On the Duality Between Quantized Time and States in Dynamic Simulation

Liya Huang and Georgios Tzounas, *IEEE Member*

***Abstract*—This letter introduces a formal duality between discrete-time and quantized-state numerical methods. We interpret quantized state system (QSS) methods as integration schemes applied to a dual form of the system model, where time is seen as a state-dependent variable. This perspective enables the definition of novel QSS-based schemes inspired by classical time-integration techniques. As a proof of concept, we illustrate the idea by introducing a QSS Adams–Bashforth method applied to a test equation. We then move to demonstrate how the proposed approach can achieve notable performance improvements in realistic power system simulations.**

***Index Terms*—Time-domain integration, quantized state system, quantum control, duality.**

I. Introduction

The numerical simulation of power system dynamics inherently requires discretization. Classical Runge–Kutta and multistep methods are based on time discretization, where the system dynamics are approximated at a sequence of discrete time points. An alternative paradigm is offered by the family of quantized state system (QSS) methods [1]–[3], where the core idea is to quantize the state variables instead of discretizing time. These methods can offer computational advantages in problems with localized activity, where some states require more frequent updates than others. This letter introduces a new perspective that exploits classical integration techniques to guide the design of novel and useful QSS-based schemes.

The family of QSS methods was systematically introduced by Kofman within the discrete event system specification (DEVS) framework [1]. The development began with the first-order quantized state system (QSS1) method, and was subsequently extended to higher-order variants designed to improve accuracy [2], [3]. Implicit versions were later developed to enhance numerical stability in stiff systems; see, e.g., [4]. The application of QSS methods to power systems has so far been limited. Existing studies have primarily considered the simulation of single converters and synchronous machines connected to an infinite bus [4], [5]. A key reason behind this limited application is the asynchronous nature of QSS methods, where, in principle, each state variable updates at its own rate. This structure complicates their use in large, coupled systems that require coordinating state updates.

In this letter, we revisit QSS methods through a new lens, drawing analogies with classical time-integration schemes, and show how this leads to new QSS-based approaches that can be used in conjunction with standard power system solvers.

The authors are with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, D04V1W8, Ireland.
Corresponding author's e-mail: georgios.tzounas@ucd.ie.

II. Duality of State and Time Discretization

The central idea behind QSS is to approximate the solution of differential equations by discretizing the states rather than time. The simplest and most intuitive variant is the first-order scheme, QSS1. Consider the scalar equation:

$$x'(t) = f(x(t)) \ , \quad x \in \mathbb{R} \tag{1}$$

QSS1 tracks $x(t)$ and updates a piecewise-constant signal $q(t)$ whenever $x(t)$ changes by a fixed amount Δq, called the *quantum*. The signal $q(t)$ acts as a quantized approximation of $x(t)$, initialized as $q(0) = x(0)$, and defined by [1]:

$$q(t) = \begin{cases} x(t) & \text{if } |x(t) - q(t^-)| \geq \Delta q \\ q(t^-) & \text{otherwise} \end{cases}$$

At any time t, the state derivative is evaluated using $q(t)$. That is, (1) is approximated as:

$$x'(t) = f(q(t)) \tag{2}$$

Since $q(t)$ changes only when $x(t)$ deviates from it by more than Δq, the right-hand side remains constant in-between, i.e., between *quantization events (QEs)*. Thus, in QSS1, $x(t)$ is piecewise linear between successive updates of $q(t)$. Let t_k denote the time of the k-th QE. At this time, we have that $q(t_k) = x(t_k)$. Then, for $t \in [t_k, t_{k+1})$, the state evolves as:

$$x(t) = x(t_k) + (t - t_k) f(x(t_k)) \ , \quad t \in [t_k, t_{k+1}) \tag{3}$$

The next QE occurs at $t = t_{k+1}$, when, by definition, the following condition is satisfied:

$$|x(t) - x(t_k)| = \Delta q \tag{4}$$

Substituting (3) in (4) at $t = t_{k+1}$, we get $\Delta t_k |f(x(t_k))| = \Delta q$, where $\Delta t_k = t_{k+1} - t_k$. The integration time step is:

$$t_{k+1} = t_k + \Delta q / |f(x(t_k))| \tag{5}$$

Equation (5) reveals a formal analogy, specifically, it is equivalent to forward Euler method applied to the following *dual system* where time evolves as a state-dependent variable:

