



21, rue d'Artois, F-75008 PARIS

<http://www.cigre.org>

**CIGRE US National Committee
2015 Grid of the Future Symposium**

**On the Need for Robust Decentralized Coordination to Support Emerging
Decentralized Monitoring and Controls Applications in Electric Power Grid.**

P. Banerjee, S. Niddodi, H. Lee, A. Srivastava and D. Bakken
The School of Electrical Engineering and Computer Science
Washington State University

Patrick Panciatici
RTE France

SUMMARY

The most common (by far) control architecture for power system monitoring and control utilizes centralized control center (CC), complemented with limited local control such as protection, transformer tap changers and reactive power control. However, with the ongoing and future transitions to the smart electric power grid, the CC based control is becoming unsuitable for the grid's changing demands. The penetration of renewable energy resources, increase in number of field sensors, addition of connected devices, real time decision support expected from energy management systems (EMS), increase in intermittent loads (battery charging for electronic vehicle and demand response) and single point failure caused by extreme events, motivates the need for supplementing and transforming the conventional CC based power system control.

The existing local controls in the power system network are initiated based on the local disturbances and limited network wide observability, which results in suboptimal system performances. The alternative architecture for power system monitoring and control is to supplement existing CC and local control with more decentralize control, whereby the decentralized nodes form a group to cooperate. Such groups must be coordinated using system information and measurements; they are often smaller group of node which facilitates fast local control and increased topological visibility for best performance from a system wide point of view. However, such coordination of decentralized logic is extremely complicated for advanced programmers with computer science degrees. For power engineers with no such background, it is likely that many boundary cases are missed (variable latencies, message arrivals, failure modes, etc).

The main contribution of this paper is overview of the trends that are causing the deployment of more decentralized power algorithms, and explain the needs for correct and

banerjee.param@gmail.com

resilient coordination. We also overview a software platform, *DCBlocks*, which we are developing in order to make 35+ years of both applied and theoretical computer science research in such coordination much more accessible to power engineers. Finally, we overview an application that uses it.

KEYWORDS

Decentralized Control, Distributed Computation, Voltage Stability

I. INTRODUCTION

The power system network typically evolved as a centrally coordinated (CC) control system [1], where the measurement data flows from sensors placed at different locations in the field to the control center and control action from the control center to the field equipment. This control scheme has been successfully operated for several decades, because power generation was predominantly aggregated and distribution system was passive and radial. The recent trends of penetration of renewable energy sources in the distribution system and availability of large data from the sensors are both the driving force for Decentralized Coordination (DC) architecture of the present and foreseeable future power system network.

The existing CC based control centers have number of power system applications like state estimation, congestion management, oscillation monitoring, supplementary FACTS controller etc. These applications are designed to function reliably with full system topology and complete network observability. The bandwidth requirement for transmitting large measurement data from sensors to CC based control centers have tremendously increased recently. On contrary the faster local control action installed in the power system have limited system visibility and may degrade the system performance while improving the local scenario. The power system stabilizer (PSS) installed at the generating stations is observed to have little impacts on the inter area oscillations [2]. The third zone protection system of distance relays may lead to cascading outages during power system oscillations and voltage instability [3]. The on-load transformer tap changers may accelerate voltage collapse due to the reverse action for some operating points [4]. The problem of sub-synchronous resonance may be initiated due to local switching of the capacitor banks in the series compensated lines [5]. The DC based control centers typically require lesser measurements which are locally available, and in turn reduce the burden on the limited bandwidth available for data communications.

The renewables integration and distributed generations are already showing limitations of CC based control. The fidelity and connectivity exponentially increases when single large power generating sources are replaced by multiple small capacity renewables. This increases the data traffic for the CC based control centers to effectively monitor and control large number of small capacity distributed generations.

