Agora: Distributed Tertiary Control of Distributed Resources

Koen Vanthournout
K.Univ.Leuven
Leuven, Belgium
kvanthou@esat.kuleuven.ac.be

Karel De Brabandere
K.Univ.Leuven
Leuven, Belgium
kdebrab@esat.kuleuven.ac.be

Edwin Haesen
K.Univ.Leuven
Leuven, Belgium
ehaesen@esat.kuleuven.ac.be

Jeroen Van den Keybus
K.Univ.Leuven
Leuven, Belgium
vdkeybus@esat.kuleuven.ac.be

Geert Deconinck
K.Univ.Leuven
Leuven, Belgium
gdec@esat.kuleuven.ac.be

Ronnie Belmans
K.Univ.Leuven
Leuven, Belgium
ronnie@esat.kuleuven.ac.be

Abstract - The current evolution towards more small, intelligent equipment, connected to the utility grid, poses new control challenges. Autonomous electricity networks (AEN's) are proposed here as a partial solution. An AEN is composed of a group of cooperating intelligent units within the same grid segment that operate fully distributed (without central controller), using standard components and public communication networks. To enable the construction of AEN's, a toolbox, Agora, was developed, which contains basic building blocks for AEN applications. As proof of concept, a distributed tertiary control application was implemented, for which the results are presented here.

Keywords - Distributed generation, distributed tertiary control, autonomous electricity networks

1 Introduction

The current evolutions in the electricity grid unveil, due to the liberalization of the market and the growing environmental concerns, a growing tendency towards a high penetration of small distributed (renewable) resources (DR). Furthermore, a strong increase in the number of intelligent apparatus is foreseen, e.g., intelligent loads, storage devices, measurement equipment, etc. All these apparatuses increasingly become able to communicate and interact with each other and the electricity grid and, in theory, are able to assist in the secure operation of the grid. However, it is still not clear how the integration in the network operation could be achieved, without overly complicating it [8]. It is believed by the authors that the solution to the integration of intelligent apparatus in the network operation should be found in distributed control, as it enables reliable, flexible network operation with limited complexity. It is the aim of this paper to present a suggestion to how this aim could be achieved.

The work presented here fits in our research effort to realize *autonomous electricity networks* (AEN); our answer to the above-formulated evolution. An AEN is a group of distributed generators, intelligent loads and storage devices in a realtime price market, capable of cooperation and control in a distributed manner, i.e., without central controller, and this based on standard components and public communication networks. Hence, we are developing a set of basic technologies, a toolbox, named Agora, that must permit this (see Section 2).

The tools, provided by Agora, were used to realize a distributed tertiary control, which also immediately serves as proof of concept for Agora itself. This application allows a group of generators, connected to the same low-voltage segment, to continuously converge to the economically optimal operation point where all generators operate at the same marginal cost, both in island operation or connected to the grid and taking into account pricing information (see section 3). Section 4 describes the test platform and Section 5 contains the obtained results from that setup, which validates our claims.

2 Agora: an AEN toolbox

An autonomous electricity network consists of a cooperating group of intelligent electricity devices in a low-voltage segment. Typically these are the intelligent devices in a company or in a village or city neighborhood. The requirements of an AEN are:

- The security of supply should be maximal.
- Voltage and frequency should remain within a predefined range around rated voltage and frequency.
- Cooperation and control should be distributed and thus achieved without central controller.
- It should be based on standard components and public communication networks (Internet, TCP/IP).
- Configuration and operator interaction must be minimal.
- It must be able to deal with a dynamic environment, i.e., new devices can join, leave or fail at any moment and should be handled automatically.

Agora consists of basic tools, which allow the realization of AEN applications. These tools are:

2.1 Droop Control

The aim of droop control is to control the voltage and frequency of the grid, without the need for communication between generators [2, 1]. It is thus a very reliable way to control grids, as it is unaffected by communication failures.

