Decentralized Grid Control in COLMENA Activity 3

Pablo de Juan Vela [1], Marco Muttoni [1], Josep Fanals i Batllori [1]
[1]eRoots Analytics, Barcelona, Spain

July 9, 2025

1 Introduction

Modern power systems undergoing a fundamental transformation, driven by the large-scale integration of renewable energy sources like solar and wind. This shift from a few centralized, predictable power plants to thousands of distributed, intermittent resources introduces significant operational challenges. The reduced system inertia and increased variability make the grid more vulnerable to disturbances, threatening its stability and reliability. Traditional, centralized control architectures are ill-equipped to manage this new level of complexity, as they suffer from communication delays and struggle to scale effectively. To address these issues, new decentralized control paradigms are essential.

COLMENA aims to enable decentralized decision-making by coordinating autonomous agents that perform specific roles. Each agent can execute certain roles to impact the system and operate based on local measurements, aggregated measurements over a group of agents, and predictions. By deploying agents at or near the grid assets themselves, COLMENA allows for rapid, localized responses to disturbances based on local measurements, while ensuring system-wide stability through coordination with neighboring agents. This modular and scalable architecture is a natural fit for managing the complex, dynamic, and heterogeneous nature of the future power grid.

While COLMENA can be applied to many grid management tasks, this project focuses on one of the most critical: frequency control. Grid frequency, which must be kept within a very tight band around its nominal value (e.g., 50 Hz), is the primary indicator of the real-time balance between electricity generation and consumption. A sudden loss of a generator or a large load can cause rapid frequency deviations. If not corrected within seconds, these deviations can

trigger protective disconnections of other grid elements, leading to cascading failures and, in the worst case, large-scale blackouts.

Distributed Model Predictive Control (MPC) is a particularly well-suited control strategy in this context (see reference [1] for a detailed explanation of the distributed MPC framework and its application to power systems). MPC anticipates future states and disturbances, optimizes control actions over a prediction horizon, and incorporates constraints in a clear manner. When implemented in a distributed fashion through COLMENA, each agent can locally solve an MPC problem to manage its area, while coordinating with neighboring agents to respect power flow constraints and system-wide objectives.

This report first explains the physical models and equations used to build the MPC, then covers the optimization background used to implement the distributed MPC, and finally analyzes the performance of the MPC control in the IEEE 39-bus grid through the ANDES simulator in order to showcase its capability for frequency control.

2 Modeling

Accurate modeling of the contingencies that are likely to take place in the grid, as well as the frequency and angle dynamics of each area, are fundamental to the design and implementation of distributed control strategies in power systems. Otherwise, the distributed control may be built on top of erroneous physical models, which can later lead to suboptimal control actions. Hence, it is central to capture how each area responds to generation-demand imbalances and how it interacts electrically with neighboring regions through phase angle differences whenever a contingency takes place. By understanding these behaviors, the proposed MPC can anticipate future states and control systems can react accordingly.

2.1 Contingency definition

In power systems, contingencies refer to unexpected events or disturbances that can significantly impact the system's stability and operation. These events can range from equipment failures to sudden changes in load or generation, and they require immediate response from control systems to maintain grid stability and prevent cascading failures [2]. Not surprisingly, the recent Spanish blackout that took place last April was a clear example of such cascading events. Contingencies can be caused by multiple, being the most common types of contingencies the following:

- **Generation contingencies:** these involve the sudden loss of generating units due to mechanical failures, protection system trips, or fuel supply issues. Generator outages can cause immediate power imbalances, leading to frequency deviations and potential voltage problems. The severity depends on the size of the lost generation relative to the system's total capacity and available reserves. Most often, loss of generation causes the frequency to drop, which has to be counteracted by the control actions of other generation units, or also the reduction or even disconnection of loads [3].
- **Transmission line contingencies:** these occur when transmission lines are disconnected due to faults, protection system operation, or physical damage (also linked weather events or equipment failure). Line outages can cause power flow redistributions, potentially leading to overloaded remaining lines and voltage violations, which can further exacerbate the frequency deviations and result in a complete blackout of the system. Restorative actions can involve the quick re-connection of the line, the re-routing of the power flow by adjusting the setpoints of nearby elements, or running an optimal transmission switching algorithm to find the best

positions of the switches and disconnectors. However, a strong power grid must also be properly planned and dedicate a budget for transmission expansion planning [4].

- **Load contingencies:** sudden changes in load demand, such as the disconnection of large industrial loads or the connection of significant new loads, can create power imbalances. While load increases typically cause frequency drops, load decreases can cause frequency rises [5]. On the other hand, loads are also employed to provide flexibility to the system, by adjusting the associated consumption to meet the requirements imposed by the transmission system operator, which are eventually monetarily compensated. Even though loads are constantly changing in the system, we define load contingencies as sudden and large spikes in demand, akin to the connection or disconnection of massive consumption points.

- **Weather-related contingencies:** extreme weather events such as storms, lightning strikes, or high winds can cause multiple simultaneous contingencies, including line outages, equipment damage, and load changes. These events often require coordinated responses across multiple control areas [6]. It is often the case that a weather-related event can simultaneously cause a generation contingency and a transmission line contingency. Hence, power grids are sometimes designed to meet not only the N-1 criterion, but also the N-2 criterion, which is the ability to withstand two contingencies at the same time.

- **Equipment failures (others):** other contingencies can be due to failures in critical equipment such as transformers, circuit breakers, or protection systems that can further lead to cascading effects throughout the system. These failures can be caused by aging, improper maintenance, or external factors [7]. In practical terms, and when seen from a modelling perspective, these equipment failures are not different from the aforementioned contingency types, as they can cause a sudden frequency variation and also a power flow redistribution.

The impact of contingencies on power system dynamics can be characterized by their severity, duration, and propagation characteristics. Severe contingencies can lead to frequency excursions that exceed normal operating limits, potentially triggering automatic load shedding or generator tripping to prevent system collapse [8].

In the context of distributed control systems like COLMENA, contingencies create the need for coordinated responses across multiple areas. The distributed MPC framework must be able to detect the impact of these events in the electrical magnitudes under measurement, assess their influence on local and neighboring areas, and coordinate appropriate control actions to restore system stability while respecting operational constraints. Figure 1 shows a schematic representation of the contingencies under study.

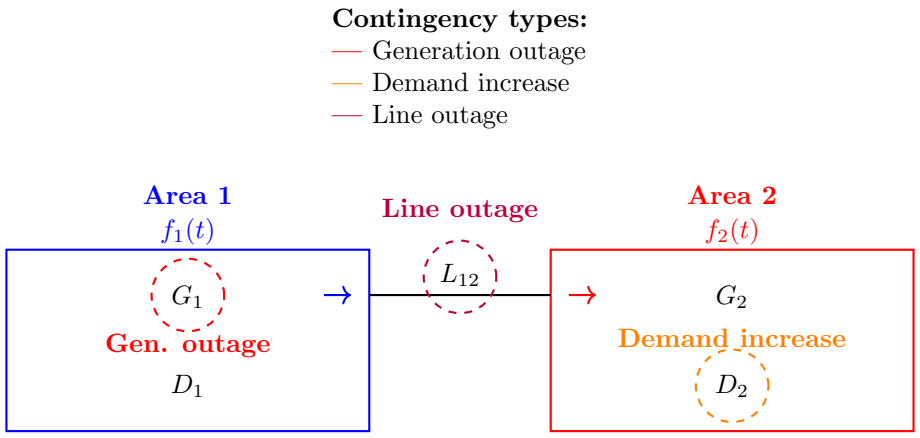

Figure 1: Schematic representation of the power system contingencies under analysis showing generation outage, load/demand increase, and line opening events in a two-area system.

2.2 Connection/disconnection definition

In modern power systems, it is increasingly common to integrate new entities, such as solar parks or other renewable energy sources. The connection of a new component means that both the new entity and the rest of the system must detect the change and adapt, ensuring that the added installation does not negatively impact, and ideally improves, the operation of the electrical grid.

