Decentralized Self-Healing Scheme for Electrical-Distribution Systems

A Project Report

Submitted in partial fulfillment of requirements
for the award of the degree of

Bachelor of Technology
In
Electrical Engineering
By
Kannedi Asha
(EE13B032)

Under the guidance of
Dr. K. S. Swarup

Department of Electrical Engineering
Indian Institute of Technology, Madras
May 2017

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and indebtedness to my project guide Dr. K. S. Swarup, in Department of Electrical Engineering, for his most valuable guidance, discussions, suggestions and encouragement, from the conception to the completion of this project. His moral support, unreserved cooperation and generosity, which enabled me to complete the work successfully, will be everlasting in my memory.

I would also like to thank my professors and friends from my undergraduate studies while at Indian Institute of Technology, Madras. The preparation and experience I gained were invaluable.

Kannedi Asha

CERTIFICATE

This is to certify that the project report entitled **"Decentralized Self-Healing Scheme for Electrical Distribution Systems",** submitted by **Ms K.Asha (EE13B032),** to the **Indian Institute of Technology, Madras,** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electrical Engineering,** is a bonafide record of work carried by her under my supervision.

Dr. K. S. Swarup
Department of Electrical Engineering
Indian Institute of Technology
IIT Madras
Chennai 600036

ABSTRACT

Self healing is a major driving force in the smart grid vision. The optimum self healing strategy is planned for the system for all possible future faults and optimum microgrids are designed. A multi agent based distribution automation solution is proposed to be used in the distribution system of self healing grids to solve service restoration as a part of the Fault Location, Isolation and Service Restoration task. The solution reduces the grid topology to an undirected weighted graph and executes a distributed implementation of Kruskal's minimum spanning tree algorithm to solve the self healing problem.

An architecture composed of physical agents is presented on a dual platform of JAVA-JADE and MATLAB, Java Agent DEvelopment Framework (JADE) to simulate multi-agent based micro grids, and the Matlab/Simulink simulation platform to design and implement power generators and power loads. This project proposes an implementation procedure comprising of Kruskal's algorithm. Test is carried on a IEEE 3feeder 16 bus system and a 12 bus system.

Keywords: Smart Grid, Self Healing, Kruskal's algorithm, multi agents, Distribution System, TCP/IP Layers, Sockets, Server/Client

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Electrical energy is one of the basic need for the economic development of the country. In day today life, importance of electricity is more in the field of agriculture, industries, transportation facilities, commercial purpose etc. and supply of electricity to these sectors are taken for granted. Due to the complexity and natural factors of the electrical system, the faults occurring along the line is inherent or even greater to the system. Electrical distribution system plays a very important role in the power system network to meet consumers demand.

The inter-microgrid coordination is achieved with a heuristic control-action behavior that is able to respond to events/disturbances. In such a way, each agent may interact and negotiate with other agents to achieve a coordinated and semi autonomous behavior.

1.1 Smart grids:

A smart grid uses digital technology to improve reliability, security, and efficiency (both economic and energy) of the electrical system from large generation, through the delivery systems to electricity consumers and a growing number of distributed generation and storage resources.A Smart Grid incorporates the benefits of advanced communications and information technologies to deliver real-time information and enable the near-instantaneous balance of supply and demand on the electrical grid.

The vision of Smart Grid is to provide much better visibility to lower-voltage networks as well as to permit the involvement of consumers in the function of the power system, mostly through smart meters and Smart Homes. A Smart Grid incorporates the features of advanced ICTs to convey real-time information and facilitate the almost instantaneous stability of supply and demand on the electrical grid.

One significant difference between today’s grid and the Smart Grid is two-way exchange of information between the consumer and the grid.Whenever an outage occurs in the distribution network, there may be partial or total block out of the system.Outages occurs due to an unpredictable interruptions exist in a distribution system, it is very difficult to bring back the power system promptly to its initial state or to an optimal target network by switching operations.

Self-healing is a major driving force in the smart grid vision. The optimum self-healing strategy is planned for the system for all possible future faults and optimum microgrids are designed. A multi agent based distribution automation solution is proposed to be used in the distribution of self healing grids to solve the service restoration part of the Fault Location, Isolation and Service Restoration task.

To improve the reliability of the electrical network by renovating its infrastructure and functionalities, distribution system operators (DSOs) are used.This renovation process is driven by needs such as penetration of renewables, distributed generation, and increasing demand for electricity, which forces the electric power system to evolve from the traditional unidirectional power flow to a newer paradigm where power flows bidirectionally.

Support for constructs, such as ontologies and speech, enables the MAS architecture.distributed control attracts the attention of the industry experts as a promising distribution automation technology for higher grid reliability due to its robustness, scalability, and efficiency

Instead of the automatic transfer switching schemes that assume each substation implements a preprogrammed local power restoration plan, the more advanced distributed control approaches propose that the affected substations can cooperatively develop and enforce adapted power restoration plans in order to achieve higher reliability and better post-fault conditions in the grid. For this reason, a MAS-based distribution automation solution is studied to solve the Fault Location, Isolation and Service Restoration problem in the context of complex scenarios of self-healing grids. The solution applies the monitoring and control functionalities provided by the MAS system, and it implements an ICT infrastructure that is optimized for efficient operation.

The benefits associated with the Smart Grid include:
More efficient transmission of electricity. Quicker restoration of electricity after power disturbances. Reduced operations and management costs for utilities, and ultimately lower power costs for consumers.Reduced peak demand, which will also help lower electricity rates.Increased integration of large-scale renewable energy systems.Better integration of customer-owner power generation systems, including renewable energy systems.Improved security.

Distribution restoration is an important task to restore loads after a fault by altering the topological structure of the distribution network while meeting electrical and operational constraints. Such a network reconfiguration process is achieved by changing open/closed states of a number of tie-switches and sectionalizing switches in the distribution system. An efficient distribution system restoration strategy to quickly determine and optimize restoration plans can significantly reduce the overall customer outage time, thereby enhancing customer satisfaction and quality of the power supplying service.

The indexes that permit evaluating power system reliability are: SAIFI (System average interruption frequency index), SAIDI (System average interruption duration index), CAIFI (Customer average interruption frequency index), and CAIDI (Customer average interruption duration index).The migration of automation intelligence in MAS-based control from the center to the grid periphery helps improve the indexes by shortening system response times as it avoids the scenario where a central control system spends much time on extensive requesting and collection of measurement and state information from different substations before the power supply restoration begins.

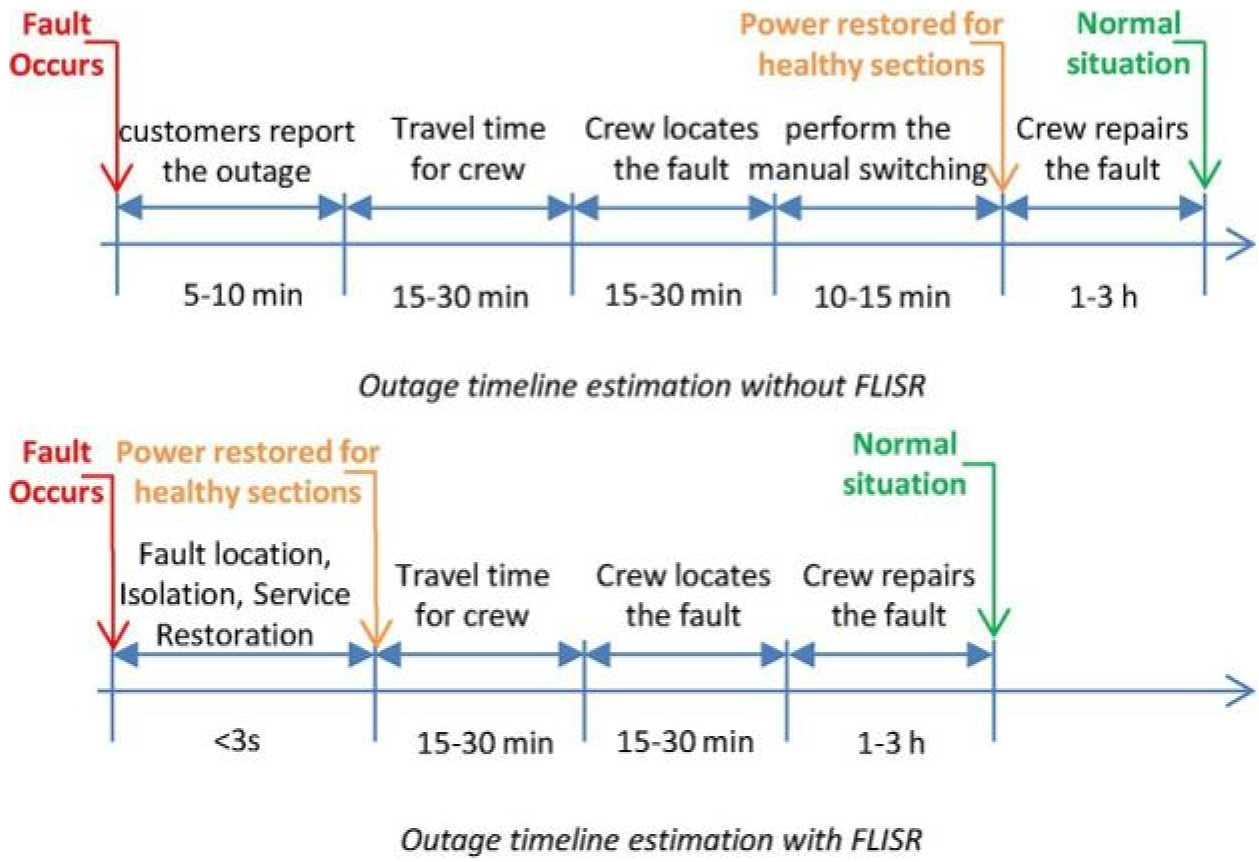


Figure 1.1 Outage timelines with and without FLISR

MAS-based control strategy suggests using the computing capacity of the substations to quickly analyze the situation. Only succinct data containing analysis results and action requests have to be sent over the communication network the substation equipment does not need to interact in real time with the control centers to get orders on how to act when the situation changes. MAS-based control is more tolerant to the communication network parameters.The implementation of decentralized control in distribution automation improves the scalability of the distribution grids because the agent-based control approach requires only that the agents maintain consistent and up-to-date information about their local area and not the entire grid.

An important phase in the FLISR process is the location of the fault, which can be done by applying different techniques such as apparent impedance measurement, direct three-phase circuit analysis, superimposed components, traveling waves, power quality monitoring data, and artificial intelligence.In the FLISR process, service restoration is a key element, and it can be generalized as a multi objective and multi constraint optimization problem. There are multiple available techniques in the literature to solve the service restoration part, which can be implemented following either a centralized or a decentralized approach.Significant effort is being made to implement FLISR following a decentralized architecture because due to single decision making software in the centralized distribution automation architectures.

We propose an efficient MAS-based FLISR solution that does not select a particular type of line and is applicable to different kinds of grid topologies, including those with bidirectional energy flows. Thus, a composition of the MAS with a minimum number of agents and agent types with clear roles is proposed. Grid topologies consisting of circuit breakers is considered but this does not affect the performance of system restoration part of the algorithm because the switching is carried out in the isolation and reconfiguration phases.

New trends in the electricity sector are demanding distribution system operators (DSOs) to improve the reliability of the electrical network by renovating its infrastructure and functionalities. This renovation process is driven by needs such as penetration of renewables, distributed generation, and increasing demand for electricity, which forces the electric power system to evolve from the traditional unidirectional power flow to a newer paradigm where power flows bidirectionally.

Self Healing is a term used to describe automated fault location, isolation and service restoration. It includes the identification of a fault, the event localization, the isolating of the faulty section and reconfiguration of the grid to re-energize faulty sections without needing manual intervention.The self healing requires communication between the intelligent controllers. There are several choices foi communication architecture. A common one is to configure one node as a local master which communicates with all the others. This means the “leaf” nodes are fairly simple but the local controller has to handle many communication channels The system described in this paper uses a communication topology that mirrors the electrical network topology. Each unit communicates only directly with its neighbours. This makes it easier to reconfigure the system, if and when more controllers are added.

The system uses write requests that are only sent when values change. This mechanism is equivalent to using DataSets in other protocols. One important safety/security feature is that none of the units send direct requests to open or close switches remotely. Instead each message only reports the state of the sender. The receiving unit always uses its own status measurements in combination with the information in the received messages in order to make a decision whether to operate a switch.

The uptake of multi-agent systems has increased over the last few years, in terms of number of research projects. However, it is essential at this stage of maturity of research into the application of MAS that appropriate standards and guidance are available for those developing multi-agent systems in the power engineering community.