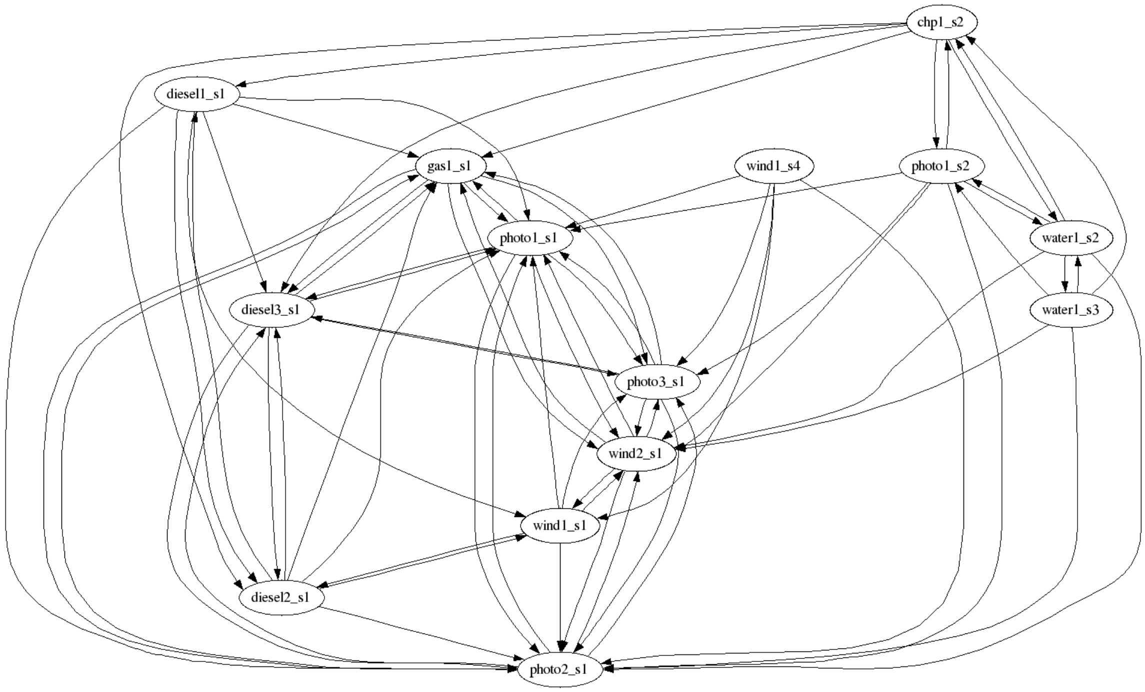

Figure 1: Snapshot of a semantic overlay network, composed of distributed generators. The '_s#' indicates the grid segment.

Droop control is in fact very similar to the primary control in utility grids [10]. However, when the primary control of distributed generators equipped with inverters is aimed, the term droop control is more often used. An inverter, equipped with a frequency and voltage droop controller, adjusts its active and reactive power output, depending on the deviation between rated and measured frequency and voltage at the point of connection with the grid, as described by the droop equations (1) and (2) and illustrated in Figure 2.

$$P - P_0 = -\frac{1}{k_p}(f - f_0) \tag{1}$$

$$Q - Q_0 = -\frac{1}{k_q}(U - U_0) \tag{2}$$

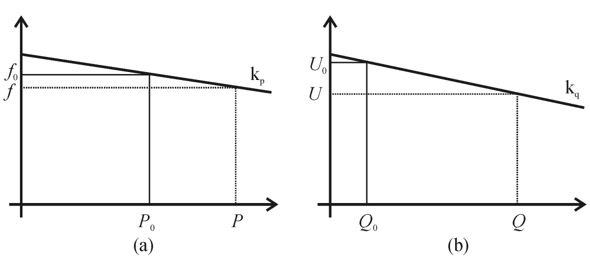

Figure 2: (a) Frequency and (b) voltage droop characteristics.