From a modeling and control perspective, the connection or disconnection of system elements, whether they are generators, loads, or other devices, can be treated as a particular case of contingency. The system must respond to these events in much the same way as it does to faults or sudden disturbances: by rebalancing power flows, adjusting control actions, and maintaining stability.

For this reason, in our framework, the connection or disconnection of components is not fundamentally different from other contingencies. Both require the system to adapt dynamically, leveraging distributed control and coordination among agents to ensure continued secure and efficient operation.

To test the system's adaptability, we consider scenarios involving the connection and disconnection of power injection sources, both generators and loads, to observe how the agents adjust their behavior. In both cases, the distributed control architecture must detect the change, coordinate the necessary adjustments, and maintain the desired performance of the grid.

2.3 Grid dynamics

In the proposed framework, each electrical area is represented using two states values:

- **Frequency** $f_i(t)$: the local frequency of area i, which reflects the balance between generation and consumption.
- **Phase angle** $\delta_i(t)$: the relative voltage phase angle of area i with respect to the reference area, governing power exchanges with neighboring areas.

This abstraction treats each area as a coherent group of generators and loads that tend to oscillate together in response to disturbances. It enables scalable distributed control by reducing the system's dimensionality.

The frequency evolution in each area is governed by the so-called swing equation, which relates powers, inertia and damping with the frequency [9]:

$$f_{i,t+1} = f_{i,t} + \frac{\Delta t}{M_i}\Big[- D_i(f_{i,t} - f_0) + \sum_{k \in \mathcal{G}_i} P_t^{\text{gen}_k} - P_{i,t}^{\text{demand}} + \sum_{j \in \mathcal{N}_i} B_{j,i}(\delta_{j,t} - \delta_{i,t})\Big] \tag{1}$$

$$=: f_{i,t}(x_t) \tag{2}$$

where:

- M_i is the aggregated inertia of area i,
- D_i is the damping coefficient of area i,
- f_0 is the nominal frequency (e.g., 50 or 60 Hz),
- $P_t^{\text{gen}_k}$ is the generation in area i,
- $P_t^{i,j}$ is the power exchanged from i to area j.

The relative angle $\delta_i(t)$ accumulates the frequency deviation with the reference area over time:

$$\frac{d\delta_i(t)}{dt} = 2\pi(f_i(t) - f_0) \tag{3}$$

This relation ensures that the angle reflects the integral of frequency deviation, which is crucial for tracking system phase shifts over time.

In this formulation, we use the DC power flow approximation to model the active power exchange between areas. This approach is widely accepted in high-voltage

transmission system analysis due to its simplicity and relatively high accuracy under typical operating conditions. The approximation is based on the following assumptions:

- Voltage magnitudes are close to their nominal values and are considered constant.
- Angle differences between buses are small enough that $\sin(\delta_i - \delta_j) \approx \delta_i - \delta_j$.
- Line resistance is negligible compared to reactance, so power losses are ignored.

Under these assumptions, the active power flow from area i to area j can be approximated by the following. This shows that active power flows are driven by angle differences and line susceptance $B_{i,j}$.:

$$P_t^{i,j} = B_{i,j}(\delta_i(t) - \delta_j(t)) \tag{4}$$

where $B_{i,j}$ represents the total susceptance between the two areas. This is calculated as the sum of the susceptances of all transmission lines directly connecting buses in area i to buses in area j:

$$B_{i,j} = \sum_{\text{line i connects area i to area j}} b_{\text{line i}} \tag{5}$$

This aggregation allows us to ignore the specific topologies of the areas during inter-area coordination, while still capturing the essential physics of power exchange via angle differences. It also reduces computational complexity, which is critical for scalable distributed optimization. Figure 2 shows the topology of a two-area power system. Given the three connections between the two system sections, the resulting admittance is $b_1 + b_2 + b_3$.

2.4 Grid controls

We build a multi-agent system where each agent is responsible for the control of a defined electrical area. Each area comprises several buses, generators, and loads, and interacts with other areas through lines. The objective of the agent is to maintain frequency stability by coordinating with neighboring agents through the sharing of information such as power flows as they influence the area frequency. Each agent is equipped with the capability to perform multiple control functions within its domain, depending on the system design and available actuators. These actions may include:

- **Power generation control**: adjusting generator outputs to balance load and maintain frequency.

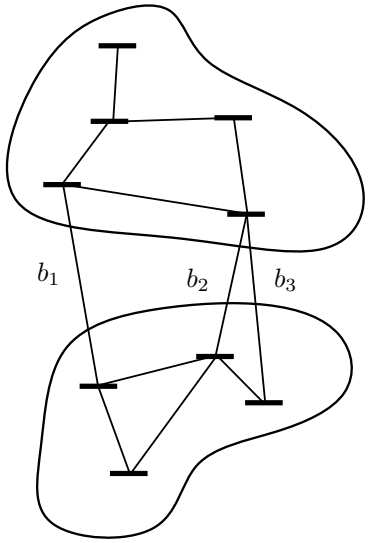

Figure 2: Topology of a simplified two-area power grid.

- **Load shedding strategies**: curtailing controllable loads during contingencies to stabilize frequency.
- **Battery or storage management**: charging or discharging storage systems to buffer fluctuations in net demand or generation.
- **Reactive power or voltage support**: engaging voltage control mechanisms to assist in voltage regulation, if needed.

In the broader COLMENA framework, agents could also be capable of activating other local roles, provided that the dynamics and interactions of those roles are explicitly defined and known by the agent. In this first implementation, we focus on a simplified setting where each agent controls the ramp of generator power outputs within its area. Specifically, the control vector u_i for agent i is defined as the difference in power output between two consecutive time steps:

$$u_i(t) = P_i^{\text{gen}}(t+1) - P_i^{\text{gen}}(t) \in \mathbb{R}^{n_{g,i}} \tag{6}$$

where $P_i^{\text{gen}}(t)$ is the vector of generator power outputs at time t, and $n_{g,i}$ is the number of controllable generators in area i. Also, the control inputs are subject to ramp-rate constraints that limit how fast the generators can increase or decrease their output:

$$u_k^{\min} \leq u_{i,k}(t)\Delta t \leq u_k^{\max} \quad \forall k \in \{1, \ldots, n_{g,i}\}, \forall t \tag{7}$$

where $u_k^{\min}$ and $u_k^{\max}$ are ramp-down and ramp-up limits for generator k.

3 Mathematical Formulation

A Model Predictive Control (MPC) is a control strategy that solves an optimisation problem at each time step to determine the best sequence of control actions over a finite future time horizon. The fundamental idea is to use a dynamic model of the system to predict its future behavior and to compute the control inputs that minimize a given cost function while satisfying physical and operational constraints.

3.1 Centralized global problem

In a distributed formulation of a power system, each area i tracks its own frequency $f_{i,t}$. Although control inputs are applied locally, the physical coupling between areas means that the frequency of area i is influenced by the overall system state, including the states of neighboring areas.

In this formulation, we express the local frequency $f_{i,t}$ as a function of the full system state vector at time t, denoted $x_t = \{\delta_{j,t}, f_{j,t}\}_{j=1}^{N}$.

State variables:

- x_t: The global state vector at time t, including:
  - $\delta_{j,t}$: voltage angle of area j
  - $f_{j,t}$: frequency of area j

Local frequency dynamics:

The frequency update in area i can now be written as:

$$f_{i,t+1} = f_{i,t}(x_t) \tag{8}$$

More explicitly, using the physical model:

$$f_{i,t+1} = f_{i,t} + \frac{\Delta t}{M_i}\Big[- D_i(f_{i,t} - f_0) + \sum_{k \in \mathcal{G}_i} P_t^{\text{gen}_k} - P_{i,t}^{\text{demand}} + \sum_{j \in \mathcal{N}_i} B_{j,i}(\delta_{j,t} - \delta_{i,t})\Big] \tag{9}$$

$$=: f_{i,t}(x_t) \tag{10}$$

This shows that $f_{i,t+1}$ depends not only on $f_{i,t}$ and $\delta_{i,t}$ (local state), but also on $\delta_{j,t}$ for $j \in \mathcal{N}_i$ (neighboring states). Hence, local frequency regulation depends on the evolution of the entire system state.