Remedial Action Schemes (RAS) are critical for future power grids. The RAS schemes are typically configured during installation, which are expected to operate after many years. The change in network topology and variation in operating point may cause the pre-configured RAS scheme to mal-function. It is highly desired that the RAS schemes dynamically updates the logic parameters considering other RAS schemes, installed in the adjacent substations. This is possible with the integration of DC architecture based control.

The Energy Management System (EMS) installed at the CC based control centers may initiate a control action for field equipment, which may be geographically apart. The latency may be significant [6], leading to unintended operating scenarios, if strict time deadline is not followed. The changing nature of the loads, such as power electronics based battery charger for electronic vehicles and converter based motor drives are also posing challenges to the existing CC based power management applications due to uncertainty and intermittent nature. Both of these require fast control actions. The latency can be substantially reduced if fast control action can be coordinated with relevant system information locally and the CC based EMS is relieved of the control action in non-critical time bounds.

The increase in Information and Communication Technology (ICT) infrastructures in the future smart grids also expose the critical power system installations to software bugs, and cyber-attacks [7]. The power system networks operated by a CC based control center are at a high risk of failures, if in case, the single point of mal-operation occurs at the CC based control center. The reliability and the security of the power system network can be enhanced considerably if the decision making points are distributed over multiple substations.

II. DECENTRALIZED MONITORING AND CONTROL OF ELECTRIC POWER GRID

The transition of different power system application from CC based architecture to DC based architecture is slow due to technology evolution and capital investments. The power system network is grouped using overlapping areas and decentralized optimal power flow control is proposed in [8]. The relaxation Lagrangian based decentralized state estimation for fast and accurate convergence is demonstrated in [9]. The comparison of the communication infrastructures for wide area measurement systems is carried out in [10], in which the latency and reliability is observed to be improved for the decentralized implementation. The decentralized reactive power control using multi-agent systems for the distribution network is proposed in [11]. The partitioning of regions based on bus voltage sensitivity and decentralized voltage regulator using fuzzy logic is proposed in [12] for improving voltage profile. The impact of renewables in the distribution grid is studied in [13], along with recommendation for proper designing and operation of distribution grid.

The control strategy of inverters with communication link [14] and without communication link [15] is proposed for stability enhancement in microgrids. The interaction of multiple grid support inverters may result in adverse impact when local volt-var control is performed [16]. A roadmap for selecting the proper control modes and settings for inverters in the distribution grid is presented in [17].

