Agent Technologies for Control Applications in the Power Grid

Angel A. Aquino-Lugo, Thomas Overbye
University of Illinois Urbana-Champaign
aaquino3@illinois.edu, overbye@illinois.edu

Abstract
Remotely controlled network devices will transform the way power systems are operated and studied. One possible application for this is the use of agent based technologies to implement decentralized control algorithms. The agents would perform local control actions instead of centralized control actions. Two cases are presented. The second case would be examined in detail. In the first case, the power losses were minimized using a decentralized algorithm and the results were similar to those found using a centralized algorithm. In the second case, transmission line overloads are relieved by controlling the load in the system. For this case, a detailed algorithm to control loads was presented to show the integration that would be required between the transmission and distribution network. This work showed that to implement decentralized control a reliable communication network within the power system will be necessary.

1. Introduction

Today, much conversation is being made about how the electric power grid would look in the future. The common consensus is that it would incorporate new technologies that would let us control the grid in a “smart” way. The problem is that many have different definitions about what “smart” grid means. A “smart” grid can be defined as the utilization of new digital and intelligent devices to replace the old analog devices in the power network. In this paper, “smart” grid relates to using those new intelligent devices to allow for remote control providing a new opportunity for decentralized control.

The challenges, whatever the definition, are enormous. The stimulus law of 2009 provides billions of dollars for smart grid funded projects and studies. Certainly the transformation of the grid would change the way it is operated and analyzed. In this paper, some new ideas on how to control the power grid in a decentralized but intelligent scheme are studied. Some examples are presented, as well as the challenges they bring to the electric power grid.

2. Agent Technologies Previous Work

Typically, the power grid system is operated and coordinated in a centralized way. Every time the power network fails the central control center determines which system elements and control actions should be implemented to either save the system from collapse or to reconfigure the system after an outage [1]. In recent years, with the introduction of remotely controlled power network devices, new possibilities for control strategies are starting to emerge. A number rely on a more local control but with decentralized algorithms. Thus the concepts of agents were studied to get more local and faster responses for applications that formerly were coordinated by a centralized control.

One of the first concepts was the self-healing of power distribution networks in combat ships. During battles, the ships can suffer severe damage to the electrical system, and in a combat situation it is important to maintain the availability of energy to the loads to keep the ship operational [2-3]. People quickly realized that this concept could be applied to power system distribution networks. In [4], the authors present a multi-agent system (MAS) approach for a decentralized solution for the power system reconfiguration problem using Matlab Simulink S-functions as agents. Following the same approach as in [1], a restoration algorithm applying an expert system type of solution was presented using Matlab Simulink and the Stateflow toolbox. In [3] an intelligent power routers (IPR) scheme was proposed where control can be detached from the central control sites, and delegated to IPRs that would be distributed over the entire electric network to initialize and coordinate control actions.

More recently, research has focused on studying centralized problems, other than power system restoration, in a decentralized scheme. For example, the optimal power flow (OPF) can be solved by using decentralized algorithms to parallelize the solution to get faster results. In [5] a parallelized OPF suitable for a coarse-grained distributed implementation is...
presented. The authors implemented the algorithm with three different mathematical decompositions in order to coordinate the distributed OPF. In [6], a distributed OPF is presented, which incorporates discrete control variables. As part of the solution the authors presented an algorithm to exchange data among different subsystems that were created during the decentralization implementation. In [7], the concept of multi-area control was presented. The authors assumed that areas are determined independently but influenced by flexible AC transmission systems (FACTS) devices. The idea is that one can solve a local optimization algorithm while assuming that the influences of other areas are constant. This concept will be studied in detail in the decentralized algorithm for minimizing losses that is presented later in Section 3. All of the decentralized OPF solutions assume that the data will be exchanged. Although the purpose of these algorithms is to parallelize the OPF solution, those notions could be used to achieve a decentralized scheme that could be implemented in real life applications. One control application is active load control. This topic is discussed in detail in Section 4.

All of these studies require some kind of communication and intelligence to manage the data, but none addresses the issue of data exchange among the different decentralized areas.

