COMBINING DISTRIBUTED SYNCHRONIZED HIGH FREQUENCY MEASUREMENTS WITH A CONTROL SYSTEM FOR SMART LOW VOLTAGE GRIDS

Gerwin HOOGSTEEN
University of Twente – Netherlands
g.hoogsteen@utwente.nl

Marco E. T. GERARDS
University of Twente – Netherlands
m.e.t.gerards@utwente.nl

Johann L. HURINK
University of Twente – Netherlands
j.l.hurink@utwente.nl

Gerard J. M. SMIT
University of Twente – Netherlands
g.j.m.smit@utwente.nl

Omar MANSOUR
Smart State Technology – Netherlands
mansour@smartstatetechnology.nl

Dennis BIJWAARD
Smart State Technology – Netherlands
dennis@smartstatetechnology.nl

ABSTRACT

This paper presents a real-time monitoring and control system for low voltage grids built with Smart State Technology's (SST) LV-Sensors and their open platform together with the TRIANA energy management methodology. This platform uses synchronized real-time measurement data from various locations in the grid. The presented control system uses this real-time data to resolve deviations in power consumption from a given planning. Based on this solution, necessary actions can be taken to avoid grid overloading and enhance the Quality of Service (QoS) to customers. The control system, implemented using DEMKit, runs on the SST LV-Sensors modules. Initial integration tests show that the solution works stable and resolves prediction errors to ensure QoS.

INTRODUCTION

The energy transition brings various challenges to low-voltage (LV) distribution grids, such as an increased risk of over/under voltage and grid overloading. With the ongoing electrification, this stress on the LV grid is expected to increase. Especially in the Netherlands, which aims to phase out the usage of natural gas rapidly, the expected shift to electric based space heating using heat pumps will have a significant effect on the load in these LV networks. Furthermore, with the integration of electric mobility, more service interruptions are expected as shown in [1]. Upgrading existing LV networks to cope with these developments is not an economically appealing solution. Instead, we aim for a low-cost solution based on modular real-time measurement units targeted for LV grids, such as presented in [2]. An energy management system (EMS) runs on top of such modules and controls flexibility offered by end-users to perform peak shaving, and thereby avoids grid overloading and service interruptions. Possible forms of flexibility are delaying the charging of electric vehicles (EV) or preheating rooms.

In this work we propose a sophisticated EMS that handles high frequency measurement data from various locations in an LV grid. Such an EMS must balance local energy production and consumption, and at the same time uses real-time measurements to maintain power quality and resolve overloading problems. We develop such a system within the Open Real Time Development Platform for Smart Grids (ORTEP) project, by combining the TRIANA concepts [3], implemented in DEMKit [4], with the open platform and LV-Sensors from Smart State Technology (SST) [5]. An overview of this system is given in the next section. The following sections present initial integration tests and performance evaluation of running DEMKit software on the SST open platform hardware. Furthermore an evaluation of the integrated control system itself is presented. At the end we draw conclusions and present future work to improve the solution.

SYSTEM OVERVIEW

The envisioned EMS combines optimized plannings of flexibility with real-time measurements from the LV-Sensors. A newly developed control mechanism combines both information streams to control attached devices. An overview of this system is depicted in Figure 1, which will be explained in this section.

Smart State Technogy LV-Sensors

The used SST LV-Sensors provide high frequency measurement data at a sampling rate of up to 128 KHz. Two flavours of the LV-Sensors are available, one for currents and one for voltages. Multiple sensor nodes can be deployed in a grid, where the common system architecture consists of a time beacon, voltage sensors, current sensors and data aggregation. The measurement samples are synchronized in time using the GPS time beacon which broadcasts GPS pulse per second (PPS) information to LV-Sensors distributed throughout the grid using a 5.8 GHz analogue transmitter. A schematic overview of a possible setup is shown in Figure 2.

