Electrical grid simulation in the COLMENA framework

Pablo de Juan Vela [1]
[1]eRoots, Barcelona, Spain

June 11, 2025

1 Introduction

COLMENA aims to enable decentralized decision-making by coordinating autonomous agents that perform specific roles. Each agent can execute specific roles to change the system and operate based on local measurements and predictions. This project uses COLMENA to deploy a multi agent control system in where each agent is responsible for the control of a specific area. The different agents use the tools available through COLMENA to ensure proper coordination for completing the goal of controlling the frequency throughout the grid.

Distributed Model Predictive Control (MPC) is a particularly well-suited control strategy in this context. 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 will first explain the physical models and equations used to build the MPC, we will then explain the optimisation background used for to implement the Distributed MPC, and finally we will use the MPC control in the grid [1] in ANDES to showcase its capability for frequency control.

2 Modeling

Accurate modeling of the frequency and angle dynamics of each area is fundamental to the design and implementation of distributed control strategies in power systems. These dynamic variables capture how each area responds to generation-demand imbalances and how it interacts electrically with neighboring regions through phase angle differences. By understanding these behaviors, the proposed MPC can anticipate future states and choose the appropriate controls actions.

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 a version of the swing equation:

$$M_i \frac{df_i(t)}{dt} = -D_i(f_i(t) - f_0) + \sum_{k \in \mathrm{Area}_i} P_t^{\mathrm{gen}_k} - \sum_{j \in \mathcal{N}_i} P_t^{i,j} - P_i^{\mathrm{demand}}(t) \tag{1}$$

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),
- $P_t^{\mathrm{gen}_k}$ 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:

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

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, this shows that active power flows are driven by angle differences and line susceptance $B_{i,j}$.:

$$P_t^{i,j} = B_{i,j}(\delta_i(t) - \delta_j(t)) \tag{3}$$

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:

$$B_{i,j} = \sum_{\text{line i connects area i to area j}} b_{\text{line i}} \tag{4}$$

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 optimisation.

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.

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.
- **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 roles 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 directly 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:

$$u_i(t) = P_i^{\text{gen}}(t+1) - P_i^{\text{gen}}(t) \in \mathbb{R}^{n_{g,i}} \tag{5}$$

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.

The control inputs are subject to ramp-rate constraints that limit how fast the generators can increase or decrease their output:

$$u_k^{\min} \leq u_{i,k}(t)\Delta t \leq u_k^{\max} \quad \forall k \in \{1, \ldots, n_{g,i}\}, \forall t \tag{6}$$

where $u_k^{\min}$ and $u_k^{\max}$ are the ramp-down and ramp-up limits (in pu/s) 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:

$$f_{i,t+1} = f_{i,t}(x_t) \tag{7}$$

More explicitly, using the physical model:

$$f_{i,t+1} = f_{i,t} + \frac{\Delta t}{M_i}\Big[- D_i(f_{i,t} - f_0) + \sum_{k \in \mathcal{G}_i} P_t^{\mathrm{gen}_k} - P_{i,t}^{\mathrm{demand}} + \sum_{j \in \mathcal{N}_i} B_{j,i}(\delta_{j,t} - \delta_{i,t})\Big] \tag{8}$$

$$=: f_{i,t}(x_t) \tag{9}$$

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:

$$\delta_{i,t+1} = \delta_{i,t} + 2\pi\Delta t(f_{i,t} - f_0) \tag{10}$$

Global Optimization Problem:

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

$$\min_{\{u_{i,t}\}} \quad \sum_{i}^{N}\sum_{t=0}^{T}\left(f_{i,t}(x_t) - f_0\right)^2 \tag{11}$$

Subject to:

$$\delta_{i,t+1} = \delta_{i,t} + 2\pi\Delta t(f_{i,t}(x_t) - f_0) \tag{12}$$
$$x_{t+1} = \text{State transition function dependent on controls and dynamics} \tag{13}$$
$$u_k^{\min} \leq u_{i,t}^{(k)} \leq u_k^{\max} \tag{14}$$
$$u_k^{\min} \leq u_{i,t+1}^{(k)} - u_{i,t}^{(k)} \leq u_k^{\max} \tag{15}$$

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.

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

With this we can redefine the global problem as follows:

$$\min_{\{u_{i,t}\}} \quad \sum_{i=1}^{N}\sum_{t=0}^{T}\left(f_{i,t}(x_{i,t}) - f_0\right)^2 \tag{16}$$
$$\text{s.t.} \quad x_{i,t} = x_{j,t}, \quad \forall (i,j) \text{ areas s.t. } B_{i,j} \neq 0,\ \forall t \tag{17}$$

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.

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.