In recent years, more attention has been given to multi-agent systems (MAS) applications as an alternative to implement decentralized control algorithms in real life. For this reason, the IEEE Power Engineering Society’s MAS Working Group presented a two-paper series [8-9] about the MAS technologies applied to the power systems. The main conclusion was that with more experience and research in the matter, a better understanding of the different standards, methodologies, and agent models needed could be achieved. With that in mind, the work presented in this paper addresses the challenges of studying these MAS technologies, and their possible application in the “smart” grid.

3. Decentralized Losses Minimization

Algorithm

Decentralized control could be used for applications other than system restoration after a blackout. One is power losses minimization in the distribution system by switching capacitors. Typically capacitors are used in the distribution system to provide reactive power support, voltage regulation and reduction of power losses. To formulate this problem, the equation used for power losses in this optimization problem will be defined. The branch loss equation is defined by:

\[ P_{\text{losses}}(x_{c}^j) = r_j \cdot |I_j|^2 \]  \hspace{1cm} (1)

where \( x_{c}^j \) is the \( i \)th capacitor in the \( k \)th system, \( r_j \) is the resistance of the \( j \)th branch and \( I_j \) is the current of the \( j \)th branch. The forward/backward method is used to solve the distribution power flow. This means that the current magnitude \( I \) is defined as:

\[
\begin{align*}
|I^j| &= (\text{real}(I^j)) + (\text{real}(I^j)) + (\text{real}(I^j)) \pm \sqrt{(\text{imag}(I^j) + (\text{imag}(I^j)) + (\text{imag}(I^j)))^2} \pm \sqrt{(\text{imag}(I^j) + (\text{imag}(I^j)) + (\text{imag}(I^j)))^2}, \\
&= (\text{real}(I^j) + (\text{real}(I^j)) + (\text{real}(I^j))) \pm \sqrt{(\text{imag}(I^j) + (\text{imag}(I^j)) + (\text{imag}(I^j)))^2}.
\end{align*}
\]  \hspace{1cm} (2)

where \( I^j \) is the load current connected at the \( j \)th branch, \( F_i \) is the current coming from other branches connected at the \( j \)th branch and \( I_{\text{incl}} \) is the current coming from the \( i \)th capacitor connected to the \( j \)th branch. Notice that these currents are at the end point of a branch. Also, for this analysis the capacitors where assumed to be constant power devices. This assumption is based on the fact that for a small time period these capacitors could be seen as a constant power source because the local region responds in a matter of seconds in order to minimize system losses.

The resulting optimization problem is a mixed-integer programming problem because the capacitors are a discrete variable that is either on or off. But for the analysis of this problem, the discrete variables are relaxed and are assumed to take constant values such that the mixed-integer programming problem can be reduced to a nonlinear programming problem and the corresponding non-integer values of the capacitors are rounded to the closest integer solution. This relaxation is allowable because the capacitors where already installed in the system and a previous analysis was performed to ensure that their operation would not create an unfeasible system state.

Now the optimization problem can be defined as follows:

\[
\min \sum_{j=1}^{N} P_{\text{losses}}(x_{c}^j)
\]

s.t. \( DPF(x_{c}^j) \) constraints  \hspace{1cm} (3)

\[ 0 \leq x_{c}^j \leq (x_{c}^j)_{\text{max}}, \]

where \( DPF \) are the distribution power-flow constraints that, in our case, are the forward/backward method equations. It is important to notice that this is a centralized optimization problem formulation. In this study, a decentralized optimization problem is considered in which each
local area (to be controlled by an agent) would optimize the local system while sharing constraints and fixed quantities with the neighbor agents. The approach and mathematical notation follows the work in [10] where the neighborhood states are kept fixed for the $k^{th}$ agent during the optimization problem solution. Each decentralized region is controlled by an agent.

To understand this decentralized optimization problem, let us introduce the set $\{xc_q \}^k$, which is the set of subsets of neighborhood states to which the $k^{th}$ subsystem is associated [10]. This set is now used to define the decentralized optimization problem as follows:

$$\min_{xc} f_k (xc) = \sum_{j=1}^{N} P_{loss_j} (xc_j^k)$$

s.t. 

$$DPF(xc_j^k | xc_{j'}^k) \text{ constraints}$$

$$0 \leq xc_j^k \leq (xc_j^k)_{max}^k,$$ \hspace{1cm} (4)

where the notation $DPF(xc_j^k | xc_{j'}^k)$ is used to represent the constraints equations of $xc_j^k$ based on the fact that the neighborhood states $\{xc_q \}^k$ are kept fixed.

