

Power Grid Transient Stability Prediction Using Wide Area Synchrophasor Measurements

J. Hazra, Ravi Kiran Reddi, Kaushik Das, Deva P. Seetharam, *Member, IEEE* and A. K. Sinha, *Member, IEEE*

Abstract—Electric power systems are prone to various kinds of transient disturbances which exist only for a fraction of second and often trigger cascading failures in. Hence it is important to detect and prevent them from spreading in time. Conventionally these events are prevented by deploying costly special protection systems (SPS). Unfortunately, in many cases SPSs mis-operate as they could not predict the stability well ahead and are designed to operate based on past experiences and extensive off-line simulations. This paper proposes an online transient stability prediction scheme based on live synchrophasor data. The novelty of the proposed method is that it accurately predicts the transient stability based on only few (10 to 12) sample fault data without solving computationally extensive electromechanical dynamics. Synchrophasor data from geographically distributed Phasor Measurement Units (PMUs) are collected, synchronized, aggregated (if required) and analyzed on a stream computing platform to predict the trajectories of the generators which are then used to predict the transient stability of the grid. Performance of the proposed scheme is evaluated on the benchmark systems and evaluation results are presented in this paper.

Index Terms—PMU, Synchrophasors, Transient Stability, Stream computing

I. INTRODUCTION

Modern power systems are continuously monitored by well-trained system operators equipped with sophisticated monitoring and control systems. Despite such precautionary measures, large blackouts are still happening quite frequently. Many of these blackouts originate as a single fault and the consequences of that fault rapidly spread through the network causing massive cascading failures due to voltage instability or generator instability (also called transient instability). While voltage instability is a relatively slow phenomenon (typically, takes a few seconds), transient instability usually happen within a fraction of second. If the unstable generators are not isolated quickly, they will drag the nearby generators also down and cause widespread blackouts. To prevent such large-scale blackouts, proper emergency control actions must be initiated as soon as the transient instability is detected.

Due to the recent advances in the areas of sensing, control, computation and communication technologies, it is becoming possible to take appropriate control actions based on fine-grained measurements of the grid. For instance, the recently developed Phasor Measurement Units (PMUs) could provide upto 120 measurements per second having microsecond accuracy and the stream computing systems enable processing large

volumes of streaming data in memory and extracting useful information from the data in realtime. Power industry interest in stream processing solutions arises from applications with requirements to acquire, analyze, and respond to enormous numbers of complex events in real-time.

Several approaches have already been proposed for transient stability prediction. For example, Scala et al [1] proposed a time domain simulator for transient stability prediction based on solving nonlinear differential equations model of the power system. Pai [2] has proposed a transient energy function for direct stability assessment by comparing the difference between the kinetic and potential energy following a disturbance. This method reduces computing time by avoiding time domain solution of non-linear differential equations. A real time online transient stability prediction method has been proposed by Rovnyak et al [3] where decision trees constructed offline are used to classify a transient swing as either stable or unstable on the basis of real-time phasor measurements. Liu et al [4] have proposed a fast transient stability prediction method using pattern recognition. The method collects an extensive database of rotor angle trajectories by simulating various power system faults and rotor angle trajectory is predicted in real-time by comparing the current trajectory with the most similar (in terms of Euclidean distances) ones in the database. Tao et al [5] introduced a data mining framework for transient stability analysis where support vector machine (SVM) and bagging algorithms are used to improve accuracy and reliability of the stability prediction. Neural networks have been employed by Amjady et al [6] for transient stability prediction where computational burden of the training phase is reduced by a hybrid intelligent system composed of a preprocessor, an array of neural networks, and an interpreter. To counter the inefficiency of common machine learning methods in learning new information, an incremental learning algorithm is proposed by Chu et al [7] to train an artificial neural network for real-time transient stability prediction.

Above mentioned transient stability prediction methods can be broadly classified into three groups i.e. time domain, direct and machine learning. Though time domain simulation is the most accurate way of predicting transient stability, it is difficult to implement in real time as it requires accurate information of the power network topology. Direct methods are faster but are not accurate. Machine learning methods are very fast but again their accuracy highly depends on offline learning. Hence, there is a need for fast and accurate transient stability prediction of Power Systems.

This paper proposes an on-line transient stability prediction method using PMU measurements. The method comprises of four steps, the first step deals with data collection, aggregation

J. Hazra, Kaushik Das and Deva P. Seetharam are with IBM Research, Bangalore, India e-mail: kaushik_msit@hotmail.com, jhazra1@gmail.com, dseetharam@in.ibm.com

Ravi K. Reddi and A. K. Sinha are with IIT Kharagpur, India e-mail: ravikiran88@gmail.com, aksinha@ee.iitkgp.ernet.in

Manuscript received August 1, 2012

and synchronization, second step predicts generator rotor angle trajectories using short-time measurement data, third step models the system based on the derived trajectories and the last step predicts the transient stability. Polynomial curve fitting technique is used to predict the generator trajectories using 10-12 data samples after the fault inception. Based on predicted trajectories, generators are clustered into critical and non-critical sets. Each cluster is then converted to an equivalent single machine and finally stability is evaluated on equivalent machines. The method is implemented in streaming environment and has been evaluated on the standard IEEE benchmark data for the New England bus with 10 machines 39 bus system and one Indian practical systems and evaluation results are presented in this paper.

