Applying Quantum Computing to Simulate Power System Dynamics' Differential-Algebraic Equations

Huynh T. T. Tran[1], Hieu T. Nguyen[1], Long Thanh Vu[2], and Samuel T. Ojetola[3]
[1]*Department of Electrical & Computer Engineering, North Carolina A&T State University*
[2]*Energy & Environment Directorate, Pacific Northwest National Laboratories (PNNL)*
[3]*Electric Power System Research, Sandia National Laboratories*
htran@aggies.ncat.edu, htnguyen1@ncat.edu, thanhlong.vu@pnnl.gov, sojetol@sandia.gov

***Abstract*—Power system dynamics are generally modeled by high dimensional nonlinear differential-algebraic equations due to a large number of generators, loads, and transmission lines. Thus, its computational complexity grows exponentially with the system size. This paper demonstrates the potential use of quantum computing algorithms to model the power system dynamics. Leveraging a symbolic programming framework, we equivalently convert the power system dynamics' differential-algebraic equations (DAEs) into ordinary differential equations (ODEs), where the data of the state vector can be encoded into quantum computers via amplitude encoding. The system's nonlinearity is captured by Taylor polynomial expansion, the quantum state tensor, and Hamiltonian simulation, whereas state variables can be updated by a quantum linear equation solver. Our results show that quantum computing can simulate the dynamics of the power system with high accuracy, whereas its complexity is polynomial in the logarithm of the system dimension. Our work also illustrates the use of scientific machine learning tools for implementing scientific computing concepts, e.g., Taylor expansion, DAEs/ODEs transform, and quantum computing solver, in the field of power engineering.**

***Keywords*—Power system dynamics, DAEs, ODEs, quantum computing, scientific machine learning.**

I. Introduction

Power system dynamics are generally modeled by a large number of differential-algebraic equations (DAEs) due to a large number of generators, loads, and transmission lines that form the network [1]. Specifically, these DAEs include a set of ordinary differential equations (ODEs) modeling the dynamics of synchronous generators along with algebraic nonlinear equations of power flow balances and Kirchhoff voltage laws for individual buses in the network. Solving such DAEs is challenging as they first need to be transformed into standard ODE forms, which is not a trivial task [2]. Then, the obtained ODEs can be solved by numerical discretization such as Euler, the Runge-Kutta, and the backward differentiation formula methods in classical computers [3] whose implementation scales exponentially with problem size and the linearization of the nonlinear terms embedded in the complex power system dynamics, leading to high computational complexity [3], [4].

Quantum computer has a different way of performing computation and possesses algorithmically superior scaling for certain problems, e.g., the Harrow-Hassidim-Lloyd (HHL) algorithm for solving linear equations [5]. The HHL algorithm is based on amplitude encoding, where a system with n variables can be represented as an n-level of quantum state $|\phi\rangle = \sum_j^n \phi_j |j\rangle$, in which ϕ_j is the amplitudes of computational basis states $|j\rangle$. Since the number of required qubits equals $\log_2(n)$, the algorithm provides an exponential memory advantage over the classical computing method [3]. Extending quantum linear equation solvers to tackle high-dimensional systems of linear ODEs are studied in [6]–[8]. Basically, they construct the quantum states proportional to the solution of the block-encoded n-dimensional system of linear equations, thus offering the prospect of rapidly characterizing the solutions of high-dimensional linear ODEs.

Unlike linear ODEs studied in [6]–[8], the power systems dynamics pose high-dimensional nonlinear DAEs. This paper studies the potential of quantum computing algorithms for simulating power system dynamics leveraging recent advances in scientific computing frameworks. We leverage symbolic programming packages from Julia/SciML [9], particularly ModelingToolkit [10], to implement and transform power systems' DAEs into an equivalent set of ODEs using the Pantelides based index reduction. Then, we employ Leyton's quantum algorithm for nonlinear cases of ODEs [11]. Specifically, the obtained nonlinear ODEs are linearized by Taylor expansion as a set of polynomial functions of state variables. By storing multiple copies of quantum states, the nonlinearities embedded in polynomial functions are then captured by amplitudes of the tensor of quantum states. Then, state variables, which are updated in the classical computer according to the Euler method, are now equivalently updated using the HHL algorithm. The complexity, however, is polynomial to the logarithm of the system dimension.

The rest of the paper is organized as follows. The mathematical model of power systems dynamics is presented in Section II. The classical method for simulating power system dynamics is presented in Section III. Section IV summarizes the quantum method to simulate the power system dynamics. Section V illustrates the procedure to implement the power system dynamics into Julia, subsequently transforming it to ODEs by ModelToolkit, and solving it by interfacing it with quantum simulator Yao/QuDiffEq. Section VI presents numerical studies on the SMIB and the three-machine nine-bus test systems. Section VII concludes the paper.

