Electric Power Systems Research 189 (2020) 106752

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Robust event-driven dynamic simulation using power flow

Aayushya Agarwal, Amritanshu Pandey*, Larry Pileggi

Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA

ARTICLE INFO

Keywords:
Dynamic simulation
Event driven
Grid device control
Quasi-steady state simulation
Temporal dependency

ABSTRACT

The power grid is a dynamic system encompassing numerous control devices with varying degrees of time constants, posing a challenge for dynamic simulators to efficiently step through time. We introduce a robust event-driven simulator to mimic dynamic simulation that exploits the inherent temporal sparsity by solving for the quasi-steady state of the grid using frequency-dependent power flow at each time-step. In this methodology, the effect of fast transient control actions is captured within the quasi-steady state while a time-dependent outer loop handles slow transients that dynamically change over time. The event-driven approach offers a fast simulation framework capable of scaling to large systems by taking time-steps based on proceeding events and uses a robust power flow simulator, Simulation with Unified Grid Analyses and Renewables (SUGAR), to solve for the quasi-steady state. We demonstrate the efficacy of this approach by simulating the effects of different control actions including automatic generation control (AGC), automatic voltage (AVR) and over excitation limits on large systems.

1. Introduction

Simulating the power grid is widely accepted as a necessity for operation, planning, and restoration of the grid. However, accurate time-domain simulations often are bottlenecked by slow simulation times for simulating large systems in real-time due to the large range of time scales attributed to the disturbances and control actions occurring in the power grid. A disturbance, such as an outage, propagates the disruption in the form of electromagnetic or electromechanical transients, and ultimately a change in the steady state through various controls taking place. These disturbances have a time scale ranging from microseconds to hourly load changes as shown by Fig. 1[1].

The range of time scale is one of the major challenges in developing an accurate, fast simulator capable of scaling to large systems (up to Eastern Interconnection size systems) while being computationally fast [2]. The grid is inherently a dynamic system constantly changing due to control actions each with a wide range of time constants. Tools combat the challenge of simulating this dynamic system by narrowing the focus to a range of time scale as well as by often compromising speed for accuracy. The result is a distinct set of tools, often incompatible with each other, that studies the behavior of the system in the vicinity of a specific time scale, ranging from steady state in power flow to short term dynamics using dynamic simulation. One such example is the suite of DSA tools [3] that offer simulation software to study stability in a range of different time scales. However, one key missing component is studying the long-term dynamics (in tens of minutes or hours) and the effect on steady state of large systems. Contingency planning, black starts and online operations are just some of the applications that benefit from studying this behavior [3]. In these uses-cases, an engineer requires a tool to understand the effect of actions that interact with automatic controllers over the course of minutes to hours. The simulation tools available today are restricted by their intended time scale to be able to simulate such a problem [4].

On one side of the spectrum, power flow is a conventional fast simulation tool, that approximates the steady state of a system and has proven to be scalable to large systems [5,6]. Advances in governor power flow have robustly improved the accuracy of the steady state by incorporating frequency deviation along with frequency control actions into the simulation [7,8]. However, it still ignores the temporal dependencies of the control actions and applies all control action at once increasing the likelihood of incorrect steady-state results [8]. In reality, various control actions within the grid are constantly being enacted with different time constants and demands and supply are constantly in flux. While an existing steady state simulation offers insight into ideal isolated scenarios, it doesn't offer a complete picture of the dynamic system response [9].

Conversely, dynamic simulation offers the accuracy to simulate transients by solving electromechanical differential algebraic equations [9] at each time step. The bottleneck for fast yet accurate transient simulation to long-scale dynamics and large systems is the large degree

* Corresponding author.
E-mail addresses: aayushya@andrew.cmu.edu (A. Agarwal), amritanp@andrew.cmu.edu (A. Pandey), pileggi@andrew.cmu.edu (L. Pileggi).

https://doi.org/10.1016/j.epsr.2020.106752
Received 4 October 2019; Received in revised form 13 April 2020; Accepted 2 August 2020
Available online 19 August 2020

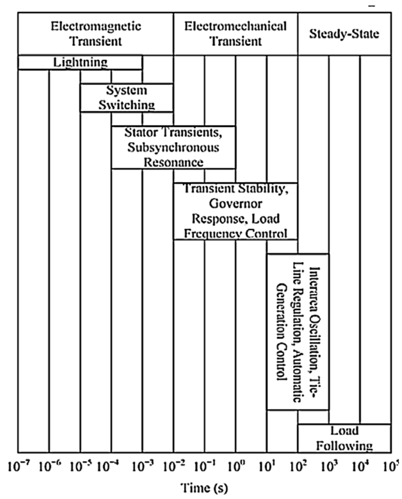

Fig. 1. Time scales of events and control actions occurring in grid [1].