1.2 Summary

Whenever an outage occurs in the distribution network, there may be partial or total block out of the system.Outages occurs due to an unpredictable interruptions exist in a distribution system, it is very difficult to bring back the power system promptly to its initial state or to an optimal target network by switching operations.Thus an intelligent agent based MAS topology is carried out for quick power restorations compared to centralized schemes.

CHAPTER 2

SELF HEALING IN SMART GRIDS

The multi agent based solution reduces the grid topology to an undirected weighted graph and executes a distributed implementation of Kruskal’s or prim’s minimum spanning tree algorithm to solve the problem. The solution is compliant with state-ofthe-art standards within smart grids, including but not limited to IEC61850. To test the performance of the algorithm, a testbed is assembled consisting of a physical dc grid model and several Arduino microcontrollers and Raspberry Pi computers. The test results show that the proposed algorithm can handle complex FLISR scenarios.three types of agents: substation control agent (SCA), load control agent (LCA), and restoration agent (RA). All these three types of agents perform protection and control functions that can be mapped to the IEC 61850 standard. The agents can be implemented on an IEC 61850- compatible IEDs.

LVrni be the load of a regular node Vrni. The load of feeder node Vnf is then

$$LVnf=allVnri \in VnrLVnri + losses.$$

To ensure that the constraint is fulfilled, the weight of adding one vertex Vdi to a tree Tj is set to be $wij =(LVjf + LVdi) / Ljmax$

and a new connection will be possible only if $wij \leq 1$.

These are the constraints considered through this reference to solve for prim’s algorithm. Similar is the case if we use Kruskal’s.

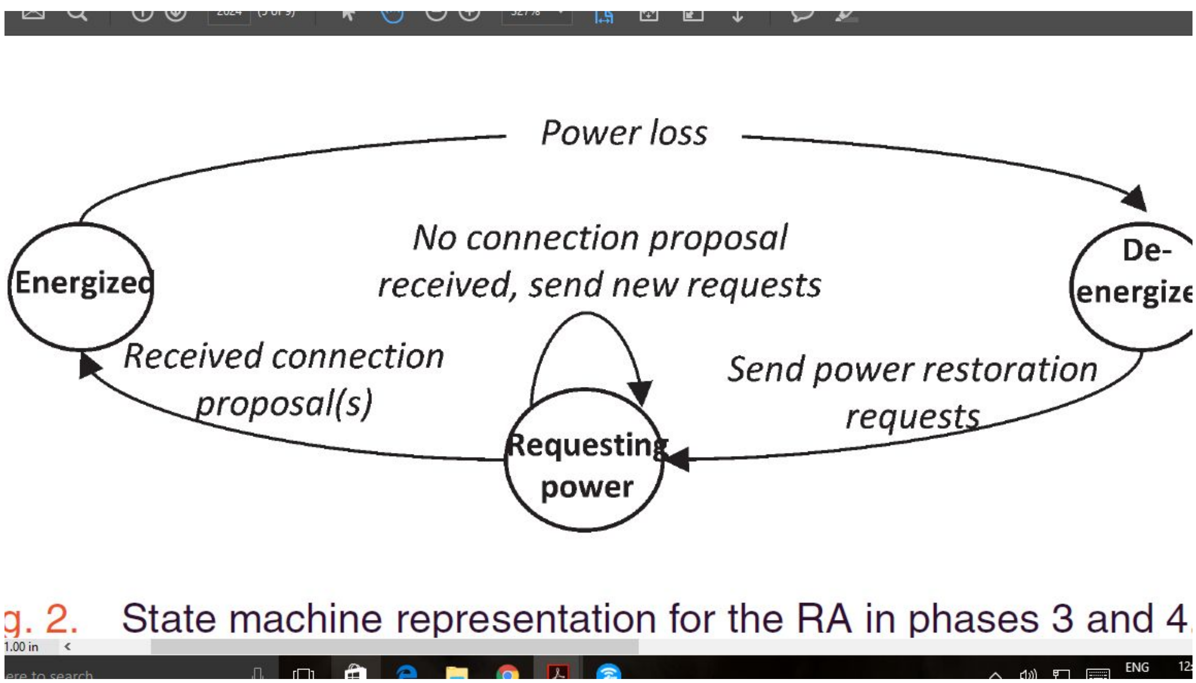


Figure 2.1 State representation of Restoration Agents

A multiagent-based distribution automation solution has been developed to restore the power supply in case of outages in a distribution network. The FLISR solution is based on Prim's algorithm, and a parallel ICT architecture is created that is optimized to support the implementation of MAS in smart grids.[1] The tests have been carried out to show that the proposed algorithm can perform the FLISR process as quickly as within several seconds of computation time and that the system is stable and can handle complex scenarios such as multiple simultaneous faults.

The power networks with multiple microgrids require flexibility and versatility in their coordination tools and decision-making tools from both technical and economical points of view.An architecture composed of physical agents is presented on a dual platform of JAVA-JADE (environment for developing agents) and MATLAB. The resilience of multiple microgrids is then demonstrated in relation to three types of disturbed operations: (i) highly variable net load, (ii) net load ramp events and (iii) net load changes during high load levels.This paper also states that each agent exhibits a certain level of autonomy over the actions that it takes and each agent has a minimally partial representation of the environment.

In this control approach, each microgrid agent implements its own model predictive control (MPC), as an economic dispatch method, which is able to dispatch the mechanical power setpoints Pmi for the synchronous machines within its control area.

With the proposed approach, each agent can be implemented with increasingly complex decision-making functionality which may be entirely decentralized & autonomous. Alternatively, each agent may interact and negotiate with other agents to achieve a coordinated and semi autonomous behavior. In such a way, each microgrid has a coordination behavior that is able to respond to disturbances in the power system. Here, the JADE platform respects that each microgrid and potentially each microsource can be managed by entirely different organizations. Furthermore, the computational platform is multi-threaded allowing simultaneous decision-making capabilities that occur in geographically distributed locations.

While the power industry has tremendous experience in operating a single large power system, they have comparatively little experience in the operation of that same grid with multiple microgrids together with potentially different controlling entities. For this reason, the MAS control architecture and approach presented here provides a simulation platform with the required flexibility and versatility to address decision-making from technical and economic points of view.

The case for using a systems approach to analyse the requirements and behaviours of the Smart Grid as well as designing relevant solutions. By linking systems thinking to agent-based modelling, they discussed how a Smart Grid can be modelled as Multi-Agent Systems by reviewing some related state-of-the-art research. This paper goes on to outline two research areas that we are developing, namely, demand response using dynamic pricing and emergent behaviours of a Smart Grid.

Agent-based designs for Smart Grids have been implemented [2] with Ghosn et al. simulating a self-healing network using JADE by the use of six agent types such as

1) Device Agents
2) Distributed Energy Resource (DER) Agents
3) Consumer Agents
4) Intelligent Prevention Control Agents
5) Intelligent Response Control Agents
6) GUI Agents

Complex problems have been addressed using this framework in a variety of fields however there has been no explicit application to Smart Grids. As discussed, the use of agent based models is suited to both analyses of the system as well as a tool to create practical solutions.The results determined the market conditions required and behaviour expectancy of households to produce effective demand management.

A two-stage procedure is proposed in order to solve the centralized self-healing scheme for electrical distribution systems. The considered self-healing actions are the reconfiguration of the distribution grid and, if needed, node and zone load-shedding.The first stage solves a mixed integer linear programming (MILP) problem in order to obtain the binary decision variables for the self-healing scheme . In the second stage, a nonlinear programming (NLP) problem is solved in order to adjust the steady-state operating point of the topology found in the first stage.

The multiobjective problem of maximizing the amount of load to be restored while minimizing the number of switching operations was considered using a heuristic algorithm.The objective function in is assembled in five parts:

1) minimizes the cost of the de-energized zones;
2) minimizes the cost of the load-shedding in the nodes;
3) minimizes the cost of the active power losses;
4) and 5) minimize the cost of the total switch operations after reconfiguring the system.

A 44 node test system was considered.

A permanent fault was simulated at zone 9 (zf = 9), which was energized by the switch located between zone 7 and 9. The proposed methodology isolated the faulty zone 9 by opening the switches between zones 7 and 9, 9 and 11, and 9 and 12. The proposed methodology transferred the unattended demand from the red feeder to the blue feeder by closing the switches between zones 11 and 12, and 22 and 52. The solution suggested a load-shedding of 21.72% at node 107 in zone 11.

The proposed methodology finds feasible and high-quality solutions for establishing the system’s operation. The solution isolates the faulty zone and minimizes the number of de-energized nodes and zones by transferring load among feeders, maintaining the network operational constraints within their limits, and disconnecting non priority loads through load and zone shedding, if necessary.the methodology isolated the faulty zones and proposed a feasible system topology to minimize the de-energized load and operate under healthy conditions. The load-shedding decreased the unattended zones and improved the flexibility of the self-healing scheme providing an efficient response to the fault. The results show that the proposed self-healing methodology is efficient and robust during the restoration of electrical distribution systems in the context of smart grids.

A graph-theoretic DSR strategy incorporating microgrids that maximizes the restored load and minimizes the number of switching operations. Spanning tree search algorithms are applied to find the candidate restoration strategies by modeling microgrids as virtual feeders [5] and representing the distribution system as a spanning tree. Unbalanced three-phase power flow is performed to ensure that the proposed system topology satisfies all operational constraints. Simulation results based on a modified IEEE 37-node system and a 1069-node distribution system demonstrate the effectiveness of the proposed approach.In this paper a graph-theoretic search algorithm is proposed to identify a post-outage distribution system topology that will achieve a minimum number of switching operations and take full advantage of the resources of microgrids.

Constraints showing that

a) the line current and bus voltage should be maintained within acceptable operating limits.

b) requires that the total load of each feeder should not exceed the maximum capacity of the supplier transformer.

c) the total load in each microgrid should not exceed the total generation capability limit of DERs are considered.

Each microgrid is modeled as a virtual feeder.

The algorithms that are used to calculate unbalanced three-phase power flow on the distribution level include Gauss-Seidel (GS) and Newton Raphson (NR) methods.

The simulation results demonstrate that microgrids enhance the recovery capability of a distribution system.this research can be enhanced by considering the priority of critical loads in microgrids during system restoration, as well as modeling the microgrid as a node in the system during restoration to allow loop topology within the microgrid.

The potential value of MAS technology to the power industry is examined. It describes fundamental concepts and approaches within the field of multiagent systems that are appropriate to power engineering applications. It also defines the technical issues which must be addressed in order to accelerate and facilitate the uptake of the technology within the power and energy sector. This paper explores the decisions inherent in engineering multi-agent systems for applications in the power and energy sector and offers guidance and recommendations on how MAS can be designed and implemented.This paper describes the various options available and makes recommendations on best practice. It also describes the problem of interoperability between different multi-agent systems and proposes how this may be tackled.

The uptake of multi-agent systems has increased over the last few years, in terms of number of research projects. However, it is essential at this stage of maturity of research into the application of MAS that appropriate standards and guidance are available for those developing multi-agent systems in the power. This paper mainly focuses on the agent communication. An agent may alter the environment by taking some action: either physically (such as closing a normally-open point to reconfigure a network).The way in which MAS provide flexibility, extensibility, and fault tolerance needs to be understood.

With agent programming, external agents can only send messages requesting the agent take some action: the autonomous agent can decide whether to fulfill the request, the priority of the task, and if other actions should also be scheduled. This can be useful in situations when an agent is receiving many requests and cannot fulfill them all within a reasonable timescale, such as with multiple requests for a processing intensive task like a load-flow calculation.

The FIPA Agent Management Reference Model covers the “framework within which FIPA agents exist,” defining standards for creating, locating, removing, and communicating with agents.One requirement of an open agent architecture is that the platform places no restrictions on the creation and messaging of agents, while a second is that some mechanism must be available for locating particular agents or agents offering particular services within the platform.