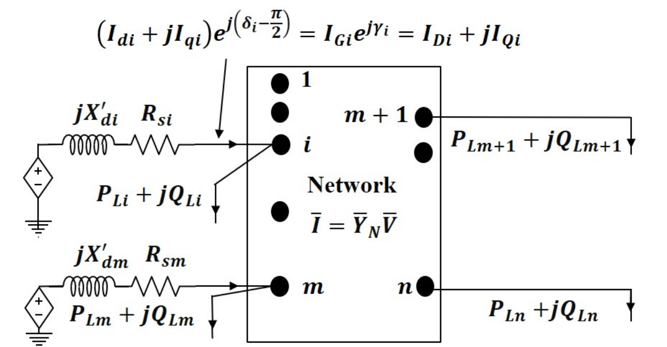


Fig. 1: Interconnection of m synchronous machine dynamic circuits and n buses with network admittance matrix $\bar{Y}_N$ [1]

II. Mathematical model

Figure 1 presents a dynamic network modeling of a generic electric power network that consists of n buses with m generators interconnected via network admittance matrix $\overline{Y}_N$ in which $Y_{ik}\angle\alpha_{ik}$ represents the ik$^{\text{th}}$ element of $\bar{Y}_N$. P_{Li} and Q_{Li} as active and reactive power demands at buses i. Generation buses are indexed from 1 to n whereas buses indexed $m+1,\ldots,n$ are load buses. Consequently, the dynamics of the electric power grid generally contain (i) a set of ordinary differential equations (ODEs) modeling the system dynamics of synchronous generators and (ii) a set of algebraic equations modeling generators' stator voltage equations and power flow balances in the network [12].

A. Dynamics of Generators

In Figure 1, the dynamic circuit of the generator in the two-axis d-q coordinates is considered as a constant voltage source behind impedance $(R_{si} + jX'_{di})$ with an IEEE-Type I exciter [1]. Its dynamics are represented by the following ODEs:

$$T'_{qoi}\frac{\mathrm{d}E'_{di}}{\mathrm{d}t} = -E'_{di} + (X_{qi} - X'_{qi})I_{qi} \tag{1a}$$

$$T'_{doi}\frac{\mathrm{d}E'_{qi}}{\mathrm{d}t} = -E'_{qi} - (X_{di} - X'_{di})I_{di} + E_{fdi} \tag{1b}$$

$$\frac{\mathrm{d}\delta_i}{\mathrm{d}t} = \omega_i - \omega_s \tag{1c}$$

$$\frac{2H_i}{\omega_s}\frac{\mathrm{d}\omega_i}{\mathrm{d}t} = T_{Mi} - E'_{di}I_{di} - E'_{qi}I_{qi} - (X'_{qi} - X'_{di})I_{di}I_{qi} - D_i(\omega_i - \omega_s) \tag{1d}$$

$$T_{Ei}\frac{\mathrm{d}E_{fdi}}{\mathrm{d}t} = -(K_{Ei} + S_{Ei}(E_{fdi}))E_{fdi} + V_{Ri} \tag{1e}$$

$$T_{Fi}\frac{\mathrm{d}R_{fi}}{\mathrm{d}t} = -R_{fi} + \frac{K_{Fi}}{T_{Fi}}E_{fdi} \tag{1f}$$

$$T_{Ai}\frac{\mathrm{d}V_{Ri}}{\mathrm{d}t} = -V_{Ri} + K_{Ai}R_{fi} - \frac{K_{Ai}K_{Fi}}{T_{Fi}}E_{fdi} + K_{Ai}(V_{refi} - V_i) \tag{1g}$$

$$\forall i = 1,\ldots,m.$$

The state variables of the generator in bus i include d-axis and q-axis internal voltages, E'_{di} and E'_{qi}, the rotor angle, δ_i, and the angular velocity, ω_i. Their dynamics are captured in equations (1a)-(1d) respectively. The state variables of the exciter of the generator i include the scaled field voltage E_{fdi}, the stabilizer rate feedback, R_{fi}, and the scaled input to the main, V_{Ri}, [13]. Their dynamics are captured in equations (1e)-(1g) respectively. The generator parameters used in (1) include d-axis and q-axis transient open-circuit time constants, T'_{do} and T'_{qo}, the synchronous and transient reactances in d- and q-axis, X_d, X'_d, X_q and X'_q, the synchronous angular speed, ω_s, the inertia constant, H, the mechanical torque, T_M, and the damping coefficient, D. The parameters of the exciter and voltage stabilizer of the generator i used in (1) include the exciter time constant, T_{Ei}, the exciter gain, K_{Ei}, exciter saturation, S_{Ei}[1], the rate feedback time constant, T_{Fi}, the rate feedback gain, K_{Fi}, the amplifier time constant, T_{Ai}, and the amplifier gain, K_{Ai}.