of time constants (ranging from milliseconds to minutes) of control actions in the grid [10–14]. This forces dynamic simulation to take small time steps to respect the fast transients even during latent periods in which the slower transients dominate. A similar issue is seen in circuits community while simulating stiff circuits such as phase locked loops [16]. During these latent periods, dynamic simulation is unable to take larger time steps that would avoid the unnecessary computation and is required to compute over the latent period. Along with the non-linearity of the differential equations, this prevents traditional transient simulation from being able to solve long-term dynamics of large systems without sacrificing speed of simulation [13]. Previous works have dynamically adjusted the time-step during periods of quiescence and have proven to aid in speed, but still fall short in scaling to real life test systems [12]. To combat the issue, real time digital simulators (RTDS) exploit the spatial and temporal sparsity in parts of the grid to parallelize the transient simulation onto multiple cores [14]. While RTDS does show improvements to scalability, it often requires large number of cores to be effective. Other works have exploited the matrix structure of dynamic simulation to readily parallelize across multiple nodes and thereby achieve simulation efficiency [15]. While these transient simulations are necessary in understanding oscillations and periods of fluctuation, the real interest in studying a system often lies in the steady state achieved throughout the span of time.

To bridge the gap between these two formulations, previous works have developed a quasi-steady state model of the transient equations, which study splices of time atat which the grid attains momentary steady state. Under the assumption of zero time-derivative for fast transients, quasi-steady state offers a compromise on accuracy for computational efficiency for long-term analysis [17–19]. While quasi-steady formulation is unable to simulate oscillations and transient effects, it provides a reasonable approximation of the change in steady-state due to these transient disturbances. Previous works have capitalized on the efficiency but have not shown to scale the formulation to large systems such as the Eastern Interconnection (80k+ buses), partly due to long term instabilities due to the quasi-steady state approximations [19]. A recent tool developed by PNNL [DCAT] [20] has capitalized on the use of power flow as an approximation of the quasi-steady state to study the effects of cascading failures. Further work has also used quasi-steady state approximations to simulate the risk associated with cascading outages [21]. In addition, previous work has also used power flow analysis to approximate the quasi steady state [22]. However, the power flow formulation should also account for steady state changes in frequency which would better model frequency control actions in the grid [8]. Nonetheless, the concept of using quasi-steady states as a compromise between accuracy and efficiency offers a potential fast and lightweight solution to simulating the steady state changes due to transient actions.

In this paper, we introduce an event-driven dynamic simulation framework that builds upon a governor power flow, SUGAR [5], as an approximation for the quasi steady-state. The framework incorporates real-time data into the governing power flow tool to provide accurate models that are capable of solving for fast transients at a time splice. To dictate simulation progression in time, the tool exploits the temporal sparsity by being driven by events rather than time steps, thereby achieving high levels of efficiency and scalability that can be shown to simulate large systems of 80k+ buses. The novelty in this work is the event-driven time-step control incorporated with an accurate quasi-steady state solver that allows fast simulation speeds. The event-driven framework has its parallels to circuit-simulators SPECS and ACES [16] that were highly efficient for simulating stiff circuits with time constants that vary with orders of magnitude, similar to the power grid.

The rest of the paper is organized as follows. Section II describes the quasi steady-state approximation that is solved using a governor power flow framework, SUGAR, as described in Section IV. Sections V describes the algorithmic framework for the event-driven platform followed by results in Section VI.

2. Quasi-steady state

The framework for event-driven dynamic analysis solves for a quasi-steady state (QSS) at each time step. The QSS approximation considers a specific time interval that justify the assumptions regarding the dynamic system, modeled as a set of differential algebraic equations as shown below [17]:

$$\dot{x} = f(z_{c,s}, z_{c,f}, \dot{z}_{c,s}, \dot{z}_{c,f}, z_d, x) \tag{1}$$

$$0 = g(z_c, z_{c,f}, \dot{z}_{c,s}, \dot{z}_{c,f}, z_d, x) \tag{2}$$

The system of equations is defined by a vector of state variables (x) that represent the voltages at each capacitor node or the current through an inductor [16]. The response of state variables due to different control actions is a result of the governing control system equations and network dependencies, which are characterized in (1) and (2), respectively. The control actions injected by a source into the network are separated into fast and slow control actions ($z_{c,\ f}$ and $z_{c,\ s}$ respectively) based on their time constants, and, along with their time derivative ($\dot{z}_{c,f}$ and $\dot{z}_{c,s}$) are responsible for a network response. $z_{c,\ f}$ and $z_{c,\ s}$ model various control actions with varying time constants including automatic voltage regulators (AVR), automatic generation control (AGC) and generator ramping. Additionally, discrete state variables (z_d), such as tap changers, change discretely during the time span.