II. WIDE-AREA SYNCHROPHASOR SYSTEM

A synchrophasor system (as shown in Fig 1) includes phasor measurement units (PMUs) to collect real-time data and a communications system (such as a private utility line, the public switched telephone network, or the internet) to deliver the data from many PMUs to a local data concentrator (usually hosted by the utility that owns the PMU) called Phasor Data Concentrator (PDC). Concentrated data are relayed on a wide-band, high-speed communications channel to a higher-capability data concentrator sometimes called Super Phasor Data Concentrator (SPDC), that feeds the consolidated data from all the PDCs into analytical applications such as a wide-area visualization, state estimator, stability assessment, alarming, etc. Synchrophasor applications need to ingest, process, and analyze continuous data streams from heterogeneous sources. The high volume of streaming data often makes it impossible to fully store and process all the data from disk. Fortunately, emerging stream computing paradigm not only capable of dealing with high volume of streaming data but also enables the extraction of new insights from data in real-time. Further, it provides functionalities like reconfigurability, scalability etc. to the applications which are key requirements for these power system applications.

This paper shows how streaming synchrophasor data could be collected, synchronized, aggregated (when required) and analyzed for real time power system application like transient stability prediction. Monitoring and analysis of these synchrophasor data let observers identify changes in grid conditions, including the amount and nature of stress on the system, to better maintain and protect grid reliability.

III. STREAM COMPUTING

Several power system applications that would need to perform very low latency computations on real-time streaming data. It needs a computational framework that can scale well with increasing large amounts of streaming data and increasingly complex and numerous applications running in parallel on this data. Similar challenges have been faced in other fields. An example is financial engineering, where split second investment decisions have to be made based on computations on large volumes of streaming data [8], often involving data analytics, pattern discovery and model training tasks similar

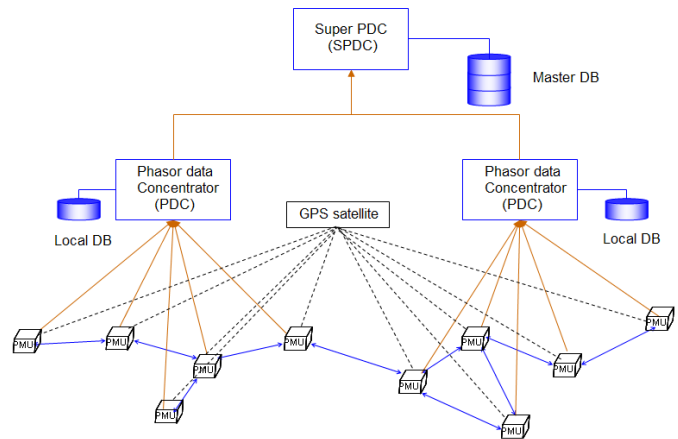


Fig. 1. A synchrophasor network consisting of PMUs, PDCs and communication network

to the case of power system or presently known as smart grid applications. A popular solution emerging for these scenarios is stream computing.

Stream programming is typically done by creating a dataflow graph [9] of operators (as shown in Fig. 2), which performs the computation required by the application. The inputs to the dataflow graph can be data from a variety of sources, such as internet sockets, spreadsheets, flat files, or relational databases, and may be consumed by one or more input operators in the graph. A synchronization primitive is an example of an operator which consumes data from multiple data streams and then outputs data only when data from each stream is read. This can be a useful operator for PMU data which can arrive at different times from different PDCs due to variable network delays and sampling times. Other relevant operators would be a fast Fourier transform (FFT) operator or a moving average operator. Each data element arriving at the input to an operator is typically treated as an event and the operator takes appropriate action when events occur at its input. Operators may be pipelined to perform in a sequential manner: output data from one operator is consumed by the next downstream operator. Operators may also perform in parallel if they do not have data dependencies: output from one operator may be consumed by two or more different operators working in parallel. These operators are typically contained within containers called stream processing elements. For fast, parallel execution, the processing elements are automatically partitioned onto parallel processors and/or machines. The optimal partitioning depends on factors such as the amount and type of data streaming through different processing elements, the resource requirements for each of them and the dependencies between them. Hiding the details of parallel programming from the user greatly improves productivity and efficiency of streaming application deployment. The flexibility of input formats, the ease of developing and connecting the operators, and the automatic compilation onto parallel processors makes stream processing attractive.

Even though stream computing languages make application development quite easy, one may need to redesign traditional

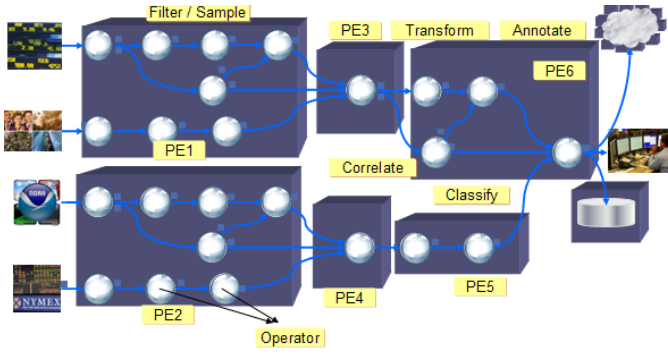


Fig. 2. InfoSphere stream processing graph

algorithms and applications to optimally use the stream processing flow. Reference [10] gives an overview of algorithmic advances and challenges in implementing efficient streaming versions of traditional algorithms. A specific example in this context is [11], where the authors implement decision tree learning for a streaming system. Real-time data will be available from increasingly numerous data sources across the grid. Stream computing frameworks hold the potential to enable scalable real-time applications that can extract information from this data to enable more complete situational awareness and very fast operational response and control.