Four categories of applications were discovered: monitoring and diagnostics, distributed control, modeling and simulation, and protection.MAS design methodologies tend to share some common features: a conceptualization phase where the problem to be solved is specified; an analysis phase; and a design phase that uses the results of the analysis phase to produce agent designs of varying detail.agents developed for the JADE platform tend to consist of three basic layers: a message handling layer; a behavioral layer; and functional layer

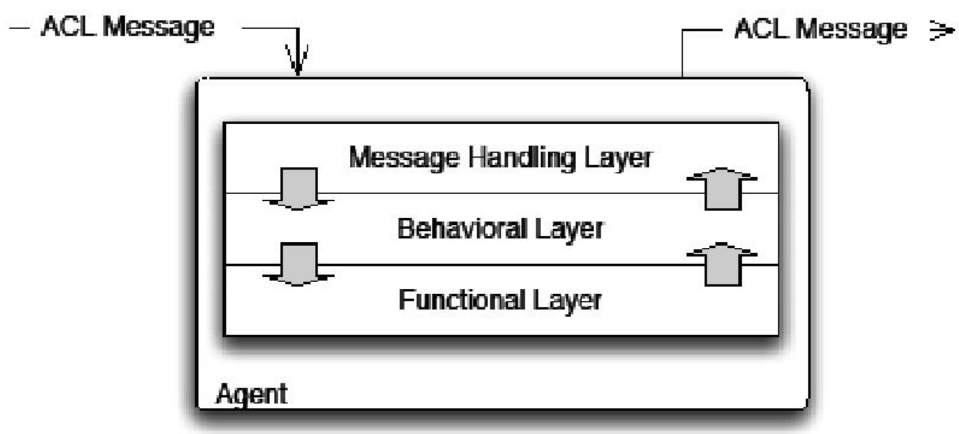


Figure 2.2 Layered architecture of JADE agents

.

JADE's support of FIPA standards and the robustness of the agents that can be implemented make it attractive, JADE also promotes a certain style of agent implementation which may not be optimal for exploiting autonomy.

Details on a MAS that restores a power system after a fault. The development of agents and behaviors of the agents are described, including communication of agents. The MAS is tested on two test systems and facilitates both full and partial restoration, including load prioritization and shedding.

An agent is executed as a single thread; however, the multithreading nature of JAVA language helps if an agent needs to execute several tasks in parallel. A power system test case was constructed in VTB, the detection of fault and restoration using a MATLAB agent environment has been described.

A message with performative "PROPOSE" is received and Conversation ID "ATCREPLY." The agent sends a close/open signal if the content is accept/decline, but normally open switch closes after some time delay to ensure load shedding in order to avoid overloading of the developed multi-agent system for service restoration of distribution network, which includes main package restoration, and with three kinds of agent whose classes are extended from the Agent class of JADE. The SimpleBehavior class of JADE was extended to realize the behaviors of different agents.

The development and implementation of a multi-agent system for load restoration has been discussed.This discussion has included description of the types of agents and their behaviors to exchange information and determine a feasible restoration path. The implementation of the agents has been demonstrated using three different test cases. The MAS features were demonstrated for full restoration, partial restoration with load shedding, and alternative path restoration. The results indicate the MAS can provide a distributed solution for restoration strategies.

In this research work the JADE implementation technique has used to isolate the fault line and has restored the nearest generator. The system model has developed in MATLAB and tested in JADE. It provides quick restoration without power loss will lead the power consumption in distributed automation power system.

The speed of the investigation of the information for discovery process is very important for reloading the power. If the average total net power has positive value, then restored part or all of the unfaulted can be possible.Identify the restore parameters as soon as possible to initiate the restoration process.A connection between an agent system in JADE and a Matlab Simulink model has been setup, which allows measurement data to be transferred from Simulink and control commands to be sent from JADE to Matlab.A connection between an agent system in JADE and a Matlab Simulink model has been setup, which allows measurement data to be transferred from Simulink and control commands to be sent from JADE to Matlab.

CHAPTER 3

PROBLEM FORMULATION OF SELF HEALING IN SMART GRIDS

Whenever an outage occurs in the distribution network, there may be partial or total block out of the system. In order to reduce interruptions to the consumers, proper switching of power lines is required and restoration of power as quickly as possible is essential. To reconfigure and determine optimal target network for minimization of the power losses, maximization of the load balance and for restoration of the power, is difficult task. The optimal target network is found through MST-Kruskal's (Minimum Spanning Tree) algorithm.

In a graph theory, T is a connected, undirected spanning tree of the graph G. That spanning tree is also a tree of the graph G(V,E). It consists of some of the lines and all the vertices of the graph G. That it doesn't forms any cycle or loop and every node lies in the tree. For the connected graph having V number of nodes, then V-1 edges be there in spanning tree. For a connected, undirected graph spanning tree is sub graph of the main graph and contains all vertices without forming any loops. If the weight is assigned to each edge, each individual weight of the edges in a spanning tree decides its weight. For a particular spanning tree the total sum of weights of each line is its weight. Minimum spanning tree or minimum weight spanning tree for a given undirected connected graph, is a spanning tree having a weight lesser or equal to other spanning tree in that graph. MST algorithms are further classified into two types they are line based and node based algorithm. One of the algorithm (line based) that is proposed by MST approach is kruskal's algorithm in a graph theory, which finds a minimum spanning tree for a connected graph with weights. This algorithm forms a tree by determining the subset of the edges that includes every vertex, and the total weight of all the edges in the tree is minimized. If the given graph is not connected, then it finds a minimum spanning tree for each connected component.

To finds the MST by taking the smallest weight in the graph and connecting the two nodes and repeating until all nodes are connected to form a tree. This process starts with the line having minimum weight. This is done by assuming line impedance as its weights. Minimum spanning tree obtained by this algorithm contains every node of the algorithm, and doesn't forms any cycle or loop. MST obtained by this algorithm, the total weight of all the lines in that tree is minimized.

The Service Restoration step in FLISR can be formulated with
Constraints as

$I_{min} < I < I_{max}$

V min < V < V max

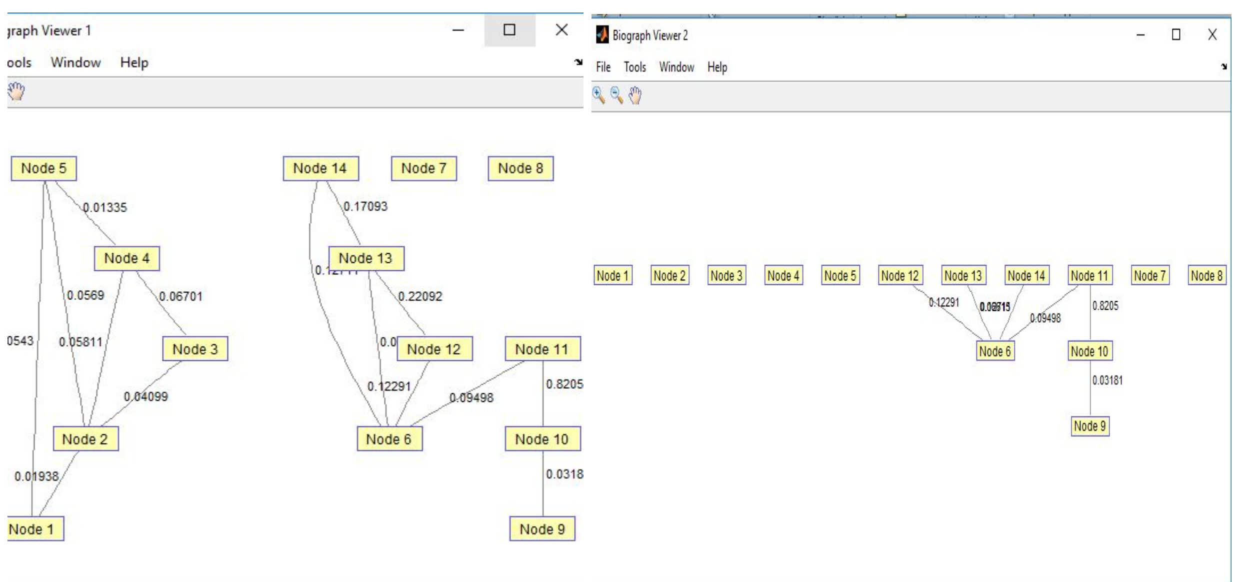
P^2 + Q^2 <= S^2

P < P max, Q < Q max

Where S is total MVA power of load and P,Q are active and reactive powers.

The solution to the FLISR task is presented following a decentralized and distributed approach based on a MAS consisting of three types of agents: substation control agent (SCA), load control agent (LCA), and restoration agent (RA).The proposed solution uses a distributed implementation of Kruskal's minimum spanning tree algorithm to solve the Service Restoration part of the FLISR task.To enable the MAS to use Kruskal's algorithm, a graph representation of the grid topology is used where the graph's edges are mapped to circuit breakers, and the graph's nodes are mapped to the areas enclosed by two or more circuit breakers as in lines and buses. The power grid capacity constraints are represented by the graph's edge weights.

This reduction is done by the use of spanning tree algorithm.



a) Before MST b) After MST

Figure 3.1 Spanning tree representation of distribution network

The Graph theoretic DSR problem consists of two major tasks finding the final optimal network topology and providing a sequence of switching operations leading to this final network topology.

The solution reduces the grid topology to an undirected weighted graph and executes a distributed implementation of Kruskal's minimum spanning tree algorithm to solve the problem as discussed in the next chapter.

A FSM (Finite State Machine) behavior of the Feeder Agent can be discussed with five states as the following

State 1:

FA receives BA message and restores prefault power values.

State 2:

Behaviour Contract Net Initiator() is invoked

State 3:

If FA is possible to restore part of the required power

State 4:

FA requests each BA to either connect/disconnect load accordingly

State 5:

FA requests one of its BAs closest to switch to connect it.

The agents such as Substation Control Agents, Restoration Agents and Load Control Agents are used for communication amongst themselves with their properties as,

SCA: It is responsible for controlling one substation equipment. It informs affected restoration agents of events and status changes in the grid, and if a fault occurs, the SCAs cooperate to locate it by comparing the measurements. If an agent (or grid operator) wants to change the state of a device (e.g., close a circuit breaker), it sends a request to the specific SCA that controls the targeted equipment.

RA: It is responsible for one node $V_i \in V$, in the grid topology graph, and its task is to ensure that the node is energized. A node is energized if it belongs to a tree T_n ($n = 1, 2, \ldots, N$) and has a power supplier, i.e., if it is either a feeder node, $V_i \in V_f$ or if it has been granted power access through another energized node.

LCA: It is responsible for one tree, T_n ($n = 1, 2, \ldots, N$), in the grid topology graph. It monitors the tree's feeder, V_{nf} , to ensure that its used capacity does not exceed its limits, $L_{V_{nf}} \leq L_{nmax}$

CHAPTER 4

METHODOLOGY OF DECENTRALIZED SELF HEALING

The buses and feeders are considered as nodes, distribution network impedance is considered as weight and distribution network line is considered as edges or lines. The total weights of all the edges in a graph are arranged in the ascending order. By these considerations the flowchart for kruskal's algorithm is as shown in fig.4.1

Reconfiguration of distribution network using kruskal's algorithm is implemented using MATLAB software. In this work, considering the IEEE standard 3 feeder 16 bus, and 1 feeder 33 bus systems. N-R method is used to perform load flow analysis for these systems.

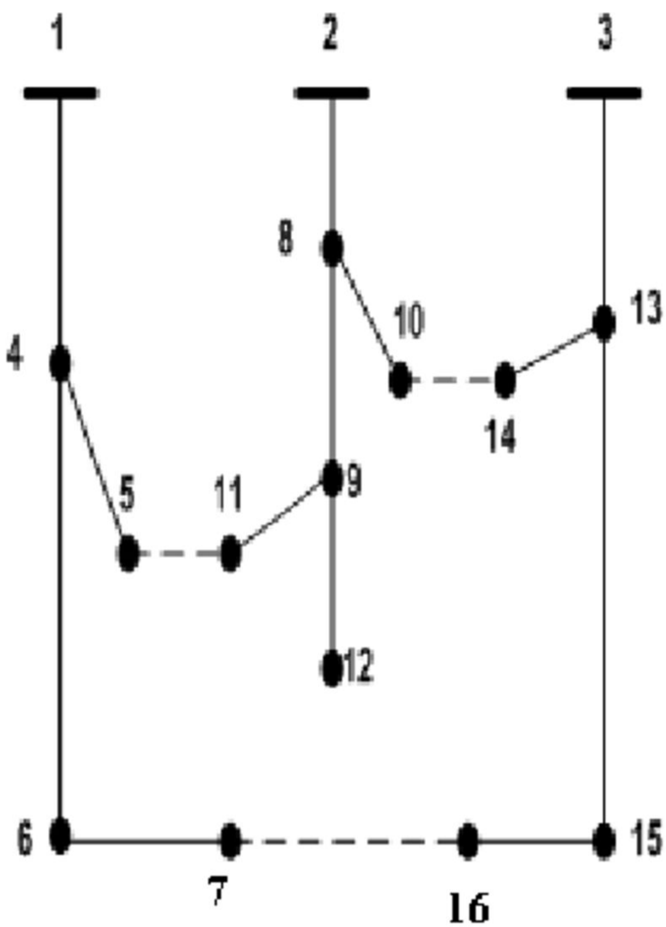


Figure 4.1 16 bus distribution system

The Newton-Raphson procedure is as follows:

Step-1: Choose the initial values of the voltage magnitudes $|V|^{(0)}$ of all n_p load buses and $n - 1$ angles $\delta^{(0)}$ of the voltages of all the buses except the slack bus.

