edgeFLEX

D1.2

Dynamic-phasor driven voltage control concept for current VPPs in large scale deployment deliverable

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 883710.

Project Name	edgeFLEX
Contractual Delivery Date:	31.03.2021
Actual Delivery Date:	30.03.2021
Author(s):	RWTH – UNIBO
Workpackage:	WP1 – Dynamic-phasor Driven Voltage Control Concepts for Dynamically Controlled VPP Solutions
Security:	P
Nature:	R
Version:	V1.0
Total number of pages:	33

Abstract
The Deliverable describes the voltage control algorithm that has been delivered to WP4 as docker container for integration into the edgeFLEX architecture. The algorithm is based on an online voltage control solution which can be used to control Distributed Generators (DGs) and Energy Storage Systems (ESSs) installed in a distribution grid.

Keyword list
Voltage control, Online control, Power flow, MQTT, VPP

Executive Summary

Improving the ability of Distribution System Operators (DSOs) to control the voltage in distribution grids allows them to improve the performance of the distribution network management and reduce the need to replace and upgrade physical assets such as transformers and cables. The implementation of distributed voltage control also leverages the utilization of existing Distributed Generators (DGs). The growing penetration of renewables will allow the DSO to perform a coordinated voltage control in which all available DGs are involved in solving local voltage problems.

The objective of WP1 is to develop voltage control software tools able to counteract voltage violations in the distribution grid. This report provides the description of the voltage algorithm implemented for the first software drop of the project and for integration into the edgeFLEX platform. The document provides preliminary results of simulations demonstrating the positive benefits of the voltage control scenario. Therefore, the application of the control service leads to a reduced amount of over-voltages and a reduction of the curtailment of the renewables, enabling a higher penetration of renewables without jeopardising system stability.

Authors

Partner	Name	e-mail
RWTH		
	Edoardo De Din	ededin@eonerc.rwth-aachen.de
	Gianluca Lipari	glipari@eonerc.rwth-aachen.de
UNIBO		
	Alessandro Mingotti	alessandro.mingotti2@unibo.it
	Roberto Tinarelli	roberto.tinarelli3@unibo.it
	Lorenzo Peretto	lorenzo.peretto@unibo.it
	Gaetano Pasini	gaetano.pasini@unibo.it
	Diego Cavaliere	diego.cavaliere2@unibo.it

Table of Contents

1. Introduction

This report is based on the result of the work done in Task 1.2, which deals with the definition and implementation of a voltage control that uses measurements received from measurement devices able to provide data in phasor configurations.

The edgeFLEX project adopts a two phased approach: Phase 1 is from month 1 to month 12 (March 2021) of the project and Phase 2 which extends from month 13 to month 36 (March 2023). This report describes the work carried out in the first phase of the project in the scope of WP1, which focuses on the development of phasor-driven Voltage Control concepts. The aim of the work in Phase 1 was to have a Minimum Viable Product (MVP) for the Voltage Control service in a standalone implementation, ready to be implemented into the edgeFLEX platform.

The voltage control algorithm described in this report represents the first drop (see Figure 1) of the control strategy to regulate the voltage in distribution grids to be developed in the frame of edgeFLEX.

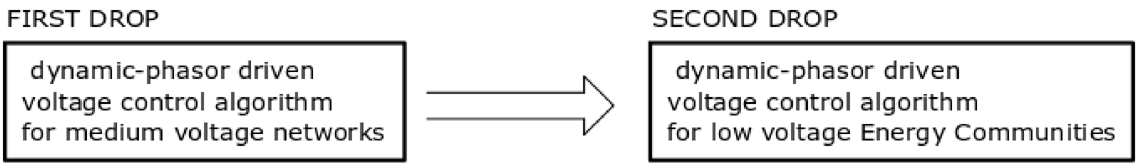

Figure 1 First and Second drops of the voltage control algorithms

In phase 2, through a series of iterative revisions via the edgeFLEX improvement model described in [1], the work of WP1 will focus on the improvement of the performances of the voltage control algorithm presented in this report and on further developing it, taking into consideration the specific requirements and needs of voltage control in low voltage networks, which will then constitute the second drop of the proposed algorithm.

The control algorithm developed for the project is a control strategy based on online measurements. That means that the control uses actual measurement values to react to sudden voltage variations. The purpose of the algorithm is to maintain the voltage within the limits defined by the DSOs by controlling the available distributed generators (DGs) installed in the electrical grid.

The control strategy developed in Task 1.2 is part of the whole edgeFLEX solution, and has to be integrated in the edgeFLEX architecture, which it is described in D4.1 [1]. Therefore, the requirements of the Task 1.2 were not only to develop the algorithm but also to realize a control able to be interfaced with the rest of the edgeFLEX platform. The control has been developed with communication interface and coupled with a power flow solver to verify the algorithm in a simulated environment.

1.1 Objective of the report

The report covers two main objectives:

- First, it describes the voltage control algorithm and the validation on a 40-nodes medium-voltage (MV) distribution grid.
- then it describes how the control algorithm has been implemented as docker container and how the communication interface has been realized in the code.

1.2 Outline of the report

The report consists of three main parts.

- Chapter 2 briefly describes the state of the art of voltage control, differentiating between offline and online solutions.
- The following chapter 3 describes the theory and the algorithm implementation of the control in edgeFLEX, as well as the results of the control tested on a 40-nodes grid where different configurations and features are highlighted.
- Chapter 4 describes the docker implementation of the control and the interface towards the edgeFLEX architecture, describing how the proposed algorithm interacts with such architecture and its components.

Finally, chapter 5 summarizes the conclusions of the report and describes the future work.

1.3 How to Read this Document

This report can be read as a standalone document. However, other deliverables can be helpful to get a better view of the concepts advanced in the edgeFLEX project and to have more details on the edgeFLEX platform. In particular, other deliverables closely related to this one are:

- D1.1 - Scenario description for dynamic-phasor driven voltage control for VPPs (M12) [2]: This deliverable starts with an overview of the state of the art in terms of VPPs and of services developed within their framework. Moreover, D1.1 introduces to the reader the dynamic-phasor driven voltage control scenario and the technical terminology adopted through the entire edgeFLEX project.
- D4.1 - Description of edgeFLEX platform design (M12) [1]: This deliverable describes the overall edgeFLEX platform where the voltage control is placed as service.

2. State of the Art of Voltage Control for distribution grid

To operate distribution grids properly, the voltages of the nodes should remain in the limits defined by the grid codes [3]. However, the growing penetration of DGs is affecting the voltage profile, increasing voltage magnitude along the nodes.

The impact of DG power injections on the electrical grid is described in [4], with the following approximated expression:

$$V = V_0 + R\,\Delta p + X\,\Delta q \tag{1}$$

Where V is the voltage magnitude of the nodes, V_0 is the voltage of the slack bus, Δp is the difference between generation and consumption in active power, Δq is the difference between generation and consumption in reactive power, which are multiplied respectively by R (real part of the impedance matrix) and X (imaginary part of the impedance matrix).

Distribution System Operators have the responsibility of maintaining the voltage of the grid within safe operational limits. To do so they can to leverage on both active and reactive power, bot generated or consumed. In this section two different control approaches to perform the control of the voltage on a distribution grid are presented.

2.1 Offline control

This approach consists in calculating the optimal solution for the electrical system (in terms of power injections and voltage control), based on load and generation forecasts on day-ahead basis. The control does not use available measurements from the distribution grid and it relies only on profiles calculated with a certain level of accuracy by means of Artificial Neural Networks or Machine Learning tools [5].

Although offline algorithms have been used in traditional power system applications, they may become not adequate in applications where high number of energy resources are installed in a large distribution network. In this condition, the fluctuations of load and generation profiles can lead to violations of the operational constraints [6] [7]. Uncertainty in generation and load can also cause deviations from the expected working conditions. Therefore, a schedule created with an offline-optimization is prone to provide not optimal solution under real circumstances.

2.2 Online control (Real-time)

To overcome the errors that are caused by load and generation uncertainties, recent works have proposed on-line feedback controllers, where system measurements are iteratively used to calculate the control output to drive the distribution grid towards an optimal solution as described in Figure 2.