Although several stream computing platforms [12], [13] have been developed recently, we are using IBM InfoSphere Streams [14], a stream processing middleware from IBM Research that supports high-performance, reconfigurable stream computing.

IV. TRANSIENT STABILITY PREDICTION

Transient stability of any generator is governed by its electric power output and mechanical power input from turbine. During fault, because of large inertia, turbine speed hence mechanical input power does not change instantaneously whereas electric power output changes abruptly. Such power imbalance creates a transient in the system which accelerates or decelerates generator rotor governed by electromechanical swing equation. Mechanical motion of any generator i is governed by the following second order swing equation:

$$\frac{H_i}{\pi f_0} \frac{d^2 \delta_i}{dt^2} + D_i \frac{d\delta_i}{dt} = PM_i - PE_i \quad i = 1, \dots, m \quad (1)$$

where, PM_i is mechanical power input, PE_i is electrical power output, δ_i is i^{th} generator rotor angle, D_i is damping coefficients, H_i is inertia constant, m is number of generators in system.

In real time, direct analysis of grid dynamics by solving differential equations is very difficult as it requires actual grid topology which may not be available and involves significant computational effort. This paper proposes a direct stability analysis method based on short-time measurement data of rotor angle which does not require grid topology and avoids to solve computationally intensive differential equations. Proposed method predicts the stability in three steps. In the first

step, generator rotor trajectories are predicted using short-time streaming data, in the second step using predicted trajectories multi-machine system is reduced to a single-machine infinite bus system, while in third step system transient stability is evaluated on the reduced order system. Each step of the method is described below.

A. Trajectory Prediction

To analyze the grid dynamics in real time, generator trajectories are predicted using short-time measurement data of rotor angle. Polynomial curve fitting is used for this purpose. In polynomial curve fitting, the rotor angle trajectory is fitted as a n^{th} order polynomial function as follows:

$$\delta(t) = a_0 + a_1 * t + a_2 * t^2 + \dots + a_n * t^n \quad (2)$$

where, $a_0, a_1, a_2, \dots, a_n$ are coefficients of polynomial. The values of $a_0, a_1, a_2, \dots, a_n$ are obtained using least squares estimation.

B. Model Reduction

Predicted trajectories are used to model the grid as an equivalent single-machine infinite bus system (SMIB). During large disturbances, entire machines in a power system can be divided into two group's namely critical group, which accelerates with respect to other group and the non-critical group [15]. In our approach, predicted trajectories are used to form two equivalent machines, one representing critical machines and other representing non-critical machines. To cluster the generators into critical and non-critical groups, post fault rotor angle deviation of each generator is computed with respect to center of inertia angle (COA) and compared with the pre-fault largest relative angle (δ_{max}^{pre}). Center of angle (COA) and pre-fault δ_{max}^{pre} are calculated as follows:

$$COA = \frac{\sum_{i=1}^n (H_i * \delta_i)}{\sum_{i=1}^n H_i} \quad (3)$$

$$\delta_{max}^{pre} = \max \sum_{i=1}^n \sum_{j=i}^n (|\delta_i - \delta_j|) \quad (4)$$

where n is number of generators in power system, δ_i and δ_j are rotor angles of i^{th} and j^{th} generators respectively.

If the relative angle between δ_i and COA is greater than δ_{max}^{pre} , then machine i is identified as a critical machine. After identification of critical and non-critical clusters, each cluster is reduced to a single machine. Let, cm represents the critical machine group and nm represents the non-critical machine group. For each group, single machine equivalents of rotor angle (δ), generator inertia (H), mechanical input power (PM) and electrical output power (PE) are formed as follows:

$$\delta_{cm/nm} = H_{cm/nm}^{-1} * \sum_{i \in cm/nm}^n (H_i * \delta_i) \quad (5)$$

$$H_{cm/nm} = \sum_{i \in cm/nm}^n H_i \quad (6)$$

$$PM_{cm/nm} = \sum_{i \in cm/nm}^n PM_i \quad (7)$$

$$PE_{cm/nm} = \sum_{i \in cm/nm}^n PE_i \quad (8)$$

The one machine infinite bus equivalent or single machine equivalent (*sme*) model of the multi-machine power system is obtained using equivalent critical and non-critical machines as follows:

$$\delta_{sme} = \delta_{cm} - \delta_{nm} \quad (9)$$

$$H_{sme} = (H_{cm} * H_{nm}) / (H_{cm} + H_{nm}) \quad (10)$$

$$PM_{sme} = H_{sme} * (PM_{cm} / H_{cm} - PM_{nm} / H_{nm}) \quad (11)$$

$$PE_{sme} = H_{sme} * (PE_{cm} / H_{cm} - PE_{nm} / H_{nm}) \quad (12)$$

C. Stability Evaluation

In this step, power-angle characteristics ($PE = P_{const} + P_{max} \sin(\delta)$) of the equivalent machine is derived using machine learning. Time domain simulations were used to generate the actual trajectories for different possible scenarios and the generated knowledge was used to tune the model parameters of power angle characterizes. The derived power-angle characteristic is used to calculate accelerating (A_1) and decelerating (A_2) areas as follows :

$$A_1 = \int_{\delta_0}^{\delta_c} (PM - P_{max} \sin(\delta)) d\delta \quad (13)$$

$$A_2 = \int_{\delta_c}^{\delta_m} (P_{max} \sin(\delta) - PM) d\delta \quad (14)$$

where, δ_c is rotor phase angle at fault clearance, δ_0, δ_c are phase angles when electrical power is equal to mechanical power.

