Pipelined Numerical Integration on Reduced Accuracy Architectures for Power System Transient Simulations

Georgios Lilis[(a)], Theodoros Kyriakidis[(a)], Guillaume Lanz, Rachid Cherkaoui, Maher Kayal

Electronics Laboratory and Power Systems Group
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
georgios.lilis@epfl.ch, theodoros.kyriakidis@epfl.ch

Abstract— **This work concerns a dedicated mixed-signal power system dynamic simulator. The equations that describe the behavior of a power system can be decoupled in a large linear system that is handled by the analog part of the hardware, and a set of differential equations. The latter are solved using numerical integration algorithms implemented in dedicated pipelines on a field programmable gate array (FPGA). This datapath is operating in a precision-starved environment since is it synthesized using fixed-point arithmetic, as well as it relies on low-precision solutions that come from the analog linear solver. In this paper, the pipelined integration scheme is presented and an assessment of different numerical integration algorithms is performed based on their effect on the final results. It is concluded that in low-precision environments higher order integration algorithms should be preferred when the timestep is large, since simpler algorithms result in unacceptable artifacts (extraneous instabilities).**

Keywords: Power system dynamics; Differential equations; Numerical simulation; Field programmable gate arrays; Fixed-point arithmetic; Numerical stability

I. Introduction

The power system is an extremely complex system with a vast array of different operations taking place continuously by different stakeholders. A common line that is found in great many of these operations is the need for simulation of the phenomena that occur in the system.

One particular type of simulation that is of great interest to power system engineers is *transient simulation* as it is very relevant to transient stability analysis (TSA) studies performed on the system. The phenomena of interest are electromechanical oscillations with frequencies in the order of 0-2 Hz and the time window of interest is normally in the order of tens of seconds. When transient behavior models are used for all the components, the resulting mathematical formulation of the problem includes a combination of a big linear system and a set of differential algebraic equations (DAE). This set of DAEs can be solved, broadly categorized, using two general solution approaches, simultaneous and partitioned [1-3]. This work follows the partitioned approach.

The major bottleneck in the partitioned approach is the solution of a sparse linear system that arises from the interconnection equations of the grid (admittance matrix). In order to tackle the problem in a computationally efficient way, the authors have been recently involved in the development of a transient simulator that is not based on general purpose computing units but on dedicated hardware instead [4-5]. This platform combines an analog and a digital computer part, i.e. it uses a *mixed-signal* electronic design.

The *analog part* acts as an extremely parallelized linear system solver with very promising timing results [4-6]. However, imprecisions in the solution of the linear system are introduced by the imperfect reconfigurable elements (potentiometers, switches) and converters (ADCs, DACs) of the analog computer. This essentially means that only an approximation of the linear system is solved instead of the original one. Initial attempts to minimize this effect have been carried out in a modular calibration scheme [6]. Scrutiny of the analog part is out of the scope of this work. It is important however to notice, that this inaccurate solving of the linear system creates an imprecise environment for the digital part.

The *digital part* handles the solution of differential equations using numerical integration on an FPGA. Computations are done in a pipelined fashion. The pipelined architecture is an attractive integration scheme for dedicated hardware platforms as it is a tradeoff between a fully parallel and a fully sequential approach. It combines merits of both in yielding substantial resource savings against the former and significant computational benefits against the latter [4]. Digital imprecision sources also exist, and include the utilization of fixed-point arithmetic on the FPGA, as well as the inherent inaccuracy of numerical integration algorithms. This way the already inaccurate datapath is further exacerbated and concerns are raised for the usability of the final results.

It is the scope of this work to present the pipelined integration architecture and to investigate the effect of different integration algorithms on the quality and the usability of the results. Since simulations are used in a vast array of contexts, e.g. in real time operation and emergency actions, the quality of the simulator results is of utmost importance for the robustness of the power system itself.