B. Kirshoff voltage and network power flow equations

Algebraic variables include the stator currents I_{di} and I_{qi} in d-q coordinates of m generators in buses $1,\ldots,m$, the magnitudes V_i and phase angle θ_i of n nodal voltages $(V_i\angle\theta_i)$ in buses $1,\ldots,n$. For generation buses $(i = 1,\ldots,m)$, I_{di}, I_{qi}, and $V_i\angle\theta_i$ are coupled with the state variables of internal generators' voltages, E'_{di} E'_{qi}, by Kirchhoff voltage law as:

$$E'_{di} - V_i\sin(\delta_i - \theta_i) - R_{si}I_{di} + X'_{qi}I_{qi} = 0 \tag{2a}$$

$$E'_{qi} - V_i\cos(\delta_i - \theta_i) - R_{si}I_{qi} - X'_{di}I_{di} = 0 \tag{2b}$$

$$\forall i = 1,\ldots,m,$$

where R_{si} denotes the stator resistance and recall that X'_{di} denotes the transient reactance of generator i. Also, the active/reactive power balances for these generation buses are:

$$I_{di}V_i\sin(\delta_i - \theta_i) + I_{qi}V_i\cos(\delta_i - \theta_i) - P_{Li} = \sum_{k=1}^{n} V_iV_kY_{ik}\cos(\theta_i - \theta_k - \alpha_{ik}) \tag{3a}$$

$$I_{di}V_i\cos(\delta_i - \theta_i) - I_{qi}V_i\sin(\delta_i - \theta_i) - Q_{Li} = \sum_{k=1}^{n} V_iV_kY_{ik}\sin(\theta_i - \theta_k - \alpha_{ik}) \tag{3b}$$

$$i = 1,\ldots,m,$$

where $I_{di}V_i\sin(\delta_i - \theta_i) + I_{qi}V_i\cos(\delta_i - \theta_i)$ and $I_{di}V_i\cos(\delta_i - \theta_i) - I_{qi}V_i\sin(\delta_i - \theta_i)$ respectively are the active and reactive power generated by the generator at bus i.

Similarly, the active and reactive power balance equations for $n - m$ remaining load buses are:

$$-P_{Li} = \sum_{k=1}^{n} V_iV_kY_{ik}\cos(\theta_i - \theta_k - \alpha_{ik}) \tag{3c}$$

$$-Q_{Li} = \sum_{k=1}^{n} V_iV_kY_{ik}\sin(\theta_i - \theta_k - \alpha_{ik}) \tag{3d}$$

$$i = m+1,\ldots,n.$$

[1]Note the exciter saturation S_{Ei} is a function of the scaled field voltage E_{fdi}, i.e., typically $S_{Ei}(E_{fdi}) = A_x e^{B_x E_{fdi}}$

C. DAE compact form

In short, the set of DAEs of power system dynamics can be written in the compact form [1], [12], [13]:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x, y) & \text{(4a)} \\ 0 = g(t, x, y), & \text{(4b)} \end{cases}$$

$$x = [E'_{di}, E'_{qi}, \delta_i, \omega_i, E_{fi}, R_{fi}, V_{Ri} \quad (i = 1, \ldots, n)]^\top,$$
$$y = [I_{di}, I_{qi}, V_i, \theta_i \quad (i = 1, \ldots, n)]^\top,$$

where vector f and g are nonlinear functions of state variables x, algebraic variables y, and time t. Equation (4a) composes ODEs modeling the dynamics of m generators (1) whereas equation (4b) encapsulates Kirchhoff voltage laws (2) and (3).

III. Classical Methods for Simulating Power System Dynamics

Given a large number of algebra equations and variables in (4b), the DAE form (4) can be of index greater than two, which is very difficult to solve [14]. Thus, they need to be equivalently transformed into the following ODEs [2]:

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \hat{f}(t, z), \tag{5}$$

where z is a new set of state variables, so it can be simulated by various methods [2]. Such transformation, unfortunately, is a challenging task due to the complexity of nonlinear power flow equations embedded in (4b). If the Jacobian $\frac{\partial g}{\partial y}$ is nonsingular, i.e., the power flow and the Kirchhoff voltage equations have a (unique) solution, which generally holds for normal operations of power systems. Based on the implicit function theorem, there exists a function $\hat{g}$ to equivalently convert (4b) into $y = \hat{g}(t, x)$. Thus, the formulation (4) can be written in the standard form (5) by substituting y by $\hat{g}(t, x)$, i.e., $\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x, \hat{g}(t, x)) = \hat{f}(t, x)$. Unfortunately, we cannot explicitly model the nodal voltages (magnitudes and phases) as explicit functions of generators' state variables. To overcome these issues, this paper employs the Pantelides algorithm, a systematic method for reducing high-indexed DAEs to lower-indexed ones by selectively adding differentiated forms of the equations already in the system[2]. This enables us to convert the power system dynamics (4) into (5) where z composes of original state variables x and parts of variables y.