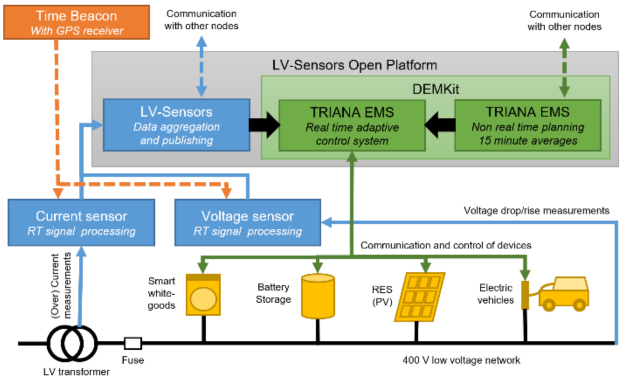

Figure 1: Overview of the proposed system for ORTEP

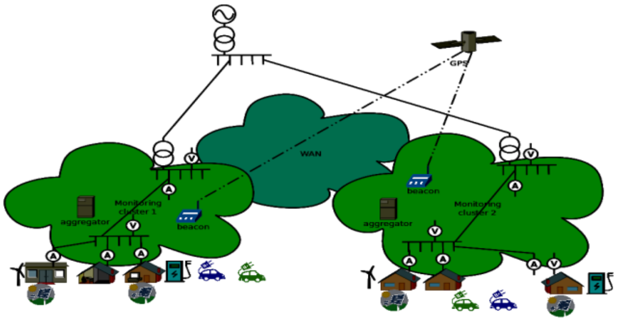

Figure 2: Example of multiple LV-Sensors in an LV network

Each LV-Sensor incorporates an ARM-based low-power embedded system running Linux. Currently, the LV-Sensors make use of an embedded Orange Pi embedded computer with and ARM processor. Although multiple variants are possible, we restrict the evaluation to an Orange Pi Zero with an Allwinner H2 Quadcore SoC and 256MB of RAM. Connected to the Orange Pi is a shield that contains an advanced multichannel ADC (with simultaneous sampling capability up to 128KHz), a digitally programmable clock oscillator, a micro-controller, and the PPS transceiver circuitry. The voltage sensors have a measurement range of 600V (230V nominal) and make use of signal transformers, while the current sensors use of split-core CTs and have a measurement range up to 600A.

Next to the provided hardware, the LV-Sensors also provide an open platform, which allows third parties to directly access real-time measurement data from the LV-Sensors, both locally and remotely. A major part of the LV-Sensors open platform concept is the DSP-framework. The DSP framework ensures that data, measurements and signal events can easily be exchanged between various sensors and aggregation units distributed throughout the grid, as well as between various algorithms. For efficient communication, the open platform makes use of two libraries: Zero Messaging Queue (ZMQ), and the structured binary format CBOR. Users of the system can run their applications directly on the hardware platform. This results in low-cost sensors suitable for LV wide area monitoring with synchrophasor based measurements and calculations suitable for e.g. remote monitoring, (dynamic) state estimation, and congestion management.

DEMKit

To implement an EMS on top of the open platform LV-Sensors, we use the DEMKit simulation and demonstration toolkit developed at the University of Twente [4]. DEMKit provides a software platform with the necessary tools for researchers to develop and validate new optimization algorithms and control mechanisms for the smart grid. The software is written in Python3 to benefit from object oriented programming, while being flexible to rapidly create prototypes of new control concepts.

A library with components is provided with DEMKit to allow the creation of such smart grid scenarios. These components include grid assets (cables and transformers), generic device classes and optimization algorithms. A scenario is a composition of multiple individual modelling components. With the modular setup, it is possible to easily create a scenario and compare various control and optimization algorithms. Furthermore, components that model the behaviour of virtual devices can be replaced by components that interact with their real world counterpart. This way, it is possible to perform hardware-in-the-loop (HIL) simulations or deploy demonstrators with the DEMKit software for validation. The main goal of this paper is to show the results of running DEMKit simulation scenarios in real life and to find bottlenecks in this approach.

Combining LV-Sensors and DEMKit

In this paper we use the TRIANA energy management methodology [3]. This methodology employs a model predictive control (MPC) approach by based on predictions of the energy profile and possible flexibility. The predictions are used as input for a mathematical optimization framework to create a near optimal planning for usage of flexibility for e.g. the next day. Such a planning often uses discrete time intervals of 15 minutes. Real-time measurements from the LV-Sensors are used to take proper control actions, given the planning and the current state of the grid.