The structure of this paper is as follows. Section II presents the mathematical formulation of the transient simulation problem alongside the necessary mathematical tools to solve it. An overview of the dedicated hardware is

---

(a) G. Lilis and T. Kyriakidis contributed equally to this work.
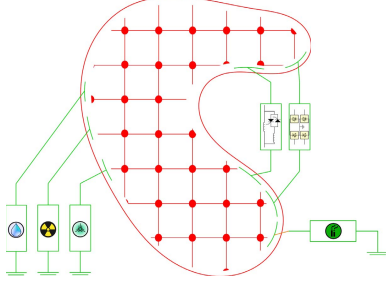
IEEE
computer
society

Fig. 1.   Abstraction of a power system

given in section III, with emphasis on the pipelined integration implementation. A comparative assessment of different integration algorithms is carried out in section IV and conclusions are drawn in section V.

II.   The Formulation of the Problem of Transient Simulation

The two major constituent components of the power system are the interconnecting components (branches, DC links, transformers) and the elements that are connected to the buses (generators, loads, shunt compensation devices). The former are called *grid* components and the latter *injection* components and are shown in fig. 1 in red and green respectively. Given that the study is confined to electromechanical phenomena in the order of Hz, all fast continuous dynamics of the system, such as electromagnetic phenomena with time constants in the order of milliseconds, can be neglected. Readers interested on detailed analysis on the derivation of equations for transient simulation are refered to [7]. Hereunder the final results are presented for clarity sake.

The grid part of the system is modeled by a big sparse linear system

$$_{(n\times 1)}\underline{I} = {}_{(n\times n)}\underline{Y}_{(n\times 1)}\underline{V} \tag{1}$$

Where $\underline{I}$ are the complex bus complex current injections (known), $\underline{Y}$ is the grid admittance matrix (known), and $\underline{V}$ are complex bus voltages (unknown).

The behavior of each injection is governed by a system of differential algebraic equations.

$$S_i = \begin{cases} \dot{x}_i = f_i(x_i, y_i, \lambda_i) \\ 0 = g_i(x_i, y_i, \lambda_i) \end{cases} \tag{2}$$

Where x_i a set of element-specific state variables and y_i a set of element-specific instantaneous variables. Generally, $\partial g_i / \partial y_i \neq 0$, so the DAEs are of index one and f_i can be expressed only as a function of x_i [8]. The interface between the injections (2) and the grid (1) is the complex bus voltage V_i and a complex bus current injection I_i both of which usually appear explicitly on y_i.

The complete system of equations (1) and (2) for all i, constitutes the ensemble of the equations for the problem to solve.

A. Partitioned Solution

The partitioned approach to solve the system of equations is a standard option for many industrial-grade programs [2] and is presented schematically in fig. 2. At
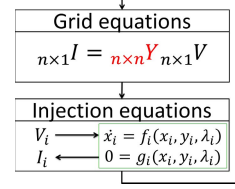

Fig. 2.   Partitioned solution of power system equations

each time step of the integration, the nodal flow algebraic equation of the grid (1) is solved and then the dynamic behavior of the injections is determined (2). As the latter includes a differential part, its solution involves numerical integration algorithms.

B. Linear System Solving

Equation (1) is a large complex sparse linear system that has to be solved at every iteration of the partitioned scheme. When the computation is performed on general purpose (GP) conventional hardware CPUs, well-established direct and iterative techniques exist to solve such large sparse linear systems [9]. However running the algorithm on GP CPUs is inherently problematic given that the datapath of the latter has not been designed to handle efficiently the vectorial operations necessary in linear system solving.

Specialized datapath platforms dedicated to linear system solving have been investigated by researchers for long [10-11].Power system specific platforms have also been created and results have been encouraging [12-15]. However all of the aforementioned are in the digital domain. The authors of this work have explored the use of analog (unconventional) computing as a means of solving the linear system [4-5]. The sparse matrix Y is handled by an analog lattice computer that is briefly described in section III.