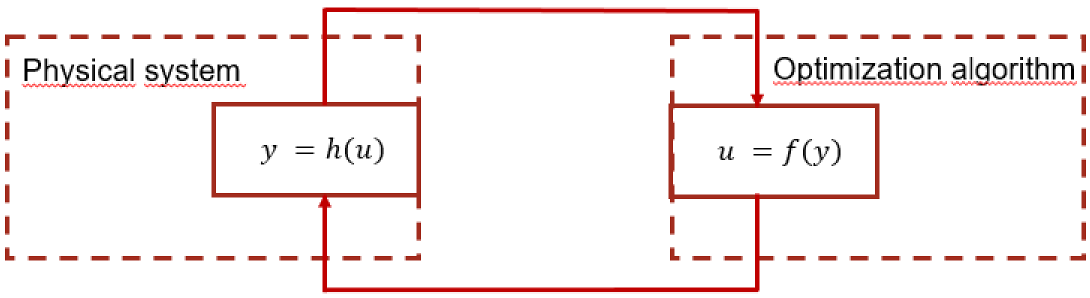

Figure 2 Online control concept

Centralized online optimization has been proposed to solve the optimal power flow (OPF) cost function, where the OPF problem is static [8] [9] or time-varying [10]. The centralized approaches require a communication infrastructure to collect measurements and transmit the control outputs for a portion of a distribution grid. In addition, being voltage control a localized problem, distributed solutions can be also used to solve the task of voltage control. In this case, controllable units communicate with a limited number of devices.

We refer to Centralized/Distributed Controllers if the controllers exchange communication messages or to Decentralized Controllers if no information is exchanged as shown in Figure 3.

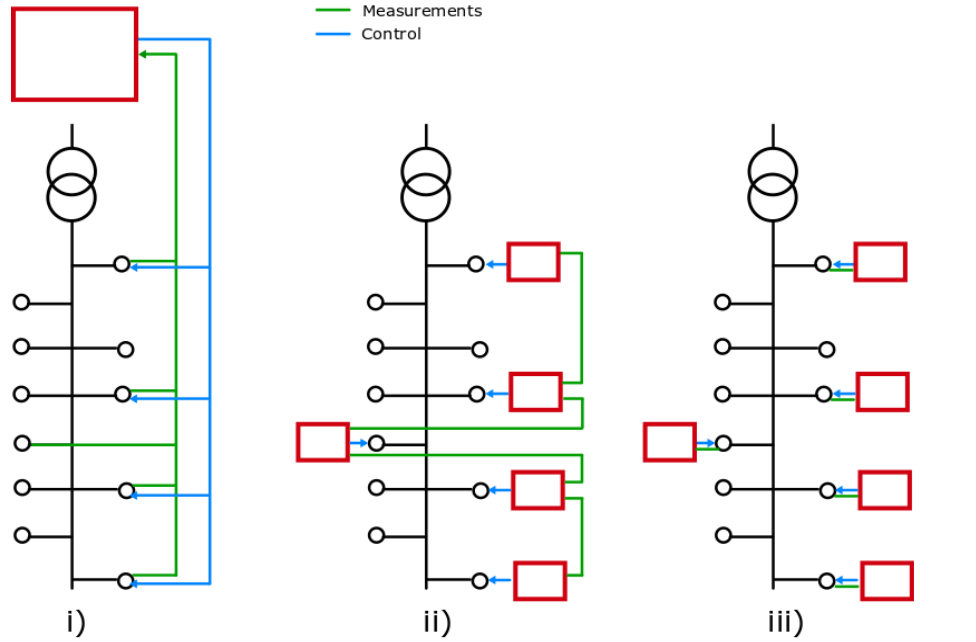

Figure 3 Different online control configurations: i) centralized ii) decentralized iii) distributed

Decentralized controllers are required to be applied to PV systems in MV systems. The control strategy measures the voltage of the point of connection of the installed PV with the main grid and it calculates reactive power output based on a local feedback law. Although decentralized strategies have been already included in German distribution grid code [3], documents [11] and [12] have demonstrated that the decentralized strategy might underperform when compared with centralized or distributed approaches. In purely decentralized approaches, system conditions and control output constraints can lead to a non-optimal solution for the system that can eventually lead to the inability to maintain the voltage in the limits. Therefore, from a DSO perspective, the application of a voltage control on a centralized or distributed configuration (a system level solution) is more attractive than the decentralized (local solution), given that it produces a solution that is global.

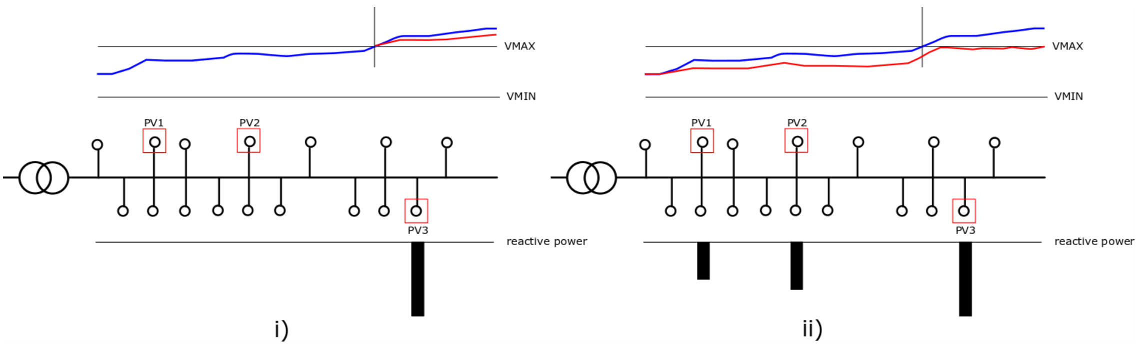

Figure 4 Comparison between i) local solution ii) system level solution

A visual representation of this concept is described in Figure 4, where the local solution is compared with a system level solution. In the first case, the voltage control produces a reactive power set-point only on PV3, because is the only one where the overvoltage happens. In the second case, the voltage control coordinates all the installed PVs to contribute to decrease the overall voltage on the feeder, solving then overvoltage problem.

3. Dynamic-phasor driven voltage control

For the development of the edgeFLEX Voltage Control algorithm we have chosen to adopt a coordinated online control approach, as described in the previous chapter, where real time measurements coming from field devices are used to identify voltage violations in a distribution network and to define control setpoints for the controllable assets in the field. The control strategy is based on a centralized system level solution that solves an online optimization.

The dynamic-phasor driven voltage control refers to the voltage control algorithm described in Section 3.1, where the voltage measurements of the nodes of the grid are phasors. In the scope of edgeFLEX project a new concept of Phasor Measurement Unit (PMU) has been developed. The edgePMU, as described in [1], has been developed during Phase 1 of the project and will be used in edgeFLEX as data source for the voltage control algorithm. The edgePMUs will provide phasor measurements of the voltage with a high reporting rate. Therefore, the voltage control algorithm has been developed considering the availability of voltage magnitude values obtained by phasors measurements.

3.1 Algorithm Description

The objective of the control algorithm is to minimize the injections of active and reactive power, coming from the Distributed Generators (DGs) and Energy Storage Systems (ESSs), that are required to keep the voltage within the limits. The contributions of active and reactive power are weighted with matrices R and X shown in (2), which are related to the ability of the grid nodes to contribute to the voltage magnitude reduction. The control algorithm considers the regulation of both the active and reactive power for optimizing the voltage. This relies on the fact that, in the distribution system, active and reactive power can have an impact on the resulting grid voltage, due to the similar order of magnitude of resistances and reactances of the lines.

The implementation of the control, developed for the first drop of the algorithm, is based on the dual-ascent method, a tool for solving convex optimization problems based on dual decomposition theory. This choice is also supported by [6], where it is stated that approaches decomposing centralized voltage optimization problems into local subproblems that need to be coordinated through communication have quite high computation and communication load.

The contribution of the active and reactive powers is expressed in squared terms, so that a quadratic formulation and therefore convex can be obtained.

$$J = \Delta p^{ESS^T} R \, \Delta p^{ESS} + \Delta p^{DG^T} R \, \Delta p^{DG} + \Delta q^{DG^T} X \, \Delta q^{DG} \tag{2}$$

Where Δp^{DG} indicates the curtailment of active power generation and Δq^{DG} the increase of absorbed/injected reactive power at the generation nodes.

Therefore, the optimization problem is formulated as follows:

$$\begin{aligned} &\min \Delta p^{ESS^T} R \, \Delta p^{ESS} + \Delta p^{DG^T} R \, \Delta p^{DG} + \Delta q^{DG^T} X \, \Delta q^{DG} \\ &s.t \quad \Delta p^{ESS}_{MIN} \le \Delta p^{ESS} \le \Delta p^{ESS}_{MAX} \\ &\qquad \Delta p^{DG}_{MIN} \le \Delta p^{DG} \le \Delta p^{DG}_{MAX} \\ &\qquad \Delta q^{DG}_{MIN} \le \Delta q^{DG} \le \Delta q^{DG}_{MAX} \\ &\qquad v_{MIN} \le v \le v_{MAX} \end{aligned} \tag{3}$$

where v is the vector of voltage magnitudes of the phasor measurements of the nodes of the grid.

