Simplified Simulation of Digital Controllers in Power System Dynamic Studies

Mehran Jafari, Petros Aristidou
Dept. of Electrical Eng., Computer Eng., & Informatics
Cyprus University of Technology, Limassol, Cyprus
mm.jafari@edu.cut.ac.cy

Gautier Bureau, Marco Chiaramello
Adrien Guironnet, Patrick Panciatici
Réseau de Transport d'Électricité (RTE)
Paris, France

***Abstract*—Time-domain simulation of power system dynamics requires solving large, sparse systems of nonlinear, stiff differential-algebraic equations. Traditional simulation tools have primarily emphasized solution accuracy; however, in the presence of digital controllers, such simulations remain computationally intensive as they must handle many discontinuities. In many practical applications, obtaining an approximate system response from the detailed model is sufficient, reducing the need for full-scale, high-fidelity simulations using small time steps. Simplified simulations provide a means to capture system dynamics quickly. However, the numerous discontinuities arising from the operation of digital controllers can become problematic for a simplified simulation, leading to significant inaccuracies and even divergence of the solver. This paper proposes a simplified simulation method suitable for power systems controlled by digital controllers, aiming to mitigate the accuracy loss of simplified simulation in the presence of multiple controllers.**

Index terms– differential-algebraic equations, digital controllers, simplified simulation, interpolation methods, discontinuities.

I. Introduction

Digital controllers are increasingly deployed in modern power systems for precise control of power electronic devices and grid stability. Typically, simulating such a system requires handling a large set of hybrid differential-algebraic equations (DAEs) of the continuous system [1] alongside the difference equations of digital controllers, which are discrete in nature [2]. However, time-domain simulation of systems containing multiple digital controllers presents significant numerical challenges [3] due to discrete sampling events that introduce numerous discontinuities during the simulation horizon [4].

The discontinuities arising during the simulation can be categorized into state events and time events [4]. A state event occurs when certain conditions are met, such as a state variable reaching a limit; therefore, its time of occurrence cannot be predicted before the simulation starts. By contrast, time events are predictable since their occurrence times are known in advance. The sampling action of digital controllers is a classic example.

Conventional solvers handle discontinuities by precisely aligning time steps with event instances through iterative step-size reductions [5]. While accurate, this approach becomes computationally prohibitive in systems with numerous controllers operating at high sampling frequencies, as demonstrated in [6].

Simplified simulation methods have emerged as promising alternatives to accelerate these simulations while maintaining acceptable accuracy levels [7]. For some applications, such as dynamic security assessment that requires simulating long-term system behavior against many contingencies, a quick simulation may be preferable to a slower but more accurate one. In that case, a simplified simulation can be achieved either through model simplification or by integrating with large time steps [8].

Model simplification is useful for reducing the computational burden, leading to faster simulations. For example, a quasi–steady-state approximation of the power system's fast dynamics reduces both the size and the stiffness of the system [9]. Another example is to replace the digital controllers with their analog equivalents [10]. Removing the digital controllers' difference equations and using DAEs instead eliminates the time events arising from controller sampling. Thus, the controllers' DAEs can be combined with the system's DAEs and solved together using a variable-step method.

Alternatively, the process of solving the model can be simplified [11]. The Simplified Simulation Method (SSM), proposed in [8], accelerates simulations by taking large time steps and deferring discontinuity handling to time step boundaries rather than interrupting integration. This approach avoids frequent step size reductions but introduces temporal shifts in event processing. While effective for sparse state events [7], SSM exhibits critical limitations when applied to digital controllers:

1) **Sequential dependency**: Digital controllers require recursive computation where each output depends on both previous states and system feedback. SSM can only process the first sampling event per time step, discarding subsequent updates.
2) **Feedback latency**: Large time steps prevent access to intermediate system states needed for accurate controller input calculations, particularly at high sampling rates relative to simulation step sizes.

These limitations cause progressive error accumulation and phase delays, rendering conventional SSM unsuitable for systems with multiple controllers or fast sampling requirements.

This paper first addresses the challenges that SSM faces when simulating power systems containing digital controllers. It then proposes a novel interpolation-based simplified simulation approach suitable for capturing power system dynamics comprising digital controllers. The proposed method aims to address the accuracy and performance issues of SSM in systems with multiple digital controllers. Therefore, the contributions of this paper can be highlighted as follows:

- Addressing the accuracy and performance issues of SSM while facing digital controllers.
- Proposing a novel simplified simulation approach suitable for time-domain simulation of power systems in the presence of digital controllers.

The rest of the paper is organized as follows. In Section II, the fundamentals of SSM are briefly reviewed. The interpolation-based simplified simulation approach for handling digital controllers is proposed in Section III. Section IV provides numerical case studies to showcase the performance and accuracy of the proposed method compared to SSM. Finally, conclusions are drawn in Section V.

II. Simplified Simulation Method

To discuss SSM, a continuous system controlled by a digital controller is assumed, as shown in Fig. 1. The state variables vector, denoted by $\boldsymbol{y}(t)$, is sampled and converted to the discrete signal $\boldsymbol{y}(kT)$ using an analog-to-digital converter block to be fed back to the digital controller. The discrete output of the digital controller $\boldsymbol{e}_k$ is converted to a continuous signal using a digital-to-analog converter block to be used as the continuous system input.