C. Numerical Integration

Numerical integration in the partitioned scheme is used to determine the behavior of the injections according to (2). Let an ordinary differential equation be $\dot{x} = f(x)$, and suppose an initial value of $x(0) = x_0$. Then the solution of the above along time can be expressed as $x(t) = \varphi_t(x_0, t_0)$. The point of numerical integration algorithms is to approximate this perfect analytical trajectory by a sequence of points x_k that correspond to respective time instants t_k. Generally, numerical integration algorithms can be defined in two broad categories, *implicit* and *explicit* methods. In order to calculate the value of the variable in the next step, implicit methods solve a system that involves this value, while explicit methods provide a direct, and hence explicit, solution for the next step.

Obviously, implicit methods are computationally much more demanding. This comes though with the benefit of having much better numerical properties, in terms of numerical stability, i.e. explicit algorithms perform worse when used for solving stiff equations. Stiffness is the property exhibited by some differential equations, for which, the solution to be accurate enough, the time step h has to be very small [16]. Stiffness is often due to the fact that in the real system there are phenomena with very different time scales. For the problem in question for this work, i.e.
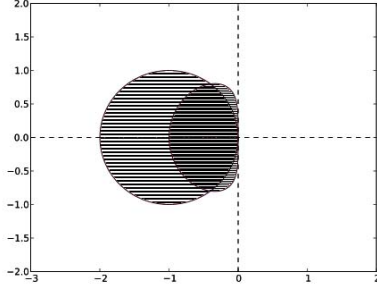
Fig. 3.   Stability region of Forward Euler method (in sparse stripes),
and of 2-step Adams-Bashforth method (in dense stripes)

transient simulation, this is not the case as the dynamics are confined to the range of the electromechanical time constants of the generators of the network [8]. So, in this work, only explicit algorithms are relevant, benefiting from their ease of implementation and their computational simplicity.

Fundamental properties of integration algorithms include order and stability.

The *order* of a method is a measure of how well the difference equation defining the method approximates the differential equation. This can be quantified by the local truncation error $\delta = x_{k+1} - x(t_k + h)$, where x_{k+1} is given by the formula of the method. A method has order p if the local error is of order $O(h^{p+1})$ as the time step h goes to zero. Intuitively, the higher the order of the method, the smaller the propagation of the error across time steps.

Stability directly relates to the concept of stiffness, as stated earlier in this subsection. A common method to assess the stability of a method, is to apply it to the standard test equation $\dot{y} = ky$, $y_0 = y(0) = 1, k \in \mathbb{C}$ .The analytical solution of which is $y(t) = e^{kt}$

This is a test of how well the method can follow dynamics with time constants dictated by k. When applied on this problem, the solution for the next step has the general form (by induction)

$$y_k = \mathcal{F}(k \cdot h)^k \cdot y_0$$

So, for $\lim_{k \to \infty} y_k = 0$ it has to be $|\mathcal{F}(z)| < 1$, where $z \triangleq k \cdot h \in \mathbb{C}$. The latter defines the stability region for the method, in the complex plane for z. A smaller stability region normally means that smaller time steps have to be adopted to prevent instability from occurring.

There is a multitude of explicit methods available. This comparative study will focus on the Forward Euler (FE) (1st order) and the 2-step Adams-Bashforth method (AB2) (2nd order), due to the simplicity of the first and popularity of the second for transient simulation applications [8].

FE: $x_{k+1} = x_k + h \cdot f(x_k)$ (3)

AB2: $x_{k+1} = x_k + h \cdot \left[\frac{3}{2} \cdot f(x_k) - \frac{1}{2} \cdot f(x_{k-1}) \right]$ (4)

The stability region of the two methods is shown in fig. 3. AB2 has a smaller region of stability, however, when stable, it is expected to perform better than FE because of its higher order. The latter can be of great importance in our application that is described in section IV, since the
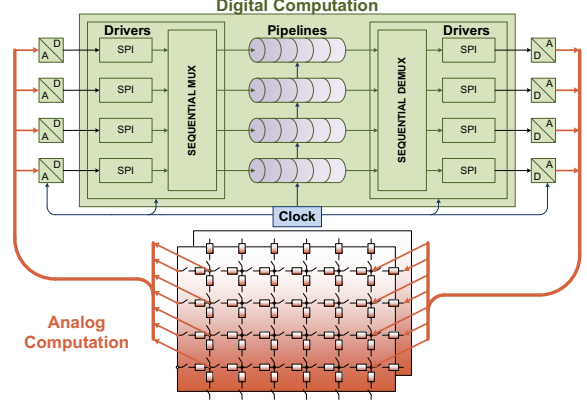

Fig. 4.   Architecture of the mixed-platform system

implementation of the algorithm resides in a low-accuracy environment.