The system is stable if A_2 is greater than A_1 . The difference between A_2 and A_1 gives the margin of stability.

The steps of the simulation are as follows:

- 1: Obtain real time measurements about rotor angles and powers of various generators using PMU.
- 2: Find maximum value of relative angle between any two generators (δ_{max}^{pre}) during pre-fault conditions.
- 3: Find center of inertia angle (COA) of the system at each time stamp.
- 4: Obtain relative rotor angles with respect to COA at each time stamp for all the generators.
- 5: Use polynomial curve fitting method to obtain relation for relative rotor angles. Use the relation to predict the future relative rotor angles.
- 6: Use the predicted rotor angles and δ_{max}^{pre} to identify critical and non-critical machines.

- 7: Form single machine equivalents of critical machines and non-critical machines to obtain two machine system.
- 8: Form single machine equivalent of the system using critical and non-critical equivalent machines.
- 9: Use electric power and rotor angle values of equivalent model at different time stamps to obtain relation between electric power and rotor angle.
- 10: Use $P_e - \delta$ curve and mechanical power of equivalent model to obtain acceleration and deceleration areas.
- 11: Predict the stability of the system using acceleration and deceleration areas. If the system is stable obtain the stability margin.
- 12: Obtain new set of reading from PMUs and repeat the entire process from step 3.

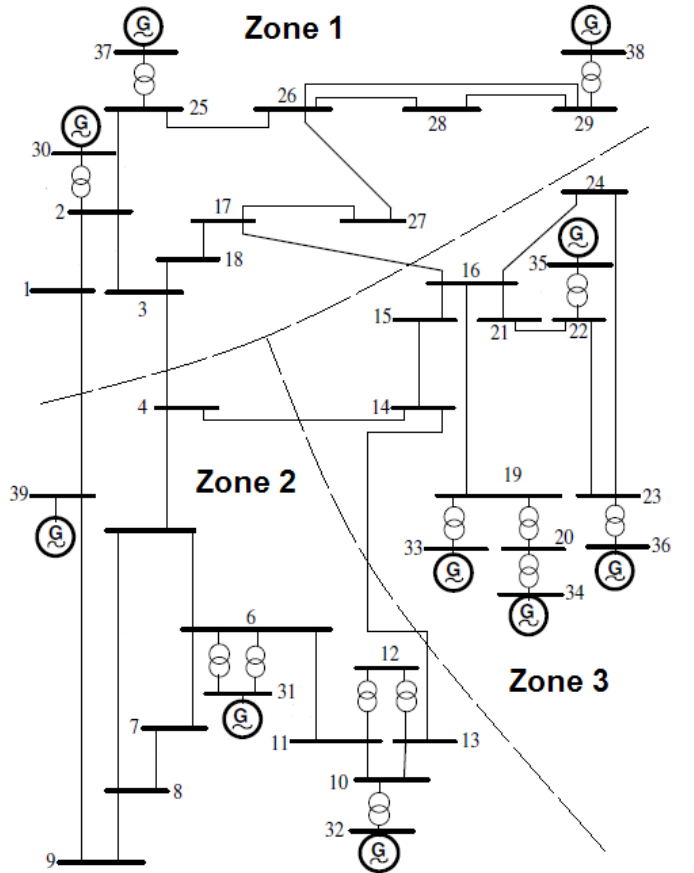


Fig. 3. New England 39 bus test system

V. STREAMS IMPLEMENTATION

In this paper, IBM InfoSphere Streams [14] which supports high-performance, reconfigurable stream computing is implemented. In InfoSphere Streams computing, applications can be scaled to a large number of compute nodes and can interact at runtime through stream importing and exporting mechanisms. InfoSphere Streams applications take the form of dataflow processing graphs consist of a set of processing elements (PEs) connected by streams, where each stream has a fixed schema and carries a series of tuples. The PEs are containers

that host operators implementing data stream analytic, and are distributed on compute nodes. Compute nodes are organized as a shared-nothing cluster of workstations or as the execution nodes in a large supercomputer such as the IBM Blue Gene. PEs communicate with each other via their input and output ports, connected by streams.

Fig. 4 shows SPADE (stream processing application declarative language) application graph for transient stability prediction. The application graph consists of 7 blocks and each block does a part of the analysis and adds the derived features to the forwarded streams. Transient stability application requires only generators data, hence will subscribe generator data from each phasor data concentrator (PDC). The application reads the generator data from PDCs using *Source* operator in block 1 of SPADE graph in Fig. 4. *Source* operator supports universal resource locators (URIs) like file, UDP datagram-based socket, and TCP socket connection and creates stream from data flowing from the external sources. *Source* operator has the flexibility to read selected attributes in a tuple. As this application needs generator rotor angle (δ) and electric power output (PE) data, *source* operator will select only these two attributes along with the time stamp. Each PMU at generating stations feeds at a rate of 100 Hz and hence, needs synchronization among different samples. In this application *Barrier* operator as shown in block 2 is used for synchronization purpose. *Barrier* consumes tuples from multiple streams, outputting a tuple only when a tuple from each of the input streams has arrived. Synchronized generator data are then used in block 3 for rotor trajectory prediction using *Functor* operators. *Functor* operator could be used for performing tuple-level manipulations such as filtering, projection, mapping, attribute creation and transformation. In trajectory prediction block, first *Functor* is used to find the fault initiation and clearance times, maximum angle difference δ_{max}^{pre} between any two generators and center of angle COA for the generators. Then first column of *Functor* operators is used to isolate the individual generator for parallel processing. Second column of *Functor* operators is utilized to predict the individual rotor angle trajectories concurrently. For any generator, trajectory is predicted using current sample and past few samples. InfoSphere stream could access any attribute x of past tuple n using the notation $\hat{n}.x$. In block 4, predicted trajectories of the generators are combined using *Barrier* operator and clustered into critical and non-critical machines using *Functor* and *Join* operators. Equivalent critical and non-critical machines are then reduced to single machine infinite bus system in block 5. Finally, in block 6 transient stability is evaluated on the equivalent system within the *Functor* operator. Output results are then sent to components that are outside of a InfoSphere Streams system. The *Sink* operator is used to perform this externalization that can write to files, sockets and other external devices.