Since there is no explicit interaction between ESSs and the DG, the solution of the minimization problem can be decoupled into three separate problems:

- $\Delta v(\Delta\, p^{ESS})$ minimization:

$$\begin{aligned} &\min \Delta\, p^{ESS^T} R\, \Delta\, p^{ESS} \\ s.t \quad &\Delta\, p_{MIN}^{ESS} \le \Delta\, p^{ESS} \le \Delta\, p_{MAX}^{ESS} \\ &v_{MIN} \le v \le v_{MAX} \end{aligned} \tag{4}$$

- $\Delta v(\Delta\, p^{DG})$ minimization:

$$\begin{aligned} &\min \Delta\, p^{DG^T} R\, \Delta\, p^{DG} \\ s.t \quad &\Delta\, p_{MIN}^{DG} \le \Delta\, p^{DG} \le \Delta\, p_{MAX}^{DG} \\ &v_{MIN} \le v \le v_{MAX} \end{aligned} \tag{5}$$

- $\Delta v(\Delta\, q^{DG})$ minimization:

$$\begin{aligned} &\min \Delta\, q^{DG^T} X\, \Delta\, q^{DG} \\ s.t \quad &\Delta\, q_{MIN}^{DG} \le \Delta\, q^{DG} \le \Delta\, q_{MAX}^{DG} \\ &v_{MIN} \le v \le v_{MAX} \end{aligned} \tag{6}$$

The process to obtain the feedback control output for the ESSs is based on the well-known duality theory [13] and takes inspiration from the optimization method presented in [12] for the control of the DG reactive power.

To solve the optimization problem previously defined, that has been divided into three subproblems ($\min \Delta\, p^{ESS^T} R\, \Delta\, p^{ESS}$, $\min \Delta\, p^{DG^T} R\, \Delta\, p^{DG}$, $\min \Delta\, q^{DG^T} X\, \Delta\, q^{DG}$), a distributed method based on dual decomposition and dual ascent gradient has been applied [13] [12].

The Lagrangian of the problem is defined in [4] where the Lagrangian multipliers are vectors associated to the voltage and power constraints. From the theory of duality, the algorithm is based on the iterative execution of the following steps:

- Dual-ascent steps on the dual variables associated to voltage constraints.
- Dual-ascent steps on the dual variables associated to power constraints.
- Unconstrained minimization on the primal variable.

The Lagrangian multipliers are updated separately for each node, therefore the complete solution of the system has to be carried out only for the unconstrained minimization problem.

The unconstrained minimization can be thus expressed by calculating $\nabla_x L$ where x is the controllable variable and L the lagrangian of the problem.

By applying some algebraic manipulation, the final vector of set-points can be obtained in a distributed fashion, given that the calculation depends on a matrix that has a particular sparse structure that, for each considered node, brings dependencies only from its neighbouring nodes.

This matrix multiplies the difference between Lagrangian multipliers of relative neighbours, presenting however a possible drawback, since the updated values of the multipliers should be exchanged at each internal iteration of the solution of the primal variable. To overcome this problem, the algorithm has been modified to get updated multipliers from the neighbours only when a new set of measurements is received (and therefore when the algorithm is called), keeping them constant throughout the internal iterations of steps 2) and 3). In this way, the number of times that the multipliers are exchanged can be considerably reduced.

3.2 edgeFLEX Voltage Control algorithm implementation

The algorithm has been implemented in Python and it is composed of different functions, dedicated to different aspects of the control application.

The function **system_info** reads from a file the topology information of the grid. The function is called when the control algorithm is started, since the topology of the grid is supposed to remain the same. The result of the function is a dictionary containing all the grid data (see A.1.1).

To initialize the control algorithm, a function **initialize_control** is used to calculate the matrix R and X and the coefficients of the algorithm. All these values depend on the nodes that are actively participating to the voltage control, meaning that the function must be called (and the algorithm re-initialized) every time the active nodes change. The same approach is also used for initialization of the ESSs, where the location and number of active ESSs are provided to the function that initialize the control A.1.2.

The function **initialize_control** uses as input the result of **system_info** and the list active_nodes and active_ESS, as described in Figure 5 and Figure 6.

After the initialization, the algorithm reads the voltage of the active nodes and of the active ESSs and their available apparent power (pv_input), which, in case of voltage remaining in the limit, is supposed to be all injected as active power to the grid. The function is called in the main file as in A.1.3, where the function **control_** includes all the control functions solving the problems defined in Section 3.1. The description of the function is in A.1.4.

The control calculates the active and reactive power set-points each time a new set of measurements is received. The structure of the code is defined in Figure 5 and Figure 6.