The stability region of the two methods is shown in fig. 3. AB2 has a smaller region of stability, however, when stable, it is expected to perform better than FE because of its higher order. The latter can be of great importance in our application, since the implementation of the algorithm resides in a low-accuracy environment (see also section IV).

III.   TRANSIENT STABILITY EMULATOR

A synoptic view of the emulator is seen on fig. 4. It can be separated into two parts, the analog one that handles the solution of the linear system, shown in red, and the digital one that deals with the numerical solution of the differential algebraic equations, shown in green.

A. Grid Equations

Mapping the topology of a power grid on a miniaturized electronic implementation has been proved feasible in [4]. Starting from (1) and neglecting the conductance under the assumption $\|G\| \ll \|B\|$ for the transmission system:

$$\begin{aligned} Re\{\underline{I}\} &= G \cdot Re\{\underline{V}\} - B \cdot Im\{\underline{V}\} \\ Im\{\underline{I}\} &= G \cdot Im\{\underline{V}\} + B \cdot Re\{\underline{V}\} \end{aligned} \xrightarrow{G \approx 0} \begin{aligned} Re\{\underline{I}\} &= -B \cdot Im\{\underline{V}\} \\ Im\{\underline{I}\} &= B \cdot Re\{\underline{V}\} \end{aligned} \quad (5)$$

In (5), B defines two identical separate grids that link voltage to current quantities. These two grids are built using reconfigurable discrete electronic elements. The connectivity pattern of the analog components can be modified by the use of electronic switches, thus different real-world topologies can be accommodated on the analog lattice. The computing forces of this analog computer are the laws of Ohm and Kirchhoff. This way, intense parallelization is achieved.

Source of inaccuracy

This speed improvement comes at the expense of accuracy. The source of inaccuracy is the imprecise mapping of power system quantities to electronic quantities. This is because of the quantization (finite resolution) and the inherent imprecision of the analog elements. Ideally the real system to be solved would be the one of (5). Instead, the result given by the analog solver corresponds only to an approximation of the original system
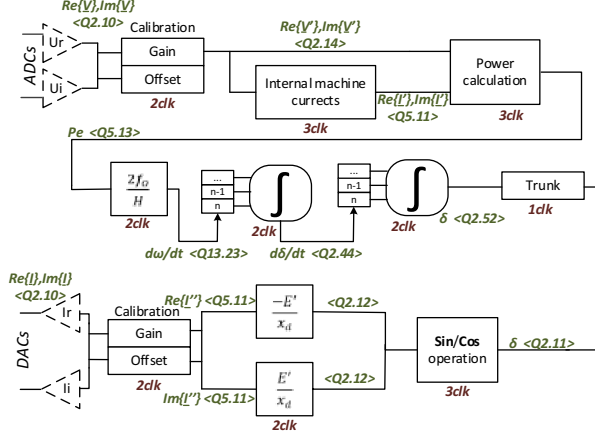
Fig. 5. Synthesized FPGA blocks with respective datapath width and block delay in clock cycles

$$Re\{\hat{I}\} = -\hat{B} \cdot Im\{\hat{V}\}$$
$$Re\{\hat{I}\} = \hat{B} \cdot Re\{\hat{V}\} \tag{6}$$

B. Injection Equations

The hardware that hosts the ensemble of the digital computations is an Altera Cyclone III FPGA.