Step-2: Use the estimated $|V|^{(0)}$ and $\delta^{(0)}$ to calculate a total $n - 1$ number of injected real power $P_{calc}^{(0)}$ and equal number of real power mismatch $\Delta P^{(0)}$.

Step-3: Use the estimated $|V|^{(0)}$ and $\delta^{(0)}$ to calculate a total n_p number of injected reactive power $Q_{calc}^{(0)}$ and equal number of reactive power mismatch $\Delta Q^{(0)}$.

Step-3: Use the estimated $|V|^{(0)}$ and $\delta^{(0)}$ to formulate the Jacobian matrix $J^{(0)}$.

Step-4: Solve for $\delta^{(0)}$ and $\Delta |V|^{(0)} \div |V|^{(0)}$.

Step-5 : Obtain the updates from

$$\delta^{(1)} = \delta^{(0)} + \Delta\delta^{(0)} \qquad |V|^{(1)} = |V|^{(0)}\left[1 + \frac{\Delta|V|^{(0)}}{|V|^{(0)}}\right]$$

Step-6: Check if all the mismatches are below a small number. Terminate the process if yes. Otherwise go back to step-1 to start the next iteration with the updates

Flowchart for Kruskal's algorithm

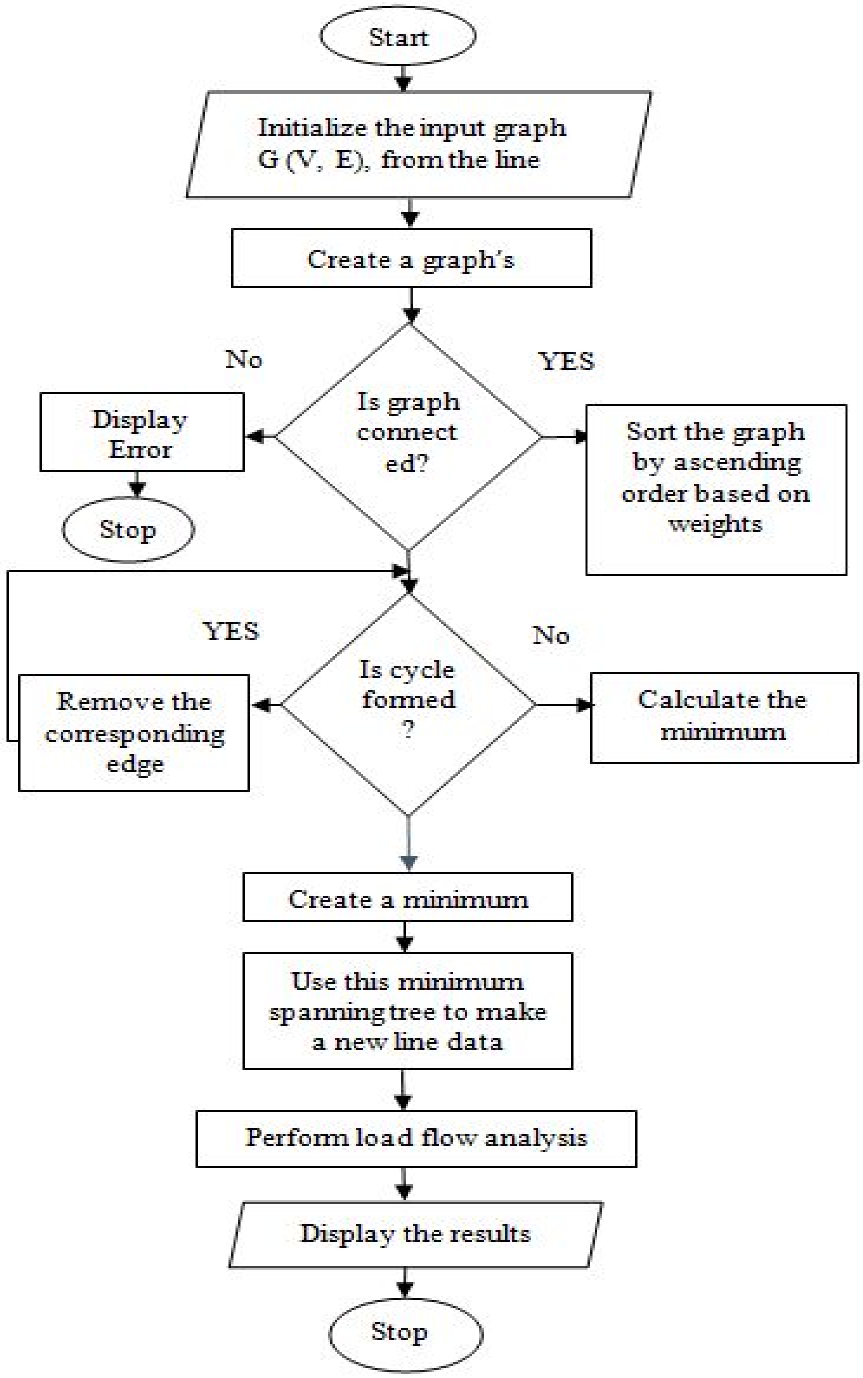


Figure 4.2 Flowchart showing kruskal's algorithm

JADE and MATLAB., These two independent platforms are connected using socket programming technique.

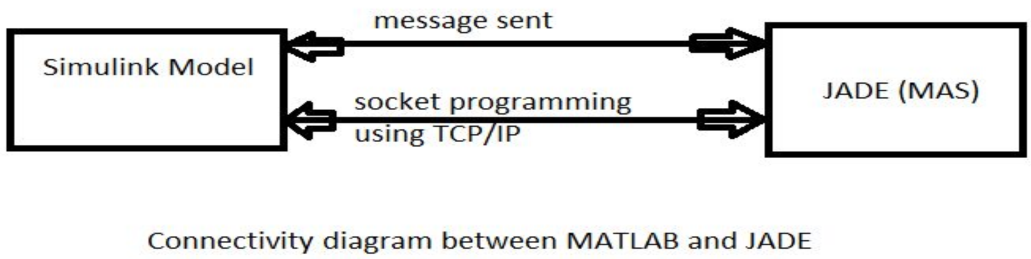


Figure 4.3 Connectivity diagram between MATLAB and JADE

An agent is a computer system that acts in an environment. It has the capability to making self-directed action in the environment in order to meet its goals. When these agents are comprise one or two are called Multi Agent System. These agents can interact with each other to find the desired goal.The Java Agent Development Environment (JADE) toolkit have three parts , library of classes assisting for agent development, a runtime environment with FIPA-specified for agent management services and a set of graphical tools for monitoring and debugging purposes.

In MAS the Agents can interact with each other to find the result/goal, they share common information and may achieve the goal with other Agents. Collaboration is about the allocation of tasks and resources between multiple agents, whether decentralized or centralized technologies. Designing an agent-based system is not just about designing the agents, it is also about designing the agent environment and interaction. If the data is particularly complex, novel, or risky, are the people, decision makers, supported by smart software support systems.

Agent:

An agent is merely “a software(or hardware) entity that is situated in some environment and is able to autonomously react to changes in that environment.” The environment is simply everything external to the agent, may be physical (e.g., the power system).An agent may alter the environment by taking some action: either physically (such as closing a normally-open point to reconfigure a network).Placing copies of the same agent in different environments will not affect the reasoning abilities of each agent nor the goals it was designed to achieve.Autonomy says that an agent exercises control over its own actions,....i.e., it can initiate or schedule certain actions for execution.

Autonomy is therefore the ability to schedule action based on environmental observations.The notion of an agent is meant to be a tool for analyzing systems.

Socket:

Agent communication is done through socket programming(TCP/IP layer). A socket is one end-point which makes communication between two programs which are running on the same machine or over network. Sockets are used to represent the connectivity between client and server. Socket is bound to an IP address and port number so that the TCP layer can identify the application.

Normally, a server runs on a specific computer and has socket which is bound to a specific port number. The server waits from client side for listening to the socket and makes a connection request. On the client end the client knows the hostname or IP address of the server and the port number of the server listening. Making a connection request the client program tries to negotiate with the server program on the hostname/IP address and port number. When connection is established between server and client, Client used that socket to communicate with server (read/write).

A messaging protocol is developed of two types as Server message and Client message. Client actually initiates the process with the server process. Server process will bind to particular port number and wait for incoming client connections.

Socket is the endpoint making communication between 2 programs and here they are client and server. They have particular ip address and port numbers. The negotiation can be shown in the form of following figure.

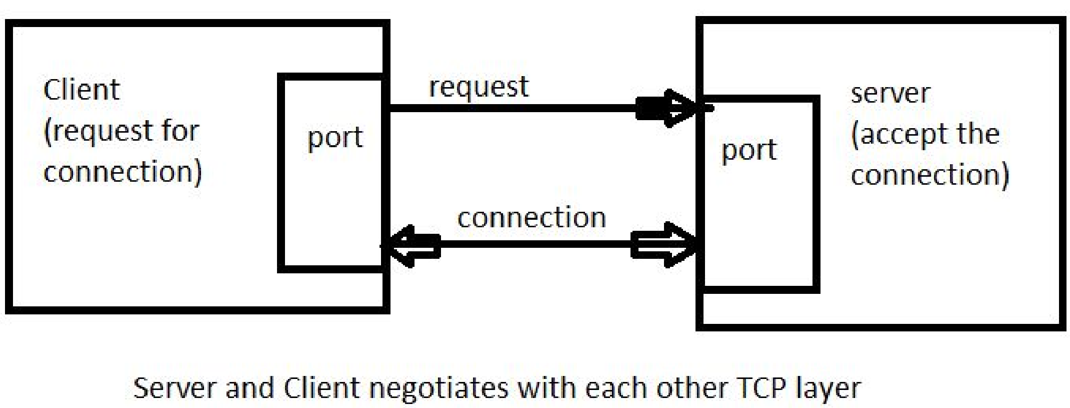


Figure 4.4 Diagram showing server and client negotiation

Chapter 5: Implementation of decentralized self healing for smart grids

The outline for Kruskal's algorithm is shown as below

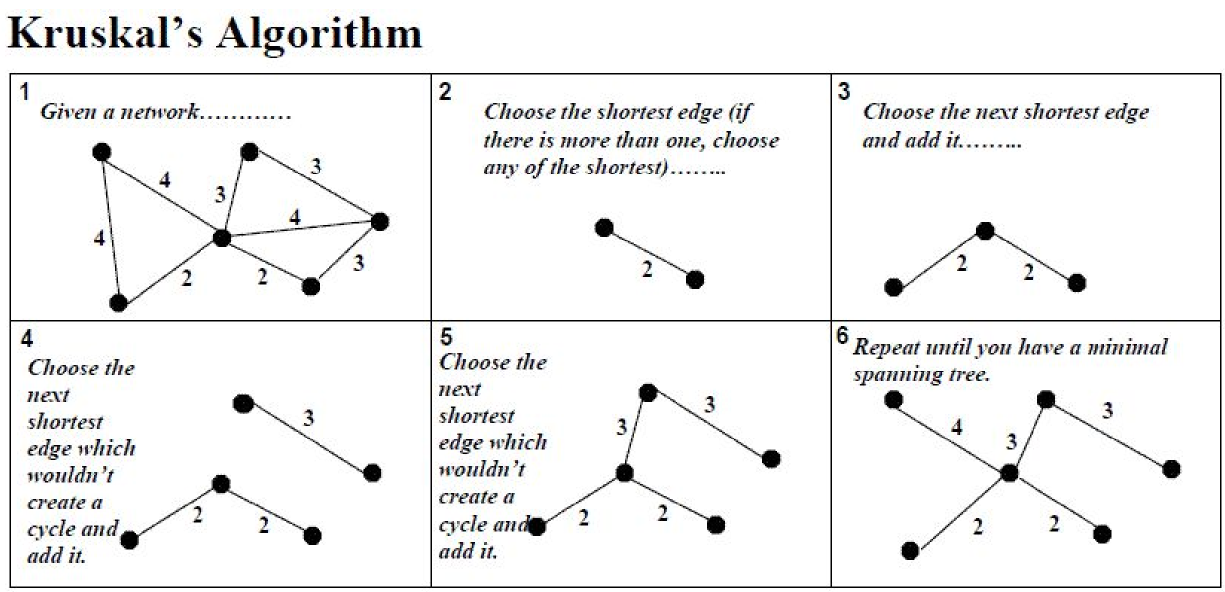


Figure 5.1 Outline of Kruskal's algorithm for a network

Socket endpoints are implemented in Java in a class called java.net package. JADE use these command (open-accept-read-write-close) for server\client socket

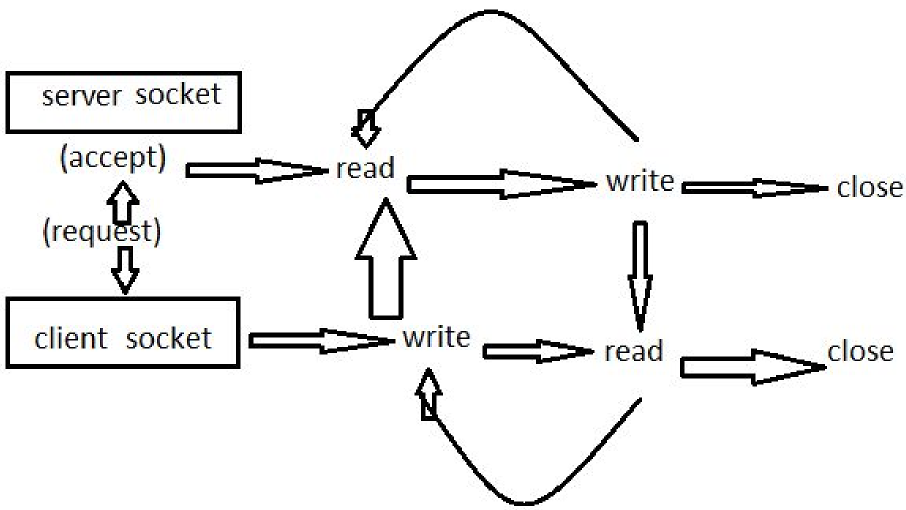


Figure 5.2 Commands followed in socket endpoints

The operations performed by socket are Open, accept connection, send data, receive data, close connection.