The formulation (5) is now in the standard form of ODEs and can be solved by numerical discretization methods [2] in classical computers such as the Forward Euler method:

$$\frac{z(t+\Delta) - z(t)}{\Delta} \approx \hat{f}(z(t) \iff z(t+\Delta) = z(t) + \Delta \hat{f}(z(t)), \tag{6}$$

which is indeed a linear equation.

[2]https://ptolemy.berkeley.edu/projects/embedded/eecsx44/lectures/Spring2013/modelica-dae-part-2.pdf

IV. Quantum Computing Method for Simulating Power System Dynamics

The linear equation (6) implies the potential application of advances in quantum computing for solving linear equations [5], for tackling the power system dynamics of a large-scale network. This includes the following steps [11].

A. Data encoding:

Let N be the length of the state vector z. Using amplitude encoding, we can encode z into amplitudes of an n-qubit quantum state with $n{=}\log_2 N{+}1$. Since the classical information z now forms the amplitudes of a quantum state, the input z needs to be normalized such that $\sum_{j=1}^{N} |z_j|^2 = 1$. Thus, the original state vector z becomes the state of n-qubits, i.e.,

$$z \longrightarrow |z\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}} \sum_{j=1}^{n} z_j |j\rangle, \tag{7}$$

where $|j\rangle$ is the computational basis for the Hilbert space. Thus, in principle, the forward Euler equation (6) can be equivalently implemented in a quantum computer as follows:

$$|z(t+\Delta)\rangle = |z(t)\rangle + \Delta |\hat{f}(z(t))\rangle. \tag{8}$$

The remaining issue in (8) is how to construct the quantum state $|\hat{f}(z(t))\rangle$ with n-qubits from $|z(t)\rangle$:

$$z \longrightarrow |z\rangle \longrightarrow |\hat{f}(z(t))\rangle. \tag{9}$$

This can be achieved in quantum computers by combining Taylor expansion and mathematical tensor, described next.

B. Reformulate the state function using Taylor Expansion

The second-order Taylor polynomial approximation of the j-th element in $\hat{f}(z)$ at the point $\bar{z}$ (operating or initial state of the power systems such that $\hat{f}_j(\bar{z})$ is already determined) is as follows:

$$\hat{f}_j(z(t)) \approx \hat{f}_j(\bar{z}) + \frac{\partial \hat{f}_j(z(t))}{\partial z}(z(t) - \bar{z}) + \frac{1}{2}(z(t) - \bar{z})^\top \nabla^2 \hat{f}_j(z(t))(z(t) - \bar{z}) + O(\Delta^2), \tag{10}$$

which can be rewritten as follows:

$$\hat{f}_j(z(t)) = \underbrace{\hat{f}_j(\bar{z})}_{a^j_{0,0}} + \sum_{k=1}^{N} a^j_{0,k} z_k(t) + \sum_{v,k=1}^{N} a^j_{v,k} z_v(t) z_k(t)$$

where $a^j_{v,k}, v = 0, \ldots, N, k = 0, \ldots N$ are coefficients resulted from Taylor expansion. Thus, by adding extra variable $z_0 = 1$, we have the following compact form of $\hat{f}_j$:

$$\hat{f}_j(z(t)) = \sum_{v,k=0}^{N} a^j_{v,k} z_v(t) z_k(t). \tag{11}$$

Computing $\hat{f}_j(z(t))$ requires encoding all monominals $z_v(t)z_k(t), \forall v, k$.