Angle dynamics:

In the discretized time formulation that is adopted, the angle dynamics is given by:

$$\delta_{i,t+1} = \delta_{i,t} + 2\pi\Delta t(f_{i,t} - f_0) \tag{11}$$

Global optimization problem:

The global frequency control is equivalent to the following optimal control problem:

$$\min_{\{u_{i,t}\}} \quad \sum_{i}^{N}\sum_{t=0}^{T} \left(f_{i,t}(x_t) - f_0\right)^2 \tag{12}$$

Subject to:

$$\delta_{i,t+1} = \delta_{i,t} + 2\pi\Delta t(f_{i,t}(x_t) - f_0) \tag{13}$$

$$x_{t+1} = \text{State transition function dependent on controls and dynamics} \tag{14}$$

$$u_k^{\min} \leq u_{i,t}^{(k)} \leq u_k^{\max} \tag{15}$$

$$u_k^{\min} \leq u_{i,t+1}^{(k)} - u_{i,t}^{(k)} \leq u_k^{\max} \tag{16}$$

Where we have $u_{i,t} \in \mathbb{R}^{N\times T}$ the set of control variables. From these formulations we introduce the local state for area i, $x_{i,t}$. The variable is defined as the set of local state variables of area i and the state variables of the neighboring areas.

$$x_{i,t} = (\delta_{i,t},\ f_{i,t},\ \{\delta_{j,t}\}_{j\in\mathcal{N}_i}) \tag{17}$$

With this we can redefine the global problem as follows:

$$\min_{\{u_{i,t}\}} \quad \sum_{i=1}^{N}\sum_{t=0}^{T} \left(f_{i,t}(x_{i,t}) - f_0\right)^2 \tag{18}$$

$$\text{s.t.} \quad x_{i,t} = x_{j,t}, \quad \forall (i,j) \text{ areas s.t. } B_{i,j} \neq 0,\ \forall t \tag{19}$$

In this formulation, the objective function itself is unchanged only the dependencies of the frequency values $f_{i,t}$ changes to a function of the local state $x_{i,t}$. The key modeling change introduced here is the inclusion of a coupling constraint of the form $x_{i,t} = x_{j,t}$ for all connected pairs of areas (i,j) and time t. Given that $f_{i,t}$ directly depends on $\delta_{j,t}$ this can be simplified to the following set of coupled constraints, where $\delta_{j,i,t}$ is the value of the area's i angle seen in the area's j subproblem:

$$\delta_{i,i,t} = \delta_{j,i,t} \quad \forall t, i \quad \forall j \in \mathcal{N}_i \tag{20}$$

This constraint enforces that the local copies of shared state variables maintained by each agent (e.g., voltage angles or neighboring states) must agree

with those of their neighbors. Although each agent i solves its problem using its local state $x_{i,t}$, coordination is required so that all local views converge to a common global state.

This structure enables the use of distributed optimization techniques, such as the Alternating Direction Method of Multipliers (ADMM), where each agent optimizes independently and the consistency across agents is enforced iteratively through dual variables and auxiliary updates [10]. The use of ADMM to solve a frequency control problem is inspired by [11] and [12]. Readers are invited to check its mathematical development in the Appendix.

4 MPC Role Definition

In multi-agent architectures and in the context of COLMENA a role is usually tied to a local behavior with a narrow task that is activated when a specific value or KPI is not respected. In this context, the distributed MPC can be seen as a policy coordinating multiple roles as much a role, given that various devices controlled through MPC within each specific area can offer different reactions to a grid disturbance. For this we propose two different solutions that limit the MPC to specific circumstances making it more akin to a proper role than a policy.

4.1 Frequency area response

In this configuration, the distributed MPC is deployed as a reactive role rather than a permanent coordination policy. When frequency takes abnormal values, which could potentially signal instability, as detected by the associated KPI being surpassed, COLMENA activates this role tasked with coordinating corrective actions within that area.

The COLMENA framework can select the most computationally capable agent within the affected area to execute the distributed MPC coordination logic. This agent acts as the local coordinator, solving the area-wide optimal control problem while communicating with neighboring agents to respect tie-line interactions and boundary constraints. This structure means that a single agent by area is executing the frequency area response role.

The role's objective is to minimize frequency deviations in the local area by adjusting controllable resources (e.g., generation and controllable loads) over a prediction horizon. The optimization prioritizes frequency recovery while incorporating ramp limits and operational constraints.

Within this context, although the distributed MPC involves system-wide coordination and may interact with multiple roles, it behaves as a role in this context because:

- It is activated only in response to a specific condition (frequency instability).
- It has a well-defined and limited objective: frequency stability.
- Changing the objective function (e.g., minimizing cost instead of deviation) or time scale (e.g., minutes vs seconds) would create a role with a different objective, not simply a parameter change within a generic policy.

If instead of minimizing the frequency deviation, the agent optimized for economic dispatch over a longer time scale such as 10 minutes, the same underlying MPC engine would constitute a completely different role, for instance, a `CostOptimizationRole`. This highlights that roles are defined by their specific objectives, context, and triggers, and not merely by the algorithm they run.

Within COLMENA's architecture, the distributed MPC can straddle the line between coordination mechanism and agent behavior, depending on how it is contextualized and activated. By bounding its scope through KPIs and objectives, it becomes operationally meaningful as a discrete role. Table 1 summarizes the characteristics of the described role.

Aspect	**Description**
Role Name	`FrequencyAreaResponseRole`
Objective	Minimize frequency deviation in the local area by optimally adjusting controllable resources over a prediction horizon.
Activation Condition	Triggered when the frequency KPI is violated, i.e., $\|f_i - f_0\| > \Delta f_{\max}$ for any agent i in the area.
Scope	Local to an area, with coordination across neighboring agents.
Agent Requirement	Computational power and specific solver
Time Scale	Fast-acting (typically sub-minute) response to stabilize frequency
Key Dependencies	Access to local state measurements (frequency, power injection), ramp-rate limits, and communication with neighboring agents.

Table 1: Summary of the `FrequencyAreaResponseRole`.

The following pseudocode describes the online control loop executed by each local agent in area i when the distributed MPC role is activated.

Algorithm 1 Local Agent i — Online Distributed MPC

Initialization: Obtain initial state $x_{i,0}$ and initialize dual variables and neighbor estimates.
while Frequency KPI is broken **do**
    while not converged and max iterations not reached **do**
        Receive latest state estimates from neighbors $j \in \mathcal{N}_i$
        Solve local MPC optimization problem over horizon $t \to t + T$
        Exchange updated local state estimates with neighbors
        Update dual variables (consensus step)
    end while
    Apply first control input $u_{i,t}$ to the system
    Measure or estimate new local state $x_{i,t+1}$
    Shift horizon and prepare for next step
end while

4.2 Specific implementation

In this section we discuss how the different tools provided by COLMENA are used to set up the grid. Specifically, how the data and contexts decorators are used by the Distributed MPC role to be implemented in parallel. The parallel implementation is key to the distributed nature of the MPC.

Contexts decorators

The use of COLMENA's contexts fits quite well with the concept of areas. In the context of electrical grids, areas are defined by a set of buses and the electrical devices connected to those buses. Here we consider agents to belong to specific contexts when they are part of an area. For example, we could have contexts defined by the agents ID beforehand, or if we consider the initial prototype, if an agent is paired to an electrical device, then the agent would belong to that area's context.