Algorithm to implement socket :

Step 1: Find the IP/local host and port Number of JADE Server
Step 2: Create a TCP Socket
Step 3: Connect to the Socket JADE Server (Server must be up and listening)
Step 4: Send the String to JADE Server through Socket
Step 5: Receive the String from JADE Server via Socket
Step 6: Do define process on after receiving the data
Step 7: Repeat the step 4 until all jobs done
Step 8: close the connection

A concept of a power grid composed of multiple interacting microgrids is very much a physical power grid architecture of the future.The MAS makes decentralized but coordinated decisions under disturbances.MATLAB executes a time domain simulation of the power grid transients. The power system state variables are sent back to the MAS via the facilitator agent interface.

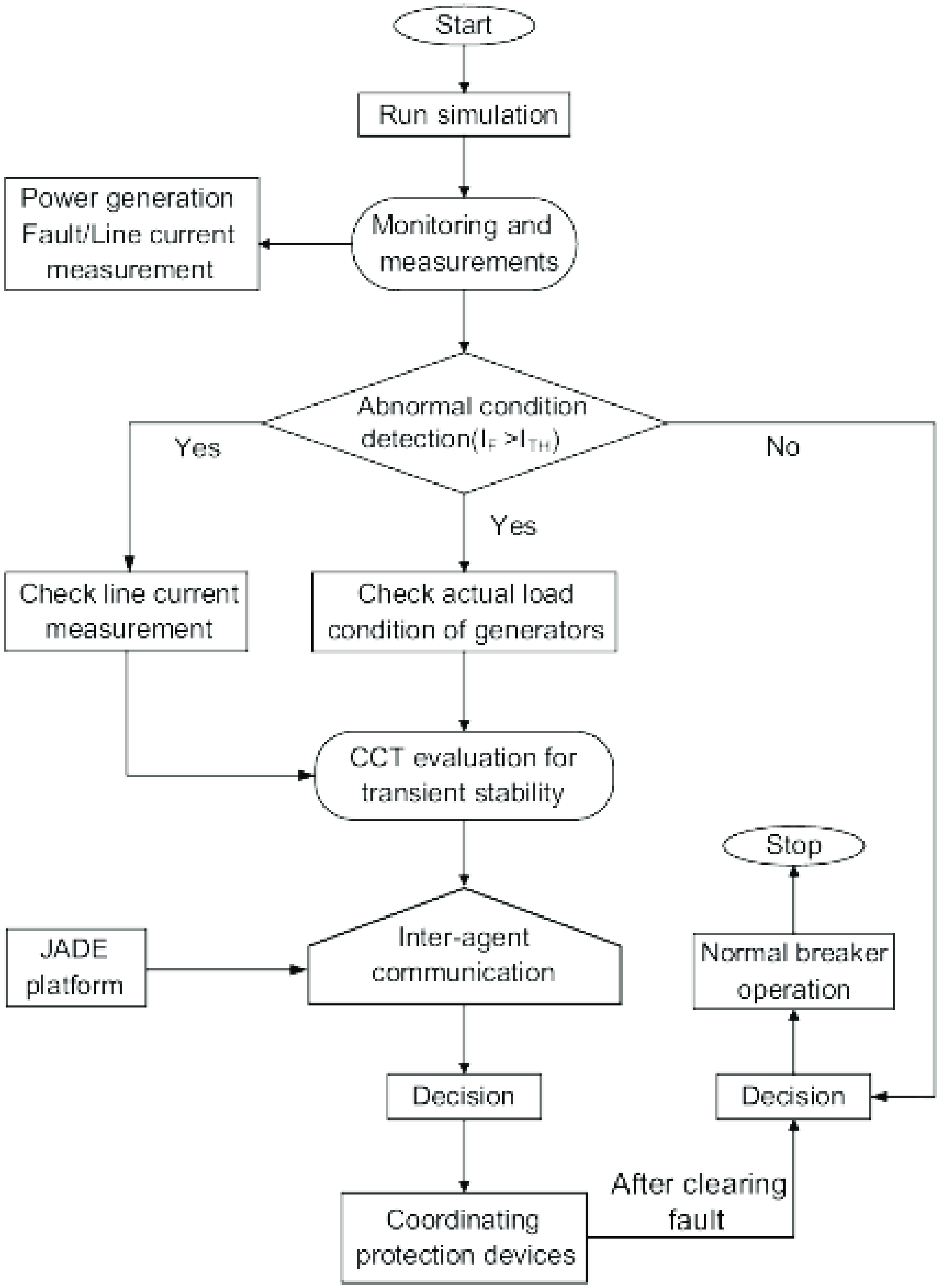


Figure 5.3 Flowchart of the implementation of proposed approach

The proposed multi-agent system coordination approach is built upon a hybrid platform in which a physical layer implemented in MATLAB is controlled by a coordination layer implemented within the Java Agent DEvelopment framework (JADE)

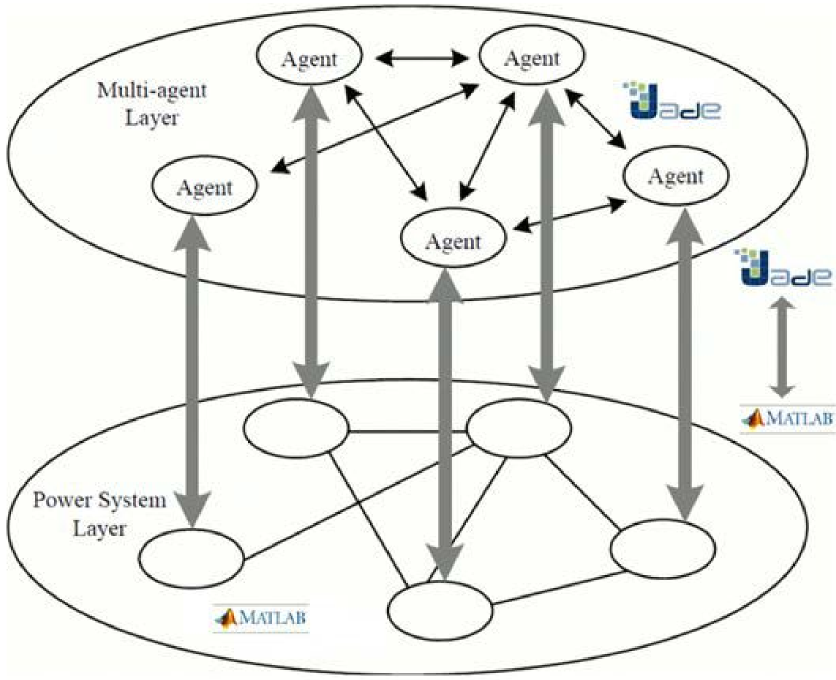


Figure 5.4 Diagram of communication between JADE and MATLAB

System model is to be developed in MATLAB and tested in JADE. Distributed generator is small power supply unit which supplies to local load. To maintain minimal operating capability under fault conditions. Therefore, it is necessary to develop energy distribution control techniques. To achieve this goal it is necessary to make a series of decisions and control actions.

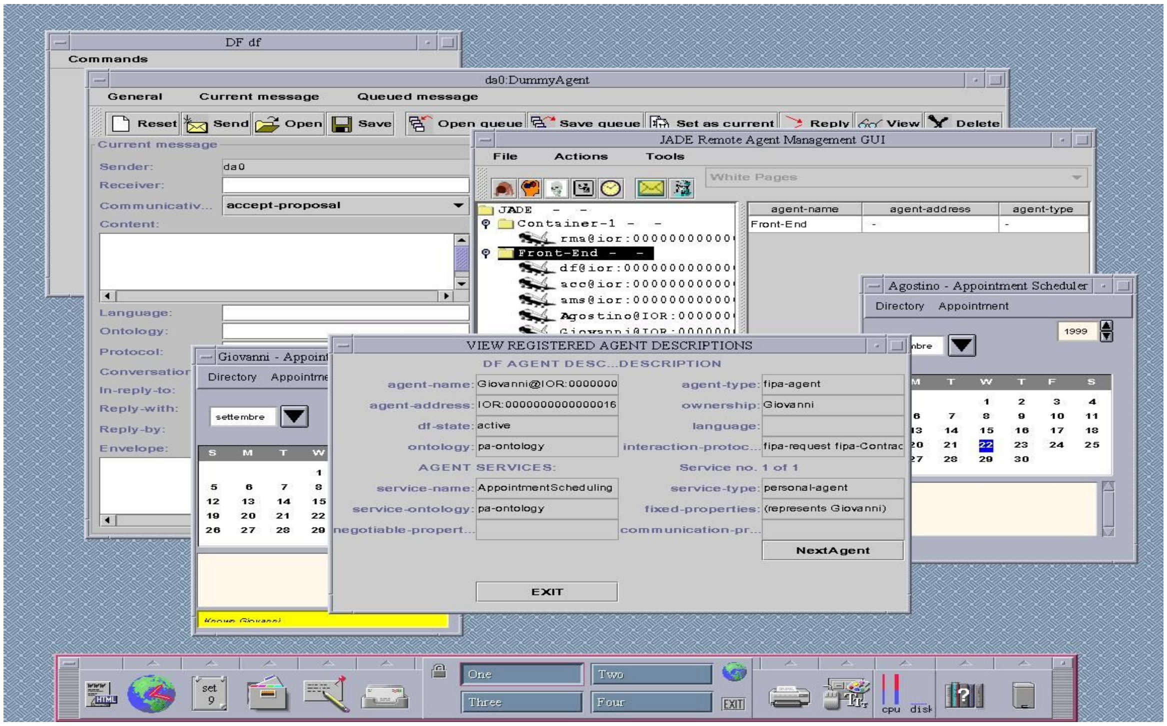


Fig 5.5 JADE software settings to be considered

Implement a test bus system in Matlab simulink. This test system is simulated using Matlab Simulink and MAS is done in JADE . Standard IEEE 3 feeder, 16 buses radial distribution system is as shown in fig 5.5, chosen as system1. Power to the system is fed through 3 feeders, designated as feeder 1, feeder 2, and feeder 3. It has total 16 switches having 13 sectionalizing and 3 tie switches. It has 13 load center and different types of loads are connected. The system has base value 100 MVA and base voltage of 20 kV.