In our case, if we leave the neighborhood states $\{xc_q \}^k$ fixed each $k^{th}$ agent would see that the power coming from the $(k+1)^{th}$ neighbor state would be constant and the $(k-1)^{th}$ neighborhood state network can be modeled as a Thevenin equivalent circuit at the interconnection bus with the $k^{th}$ state. This is important because each agent would be solving an optimization algorithm based on the fact that their neighbor’s states are fixed. Thus, each agent would need to know the Thevenin equivalent voltage and impedance at the start of the local subsystem feeder in order to perform the local analysis. In other words, the $(k)^{th}$ neighbor agent sends its Thevenin equivalent circuit up to the point of connection to the $(k+1)^{th}$ neighbor agent. As the distribution system is single-phase and balanced, the equivalent of a $k^{th}$ subsystem was a Thevenin equivalent circuit. For this Thevenin equivalent circuit, the loads and the capacitors are modeled as parallel shunts connected to the buses. In Figure 1 a representation of this assumption on a radial feeder with 3 agents is presented.

This decentralized optimization problem can now be solved by implementing a modified form of the penalty augmented-cost function for the $k^{th}$ agent subsystem optimization,

$$F_k(xc^k,\beta_k | \{xc_q \}^k) = \beta_k f_k(xc^k) + P_k(xc^k | \{xc_q \}^k),$$ \hspace{1cm} (5)

where $\beta_k$ is a local penalty parameter and $P_k$ is a penalty function for constraints given in (4). $P_k$ can be decomposed in

$$P_k(xc^k | \{xc_q \}^k) = P_k(xc^k | \{xc_q \}^k) + P_{ink}(xc^k | \{xc_q \}^k),$$ \hspace{1cm} (6)

where $P_{ink}$ correspond to the local constraints and $P_{ink}$ correspond to the interconnected quantities. But since in this case these quantities ($P_{ink}$) can not be directly controlled by the $k^{th}$ agent, $P_{ink}$ would have no constraints and will be directly enforced by the $P_{ink}$ constraints.

If these assumptions are implemented, each agent should exchange information to the neighbor agents after each decentralized optimization problem is performed. The agent at the start of the feeder would provide the equivalent Thevenin voltage magnitude and angle, as well as the Thevenin impedance, to its $k^{th}$ connected agent. The agents between two neighboring agents should receive the Thevenin equivalent from the $(k-1)^{th}$ agent and the constant PQ coming from the $(k+1)^{th}$ agent. This constant PQ represents the power flowing from the $k^{th}$ subsystem to the $(k+1)^{th}$ subsystem through the feeder directly connected to the $k^{th}$ agent subsystem. At the same time, the $k^{th}$ agent should send the constant PQ at the connection point to the $(k-1)^{th}$ agent and also send the Thevenin equivalent circuit to the $(k+1)^{th}$ agent. If the agent is the last bus of the feeder it would receive only the Thevenin equivalent circuit from the $(k-1)^{th}$ agent and would send the constant PQ at the connection point to the $k^{th}$ agent.

This decentralized optimization problem stops when the change in the constant PQ exchange between agents is less than some tolerance.

### 3.1 Ten, Thirteen and Thirty-Four Bus Feeder Loss Minimization Examples

The decentralized optimization algorithm was first implemented on a 10-bus feeder (Figure 2). In this
case, the results will always converge to a feasible solution such that the DPF constraints will always be satisfied. This means that these constraints could be taken out of the optimization algorithm, thus simplifying the solution of the optimization algorithm. Also the solution of the radial distribution system was calculated by the implementation of the forward/backward method in a Matlab code.

From Section 3, all of the loads are modeled as constant PQ devices and the capacitors as constant Q devices. In the first case (C1), three agents are placed so as to control the capacitors in the feeder. The first agent controls capacitor 1 connected at bus 3 and is responsible for buses 1 through 3. The second agent is responsible for capacitor 2 connected to bus 6 and only controls buses 3 through 6. The third agent is responsible for capacitor 3 connected at bus 10 and controls buses 6 through 10. The results are presented in Table 2 and the decentralized results are compared with a centralized optimization solution. It is important to mention that capacitors 1 have a maximum kVA of 3000kVA and capacitors 2 and 3 have a maximum of 1500kVA. The capacitors were rounded up to the nearest hundred after the optimization algorithm solution converged.