We can use contexts to send control messages from the agent running the Distributed MPC to other agents. For instance, instead of the MPC including the dynamics of certain roles, the MPC can set objectives that then other roles inside the area can react to. For this case we define *Grid Areas* a context where a set of COLMENA agents are interlinked depending on their Area.

In the context of the distributed MPC, we can use this *Grid Areas* to define a Data channel with that scope. This channel makes it possible to store the dual variables of each area independently of the agent that is running the MPC role. With this implementation we could run the MPC in different agents in the same area without problems because the new agent executing the role could read the

saved duak variables and keep running the MPC where the previous agent left. This is not available right now since we would need to ensure that one and only agent per area is running the MPC role. However, we expect this functionality to be available eventually which would make the execution of this role much more dynamic.

An example of the code being

```
class GridAreas(Context):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)

    def locate(self, device):
        agent_id = os.getenv('AGENT_ID')
        #agent_2_area a predefined dictionary
        agent_area = agent_2_area[agent_id]
        id = {'id':agent_area}
        print(json.dumps(id))

class Distributed_MPC(Role):
    @Requirements('AREA')
    @Metric('frequency')
    @Context(class_ref = GridAreas, name='grid_areas')
    @Data(name = 'dual_vars', scope = 'grid_areas/id=.')
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        [...]


    @Persistent()
    def behavior(self):
        self.dual_vars_values = self.dual_vars.get()

        #We solve the local MPC and update the dual vars
        [...]

        self.dual_vars.publish(self.dual_vars_values)
```

Data decorators

We define three different types of data channels to share information between the different agents running the ADMM:

- **State horizon zata:** channel where the copies of all the coupled primal

variables are sent between agents, that is the $\hat{\delta}_{i,j,t} \quad \forall i, j, t$.

- **Dual variables data:** channel where the dual variables are stored, this specific channel is scoped with respect to the context *Grid Areas*. This makes it so that each area stores their own copy of the dual variables. Such an approach can be leveraged so that different agents in the same area could run different iterations of the online control.
- **Error data:** channel where the agents update the dual error of the ADMM algorithm. When the error goes below a specific tolerance we consider the ADMM has converged. Then the agents send the control actions to the ANDES simulation and wait until the next step.

Coordination

The different agents read the state horizon channel to get the values for the primal variables of other agents and then post their updated primal values. To avoid agents overwriting the changes of other agents, we also add a coordination feature. This consists in different agents having to update various data channels depending on the context. We also consider defining a new coordinating agent to perform this task, which could be implemented in later phases, although it was deemed unnecessary at this stage.

In this implementation of the MPC, we define n_{agents} data channels where n_{agents} is the number of agents. The agent i reads the primal variables from the data channel i and publishes the data that it wants to share in the data channel $i + 1$, except the agent number n_{agents} that publishes in the data channel 1. The data that each agent publishes is the set of values $\hat{\delta}_{i,j,t} \forall i, j, t$. This means the agent does not only publish his primal values but also the ones his received in previous iterations, this way the data can reach every agent.

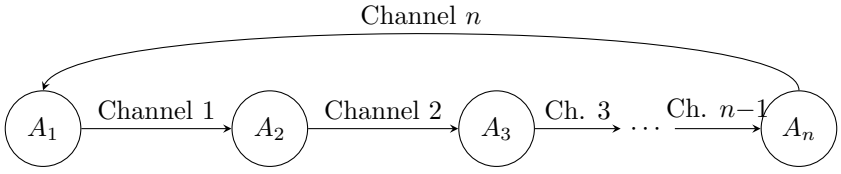

Figure 3: Data flow among agents in MPC implementation.

Additionally, for tracking the convergence of the ADMM we define a data channel called global error. The data channel tracks the error of every area in the form of a dictionary where the area's number is the key and the value is that area's error. To syncronise the different agent we also define an attribute *to_publish*. In every iteration the agent in area i can only publish if the value of *to_publish* saved in the data is equal to its areas number, if not it waits. When

it can finally publish it publishes its error and updates *to_publish* to $i + 1$. Effectively, this means that the agents publsh the error in the same order as their areas. In practice, this scheme works but its quite inefficient time wise, so we chose to just run the ADMM for a fixed number of iterations.

Activation

The distributed frequency control role is activated when the KPI related to the frequency control is broken, that is, when at least two frequency values are outside the allowed range. The KPI is defined as ($\omega \in [0.999, 1.0001]$) and is monitorised by the monitoring role executed persistently. When the role is activated, it first communicates with ANDES and learns the initial conditions for the problem. It then solves a single iteration of the local problem. In the prototype, the other iterations are also executed by the same agent. The goal however is to potentially use COLMENA's resource control to run the solver, for example, in the most computationally powerful agent available. This could be possible thanks to the different communications channels already storing the necessary data independently of the agents.

4.3 Testing environment

The testing was carried out on a local machine, where the agents in the COLMENA framework run in Docker containers, maintaining the decentralized nature of the system. These agents communicate locally with the ANDES simulator, which is also hosted on the same machine. The use of Docker containers for the agents ensures that the system remains modular and scalable, allowing for independent operation and coordination between agents while still adhering to decentralized principles.

4.3.1 Andes-COLMENA Interaction

The COLMENA agents and the ANDES simulation run in separate containers and communicate through HTTP requests over the local network. COLMENA agents request the current state of the grid from the ANDES simulator, compute their local control actions, and update their internal states accordingly. Similarly, the agents send the control actions to COLMENA changing the grid simulation dynamically. This communication is handled asynchronously, enabling the agents to operate in parallel and maintaining the system's decentralized nature.

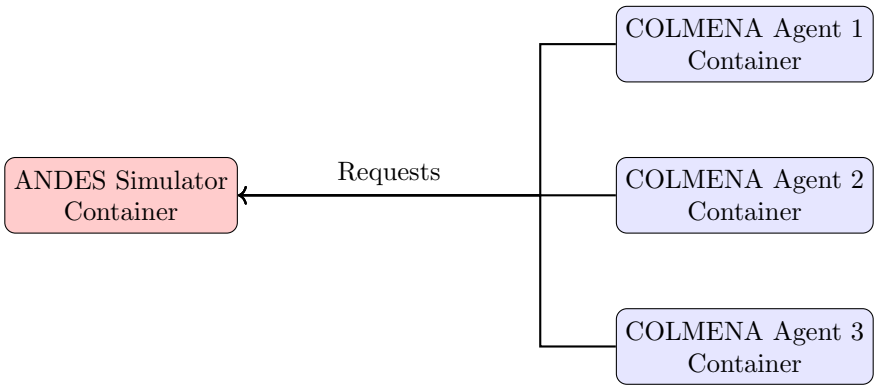

Figure 4: Interaction between COLMENA agent containers and the ANDES simulator container via requests.

Andes Simulator deployment

To run the Andes simulation in a separate docker container we run the following code, with `config.py` the file that can be modified for different simulation scenarios and `/home/output_plots` the folder where the simulation plots will be saved.

```
docker run \
  -p 5000:5000 \
  -v $(pwd)/config.py:/home/colmenasrc/config/config.py \
  -v $(pwd)/home/output_plots:/home/output_plots \
  pablodejuan/andes_src
```

The configuration file can have the following aspect. The attribute "failure" describes the type of perturbation that will happen in the simulation, it can have the following values: "line", "generator", "load". A model configuration file is present in the main directory of the repository as an example. We can also change other values relative to the simulation such as the simulation time, the final time, the disturbance time and other parameters.

```
class Config:

    andes_url = "http://127.0.0.1:5000"
    case_name = "ieee39" # case_name = "kundur" "ieee39" "npcc"
    case_path = f"{case_name}/{case_name}_modified.xlsx"
    failure = 'load'
```

```
# Simulator
tstep = 0.05
tf = 20.0

# Disturbance
td = 5.0
[...]
```

It is important to remark that given the MPC long solving time, we are forced to slow down the simulation once the MPC role is activated in order for the control actions to be meaningful. We are working on speeding up the MPC in order for the system to be fully online and in parallel.