For planning, the Profile Steering approach [4] is used. It optimizes the energy profile using a predefined desired energy profile with minimum and maximum bounds (e.g. congestion limits) as target. In an iterative fashion, it updates device profiles to minimize the Euclidean distance between the desired profile and the planned profile. Furthermore, tailor made device optimization algorithms, see e.g. [6], are used to create a planning for each individual device. The use of MPC allows the control system to make informed decisions concerning short- and long-term objectives and available flexibility. Thereby it avoids greedy use of flexibility, which potentially leads to larger problems in the future. An example is discharging a battery too greedily now, which may result in an empty battery that cannot resolve severe overloading problem in the future. Such a situation would result in a reduced Quality of Service (QoS) to end-users by either performing load-shedding or a service interruption.

Such a situation is what our solution aims to prevent. However, as predictions are never completely accurate, a mechanism to deal with prediction errors is required. Prediction errors occur in both the time domain (e.g. a cloud passing by later, affecting PV production) and the energy domain (e.g. total produced energy is lower than predicted). In [4] it is shown that an event-driven variant of Profile Steering is able to adequately deal with prediction errors from both domains. This method continuously incorporates updated predictions and performs partial re-planning to keep the overall power profile as close as possible to the original planned profiles. The Profile Steering algorithms form the green right hand block of TRIANA in Figure 1.

The LV-Sensors open platform aggregates data from multiple sensors in the grid. All incoming samples are synchronized and optionally signal processing is performed. The resulting output samples are streamed via ZMQ to the TRIANA methodology.

The left hand TRIANA block in Figure 1 concerns the online control algorithm. As events in the smart grid play on much smaller time scales than the 15 minute interval planning, we use an online control concept similar to [7]. The control mechanism continuously tries to minimize the error between the planned power consumption and the measured power consumption by updating the power output of a controllable asset. On the longer time-scale, the event-based Profile Steering algorithm makes sure that sensible trade-offs are made to keep the operation of the grid within bounds for the longer term. On the other hand, if the asset has insufficient flexibility to resolve dangerous situations, the controller does as much as it can and immediately signals other controllers for assistance. Such a signal could be used to e.g. perform load-shedding as a last resort to prevent e.g. overloading.

EXPERIMENT 1: INTEGRATION TESTS

Within the ORTEP project, we further develop the capabilities of DEMKit to operate in demonstration projects. Among those developments is the addition of ZMQ interfaces for distributed optimization across multiple control nodes. Furthermore, for the online control mechanism we use the real-time measurements from the LV-Sensors as input. Hereby, the distribution of SST LV-Sensors in a grid matches the decentralized architecture of the Profile Steering approach.

Test 1: control-loop performance

The first step to real deployment of such a system is the ability to receive the measurement data in the control system itself. As the data must be processed at a high rate, we developed a new component for the DEMKit software, which runs in a separate process. This in contrast to other components that run synchronized with the discrete time intervals used in simulations. This component reads the data signal from the LV-Sensors by subscribing to the ZMQ channels defined in the open platform specification. The online control process steers a controllable asset as quickly as possible to match the planned power value.

To test the integration, we realized the execution of the algorithm directly on the LV-Sensors hardware. For this test, we did not have a controllable asset available, so we used a modelled battery within the DEMKit simulation environment instead. The controller received real measurement data from an external SST LV-Sensors module in the grid. In this test we evaluated the amount of samples the system is able to process within an hour using the online algorithm on the system. Furthermore we evaluated how fast new control actions can be executed. The simple method just reads data and calculates the control action. In an extended version of the test we sent a control signal back to an external system (in our case a home automation platform) to check the rate at which external systems can be controlled through an HTTP REST API. The used control rule for this test is a simple feed-back loop:

$$P_t^{control} = P_t^{target} - P_t^{measured} + P_{t-1}^{control}$$

Here, the setpoint for the controlled power ($P_t^{control}$) in time interval t depends on the current *target* (the result from the offline optimization) and *measured* value from the sensors. This gives the difference that must be added to the previous control action $P_{t-1}^{control}$.