The use of droop controlled DR is today mostly confined to small island grids, but increasing penetration of DR in utility grids will inevitably lead to the increased use of droop controlled inverters. Nowadays, droop-like control of large wind turbines in utility grids already exists in countries as Germany and Denmark.

2.2 Semantic Overlay Networks

Classic control systems are master-slave systems. Yet this type of control is unsuitable for AEN's. Since an AEN is potentially composed of devices of different owners, a central controller and the accompanying responsibilities should be avoided if possible, the more since such a controller forms a single point of failure. AEN's can contain a large number of components, which makes the central controller also a performance bottleneck. Therefor, a semantic network was designed as a control basis for AEN's instead.

An overlay network is a virtual communication network, consisting of a directed graph, build on top of another network (see Figure 1). Typically, such an overlay network consists of a group of Internet-connected nodes, that have knowledge of a (small) subset of other nodes in the group (the edges in the overlay graph) and with whom they communicate. The most well-known systems that incorporate such a network are the peer-to-peer file-sharing systems, e.g., Gnutella [6], Freenet [3], Chord [9], etc. The main properties of an overlay network is that it is scalable, self-organizing and fully distributed: global properties are achieved through local interactions. Furthermore, they degrade gracefully in the advent of failures, restore automatically after repair and can cope with dynamic environments [11], which is required for AES's, as stated above. They also require only a minimum of configuration data: the address of any other active node is sufficient to add a new node to the system; from there on, the self-organization takes over.

A semantic overlay network [12], is a network in which the participants self-organize into a graph in which nodes of equal functionality are close to each other. This is achieved by comparing XML description files, e.g.:

```
<?xml version="1.0"?>
<entityDescription>
  <description>
    <IntelligentDevice>
```

```
      <deviceOwner>ELECTA</deviceOwner>
      <deviceSegment>Arenberg-ESAT</deviceSegment>
      <intelligentDeviceType>
        <electricalDevice>
          <generator>
            <windTurbine/>
          </generator>
        </electricalDevice>
      </intelligentDeviceType>
      <workingStatus>1</workingStatus>
      <ElectricalDevice>
        <maximumPowerInW>10000</maximumPowerInW>
        <minimumPowerInW>0</minimumPowerInW>
        <actualPowerInW>2500</actualPowerInW>
        <ManageableDevice>
          <ActiveManageableDevice>
           <gossipingPeriodTimeInmsec>10000</g...>
           <measurementsUpdatePeriod>3000</mea...>
           <costProfile>
            <generatorPowerData>
             <genPowerDataEntry>0</genPowerDat...>
             <genPowerDataEntry>10</genPowerDa...>
            </generatorPowerData>
            <generatorMarginalCostData>
             <genMargCostDataEntry>0</genMargC...>
             <genMargCostDataEntry>10</genMarg...>
            </generatorMarginalCostData>
           </costProfile>
          </ActiveManageableDevice>
        </ManageableDevice>
      </ElectricalDevice>
    </IntelligentDevice>
  </description>
</entityDescription>
```

Figure 1 contains the graphical representation of a snapshot of such a semantic network, built with online distributed generators. It illustrates that generators of the same type and generators within the same grid segment are close to each other.

The advantage of using a semantic overlay network as the basis of an AEN, is that it allows the construction of a communication structure, that requires little configuration, yet allows the implementation of distributed algorithms for the control and coordination of the nodes it contains.

2.3 Gossiping

Gossiping (a.k.a. epidemic algorithms), first introduced in [4], is a technique to quickly disseminate data in a (large) distributed network. The basic mechanism consists of each node periodically and randomly selecting a neighbor with which it consecutively exchanges data. This data spreads then into the network at an exponential rate. Next to spreading data swiftly, gossiping can also be used to construct aggregation algorithms [7], i.e., distributed algorithms that obtain global information on a distributed network for local nodes, without central coordination, e.g., the size of the network or, if each node proposes a value, the average of those values.

Gossiping is a powerful paradigm for distributed systems and can be the enabling technology for data-dissemination, data-collection and for distributed control algorithms in autonomous networks, which we demonstrate by using a variant of gossiping to realize distributed tertiary control (see Section 3) in an AEN.