In the second scenario the agents and the capacitors are placed in a modified IEEE 13-bus distribution system (Figure 3). The voltage regulator at the start of the feeder is not modeled, the system is modeled with balanced loads, and the distributed loads and transformers are eliminated. In this case (C2), the first agent controls capacitor 1 connected at bus 6 and is responsible for buses 1 through 6. The second agent controls capacitors 2 and 3 connected to buses 9 and 11 respectively, and is responsible for buses 7 through 13. It is important to mention that capacitors 1 and 2 have a maximum kVA of 200kVA and capacitor 3 has a maximum of 100kVA.

In the last scenario, the agents and the capacitors were tested in a modified IEEE 34-bus distribution system (Figure 4). This scenario has the same modifications as in the IEEE 13-bus system. In this case (C3) there are four agents. The first controls capacitor 1 connected at bus 5 and is responsible for buses 1 through 7. The second controls capacitor 2 connected to bus 12 and is responsible for buses 8 through 16. The third controls capacitor 3 connected to bus 22 and is responsible for buses 17 through 24. The fourth controls capacitors 4 and 5 connected to buses 27 and 29 respectively, and is responsible for buses 25 through 34. It is important to mention that capacitors 1 to 4 have a maximum kVA of 100kVA and capacitor 5 a maximum of 200kVA.

In Table 1, the results for all three cases are presented. The decentralized and the centralized algorithms converge to similar capacitors sizes after the capacitors were rounded to the nearest integer. The kVA of the capacitors was the same for all cases. As a result, the power losses (Table 2) of the decentralized and centralized optimization algorithms have the same magnitude for all of the cases. The results from this algorithm show that using decentralized algorithms could achieved a solution close to the one obtained from a centralized algorithm. In these cases agents were implemented to control the capacitors and to exchange data throughout the regions. The problem found that the entire algorithm iterates for about 4 to 8 times depending of the case. These iterations create a long
simulation time of more than 30 seconds if the agent communication is included. The communication protocol presented in the Section 4.3.1 is suitable for running decentralized optimization algorithms.

### Table 1 Capacitor kVA comparison between decentralized and centralized optimization algorithms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C1: Bus 3</td>
<td>3000</td>
<td>3000</td>
<td>3000</td>
<td>3000</td>
</tr>
<tr>
<td>C1: Bus 6</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
<td>1500</td>
</tr>
<tr>
<td>C1: Bus 10</td>
<td>450.8</td>
<td>0</td>
<td>551.4</td>
<td>0</td>
</tr>
<tr>
<td>C2: Bus 6</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>C2: Bus 9</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>C2: Bus 11</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C3: Bus 5</td>
<td>100</td>
<td>100</td>
<td>84.6</td>
<td>100</td>
</tr>
<tr>
<td>C3: Bus 12</td>
<td>88.7</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C3: Bus 22</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C3: Bus 27</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C3: Bus 29</td>
<td>111.6</td>
<td>200</td>
<td>182.2</td>
<td>200</td>
</tr>
</tbody>
</table>

### Table 2 Loss comparison between decentralized and centralized optimization algorithms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>688.4</td>
<td>688.4</td>
<td>783.4</td>
</tr>
<tr>
<td>C2</td>
<td>87.84</td>
<td>87.84</td>
<td>124.1</td>
</tr>
<tr>
<td>C3</td>
<td>75.07</td>
<td>75.07</td>
<td>107.3</td>
</tr>
</tbody>
</table>

### 4. OPF Load Control with Distributed Agents

In Section 2, various algorithms to perform decentralized OPF were presented. These were chosen because they can be parallelized to speed up the solution of larger systems. Also it was assumed that communication was to be implemented in order to exchange information between different regions of the decentralized algorithm. The communication is presented in two simple examples. The analysis shows the requirements necessary to implement that kind of solution and which simulation environments are suitable for this application.

#### 4.1. Distributed Agents and Load-Control OPF

In this paper a similar and related problem to decentralized OPF is investigated. When a transmission line is out of service in a power system, it can create line-flow overloads in other lines that are in service. Line overloads in a transmission system may prevent power transfer. To solve this problem, the use of distributed agents can be implemented to coordinate a solution to relieved the line overloads. In order to perform this task the power system is divided to regions (Figure 5).