Let's define the continuous system using a set of DAE. This leads to an initial-value problem (IVP):

$$\begin{aligned} \mathbf{0} &= \boldsymbol{F}(\dot{\boldsymbol{y}}(t), \boldsymbol{y}(t), e(t)) \\ \boldsymbol{y}(0) &= \boldsymbol{y}_0,\ e(0) = e_0 \end{aligned} \tag{1}$$

Discretizing (1) using a numerical integration method results in non-linear algebraic equations that can be solved over a time step using a Newton method [12]. The time step size is given as $h_n = t_n - t_{n-1}$. The solution for the state variables at each time step, $\boldsymbol{y}(t_n) = \boldsymbol{y}_n$, can be obtained by forming the following residual function:

$$\boldsymbol{g}(\boldsymbol{y}(t_n), e(t_n)) = 0 \tag{2}$$

As mentioned before, the digital controller can be described using difference equations:

$$\boldsymbol{e}_k = \boldsymbol{\zeta}(\boldsymbol{e}_{k-1}, \boldsymbol{y}(kT)) \tag{3}$$

where $\boldsymbol{\zeta}$ is the controller function defining the sequential relation between the controller output $\boldsymbol{e}_k$ to its previous one $\boldsymbol{e}_{k-1}$ at the k-th sampling with sampling period T.

The step reduction method (SRM) addresses each time event by reducing the integration step size h_n to precisely "land" on the discontinuity [13]. In other words, the solver step size is adjusted in a way that it will be equal to the controller sampling time at that instance. Once the time step is adjusted, the controller output, $\boldsymbol{e}_k$, is computed, and the system equations are subsequently integrated using $\boldsymbol{e}(t) = \boldsymbol{e}_k$ as the input. Consequently, (3) is solved prior to (1), with $\boldsymbol{e}(t)$ assumed constant throughout the integration of the time step (zero-order-hold).

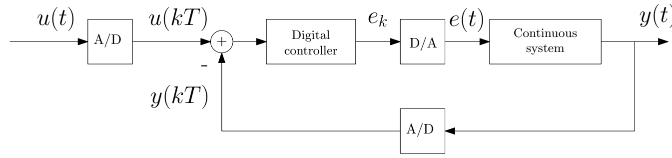

Fig. 1. A continuous system under control by a digital controller scheme

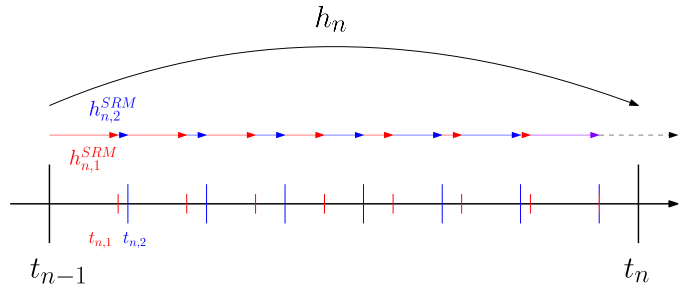

Fig. 2. Schematic of integration using the step reduction method. The black vertical lines denote the simulation time steps, while the blue and red vertical lines denote the digital controllers' sampling actions.

SRM offers the highest accuracy, yielding results closest to actual system behavior. However, it is computationally intensive, as the maximum allowable time step size is constrained by the controllers' sampling rates [14]. For instance, Fig. 2 shows the time stepping using SRM when two digital controllers distinguished by different colors are present. The time step h_n that would have been taken in the absence of events is overridden by a series of new, smaller time steps $h_{n,p}$ selected to land on the controllers' events, where p denotes the index of the event within the time step. Similarly, $t_{n,p}$ denotes intermediate times within the time step t_n that coincide with the sampling times kT.

In contrast to SRM, which reduces the simulation time step to align with the controller's sampling period, SSM proposed in [8] shifts the controller's sampling forward to match the simulation step. More precisely, it advances the digital controller's control action to coincide with the next simulation step, as it is illustrated in Fig. 3 for a system with two digital controllers. Therefore, as there is no time step size reduction, SSM is computationally efficient.

However, SSM can only shift the first sampling action per time step to the end of the time step. The reason is that only for the first event are both the feedback from the system $\boldsymbol{y}(kT)$ and the previous controller action $\boldsymbol{e}_{k-1}$ known. Although the sampling action occurs in the middle of the time step, both the feedback and the previous controller output remain constant after the last sampling in the previous time step due to the zero-order hold.

Fig. 4 shows the time stepping and the controller outputs and inputs, where $\boldsymbol{e}_{n,p}$ denotes the p-th controller output in the

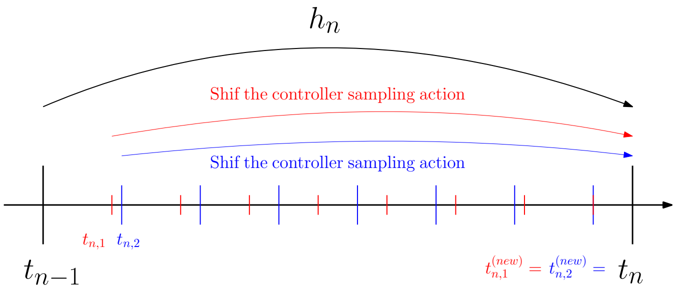

Fig. 3. Schematic of integration using the simplified simulation method. The black vertical lines denote the simulation time steps, while the blue and red vertical lines denote the digital controllers' sampling actions.