COLMENA Agent deployment

To deploy the COLMENA agents we just need to run the script `colmena_mpc.py` specifying the COLMENA service that we want to deploy and the grid we want to deploy it to. In the existing use case we use `service_file=mpc_multiple_areas.py grid_name=ieee39`. Therefore we initialise it with the following code

```
service_file = mpc_multiple_areas
#service_file can be changed for other services (such as mpc_consensus, mpc_kpi_driven)
grid_name = ieee39
docker_user = #YOUR_DOCKER_USER
colmena_dir = #COLMENA_INSTALLATION_DIRECTORY
python_dir = #YOUR_PYTHON_OR_VIRTUAL_ENVIRONMENT_DIRECTORY
python3 colmena_services.py
```

This code deploys the necessary agents and roles but does not build the service since this has been done previously. The agents run on specific docker images `pablodejuan/agent_src` that have access to the necessary code for running the MPC as well as the IPOPT solver [13]. The number of agents deployed as well as their requirement depends on the chosen grid.

4.4 COLMENA justification

The COLMENA framework is a good fit for frequency optimization mainly because it gets around the delays of a central controller. Frequency problems start locally, but a central system has to wait for data from all over the grid before it can react. By the time it sends commands back, precious time is lost.

COLMENA lets local agents act immediately based on their own measurements, which is much faster. They still coordinate with their neighbors, so their quick fixes work together to stabilize the whole grid instead of just creating new problems elsewhere.

A decentralized control system is generally speaking a much more practical way to handle modern grids. As the grid is full of different devices renewables, HVDC links, switches, among others, trying to update one giant, central control system for every new piece of equipment is deemed impractical. With COLMENA, a new agent with the right role can be deployed for that equipment. This makes the system more flexible and robust. If one agent fails, the others can still operate and pick up the slack, so there is no single point of failure that can take the whole system experience a blackout.

5 Case Studies

We consider the IEEE 39-bus grid, whose data can be found in [14], to perform multiple case studies and analyze how the COLMENA framework performs in controlling the frequency. The grid consists of 39 buses, 10 generators, 14 lines, among others. The calculations are performed in Python using the open-source package ANDES [15]. This grid is divided in two different areas of similar size, where each bus belongs to one specific area. The setup is identical to the one explained in the previous deliverable: ANDES is set up as an app that simulates the grid in real time. This app is also prepared to receive queries to from COLMENA regarding the state on the grid, akin to taking measurements of specific metrics on the grid. Additionally, it is also possible to modify the parameters of the simulation similar to a role acting on the grid. Figure 5 shows the diagram of the IEEE 39-bus grid with two areas.

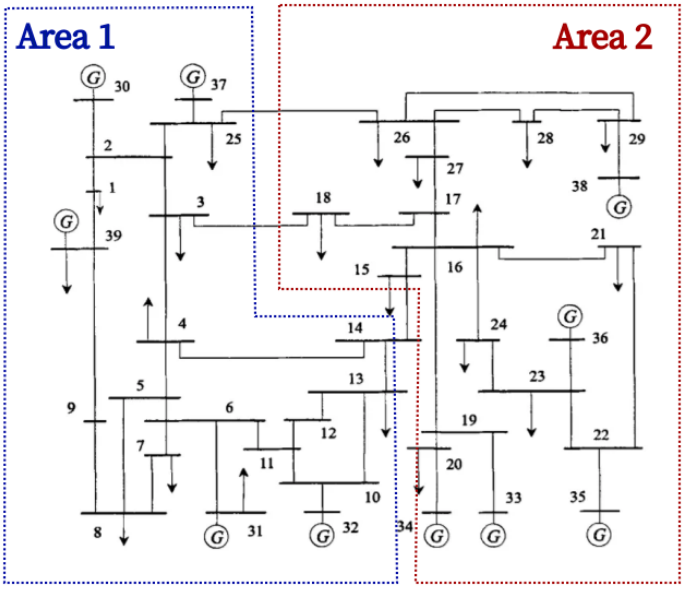

Figure 5: Diagram of the IEEE 39-bus grid with two area.

From the COLMENA side, we employ multiple agents as Docker containers deployed in a single computer. We deploy two agents per area, where these agents can run both the Distributed MPC role and the roles presented in the previous deliverables (if they respect the requirements). In each area we deploy:

- Agent 1: Agent with requirement ('AREA')

On the simulation side, we start the simulation with a grid in steady-state conditions. We set up multiple scenarios with the objective of studying how the different roles act in the grid to counteract this perturbation. The distributed MPC is configured with a horizon of 3 seconds and a time step of 100 ms.

5.1 Case study 1 - Load decrease

We define a perturbation at $t = 5s$ that consists in a load suddenly disconnecting, this sudden loss is equivalent to losing 600 MW and 200 Mvar. This perturbation located at bus 3 provokes an increase in frequency. The distributed frequency optimizer role activates 1 seconds after the KPI (around t = 6.5s) is broken given the required time to set up. Once the role is activated, the frequency is brought back closer to the nominal value. The grid does not come back to exactly the nominal frequency because of the assumptions made in the modelling of the grid (mainly, modelling the system as linear when it is inherently non-linear). The final oscillations could then be cancelled by the Automatic Response Control. The comparison of the frequency evolution with and without the distributed MPC is shown in Figure 6 and 7.

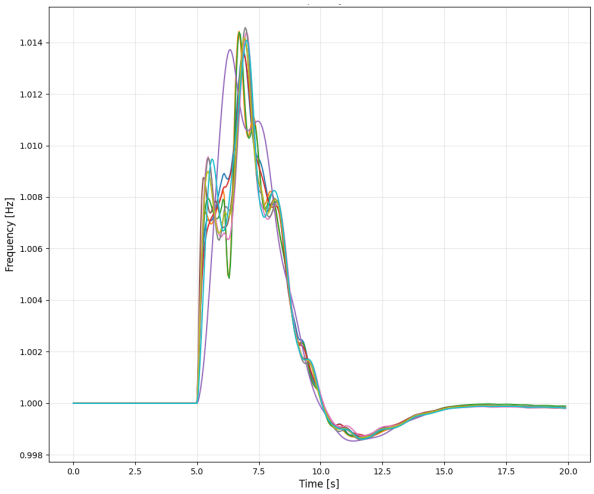

Figure 6: Frequency evolution for study case 1 with MPC

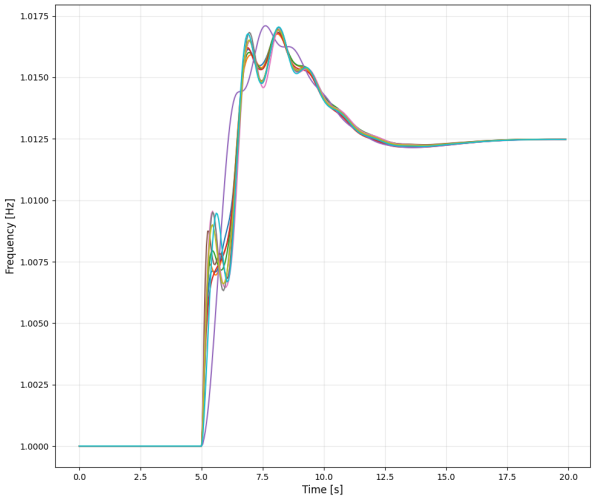

Figure 7: Frequency evolution for study case 1 without MPC

5.2 Case study 2 - Generator outage

For this case study we define the perturbation as a sudden loss of generation. The synchronous generator *GENROU 1* disconnects and stops producing 436 MW and 140 Mvar. The activation of the MPC role follow the same principle.

The loss of the generator produces a decrease in frequency that is counteracted by the Agents actions. The comparison of the frequency evolution with and without the distributed MPC is shown in figures 8 and 9. We see that the area's frequency go back to values very near to 1 p.u. but not quite. This is again likely due to a numerical error.