VI. SYSTEM EVALUATION

Application was tested on the standard IEEE benchmark New England 10 machine 39 bus test system as shown in Fig. 3 and 390 bus NREB (Northern Region Electricity Board)

system having 754 transmission lines, 205 transformers, 323 generators, 1 Static VAR Compensator (SVC) and 1 HVDC link. It was assumed that each bus or substation has a PMU sensor and reports to the local phasor data concentrator. These PDCs in turn forward the measurements to a central super PDC (SPDC) that hosts the transient stability application. To generate the PMU data a time domain transient stability analysis simulator was used where system was assumed to operate at 50Hz and sampling frequency was chosen as 100Hz. Random faults were simulated and generated time series data are ingested by InfoSphere streams to run the application. As in general any fault is cleared within 6-7 cycles (2 cycle for relay operation and 4-5 cycles for circuit breaker operation), in all test cases we have used 6 cycle ($=6*20ms=12$ samples) data for predicting the rotor angle trajectories of the generators.

In order to choose the right order of the polynomial curve fitting, several test cases were analyzed for different test systems. Figures 5, 6, 7 and 8 show the predicted trajectories of rotor angles (of most and least affected generators) using different orders of polynomial curve fitting for a dead short-circuit at bus 16 in 39 bus system. To verify the accuracy, predicted trajectories are compared with the actual trajectories obtained using time domain solution of the differential equations involve in the transient. From the simulation results, it is clear that performance does not necessarily improves with the increase in order of the polynomial. For example, Fig 8 shows that 6th order polynomial curve fitting cannot predict trajectories as good as 4th or 5th order can predict. For 6th order polynomial, the coefficients obtained using the small measurement period of 120 ms sample set fail to capture the trajectories. However, higher order polynomial may perform well by increasing the measurement period but this will lead to decrease in time available for emergency control actions before the system loses stability.

Performances of different trajectory prediction methods are also compared. Fig. 9 and Fig. 10 show the trajectory predictions for New England system and NREB system respectively. For new England system temporary fault was simulated at bus 9 whereas for NREB system fault was simulated at bus 1228. In each case, trajectory of only the most affected generator is presented. From Fig. 9 and Fig. 10 it is clear that 4th and 5th order polynomial curve fittings predict the trajectory more accurately than other methods. However, 4th order polynomial is preferable as it predicts the trajectories most accurately and requires less number of co-efficient calculation which saves computational time.

A comparative study of the range of rotor angle prediction was made considering a tolerable limit of 2^0 . Prediction ranges of Taylor series expansion, Ohura et al's method [16], autoregression method and polynomial curve fitting (CF) method for both the test systems are presented in Table I. From Table I it is clear that 4th order polynomial curve fitting outperforms others.

To identify critical and non-critical set of generators for the given fault, predicted rotor angle trajectories with respect to center on angle of all the generators of New England System are plotted in Fig. 11. Fig. 11 shows that in this case two generators (6 and 7) swing fast with respect to the rest of the