Case	CPU [%]	Memory [%]	Samples in total	Samples /second
NoComm	40-50	9.3	170856	47.46
Comm	100	9.4	152677	42.41

Table 1: Results of the control-loop performance evaluation

Profiling of the tests (Table 1) shows that the system is highly capable of performing this test. Without control to an external system (*NoComm*), it is able to process 47.46 samples per second on average out of the maximum achievable 50.00. The load on the CPU is around 40-50% and memory utilisation is around 9.3%. When sending control signals to a second system via an HTTP REST API (*Comm*), the CPU load increases to 100%. This also results to a slight decrease in handled samples. The RMSE with respect to the target is calculated to be 30.96 W. Figure 3 also shows that the system is capable of responding quickly to rapid power consumption changes by providing a smooth overall power profile in general.

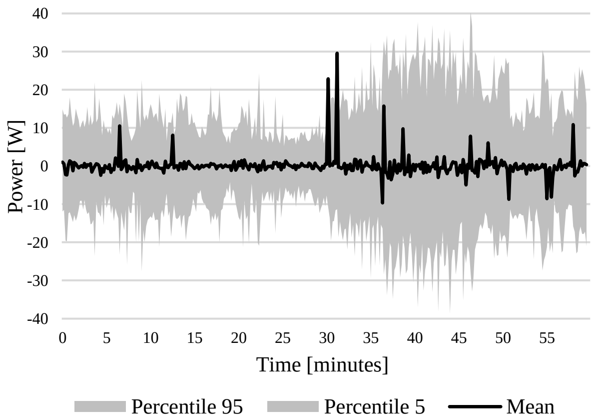

Figure 3: Resulting mean total power and 5th and 95th percentile deviation from target

We note that the test software is written in Python3 scripts, which allows for fast deployment of the concept, but leaves room for performance improvement using compiled code instead. Furthermore, it is likely that assets are also limited in the rate at which they accept control signals.

Test 2: Prediction and control algorithms performance

The second test is to run the optimization and control algorithms of a typical household on the LV-Sensors hardware. With this test we evaluate whether the hardware is powerful enough to serve as a Home Energy Management System (HEMS). A simulation scenario of a household is loaded in DEMKit to run on the LV-Sensor hardware. This scenario includes all components of a future smart household and corresponding optimization algorithms to investigate whether the hardware is able to provide the required computation power. As setup, the standard single-house scenario of DEMKit is used, which includes the following components:

- a static uncontrollable load,
- a PV panel setup,
- a controllable washing machine,
- a controllable dishwasher,
- battery storage system,
- electric vehicle,
- thermal model with a heat pump.

The simulation step-size is set to 60 seconds and one week (7 days) is simulated.

	Test 2	Test 3
CPU usage	100 %	38 %
Memory usage	10.1-10.7 %	16.4-17.0 %
Model training	9.8 s	10182.9 s
Controllers	44.1 s	35.3 s
Device sim.	157.4 s	227.4 s
Writing output	1332.3 s	3630.0 s

Table 2: System load and time spent in different functions during test 2 and test 3

The profiling results (Table 2, Test 2) show that the hardware platform is capable of acting as a HEMS. The single-threaded simulation requires between 10.1% and 10.7% of memory and takes in total 1544 seconds to complete. This leaves enough computation time for the real-time control loop and to deliver a smooth user experience. The statistics also reveal that most time (1332 seconds, 86%) is spent in storing data. The pure control algorithms require only 44 seconds (3%). In comparison, the same test requires approximately 50 seconds in total on a desktop with an Intel Core i7 3770.

Test 3: Optimization algorithms performance

The third integration test is similar to the second test. This test focusses on a use-case with real historical data to evaluate the required computation time to train the models used for predictions on the LV-Sensors hardware. Furthermore, this test is used to validate whether algorithms run fast enough to provide a seamless user-experience. For this we run the system for 10 hours using an interval length of 10 seconds. This scenario consists of the following components:

- prediction algorithms for the uncontrollable load using historical data,
- model training and prediction algorithms for the power production of a solar panel,
- optimization of a battery.