3 Distributed Tertiary Control

Building on the Agora tools, the distributed tertiary control of a group of generators is realized. The generators must be connected to the same segment of the distribution grid and are equipped with a standard processor and an Internet connection (TCP/IP) (see Figure 4).

For simplification purposes, we assume that the set of generators is always connected to the same segment. This excludes the use of breakers within a segment, yet the assumption is justifiable, since the system here proposed is able to cope with the (frequent) joining of new generators, with their leaving and other runtime changes. As such, the system is easily extendable to a breaker-equipped segment fed by multiple transformers on the condition that a mechanism is provided that informs the generators of the current segment configuration. However, such an information mechanism is out of the scope of this proof-of-concept.

Two operation modes are possible: island operation or interconnected operation. If the segment is interconnected with the grid, we assume that a metering device is positioned at the point of coupling between the segment and the utility grid. This meter should measure the power flows into or from the considered segment and must contain pricing information. The power flow data is used to prevent line overload, the pricing information to operate at economic optimum. Pricing can be fixed and pre-set or could be up-to-date pricing information, received from an energy stock exchange. The mechanism to obtain it, however, is out of the scope of this work.

We assume that the power production of the segment is small compared to the total nominal power of the (stiff) grid. As such, the influence of the autonomous electricity network on the frequency of the grid is negligible.

3.1 Adding the Pieces

Every generator is equipped with a droop control. This ensures grid stability in island operation and enables the use of public communication networks. Only non-critical control is based upon (unreliable) Internet communication, which ensures the system's survival during communication failures. The AEN will degrade, i.e., operate at a non-optimal operation point, but will not fail.

The generators are also equipped with agent software, required to have them self-organize into a semantic overlay network. This overlay network is then used by a gossiping algorithm for distributed tertiary control.

Tertiary control enables the economic operation of the generators in the AEN, which is obtained via a marginal cost optimization of all generators [1]. If the AEN is grid-connected, the meter at the point of coupling between the AEN's segment and the utility grid is behaving as a generator (or load), with a marginal cost equal to the price of the electricity.

[1] If manageable loads would be present in the microgrid, their marginal cost functions can also incorporated in the cost optimization problem, being considered as negative generators.

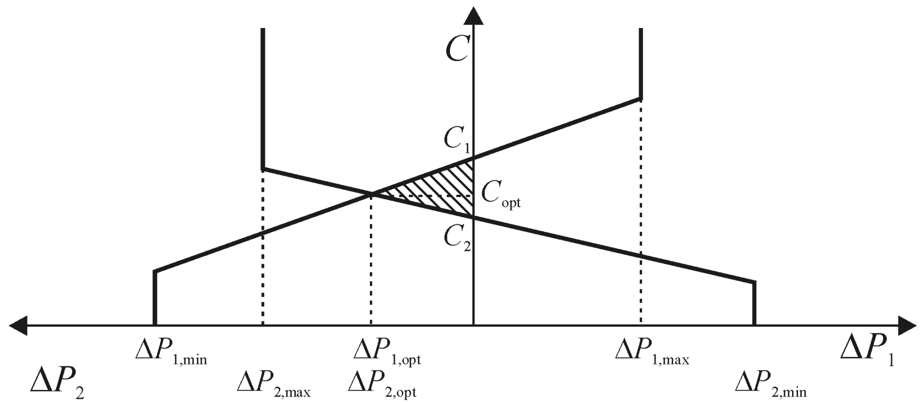

Figure 3: Marginal cost matching of two generators.

As long as the point of coupling is not at its power limits, the economic optimum is dictated by the electricity price of the power flowing through that point of coupling. All generators operate at the point where their marginal cost is equal to this electricity price. Any generator having a marginal cost lower than the power price across its whole power range, operates at maximal power output. Generators having a marginal cost being higher than the power price across its whole power range, operate at zero power output.