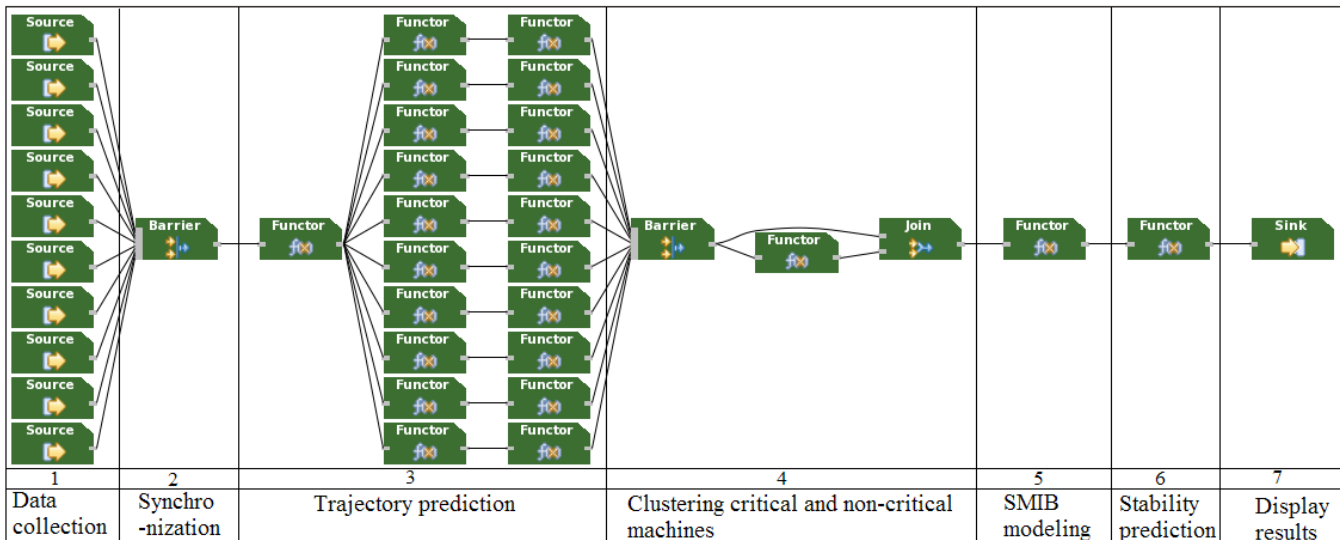


Fig. 4. Application graph for transient stability prediction

TABLE I
RANGE OF PREDICTION WITH ERROR LESS THAN 2⁰

Method	Range of Prediction (milliseconds)	
	39 bus system	NREB system
Taylor series expansion	80	125
Method by Ohura et al	165	195
Autoregression Method	210	200
3rd order polynomial CF	195	180
4th order polynomial CF	285	260
5th order polynomial CF	270	260

generators. Hence, generators 6 and 7 are grouped as critical machines and others are treated as non-critical machines. To evaluate the validity of this grouping actual trajectories with respect to COA are presented in Fig. 12 which clearly shows predicted trajectories with respect to COA are highly accurate. These trajectories are used to model the grid as a single machine infinite bus system and grid stability is evaluated on this reduced grid model as discussed in Section V. Stability prediction method gives an indication of whether grid is going to stable or not along with a stability margin. Sliding window is used for stability prediction. When a new tuple comes in, old tuple is dropped and stability is evaluated. Fig. 13 shows a case study where fault exists 9 cycles in the system and system becomes unstable. Fig. 13 predicts that transient stability margin of the system will be reduced to zero at 0.48 second. This monotonic reduction in stability margin could be a useful indication of grid collapse and as the proposed method predicts such grid behavior well ahead (around 300ms) of time, it could be very useful to take emergency action in real time.

An execution time comparison was made between stream computing based transient stability prediction method and time domain simulation on a single node Linux machine. While time domain simulation method takes 1670 ms to predict the stability of New England system, stream computing based method takes only 1 ms for stability prediction. As transient instability typically happens in 500-600ms after fault initiation, and proposed method can predict the stability within 200 ms

(typically 75ms for communication delay+120 ms fault data collection+1 ms computation) after fault initiation, 300-400 ms could be very crucial for triggering the emergency control. This huge computational improvement makes the proposed method applicable for real time implementation. In-built distributed computational capability of InfoSphere streams could be easily exploited to increase the scalability for real time application.

VII. CONCLUSIONS

This paper described a synchrophasor based online transient stability prediction using the stream computing paradigm. Simulation results presented in this paper show that the proposed method predicts transient stability well ahead of time. The key benefits of this method are that it does not require power system models and can accurately predict the stability in real time based on streaming sensor data. As the method could predict the grid stability well ahead of time, it provides sufficient time to take emergency control actions to prevent cascaded outage leading to system blackout. Because of inherent scalability of stream computing, the application could be easily deployed to handle any real life power network.

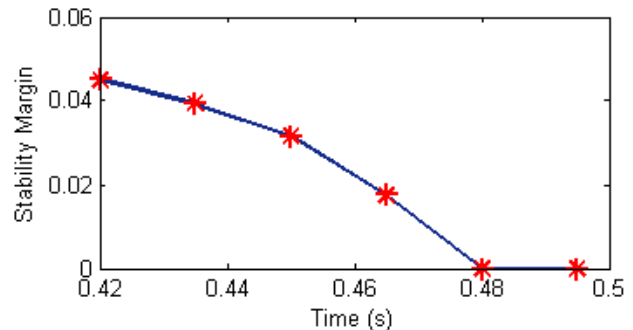


Fig. 13. Transient stability margin prediction for 39 bus system

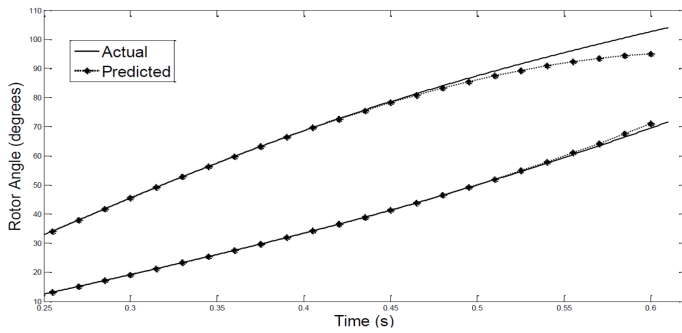


Fig. 5. Predicted rotor angle trajectories for 39 bus system using 3rd order polynomial curve fitting

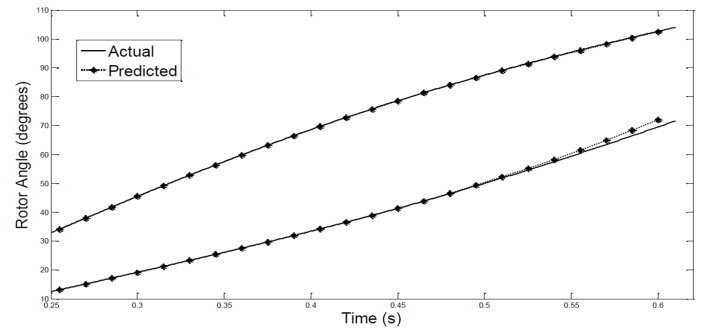


Fig. 6. Predicted rotor angle trajectories for 39 bus system using 4th order polynomial curve fitting