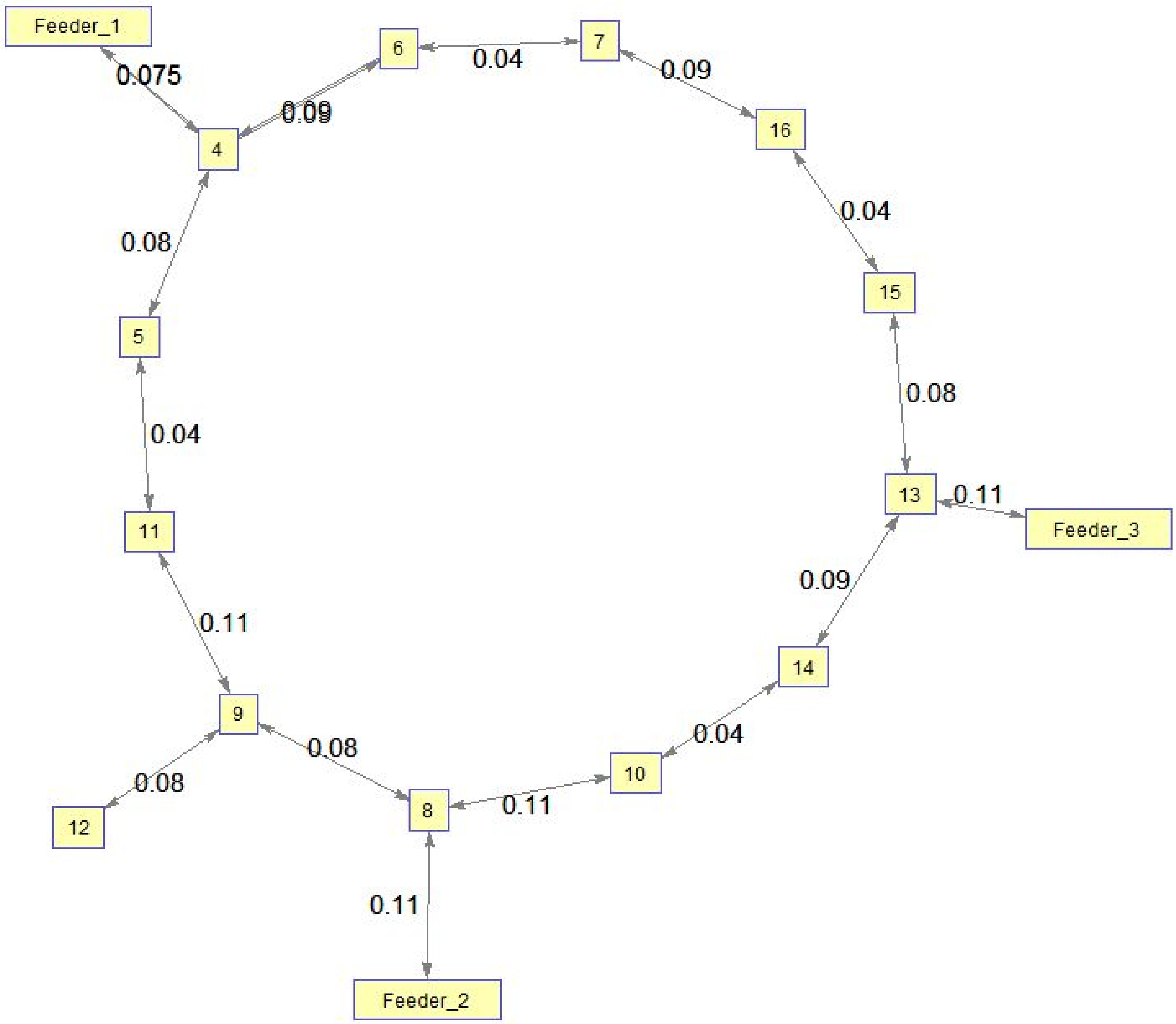


Figure 5.6 Graphical representation of IEEE 3 feeder, 16 buses radial distribution system

Chapter 6: Case Study and simulation Results

EXAMPLE:

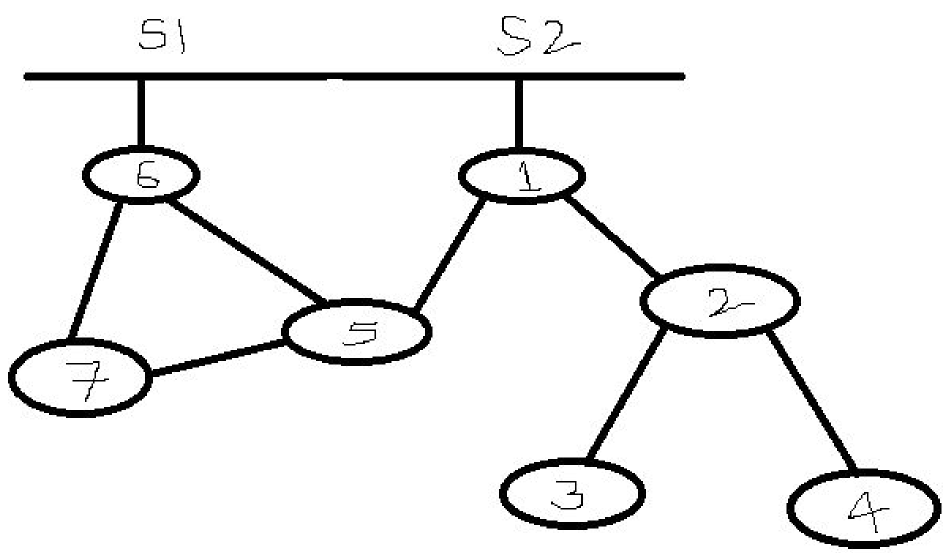


Figure 6.1 7 Bus agent system

Data of this 7 bus agent system is taken as,

No	Neighbours	Pgi	Pli	Xi
1	2,5	200	0	200
2	1,3,4	0	120	-120
3	2	40	0	40
4	2	0	80	-80
5	1,6,7	0	60	-60
6	5,7	120	0	120
7	5,6	0	100	-100

We can notice that after application of Kruskal's algorithm we have minimized the power losses.
Consider a 12 bus system with data as follows

Line Data of 12-Bus System

Branch no	Sending end	Receiving end	R (ohms)	X (ohms)
1	1	2	1.093	0.455
2	2	3	1.184	0.494
3	3	4	2.095	0.873
4	4	5	3.188	1.329
5	5	6	1.093	0.455
6	6	7	1.002	0.417
7	7	8	4.403	1.215
8	8	9	5.642	1.597
9	9	10	2.890	0.818
10	10	11	1.514	0.428
11	11	12	1.238	0.351
12	8	12	1.560	0.465

Load data of 12-Bus System

Node no	PL (kW)	QL (kVAR)
1	0	0
2	60	60
3	40	30
4	55	55
5	30	30
6	20	15
7	55	55
8	45	45
9	40	40
10	35	30
11	40	30
12	15	15

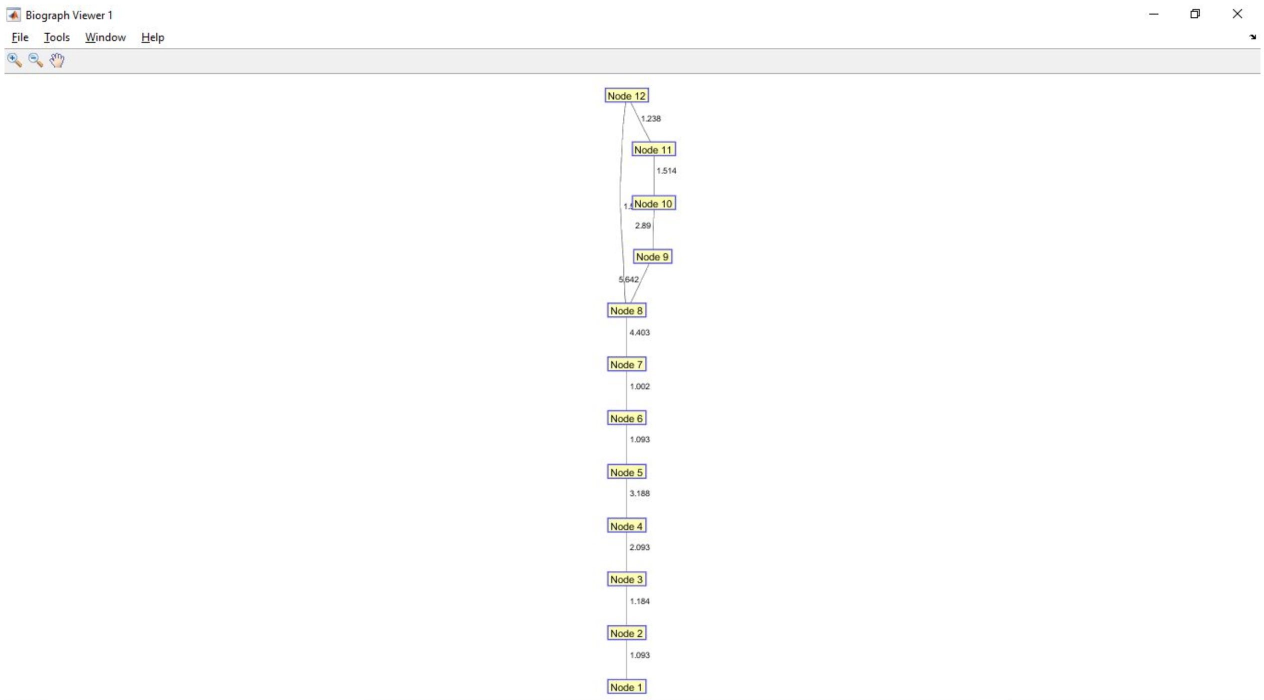


Figure 6.2 12 bus system before applying kruskal's

After MST using Kruskal's we obtain the representation as

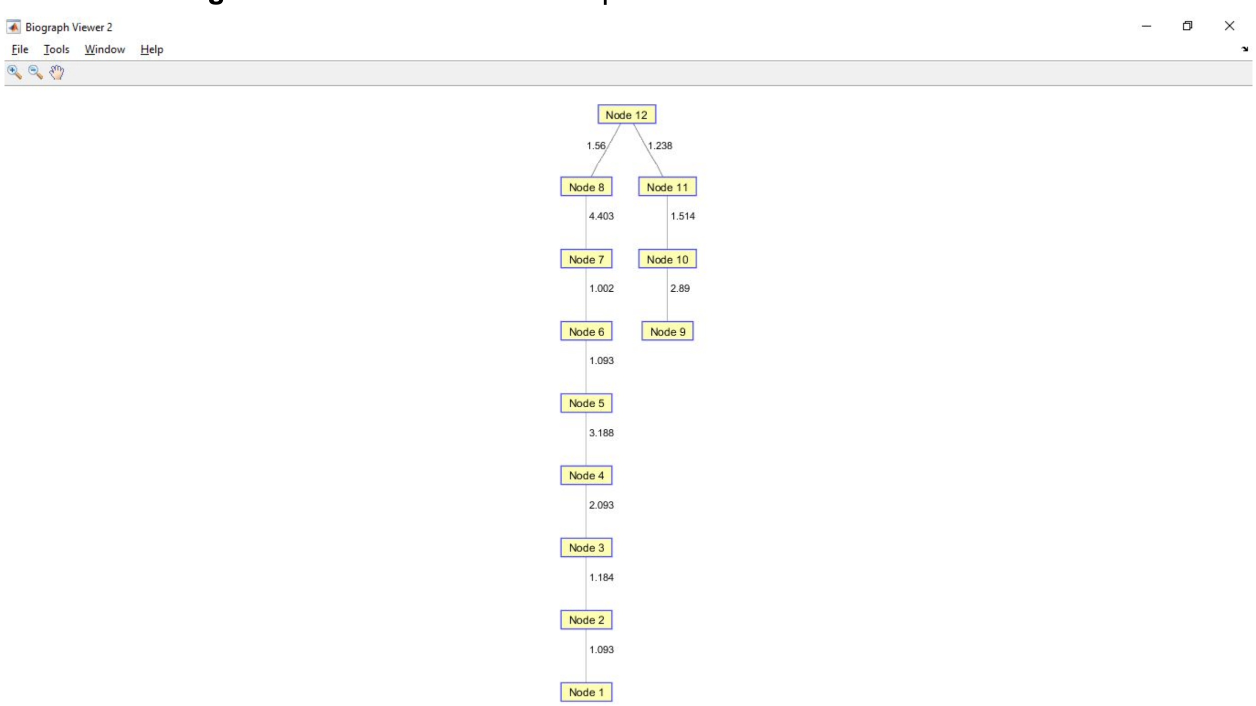


Figure 6.3 12 bus system after MST applying kruskal's

Node no	Voltage Magnitude	Power losses
1 2 3 4 5 6 7 8 9 10 11 12	1.00000 0.99433 0.98903 0.98057 0.96982 0.96653 0.96374 0.95530 0.94727 0.94446 0.94356 0.94335	Total real power loss = 20.71 kW Total reactive power loss = 8.04 kVAr

The solution of the load flow has been given in above Table. It took three iterations to converge by the proposed NR method to converge.