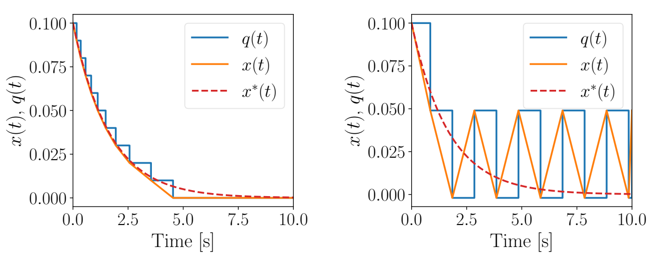
$$t'(x) = \phi(t(x)) \tag{6}$$

with $\phi = 1/|f|$. That is, $t_{k+1} = t_k + \Delta q \ \phi(t_k)$. This dual perspective allows us to interpret QSS methods as discrete approximations of (6). The following implications are relevant: (i) if $x' = 0$, then $t' \to \infty$. In other words, QSS jumps directly to the final simulation time if the system has reached steady state, skipping unnecessary computations. (ii) if $x' \to \infty$, then $t' \to 0$. In other words, QSS becomes inefficient in the presence of high-frequency noise or instability. A trade-off between efficiency and temporal accuracy can be achieved by enforcing a suitable minimum time increment.

The duality above offers valuable insight into the numerical behavior of QSS methods. In particular, since (5) mirrors forward Euler applied to (6), it follows that QSS1 inherits its numerical instability. In this case, however, the instability does not cause unbounded growth of the state error, but rather *destabilizes the QE timing*: when Δq is too large, computed event times may deviate significantly from the point at which the state actually crosses the quantization thresholds. This motivates the design of novel higher-order methods. In this letter, we introduce a second-order Adams–Bashforth variant:

$$t_{k+1} = t_k + \Delta q \left[1.5\phi(t_k) - 0.5\phi(t_{k-1})\right] \tag{7}$$

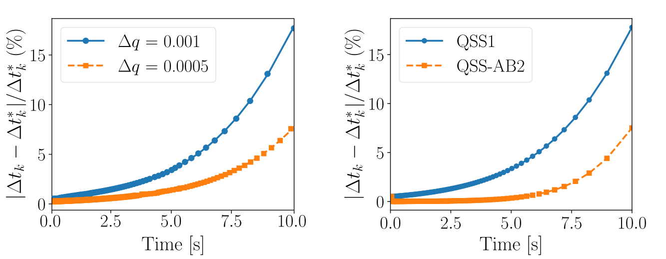
Note that (7) improves the derivative approximation and thus the timing of QEs. Thus, it is conceptually different from, e.g., [2], which focuses on state reconstruction by making $q(t)$ piecewise linear. These two enhancements are independent and can be conveniently combined. For simplicity, in this letter we retain the piecewise-constant (QSS1) structure of $q(t)$.



(a) $\Delta q = 0.01$. (b) $\Delta q = 0.051$.

Fig. 1: Test equation $x' = -0.6x$: QSS1 solution.

As an illustrative example, Fig. 1 shows the numerical solution of the test equation $x'(t) = -0.6x(t)$, with $x(0) = 0.1$, obtained using QSS1. For reference, the exact analytical solution $x^*(t)$ is also included. As expected, increasing Δq leads to larger numerical errors. For sufficiently large Δq (in this case for $\Delta q > 0.05$), QSS1 is no longer able to track the exponential state decay. The effect of the quantum size Δq on the QE timing accuracy of QSS1 is illustrated in Fig. 2a. The relative error $|\Delta t_k - \Delta t_k^*|/\Delta t_k^*$ quantifies the deviation between the computed time step Δt_k and the exact interval Δt_k^* at which $x(t)$ crosses the quantization thresholds. As shown in Fig. 2b, QSS-AB2 significantly reduces this error, providing improved event timing accuracy under the same Δq.



(a) QSS1. (b) $\Delta q = 0.001$.

Fig. 2: Test equation $x' = -0.6x$: Event quantization timing errors.

III. QSS-Based Time Step Control

In the preceding analysis we used a scalar equation to illustrate the key idea of viewing QSS as a time-discretization applied to a dual system, where the roles of time and state are interchanged. We now generalize this perspective to systems of equations, focusing on dynamical models that describe the behavior of power systems. These models can be expressed in the form of explicit differential algebraic equations (DAEs):

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{x}'(t) \\ \boldsymbol{y}'(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t)) \\ \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{y}(t)) \end{bmatrix} \tag{8}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ and $\boldsymbol{y}(t) \in \mathbb{R}^m$ are the state and algebraic variables, respectively. The functions $\boldsymbol{f}$ and $\boldsymbol{g}$ define the differential and algebraic equations, while $\mathbf{I}_n$ is the $n \times n$ identity matrix and $\mathbf{0}_{n\times m}$ the $n \times m$ zero matrix. We associate a local dual system to each differential equation, as follows:

$$t_i'(\boldsymbol{x}, \boldsymbol{y}) = \phi_i(t_i(\boldsymbol{x}, \boldsymbol{y})) \tag{9}$$

where $\phi_i = 1/|f_i(\boldsymbol{x}, \boldsymbol{y})|$ for $i \in \{1, \ldots, n\}$. By applying (7), we obtain n candidate local time steps:

$$\Delta t_{k,i} = \Delta q[1.5\phi_i(t_{k,i}) - 0.5\phi_i(t_{k-1,i})] \tag{10}$$

Implementing such asynchronous system updates raises two issues. First, algebraic equations do not have finite, nonzero time constants, preventing the application of QSS-based stepping to the full DAE model. Second, each QE may affect multiple variables whose dynamics are coupled, but the causal ordering of their updates is not uniquely defined and thus depends on heuristics.