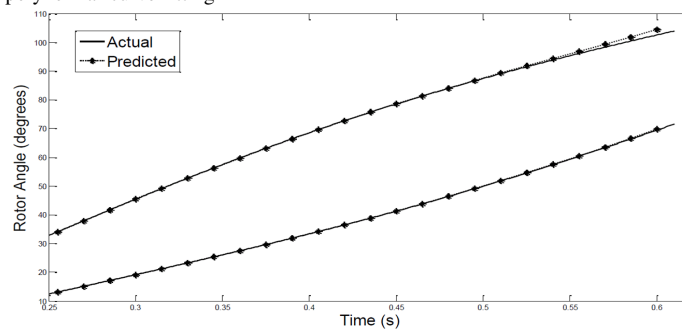


Fig. 7. Predicted rotor angle trajectories for 39 bus system using 5th order polynomial curve fitting

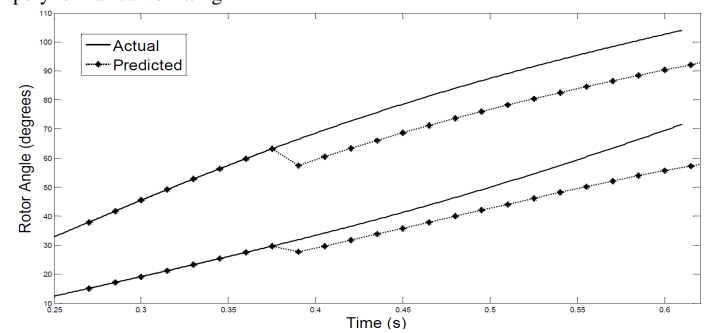


Fig. 8. Predicted rotor angle trajectories for 39 bus system using 6th order polynomial curve fitting

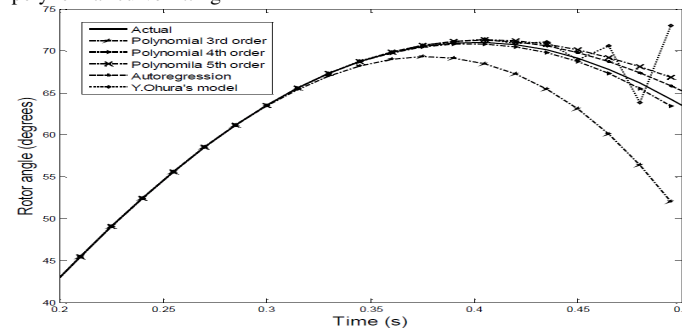


Fig. 9. Predicted rotor angle trajectories for 39 bus system using 4th order polynomial curve fitting

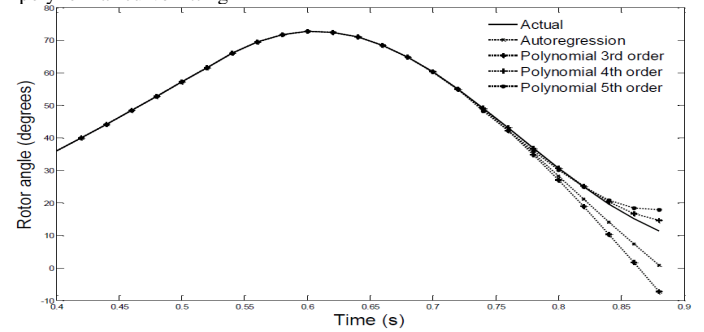


Fig. 10. Predicted rotor angle trajectories for NREB bus system using the model by Ohura et al

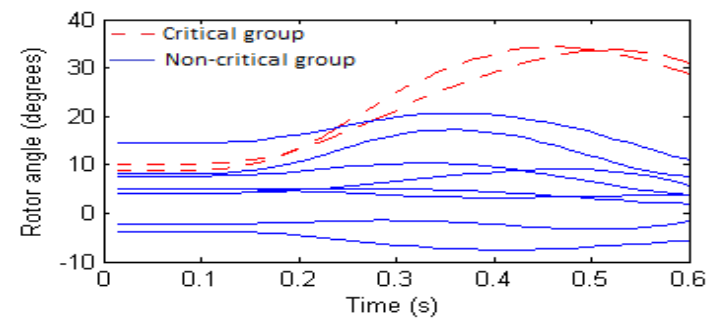


Fig. 11. Predicted rotor angle trajectories for 39 bus system with respect to COA