![Figure 5 Power system divided in regions](image)

Each region (TR_A) is divided into transmission and distribution regions (Figure 6). The transmission region (TR_A) is responsible for network devices that can be controlled. In this case, it was assumed that some of the loads can be controlled. The agents at each load bus (B_A) would know at a specific time the amount of load that is connected to that bus, as well as the load amount that can be controlled. These bus agents (B_A) communicate with many of the distribution network devices and obtain and exchange data among the distribution network agents. One of these distribution network devices is the smart meter that is part of the advance meter infrastructure (AMI). Using data obtained from the AMI smart meter, the bus agent knows the amount of load that can be...
controlled (Figure 6) The AMI data would be collected by distribution agents (D_A) that exchange information among themselves and the B_A. Two type of load control can be performed. One is the pluggable hybrid connected to the grid to inject power and the other is the disconnection of loads for shedding purposes, but only for specific situations such as the line-overload case presented in Section 4.2. Based on the information collected at each bus agent, the region agent performs a local load OPF to determine the amount of load that needs to be connected (in case of pluggable hybrids) or disconnected (for load shedding) to relieve the overloaded lines. Also the regional agent negotiates with other regional agents in the transmission grid if a solution is not obtained. For this last case a decentralized OPF looks for an optimal solution. This decentralized OPF is the subject of future research.

![Figure 6 Transmission region and distribution region (TN is transmission network and DN is distribution network)](image)

4.2. Load Control OPF Formulation

To show how the proposed control algorithm would be implemented in real life, a small case example is presented. A load control OPF was implemented. The optimization problem can be defined as follows:

\[
\min \sum_{j=1}^{N} L_j
\]

s.t. \(PF\) constraints

\[
L_{j\min} \leq L_j \leq L_{j\max},
\]

\[
L_{\min} \leq L_j \leq L_{\max}
\]

In this load OPF, the amount of load \((L_j)\) to be controlled is minimized while satisfying the power flow constraints \((PF)\) and the transmission line flow \((L_{FL})\) limit. To satisfy the line-flow limit the sensitivity of the power flow in a line \(l\) after a change in power at a bus \(j\) was calculated. The controllable loads can be controlled with a minimum and a maximum, meaning that this information would be available at the moment of optimization. An overloaded line case implementing the local OPF while interacting at the same time with the distributed intelligent agents is presented. The agents gather information about the amount of load that can be controlled at a certain moment in time. Remember that this information is collected from the smart meters and the distribution agents that are currently distributed in the distribution power network. These agents were simulated using JADE (Java Agent Development Framework).

4.3. Agent Simulation in JADE and OPF Algorithm

JADE is a JAVA framework for developing FIPA (foundation for intelligent physical agents) compliant agent applications and is one of the most widespread agent-oriented and completely distributed middleware systems to create agents. The framework provides a flexible infrastructure that allows easy extension with add-on modules and is one of the platforms proposed by the IEEE Power Engineering Society’s MAS Working Group [10-11]. Agents can be created and simulated using the JADE platform, and the power distribution system can be modeled using Matlab. Thus a connection with Matlab can be established to obtain power-flow and OPF optimization results.

The following OPF algorithm was implemented with the simulated JADE agents:

Step 1) Each bus agent (B_A) gets the bus voltage magnitude, angle, and lines power flows of directly connected lines to the bus from Matlab. This information is used in the Load OPF.

Step 2) Each B_A sends data to the transmission region agent (TR_A) every time there is a change in the data obtained from Matlab. Part of the data includes the information about load that can be controlled. The agent-to-agent communication protocol is explained in Section 4.3.1.
Step 3) If a line outage or a line overflow is detected, then the regional agent performs the Load OPF with the most recent available data. Once a solution is obtained, the result is sent to each B_A.

Step 4) After the B_A verify that the amount of load requested by TR_A can be controlled, the B_A performs the control. A new power flow is obtained and the data is collected by the B_A and sent again to the TR_A.

Step 5) Once all of the B_A agents perform the requested load control, the algorithm stops.

4.3.1 Agents Communication Protocol. It is important to notice from this algorithm that the transmission region agent (TR_A) and the bus agents (B_A) are exchanging messages and data among themselves. The agents implemented the FIPA Contract Net Interaction protocol [11]. This protocol has the following algorithm.