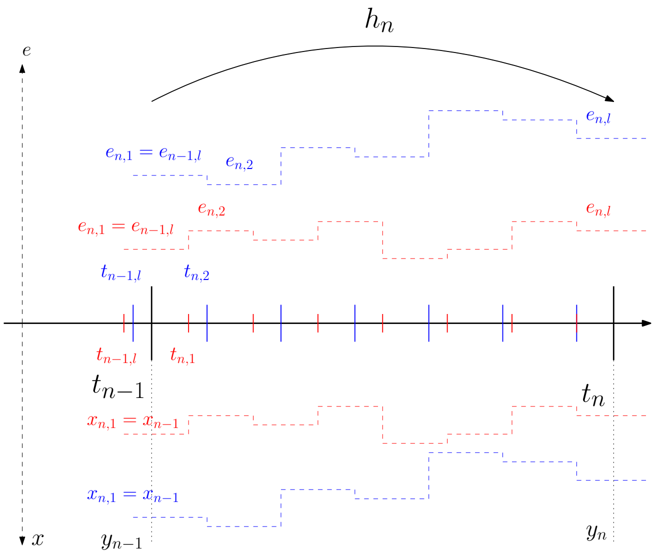

Fig. 4. Schematic of integration using the simplified simulation method and the intermediate controller inputs $x_{n,p}$ and outputs $e_{n,p}$. The red and blue inks indicate two separate controllers.

time step t_n, $\boldsymbol{x}_{n,p}$ is the p-th feedback (controller input) in the same time step. Also, $\boldsymbol{e}_{n,l}$ denotes the last controller output in the time step t_n while $\boldsymbol{e}_{n-1,l}$ is the last controller output of the previous time step. It should be noted that $\boldsymbol{x}$ is a subset of state variables $\boldsymbol{y}$ monitored by the controller. Considering (3), the first feedback of the controller $\boldsymbol{x}_{n,1}$ is computable as it is equal to $\boldsymbol{x}_{n-1}$, which is known from the previous time step. The same logic applies to the first controller output $\boldsymbol{e}_{n,1}$, which is considered constant and equal to the last controller output in the previous time step, $\boldsymbol{e}_{n-1,l}$. However, subsequent controller outputs cannot be obtained because the feedback at the next sampling instants is not known. For the sake of comparison, the pseudocode of SRM and SSM is listed in Fig. 5.

Beyond delaying and handling only the first controller action, ignoring the remaining actions further degrades accuracy. As the number of digital controllers and the number of time events falling within each time step increase, more events are discarded, and the inaccuracy grows. Therefore, there is a need for a simplified simulation framework capable of simulating systems containing digital controllers with acceptable accuracy.

III. The Proposed Simplified Interpolation-Based Method

To address the inaccuracy issue of SSM arising from ignoring the time events, we propose an enhanced Simplified Interpolation-Based Method (SIBM) incorporating two key innovations:

1) **Recursive controller evaluation**: Considering (3), it can be seen that the controller output depends on the previous one. This means that the function $\boldsymbol{\zeta}$, which calculates the controller output, can be called recursively as many times as the controller samples in the time step h_n, instead of being called only once as SSM does. Therefore, in the proposed method, controller outputs are computed at each virtual sampling instant within the time step through iterative application of difference equations, maintaining temporal dependencies between successive states.
2) **Adaptive state interpolation**: Instead of using the outdated feedback $\boldsymbol{y}(kT)$ from the previous time step, an interpolation scheme estimates intermediate system states from previous time step solutions, providing estimates of the feedback inputs for controller computations without requiring step size reduction.

This dual approach enables complete processing of all sampling events within each time step while preserving the computational advantages of simplified simulation and not reducing the time step. The interpolation mechanism ensures controller inputs remain synchronized with virtual sampling instants, fundamentally addressing SSM's feedback latency issue.

Fig. 6 shows the scheme of SIBM, where the red arrows show the shifting of events to the end of the time step, the blue dashed arrows indicate interpolating the feedback for the controller $\boldsymbol{x}_{n,p}$, and the green arrow is the consecutive calculation of controller outputs stacking up until the last one $\boldsymbol{e}_{n,l}$.

To better discuss SIBM, an interpolation function is used to estimate the feedback coming from the continuous system:

$$\boldsymbol{x}_{n,p} = \boldsymbol{w}_n(t_{n,p}), \quad \forall p \in [1, l] \tag{4}$$

where w_n denotes the interpolation function at the time step t_n. To bound the error of the integration method, it should have a higher order of local error than the interpolation polynomial [15]. Then, considering (3), it can be rewritten with the interpolated values instead of the system's feedback:

$$\boldsymbol{e}_{n,p} = \boldsymbol{\zeta}(\boldsymbol{e}_{n,p-1}, \boldsymbol{x}_{n,p}) \tag{5}$$

Now, the controller outputs can be calculated one after another until the last one $\boldsymbol{e}_{n,l}$ is computed. It should be noted that the method is agnostic to the controller's transfer function order, as SIBM only relies on the difference equations of the controller. Finally, $\boldsymbol{e}_{n,l}$ is fed back to the continuous system as

Algorithm 1 SRM

Require: $t_0, t_{\text{end}}, y_0, e_0$
$n \leftarrow 0$
while $t_n < t_{\text{end}}$ **do**
  Identify all sampling instants $\{t_{n,1}, \ldots, t_{n,l}\}$ in $(t_{n-1}, t_n]$
  for $p = 1$ **to** l **do**
    $h_{n,p} \leftarrow t_{n,p} - t_{n,p-1}$
    Solve (1) over $h_{n,p}$
    $e_{n,p} \leftarrow \zeta(e_{n,p-1},\, y_{n,p})$
  end for
  $e_n \leftarrow e_{n,l}$
  $n \leftarrow n + 1$
end while

Algorithm 2 SSM

Require: $t_0, t_{\text{end}}, y_0, e_0, h_{\max}$
$n \leftarrow 0$
while $t_n < t_{\text{end}}$ **do**
  $h_n \leftarrow h_{\max}$
  $x_{n,1} \leftarrow x_{n-1}$ ▷ stale FB
  $e_{n,1} \leftarrow \zeta(e_{n-1,l},\, x_{n,1})$
  Discard $e_{n,2}, \ldots, e_{n,l}$
  Solve (1) over h_n with input $e_{n,1}$
  $e_n \leftarrow e_{n,1}$
  $n \leftarrow n + 1$
end while

Algorithm 3 SIBM (proposed)