C. Nonlinear transformation of state function

A Quantum computer captures all values $z_v(t)z_k(t), \forall v, k$ in the probability amplitudes of the tensor product:

$$|z\rangle|z\rangle = \frac{1}{2}\sum_{v,k=0}^{N} z_v z_k |vk\rangle, \quad (12)$$

We now need an operator A to assign corresponding coefficients, $a^j_{v,k}$, to individual monomial $z_v z_k$:

$$A = \sum_{j,v,k=0}^{N} a^j_{v,k}|j0\rangle\langle vk|. \quad (13)$$

Acting on $|z\rangle|z\rangle$ gives us the information of $\hat{f}(z)$:

$$A|z\rangle|z\rangle = \frac{1}{2}\sum_{j,v,k=0}^{N} a^j_{v,k} z_v z_k |j\rangle|0\rangle = \frac{1}{\sqrt{2}}|\hat{f}(z)\rangle|0\rangle. \quad (14)$$

The remaining question is how to simulate $A|z\rangle|z\rangle$ in a quantum computer, which is described next.

D. Simulation in a quantum computer

In order to simulate $A|z\rangle|z\rangle$, we can simply set up the following Hamiltonian using the well-known trick of von-Neumann measurement prescription [15]:

$$H = iA \otimes |1\rangle_P\langle 0| - iA^\dagger \otimes |0\rangle_P\langle 1|, \quad (15)$$

where $A^\dagger$ is the adjoin of A and P is the qubit pointer. If we simulate H in the quantum computer with the initial state $|z\rangle|z\rangle|0\rangle_P$, the quantum system will evolve according to H for a time ϵ and reach the steady state:

$$|\Psi\rangle = e^{-i\epsilon H}|z\rangle|z\rangle|0\rangle = \sum_{j=0}^{\infty}\frac{(-i\epsilon H)^j}{j!}|z\rangle|z\rangle|0\rangle$$
$$= |z\rangle|z\rangle|0\rangle + \epsilon A|z\rangle|z\rangle|1\rangle - \ldots, \quad (16)$$

where the second term contains $A|z\rangle|z\rangle$. In other words, we simply need to measure the state quantum state Ψ of the n-qubits and post-select on $|1\rangle$ to produce $|\hat{f}(z)\rangle$. As $|\hat{f}(z)\rangle$ can be simulated, the Forward Euler linear equation (6) can be solved by the HHL algorithm [5].

V. Implementation with Scientific Machine Learning and Quantum Simulation Tools

We leverage recent advances in scientific machine learning tool-kits to implement the aforementioned scientific computing algorithms, particularly mathematical concepts such as Taylor expansion, ODEs/DAEs, and quantum algorithms. Figure 2 shows the implementation workflows. The DAEs of power system dynamics are implemented in Julia's computing environment. We then use the ModelToolkit, a symbolic equation-based model system, to convert DAEs to ODEs by index reduction method, i.e., transforming the DAEs into MTK symbolic representation and using *dae_index_lowering* to generate the index-1 form, particularly formulation (4). The ODE model obtained by MTK can be either solved by ODE numerical methods in a classical computer or quantum

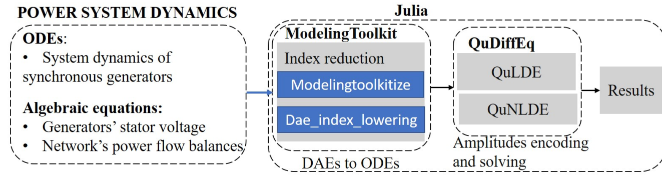


Fig. 2: Implementation's workflow.

computing solvers in the QuDiffEq package that employs Yao as a quantum simulator. We employ the QuNLDE solver to linearize the ODE systems and perform amplitude encoding and Hamiltonian simulation.

VI. Numerical Results

We simulate the power system dynamics through two case studies, particularly (i) a single-machine infinite bus system and (ii) the Western System Coordinating Council (WSCC) three-machine nine-bus system. The time step for linearization is set as 0.01s. The simulation setup is based on [13] and [1] (Chapter 7, Example 7.1), respectively. We do not model the dynamics of the governor and ignore the limit constraint on V_{Ri} in this current work. We compare the simulation gap between the quantum and classical methods using the root means square error.

A. Single machine infinite bus system

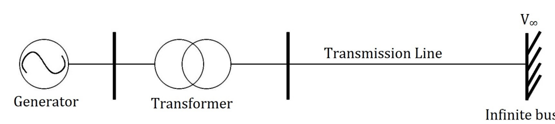
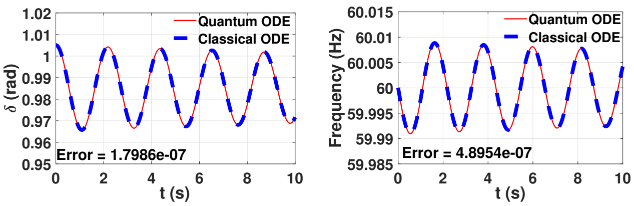


Fig. 3: Single machine infinite bus system [13]