Pseudo code for 12 bus system in MATLAB:

```
W = [1.093 1.184 2.093 3.188 1.093 1.002 4.403 5.642 2.890 1.514 1.238];
DG = sparse([1 2 3 4 5 6 7 8 9 10 11],[2 3 4 5 6 7 8 9 10 11 12],W);
UG = tril(DG + DG')
view(biograph(UG,[],'ShowArrows','off','ShowWeights','on'))
[ST,pred] = graphminspantree(UG)
view(biograph(ST,[],'ShowArrows','off','ShowWeights','on'))
```

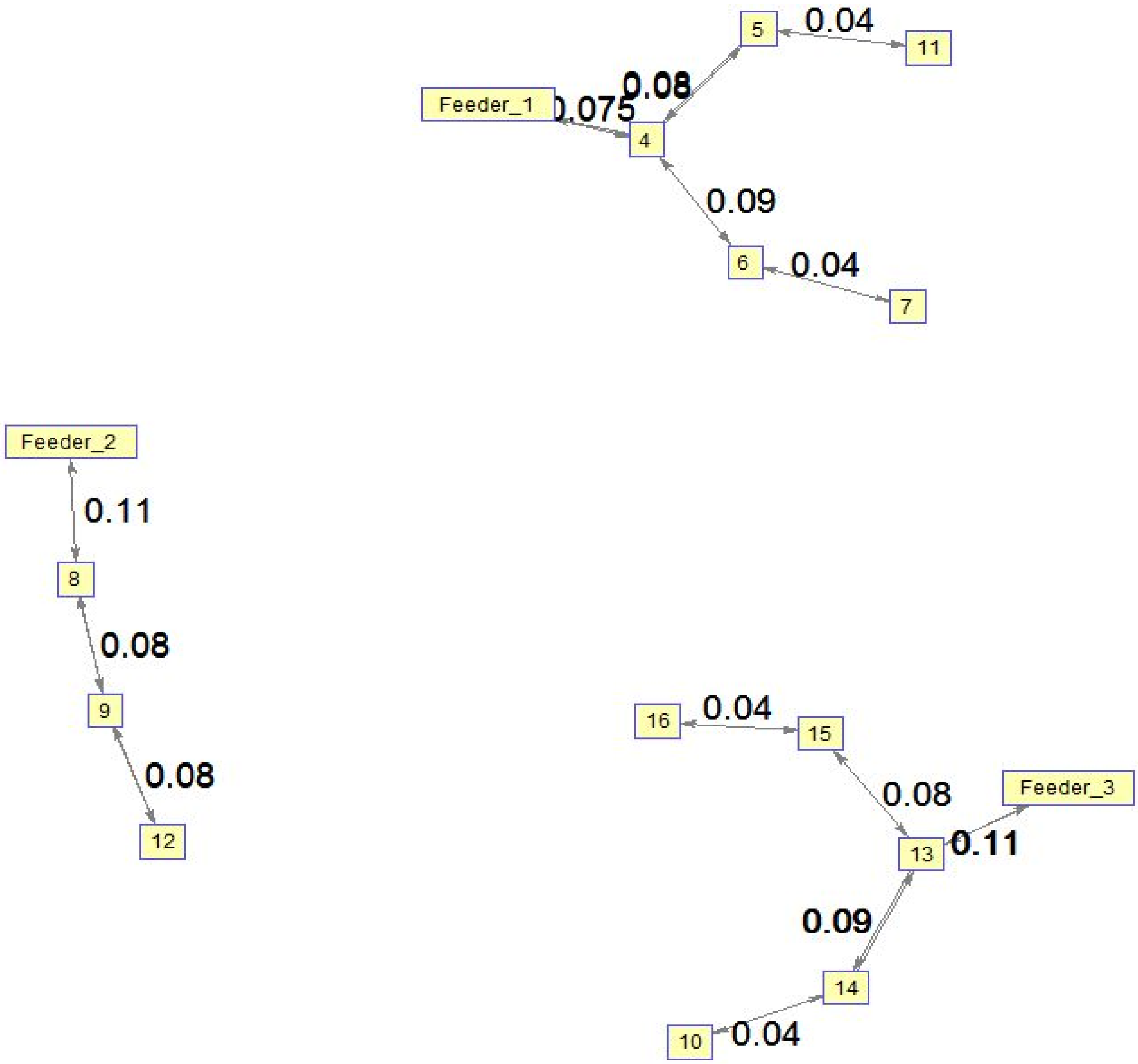


Figure 6.4 MST of 3 feeder, 16 bus system using Kruskal's algorithm

A minimum impedance or shortest path is determined by implementing the kruskal's algorithm in MATLAB. The graphical representation of 16 bus system shown is with total weight of 1.265. The total weight of the minimum impedance path is 0.73. The shortest path for 16 bus system is shown in above figure.

For 3 feeder, 16 bus system line losses are considerably reduces from 0.299 MW to 0.126MW and 0.365 MVAR to 0.157 MVAR.

	Without Kruskal's algorithm		With Kruskal's algorithm	
16 BUS SYSTEM	MW	MVAR	MW	MVAR
LOAD	28.7	5.9	28.7	5.9
GENERATION	28.99	6.26	28.68	5.916
LINE LOSSES	0.299	0.365	0.126	0.157
TOTAL WEIGHT	1.265		0.955	

From the above table it can be observed that total weight has been reduced. Therefore using Kruskal's algorithm was effective to find minimal weight agents nearby in case of power restoration.

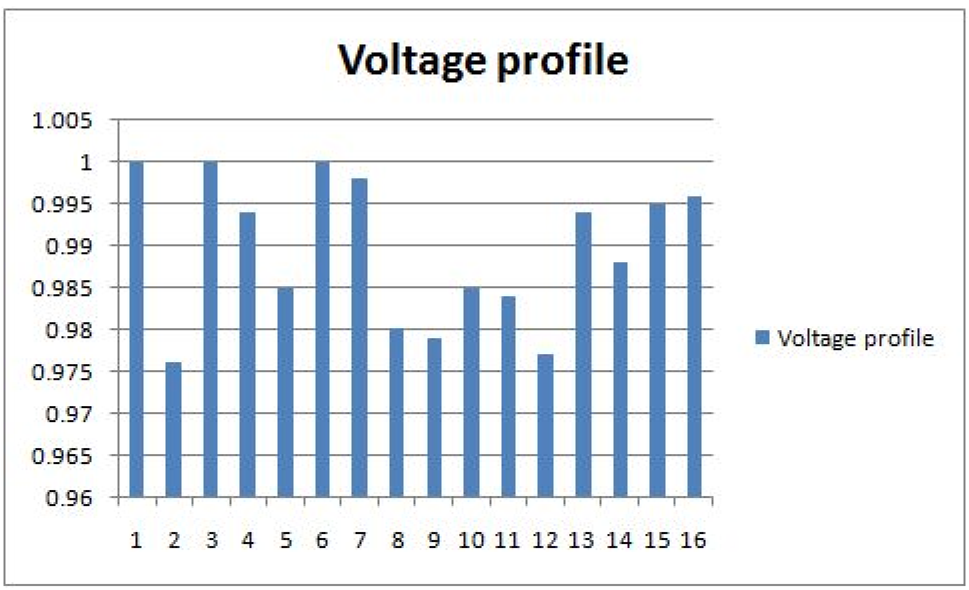


Figure 6.5 Voltage magnitude of buses for 3 feeders 16 bus system

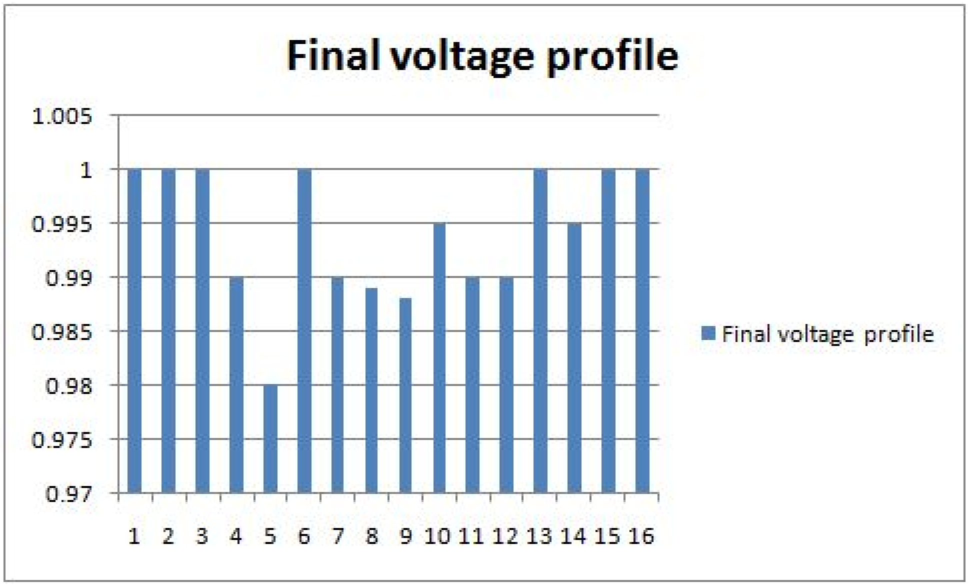


Figure 6.6 Voltage magnitudes of each bus for 3 feeders 16 bus system after getting minimum impedance path

By comparing fig 6.5 and fig 6.6 it can be observed that voltage level has been brought to the constraint level and fluctuations were low.

Summary

It shows less variations and also losses are reduced because shortest path is easily obtained by the proposed algorithm. By observing the simulation analysis, it clearly shows that variation of voltage magnitude characteristics with respective buses is almost straight line. Loads are supplied and losses are also minimized.

When a fault is introduced at bus 5,
figures 6.7 and 6.8 explain the power and voltage variations across the bus and then later how self healing took place and the power has been restored.

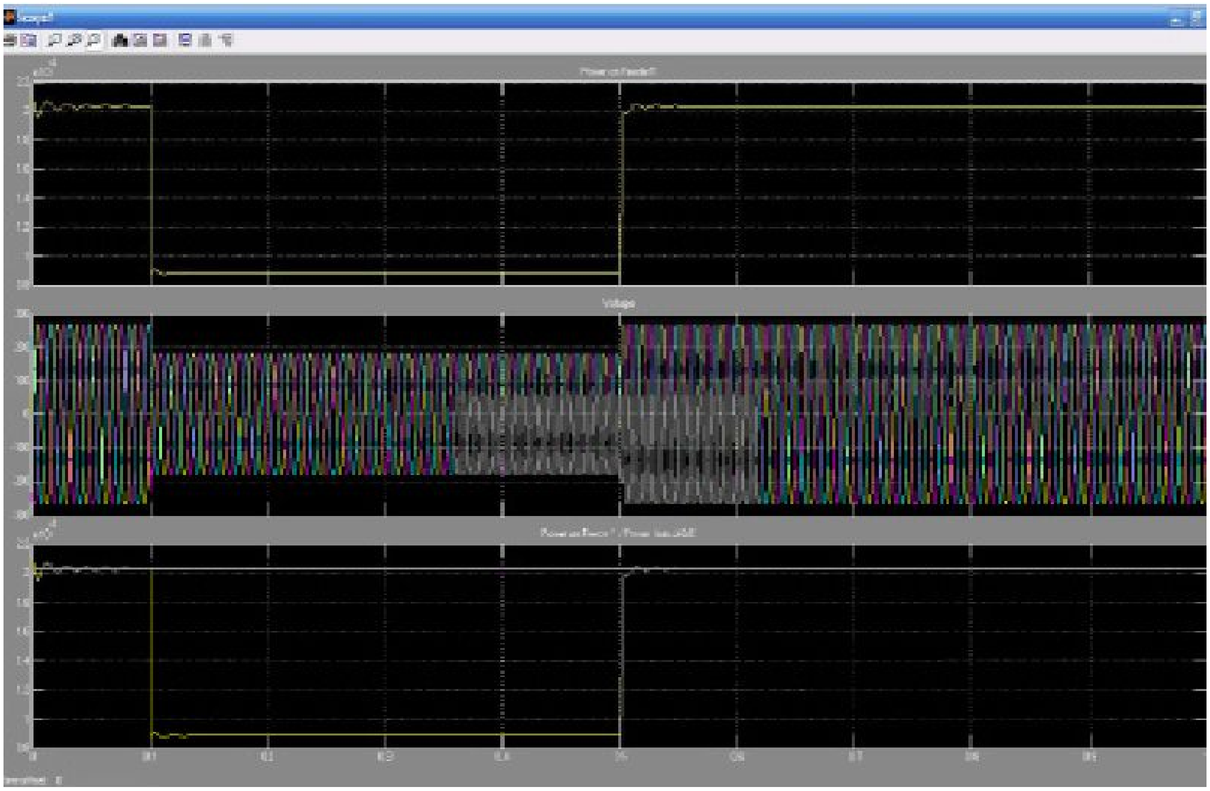


Figure 6.7 Result of fault occurrence

And after the implementation is done and the power is restored,
Then the waveform looks as,