Require: $t_0, t_{\text{end}}, y_0, e_0, h_{\max}, w_n$
$n \leftarrow 0$
while $t_n < t_{\text{end}}$ **do**
  $h_n \leftarrow h_{\max}$
  for $p = 1$ **to** l **do**
    $x_{n,p} \leftarrow w_n(t_{n,p})$ ▷ (4)
    $e_{n,p} \leftarrow \zeta(e_{n,p-1},\, x_{n,p})$ ▷ (5)
  end for
  Solve (1) over h_n with input $e_{n,l}$
  $e_n \leftarrow e_{n,l}$
  $n \leftarrow n + 1$
end while

Fig. 5. Pseudocode comparison of SRM, SSM, and the proposed SIBM for one simulation horizon. The key difference is how each method handles the l controller sampling events within a time step h_n: SRM reduces the step to each event; SSM processes only the first using stale feedback; SIBM processes all l events using interpolated feedback $w_n(t_{n,p})$ without step reduction.

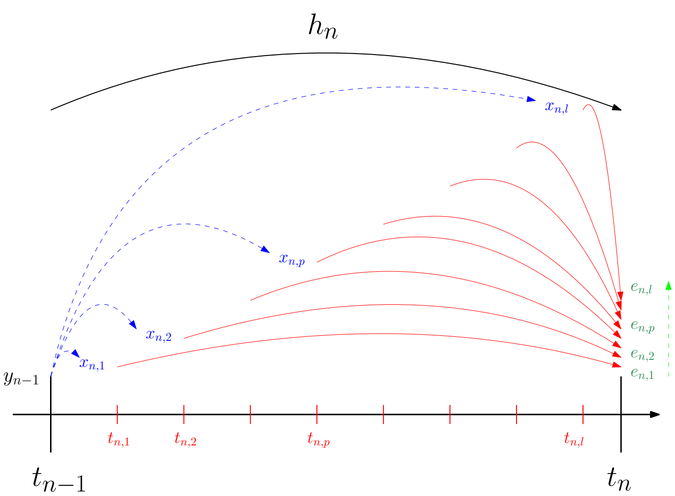

Fig. 6. Time stepping using SIBM in a simulation with one digital controller. The blue ink indicates the interpolated variables. The red arrows indicate the shifting of samples. The green arrow shows the sequence of controller outputs being calculated.

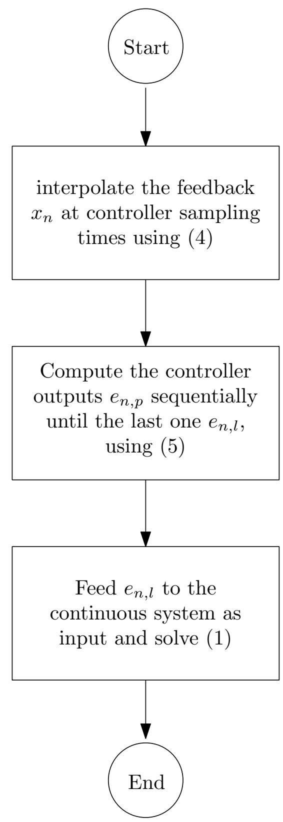

Fig. 7. Block diagram of SIBM.

the input (see (1)) and the Newton method is used for obtaining $\boldsymbol{y}_n$. To further clarify, the pseudocode of all the methods and the block diagram of the SIBM stages of interpolation and sequential calculation of controller outputs to solve (1) for one time step is shown in Fig. 5., and Fig. 7.

IV. Numerical Experiments and Validation

In this section, a case study is considered to first showcase the challenge of SSM and its accuracy loss when facing many time events in each time step, then the accuracy and performance of SIBM, performing the same simulation, are explored.

The two-area Kundur system [16] is considered as the test system for numerical simulations, as shown in Fig. 8. It has four synchronous generators, each under control by a digital exciter [17] and a digital governor [18], illustrated in Fig. 9 and Fig. 10, respectively.

A variable-step predictor–corrector integration scheme, based on the second-order Adams–Bashforth and Adams–Moulton pair, is employed for all the simulations [19], and for all three methods: SRM, SSM, and SIBM. For the

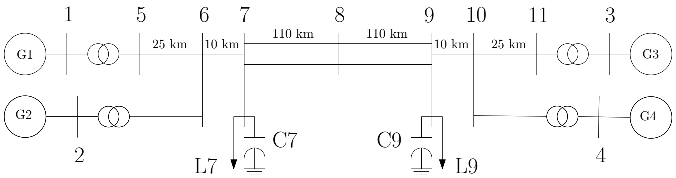

Fig. 8. Schematic of the two-area Kundur test system

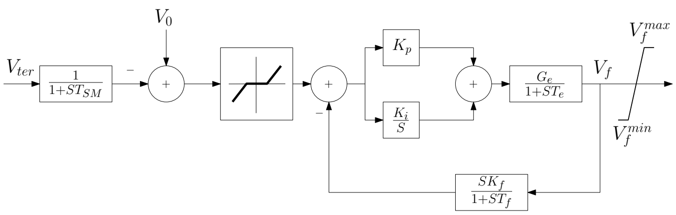

Fig. 9. Schematic of the digital exciter of the two-area Kundur test system

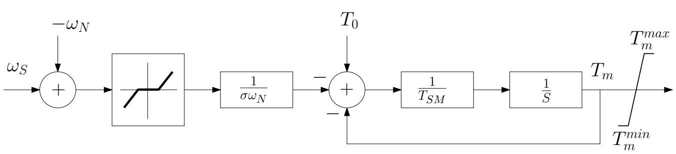

Fig. 10. Schematic of the digital governor of the two-area Kundur test system

SIBM approach, a second-order Taylor expansion polynomial is applied for interpolation. The same set of system DAEs and controller difference equations is solved across all three methods. Uniform quantization with 18-bit resolution is assumed for both the analog-to-digital (A/D) and digital-to-analog (D/A) components [20]. The time step adjustment factors are set to 1.25 for step size increases and 0.5 for decreases, with the minimum and maximum allowable time steps defined as 1 ms and 1 s, respectively. All models and solution algorithms are implemented in MATLAB [21].