In the partitioned scheme of fig. 2, the solution of the equations of each injection is independent from the others. The set of injections E is partitioned in K equivalence classes E_k of structurally similar equations. For all i, j members of class E_k, the following holds.

$$f_i|_{\lambda_i = \psi_i(\lambda_k)} = f_j|_{\lambda_j = \psi_j(\lambda_k)}$$
$$g_i|_{\lambda_i = \psi_i(\lambda_k)} = g_j|_{\lambda_j = \psi_j(\lambda_k)} \quad \forall\, i, j \in E_k \tag{7}$$

Where $\lambda_k, k = 1 \dots K$ is selected so that $\exists \psi_m: \lambda_m = \psi_m(\lambda_k)\, \forall m \in E_k$. The above relation means that differences between DAEs of in-class injections are limited to parameters λ. In the emulator implementation there exists one computational module C_k per equivalence class E_k that is able to process injections of that class.

C_k computations are performed in a pipelined fashion. At each pipeline clock cycle, the parameters of the system to be solved are iterated across class injections $\lambda_i, i \in E_k$. As a result the system $S_k(\lambda_i)$ is solved, for all the injections of the class. After all pipelined computations are finished, the nodal injections are concurrently updated and grid computations are performed for the next time step.

Integration modules in these computational pipelines implement the FE and AB2 algorithms. A simple example for the above is illustrated on fig. 5, which shows the pipeline that handles generators that are modeled using the classical model.

Sources of inaccuracy

There are two main inaccuracy sources in the digital part of the hardware. The use of fixed-point arithmetic and the inherent inaccuracies of the integration algorithms.
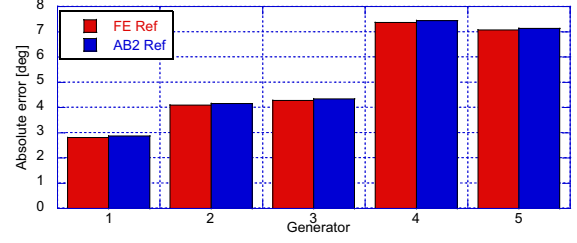

Fig. 6. Minimal time-step emulation error compared to simulation reference

The characteristic of *fixed-point arithmetic* is the fixed number of decimal bits in the representation of a real number. There are two benefits from adopting it in a datapath: (a) faster execution of arithmetic operations and (b) lower silicon footprint of arithmetic modules, i.e. resource utilization in the FPGA. However, this comes at the cost of precision loss. For example the multiplication of two $Qm.f$ numbers (m and f are the number of digits before and after fixed point respectively) is a $Q2m.2f$ number. So, for the computational datapath to remain of bounded width in a series of computations, extra bits have to discarded, i.e. numbers have to be rounded or truncated. Fig. 5 shows a detailed view on the fixed-point datapath. The width of the pipeline appears in green, and the time spent on each block (in FPGA clock cycles) is shown in red. In order to bound truncation inaccuracies, most datapath truncations occur in the dedicated block "trunk" in the figure. Minor truncations occur also elsewhere, but care is taken to minimize the amount of information lost.

Another source of inaccuracy are the *numerical integration algorithms,* an overview of which was given in section III-B. The two algorithms that have been implemented in the pipelines are FE and AB2. The quality of the approximation of the numerical algorithm becomes of critical importance when the environment has many sources of inaccuracy. This will be demonstrated with a series of comparative studies between FE and AB2 in the next section.

IV. COMPARATIVE STUDIES

For the following studies the IEEE 18-bus benchmark system is used. It consists in total of 5 generators, 8 loads and 24 branches.

A. Assessment of absolute precision for different time-steps

The most accurate solution of the emulator comes using a minimum time-step, and only after the calibration of the analog part [4].

