorithm for the optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 37, no. 1, pp. 666–679, 2021.
- [7] J. A. Momoh, R. Adapa, and M. El-Hawary, "A review of selected optimal power flow literature to 1993. i. nonlinear and quadratic programming approaches," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 96–104, 1999.
- [8] P. Fortenbacher and T. Demiray, "Linear/quadratic programming-based optimal power flow using linear power flow and absolute loss approximations," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 680–689, 2019.
- [9] D. N. d. Silva, B. R. Pereira Júnior, and G. R. M. d. Costa, "A novel milp-newton approach for power flow analysis," *IET Generation, Transmission & Distribution*, vol. 15, no. 3, pp. 518–532, 2021.
- [10] A. F. Zobaa, S. A. Aleem, and A. Y. Abdelaziz, *Classical and recent aspects of power system optimization*. Academic Press, 2018.
- [11] A. Ali, A. Hassan, M. Keerio, N. H. Mugheri, G. Abbas, M. Hatatah, E. Touti, and A. Yousef, "A novel solution to optimal power flow problems using composite differential evolution integrating effective constrained handling techniques," *Scientific Reports*, vol. 14, no. 1, p. 6187, 2024.
- [12] A. A. Z. Diab, A. M. Abdelhamid, and H. M. Sultan, "Comprehensive analysis of optimal power flow using recent metaheuristic algorithms," *Scientific Reports*, vol. 14, no. 1, p. 13422, 2024.
- [13] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2020, pp. 1–5.
- [14] D. Owerko, F. Gama, and A. Ribeiro, "Optimal power flow using graph neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5930–5934.
- [15] S. Zeng, A. Kody, Y. Kim, K. Kim, and D. K. Molzahn, "A reinforcement learning approach to parameter selection for distributed optimal power flow," *Electric Power Systems Research*, vol. 212, p. 108546, 2022.
- [16] M. R. Narimani, D. K. Molzahn, D. Wu, and M. L. Crow, "Empirical investigation of non-convexities in optimal power flow problems," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 3847–3854.
- [17] Z. Kaseb, M. Möller, P. P. Vergara, and P. Palensky, "Power flow analysis using quantum and digital annealers: a discrete combinatorial optimization approach," *Scientific Reports*, vol. 14, no. 1, p. 23216, 2024.
- [18] B. Boddu, M. H. Krishna, N. Kumar, N. Singh, A. M. Lafta *et al.*, "Quantum computing approaches to optimal power flow in complex grid networks," in *2023 International Conference on Power Energy, Environment & Intelligent Control (PEEIC)*. IEEE, 2023, pp. 752–756.
- [19] S. F. Hafshejani, M. M. Uddin, D. Neufeld, R. Benkoczi, and D. Gaur, "Quantum algorithms for optimal power flow," in *2025 15th International Conference on Power, Energy, and Electrical Engineering (CPTEE)*. IEEE, 2025, pp. 295–300.
- [20] F. Gao, G. Wu, S. Guo, W. Dai, and F. Shuang, "Solving dc power flow problems using quantum and hybrid algorithms," *Applied Soft Computing*, vol. 137, p. 110147, 4 2023.
- [21] F. Amani and A. Kargarian, "Quantum optimization for optimal power flow: Cvqls-augmented interior point method," *IEEE Transactions on Smart Grid*, 2025.
- [22] F. Feng, Y. Zhou, and P. Zhang, "Quantum power flow," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3810–3812, 2021.
- [23] F. Feng, Y.-F. Zhou, and P. Zhang, "Noise-resilient quantum power flow," *iEnergy*, vol. 2, no. 1, pp. 63–70, 2023.
- [24] Z. Kaseb, M. Möller, G. T. Balducci, P. Palensky, and P. P. Vergara, "Quantum neural networks for power flow analysis," *Electric Power Systems Research*, vol. 235, p. 110677, 2024.
- [25] P. Pareek, A. Jayakumar, C. Coffrin, and S. Misra, "Limitations of fault-tolerant quantum linear system solvers for quantum power flow," *IEEE Transactions on Power Systems*, 2025.
- [26] J. Liu, H. Zheng, M. Hanada, K. Setia, and D. Wu, "Quantum power flows: From theory to practice," *Quantum Machine Intelligence*, vol. 6, no. 2, p. 55, 2024.
- [27] M. R. Magar, D. Pandit, D. Nguyen, and N. Nguyen, "Dc optimal power flow in unit commitment using quantum computing: An admm approach," in *2024 56th North American Power Symposium (NAPS)*. IEEE, 2024, pp. 1–6.
- [28] C. Hartmann, J. Zhang, C. D. G. Calaza, T. Pesch, K. Michielsen, and A. Benigni, "Quantum annealing based power grid partitioning for parallel simulation," *IEEE Transactions on Power Systems*, 2025.
- [29] E. Pelofske, A. Bärtschi, and S. Eidenbenz, "Short-depth qaoa circuits and quantum annealing on higher-order ising models," *npj Quantum Information*, vol. 10, no. 1, p. 30, 2024.
- [30] C. C. McGeoch, *Adiabatic quantum computation and quantum annealing: Theory and practice*. Springer Nature, 2022.
- [31] T. Morstyn, "Annealing-based quantum computing for combinatorial optimal power flow," *IEEE Transactions on Smart Grid*, vol. 14, no. 2, pp. 1093–1102, 2022.
- [32] R. Wagle, G. Tricarico, A. F. S. Melo, V. Rosero-Morillo, A. Shukla, F. Gonzalez-Longatt *et al.*, "Real-time cyber-physical power system testbed for optimal power flow study using co-simulation framework," *IEEE Access*, vol. 12, pp. 150 914–150 929, 2024.
- [33] R. Wagle, P. Sharma, C. Sharma, M. Amin, J. L. Rueda, and F. Gonzalez-Longatt, "Optimal power flow-based reactive power control in smart distribution network using real-time cyber-physical co-simulation framework," *IET Generation, Transmission & Distribution*, vol. 17, no. 20, pp. 4489–4502, 2023.
- [34] S. Chanda, M. Mohanpurkar, and R. Hovsapian, "Architecture for quantum-in-the loop real-time simulations for designing resilient smart grids," in *2023 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*. IEEE, 2023, pp. 1–6.
- [35] S. Talkington, D. Turizo, S. Grijalva, J. Fernandez, and D. K. Molzahn, "Conditions for estimation of sensitivities of voltage magnitudes to complex power injections," *IEEE Transactions on Power Systems*, vol. 39, pp. 478–491, 1 2024.
- [36] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, no. 3, pp. 859–869, 1974.
- [37] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1 1996, vol. 7632.
- [38] M. A. Schalkers and M. Möller, "Momentum exchange method for quantum boltzmann methods," *Computers & Fluids*, vol. 285, p. 106453, 2024.