The QSS approximation simplifies the differential state equations above by studying a particular segment in time that is long enough to consider the state variables' response to fast transient control actions ($z_{c,\ f}$) to reach steady state due to the short period of oscillations and transients. However, this time scale is also short enough to approximate the slower control actions to remain stationary ($z_{c,\ s}$, z_d) thereby setting $\dot{z}_{c,s}$ to zero. For example, a slow generator linear ramp is viewed as a constant during a short time period during which the state variables have reached quasi-steady state. The approximation is that during the time the state variables take to settle, particular control actions are slow enough to be viewed as constant during that period. The approximation simplifies the Eqs. (1) and (2) to solve for the quasi-steady state for the state variables as follows:

$$0 = f_{ss}(z_{c,s}, z_{c,f,SS}, \dot{z}_{c,s}, z_d, x_{QSS}) \tag{3}$$

$$0 = g(z_c, z_{c,f,SS}, \dot{z}_{c,s}, z_d, x_{QSS}) \tag{4}$$

$$\dot{z}_{c,s} = 0 \tag{5}$$

where, f_{ss} is a function dictating the steady state values of the state variables, x_{QSS} and response of fast control actions, $z_{c,\ f,\ SS}$. These

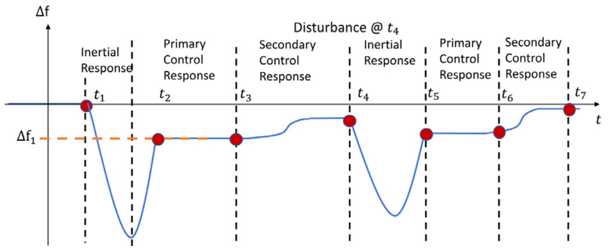

Fig. 2. Frequency response of system with primary and secondary frequency control due to disturbances.

approximations enable the simulation to solve for the states at the times shown by the red markers in Fig. 2. The figure demonstrates the primary and secondary generator responses on a testcase where the quasi-steady states are solved for using the approximations in (3)-(5).

A discrete change in z_d or z_c causes an impulse in the differential Eqs. (1) and (2) as well, that will lead to a new quasi-steady state. For example, discrete events such as tap changers, over- or under- voltage tripping will cause an impulse that propagates into a change in steady state. Even a discrete change in the slower control actions will create a discrete change in the function f_{ss}, thereby creating a new set of differential equations. We define the discrete change as an event, and robustly simulate the quasi-steady state achieved after this event once the oscillations and transients have settled.

3. Power flow quasi-steady state

In order to model the quasi-steady state after an event, the QSS dynamic equations given by (3)-(5) must be solved efficiently and accurately. Power flow has been shown to solve for a steady state of a grid where all the variables have a zero-time derivative [22], as shown in Fig. 3, in which power flow solution, indicated by the red marker, accurately describes the steady state of an induction motor.

The QSS Eqs. (3) and (5) are similarly formulated to find the steady state of the short-term variables (x and y) with the assumption of stationary control variables (z_d and z_c). This implies that power flow can be used to solve for the QSS given stationary long term and discrete variables. However, traditional power flow simulation tools have shown to lack robustness (for large scale and ill-conditioned networks) [5] and physical accuracy (amongst others due to lack of frequency information) [8].

3.1. SUGAR power flow

Traditional power flow simulation relies on good initial conditions close to the final state to converge to a solution. However, in the dynamic event-driven framework, an event may cause the system to drastically change its state, thereby making the pre-event state an inadequate initial condition. SUGAR uses circuit heuristics with state variables as real and imaginary voltages and currents to achieve a level of robustness [5] capable of solving large testcases representing the Eastern Interconnection, regardless of initial conditions.

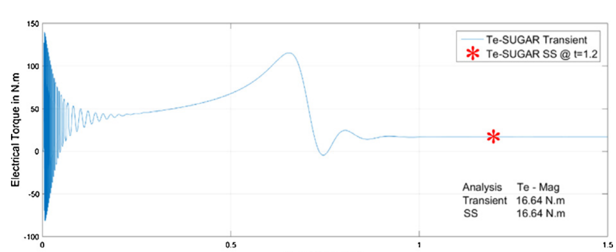

Fig. 3. Steady state of induction motor through transient simulation (blue) and power flow (red marker) [21].

3.1.1. Governor power flow

The robust and scalable circuit formulism in SUGAR, was extended to model the frequency deviation which allows the tool to solve for the steady state due to frequency control actions such as primary and secondary control. Unlike traditional power flow that assumes the grid is operating at nominal frequency, a governor power flow [8] introduced a frequency deviation variable (Δf) to accurately model QSS.