Step 1) An initiator agent requests a task to be performed by other agents. Then the initiator calls for proposals (CFP) to the participant agents.

Step 2) The participant agents received the CFP and can either tell the initiator that they can perform the proposal or refuse it.

Step 3) If the participant agent indicates that the proposal can be satisfied, it sends that message to the initiator.

Step 4) The initiator confirms the participation proposal.

Step 5) If the participant agent receives the confirmation of the proposal, then the participant agent performs the task.

Step 6) After the task is performed, the participant agent informs the initiator that the task was performed.

Step 7) The initiator confirms that it has received the message that the task was performed. After this, no further requests to perform that task are sent to any other agent.

This interaction happens every time a message is exchanged between agents. These messages are similar to human communicative acts. More details on the message protocol and language communication acts are presented in [11].

4.4. Case Study for the OPF Algorithm

The example power system used in the OPF case is presented in Figure 4. This case results from the disconnection of line 2-5 because of an outage. The affected system has one overloaded line (2-6) at 92%; but desired value is to be below 84%. The agents are controlling three loads that are also identified in Figure 7. The amount of load that can be controlled at each of the buses, as well as the results, are shown in Table 3.

After the load OPF was calculated the most severe line 2-6 overload was reduced from 92% to 84%. Table 3 shows the results of the OPF, as well as the original and controllable loads. In order to achieve this goal the net load at bus 6 was reduced from 110 MW to 83.48MW. This result was obtained by connecting some pluggable hybrid cars that were simulated as a generator. The net effect on the bus is represented as a load shedding because this injection of pluggable hybrids would be in the distribution network and would be seen in the transmission network as a change in the bus injection. But by coordinating the load response and the resulting bus injection change, the desired line flow can be obtained. Note that the load in bus 7 also has a net load change. This is a consequence of satisfying the line 3-7 loading constraint of 85%. It is important to mention that these results shown in Table 3 were obtained using the agent scheme presented in Figures 5 and 6.

Figure 7 Case study and controllable loads

Table 3 Results for the OPF Algorithm

<table>
<thead>
<tr>
<th>Bus</th>
<th>Original Load (MW)</th>
<th>Amount of Load Controllable (MW)</th>
<th>Net Load after OPF (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>80</td>
<td>30</td>
<td>80</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>30</td>
<td>83.48</td>
</tr>
<tr>
<td>7</td>
<td>130</td>
<td>50</td>
<td>80</td>
</tr>
</tbody>
</table>

The bus agents (B_A) gather the data measurements from the connected bus. The regional agents (TR_A) receive the data measurements from the bus agents and used this information to run the load OPF of the entire region. For a future implementation, a larger and more extensive power network will be used to incorporate a decentralized optimization algorithm among the different transmissions regions (TR_A). In this case a solution
will be obtained by coordinating the cooperation from the agents in the different transmission regions. This type of analysis can help decide which solution is more suitable, the decentralized or the centralized approach.

4.5. Second Case Study for the OPF Algorithm

The example presented in Section 4.4 is a worst case scenario as the amount of load being controlled is significant. To illustrate a more realistic case in which control of the loads would be reasonable, the following example is presented.

Line 3-7 has a real power limit of 82.3 MW and at the moment is just above that limit with a power flow of 82.35 MW. There are some penalties to the utility if the power exceeds that real power limit constraint. As it is just a small violation of the limit, this is a problem that can be solved easily by controlling the loads of the system.

The same algorithm using the agents that was presented in Section 4.4 is performed. Now it will show how the agents would have to interact with the distribution agents (D_A). After the load control OPF is performed, the algorithm determined that the load at bus 7 has to be reduced by 140 kW. TR_A sent the request to B_A and the B_A agreed to control that amount of load, the B_A also performed a load OPF to determine which loads at the distribution network have to be controlled. For this case, the same 13-bus feeder used in the loss minimization case was implemented as the distribution network. There were two agents just as in Section 3.1, one controlling buses from 1 to 6 and the other buses 7-13. The loads to be controlled are located at buses 6, 7, 8, 9, 11 and 12 (Figure 8). The first agent controls the load at bus 6 and the rest of the loads are controlled by the second agent. The results are presented in Table 4. Again it is assumed that this result was obtained by connecting some pluggable hybrid cars that were simulated as a generator.