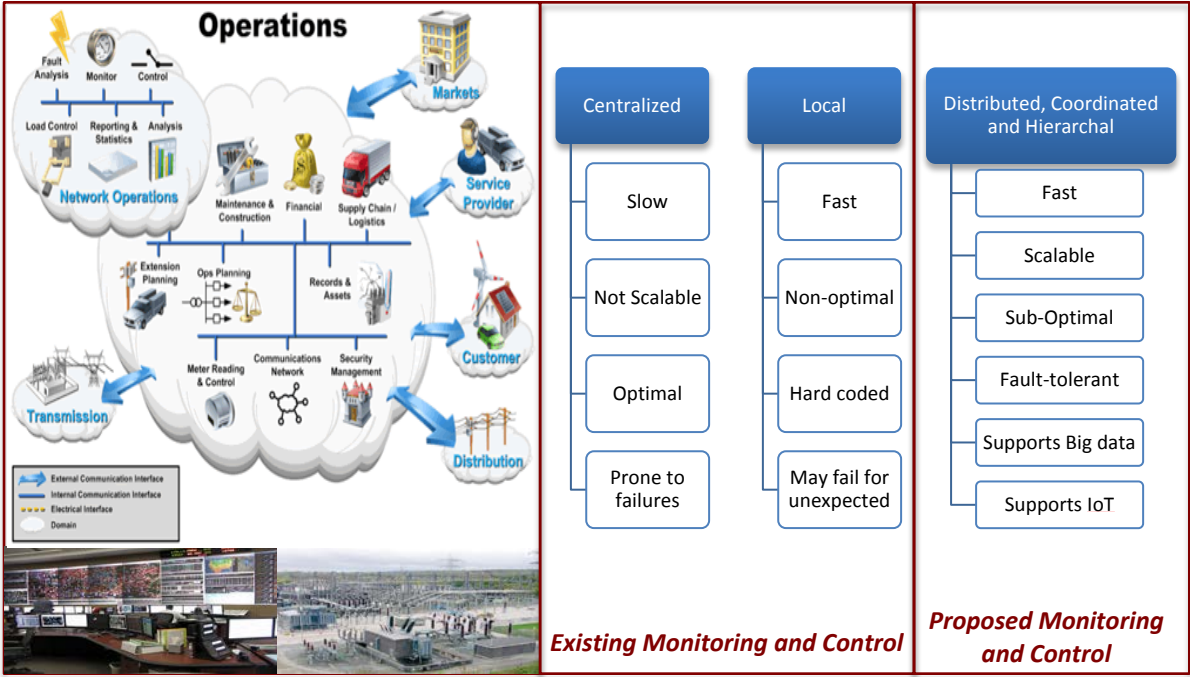


Fig. 1: Salient features of future grid and comparison of existing and proposed monitoring and control strategies.

The future power grid will experience the interaction of large number of third party entities like markets, service providers, prosumers, and distribution, transmission and network operations as shown in Fig. 1. Such smart power system network will not be centralized as it is slow, non-scalable and prone to failures. The local control, which is non-optimal, hardcoded and not fault tolerant is also not a choice. The distributed, coordinated and hierarchal architecture is the possible architecture suited for such a system as it will be fast, scalable and fault tolerant. The robust computational algorithms available for DC based control are also required for future power grid application. The communicating systems in the DC architecture also support Big Data and Internet of Things technology, for handling communicating sensors, controllers, switches and servers.

III. DISTRIBUTED COORDINATION ARCHITECTURE

The DC architecture is based on distributed system in which application logic (“processes”) is installed in multiple computers in the network trying to achieve a well-defined objective using a group messaging interface. Such message passing mechanisms come from the field of distributed computing, which asks the question “how can we best use computational resources and computer networks in order to help distributed process” [18].

The communicating systems utilized in DC architecture are heterogeneous in various aspects. The hardware platform may differ depending on the micro architecture. The operating system running above the processor may also differ among different communicating systems. The application development platform may also be different among various systems. The user interface (UI) for similar power system application also differs from one vendor to another. This vast heterogeneity of the ingredients system in a DC architecture is handled by middleware which provides seamless mechanism and higher level of abstraction for the end user applications such that interoperability is provided across programming language, operating system, and CPU architecture. The middleware in the DC architecture allows for concealment of separation of components, so that the system is perceived as a whole instead of collection of independent components. It allows for opaqueness in terms of access and location of resources. Remote and local resources can be accessed using identical operation without knowing its actual physical location. It is fault tolerant and can also support failure recovery either through checkpoint and rollback mechanism or through state machine replication.

An example DC architecture system is shown in Fig 2. It consists of front-end users, dispatchers and groups of workers responsible for performing specific application tasks requested by the frontend users. The group of worker can constitute to a work site. The request from the frontend user is received by the corresponding dispatcher which then forwards it to its workgroup and collects the result and sends it back to frontend user. Depending on the amount of workload, the dispatchers can communicate with each other to request computational resources (workers) from adjacent sites. The dispatchers need not be aware of the IP address of other dispatchers and can discover each other through naming service. Further, the dispatchers sends the work requests to the workers using reliable multicast by maintaining the proper ordering of events. The various systems in the DC architecture can communicate with each other without the requirement of global clocks. Time synchronized events can be performed using logical clocks or establishing happened-before ordering of events.