A. Slow controllers

In the first scenario, controllers with relatively slower sampling rates are employed. Consequently, fewer time events occur within each simulation step, and a smaller number of these events are neglected by the SSM. In this case, the sampling periods T of the digital controllers are set to 210 ms, 220 ms, 230 ms, and 240 ms for the governors, and 41 ms, 42 ms, 43 ms, and 44 ms for the exciters.

A 200 ms short-circuit fault is applied at bus 3 and simulated using all three methods: SRM, SSM, and SIBM. The resulting simulation outputs are presented in Figs. 11, 12, 13, and 14, corresponding to the voltage at bus 1, the speed deviation, governor output, and exciter output of generator 3, respectively. The results indicate that, while the SSM exhibits minor accuracy degradation, the SIBM closely follows the correct trajectory, showing accuracy nearly identical to SRM. It can be observed that the Exciter reaches its maximum limit of 2 pu after the fault, and SIBM follows the contingency with accuracy similar to that of SRM, highlighting its effectiveness under contingencies and non-linearity.

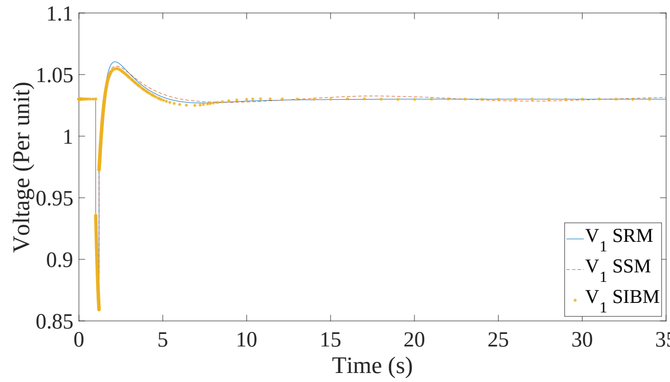

Fig. 11. Voltage of bus 1 of the two-area Kundur test system with fast controllers.

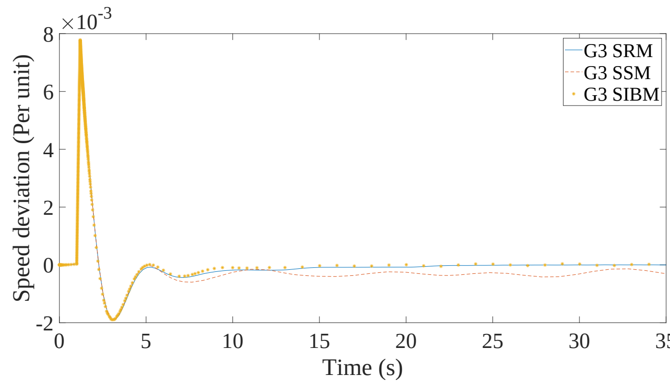

Fig. 12. Speed deviation of generator at bus 3 of the two-area Kundur test system with slow controllers.

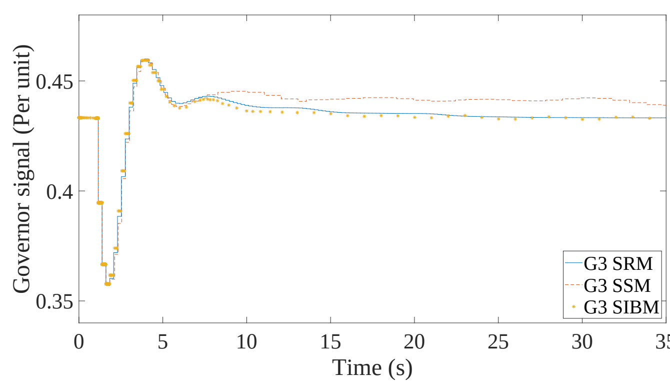

Fig. 13. Governor output of generator 3 of the two-area Kundur test system with slow controllers.

The performance results of the simulations using all the methods are listed in Table I, in terms of the number of Newton iterations and average run times. As expected, SSM is the fastest method. SIBM is the second fastest, with performance similar to SSM, and SRM is the slowest as it takes the smallest time steps. The step size results for the methods are illustrated in Fig. 15. As evident, SSM reduces the step size a couple of

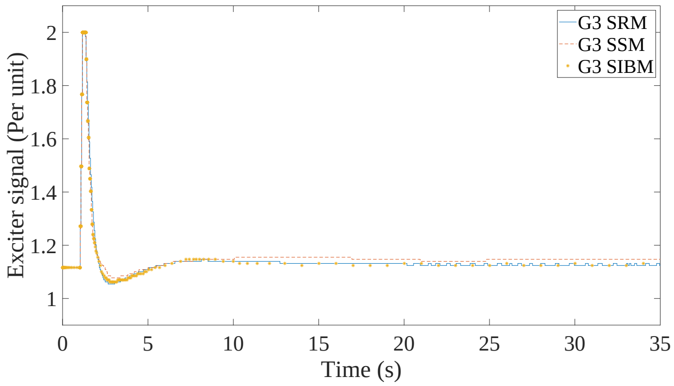

Fig. 14. Exciter output of generator 3 of the two-area Kundur test system with slow controllers.