The results of running the system for 10 hours are provided in Table 2 (Test 3). From these results it is immediately clear that the algorithms to train the models using historical data are the main bottleneck. This is explained by the heavy interaction with an external database and the fact that the code of these algorithms have not been optimized yet. However, they can easily run in a separate thread. The control and device interfacing code only require modest processing time (35.3 and 227.4 seconds respectively), which is less than 1% of the discrete time interval length of 10 seconds on average. Hence, enough processing power is left to run the synchronized parts of DEMKit at interval lengths of 1s and provide the computation power for a smooth user experience.

EXPERIMENT 2: DEMONSTRATOR SETUP

Next to integrating DEMKit with the LV-Sensors, it is also important to validate that the envisioned control mechanism operates stable. The algorithms are backed by mathematical proofs and are extensively tested in various case studies [4, 6]. However, these have not been tested in a practical setting where more prediction errors may occur compared to synthetic data used in the aforementioned case studies. Furthermore, when deploying the system as a HEMS, users also interact with the system. Hence, to validate the system in a realistic environment, we have set up DEMKit as a demonstrator in a pilot household. Within this scenario all components of the proposed EMS are deployed.

For this demonstration setup, a scenario was created in DEMKit, such that it can be used to validate it before real deployment. The scenario contains the following components:

- uncontrollable load with historical data,
- PV setup with historical power and weather data,
- a virtual battery,
- a controllable (deferrable) washing machine.

Furthermore, multiple scenarios are created to evaluate the added value of the proposed solution with online-control using LV-Sensors measurements. These are:
ORTEP Control (OC): The system as presented in the paper with real-time measurements and online control.
Greedy Control (GC): No optimization used, only a greedy control strategy for the battery.
No Control (NB): Business as usual without a battery

To test the system in real, with the different scenarios in parallel, we use a micro-server (Intel Celeron J4105 with 8GB of RAM) to avoid processing power bottlenecks for this test. The simulation scenario is copied to the target system and only minimal effort was required to convert the simulation scenario into a real demonstration by replacing a few components in the scenario and adjusting some DEMKit parameters. This shows that the simulation scenario can be transferred to the real test-bed easily. One drawback of this approach is that DEMKit is still mostly single-threaded at this moment. Therefore, we reduced the control action speed by using discrete time intervals of 10 seconds to provide computing time for the (re-)planning algorithms.

To monitor its operation, the control system connected to a home automation system (Home Assistant). This automation system allows us to control a non-smart washing machine using a controllable switch between its power cord and the wall socket. For this, the power supply to the washing machine is cut off as soon as its starts its program. An interface is created in Home Assistant to set a deadline for the washing machine, such that the optimization algorithm can find the optimal start time that respects the user comfort constraints. When this optimized time is reached, the control system toggles the switch to power the machine, which then continues its operation.

Demonstrator setup results

Figure 4 shows the overall resulting profile of the three tested scenarios during the 18th and 19th of December 2018. As can be expected, **OC** results in the flattest profile, followed by **GC**. The lack of predictive control with **GC** results in an empty battery and hence a power peak, around 8:00 on the first day and 20:00 on the second. The predictive control of the proposed solution (**OC**) prevents this by leaving approximately 1kWh of energy left in the battery to be able to shave this peak. As a result, the **GC** strategy results in the same high peaks as **NB** since it cannot deliver flexibility when it matters. These worst-case power values for the three cases are given in Table 3.