3.1.2. Primary frequency response

Each synchronous generator model in the governor power flow included an additional term ΔP_G representing the change in active power due to primary frequency response (6) [8].

$$\Delta P_G = -\frac{P_R}{R}\Delta f \tag{6}$$

where P_R and R are parameters describing the inertial and droop response of a generator. The model must also respect generator active power limits (7)-(9) to realistically simulate the behavior of primary control.

$$\Delta P_G^{MIN} \leq \Delta P_G^p \leq \Delta P_G^{MAX} \tag{7}$$

$$\Delta P_G^{MIN} = P_G^{MIN} - P_G^{SET} \tag{8}$$

$$\Delta P_G^{MAX} = P_G^{MAX} - P_G^{SET} \tag{9}$$

where P_G^{MAX} and P_G^{MIN} are a generator's active power limits and P_G^{SET} is the set active power.

To incorporate the limits within the power flow framework, [8] introduced a continuous function that modeled the primary frequency response while respecting the active power limits, as shown by Fig. 4. Continuous functions have been previously shown to improve convergence of power flow simulations, as the underlying non-linear numerical solver, Newton-Raphson, uses the first derivatives to project the solution step necessary to solve the set of nonlinear equations.

The continuous function in Fig. 4 models the primary frequency response characteristics in Region 3. The two quadratic regions, Region 2 and 4, are quadratic approximations that allows the function to be continuous.

In addition, slack generators are similarly modeled with primary frequency response (ΔP_S^p) with the added constraint

$$(P_S^{SET} + \Delta P_S^p) = V_R^s I_R^s + V_I^s I_I^s \tag{10}$$

Unlike the slack model in traditional power flows, (10) constrains the slack generator to deliver a finite active power and enforces realistic frequency behavior, which serve to improve the accuracy of the QSS.

3.1.3. Secondary frequency response

While the primary frequency response responds quickly, it does not restore the grid frequency to the nominal value. The secondary frequency response, automatic generation control (AGC), is able to adjust generators to restore the grid to the ideal state. The response is initiated by an error known as the Area Control Error (ACE) (11).

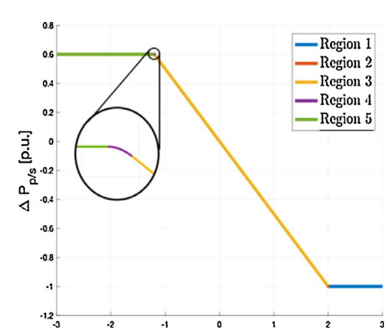

Fig. 4. Continuous primary response model with active power limits.

Accordingly, each participating generator is assigned a participation factor, κ, that controls the adjustment in active power required, ΔP_G^s. Previous work modeled the interaction between ACE and AGC as (11) and (12) respectively [9]:

$$ACE = (\Sigma P_{scheduled} - \Sigma P_{actual}) + \ 10\beta\Delta f_1 \tag{11}$$

$$\Delta P_G^s = \kappa^* ACE \tag{12}$$

where β $[\frac{MW}{0.1Hz}]$is the frequency bias constant, and Δf_1 is the frequency deviation from nominal measured when the secondary control is activated [9]. Also, $(\Sigma P_{scheduled} - \Sigma P_{actual})$ measures the deviation of the net exchange of active power between areas from the scheduled net exchange.

3.2. Load modeling

In order to better approximate the QSS, the introduction of frequency deviation as a variable is required to better model changes in load behavior.

3.2.1. Frequency dependent loads

The governor power flow can be extended to compute the frequency dependence on load values (P_L and Q_L), where these values can either be interpreted as PQ or ZIP models. Previous work [9], demonstrated that the dependence of frequency can be modeled as multiplying the nominal load parameters ($P_{PQ/ZIP}$ and $Q_{PQ/ZIP}$) by a term as given below:

$$P_L = \ P_{PQ/ZIP}(1 + K_{pf}\Delta f) \tag{13}$$

$$Q_L = \ Q_{PQ/ZIP}(1 + K_{qf}\Delta f) \tag{14}$$

where (K_{pf} and K_{qf}) are load specific parameters that describe the linear relation between change in frequency and change in active and reactive power respectively.