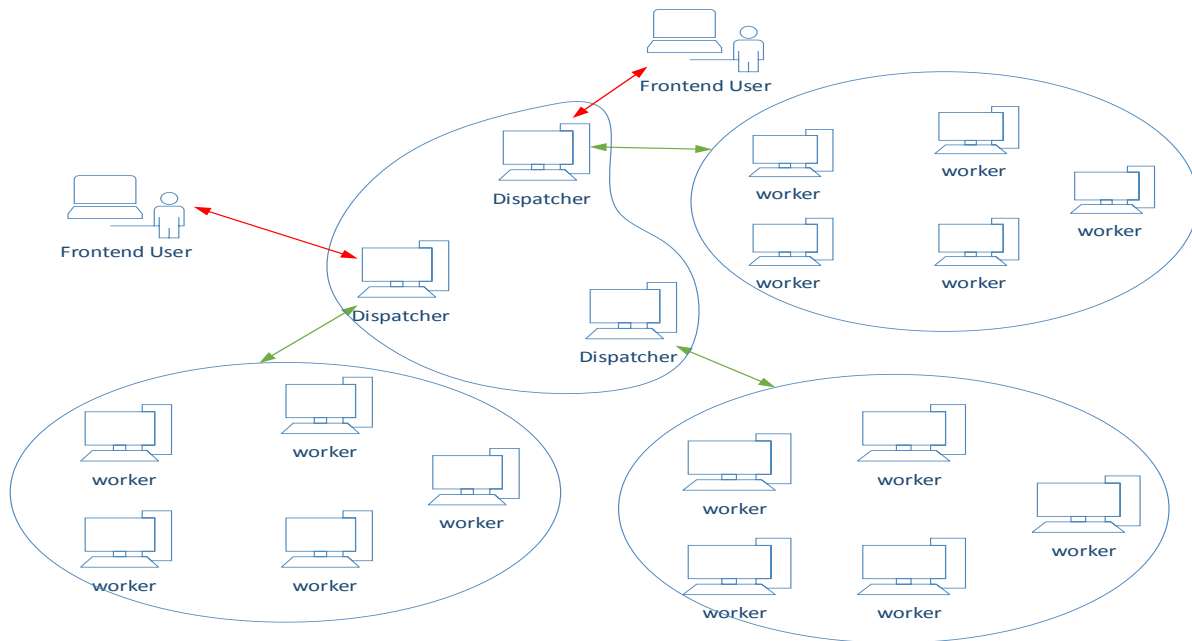


Fig. 2: An example DC Architecture

A handful of common distributed coordination problems from the literature can be adopted to develop decentralized power system applications. These include the following:

- **Consensus / Agreement:** The groups in a DC architecture require to agree on a scalar value. This mechanism of agreement in a distributed environment is called consensus. The various consensus algorithms available are Simple Consensus [19], K-set[19], Paxos [19], Raft [20], Raptor [21] and Interactive Consistency (IC) [22].
- **Election:** Processes can elect one process among them to make decisions or calculations for the group. The decision criteria can be based on processor identifier or any other application specific decision function. Bully [23] and Ring based [23] election algorithm can be possibly used for leader election.
- **Mutual Exclusion:** Ensuring that only one process is performing certain steps such as accessing a shared resource. The Ring based [23], Ricart and Agrawala [23], Maekawa's Voting Algorithm [23], Leslie Lamport's algorithm [24] and Suzuki-Kasami's Broadcast Algorithm [25] may be used for mutual exclusion.
- **ABCAS**T: multicast ($1 \rightarrow$ many) communication that delivers messages to the group in the same order and atomically (all processes get a message or none do) despite multiple concurrent senders, the order messages arrive from lower layers such as UDP, etc.
- **Voting:** processes choose one value from a set of values. This can include average functions such as mode, median, mean, and ALL-NOT-FAILED, as well as other. A possible variant here is **fusion/aggregation**, which is calculating value(s) from a set of values rather than just choosing one. An extreme example would be complex event processing (CEP).
- **Interactive Consistency (IC):** Processes agree on a vector of values (one sent by each process)
- **Group Membership:** This handles which processes are now in a given group (formed for a given purpose, perhaps only for a few minutes)
- **Group discovery/formation:** peer-to-peer mechanisms for a decentralized agent that wishes to support the grid (e.g., voltage support by a plug-in distributed energy resource) to form a group (based mainly on CPS domain topology but possibly also (for electricity)