However, when the power flow through the point of coupling is at its limits, the local price (being equal to the marginal cost of the unconstrained generators) will be higher or lower than the electricity price at the point of coupling, depending on whether the grid acts as a generator or load to the microgrid. Also, when the microgrid is operating isolated from the utility grid, local price will be different from the electricity price of the utility grid.

In island mode, the droop control will ensure energy balance. The gossiping algorithm, on its turn, ensures that the energy demand is covered as economically as possible with the available generators.

3.2 Gossiping Based Tertiary Control

Every node periodically selects from its current neighbors in the semantic overlay network a random node that fulfils the gossiping requirements, i.e., the target node is a generator in the same grid segment. With this gossiping target a TCP connection is set up, to allow the execution of a gossiping step. A gossiping step consists of the matching of the marginal costs of the two gossiping generators in a way that the total power production of these two generators remains the same. The periodic gossiping of each node will then force the system to converge to an optimal state, i.e., all generators operate at the same marginal cost.

The matching of the marginal costs of two generators can be understood as depicted in Figure 3. The generator units exchange data and calculate a power offset based on their own data and its gossiping partner's data. Stating that the total power production of two generators has to remain equal is stating that in each gossiping step the two power offsets ΔP have to be opposite. The data that is sent is a piece of the marginal cost function, relative to the actual produced power. The marginal cost function is considered piecewise linear. The exchanged data consists of three successive line pieces or four points in x-y coordinates, i.e. eight values which have to be transmitted by each generator. The centerpiece is the power range in which the generator actually operates. If there is no left or right piece (as is the case in Figure 3), the first or fourth point coordinates will be respectively equal to the second or third point coordinates. Finding the power offset at which marginal costs are equal can graphically be interpreted as finding the intersection of two curves, one being the marginal cost function of the generator itself, the other being the marginal cost curve of the gossiping partner flipped across the Y-axis. In this way the power offset calculated by the gossiping partner will be exactly opposite. If no intersection is found, at least one of the generators is working at its maximum production or has to be turned off. The new power output at which the generator has to be regulated is the previous power output plus the calculated power offset. Each gossiping step will introduce an economic optimization gain, represented by the hatched area in Figure 3. Any other power offset will create a less optimal situation.

3.3 The Gossiping Protocol

After the initiating node of a gossiping step sets up a TCP connection with the (randomly selected) target node, both nodes execute following symmetric protocol. The symbols used are: u: the gossiping node, v: the gossiping target node, $\mathcal{P}$: the current active power, G: the set of gossiping variables (the four coordinates) and $\mathcal{O}$: the power offset for the droop control.

GOSSIPING(u, v)

1) $establishConnection(u, v)$
2) $\mathcal{P}_u = getPower(u)$
3) $G_u = calculateGossipingVars(\mathcal{P}_u)$
4) $send(G_u \rightarrow v)$
5) $receive(G_v \leftarrow v)$
6) $disconnect(u, v)$
7) $\mathcal{O}_u = calculateOffset(G_u, G_v)$
8) $setOffset(\mathcal{O}_u)$

4 Experimental Setup

The used lab setup (see Figure 5) is composed of three voltage-source inverters [5] in island operation. The inverters represent the front-end of DR units. The DR could be a generator, such as a PV system, a wind generator or a microturbine, or a storage unit, such as a battery unit or a fuel cell. In the laboratory, each inverter is fed by a DC source, being representative for a DR unit.

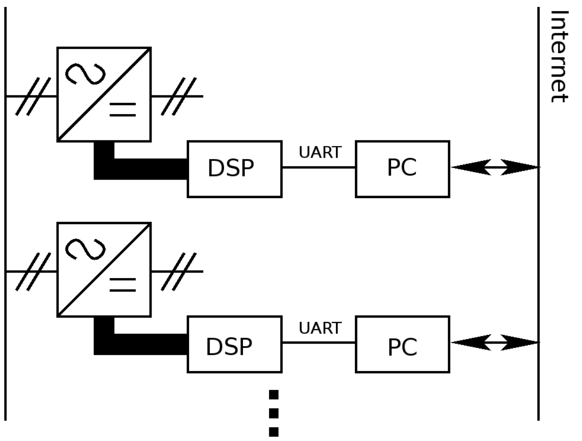

Figure 4: The experimental setup: each inverter is controlled by a DSP, which is connected via a serial line to a standard Internet-connected PC, on which the semantic overlay network and the gossiping software runs.