3.2.2. UFLS/UVLS

The frequency change term also enables SUGAR to implicitly model UFLS by a continuous function that describes the automatic load relief when the frequency deviation is below a certain threshold. Similarly, UVLS is modeled as a continuous function that relieves the load when the voltage is below a certain threshold [22]. This changes the load active and reactive power as shown:

$$P_L = P_L^{set}(1 - \alpha^{UFLS})\ (1 - \alpha^{UVLS}) \tag{15}$$

$$Q_L = Q_L^{set}(1 - \alpha^{UFLS})(1 - \alpha^{UVLS}) \tag{16}$$

where α^{UFLS} and $\alpha^{UVLS} \in [0, 1]$ are load shedding terms that turn off the load based on the frequency or voltage at the bus, respectively. When the frequency deviation is below a threshold of f_{set} or the voltage at the bus has reached a state below V_{SET}, $\alpha^{UFLS}$and α^{UVLS} should take a value of 1, thereby setting P_L and Q_L to 0. The continuous function for $\alpha^{UFLS}$and α^{UVLS} are is shown in Fig. 5 which is first derivative continuous due to the additional patching regions of Regions 2 and 4.

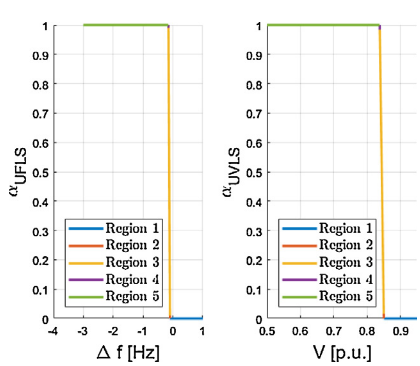

Fig. 5. Continuously differentiable model for UFLS.

3.3. Incorporating direct measurements

The SUGAR framework is also capable of integrating direct real-time measurements from state estimation, thereby further improving the QSS models. Generally, measurements are given in the form of injected or consumed active power, P_M, injected reactive power Q_M, a voltage magnitude V_M and the power factor, $\cos(\theta_M)$ [24]. These measurements have been shown to be easily integrated into SUGAR power flow, which can aid in understanding changes in the state. Any incorporation of direct measurement can be considered as a discrete change in the network, as the network parameters such as load values or network impedances may be updated. In order to stay current in an online study, the event-driven framework will re-compute the QSS each time there is a new state given by the state estimator.

One of the largest inaccuracies within power flow as an approximation of the quasi steady state is the use of PQ and ZIP load models. Previous work [24] have shown that these load models often result in the wrong voltage sensitivity, making it inaccurate during a dynamic simulation. To improve accuracy of quasi steady state, we employ the use of BIG load models [23] which have demonstrated a higher level of accuracy in transients. Comprised of a susceptance, current source and conductance, the BIG load model is able to accurately predict the variation in load due to voltage during the time series, unlike traditional PQ and ZIP models. The BIG model is especially effective for an event-driven dynamic analysis, as quasi steady state is continuously being updated and requires proper sensitivities to simulate the effect of control actions and discrete network changes.

3.4. Control actions in power flow

In order to differentiate between short-term and long-term control actions, we introduce a user-defined maximum time constant, τ_{max}, that is greater than the short-term dynamic control action time constants. By properly defining τ_{max}, we are able to quantify which variables, defined as (z_c) can use the QSS assumption of being stationary. The user defined τ_{max}, will be subject to lower bound, defined as the time constant of the slowest power flow control action, to ensure that power flow can still be a valid QSS approximation. A governor power flow includes essential control actions such as AVR and primary control to represent an accurate QSS. These control actions have an associated time constant, and we can define the maximum of those as:

$$\tau_{PFmax} = \max(\tau_{\text{AVR}},\ \ \tau_{Droop}) \tag{17}$$

In governor power flow simulation, SUGAR considers a user-defined $\tau_{max} \geq \tau_{PFmax}$. Given a τ_{max}, we can now include steady state models of control systems with time constants less than τ_{max} into a single power flow event.

4. Event-driven steps

Governor power flow is able to robustly solve for a quasi-steady state after an impulse change in discrete or long-term variables, denoted as an event. Although it is unable to capture the oscillations or fluctuations before reaching the QSS, the simulation platform focuses on the change in steady states as a result of control actions. Often in long-term dynamic studies engineers are concerned with the effect of transient control actions on steady state of the grid.

Using the QSS approximation, the event-driven dynamic simulation exploits the temporal sparsity of each event to simulate the response of interest. This allows the simulator to overlook transient oscillations by considering fast control actions to reach steady state within a single power flow simulation and focus on the QSS response due to slower events such as discrete changes in the network or other frequency dependent and operator-based control actions. This framework overlooks latent periods in which the grid states are not changing significantly due to slow control actions, and thereby significantly reducing the number of model evaluations.

To efficiently track the progression of events, we introduce an event queue, E_Q, that is sorted by the time at which the event will occur. Slower control actions such as large-mass nuclear generator ramp ups will include two events; one for beginning the control and one for finishing. On the other hand, scheduled discrete changes in z_d, are sorted in the queue by the time they will occur. Other discrete changes that result from the previous QSS, such as over-loading of line, are determined at which time they occur and then inserted into the event queue.