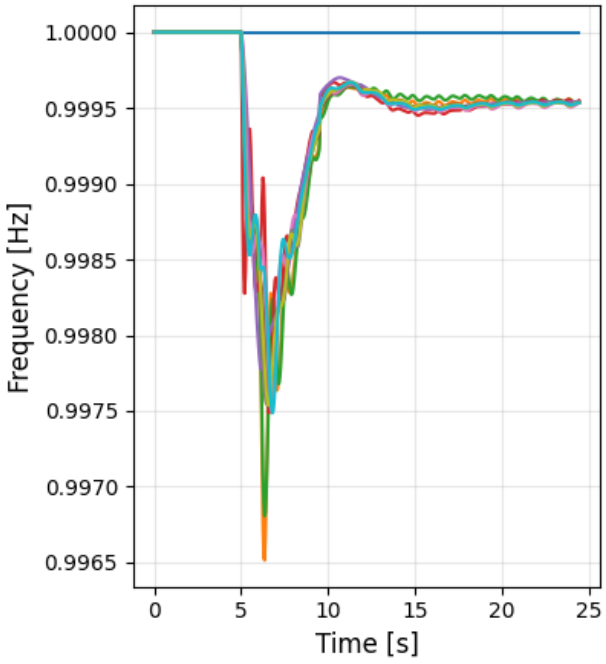

Figure 8: Frequency evolution for study case 2 with MPC

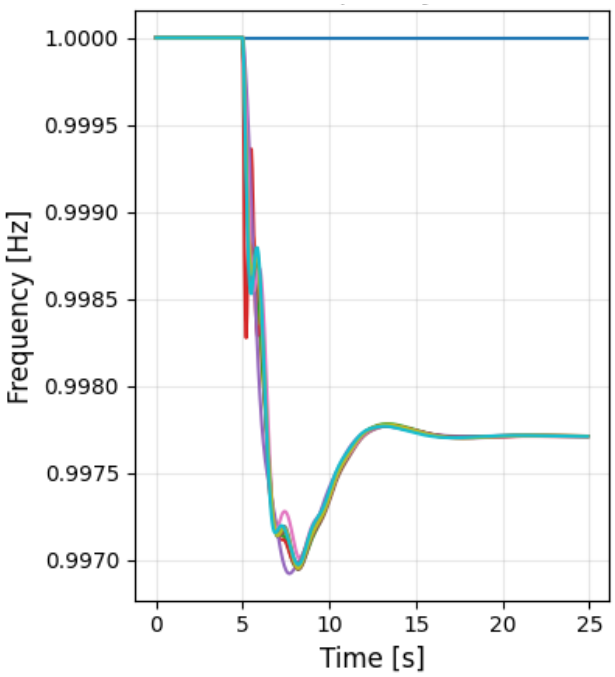

Figure 9: Frequency evolution for study case 2 without MPC

5.3 Case study 3 - Line outage

For this case study we define the perturbation as a line outage, the line disconnected is the one between buses 1 and 2. This perturbation produces some small oscillations, but that don't through the frequency off balance, specially compared to the previous tests. In this case the MPC control performs worse that the automatic control already in the grid. This is a sign that for small perturbations that are dampened quickly the MPC control should probably not be used. Changing the KPI so that the role only activates when the KPI has been broken for long enough (around 5 seconds) can solve this.

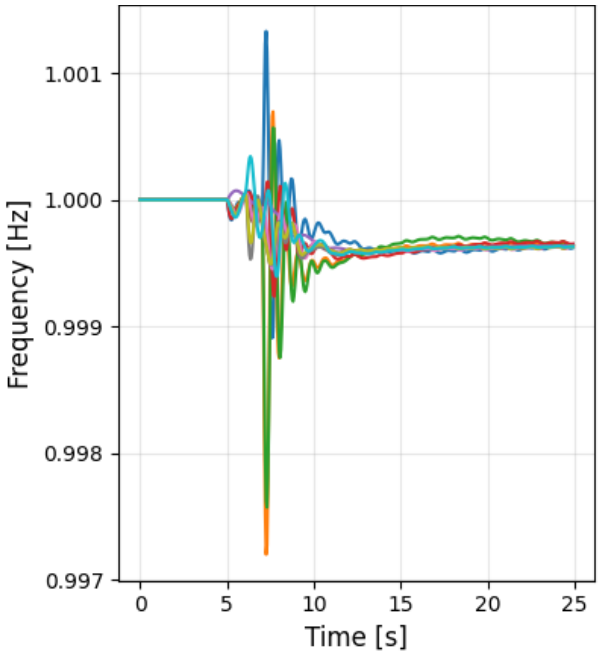

Figure 10: Frequency evolution for study case 3 with MPC

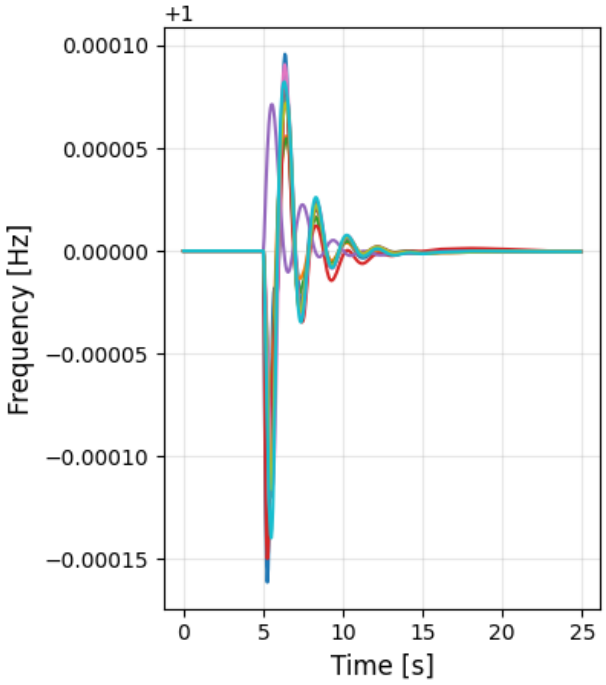

Figure 11: Frequency evolution for study case 3 without MPC

6 Milestones

This brief chapter summarizes the milestones of the project associated with this deliverable corresponding to Activity 3.

Milestone 1 - Prototype code delivery

The final version of the prototype is now fully functional and has been successfully deployed. This version leverages the tools provided by COLMENA to enable scalable, modular deployment of decentralized control algorithms. It has been extensively tested using multiple contingency scenarios, including generator outages, line outages and loads increases to validate the robustness of power converter responses under possible stress conditions. The response of the COLMENA control in these scenarios validates the capacity of the roles to respond to perturbations and restore the original frequency. The different testcases along with the prototype are available in the following online repository [16].

Milestone 2 - Prototype deployment

The prototype has been successfully deployed on an eRoots machine, following a collaborative effort between eRoots engineers and BSC staff working to overcome possible difficulties during the development. We ensured compatibility with the required software stack and performance expectations. The specific ANDES app tasked with running the simulation is also available as a Docker container in case it is needed further testing is needed.

7 Conclusions and Future Work

The results from the simulations validate that deploying frequency control using the COLMENA framework is an effective approach for maintaining grid stability. The system successfully managed frequency deviations, responding to disturbances in a timely manner and restoring the frequency to its nominal value. This demonstrates the robustness and effectiveness of COLMENA's decentralized control strategy, confirming its potential as a reliable solution for frequency regulation in power grids.

While the current study on the IEEE 39-bus system demonstrates the fundamental viability of the distributed MPC, the most critical next step is to rigorously evaluate its performance at scale. Therefore, the future work will be centered on scaling the system to handle larger and more complex grid structures to ensure its readiness for real-world applications. This will involve the following key activities:

The primary objective is to test the limits of the distributed MPC controller by applying it to significantly larger power systems. This transition is not merely about increasing the number of buses but also about introducing greater topological complexity.