TABLE I
PERFORMANCE COMPARISON BETWEEN SRM, SSM, AND SIBM FOR THE KUNDUR SYSTEM WITH SLOW CONTROLLERS IN TERMS OF THE NUMBER OF NEWTON ITERATIONS AND RUNTIME

Method	SRM	SSM	SIBM
Newton iterations	16813	1198	1214
Average runtime (s)	66.14	4.72	5.13

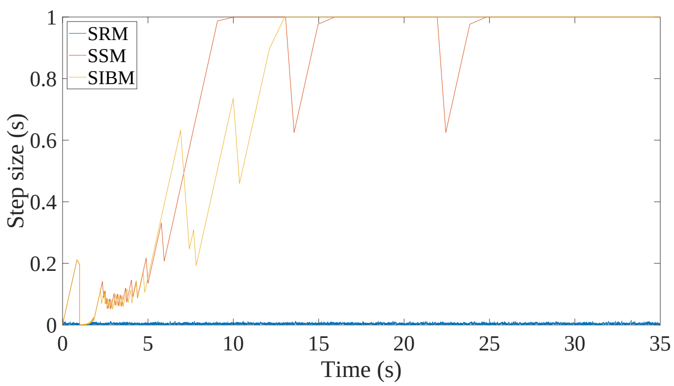

Fig. 15. Step size results for all the methods simulating the two-area Kundur system with slow controllers. The minimum and maximum time step size limits are equal to 1 ms and 1 s, respectively, for all the methods.

times due to accumulated error estimates. However, it reaches the maximum time step first. The figure also shows how much the step size is limited for SRM compared to SSM and SIBM.

B. Fast controllers

The same simulation is repeated using controllers with faster sampling rates. In this scenario, the sampling periods T of the digital controllers are set to 91 ms, 92 ms, 93 ms, and 94 ms for the governors, and 11 ms, 12 ms, 13 ms, and 14 ms for the exciters.

The simulation results are shown in Figs. 16, 17, 18, and 19, for the voltage of bus 1, the speed deviation of the third generator, the governor output of the third generator, and the exciter output of the third generator, respectively.

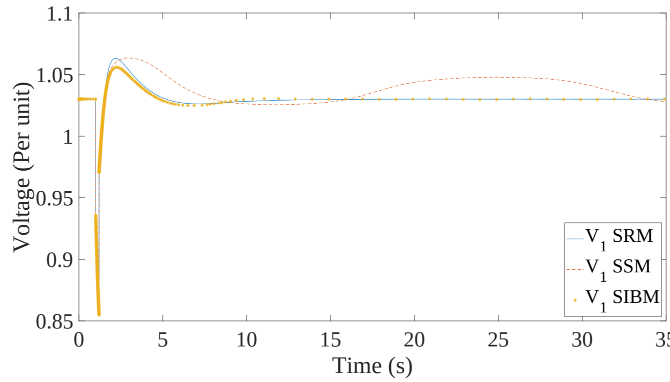

Fig. 16. Voltage of bus 1 of the two-area Kundur test system with fast controllers.

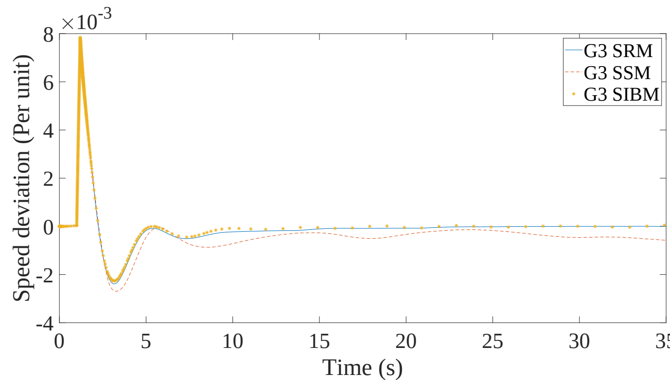

Fig. 17. Speed deviation of generator at bus 3 of the two-area Kundur test system with fast controllers.

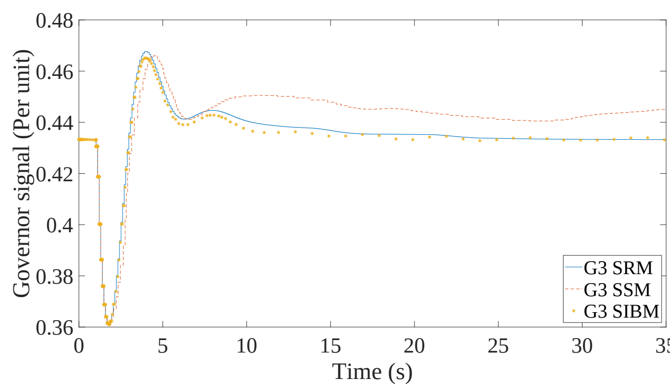

Fig. 18. Governor output of generator 3 of the two-area Kundur test system with fast controllers.

In this scenario, the SSM exhibits difficulty in reaching the steady state and accurately capturing system fluctuations. This issue can be mitigated by reducing the maximum allowable time step, thereby limiting the number of events occurring within each step. However, such an adjustment compromises the computational efficiency of the SSM, negating its primary advantage. In contrast, the SIBM once again demonstrates

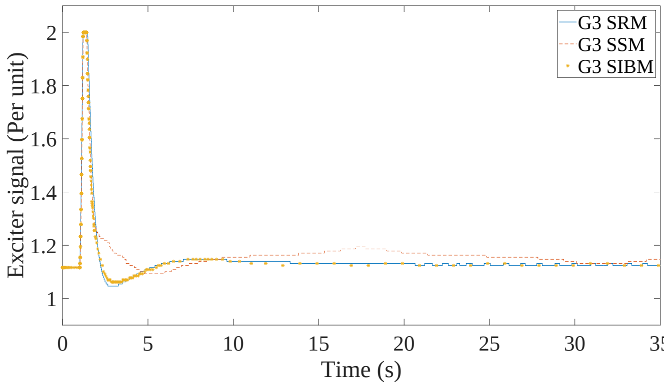

Fig. 19. Exciter output of generator 3 of the two-area Kundur test system with fast controllers.