A metric to quantify the relative precision of two simulators is the maximum difference of the angle of the same generator in the trajectories coming from the two simulators for the same scenario[6]. $L_i^\infty = \max_t |\Delta\delta_i(t)|$. Fig. 6 presents L_i^∞ for all generators in the test system. Simulations are run on the emulator using a time-step $h = 60\,\mu s$, for both FE and AB2. This value is orders of magnitude less than common practice in TSOs [17], so relative accuracy is expected to be the maximum possible.
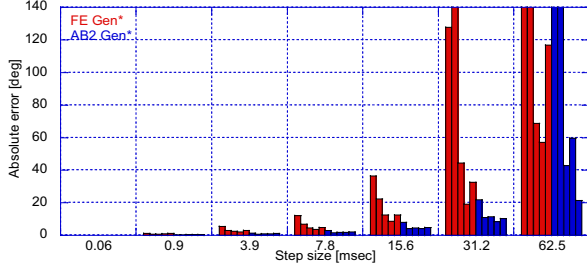
Fig. 7.   Emulation error due to digital imprecision
for different time steps

Fig. 8.   FE instability while AB2 succeeds in retaining
the stability of the numerical solution

Results coming from a PC software 4th-order Runge-Kutta (RK4) implementation are used as a "perfect angle solution" reference. Discrepancies between the best possible analog solution and the software reference can be attributed exclusively to analog imprecision.

As the time step of the hardware emulator increases, its accuracy decreases. This deterioration is due to the effect of digital imprecisions that are gradually stacked even on the best possible case of the minimum time step. Fig. 7 shows this added inaccuracy against the minimum timestep case that is taken as a relative reference. L_i^∞ for generators are displayed in groups of time steps. Naturally, as higher time steps are used the accuracy decreases. However this decrease in the quality and thereof usability of the results, is less pronounced for AB2 than for FE. For example, using a timestep of 15.6 ms yields acceptable results for AB2, but FE instead collapses. In cases of very high time step (e.g. 62.5 msec), both algorithms collapse and a quantitative assessment of L_i^∞ is irrelevant.

A selected case where the contrasting behavior of FE and AB2 is well illustrated is shown on fig. 8, which shows a detailed view on the trajectory of the internal machine angle of a generator after a perturbation. Differences on the amplitude and the exact time instant of the first peak of the trajectory are hinting the global accumulated truncation error of the algorithms. It is clear that the AB2 trajectory is way closer to the "real" one coming from RK4 on software. As the phenomenon evolves, local truncation errors further accumulate in the FE case, and finally render the trajectory unstable after the second peak. It is crucial to note, that this
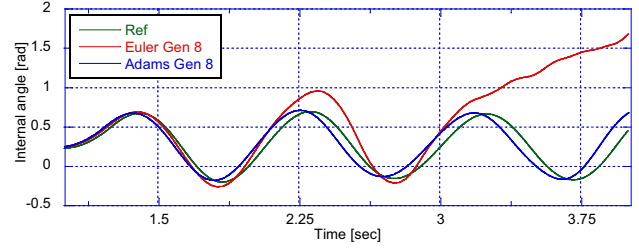
TABLE I
N-1 CONTINGENCIES RESULTS FOR DIFFERENT ALGORITHMS AND TIME STEPS

contingency	sim	FEa	ABa	FEb	AB b	FEc	ABc
250ms @br#1	S	S	S	**U**	S	**U**	S
250ms @br#6	S	**U**	S	**U**	S	**U**	S
250ms @br#10	S	S	S	S	S	**U**	S
250ms @br#13	S	S	S	**U**	S	**U**	S
250ms @br#17	S	S	S	**U**	S	**U**	S
250ms @br#20	S	S	S	S	S	**U**	S
250ms @br#31	S	S	S	**U**	S	**U**	S
410ms @br#32	U	U	U	U	U	U	U
250ms @br#33	S	S	S	**U**	S	**U**	S

sim: RK4 implemented purely on software and taken as reference; FE*: Forward Euler implementations on hardware emulator; AB*: 2-step Adams-Bashforth implementations on hardware emulator; *a: 7.8 ms time step; *b: 15.6 ms time step; *c: 31,3 ms time step
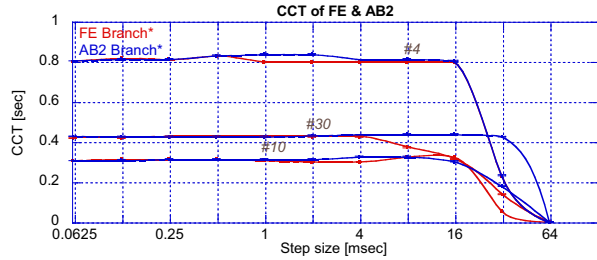
instability is a numerical artifact of the integration algorithm (FE) and it is not owed to a real-life phenomenon. This clearly demonstrates that similar results given by FE are unusable and potentially dangerous for real world operation analyses. This issue is further analyzed in the next section.