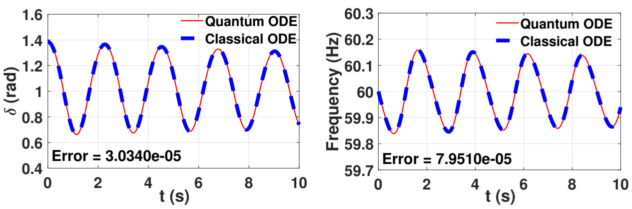
Figure 3 depicts a single-machine infinite bus (SMIB) system, whose dynamics are in the simple ODE form [13]:

$$\begin{cases} \dfrac{d\delta}{dt} = \omega - \omega_s & (17a) \\ \dfrac{2H}{\omega_s}\dfrac{d\omega}{dt} = P_M - \dfrac{E'_q V}{X_{d\Sigma}}\sin(\delta) - \dfrac{D(\omega - \omega_s)}{\omega_s} & (17b) \end{cases}$$

The parameters used in (17) include the initial constant, H = 15 kWs/kVA, synchronous angular speed, ω_s = 376.9911 rad/s, the mechanical power input of the synchronous machine, P_M = 1 p.u., the voltage magnitude on the machine by q-axis, E'_q = 1.0566 p.u., the voltage of infinite bus, V = 1 p.u., the total reactance by d-axis of the machine, $X_{d\Sigma}$ = 0.8805 p.u., and the damping constant, D = 1.2. The SMIB system is equilibrium at δ_0 = 0.9851 rad and $\omega_0 = \omega_s$. We consider two cases when the initial generator's angle departs from its equilibrium a) small disturbance, which $\Delta\delta = 0.02$, the new $\delta_0 = 1.0051$ and b) large disturbance, which $\Delta\delta = 0.4$, the new $\delta_0 = 1.3851$, while keeping $\omega_0 = \omega_s$. The system oscillations derived from the initial values are shown in Figure 4. The solid line in Figure 4 represents the values of quantum computing, and the dashed line is the values of the classical method. It can be seen that for both situations, quantum computing has the same value as the classical method with an almost zero error gap. The obtained results are also in accordance with the result reported in [13].



(a) Small disturbance



(b) Large disturbance

Fig. 4: Results of SMIB system

B. WSCC three-machine nine-bus system

The WSCC system is shown in Figure 5 whose parameters and its equilibrium can be found in [1]. Its DAE system has 21 state variables and 24 algebraic variables, which are converted into ODEs with 39 state variables using Modelingtoolkit. At $t = 0$, we consider an initial overvoltage occurs in the generator G1, i.e., its internal q-axis voltage, $E'_{q,1}(0)$, is 0.45pu larger than the normal value, and its angle, $\delta_1(0)$, is 0.01pu increased from the equilibrium. Then, at $t = 20s$, we consider the disturbances of load with two cases:

- *Small disturbance:* demands at B5, B6, and B7 increase by $0.2 + j0.05$pu, 0.15pu, and 0.2pu respectively.
- *Large disturbance:* demands at B5, B6, and B7 increase by $0.5+j0.25$pu, $0.3+j0.05$pu, and $0.4+j0.05$pu respectively.

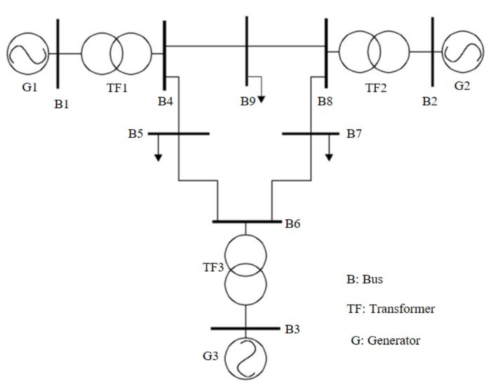
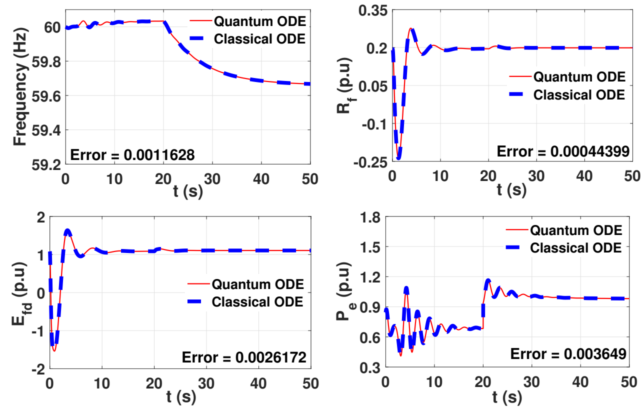
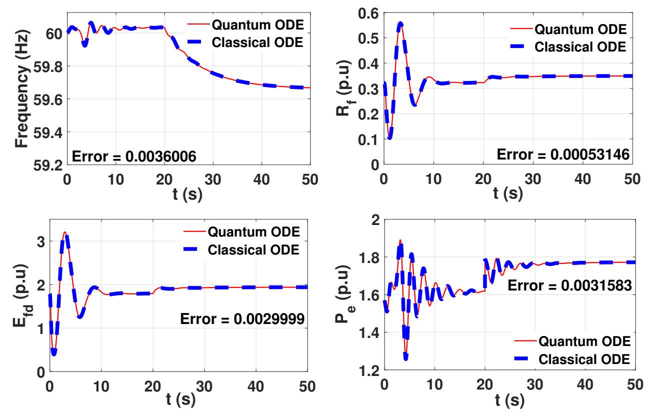