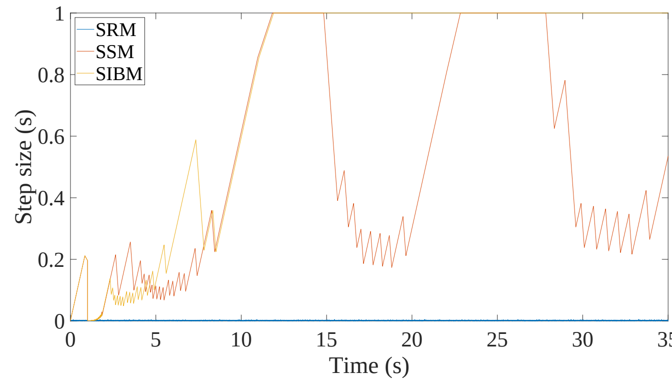

Fig. 20. Step size results for all the methods simulating the two-area Kundur system with fast controllers. The minimum and maximum time step size limits are equal to 1 ms and 1 s, respectively, for all the methods.

TABLE II
PERFORMANCE COMPARISON BETWEEN SRM, SSM, AND SIBM FOR THE KUNDUR SYSTEM WITH FAST CONTROLLERS IN TERMS OF THE NUMBER OF NEWTON ITERATIONS AND RUNTIME

Method	SRM	SSM	SIBM
Newton iterations	50736	1345	1235
Average runtime (s)	195.65	5.32	5.23

TABLE III
PERFORMANCE DROP RATE BETWEEN TWO SCENARIOS FOR SRM, SSM, AND SIBM FOR THE KUNDUR SYSTEM IN TERMS OF THE NUMBER OF NEWTON ITERATIONS AND RUNTIME

Method	SRM	SSM	SIBM
Newton iterations drop ratio	3.01	1.12	1.01
Runtime drop ratio	2.95	1.12	1.02

superior accuracy, closely following the SRM trajectories. In addition, once again, it can be noticed that the exciter maximum limit is hit after the fault, during which SIBM follows the SRM trajectory with good precision.

The performance results for the simulations with fast controllers are summarized in Table II. As can be seen, the SIBM is the fastest method, while the SSM requires a greater number of Newton iterations. Two factors contribute to the increased number of iterations in the SSM. First, the method demands more Newton iterations to achieve convergence. Additionally, the SSM frequently reduces its time step to prevent excessive error estimates. This behavior is illustrated in Fig. 20, which presents the time step profiles for all methods. As observed, the SSM periodically decreases its time step to maintain the error estimate below the prescribed threshold while attempting to reach the steady state.

Another interesting point regarding the performance of the methods is that SRM is three times slower compared to the first scenario with slower controllers, as shown in Table III, which lists the performance drops in terms of run times and Newton iterations. The reason is that the step sizes are heavily constrained by the small intervals between the controllers' samplings. The other methods do not experience that degree of performance drop, losing only a few seconds, as they still jump over multiple discontinuities. Meanwhile, SIBM has the lowest performance-drop ratio.

It should be noted that, for the sake of comparison, the maximum step size limit for all methods has been kept equal to 1 second. Nevertheless, the accuracy issues of SSM can be improved with time step size reductions to prevent an excessive number of events within time steps. However, this removes SSM's advantage of high performance. With severe step size reductions, its accuracy and performance come close to SRM, while SIBM alleviates this need. To better illustrate this, the same simulation with the fast controllers is repeated using SSM with different maximum time step size limits, equal to 1, 0.1, and 0.01 s, denoted as SSM1, SSM2, and SSM3, respectively. The voltage of bus 1 and the governor output of the generator 3 are depicted in Fig. 21 and Fig. 22, respectively. It can be seen that the accuracy issue is resolved; however, the performance drops as can be read from Table IV, which summarizes the number of Newton iterations and the runtime of these simulations.

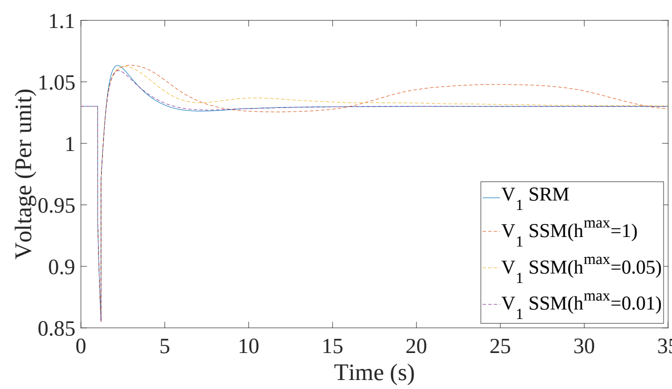

Fig. 21. simulation of voltage of bus 1 of the two-area Kundur test system with fast controllers using SRM and SSM with different maximum step size limits.

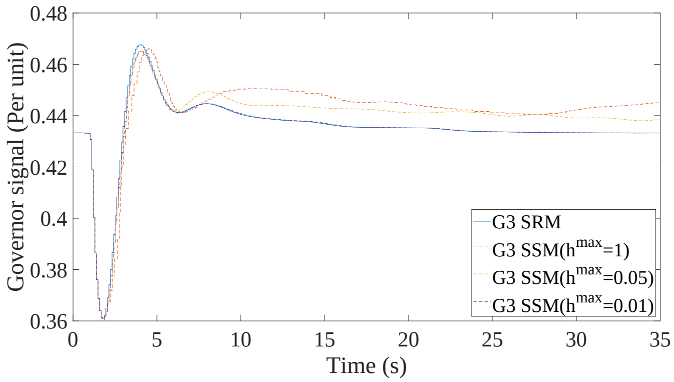

Fig. 22. simulation of governor output of generator 3 of the two-area Kundur test system with fast controllers using SRM and SSM with different maximum step size limits.