In this letter we address these issues by imposing a global time step for the entire system. In particular, we compute $\Delta t_{k,i}$ for each equation from (10) and select:

$$\Delta t_k = \min\{\Delta t_{k,i}\} \ , \quad i \in \{1, \ldots, n\} \tag{11}$$

This allows combining novel QSS-based schemes like (10) with standard solvers such as the Trapezoidal Method (TM).

We note that in this formulation, QEs are not enforced; instead, Δq acts as a parameter that governs time step-size control. This ensures that no state overshoots its quantization threshold, but can lead to conservative results, as all equations are synchronized with the most sensitive one. To reduce conservativeness and improve efficiency, we allow the quantum size Δq to vary dynamically during the simulation, thereby indirectly controlling the timing of QEs. Specifically, we propose updating the Δq using a PI control rule, directly inspired by classical step-size adaptation strategies based on local error estimates [6], [7]:

$$\Delta q_{k+1} = (\mathrm{tol}/|\sigma_k|)^{\alpha} \left(|\sigma_{k-1}|/|\sigma_k|\right)^{\beta} \Delta q_k \tag{12}$$

where σ_k is the estimated event timing error at t_k; tol is a prescribed error tolerance; and α, β are control parameters.

The combination of (10)–(12), that is, time-step selection based on duality-inspired QSS schemes, standard implicit solvers, and adaptive quantum size control, defines a novel framework that we further explore in the next section.

IV. Case Study

This section presents simulation results based on a dynamic model of the all-island Irish transmission system (AIITS) [8].

The model consists of 1503 buses, 1,851 lines and transformers, 22 synchronous machines, 169 wind power plants and 245 loads. Simulations are carried out with Dome [9], on a computer equipped with an Intel Xeon E3-1245 v5 processor, 16 GB of RAM, and a 64-bit Linux OS.

Two disturbances are considered: 1) a three-phase fault at bus 1238 occurring at $t = 1$ s and cleared by tripping a line connected to the faulted bus after 80 ms; 2) a loss of 142.06 MW load connected to buses 91-102. All tests are carried out using TM as the underlying integration method, considering a simulation duration of 20 s. For each disturbance, the following configurations are evaluated: i) fixed time step; ii) QSS1-Sync, where $\Delta t = \min\{\Delta q \phi_i(t_{k,i})\}$, $i \in \{1, \ldots, n\}$, with constant Δq; iii) QSS-AB2, where Δt_k is defined through (10), (11), with constant Δq; iv) QSS-AB2-Ad, where Δt_k is defined through (10), (11), with adaptive Δq updated via the PI control rule (12), using parameters $\alpha = 0.5$, $\beta = 0$, $\text{tol} = 0.2 \times 10^{-1}$. For QSS-AB2-Ad, the initial Δq is set to 0.2 and a maximum value of 4 is imposed to avoid excessive time steps. The error σ_k is taken as the difference between Δt_k computed with Δq_k and $\Delta q_k/2$, similar to embedded error estimates used in variable-step methods.

TABLE I: Three-phase fault: Performance comparison.

Time step	Quantum	Time [s]	Avg. error [$\times 10^{-5}$]
Fixed [$\Delta t = 0.001$ s]	-	32.2	-
Fixed [$\Delta t = 0.01$ s]	-	5.8	42.92
QSS1-Sync	Fixed [$\Delta q = 0.01$]	9.2	42.76
QSS1-Sync	Fixed [$\Delta q = 0.24$]	5.6	71.89
QSS-AB2	Fixed [$\Delta q = 0.24$]	4.6	1.29
QSS-AB2-Ad	PI control	4.1	0.14

TABLE II: Load loss: Performance comparison.