Therefore, after solving for a quasi-steady state, the simulation framework will take a step-in time to the next event in E_Q. It is vital to ensure that the framework does not take a step that skips an event of interest, as the temporal dependency on the skipped event may change subsequent states.

To improve the simulation time required to solve for each QSS, the governor power flow solver considers the previous time step's state as an initial condition for the Newton-Raphson power flow solver. Many events only cause a large change in certain areas of the grid, therefore by using the previous time step state as an initial condition, the voltages of the network that do not experience much change will solve within few iterations. Using the circuit formalism also allows guaranteed convergence through the use of homotopy methods in case the previous state is not a good initial condition [5].

4.1. Backtracking

Over the course of the dynamic simulation, slow control actions are gradually changing over events. This gradual change may induce a discrete event such as over-loading of line. However, the event-driven framework would have skipped over the exact time at which this occurs, resulting in skipping multiple events at once. This violates the temporal dependencies that allow the event-driven framework to accurately simulate long-term dynamics. To amend this, we step back in time till we have found the first event caused by the slow control system.

In order to effectively find the time at which the event occurred we use a binary search to take a time step back equal to half the time step take. This backtracking algorithm efficiently searches for the event, however if it finds that two events have occurred within a period of τ_{max}, then we can approximate the events to have happened simultaneously. The overall flow of the framework is shown in Fig. 6.

5. Results

The algorithm described in Fig. 6 offers a robust framework that simulates a series of quasi-steady states during a pre-defined time interval. The framework steps through time by solving for the quasi-steady state for each event (past the minimum time constant) that causes a sufficient change in the QSS dynamic equations given in (3)-(5). Along with an accurate representation of a QSS using governor power flow, the result is a simulation platform that mimics dynamic simulation with a high level of efficiency.

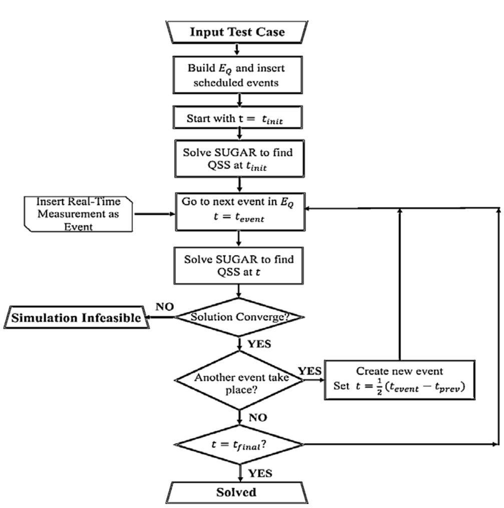

Fig. 6. Event-driven framework algorithm.

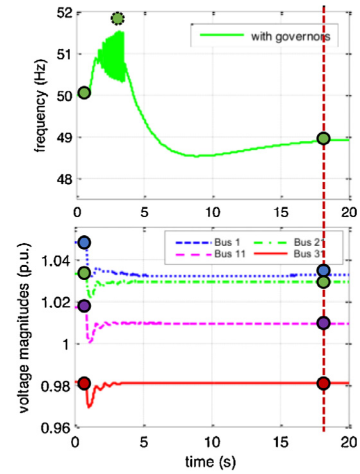

Fig. 7. Comparison of dynamic simulation and event-driven simulation of 39-bus system [25] (© [2017] IEEE). Manuscript results superimposed on top of reference results.

5.1. 39-Bus comparison with dynamic simulation

To justify the accuracy of the event driven framework, we compare the quasi-steady states after a primary response control during a disturbance with a dynamic simulation. Previous work created a dynamic model of IEEE testcases [24], in particular a dynamic simulation of a 39-bus system that experiences a 10% increase in load resulting in a transmission line tripping connecting bus 39. In the experiment, at $t = 0$ s, shown in Fig. 7 the system was in steady state, but after 1 second the load steps up by 10% to increase the frequency by consuming more active power. This resulted in a quasi-steady state, at $t = 4$ with a frequency of 51.8 Hz, there by tripping a line connecting bus 39 (with a load on the bus). The system then reaches a steady state after the transient effects at $t = 17$ s.

The event-driven approach is able to accurately simulate the quasi-steady states reached after the oscillations settled, shown in Fig. 7. Using the governor power flow, the framework predicted the increase in frequency due to the increase in load at $t = 1$ to 51.8 Hz (shown by the dashed green marker), thereby causing a transmission line to trip. The framework backtracked to include this event with the first event, and together caused an overall frequency decrease to reach a steady state at $t = 17$ s (indicated by the solid green marker). The markers over-layed on the transient analysis performed by [25] in Fig. 7 indicate the quasi-steady states reported from the event-driven framework matched the dynamic response.