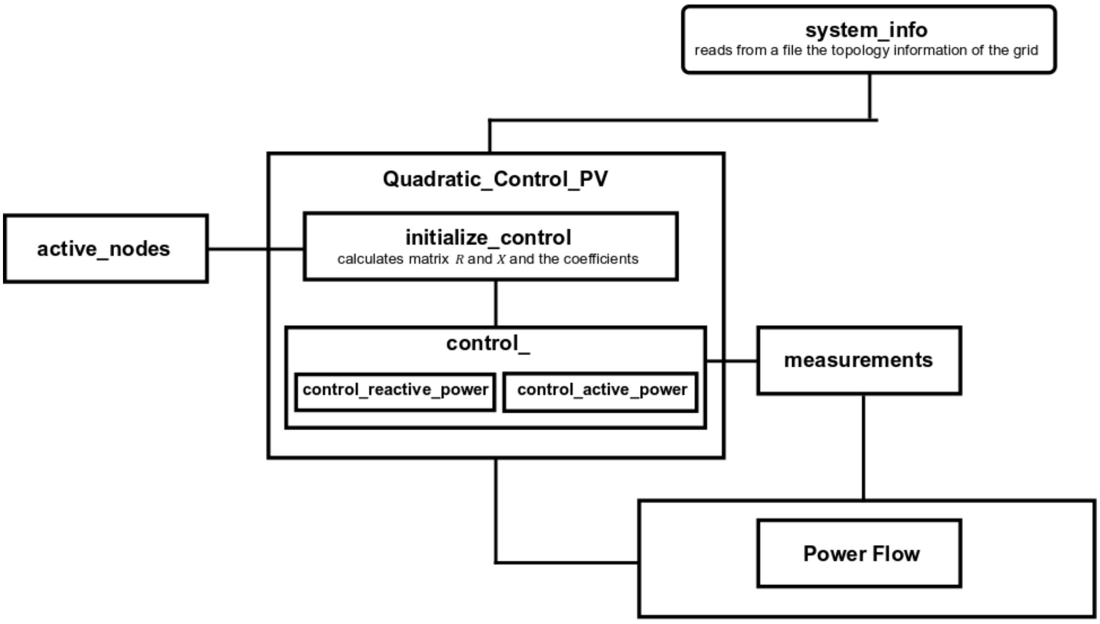

Figure 5 PVs control algorithm implementation

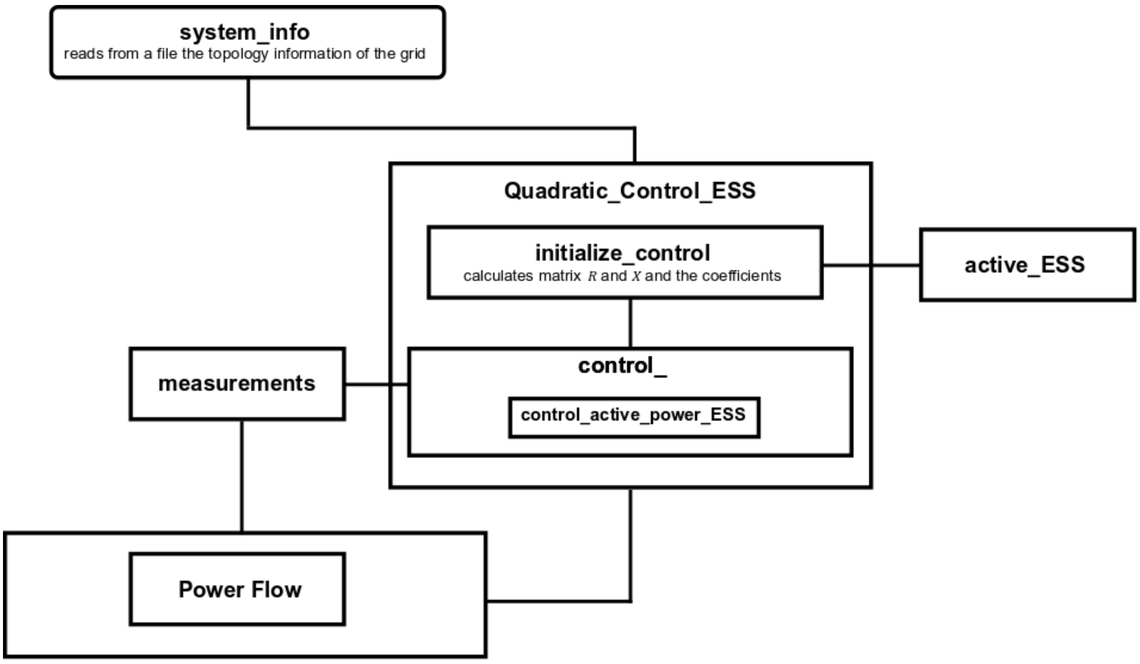

Figure 6 ESSs control algorithm implementation

3.3 Preliminary results

This section shows some preliminary results of the application of the proposed control strategy obtained in the lab. The purpose of this section is to demonstrate the ability of the control solution to compensate for some voltage violations. The results presented in the following sections refer to the use cases defined in deliverable D1.1 [2]:

- Use Case 1: Impact of DGs penetration in creating overvoltages
- Use Case 2: Mixed control with PVs and ESSs
- Use Case 3: Dynamic change of the devices under control

3.3.1 Simulation Setup

The simulation setup described in Figure 7 has been used for obtaining the preliminary results reported in this deliverable. The voltage control service, running on a docker container, exchanges measurements and set-points with the power flow simulation (the grid simulator). At the same time, simulation data is stored in a database (influxDB) for the grafana visualization. The detailed description of the implementation of the voltage control service into the edgeFLEX architecture is reported in D4.1 [1].

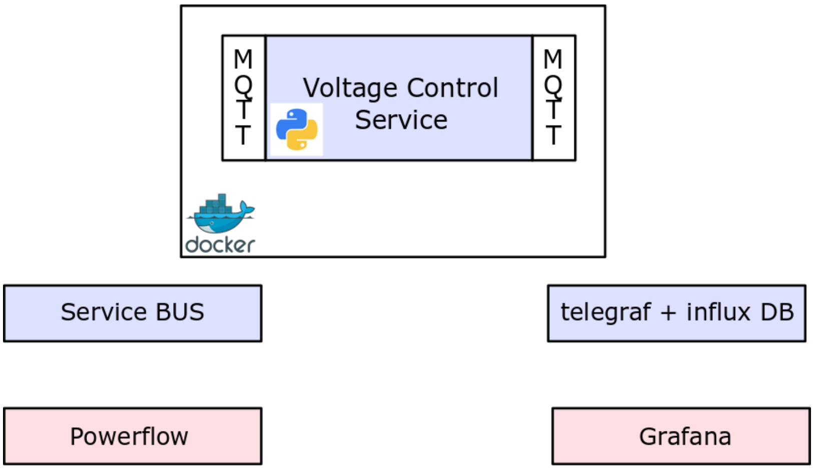

Figure 7 Simulation Setup

3.3.2 Grid under test

The grid under test is a 40-node MV distribution grid, where on each node PV and ESS are installed and are available for the voltage control. The grid is shown in Figure 8.

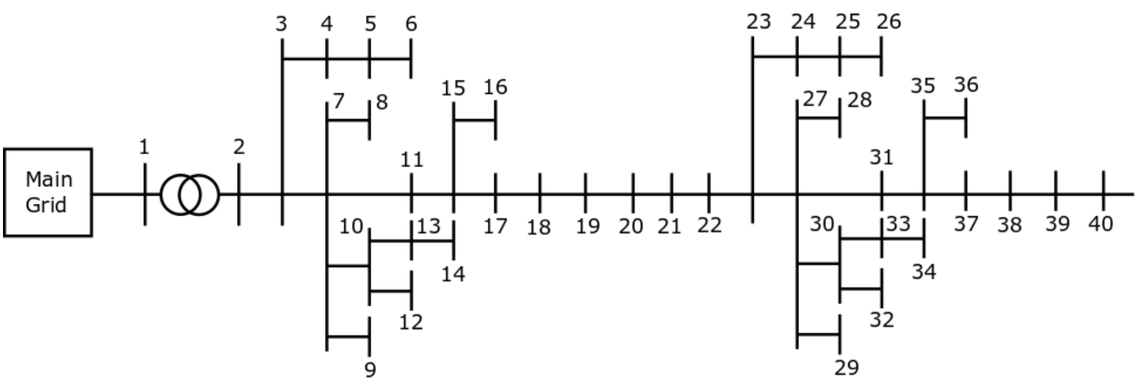

Figure 8 40 Nodes MV distribution grid

The nominal voltage for the gird under test is $V_{nom} = 20\ kV$. The limits considered in these tests are based on the [14], meaning that the voltage should remain between $\pm 10\ \%$ of the nominal value.

The load and generation profiles utilized in the simulations are shown in Figure 9 and Figure 10, respectively.

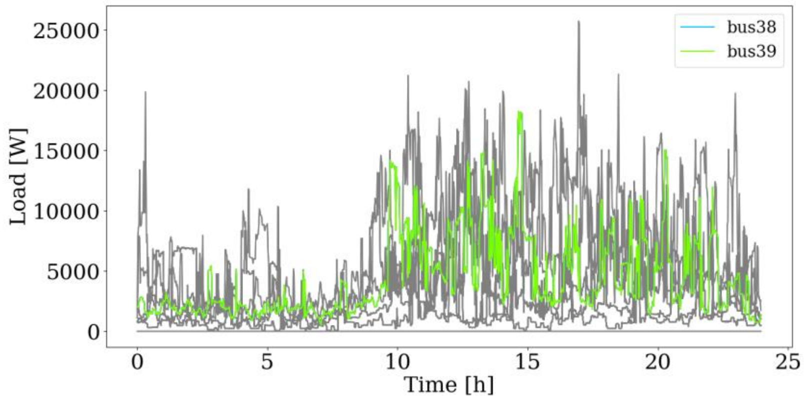

Figure 9 Load profile during 24 hours

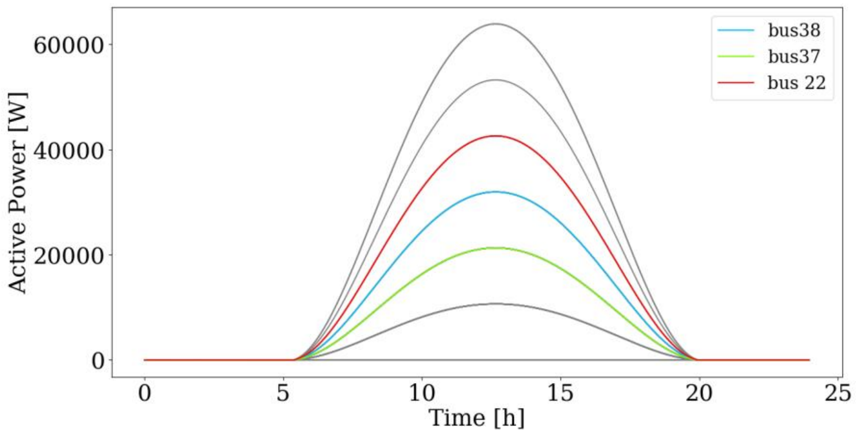

Figure 10 Generation profile during 24 hours

3.3.3 Use Case 1: Impact of DGs penetration in creating overvoltages

This use case considers a system where only PVs are installed and their power injection is not controlled. In this condition, the prosumers cannot directly participate to the voltage control. This is an extreme scenario and rather not realistic, because some new PVs have internal control to limit power injection and also because they automatically disconnect if needed. However it represents a limit case useful as a reference to assess the benefits of the voltage control service.

Therefore, this test is used to demonstrate how large is the voltage fluctuation that the generation and load profiles applied to the grid can produce.

The test performs the simulation of 24 hours with a different percentage of DGs penetration. To achieve that, the total amount of DGs generation is reduced as follows:

- 100 % reduction to obtain a 0 % DGs penetration;
- 50 % reduction to obtain 50 % DGs penetration;

- 0 % reduction to obtain 100 % DGs penetration.