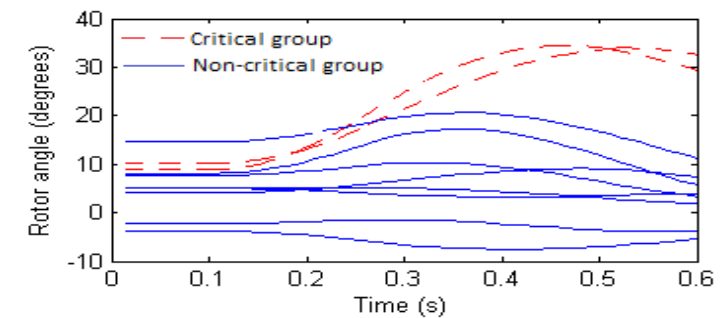


Fig. 12. Actual rotor angle trajectories for 39 bus system with respect to COA

VIII. REFERENCES

- [1] M. La Scala, R. Sbrizzai, F. Torelli, and P. Scarpellini, "A tracking time domain simulator for real-time transient stability analysis," *IEEE Trans on Power Systems*, vol. 13, no. 3, pp. 992–998, 1998.
- [2] M. A. Pai, *Energy Function Analysis for Power System Stability*. Boston, MA:Kluwer, 1989.
- [3] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision trees for real-time transient stability prediction," *IEEE Trans on Power Systems*, vol. 9, no. 3, pp. 1417–1426, 1994.
- [4] X. D. Liu, Y. Li, Z. J. Liu, Z. G. Huang, Y. Q. Miao, Q. Jun, Q. Y. Jiang, and W. H. Chen, "A novel fast transient stability prediction method based on pmu," in *Power & Energy Society General Meeting*, 2009, pp. 1–5.
- [5] X. Tao, H. Renmu, W. Peng, and X. Dongjie, "Applications of data mining technique for power system transient stability prediction," in *Proceedings of the 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies*, 2004, pp. 389–392.
- [6] N. Amjady and S. F. Majedi, "Transient stability prediction by a hybrid intelligent system," *IEEE Trans on Power Systems*, vol. 22, no. 3, pp. 1275–1283, 2007.
- [7] X. Chu and Y. Liu, "Real-time transient stability prediction using incremental learning algorithm," in *IEEE Power Engineering Society General Meeting*, 2004, pp. 1–5.
- [8] H. Wu, B. Salzberg, and D. Zhang, "Online event-driven subsequence matching over financial data streams," in *ACM SIGMOD Intl. Conf. on Management of Data*, 2004.
- [9] B. Gedik, H. Andrade, K. L. Wu, P. Yu, and M. Doo, "Spade: The system's declarative stream processing engine," in *SIGMOD*, 2008, pp. 1123–1133.
- [10] S. Muthukrishnan, *Data streams: algorithms and applications*. Now Publishers, 2005.
- [11] G. Hulthen, L. Spencer, and P. Domingos, "Mining time changing data streams," in *ACM Conf. on Knowledge Discovery and Data Mining*, 2001.
- [12] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. S. Maskey, A. Rasin, E. Ryzkina, N. Tatbul, Y. Xing, , and S. Zdonik, "The design of the borealis stream processing engine," in *Proc. CIDR*, 2005, pp. 277–28.
- [13] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein, and J. Widom, "Stream: The stanford stream data manager," *IEEE Data Engineering Bulletin*, vol. 26, no. 1, pp. 19–26, 2003.
- [14] R. Rea and K. Mamidipaka, "IBM infosphere streams: Enabling complex analytics with ultra-low latencies on data in motion," IBM, White Paper, 2009.
- [15] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient stability of power systems, a unified approach to assessment and control*. Springer Publications, 2000.
- [16] Y. Ohura, M. Suzuki, K. Yanagihashi, M. Yamaura, K. Omata, T. Nakamura, S. Mitamura, and H. Watanabe, "A predictive out-of-step protection system based on observation of the phase difference between substations," *IEEE Trans Power Delivery*, vol. 5, no. 4, pp. 1695–1704, 1990.

IX. BIOGRAPHY

J. Hazra received his Phd from Indian Institute of Technology, Kharagpur in 2008. His thesis was on Identification and Prevention of Catastrophic Failures in Power Systems. He worked as a post doctoral researcher at SUPELEC, France from 2008 to 2010. During post doctoral research He worked on designing HVDC controllers for improving grid stability. From April, 2010, he is working with the Energy and Utility group at IBM Research. His research interest includes smart grid, network congestion management, blackout prediction, HVAC, renewable energy, etc.

Ravi Kiran Reddi received his M.Tech(Master of Technology) in Power System Engineering from Indian Institute of Technology, Kharagpur in 2011. His thesis was on Real Time Large Signal Rotor Angle Stability using Synchrophasor Measurements. From July, 2011, he is working with BHEL Corporate Research and Development. His research interest includes Power system protection, Substation Automation, Grid Stability Control using HVDC, Wide Area Monitoring and Control of Power Systems Using Synchrophasor Measurements, Smart Grid.

Kaushik Das received his M.Tech(Master of Technology) in Power System Engineering from Indian Institute of Technology, Kharagpur in 2011. His thesis was on Real Time State Estimation incorporating Synchrophasor Measurements. From July, 2011, he is working with the Energy and Utility group at IBM Research. His research interest includes Wide Area Monitoring and Control of Power Systems Using Synchrophasor Measurements, Integration of Renewable Energy Resources, Smart Grid, Application of Stream Computing in Power Systems.

Deva P. Seetharam leads IBM India's Smarter Energy research initiatives. Previously, he led France Telecom's electronic paper display research group in Boston. He has also worked on various embedded wireless networking technologies including active RFID and wireless sensor networks. On the entrepreneurial side, he co-founded RadioSherpa, a company that developed innovative HD Radio technologies and applications, and he was instrumental in selling the company to RadioTime Inc. He holds a MS degree from the MIT Media Laboratory and a BE degree from the Bharathiar University.

Avinash K. Sinha joined IIT Kharagpur in 1984, where he is currently a Professor in the department of Electrical Engineering. He was head of this department from 2007-2010. He was a Visiting Professor at Washington State University, Pullman (2001-2002 and 2011). He received IBM Open Collaborative Faculty Award for the years 2010 and 2011. At IIT Kharagpur he is leading the Power Systems group and his team has developed a digital simulator for power systems. His research interests include Power System Analysis, Simulation of Power System Dynamics, AI Applications to Power Systems, Synchro-phasor applications to power Systems.