- **Progressive system scaling:** we will systematically move from the current 39-bus system to well-established, larger benchmarks, such as the NPCC 140-bus system, or potentially reaching up to the PEGASE 1000+ bus grids.
- **Handling increased complexity:** these larger grids introduce challenges that are not present in smaller systems, such as higher mesh-like interconnectivity, the presence of multiple, weakly connected control areas, and more complex power flow patterns. We will assess how the distributed control framework adapts to these more realistic and challenging structures.
- **Expanding the number of control areas:** scaling will involve partitioning these larger grids into a greater number of control areas, with more numerous agents. This will directly test the scalability of the coordination mechanism (ADMM), as the number of interacting agents and required communication channels will increase substantially.
- **Testing different consensus schemes:** In this we impemented the ADMM algorithm without any type of coordinator. In the following work we want to compare this approach with the use of a coordinating agent that centralizes some of the data and simplifies some of the communication complexity.

- **Combining the previous roles:** Sacaling the prototype in the typos of roles deployed inside the agents.

As the system size grows, it is crucial to quantify the impact on performance to ensure the framework remains practical for real-time operation. We will evaluate performance along several key dimensions:

- **Computational speed and efficiency:** we will measure the number of iterations required for the distributed agents to reach a consensus. A key research question is whether the convergence rate scales linearly or sub-linearly with the number of agents, which is critical for maintaining performance. We will also track the time required to solve the optimization problem at each time step. This time should remain well below the control interval (e.g., a few seconds) to be viable for online control.
- **Control accuracy and precision:** for systems of a manageable size (like the IEEE 118-bus), we will compare the results of the distributed MPC against the performance of the non-MPC case. This will allow us to quantify any loss of optimality and ensure the control actions remain precise and effective. We will analyze the controller's ability to maintain grid stability (frequency, voltage profiles, line flows) in these larger, more inertial systems, to make sure no adverse dynamics are introduced by the distributed decision-making process.
- **Communication overhead:** we will analyze the data exchange requirements between agents. As the number of areas and interconnections grows, the communication network's load increases. We will measure the volume of data that needs to be exchanged per iteration to identify potential bottlenecks and inform the design of efficient communication protocols.

A Distributed MPC via ADMM

In order to enforce consistency across areas while allowing decentralized computation, we adopt the Alternating Direction Method of Multipliers (ADMM) to solve the distributed MPC problem. At each iteration, each agent i solves a local optimal control problem over its prediction horizon. This problem incorporates local dynamics and objectives, along with augmented terms that enforce consensus on shared variables (e.g., voltage angles) with neighboring agents.

The augmented Lagrangian $\mathcal{L}_i$ for agent i combines the local objective with terms enforcing agreement on shared voltage angle variables $\delta_{i,t}$ with its neighbors. Here $\hat{\delta}_{i,j,t}$ is the copy of the variable $\delta j, t$ seen from the agent in area i. The dual variables $\lambda_{i,j,t}$ and $\lambda_{j,i,t}$ enforce the consensus constraints between agent i and neighbor j. The penalty term weighted by ρ further encourages agreement across shared variables through a quadratic penalty.

$$\begin{aligned}\mathcal{L}_i = \sum_{t=0}^{T}(f_{i,t} - f_0)^2 + \sum_{j\in\mathcal{N}_i}\sum_{t=0}^{T}\Big[\lambda_{i,j,t}(\delta_{i,t} - \hat{\delta}_{j,i,t}) + \lambda_{j,i,t}(\delta_{j,t} - \hat{\delta}_{i,j,t})\Big] \\ + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\sum_{t=0}^{T}\Big[(\delta_{i,t} - \hat{\delta}_{j,i,t})^2 + (\delta_{j,t} - \hat{\delta}_{i,j,t})^2\Big] \qquad (21)\end{aligned}$$

With this we can define a local subproblem for agent i. At each iteration of the ADMM algorithm agent i will solve minimizes the augmented Lagrangian with respect to its own local decision variable $\{u_{i,t}\}$ over the prediction horizon. The subproblem has the following expression:

$$\begin{aligned}\min_{\{u_{i,t}\}} \quad & \sum_{t=0}^{T}(f_{i,t} - f_0)^2 \\ & + \sum_{j\in\mathcal{N}_i}\Big[\lambda_{j,i,t}(\delta_{j,t} - \hat{\delta}_{i,j,t}) + \lambda_{j,i,t}(\hat{\delta_{j,t}} - \hat{\delta}_{i,j,t})\cdot\Big] \\ & + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\sum_{t=0}^{T}\Big[(\delta_{i,t} - \hat{\delta}_{j,i,t})^2 + (\delta_{j,t} - \hat{\delta}_{i,j,t})^2\Big] \qquad (22)\end{aligned}$$

Subject to the following local constraints:

$$u_k^{\min} \le P_{t+1}^{\text{gen}_k} - P_t^{\text{gen}_k} \le u_k^{\max} \quad \forall k \in \mathcal{G}_i,\ \forall t \qquad (23)$$

$$P_{\min}^{\text{gen}_k} \le P_t^{\text{gen}_k} \le P_{\max}^{\text{gen}_k} \quad \forall k \in \mathcal{G}_i,\ \forall t \qquad (24)$$

Interpretation:

- The first term penalizes local frequency deviation over the prediction horizon.
- The second term introduces dual variables $\lambda_{i,j,t}$ and $\lambda_{j,i,t}$ that enforce agreement between shared voltage angle variables $\delta_{i,t}$ and $\delta_{j,t}$ across neighboring agents.
- $\hat{\delta}_{j,i,t}$ is the angle value received from neighbor j during the previous iteration.
- The last term is a quadratic penalty that reinforces consensus via the parameter ρ.

A.1 Local problem modifications

A. Damping term

We add an additional term damping term to the cost function when solving the local MPC. It penalizes deviations from the previous iteration $k-1$ of the primal variables. The use of this term is heavily influenced by [17].

$$\frac{\beta}{2}\|\delta_{i,t} - \delta_{i,t}^{k-1}\|^2 \quad \forall i, t \tag{25}$$

where $\delta_{i,t}$ is the area's decision variable for the angle and $\delta_{i,t}^{k-1}$ is its value in the previous iteration, and $\beta > 0$ is a damping coefficient. This modification biases the optimizer towards solutions close to the prior iteration, it stabilizes convergence and avoids oscillations in the horizon state.

B. Adaptable ρ

Instead of using a fixed penalty parameter ρ, we adjust it based on the ratio of primal and dual residuals. This helps balance the convergence of both residuals and improves performance across different problem scales[10]. The update rule is:

$$\rho^{k+1} := \begin{cases} \tau\rho^k & \text{if } \|r^k\| > \mu\|s^k\| \\ \rho^k/\tau & \text{if } \|s^k\| > \mu\|r^k\| \\ \rho^k & \text{otherwise} \end{cases} \tag{26}$$

Here, r^k and s^k are the primal and dual residuals respectively, and $\tau > 1$, $\mu > 5$ are tuning parameters. The residuals in iteration k are defined as follows:

$$r_{j,i}^k := \delta j, i^k - \delta_j^k \tag{27}$$

$$s_{j,i}^{k} := \rho \left(\delta_{j}^{k} - \delta_{j}^{k-1} \right) \tag{28}$$

C. Cost function scaling

In practice, the magnitudes of the objective terms across agents can differ significantly. Specially, when dealing with the frequency in per unit the cost function can be of the order of 10^{-6}. This means that if the cost function is not properly scaled the penalty and lagrangian term override the capcity of the solver to converge. This results on reaching consensus easier but with suboptimal horizons. Thats we redefine the frequency cost as follows. In the simulation we use $\gamma = 10^{9}$.

$$\gamma ||f_{i,t} - f_{nom}||_{2}^{2} \tag{29}$$

This subproblem defines the core of the distributed MPC framework under ADMM. Each agent optimizes locally, exchanges information with its neighbors, and updates its variables based on the augmented Lagrangian terms, gradually driving the overall system toward a consensus-based optimal trajectory.