KPI1 is calculated as:

$$\max_{n \in N} \frac{\sum_k^H \{k \mid V_n[k] > V_{limit}\}}{\sum_k^H k} \qquad (7)$$

Where N are the total nodes of the grid, H is the length of the 24 hours in terms of samples k.This KPI quantifies the percentage of time, over 24 hours, in which the voltage exceeds the safety operational limits. Higher values of KPI1 correspond to more severe instability of the grid.

3.3.3.1 Test 0 % DGs penetration:

The first test is performed with 0 % of DGs penetration, meaning that there are no power injections from the DGs. The result is a power profile that for most of the time is below the nominal voltage but far from the lower limit. Therefore, the resulting KPI1 is 0 %.

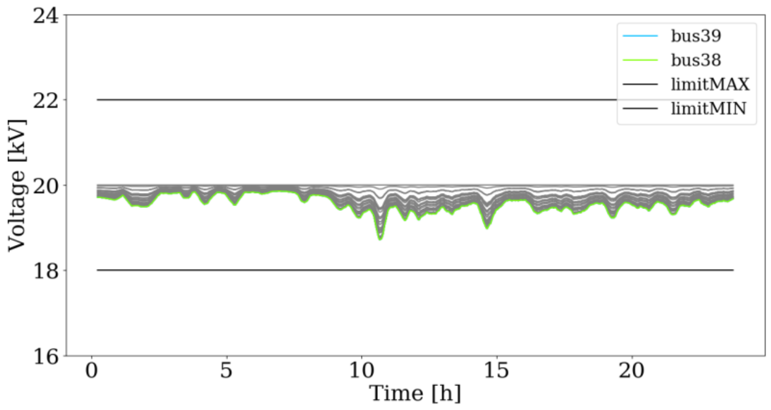

Figure 11 Voltage profile with 0 % DGs penetration

3.3.3.2 Test 50 % DGs penetration:

The second test is performed with 50 % of DGs penetration, meaning that the power injections of the DGs are half of the maximum available. The result in Figure 12 is a power profile that for most of the time is over the nominal voltage but far from the upper limit. Therefore, the resulting KPI1 is 0 %.

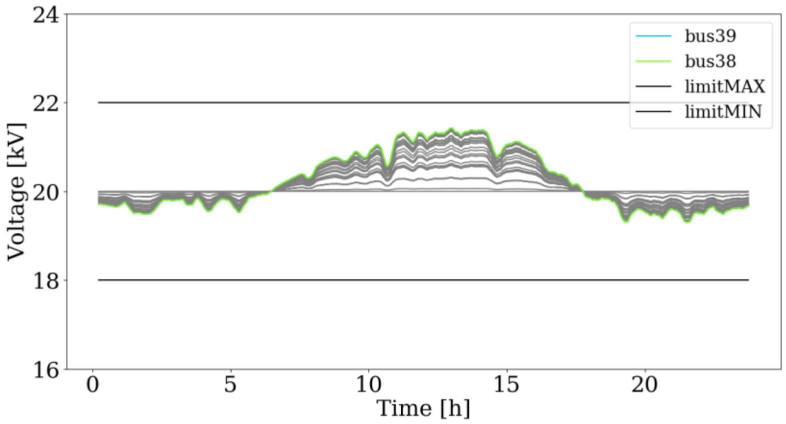

Figure 12 Voltage profile with 50 % DGs penetration

3.3.3.3 Test 100 % DGs penetration:

Figure 13 shows the voltage profiles when the voltage control is not applied on a 100 % DGs penetration, highlighting that during the generation peak the voltage of many nodes is over the limits. Therefore, the resulting KPI1 is 26 %.

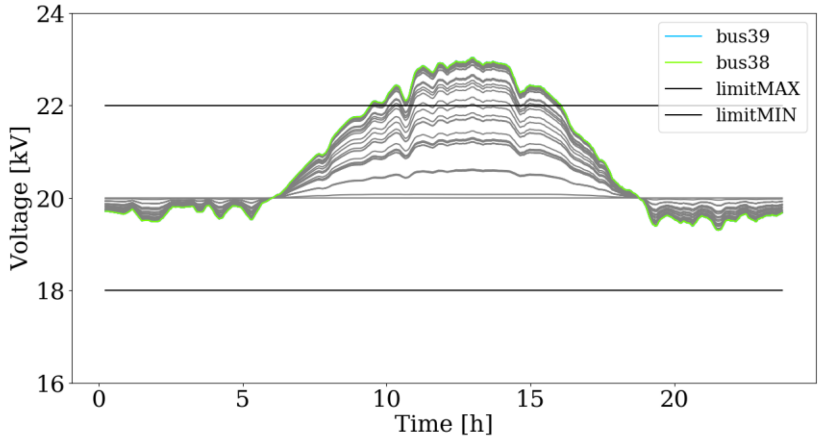

Figure 13 Voltage profile with 100 % DGs penetration

3.3.4 Use Case 2: Mixed control with PVs and ESSs

In this second use case we consider the application of the voltage control algorithms in three different configurations:

- Only PV curtailment
- PV curtailment and reactive power
- With Energy Storage Systems

The objective of this use case is to demonstrate the beneficial impact of using the reactive power control of the PVs and active power control of the ESSs in reducing the amount of active power curtailments. The following KPIs also described in D1.1 are used:

KPI2 is calculated as:

$$\begin{cases} \max\limits_{n \in N} \frac{\sum_k^H \left[P_{PV}^{curtailed}[k] - P_{PV}^{max}[k]\right]}{\sum_k^H P_{PV}^{max}[k]} & if \quad P_{PV}^{max}[k] \neq 0 \\ 0 & if \quad P_{PV}^{max}[k] = 0 \end{cases} \tag{8}$$

Which calculates the amount of active power curtailment in relation to maximum available power.

Additionally, another KPI, **KPI3,** is defined as the number of PVs to which the active power curtailment is applied.

3.3.4.1 Only PV curtailment

In the first configuration the active power curtailment is applied on many of the installed PVs, resulting in the active power injections described in Figure 14. Figure 15 shows the voltage profiles, which are kept below the limits by the voltage control. The maximum active power curtailment, which represents the constraints for the **control_active_power** function, is calculated each time step as:

$$P_{curtailment,max} = -P_{available} \tag{9}$$

Equation (9) implies that the active power generation can be curtailed until the extreme case of reducing it to zero.

The resulting KPIs are:

- **KPI2** = -52 %
- **KPI3** = 13

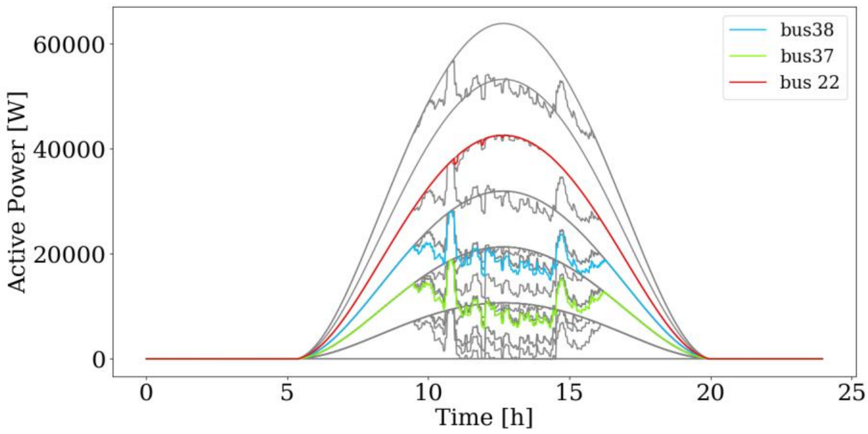

Figure 14 Active power generated by PVs after curtailment

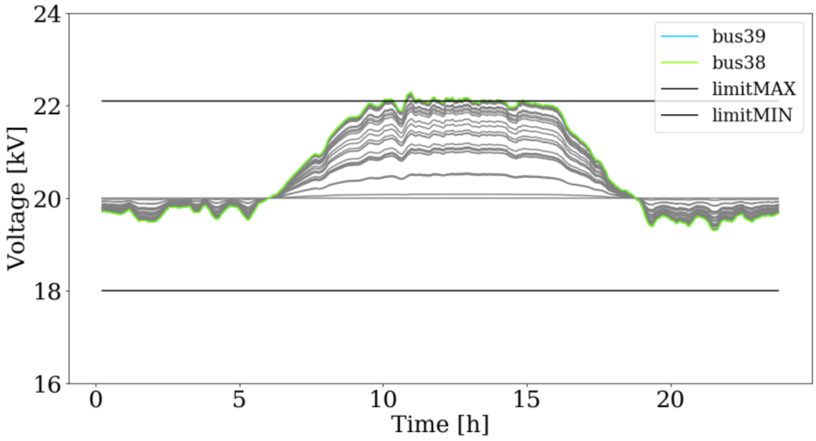

Figure 15 Voltage profile with voltage control applied

It can be observed that in this test case more than half of the available power generation is curtailed, thus lost, to keep the voltage within the desired operational limits. Additionally, 13 PVs have to participate in the active power curtailment to reduce the voltage magnitude.