The inverters are controlled by digital signal processor (DSP TI C6711), running on the droop software, which is in turn connected via a UART serial line to a standard i386 architecture PC, running Linux 2.6. The semantic overlay and gossiping software is located on these PC's, which are interconnected by the Internet (see Figure 4). All Internet communication is via XML messages, to ensure interoperability, and are sent using the TCP/IP protocol. All TCP/IP operations are enhanced with a custom application-level time-out, to ensure timely detection and handling of communication faults or failed nodes [11].

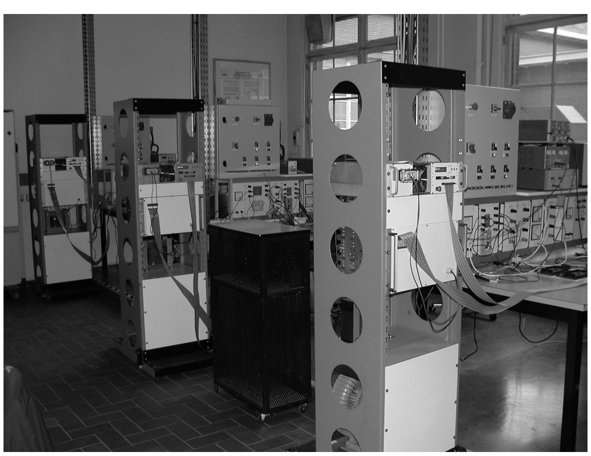

Figure 5: Picture of the three inverters.

5 Results

The concept is validated in a laboratory experiment, comprising an island grid with three inverters as generators, connected in parallel to feed a resistive load. The marginal cost of each inverter, rated at 1.25 kW, as a function of the power output is shown in Figure 6. During the experiment, the load is changed several times (see Figure 7). As a result of this load change, the frequency deviates from the rated frequency (not shown on the figure). The droop control of all three inverters react to this frequency changes by supplying either more or less power, in order to keep the balance between produced and consumed power, and to ensure that the frequency and voltage of the grid remain close to the rated frequency (50 Hz) and voltage (230 V).

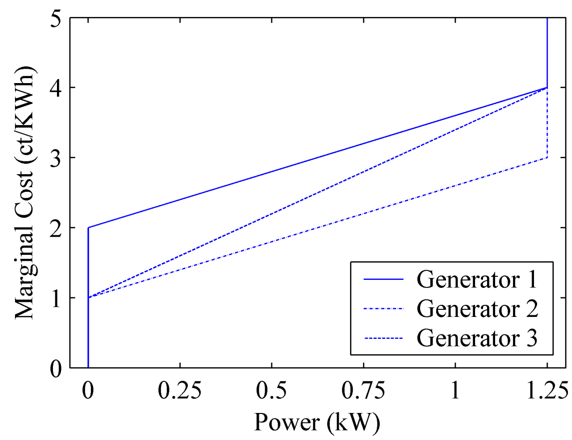

Figure 6: Marginal cost entries for the three generators.

The local control of each inverter not necessarily leads to equal marginal costs of the power supplied. Therefore, the distributed tertiary control continually optimizes the system by ensuring that the marginal costs of the inverters become equal, unless the power limits of the inverters are reached. As each node initiates a gossiping step every 3 seconds, it takes several seconds until the system settles at the new marginal cost[2].