Fig. 5: WSCC three-machine nine-bus system [1], [13]

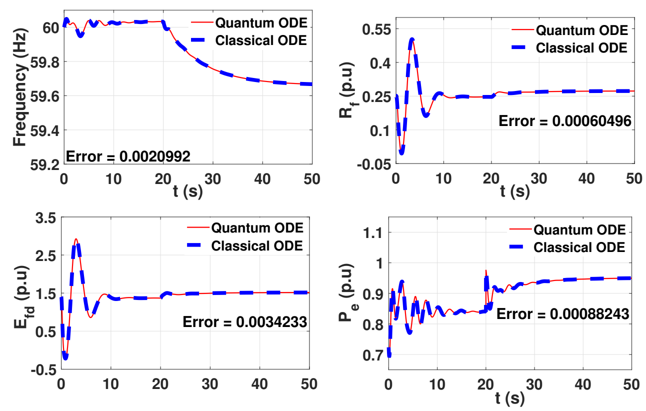
Figures 6 and 7 show the changes of variables (frequency, $f_i = \frac{\omega_i}{2\pi}$, the scaled field voltage, E_{fdi}, and the stabilizer rate feedback R_fi of the exciter system, and active power output, P_{ei} corresponding to generators G1, G2, and G3



(a) Generator 1



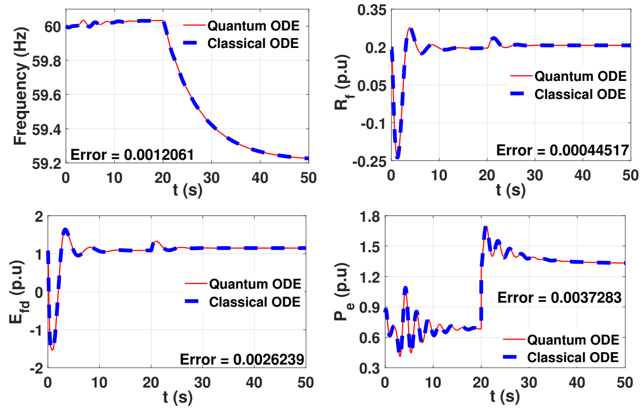
(b) Generator 2



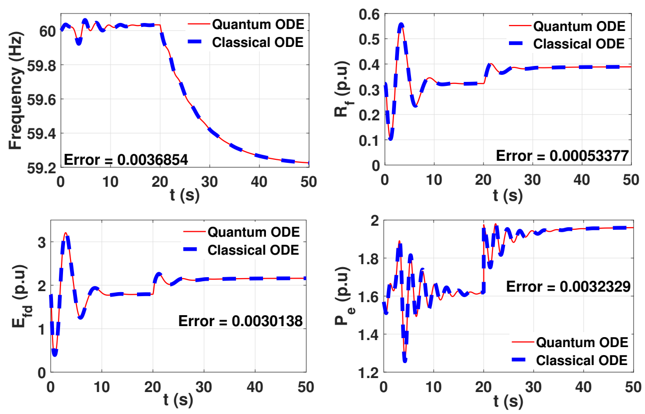
(c) Generator 3

Fig. 6: The generic DAE model with small changing loads.