3.3.4.2 PV curtailment and reactive power

The second configuration consists of a system where only PVs are installed but their power injection is controlled by the algorithm. In this case, the prosumers can participate to the regulation of the voltage by receiving set-points of reactive power and active power curtailment. The maximum value of reactive power depends on the available generated power, such that the $cos\phi$ is always equal or greater than 0.95. Therefore, the maximum available reactive power, which represents the constraints for the **control_reactive_power** function, is calculated each time step as:

$$Q_{max} = \sin(\phi_{max}) \cdot P_{available} \tag{10}$$

Figure 16 shows the reactive power absorption of the PVs for performing voltage control, which results in a reduction of the active power curtailment (Figure 17) while keeping the voltage below the limits (Figure 18).

The resulting KPIs are:

- **KPI2** = -34 %
- **KPI3** = 9

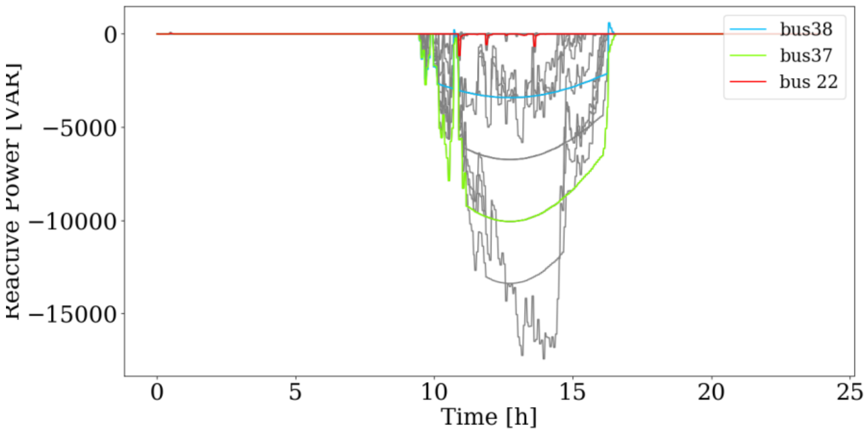

Figure 16 Reactive power set-points for installed PVs

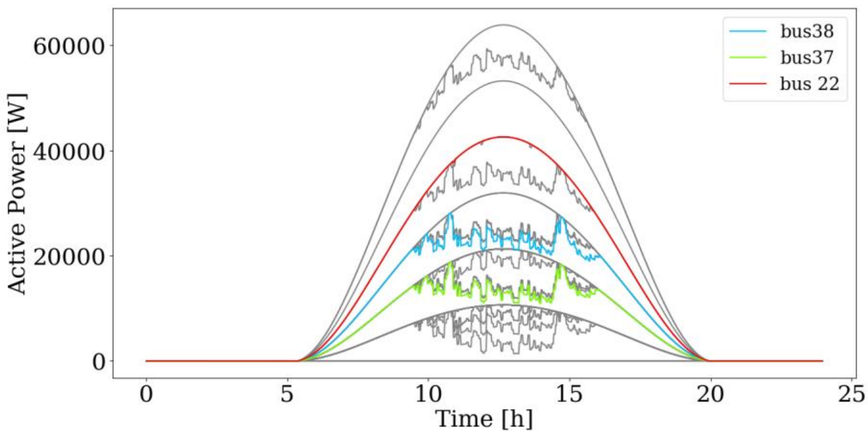

Figure 17 Active power generated by PVs after curtailment

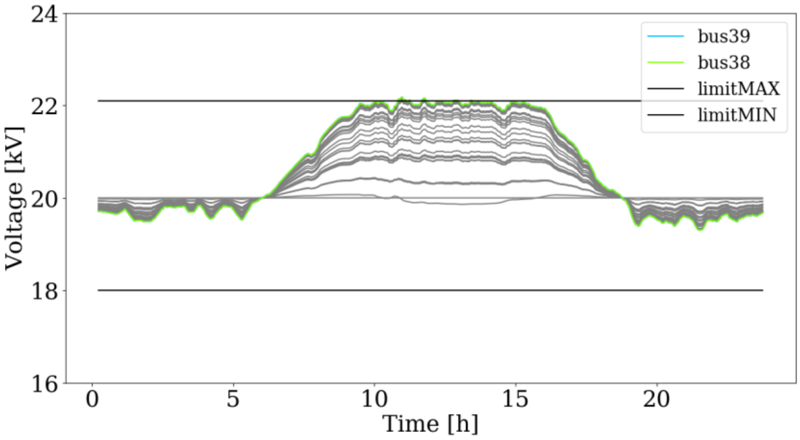

Figure 18 Voltage profile with voltage control applied

With respect to the previous test case, it can be noted that the total amount of curtailed active power is reduced to just 34%, thanks to the usage of both active power curtailment and reactive power regulation in the control strategy. Moreover, only 9 PV need to participate in the voltage control.

3.3.4.3 With Energy Storage Systems

In the third configuration the control of the voltage is also performed with the ESSs. The maximum available active power of the ESS that are the constraints for the **control_active_power_ESS** function is calculated each time step as:

$$P_{min}^{ESS} < P^{ESS} < P_{max}^{ESS} \tag{11}$$

The effect of the control of the ESSs in Figure 19 in combination with the control of the reactive power injections clearly reduce the amount of active power curtailment as described in Figure 20.

The resulting KPIs are:

- **KPI2** = -30 %
- **KPI3** = 6

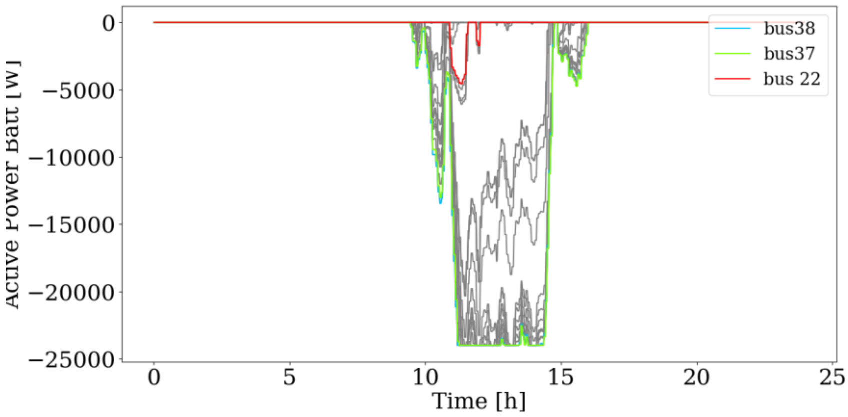

Figure 19 Active power set-points for installed ESSs

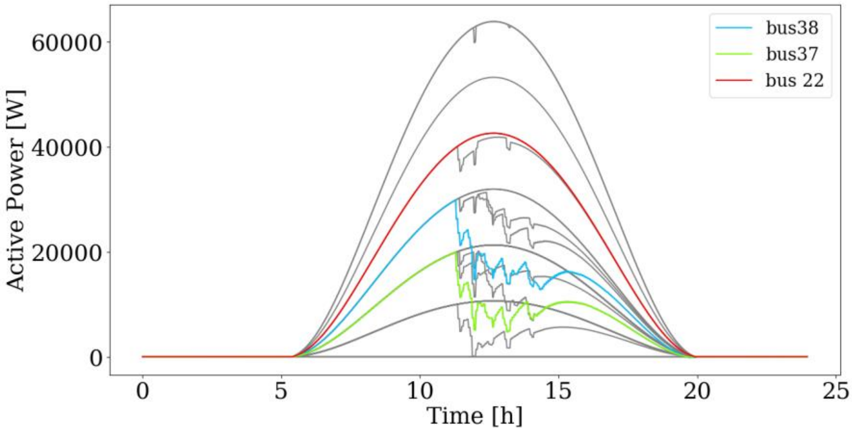

Figure 20 Active power generated by PVs after curtailment

In this case we can notice a further reduction of the curtailed active power from PV, thanks to the usage of local storage systems. More importantly, the number of involved PV systems is further reduced, showing how this solution can better identify which assets should participate in the voltage control and maximize their impact in the execution of the desired control action.