including capabilities (battery charge) and possibly even ICT topology), e.g. to support voltage in a given (fairly small) area. These groups may need to discover what other peers are “nearby” (in domain topology terms) to be able to coordinate within peer groups or possibly a parent group. Note: there is a lot of prior work on ad hoc mobile networks that we will adapt to the needs of the transportation and electricity domains.

- **Supply Agreement:** Each prosumer provides M values (e.g., how much it can support the grid in the next M time increments). The total N prosumers in a group will lead to $[N][M]$ dimension matrix, which need to be processed by the communicating systems in the DC architecture.

IV. THE CHALLENGES IN DC ARCHITECTURE

The group messaging interface is the primary mechanism of communication for different sets of groups in a DC architecture. The problems faced by the programmer when trying to program pieces of logic that must be spread across a network, especially a wide-area network outside of a single substation, include the following:

Variable delays: The sharing of data among different groups in DC architecture may not arrive synchronously. The latency associated for data transmission varies between different groups resulting in unbounded time for the arrival of the data packets. This may result in incorrect ordering of messages arriving at each process in the group of processes that are coordinating.

Partial Failures: The communicating systems in the DC architecture may continue to transmit some incorrect measurement data due to anomalies in some of the sensors. There are also chances of omission of transmitted data at frequent intervals. Some systems may be healthy and may be sending correct data, but few may have failed and crashed. Network partition may result in systems on opposite ends of network partition to make contradictory updates. Hence DC architecture should employ failure detectors to detect these failures and should be designed to be fault tolerant.

Byzantines failures: There may be cases where in failures observed in the communicating system might be arbitrary in nature, in which a process can arbitrarily omit to send a message, or send conflicting messages to different processes, or send duplicate messages etc. These type of failures can be unintentional (due to software or hardware malfunction) or intentional (malicious) and is difficult to detect and recover from them. Hence it is necessary to have signed messages to guarantee authenticity of sender.

Race condition: The communicating systems in a DC architecture may be designed to control physical power system equipments. The inverters or the governors equipped to the generators in a particular group require droop control to operate with the other inverters or governors in the same group or the adjoining groups. Lack of proper coordination and agreement might sometimes lead to the race condition for the inverters [16] and the governors spread across different groups in the DC architecture.

Inconsistent decision: The algorithms used for consensus and election in DC architecture may sometimes result in unstable operating point for the power system network. Or there may be situations where there is no possibility to arrive at a common agreement/consensus among the processes in a group. As a result, proper fallback actions need to be implemented to avoid catastrophic errors to the ad-hoc nature of different groups in the DC architecture

V. DISTRIBUTED VOLTAGE STABILITY EXAMPLE

The various power system applications which can be scaled to DC architecture are state estimation, voltage stability, RAS, etc. In this work, voltage stability is selected as an example for implementation using DC architecture. The 30 bus power system is modeled in DETERLab [26] to test the system with different failures models. DETERLab has many hundreds of computational nodes that can be arranged in varying topologies, including special nodes that delay, drop, or corrupt. It has great potential for cyber-physical R&D [27].

This prototype is implemented by integrating voltage stability application with a decentralized coordination framework called DCBlocks (being developed as open source), which provides features like group management, leader election, consensus, ABCAST etc. The DCBlocks framework is developed using Akka Java [28] which is one of the common toolkits used for building distributed applications. Akka toolkit uses actor based model where in messages are exchanged between actors asynchronously. The actors can reside in the same system or in remote systems and interact with each other without the knowledge of their exact physical location. Akka also provides support for group (called clusters) membership and group failure monitoring. It has robust fault handling mechanism with well-defined supervision hierarchy and inbuilt resolution strategy for some of the failures like crash failures etc. There are other toolkits available like JBOSS, Spread etc which provide similar features. DCBlocks is built on top of this platform and supports additional features like leader election, consensus etc.