B. Effect on n-1 contingency analysis

A common procedure, performed during TSA studies by the system operator, is *n-1 contingency analysis*. In the latter, the operator is interested in knowing whether the system retains its stability, after a perturbation is applied on one of its elements. The outcome of this study are simple boolean answers (stable/unstable) for each of the contingencies in the test set.

Table I summarizes the results of a n-1 contingency analysis for the test system, for a selected set of branch contingencies. Under the *sim* column results coming from a purely software RK4 simulator are shown and taken as reference. It is clearly seen that as the time steps become larger (a→b→c), FE implementations fail to be in accord with the reference, while AB2 implementations succeed in doing so. Boolean results in red bold fonts show this mismatch between FE results and the reference.

C. Effect on CCT analysis

Another commonly performed operation by TSO for which transient simulations are done are Critical Clearing Time (CCT) analyses. CCT determines the maximum duration of a fault on a branch for which the system (even marginally) maintains its stability. The most common algorithm to do so is to perform transient simulations in a binary search fashion for different fault durations. Potentially erroneous results for this type of analysis can be critical for the safety of the system, since many of the coordinated protection schemes on power systems rely on CCT analyses.

Fig. 9 presents CCT results for branches #4, #10 and #30. It is clear from the graphs that AB2 is able to provide reasonable results ($\pm 2.2\%$ discrepancy against the reference) with time steps as high as $\sim 16\,ms$. On the contrary, for FE results to be in the same range, step sizes of $\leq 2\,ms$ have to be used. Since CCT analysis is a repetitive procedure due to the binary search scheme (~ 10 transient simulations for each point of the figures), higher time steps result in considerable computational savings.

This phenomenon becomes even more pronounced when the accuracy of the environment surrounding the numerical integration blocks is further reduced. A similar set of tests

Fig. 9.   CCT for branches #4 ,#10, #30 with
varying time steps using FE & AB2

Fig. 10.   CCT for branches #30 with varying time steps
using FE & AB2, in a minimal accuracy environment

were performed for the power system, this time without the calibration of the analog part (as per [6]). This makes the approximation of the solution of the linear system (6) even wilder, hence reducing the overall accuracy of the solver.

Fig 10 present relevant results for branch #30, for the two algorithms. It is clear that AB2 is far less affected by the drop of accuracy of the linear solution compared to FE. For the latter, we observe results that are off by a margin of 8% against the reference, for time steps as low as 4 ms.

V.   CONCLUSIONS

In this paper, an introduction to transient stability simulation of power systems was given. The formulation of the problem was presented, alongside necessary power system fundamentals, as well as relevant mathematics. A brief overview of a hardware solver dedicated to the problem was given and sources of inaccuracy were identified. Particular attention was given to the pipelined numerical integration scheme and to the effect of different integration algorithms on the quality of the end result.

It was concluded that in environments of high imprecision, such as the one of the dedicated emulator, the use of higher order integration algorithms is indispensable to have results of acceptable quality for higher time steps. The use of higher time steps is translated in computation time saving for the simulation overall.

Extensive empirical studies conducted for the two algorithms, for two realistic type of analyses (n-1 contingencies and CCT) showed that results coming from FE implementations in such an accuracy-starved environment cannot be trusted unless time steps in the range of few hundreds of microseconds are used. For AB2 implementations, time steps of a few tens of milliseconds produce trustable results.

REFERENCES

[1]   H. W. Dommel, and N. Sato, "Fast Transient Stability Solutions," *IEEE Trans. Power App. Syst*, vol. PAS-91, no. 4, pp. 1643-1650, 1972.