3.3.5 Dynamic change of active nodes

Another interesting aspect linked with the implementation of the proposed voltage control is the possibility of dynamically change the nodes that contribute to the voltage control. By doing so, the algorithm can dynamically adapt to any changes in the grid topology and in the availability of each controllable resource. If, for example, any node gets disconnected from the grid or any controllable resource becomes unavailable, the algorithm automatically detects the change in the topology and availability of resources and adapts to it by recalculating the control setpoints to be given to all the remaining assets.

As described in Figure 5 and Figure 6, the initialize functions receive as input the number of nodes participating in the control (active nodes for PVs and active ESS for ESSs).It is an input for the control algorithm itself and it is sent as JSON message to the algorithm via REST API.

```
reqData = {
   "data": {
      "nodes": [4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,31,32],
      “ESS_nodes”: [4,5,6,7,8,9,10,11,12,13,14,15,16,17,20,21,22,31,32],
   }
}
```

Figure 21 shows the effect of changing the number of nodes participating in the voltage control during the simulation. At the instant highlighted by the yellow box, the message above is sent to the controller and therefore Node 33 and 34 are disconnected. This determines that Node 31 and

32 increase the amount of reactive power absorption to compensate for the reduction of the controllable devices.

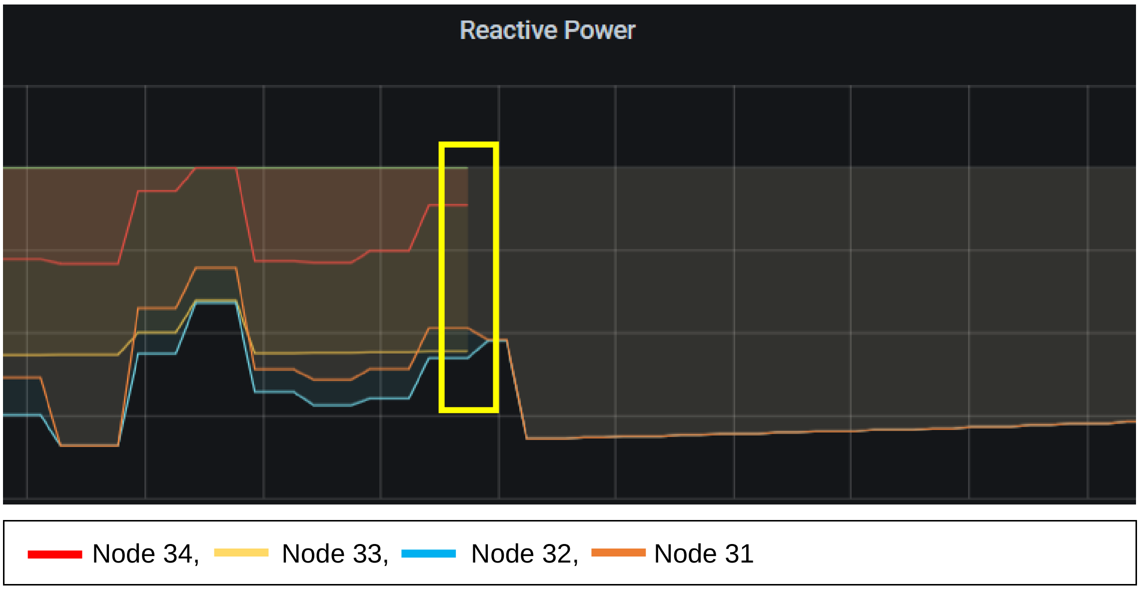

Figure 21 Disconnection of Node 33 and Node 34

4. Interaction with edgeFLEX Architecture

The Voltage control described in Section 3 is part of an architecture to be developed in the project. Therefore, the algorithm has been organised and structured to facilitate the integration in the platform. Therefore to the algorithm has been added an MQTT interface and it has the ability to be instantiated as docker container. Moreover, the voltage control has been organised as git repository to facilitate the development and integration.

4.1 Voltage control as a service

To implement the control in the edgeFLEX architecture, the code has been organised in a git repository to facilitate the development between WP1 and WP4 and to simplify the integration in the architecture.

The repository is composed of

- Control_Main.py: This is the main control file
- Cases: The folder contains grid data of test-grids
- Control_strategies: The folder contains the different functions that are called in Control_Main.py
- Setup: This folder contains all the files used to set the virtual environment
- Submodules: This folder contains some git submodules that are automatically pulled when cloning with `--recursive`.

To implement the control in a docker container, a docker-compose file is defined to execute all the required steps to run the algorithm. For example, in the docker-compose are defined all the bash commands to activate a Python virtual environment where all the required packages are defined in the folder Setup.

The voltage control service is part of the edgeFLEX architecture and it needs to be interfaced with the rest of its components. The communication protocol that has been selected to exchange data between the voltage control service and the architecture is MQTT, which in case of the service, it has been implemented by means of the dmu [15] submodule and it is described in A.2.1.

4.2 Powerflow service

To test the voltage control, a powerflow service has been implemented based on PYPOWER [16]. The service simulates the real electrical grid, receives set-points for the generators and send measurements for the voltage control service.

The repository is composed of

- runPF.py: This is the main file for the simulation
- Cases: The folder contains grid data of test-grids
- Data: The folder contains different load and generation profiles
- Setup: This folder contains all the files used to set the virtual environment
- Telegraf: This folder contains the configuration file for telegraf
- Submodules: This folder contains some git submodules that are automatically pulled when cloning with `--recursive`.

The functions that in runPF.py are used to calculate the power flow are:

- Initialize: read the grid data and the result of the function is a dictionary containing all the grid data.
- run_Power_Flow: is the function calculating the power flow based on the set-points received by the voltage control and based on the value of load and generation for each node extracted from the profile data.

As for the voltage control, the dmu [15] submodule has been used to implement the communication via mqtt. To perform a visualization useful for the preliminary tests via Grafana, the telegraf software has been used. It acts as interface between the mqtt publisher of the powerflow service and influxDB. Once the data are forwarded to InluxDB, Grafana can directly query a specific set of data for the visualization.

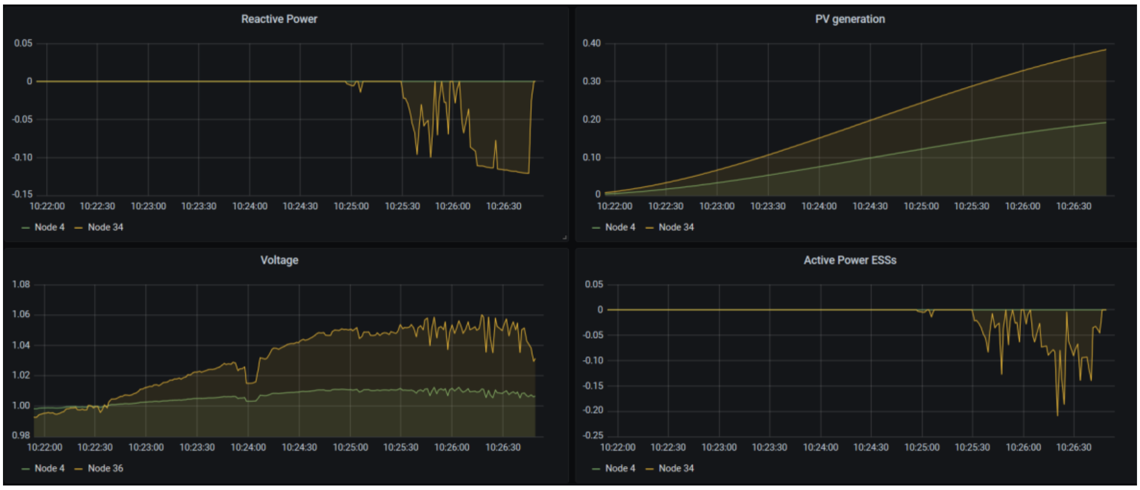

Figure 22 Grafana Dashboard

4.3 Requirements of Voltage Control as a service

- **Objective:** Control of the voltage with resources owned by the DSO.
- **Timeframe:** The control is based on measurements of voltage with a reporting rate of few minutes (or less). If power measurements are not available with the same rate, they could be received with a timeframe of 10 minutes.
- **Voltage Measurements:** Synchronized measurements with time stamps are needed for the voltage control.
- **Accuracy of Voltage Measurements:** The PMUs and power measurement units must have accuracy and granularity that are high enough to capture the fast changes in voltage (within few minute).
- **Manner:** The voltage can be executed at the substation of the DSO.
- **Data Recordings:** Data storage of voltage measurements is not required but it can be useful for the DSO to record voltage profiles to have a better understanding of the grid.