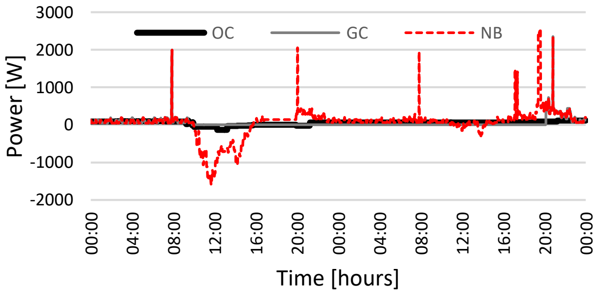

Figure 4: Final power consumption for three scenarios

	OC	GC	NB
Max. power [W]	122	2352	2555
Min. power [W]	-132	-1	-1590

Table 3: Worst case power consumption

Looking at the control actions used with **OC** (Figure 5), it is clear that, despite forecasting errors in the PV production, the control mechanism is able to closely follow the day-ahead planning (planning). Most notable is the event-based adaption of the planning around 13:00 on the first day. At this point new weather predictions indicated a lower irradiation forecast. This results in a higher intra-day planning (realization) with a higher target power. As a result, the control system makes sure that enough energy is stored to perform peak-shaving during the evening peak the next day. The measured profile (with virtual battery) matches the intra-day planning accurately.

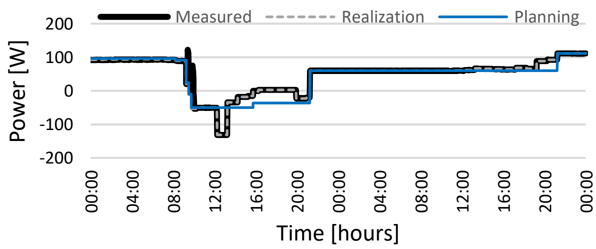

Figure 5: Control actions with the OC case

CONCLUSIONS AND OUTLOOK

In this paper we presented a smart distributed monitoring solution based on the SST LV-Sensors open platform. Connected to this platform is the TRIANA methodology for smart grids. The presented approach safeguards the operation of future low-voltage grids to guarantee a high quality of service to end-users while mitigating the effects of further electrification. For the implementation of the control mechanism, we used the DEMKit simulation and demonstration toolkit. The integration tests show that the LV-Sensors hardware delivers the required computational performance for the proposed solution. Furthermore, the total solution is capable of processing the real-time measurements adequately. A live test of the proposed control algorithm shows that the system runs stable and adapts the control setpoints adequately to prevent peaks in the power usage. Furthermore deploying a DEMKit scenario in reality, as well as integrating it with and onto the SST LV-Sensors open platform, did not cause major hurdles.

Some aspects remain to be researched and implemented for a full deployment of the presented system. Firstly, we need to extend the DEMKit software with multi-threading capabilities. The current, simulation oriented, single-threaded implementation hinders the user experience as interaction is blocked during the optimization process. The challenge here is to implement a solution that works well in both simulations and real-world deployment to minimize the time-to-deployment and ensures feedback from demonstrators is available for future simulations. For the offline planning, including the expected accuracy of the predictions is considered to increase the robustness of the resulting planning. Finally, the real-time control loop may be extended with some filtering algorithm to avoid that short term phenomena affect the online-control system in a negative way.

ACKNOWLEDGMENTS

The authors thank Dutch national program TKI iDeego (project ORTEP) and RVO for their support.

REFERENCES

[1] G. Hoogsteen, A. Molderink, J.L. Hurink, G.J.M. Smit, B. Kootstra and F. Schuring, 2017, "Charging electric vehicles, baking pizzas, and melting a fuse in Lochem", *Proceedings CIRED conference*, vol. 2017, 1629-1633

[2] O. Mansour, F. Baldinger, T. Jansen, 2011, “Challenges in smart distribution grids”, *Proceedings CIRED conference*, vol. 2011

[3] A. Molderink, V. Bakker, M.G.C. Bosman, J. L. Hurink and G.J.M. Smit, 2010, "Management and Control of Domestic Smart Grid Technology," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, 109-119

[4] G. Hoogsteen, 2017, “A cyber-physical systems perspective on decentralized energy management”, *PhD dissertation*, University of Twente

[5] O. Mansour and D. Bijwaard, 2018, www.smartstatetechnology.nl

[6] T. van der Klauw, 2017, “Decentralized energy management with profile steering: resource allocation problems in energy management”, *PhD dissertation*, University of Twente

[7] M.H.H. Schoot Uiterkamp, M.E.T. Gerards and J.L. Hurink, 2018. “Fill-level prediction in online valley-filling algorithms for electric vehicle charging”, *IEEE Innovative Smart Grid Technologies*, vol 2018, 1-6