Time step	Quantum	Time [s]	Avg. error [$\times 10^{-5}$]
Fixed [$\Delta t = 0.001$ s]	-	47.5	-
Fixed [$\Delta t = 0.01$ s]	-	6.3	153.46
QSS1	Fixed [$\Delta q = 0.01$]	10.0	153.81
QSS1	Fixed [$\Delta q = 0.24$]	5.6	189.77
QSS-AB2	Fixed [$\Delta q = 0.24$]	5.1	8.34
QSS-AB2-Ad	PI control	4.3	0.06

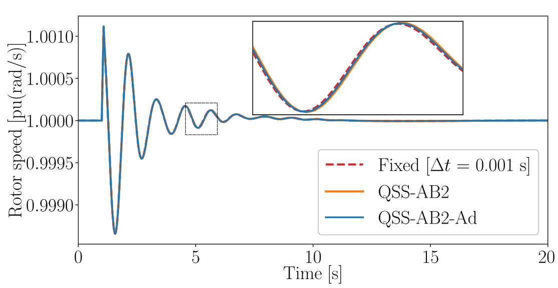


Fig. 3: Three-phase fault: Rotor speed of machine at bus 1237.

Tables I and II report the performance of the examined configurations under the examined disturbances. In both cases, the numerical error is measured for the rotor speed of a representative machine (at bus 1237), using TM with fixed step size 0.001 s as the reference. Increasing the step to 0.01 s yields faster simulations and a moderate error increase. Using QSS1-Sync with $\Delta q = 0.01$ exhibits similar accuracy to TM with $\Delta t = 0.01$ s but results in higher runtime. Increasing Δq to 0.24 improves runtime at the expense of accuracy. In contrast, QSS-AB2 with the same $\Delta q = 0.24$ achieves lower runtime and much lower error compared to TM with $\Delta t = 0.01$ s. Finally, QSS-AB2-Ad dynamically adjusts Δq to further improve both accuracy and runtime.

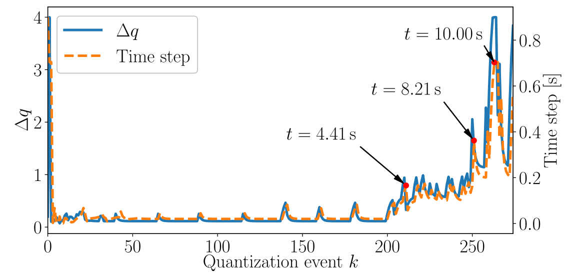


Fig. 4: Three-phase fault: Variations of Δq and Δt, QSS-AB2-Ad.

The rotor speed of the machine at bus 1237 following the fault is shown in Fig. 3. Both QSS-AB2 and QSS-AB2-Ad accurately track the reference solution, consistent with the errors reported in Table I. For the same disturbance, Fig. 4 further illustrates the behavior of QSS-AB2-Ad, showing how Δq and Δt evolve over the course of QEs during the simulation. In particular, both remain small during the transient and increase significantly as the system approaches steady state.

V. Conclusion

This letter introduces a novel perspective on quantized-state simulation by formalizing a duality between time and state discretization. To illustrate this perspective, we define a QSS-based, second-order Adams–Bashforth method and describe how it can be combined with standard implicit power system solvers to achieve significant speedups while maintaining accuracy. Further performance gains through adaptive quantum control are also discussed. The proposed approach opens directions for a new family of numerical methods that integrate time discretization and state quantization.

References

[1] E. Kofman and S. Junco, “Quantized-state systems: A DEVS approach for continuous system simulation,” *Transactions of The Society for Modeling and Simulation International*, vol. 18, no. 3, pp. 123–132, 2001.

[2] E. Kofman, “A second-order approximation for DEVS simulation of continuous systems,” *Simulation*, vol. 78, no. 2, pp. 76–89, 2002.

[3] ——, “A third order discrete event method for continuous system simulation,” *Latin American applied research*, vol. 36, no. 2, pp. 101–108, 2006.

[4] F. Di Pietro, G. Migoni, and E. Kofman, “Improving linearly implicit quantized state system methods,” *Simulation*, vol. 95, no. 2, pp. 127–144, 2019.

[5] N. Gholizadeh, J. M. Hood, and R. A. Dougal, “Evaluation of linear implicit quantized state system method for analyzing mission performance of power systems,” *The Journal of Defense Modeling and Simulation*, vol. 20, no. 2, pp. 159–170, 2023.

[6] G. Söderlind, “Automatic control and adaptive time-stepping,” *Numerical Algorithms*, vol. 31, pp. 281–310, 2002.

[7] K. Huang, Y. Liu, K. Sun, and F. Qiu, “PI-controlled variable time-step power system simulation using an adaptive order differential transformation method,” *IEEE Transactions on Power Systems*, vol. 39, no. 5, pp. 6332–6344, 2024.

[8] G. Tzounas, I. Dassios, M. A. A. Murad, and F. Milano, “Theory and implementation of fractional order controllers for power system applications,” *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4622–4631, 2020.

[9] F. Milano, “A Python-based software tool for power system analysis,” in *IEEE Power & Energy Society General Meeting*, 2013, pp. 1–5.