From time = 270 seconds onwards generator 2 is operating at its upper limit of 1.25 kW. As a result, the marginal costs of the two other generators converge (after a transient) to a level higher than the marginal cost of generator 2 at maximum output. At time = 370 seconds, the load increases again, and due to the droop control all three inverters, including generator 2, react, leading to an output power of generator 2 which is slightly larger than the rated power during a few seconds. After some time, the tertiary control ensures that the overloading of generator 2 is taken over by the two other generators, despite the higher marginal costs of these generators. Also (not shown on the figure), after each load change, the initial frequency deviation disappears slowly thanks to the tertiary control actions.

[2] If this period is decreased, convergence will be faster, but at the cost of higher network load.

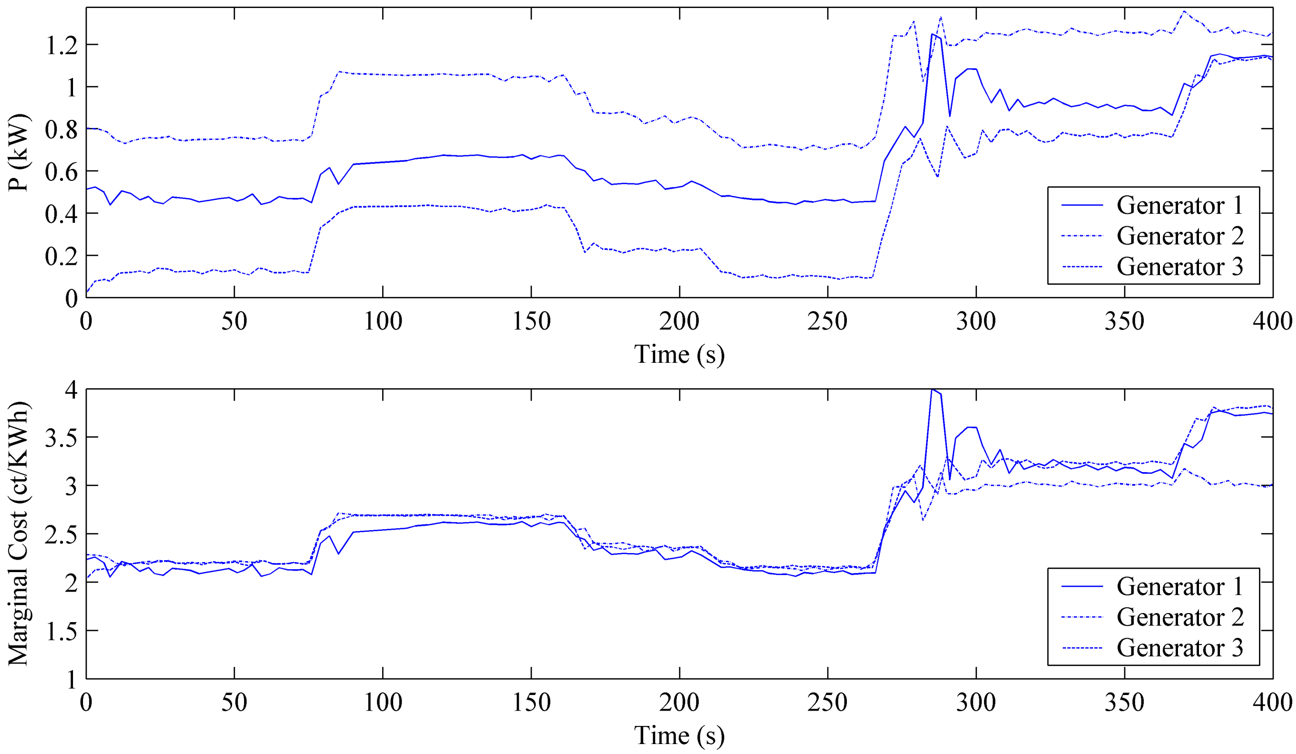

Figure 7: Actual power output and corresponding marginal costs of the generators as a function of time with changing load.