D. Smoothing and terminal cost

Additionally, we add two new terms to the cost function in order to get better results out of the distributed MPC. These new cost terms consists of a smoothing term that smooths the trajectory of the frequency:

$$\alpha_{1} \sum_{0 \leq t < T} (f_{i,t+1} - f_{i,t})^{2} \tag{30}$$

And a Terminal cost term that we define as follows:

$$\alpha_{2} (f_{i,T} - f_{nom})^{2} \tag{31}$$

Overall, these additions help us find a more suitable solution of the frequency's trajectory rather than just minimizing for the cost.

E. Redundant coupled constraints

Finally, we add additional coupled constraint to the local MPC formulation. This new constraint do not change the model mathematically but significantly improve convergence speed. The existing formulation can pose problems since it achieves consensus in the states that are further in the horizon and can lock the

problem into a suboptimal path. The new coupled constraint intent to achieve consensus between the agents also in the dynamics. They are the following in

$$P^{\text{exchanged from i to j}}_{\text{seen from area i}} + P^{\text{exchanged from j to i}}_{\text{seen from area j}} = 0 \tag{32}$$

$$\omega_{i,i,t} = \omega_{j,i,t} \tag{33}$$

We can also write this equations in terms of $\delta_{j,i,t}$. With this addition we also need to modify the local MPC with new terms.

A.2 ADMM Algorithm

Algorithm 2 Distributed MPC via ADMM

Initialize states $x \leftarrow x_0$ by getting the grid's actual state.
while consensus error $> \epsilon$ **do**
    for each agent i **do**
        Receive $\hat{\delta}_{j,i,t}$ from neighbors $j \in \mathcal{N}_i$ the copies of the shared values in the neighboring agents.
        Solve local MPC with updated dual and penalty terms.
        Send $\delta^*_{i,j,t}$ to agent's i neighbors.
    end for
    for each pair (i, j) **do**
        Update dual variables:

$$\lambda_{i,j,t} \leftarrow \lambda_{i,j,t} + \rho(\delta_{j,j,t} - \hat{\delta}_{i,j,t})$$

    end for
end while

This distributed MPC approach converges to the centralized solution while allowing each agent to optimize its control strategy independently and in parallel. The dual variables ensure consensus across shared variables, and ADMM iterations enforce global coordination.

In the prototype we aim to implement an online implementation of this ADMM by solving the ADMM algorithm iteratively. In practice this means that each area solves its local optimization problem over a finite horizon and exchanges boundary variables with neighboring areas. Once there is convergence, only the first control action of the optimal sequence is applied to the physical system. This includes updating the setpoints of generators for the next time step.

The following algorithm illustrates how each generator applies the control update by computing the power ramp to be executed in the current time step:

Algorithm 3 Online Setpoint Update using ADMM

for each generator k **do**
  Apply control ramp: $u_k(t) = P_{t+1}^{\text{gen}_k} - P_t^{\text{gen}_k}$.
  Send updated setpoint $P_{t+1}^{\text{gen}_k}$ to the generator's governor.
end for

This way, the agents are able to solve the local problem and send the control actions to the ANDES simulator, which would then provide the system operating conditions for the next time step.

References

[1] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed mpc strategies with application to power system automatic generation control," *IEEE transactions on control systems technology*, vol. 16, no. 6, pp. 1192–1206, 2008.

[2] A. K. Roy and S. K. G. Jain, "Contingency analysis in power system," Ph.D. dissertation, 2011.

[3] M. Marzband, M. M. Moghaddam, M. F. Akorede, and G. Khomeyrani, "Adaptive load shedding scheme for frequency stability enhancement in microgrids," *Electric Power Systems Research*, vol. 140, pp. 78–86, 2016.

[4] D. B. Aeggegn, A. O. Salau, and Y. Gebru, "Load flow and contingency analysis for transmission line outage," *Archives of Electrical Engineering*, pp. 581–594, 2020.

[5] W. W. Price, K. A. Wirgau, A. Murdoch, J. V. Mitsche, E. Vaahedi, and M. El-Kady, "Load modeling for power flow and transient stability computer studies," *IEEE Transactions on Power Systems*, vol. 3, no. 1, pp. 180–187, 2002.

[6] R. J. Campbell and S. Lowry, "Weather-related power outages and electric system resiliency," Congressional Research Service, Library of Congress Washington, DC, 2012.

[7] W. Li, E. Vaahedi, and P. Choudhury, "Power system equipment aging," *IEEE Power and Energy Magazine*, vol. 4, no. 3, pp. 52–58, 2006.

[8] P. Kundur, "Power system stability," *Power system stability and control*, vol. 10, no. 1, pp. 7–1, 2007.

[9] Q. Qiu, R. Ma, J. Kurths, and M. Zhan, "Swing equation in power systems: Approximate analytical solution and bifurcation curve estimate," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 1, 2020.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. [Online]. Available: `https://web.stanford.edu/~boyd/papers/pdf/admm_distr_stats.pdf`.

[11] R. R. Negenborn, "Model predictive control for the management of large-scale power systems," Accessed: 2025-06-25, Ph.D. dissertation, Delft University of Technology, 2007. [Online]. Available: `https://www.dcsc.tudelft.nl/~bdeschutter/research/phd_theses/phd_negenborn_2007.pdf`.

[12] A. Bettoni, G. Mastroddi, and M. Muttoni, "Distributed nonlinear model predictive control for district heating networks," 2021.

[13] COIN-OR, *IPOPT - interior point optimizer*, `https://github.com/coin-or/Ipopt`, Accessed: 2025-07-08.

[14] T. Athay, R. Podmore, and S. Virmani, "A practical method for the direct analysis of transient stability," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, 1979.

[15] H. Cui, "Andes, model references," 2022. Accessed: Aug. 13, 2024. [Online]. Available: `https://docs.andes.app/en/latest/modelref.html`.

[16] eRoots Analytics, *Colmena: Decentralized control framework for power grids*, `https://github.com/eRoots-Analytics/COLMENA`, Accessed: 2025-06-03, 2024.

[17] P. Olivella-Rosell, F. Rullan, P. Lloret-Gallego, and E. Prieto-Araujo, "Centralised and distributed optimization for aggregated flexibility services provision," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3551–3565, 2019. DOI: `10.1109/TSG.2018.2827684`. [Online]. Available: `https://ieeexplore.ieee.org/document/8359283`.

[18] Q. Huang, "Rlgc repository," 2022. Accessed: Aug. 13, 2024. [Online]. Available: `https://github.com/RLGC-Project/RLGC`.

[19] U. of Washington, "Power systems test case archive," 2022. Accessed: Aug. 13, 2024. [Online]. Available: `https://labs.ece.uw.edu/pstca/`.

[20] U. of Washington, "Power systems test case archive," 2022. Accessed: Aug. 13, 2024. [Online]. Available: `https://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm`.

[21] U. of Washington, "Power systems test case archive," 2022. Accessed: Aug. 13, 2024. [Online]. Available: `https://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm`.

[22] Q. Huang, "Rlgc repository," 2022. Accessed: Aug. 13, 2024. [Online]. Available: `https://github.com/RLGC-Project/RLGC`.

[23] S. Qi and F. Milano, "Andes: A python tool for power system modeling and analysis," *SoftwareX*, vol. 11, p. 100 453, 2020.

[24] G. C. Ejebe and B. F. Wollenberg, "Generation contingency analysis for power system security assessment," *IEEE Transactions on Power Apparatus and Systems*, vol. 98, no. 2, pp. 606–617, 1979.

[25] G. C. Ejebe and B. F. Wollenberg, "Transmission contingency analysis for power system security assessment," *IEEE Transactions on Power Apparatus and Systems*, vol. 98, no. 2, pp. 618–627, 1979.

[26] R. Billinton and R. N. Allan, "Equipment failures and their impact on power system reliability," *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 672–680, 1988.