Consider a DC network as shown in Fig. 3, which is designed to be implemented in DETERLab. It consists of a network of substations, which can be grouped together based on the electrical distance, voltage to reactive power sensitivity and reactive power availability. The substations (hereafter referred to as nodes) have the computing devices able to compute voltage stability indices using reduced network model equivalent and synchrophasor measurements from phasor measurement units (PMU). Given the intermittency of wind energy and possibility of large change in shorter time frame, voltage stability (VS) margin needs to be monitored more frequently.

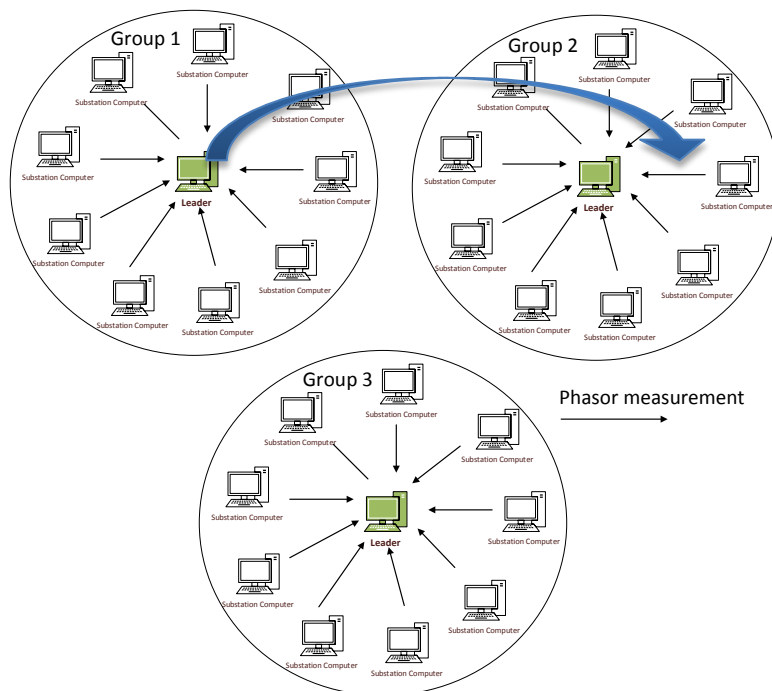


Fig. 3: A DC architecture for implementing Distributed Voltage Stability

When a voltage stability problem at a particular bus is encountered in a reduced power network, the lead computes reactive power needed and communicates all the substations within that group to provide the reactive power needed to improve the voltage stability using control actions discussed in [29]. An index 'Voltage Stability Assessment Index (VSAI)' will be computed at every load bus as:

$$VSAI(k) = \frac{\overline{VL(k)}/\overline{IL(k)}}{f(\overline{VL(k)}, \text{ReducedNetworkGroupTopology}, \text{ReactiveMargin}, \text{ZIPLoadModel}(k))}$$

where: k = Load Bus; $VL(k)$ = Voltage phasor measurement at bus ' k ';

f = function to compute system-centric network equivalent impedance as seen from bus ' k '

If the VSAI value at a bus is close to '1', it indicates that the load bus is operating close to the edge. As the VSAI is calculated for all the load buses in the system, the bus with the highest VSAI (i.e. the weakest bus) decides the VSAI of the entire system. This approach will require complete network information and PMU measurements at control center for fast monitoring of voltage stability. Lead node can also perform computation for required control actions to improve the voltage stability. For decentralized voltage stability applications, following requirements need to be met:

- The system should be able to dynamically reorganize the group based on the reactive power availability, voltage sensitivity and electrical distances.
- The election algorithms in the DC group will help in selecting a lead node having the best possible computational ability to perform state estimation and voltage stability assessment as well as control action.
- If the reactive power within the group is insufficient, the lead node should contact other lead nodes for additional reactive power (as shown in the blue arrow in the figure Fig. 3) and reorganize the group if necessary in order to acquire the required reactive power.
- This will be inherently fault tolerant and can tolerate failures up to $f = (n - 1)/2$ failures in this case. Failure types tolerated are crash, omission, value and timing failures.
- The design should employ timeout mechanism to translate crash and omission failures into timing failures using timeout. If the node does not respond within specific timeout, it is considered as failed after few trials. We will further work on developed algorithms for IEEE test topologies and compare with existing centralized voltage stability algorithm [28].

The dynamic grouping and regrouping of nodes is done using the DCBlocks framework. The leader of the group is elected using DCBlocks, which collects the vote from each substation and selects the node having the highest computational ability as the leader. Each node sends its local power measurement to the lead node which performs computation for the group. If the control action within the group is not sufficient to resolve the problem, then dynamic regrouping of the nodes (merging of two or more groups) can be done using DCBlocks. DCBlocks helps in detecting faulty nodes like for example the leader, in which case, the application can switch to secondary leader to perform leader activities.

VI. CONCLUSION

This paper highlights the recent power system monitoring and control issues faced by the CC based control centers. The several aspects of the CC based control centers, which are already experiencing problems or may soon experience limitations in the near future due to

paradigm shift in the nature of grid resources and the loads are discussed. The alternative supplementary solution to address these issues proposed in this paper is to develop distributed coordinated (DC) architecture based monitoring and control. The challenges envisaged for a DC architecture are discussed along with possible solution for power system application perspective. The voltage stability application is considered as an example for implementing using DC architecture in this paper. The requirements for implementing the distributed voltage stability in Deterlab are discussed in this paper.