5. Conclusion

The deliverable describes the algorithm that has been implemented as a service for the edgeFLEX platform. The algorithm takes advantage of the positive aspects of the online control strategies, described in Section 2.2, resulting in a very suitable solution for reacting to fast dynamics.

The optimization problem is a quadratic equation that tries to minimize the amount of active or reactive power injections required to maintain the voltage in the limits. The main problem is then divided into three subproblems to reduce the complexity of the solution.

Section 3.3 presents some preliminary results that describe the positive impact on the control of the voltage. The algorithm has been tested on a medium voltage distribution grid consisting of 40 nodes. Some realistic load and generation profiles have been used to test the grid on a realistic scenario, where the profiles have high fluctuations linked to the weather conditions and the typical customers' load profiles. The results show the set-points that are calculated by the control and transmit to the DGs, highlighting that the use of the ESSs can reduce the amount of curtailment of the active power injections of the PVs.

For the integration of the algorithm as a service in the edgeFLEX architecture, the control has been included in a docker container. Section 4 describes how the repository has been organised to facilitate the integration in the architecture and to facilitate the exchange with WP4. Together with the voltage control, WP1 has also provided a docker container with the powerflow service that contains the power flow solver simulating the grid and the communication interface to exchange messages with the voltage control.

During the second phase of edgeFLEX project, WP1 will focus on the improvement of the performances of the voltage control algorithm presented in this report and on further developing it. The continuous development effort will aim, on one hand, at improving the performances and facilitate the integration into the edgeFLEX platform, especially for its deployment in the trials. On the other hand, the development effort in WP1 will address the second drop of the voltage control algorithm, taking into consideration the specific requirements and needs of voltage control in low voltage networks.

6. List of Figures

7. References

[1] H2020 edgeFLEX, “D4.1: Description of EdgeFLEX platform design,” https://www.edgeflex-h2020.eu/progress/work-packages.html.

[2] H2020 edgeFLEX, “D1.1: Scenario description for dynamic-phasor driven voltage control for VPPs,” https://www.edgeflex-h2020.eu/progress/work-packages.html.

[3] Verband der Elektrotechnik Elektronik Informationstechnik, “Generators connected to the low-voltage distribution network: technical requirements for the connection to and parallel operation with low-voltage distribution networks,” *VDE Ref. VDE-AR-N 4105.,* 2008.

[4] E. De Din, M. Pau, F. Ponci and Monti, “A Coordinated Voltage Control for Overvoltage Mitigation in LV Distribution Grids,” *Energies,* 2020.

[5] A. A. Mamun, M. Sohel, N. Mohammad, M. S. H. Sunny, D. R. Dipta and E. Hossain, “"A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models," in IEEE Access, vol. 8, pp. 134911-134939, 2020, doi: 10.1109/ACCESS.,” 2020.

[6] D.K.Molzahn, F.Dörfler, H.Sandberg, S.H.Low, S.Chakrabarti, R.Baldick and J.Lavaei, “Asurvey of distributed optimization and control algorithms for electric power systems”. In: IEEE Transactions on Smart Grid 8.6, pp. 2941–2962.,” 2017.

[7] M. Todescato, J. W. Simpson-Porco, F. Dörfler, R. Carli and F. Bullo, “Online distributed voltage stress minimization by optimal feedback reactive power control,” *IEEE Transactions on Control of Network Systems 5.3 (2017),* p. 1467–1478, 2017.

[8] A. Hauswirth and et al., “"Projected gradient descent on Riemannian manifolds with applications to online power system optimization." 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2016.,” 2016.

[9] L. Gan and S. H. Low, “"An online gradient algorithm for optimal power flow on radial networks." IEEE Journal on Selected Areas in Communications 34.3 (2016): 625-638.,” 2016.

[10] E. Dall'Anese, S. V. Dhople and G. B. Giannakis, “"Photovoltaic inverter controllers seeking AC optimal power flow solutions." IEEE Transactions on power systems 31.4 (2015): 2809-2823.,” 2015.

[11] P. N. Vovos and et. al, “Centralized and distributed voltage control: Impact on distributed generation penetration,” *IEEE Transactions on power systems 22.1,* pp. 476-483, 2007.

[12] S. Bolognani and et. al, “"On the need for communication for voltage regulation of power distribution grids." IEEE Transactions on Control of Network Systems 6.3 (2019): 1111-1123.,” 2019.

[13] D. Bertsekas, “Nonlinear programming,” *J. Operat. Res. Society 1997,48, 334,* 1997.

[14] CENELEC, Voltage Characteristics of Electricity Supplied by Public Electricity Networks EN 50160, 2010.

[15] “DMU,” [Online]. Available: https://git.rwth-aachen.de/acs/public/automation/modules/dmu.

[16] “PYPOWER,” [Online]. Available: https://pypi.org/project/PYPOWER/.

8. List of Abbreviations

API	Application programming interface
CENELEC	European Committee for Electrotechnical Standardization
DB	Database
DG	Distributed Generator
DSO	Distribution System Operator
ESS	Energy Storage System
IEEE	Institute of Electrical and Electronics Engineers
KPI	Key Performance Indicator
LV	Low Voltage
MQTT	Message Queue Telemetry Transport
MV	Medium Voltage
MVP	Minimum Viable Product
OPF	Optimal Power Flow
PF	Power Flow
PMU	Phasor Measurement Unit
PV	Photovoltaic
REST	Representational State Transfer
VPP	Virtual Power Plant
WP	Work Package

Code references

A.1 Code related to chapter 3

A.1.1

```
grid_data = {"baseMVA":baseMVA,"branch":branch,"pcc":pcc,"bus":bus,"gen":gen,
    "nb":nb,"ng":ng,"nbr":nbr}
```

A.1.2

```
control = quadratic_control.Quadratic_Control_PV (grid_data, active_nodes)
control.initialize_control()

control_ESS = Quadratic_Control_ESS(grid_data, active_ESS)
control.initialize_control()
```

A.1.3

```
[reactive_power, active_power] = control.control_(pv_input, reactive_power, active_power,
v_gen)

active_power_ESS = control_ESS.control_(active_power_ESS, v_ess)
```

A.1.4

```
reactive_power = control_reactive_power.Voltage_Control(PV_list, q_PV, v_gen, alpha)
active_power_PV = control_active_power_PV.Voltage_Control(PV_list, active_power_PV,
v_gen, alpha_PV)

active_power_battery = control_active_power_ESS.Voltage_Control(active_power_battery,
v_ess, alpha_ESS)
```

A.2 Code related to chapter 4

A.2.1

First the mqtt server is initialized as:

```
''' Initialize objects '''
dmuObj = dmu()
''' Start mqtt client '''
mqttObj = mqttClient("mqtt", dmuObj)
```

Then the subscribers are defined, which writes into dictionaries.

```
mqttObj.attachSubscriber("/voltage_control/measuremnts/voltage","json","voltage_dict")''' S
```

```
mqttObj.attachSubscriber("/voltage_control/measuremnts/pv","json","pv_input_dict")
```

The publisher are defined as:

```
mqttObj.attachPublisher("/voltage_control/control/active_power","json","active_power_dict")
mqttObj.attachPublisher("/voltage_control/control/reactive_power","json","reactive_power_dict")
mqttObj.attachPublisher("/voltage_control/control/active_power_ESS","json","active_power_ESS_dict")
```

To read the message received by the subscriber, a function getDataSubset of the dmu object is used:

```
voltage_value = dmuObj.getDataSubset("voltage_dict")
pv_input_value = dmuObj.getDataSubset("pv_input_dict")
```

To pass the data to the publisher, the setDataSubset function of the dmu is used:

```
dmuObj.setDataSubset({"active_power":active_power_dict},"active_power_dict")
dmuObj.setDataSubset({"reactive_power":reactive_power_dict},"reactive_power_dict")
dmuObj.setDataSubset({"active_power_ESS":active_power_ESS_dict},"active_power_ESS_dict")
```