6 Conclusions and Future Work

The current traditional master-slave control systems are unlikely going to be able to cope with the large increase of small intelligent units in the future electricity grid. As an alternative, we here proposed *autonomous electricity networks*, which are composed of cooperating groups of these intelligent units within the same grid segment to achieve distributed control. To enable AEN applications, a toolbox (Agora) has been designed and implemented and to prove its validity, we used Agora to implement a distributed tertiary control algorithm, which allows groups of generators to operate at an economical optimum, i.e., they converge to the operating point where they all produce at the same marginal cost. The results of the experiments prove the effectiveness of this approach.

Future work includes further tests with larger numbers of inverters and the evaluation of the fault detection and fault handling capabilities of the AEN. Other applications that can be build, using Agora, are:

- Internet-based AEN supervision software
- power line congestion avoidance
- minimization of peak consumption
- maintenance of the power quality (voltage profile, frequency, etc.)

The realization of the supervision software and the congestion avoidance are planned, the other applications are in consideration.

Acknowledgements

The authors are grateful to the ”Instituut voor de aanmoediging van Innovatie door Wetenschap en Technologie in Vlaanderen (IWT)”, for granting a GBOU research project on embedded generation, the European Commission for support through the FP5 Dispower project (Contract No. ENK5-CT-2001-00522), the K.U.Leuven Research Council (GOA/2001/04).

REFERENCES

[1] K. De Brabandere, B. Bolsens, J. Van den Keybus, A. Woyte, J. Driesen, and R. Belmans. A voltage and frequency droop control method for parallel inverters. In *Proc. of the IEEE Power Electronics Specialists Conf. (PESC-2004)*, pages 2501–2507, Aachen, Germany, Jun 2004.

[2] M. C. Chandorkar, D. M. Divan, and R. Adapa. Control of parallel connected inverters in standalone ac supply systems. *IEEE Trans. on Industry Applications*, 29(1):136–143, Jan 1993.

[3] I. Clarke, O. Sandberg, et al. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46–66, 2001.

[4] A. Demers, D. Greene, A. Hauser, et al. Epidemic algorithms for replicated database maintenance. In *Proc. ACM Symp. on the Principles of Distributed Computing*, pages 1–12, Aug 2000.

[5] J. Van den Keybus, B. Bolsens, K. De Brabandere, and J. Driesen. Using a fully digital rapid prototype platform in grid-coupled power electronics applications. In *Proc. of the 9th IEEE Conference on Computers and Power Electronics (COMPEL 2004)*, Champaign-Urbana, USA, 10 pages, Aug 2004.

[6] Gnutella. The gnutella protocol specification. http://rfc-gnutella.sourceforge.net.

[7] M. Jelasity and A. Montresor. Epidemic-style proactive aggregation in large overlay networks. In *Proc.of the 24th Int. Conf. on Distributed Computing Systems (ICDCS'04)*, pages 102–109, Tokyo, Japan, Aug/Sep 2004. IEEE Computer Society.

[8] F. Provoost, J. M. A. Myrzik, and W. L. Kling. Setting up autonomous controlled networks. In *Proc. of the 39th Int. Universities Power Engineering Conference (UPEC 2004)*, Bristol, UK, 5 pages, Sep 2004.

[9] Ion Stoica, Robert Morris, David Liben-Nowell, et al. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, Feb 2003.

[10] UCTE. Operation handbook. http://www.ucte.org/ohb/e_default.asp, Jun 2004.

[11] Koen Vanthournout, Geert Deconinck, and Ronnie Belmans. Building dependable peer-to-peer systems. In *Supplemental volume of Int.Conf. on dependable systems and networks (DSN-2004)*, pages 297–301, Florence, Italy, Jun 2004.

[12] Koen Vanthournout, Geert Deconinck, and Ronnie Belmans. A small world overlay network for resource discovery. In *10th Int. Euro-Par Conference(Euro-Par 2004)*, Lecture notes in Computer Science Vol. 3149, pages 1068–1075, Pisa, Italy, Aug/Sep 2004. Springer, Berlin.