[2]   B. Stott, "Power system dynamic response calculations," *Proc. IEEE*, vol. 67, no. 2, pp. 219-241, 1979.

[3]   C. F. Fu, J. D. McCalley, J. Tong, "A Numerical Solver Design for Extended-Term Time-Domain Simulation," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4926-4935.
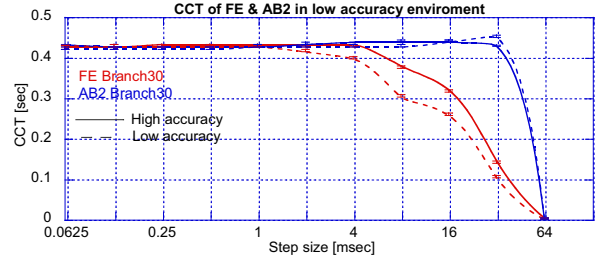
[4]   L. Fabre, G. Lanz, T. Kyriakidis, D. Sallin, I. Nagel, R. Cherkaoui, and M. Kayal, "An Ultra-High Speed Emulator Dedicated to Power System Dynamics Computation Based on a Mixed-Signal Hardware Platform," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4228-4236, 2013.

[5]   T. Kyriakidis, G. Lanz, D. Sallin, G. Lilis, L. Fabre, R. Cherkaoui, and M. Kayal, "A mixed-platform framework for Dynamic Stability Assessment," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, Vancouver, BC, 2013.

[6]   G. Lanz, L. Fabre, G. Lilis, T. Kyriakidis, D. Sallin, R. Cherkaoui, and M. Kayal, "Power network transient stability electronics emulator using mixed-signal calibration," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2013 Proc. 20th Int. Conf.*, Gdynia, Poland, 2013, pp. 369-373.

[7]   K. R. Padiyar, *Power System Dynamics: Stability and Control*. Singapore & New York, NY: John Wiley, 1996.

[8]   J. Y. Astic, A. Bihain, and M. Jerosolimiski, "The mixed Adams-BDF variable step size algorithm to simulation transient and long term phenomena in power systems", *IEEE Trans. Power Syst.*, vol. 9, no. 2, pp. 929-935, 1994.

[9]   T. A. Davis, *Direct Methods for Sparse Linear Systems*. Philadelphia: SIAM, 2006.

[10]   L. Zhuo, and V. K. Prasanna, "Scalable Hybrid Designs for Linear Algebra on Reconfigurable Computing Systems," *IEEE Trans. Comp.*, vol. 57, no. 12, Dec 2008.

[11]   G. Wu, Y. Dou, J. Sun, and G. D. Peterson, "A High Performance and Memory Efficient LU Decomposer on FPGAs," *IEEE Trans. Comp.*, vol. 61, no. 3, Mar 2012.

[12]   V. Jalili-Marandi, and V. Dinavahi, "SIMD-Based Large-Scale Transient Stability Simulation on the Graphics Processing Units," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1589-1599.

[13]   V. Jalili-Marandi, Z. Zhou, and V. Dinavahi, "Larde-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs," *IEEE Parallel Distrib. Syst.*, vol. 23, no. 7, pp. 1255-1266.

[14]   A. S. Deese, C. O. Nwankpa, "Application of analog and hybrid computation methods to fast power system security studies," *International Journal of Electrical Power & Energy Systems*, vol . 33, no. 5, pp. 1140-1150.

[15]   Nwankpa, C., J. Johnson, P. Nagvajara, Z. Lin, M. Murach, and P. Vachranukunkiet, "Special purpose hardware for power system computation," in *Power and Energy Society Meeting (PES), 2008 IEEE*, Pittsburgh, PA, Jul 20-24, 2008.

[16]   N. J. Higham, Accuracy and Stability of Numerical Algorithms. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2002.

[17]   F. P. de Mello, J. W. Feltes, T. F. Laskowski, and L. J. Oppel, "Simulating fast and slow dynamic effects in power systems," *IEEE Comput. Appl. Power*, vol. 5, no. 3, pp. 33-38, 1992.