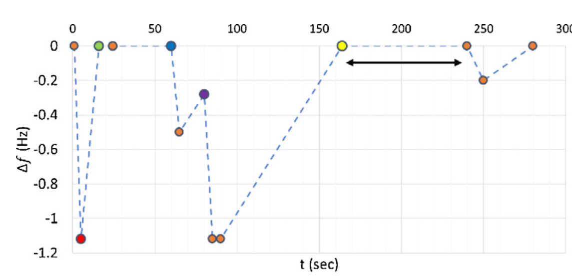

Fig. 8. Event-driven dynamic simulation of 8k bus system [26] with various control actions and disturbances.

Table 1
Control actions in 39-bus testcase.

Control Action	Location/Occurrence
AVR	All generators
Primary Frequency Response	All Generators
Secondary Response	All Generators
Generator Ramp	All Generators (10MW/ minute)
Update network file (with real time measurements)	Every 2 min2 minutes

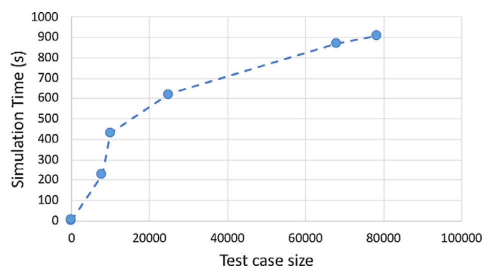

Fig. 9. Simulation time for various sizes test cases running event-driven.

5.2. Dynamic simulation of large system

Now that the approach is verified using dynamic simulation, we can show the efficacy of the framework by simulating long-term dynamics of large systems, which is a challenge for dynamic simulation tools.

We demonstrate the approach on a large 8k bus system [26], with various controls and discrete changes, listed in Table 1 to highlight the effectiveness of the framework. The system is initialized at steady state given by the original RAW file.

To initiate the dynamic sequence, all loads increased by 10% to change the frequency as shown by the figure below.

After the initial disturbance, the framework calculated the quasi steady state due to the primary response at $t = 5$ s (**red marker**), after an inertial nadir, at which time, the ACE was calculated causing generators to contribute more using AGC. The QSS of the AGC response was established at $t = 16.7$ s (**green marker**) at which the frequency deviation was 0 Hz.

The system did not see a disturbance of any kind until an update in the network (load increase, generator decrease, and incorporation of real-time measurements) caused another primary control response that resulted in a QSS at $t = 62$ s (**blue marker**). The network change used in this simulation was a uniform increase in load by 1%. At that point AGC ramped up certain generators to offset the ACE calculated at $t = 65$ s. However, during this ramp up a branch trip at $t = 80$ s (**purple marker**) caused another event that enacted primary and secondary response. The frequency control actions brought the system to a state with zero frequency deviation at $t = 164$ s (**yellow marker**). There was no updated in the network until $t = 240$ s at which the loads increased by 1% resulting in primary response that resulted in a QSS at 250 s. The latent between $t = 164$ s and $t = 240$ s, the **black arrow**, demonstrates the advantage of event-driven dynamic analysis as no QSS was calculated for that time period.

5.3. Scalability

The advantage of the approach is being able to scale to large systems within a reasonable simulation time. The framework can increase the time scale without a large penalty by using previous states as initial conditions. We demonstrate the scalability of the framework by simulating 10 min of dynamic response for various network up to 80k buses, representing a realistic Eastern Interconnection system, with a simulation time shown in Fig. 9, including multiple synthetic US grid testcases of varying sizes [27]. Each of these simulations are run on a single core of a 2.6 GHz Quad-Core Intel Core i7 Macbook Pro-without any level of parallelization.

6. Conclusion

The temporal sparsity of the control actions in the power grid is a bottleneck for dynamic simulation. We introduce an event-based dynamic simulation that captures the quasi-steady state due to each control using a governor power flow. The governor power flow incorporates frequency information and real-time measurements, thereby giving an accurate quasi-steady state of the system. By using events to dictate time steps, the event-driven approach successively solves for QSS, thereby skipping over latent periods. This approach is shown to scale to systems as large as the Eastern Interconnection for long-term time scale while solving the simulation within minutes. The tool is capable of simulating the long-term behavior of system due to a sequence of control actions by the operator while considering the response of a multitude of automatic control mechanism.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) un der award no. FA8750-17-1-0059 for RADICS program, and the National Science Foundation (NSF) under contract no. ECCS-1800812.