BIBLIOGRAPHY

- [1]. F. F. Wu , K. Moslehi and A. Bose "Power system control centers: Past, present, and future", Proc. IEEE, vol. 93, no. 11, pp.1890 -1908 2005.
- [2]. N. Mithulananthan , C. A. Canizares , J. Reeve and G. J. Rogers "Comparison of PSS, SVC, and STATCOM controllers for damping power system oscillations", IEEE Trans. Power Syst., vol. 18, no. 2, pp.786 -792 2003.
- [3]. S. H. Horowitz A. G. Phadke "Third zone revisited", IEEE Trans. Power Del, vol. 21, no. 1, pp.23 -29 2006.
- [4]. H. Ohtsuki, A. Yokoyama, and Y. Sekine, "Reverse action of on-load tap changer in association with voltage collapse", IEEE Trans. Power Syst., vol. 6, pp.300 -306 1991
- [5]. K. R. Padiyar, Analysis of Subsynchronous Resonance in Power Systems, 1999 :Kluwer
- [6]. J. W. Stahlhut , T. J. Browne , G. T. Heydt and V. Vittal "Latency viewed as a stochastic process and its impact on wide area power system control signals", IEEE Trans. Power Syst., vol. 23, no. 1, pp.84 -91 2008
- [7]. P. M. Esfahani , M. Vrakopoulou , K. Margellos , J. Lygeros and G. Andersson "Cyber attack in a two-areapowr system: Impact identification using reachability", Proc.Amer. Control Conf., pp.962 -967 2010
- [8]. G. Hug- Glanzmann and G. Andersson "Decentralized optimal power flow control for overlapping areas in power systems", IEEE Trans. Power Syst., vol. 24, no. 1, pp.327 -336 2009.
- [9]. E. Caro , A. J. Conejo and R. Minguez "Decentralized state estimationand bad measurement identification: An efficient Lagrangian relaxation approach", IEEE Trans. Power Syst., vol. 26, no. 4, pp.2500 -2508 2011
- [10]. M. Shahraeini, M. Javidi and M. S. Ghazizadeh "Comparison between communication infrastructures of centralized and decentralized wide area measurement systems", IEEE Trans. Smart Grid, vol. 2, no. 1, pp.206 -211 2011
- [11]. M. E. Elkhatab , R. E. Shatshat and M. M. A. Salama "Decentralized reactive powercontrol for advanced distribution automation systems", IEEETrans. Smart Grid, vol. 3, no. 3, pp.1482 -1490 2012
- [12]. H. Mehrjerdi, S. Lefebvre, M. Saad and D. Asber "A decentralized control of partitioned power networks for voltage regulation and prevention against disturbance propagation", IEEE Trans. Power Syst., vol. 28, no. 2, pp.1461 -1469 2013
- [13]. J. Romero Aguero "Review of Impacts and Solutions for Integration of Renewable Intermittent Distributed Generation in Traditional and Future Power Distribution Grids", Grid of the Future Symposium, Houston, 2014
- [14]. Hao Liang, Bong Jun Choi, Weihua Zhuang and Xuemin Shen "Stability enhancement of decentralized inverter control through wireless communications in microgrids", IEEE Trans. Smart Grid, vol.4, no.1, pp.321,331 2013.
- [15]. P. H. Divshali, A. Alimardani, S. H. Hosseinian and M. Abedi, "Decentralized Cooperative Control Strategy of Microsources for Stabilizing Autonomous VSC-Based Microgrids," IEEE Trans. Power Syst., vol.27, no.4, pp.1949-1959 2012
- [16]. H. Li, J. Smith, M. Rylander "Multi-Inverter Interaction with Advanced Grid Support Functions", Grid of the Future Symposium, Houston, 2014
- [17]. M. Rylander, J. Smith, H. Li, "Determination of Smart Inverter Control Settings to Improve Distribution System Performance", Grid of the Future Symposium, Houston, 2014
- [18]. G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair. Distributed Systems: Concepts and Design, 5ed. Boston: Addison-Wesley, 2011
- [19]. L. Lamport, "Generalized consensus and Paxos," Microsoft Research, Tech. Rep. MSR-TR-2005-33, Mar. 2005.

- [20]. D. Ongaro and J. Ousterhout, “In Search of an Understandable Consensus Algorithm,” Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference, Philadelphia, PA: USENIX Association, 2014, pp. 305–320.
- [21]. I.R.O. A. Barros M. Weigang, “Computational Models, Software Engineering and Advanced Technologies in Air Transportation Raptor”, Engineering Science Reference, 2009.
- [22]. M. Pease, R. Shostak, and L. Lamport. 1980. Reaching Agreement in the Presence of Faults. J. ACM 27, 2 (April 1980), 228-234.
- [23]. J. Dollimore T. Kindberg G. Coulouris and G. Blair, “Distributed Systems: Concepts and Design”, Addison-Wesley, 2011.
- [24]. L. Lamport, “Time, Clocks, and the Ordering of Events in a Distributed System,” Commun. ACM, vol. 21, Jul. 1978, pp. 558–565
- [25]. I. Suzuki and T. Kasami, “A Distributed Mutual Exclusion Algorithm,” ACM Trans. Comput. Syst., vol. 3, Nov. 1985, pp. 344–349.
- [26]. DETER Project, <http://deter-project.org/>
- [27]. Ryan Goodfellow, Robert Braden, Terry Benzel, and David Bakken. “First Steps Toward Scientific Cyber-Security Experimentation in Wide-Area Cyber-Physical Systems”, in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, ACM, Oak Ridge, TN, January 2013.
- [28]. AKKA JAVA Documentation, available online <http://akka.io/docs/>.
- [29]. S. Biswas, “Synchronphasor Based Voltage Stability Monitoring and Control of Power Systems”, Ph.D. Thesis, Washington state University, September 2014.