TABLE IV
PERFORMANCE COMPARISON BETWEEN SRM AND SSM WITH DIFFERENT MAXIMUM STEP SIZE LIMITS FOR THE SIMULATION OF THE TWO-AREA KUNDUR SYSTEM WITH FAST CONTROLLERS IN TERMS OF THE NUMBER OF NEWTON ITERATIONS AND RUNTIME

Method	SRM	SSM1	SSM2	SSM3
Newton iterations	50736	1345	1890	4683
Runtime (s)	195.65	5.32	7.14	17.69

V. CONCLUSION

In this paper, it was shown that the traditional simplified simulation method loses accuracy when multiple fast digital controllers are present, as it cannot shift more than one event per controller to the end of each time step. Therefore, a novel, simplified interpolation-based simulation method was proposed for simulating power systems with digital controllers, capable of shifting all controller actions to the end of each time step and computing them sequentially. It employs interpolation to obtain accurate estimates of system feedback, thereby improving overall simulation accuracy compared to the traditional simplified approach. It also exhibits better performance than the regular simplified simulation approach, as it accumulates less error over the time steps and thus maintains larger time steps over the simulation.

For future work, large-scale simulations for systems with more controllers can be conducted to assess how the computational cost gap between the methods scales with system size. It is expected to see that the SIBM's performance overhead is proportional to the number of sampling events per step and is independent of the power system size, and the Newton iterations remain the main part of the computation cost.

REFERENCES

[1] F. Milano, *Power system modelling and scripting*. Springer Science & Business Media, 2010.

[2] M. Otter, H. Elmqvist, and S. E. Mattsson, "Hybrid modeling in Modelica based on the synchronous data flow principle," in *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (CACSD)*. IEEE, 1999, pp. 151–157.

[3] M. Jafari, G. Bureau, M. Chiaramello, A. Guironnet, P. Panciatici, and P. Aristidou, "Handling of computationally demanding digital controllers in power system dynamic simulations," in *2024 3rd International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED)*. IEEE, 2024, pp. 1–5.

[4] D. Ellison, "Efficient automatic integration of ordinary differential equations with discontinuities," *Mathematics and Computers in Simulation*, vol. 23, no. 1, pp. 12–20, 1981.

[5] M. Jafari, G. Bureau, M. Chiaramello, A. Guironnet, P. Panciatici, and P. Aristidou, "Modeling of digital controllers in electric power system dynamic simulations," in *2023 IEEE Belgrade PowerTech*. IEEE, 2023, pp. 01–06.

[6] ——, "Fast and accurate simulation of smart digital controllers in power system dynamic studies," *Sustainable Energy, Grids and Networks*, vol. 42, p. 101665, 2025.

[7] D. Fabozzi and T. Van Cutsem, "Simplified time-domain simulation of detailed long-term dynamic models," in *2009 IEEE Power & Energy Society General Meeting*. IEEE, 2009, pp. 1–8.

[8] D. Fabozzi, A. S. Chieh, P. Panciatici, and T. Van Cutsem, "On simplified handling of state events in time-domain simulation," in *Proceedings of the 17th Power Systems Computation Conference (PSCC)*, 2011. [Online]. Available: https://orbi.uliege.be/handle/2268/91650

[9] T. Van Cutsem and C. Vournas, *Voltage stability of electric power systems*. Springer Science & Business Media, 2007.

[10] M. Jafari, G. Bureau, M. Chiaramello, A. Guironnet, P. Panciatici, and P. Aristidou, "Methods for incorporating digital controllers in power system dynamic simulations," *Electric Power Systems Research*, vol. 235, p. 110827, 2024.

[11] ——, "Decoupled interpolation-based method for numerical simulation of digital controllers," in *2024 IEEE International Systems Conference (SysCon)*. IEEE, 2024, pp. 1–6.

[12] M. B. Carver, "Efficient integration over discontinuities in ordinary differential equation simulations," *Mathematics and Computers in Simulation*, vol. 20, no. 3, pp. 190–196, 1978.

[13] F. Zhang, M. Yeddanapudi, and P. J. Mosterman, "Zero-crossing location and detection algorithms for hybrid system simulation," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 7967–7972, 2008.

[14] M. Jafari, G. Bureau, M. Chiaramello, A. Guironnet, P. Panciatici, and P. Aristidou, "A Modelica IBM implementation for fast simulation of digital controllers in power systems," in *2024 Open Source Modelling and Simulation of Energy Systems (OSMSES)*. IEEE, 2024, pp. 1–6.

[15] S. Glumac and Z. Kovačić, "Defect Analysis of a Non-Iterative Co-Simulation," *Mathematics*, vol. 11, no. 6, p. 1342, 2023.

[16] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-Hill New York, 1994, vol. 7.

[17] IEEE, "IEEE recommended practice for excitation system models for power system stability studies," IEEE, Tech. Rep. IEEE Std 421.5-2016, 2016.

[18] P. Pourbeik, "Dynamic models for turbine-governors in power system studies," IEEE Power & Energy Society, Technical Report PES-TR1, 2013.

[19] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, 1998.

[20] K. Sayood, *Introduction to data compression*, 5th ed. Morgan Kaufmann, 2017.

[21] The MathWorks Inc., "MATLAB version 9.10.0 (R2021a)," Natick, Massachusetts, United States, 2021. [Online]. Available: https://www.mathworks.com