Table 4 Results for the Dist. Load Control OPF Case 1

<table>
<thead>
<tr>
<th>Bus</th>
<th>Original Load (kW)</th>
<th>Amount of Load Controllable (kW)</th>
<th>Net Load after OPF (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2.30</td>
<td>0.50</td>
<td>2.15</td>
</tr>
<tr>
<td>7</td>
<td>1.925</td>
<td>0.40</td>
<td>1.725</td>
</tr>
<tr>
<td>8</td>
<td>1.70</td>
<td>0.40</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>0.68</td>
<td>0.30</td>
<td>0.48</td>
</tr>
<tr>
<td>11</td>
<td>1.7</td>
<td>0.50</td>
<td>1.35</td>
</tr>
<tr>
<td>12</td>
<td>1.28</td>
<td>0.30</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Once the solution is obtained by the B_A, the results are sent to the D_A in order for them to control the load. Then each D_A sends a confirmation to the B_A after the load control is performed. The B_A sent a confirmation to the TR_A to stop the algorithm, only after all of the D_As have confirmed that loads were controlled.

Figure 8 Dist. Network with Controllable Loads

The last simulation was performed using the simulated agents in JADE integrated with a real power system simulation run in Matlab. It is important to show the implications of these algorithms on both power networks. This type of analysis was not considered in the past but certainly is going to be in the future because a detail amount of load data composition will be available.

The simulation took about 25 seconds to run for two reasons. Two different networks were simulated. Thus two different connections to Matlab are needed and there are agents in the distribution and the transmission network. Every time a B_A communicates with the TR_A, the TR_A responds to each B_A message one at a time. The messages are in a query, and once the message is addressed, another one is addressed. Each agent communication has its own message ID and each agent has its own name, so the agents can keep track to whom they are communicating with. All of this takes time to verify. For this type of algorithm each B_A communicates with the TR_A at least five times. This also is the case between the B_A and the D_A inside the distribution network. This analysis was implemented in a relatively small case, thus if these results are extrapolated to larger systems, the simulation time increases as well as the complexity of the problem. For these larger cases, it is better to simulate different regions of the grid in different computers so that each computer can parallelize the solution of the algorithm.
5. Discussion of Results and Challenges

The work presented is a brief view of what can be done with the incorporation of agent-type technologies into the power grid. The algorithms presented are decentralized. In order for them to work they require communication to exchange data in a reliable manner.

Based on the FIPA Contract Net Interaction protocol used to exchange messages among agents it can be seen that at least five messages are required. If for any reason some of these messages are lost, the communications have to start from the original step 1. The task needs to be performed in a way that would not create more problems to the grid. Thus, the protocol provides confirmation of messages as tasks are being performed. This was implemented in a simplified communication example, but the protocol could be extended to verify each message and request, incrementing the number of messages.

As was discussed in the Section 4, the decisions made in the distribution network can directly affect the transmission grid and vice versa. That is the reason a good coordination among the power networks has to be implemented. In order to handle the large amount of data needed, there would have to be some data collection and aggregation points in which data from different regions is collected and analyzed before being sent to the distribution control center. But to do that a good communication and data network that consider the cyber-security has to exist. The cyber security implications of the smart grid are enormous not only for the agent-type technologies but also for the amount of smart devices that are going to be deployed in the distribution network. This subject is currently being addressed by research groups in the US [12-13] and is a constant topic of discussion in the US Congress when they talk about the smart grid.

The work presented here is an example of what can be done with agent-based technologies in the power grid, but the challenges are even greater. For example, there is no power system software that can simulate decentralized algorithms in a way that would allow the study of agent-based technologies in an effective way. Although the JADE platform provided a good tool to simulate the distributed agents, Matlab was necessary to allowed integration of the agents to complement the study. Another challenge is the amount of computer science concepts, like networking, that the new power engineers need to know.

6. Conclusions

This paper presented two decentralized algorithm that would require the implementation of distributed intelligent agents.

The first algorithm results were found to be similar to those using a centralized algorithm. The second algorithm relieved transmission line overloads by controlling loads. The results show that this type of control is possible but would require new communication algorithms.

Addressing larger systems such as the smart grid will change the way that power engineers approach the solution to power systems problems.

In future work, the distributed agents will be implemented in larger transmission and distribution power networks in order to test a communication network.

7. References