References

[1] P. Chirapongsananurak, Santoso, Multi-time-scale simulation tool for renewable energy integration analysis in distribution circuits, Inventions (2017).
[2] J. Shu, W. Xue, W. Zheng, A parallel transient stability simulation for power systems, IEEE Trans. Power Syst. 20 (2005).
[3] L.Wang J.Tong, "Design of a DSA tool for real time system operations,", IEEE Int. Conf. Power Syst. Technol. (2016) Oct..
[4] Dan Yang, Venkataramana Ajjarapu, A decoupled time-domain simulation method via invariant subspace partition for power system analysis, IEEE Trans. Power Syst. 21 (2006).
[5] A. Pandey, et al., Robust power flow and three phase power flow analyses, IEEE Trans. Power Syst. (2018).
[6] H.W. Dommel, W.F. Tinney, W.L. Powell, Further developments in Newton's method for power system applications, IEEE Winter Power Meet. Conf. Pap. No. 70 CP 161-PWR (1970) January.
[7] G.O. Alves, et al., A new governor power flow formulation based on the current injections method, Int. J. Electr. Power Energy Syst. (2019).
[8] A. Agarwal, A. Pandey, M. Jereminov, L. Pileggi, Implicitly modeling frequency control within power flow, IEEE ISGT Eur. (2019).
[9] P. Kundur, Power System Stability and Control, McGraw-Hill, Inc., 1994.
[10] G. Gurrala, et al., Parareal in time for fast power system dynamic simulation, IEEE Trans. Power Syst. 31 (3) (2015).
[11] Renke Huang, et al., Faster than real-time dynamic simulation for large-size power system with detailed dynamic models using high-performance computing platform, Power Energy Soc. General Meet. 2017 IEEE (2017) 1–5.
[12] A. Kurita, et al., Multiple time-scale power system dynamic simulation, IEEE Trans. Power Syst. vol.8, (no.1) (1993).
[13] C. Dufour, J. Mahseredjian, J. Belanger, A combined state-space nodal method for the simulation of power system transients, IEEE 2011 PES General Meet. (2011).
[14] C. Dufour, V. Jalili-Marandi, J. Bélanger, L. Snider, Power system simulation algorithms for parallel computer architectures, IEEE 2012 PES General Meet. (2012).
[15] D. Fabozzi, A.S. Chieh, B. Haut, T. Van Cutsem, Accelerated and localized Newton schemes for faster dynamic simulation of large power systems, IEEE Trans. Power Syst. (2013) Nov..
[16] L. Pileggi, R. Rohrer, C. Visweswariah, Electronic Circuit & System Simulation Methods, McGraw-Hill, Inc, New York, NY, USA, 1995.
[17] T.V. Cutsem, Voltage Stability of Electric Power Systems, Kluwer Academic Publishers, Boston/London/Dordrecht, 1998.
[18] T.V. Cutsem, M.E. Grenier, D. Lefebvre, Combined detailed and quasi steady-state time simulations for large-disturbance analysis, Int. J. Electr. Power Energy Syst. (2006).
[19] X.Z. Wang, H.D. Chiang, Numerical investigations on quasi steady-state model for

voltage stability: limitations and nonlinear analysis, Int. Trans. Electr. Energy Syst. (2014).
[20] A.B. Vyakaranam, et al., A study of the impact of peak demand on increasing vulnerability of cascading failures to extreme contingency events, IEEE 2017 PES General Meet. (2017).
[21] Emanuele Ciapessoni, et al., Benchmarking quasi-steady state cascading outage analysis methodologies, IEEE PMAPS (2018).
[22] A. Pandey, M. Jereminov, X. Li, G. Hug, L. Pileggi, Unified power system analyses and models using equivalent circuit formulation, IEEE 2016 PES ISGT USA (2016).
[23] A. Pandey, A. Agarwal, M. Jereminov, M.R. Wagner, D.M. Bromberg, L. Pileggi, Robust sequential steady-state analysis of cascading outages, IEEE 2019 PES ISGT Eur. Conf. (2019).
[24] B. Hooi, H.A. Song, A. Pandey, M. Jereminov, L. Pileggi, C. Faloutsos, StreamCast: fast and online mining of power grid time sequences, Proc. SIAM Int. Conf. Data Min. (2018) 531–539.
[25] P. Demetriou, et al., Dynamic IEEE test systems for transient analysis, IEEE Syst. J. (2015).
[26] PSSE34 Example Test Case “bench2.raw”. Available with download of PSSE34.
[27] A.B. Birchfield, T. Xu, K.M. Gegner, K.S. Shetye, T.J. Overbye, Grid structural characteristics as validation criteria for synthetic networks, IEEE Trans. Power Syst. (2017).