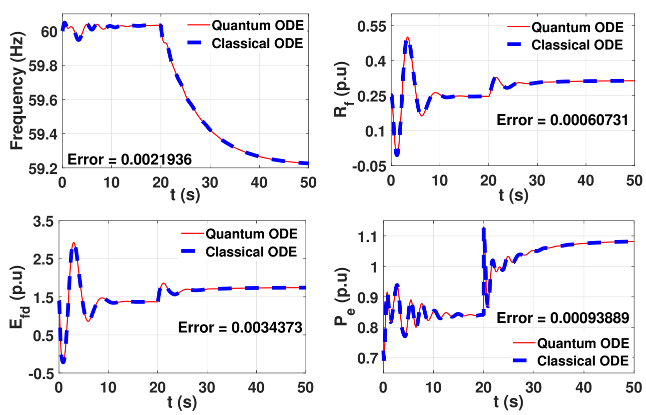
$(i = 1, 2, 3)$ respectively). It is observed that $t = 0$s to $t = 20$s, all generators' variables oscillate for a few seconds from the initial values before reaching new equilibrium values. A larger change in loads leads to a higher change in generators' variables. Note that in the setting, we do not model the governor, so the mechanical torque T_M in (1d) is fixed. Thus, the frequency reduces and converges to a new equilibrium point of less than 60Hz due to the increase in load as expected. In both periods ($0 \leq t < 20$ and $t \geq 20$) of two studies, quantum solvers simulate well the system dynamics, and the gaps between quantum and classical computing methods are



(a) Generator 1



(b) Generator 2



(c) Generator 3

Fig. 7: The generic DAE model with large changing loads.

quite small. These results demonstrate the potential of quantum computing for modeling system dynamics.

VII. Conclusion

This paper presents the method of quantum computing to simulate the power system dynamics, which is typically represented as a set of DAEs and can be converted into a set of ODEs using the index reduction method. Their data can be encoded into quantum computers by using amplitude encoding, which requires a number of qubits as a logarithm of the number of state variables. To this end, the nonlinear ODEs need to be linearized as a set of polynomial functions of state variables to be simulated in quantum computers by amplitudes of tensors and tailored Hamiltonian tricks. The linear update in traditional ODE solvers can be replaced by quantum-based linear equation solvers, such as the HHL algorithm, with proven quantum advantages. Our numerical results demonstrate the potential of quantum computing in modeling power system dynamics with high accuracy. Additionally, we also illustrate the use of scientific machine learning tools, particularly symbolic programming, to implement scientific computing concepts, e.g., Taylor expansion, DAEs/ODEs transform, and quantum computing solver, in the field of power engineering.

VIII. Acknowledgement

This research is supported by the Alfred P. Sloan Foundation #10358 and the NCAT's Intel Foundation Gift.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

References

[1] P. W. Sauer, M. A. Pai, and J. H. Chow, *Power system dynamics and stability: with synchrophasor measurement and power system toolbox*. John Wiley & Sons, 2017.
[2] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. Siam, 1998, vol. 61.
[3] O. Kyriienko, A. E. Paine, and V. E. Elfving, "Solving nonlinear differential equations with differentiable quantum circuits," *Physical Review A*, vol. 103, no. 5, p. 052416, 2021.
[4] J. Stiasny, S. Chatzivasileiadis, and B. Zhang, "Solving differential-algebraic equations in power systems dynamics with neural networks and spatial decomposition," *arXiv preprint arXiv:2303.10256*, 2023.
[5] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical review letters*, vol. 103, no. 15, p. 150502, 2009.
[6] D. W. Berry, "High-order quantum algorithm for solving linear differential equations," *Journal of Physics A: Mathematical and Theoretical*, vol. 47, no. 10, p. 105301, 2014.
[7] A. M. Childs, R. Kothari, and R. 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, 2017.
[8] T. Xin, S. Wei, J. Cui, J. Xiao, I. Arrazola, L. Lamata, X. Kong, D. Lu, E. Solano, and G. Long, "Quantum algorithm for solving linear differential equations: Theory and experiment," *Physical Review A*, vol. 101, no. 3, p. 032307, 2020.
[9] C. Rackauckas and Q. Nie, "Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia," *Journal of Open Research Software*, vol. 5, no. 1, p. 15, 2017.
[10] Y. Ma, S. Gowda, R. Anantharaman, C. Laughman, V. Shah, and C. Rackauckas, "Modelingtoolkit: A composable graph transformation system for equation-based modeling," 2021.
[11] S. K. Leyton and T. J. Osborne, "A quantum algorithm to solve nonlinear differential equations," *arXiv preprint arXiv:0812.4423*, 2008.
[12] D. J. Hill and G. Chen, "Power systems as dynamic networks," in *2006 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2006, pp. 4–pp.
[13] S. Wang, W. Gao, and A. S. Meliopoulos, "An alternative method for power system dynamic state estimation based on unscented transform," *IEEE transactions on power systems*, vol. 27, no. 2, pp. 942–950, 2011.
[14] L. Petzold, "Differential/algebraic equations are not ode's," *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 3, pp. 367–384, 1982.
[15] A. Peres, *Quantum theory: concepts and methods*. Springer, 1997.