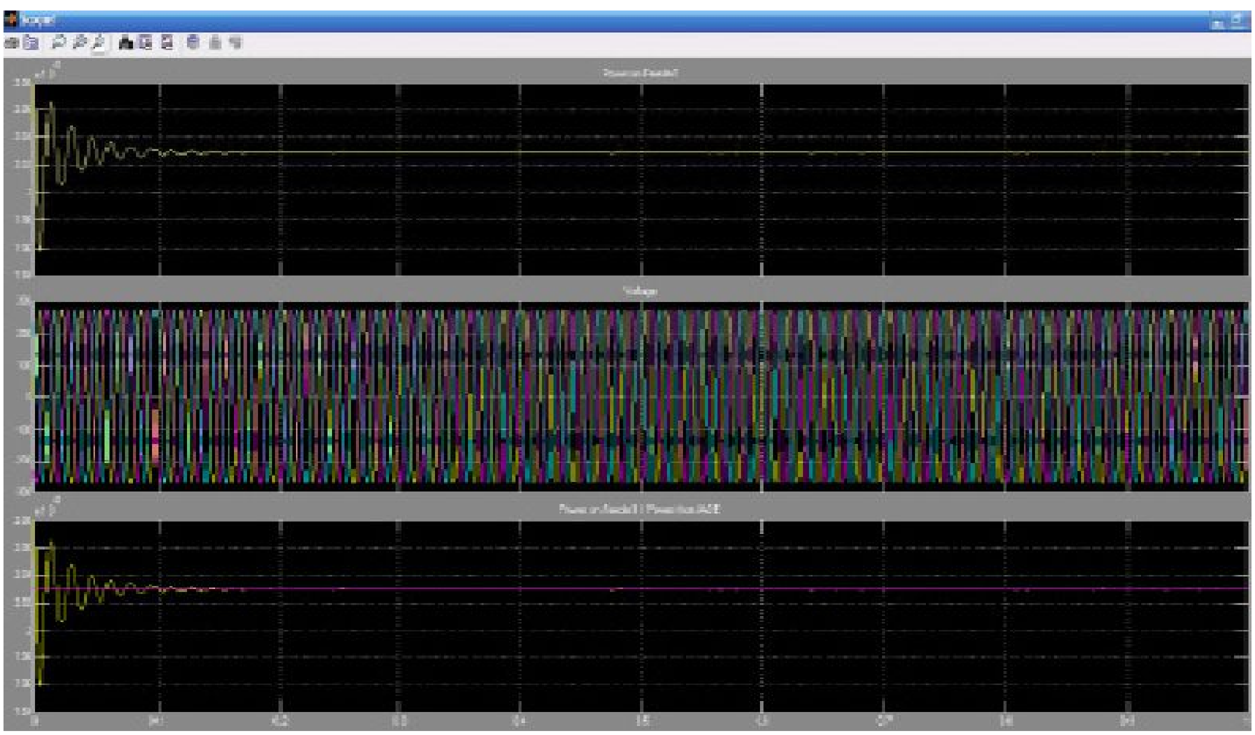


Figure 6.8 Result of power restoration after the fault

Chapter 7
CONCLUSIONS AND REFERENCES

CONCLUSIONS

7.1 Summary

In this research to find decentralized schemes,a simple algorithm has been presented. According to our results we have minimized the power losses in a 33 Bus System using Graph theory concept. Outages and faults are inevitable in distribution power systems. In the process of isolation of the fault, some unfaulted areas lose power. Restoring power to these out-of-service areas as soon as possible is essential. The restoration problem is formulated as an objective function satisfying the system constraints. The objective is to maximise the supply of power for as many loads as possible by giving priority to critical or vital loads.The Graph theoretic DSR problem consists of two major tasks: finding the final optimal network topology and providing a sequence of switching operations leading to this final network topology.After the unbalanced three-phase power flow is performed, the injected real and reactive power at the slack bus will be compared with the generation limits of DERs in the isolated microgrid.

An intelligent agent based MAS have designed for automatic distribution system.A connection between an agent system in JADE and a Matlab Simulink model has been setup, which allows measurement data to be transferred from Simulink and control commands to be sent from JADE to Matlab.The test system is simulated using Matlab Simulink Systems toolbox, and the MAS were implemented using Java programming language and JADE platform. A distribution network carries electricity for transmission network to customers through facilities such as substations, buses and feeders.With this methodology of using Kruskal's algorithm for minimum impedance path, percentage reduction in losses is found to be 17.3%MW and 20.95%MVAR and total weight is reduced from 1.265 to 0.73 for 3 feeder, 16 bus system.This methodology provides the way to calculate parameters of load and also provides the information about feeder overload condition without disturbing the existing network. This technique can be used to reconfiguration and restoring the power in distribution network.Today, power system industry has to be innovative to tackle the many challenges presented by modern power systems consisting of complex interconnections of multiple microgrids.

In this work, an approach for applying agent technology to service restoration of a distribution network has been presented. Two fault scenarios have been considered to verify the capability of the MAS. Furthermore, the potential for simulating a multi agent system behavior linked to a physical model in Matlab Simulink has been investigated. A connection between an agent system in JADE and a Matlab Simulink model has been setup, which allows measurement data to be transferred from Simulink and control commands to be sent from JADE to Matlab.

With the proposed approach, each agent can be implemented with increasingly complex decision-making functionality which may be entirely decentralized and autonomous. Alternatively, each agent may interact and negotiate with other agents to achieve a coordinated and semi autonomous behavior. In such a way, each microgrid has a coordination behavior that is able to respond to disturbances in the power system. Here, the JADE platform respects that each microgrid and potentially each microsource can be managed by entirely different organizations.

Increased public safety will be another benefit of the self-healing smart grid. Grid reconfigurations will quickly de-energize downed wires. Restoring power faster to more people will reduce the impact to customers who rely on the grid for medical necessities, as well as maintaining HVAC to elder care facilities. And, fewer outages reduce the opportunities for criminal acts and civil disturbances. Power quality defects represent another large cost to society, estimated to be in the tens of billions of dollars. The self-healing grid will detect and correct many power quality issues. The self healing grid will accommodate multiple green resources, both distributed and centralized, resulting in substantial reductions in emissions. And, a more efficient grid equates to lower electrical losses.

7.2 Scope for future work

Future work can build upon these purely autonomous decisions with inter-microgrid negotiations that rely on agent’s interaction. In this way, this work presents many opportunities for future developments in the domain of resilient self-healing power grids..Future work should also take into account the interconnection between a microgrid and the main grid.

REFERENCES

[1] M. Eriksson, M. Armendariz, O. O. Vasilenko, A. Saleem and L. Nordström, "Multiagent-Based Distribution Automation Solution for Self-Healing Grids," in IEEE Transactions on Industrial Electronics, vol. 62, no. 4, pp. 2620-2628, April 2015.

[2] Lin, Zhenzhi, Fushuan Wen, and Yusheng Xue. "A Restorative Self-Healing Algorithm for Transmission Systems Based on Complex Network Theory." IEEE Transactions on Smart Grid 7.4 (2016): 2154-2162.

[3] Dave, S. J. K., Sooriyabandara, M., & Yearworth, M. (2011). A Systems Approach to the Smart Grid. In First International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY 2011), Venice, Italy, May 22-27, 2011. (pp. 130 - 134).

[4] Arefifar, Seyed Ali, Yasser Abdel-Rady I. Mohamed, and Tarek HM EL-Fouly. "Comprehensive operational planning framework for self-healing control actions in smart distribution grids." IEEE Transactions on Power Systems 28.4 (2013): 4192-4200.

[5] J. Li, X. Y. Ma, C. C. Liu and K. P. Schneider, "Distribution System Restoration With Microgrids Using Spanning Tree Search," in IEEE Trans*actions on Power Systems,* vol. 29, no. 6, pp. 3021-3029, Nov. 2014

[6] T. Nagata, Y. Tao, H. Sasaki, and H. Fujita, "A multi-agent approach to distribution system restoration," in Proc. IEEE/Power Eng. Soc. General Meeting, Toronto, ON, Canada, Jul. 2003.

[7] Karnouskos, Stamatis, and Thiago Nass De Holanda. "Simulation of a smart grid city with software agents." Computer Modeling and Simulation, 2009. EMS'09. Third UKSim European Symposium on. IEEE, 2009.

[8] S. D. J. McArthur et al., "Multi-agent systems for power engineering applications—Part II: Technologies, standards, tools for building multiagent systems," IEEE Trans. Power Syst., vol. 22, no. 4, pp. 1753–1759, Nov. 2007.

[9] J. M. Solanki, S. Khushalani, and N. N. Schulz, "A multi-agent solution to distribution systems restoration," IEEE Trans. Power Syst., vol. 22, no. 3, pp. 1026–1034, Aug. 2007

[10] S. Chouhan, H. Wan, H. J. Lai, A. Feliachi, and M. A. Choudhry, "Intelligent reconfiguration of smart distribution network using multi-agent technology," in Proc. IEEE Power Energy Soc. Gen. Meet., Calgary, AB, Canada, Jul. 2009, pp. 1–6.

[11] Java Agent Development Framework (JADE). [Online]. Available: http://jade.tilab.com/

[12] Foundation for Intelligent Physical Agents (FIPA), 2007. [Online].
Available: http://www.fipa.org

[13] A. L. Dimeas and N. D. Hatziargyriou, "Operation of a multi-agent system for microgrid control," IEEE Trans. Power Syst., vol. 20, no. 3, pp. 1447–1455, Aug. 2005.

[14] Joseph B. Kruskal, Jr. Proceedings of the American Mathematical Society
Vol. 7, No. 1 (Feb., 1956), pp. 48-50

[15] M. M. Nordman and M. Lehtonen, “An agent concept for managing electrical distribution networks,” IEEE Trans. Power Del., vol. 20, no. 2, pp. 696–703, Apr. 2005

[16] Kashyap, Neelabh, et al. "Automated fault location and isolation in distribution grids with distributed control and unreliable communication." IEEE Transactions on Industrial Electronics 62.4 (2015): 2612-2619.

[17] Cavalcante, Patricia L., et al. "Centralized self-healing scheme for electrical distribution systems." IEEE Transactions on Smart Grid 7.1 (2016): 145-155.

[18] Anisha, K., and M. Rathina Kumar. "Jade Implementation of Power Restoration in Automated Distributed System." Indian Journal of Science and Technology 8.19 (2015).

APPENDIX

```
set CLASSPATH=%CLASSPATH%;.;c:\jade\lib\jade.jar;
       c:\jade\lib\jadeTools.jar; c:\jade\lib\Base64.jar;
                         c:\jade\lib\http.jar;c:\jade\lib\iiop.jar
setenv JADE_LIB  "${HOME}/jade/lib"
   setenv CLASSPATH ".:${JADE_LIB}/jade.jar:${JADE_LIB}/iiop.jar:${JADE_LIB}/http.jar\
       :${JADE_LIB}/Base64.jar:${JADE_LIB}/jadeTools.jar"
//Search agents if you already know the address:
 AID myID = getAID();
       for (int i=0; i<agents.length;i++)
       {
           AID agentID = agents[i].getName();
           System.out.println(
               ( agentID.equals( myID ) ? "*** " : "    ")
               + i + ": " + agentID.getName()
           );
       }

//To get arguments:
 import jade.core.Agent;
     public class ParamAgent extends Agent
     {
        protected void setup()
        {
            Object[] args = getArguments();
            String s;
            if (args != null) {
                for (int i = 0; i<args.length; i++) {
                    s = (String) args[i];
                    System.out.println("p" + i + ": " + s);
                }
                // Extracting the integer.
                int i = Integer.parseInt( (String) args[0] );
                System.out.println("i*i= " + i*i);
            }
        }
     }
//Client message to find agents:
 ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
     msg.setContent( "L<M" );
     for (int i = 1; i<=agents.length; i++)
        msg.addReceiver( new AID( "store" + i, AID.ISLOCALNAME) );
     send(msg);
//Server message:
             public void action()
```

```
            {
              ACLMessage msg = receive();
              if (msg!=null) {
                  System.out.println( " - " +
                     myAgent.getLocalName() + " <- " +
                     msg.getContent() );
   //Sever reply:
                  ACLMessage reply = msg.createReply();
                  reply.setPerformative( ACLMessage.INFORM );
                  reply.setContent(" yes" );
                  reply.send();
               }
              block();
            }
//If message is received, then:
 public class Receiver extends Agent
       {
           protected void setup()
           {
              addBehaviour(new CyclicBehaviour(this)
              {
                   public void action()
                   {
                      ACLMessage msg= receive();
                      if (msg!=null)
                          System.out.println( " - " +
                             myAgent.getLocalName() + " <- " +
                             msg.getContent() );
                      block();
                   }
              });
           }
```