3.2 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.

$$\begin{aligned}\mathcal{L}_i = \sum_{t=0}^{T}(f_{i,t} - f_0)^2 + \sum_{j\in\mathcal{N}_i}\sum_{t=0}^{T}\Big[\lambda_{i,j,t}(\delta_{i,t} - \hat{\delta}_{j,i,t}) + \lambda_{j,i,t}(\delta_{j,t} - \hat{\delta}_{i,j,t})\Big] \\ + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\sum_{t=0}^{T}\Big[(\delta_{i,t} - \hat{\delta}_{j,i,t})^2 + (\delta_{j,t} - \hat{\delta}_{i,j,t})^2\Big]\end{aligned} \tag{18}$$

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:

$$\begin{aligned}\min_{\{u_{i,t}\}} \quad & \sum_{t=0}^{T}(f_{i,t} - f_0)^2 \\ & + \sum_{j\in\mathcal{N}_i}\Big[\lambda_{j,i,t}(\delta_{j,t} - \hat{\delta}_{i,j,t}) + \lambda_{j,i,t}(\widehat{\delta_{j,t}} - \hat{\delta}_{i,j,t})\cdot\Big] \\ & + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\sum_{t=0}^{T}\Big[(\delta_{i,t} - \hat{\delta}_{j,i,t})^2 + (\delta_{j,t} - \hat{\delta}_{i,j,t})^2\Big]\end{aligned} \tag{19}$$

Subject to the following local constraints:

$$u_k^{\min} \leq P_{t+1}^{\mathrm{gen}_k} - P_t^{\mathrm{gen}_k} \leq u_k^{\max} \quad \forall k \in \mathcal{G}_i,\ \forall t \tag{20}$$

$$P_{\min}^{\mathrm{gen}_k} \leq P_t^{\mathrm{gen}_k} \leq P_{\max}^{\mathrm{gen}_k} \quad \forall k \in \mathcal{G}_i,\ \forall t \tag{21}$$

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.

Local Problem Modifications

A. Damping Term We addan 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.

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

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 stabilize convergence and avoids scillations in the horizon state.

B. Adaptable ρ Instead of using a fixed penalty parameter ρ, we adaptively 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. The update rule is:

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

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:

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

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

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 $\gamma = 10^9$.

$$\gamma||f_{i,t} - f_{nom}||_2^2 \tag{22}$$

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 Finally, 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:

$$\alpha_1 \sum_{0 \le t < T} (f_{i,t+1} - f_{i,t})^2 \tag{23}$$

And a Terminal cost term that we define as follows:

$$\alpha_2 (f_{i,T} - f_{nom})^2 \tag{24}$$

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

3.3 ADMM Algorithm

Algorithm 1 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 2 Online Setpoint Update using ADMM

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

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. 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 stability is unusual, indicated by the associated KPI being broken, 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 controllable loads) over a prediction horizon. The optimization prioritizes frequency recovery while incorporating ramp limits and operational constraints.

With 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 frequency deviation, the agent optimized for economic dispatch over a longer time scale (e.g., 10 minutes), the same underlying MPC engine would constitute a completely different role, such as 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.

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., $\lvert f_i - f_0 \rvert > \Delta f_{\max}$ for any agent i in the area.
Scope	Local to an area, with coordination across neighboring agents.
Agent Requirement	Computational Power, Specific Solver
Time Scale	Fast-acting (typically sub-minute) response to stabilize frequency
Key Dependencies	Access to local state measurements (frequency, generation), 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 3 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

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

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 belong to 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 example, 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 each colmena agent is associated with their

Data Decorators

We define 3 different types of data channels to share information between the different agents running the ADMM:

- State Horizon Data: Channel where the copies of all the coupled primal variables are stored, 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. This 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 reached convergence. Then the agents send the control actions to the ANDES simulation and wait until the next step.

Coordination

The different agents read this channel to get the values for the primal variables and the copies of the primal variables and then post their updates primal values. To avoid agents overwriting the changes of other agents we also add a coordination feature. This consists in the different agents having to update the data channels in a specific order. We add this coordination requirement to both the Error and State Horizon data channels. We also considered defining a new coordinating agent to perform this task that receives the different agents messages but decided against it.

5 Case Study and Results

We consider the IEEE39 **grids:ieee39** grid to perform multiple case studies and see how the COLMENA frameworks performs in controlling the frequency. The grid consists of 39 buses, 10 generators, 14 lines and others electrical devices responsible for the electrical control. The line is simulated in python using the open source package ANDES [2]. This grid is divided in two different areas of similar size, where each bus belongs to one area or the other. 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 set up to receive queries to from COLMENA on the state on the grid, akin to making measurement of specific metrics on the grid. And it can also change the parameters of the simulation similar to a role acting on the grid.

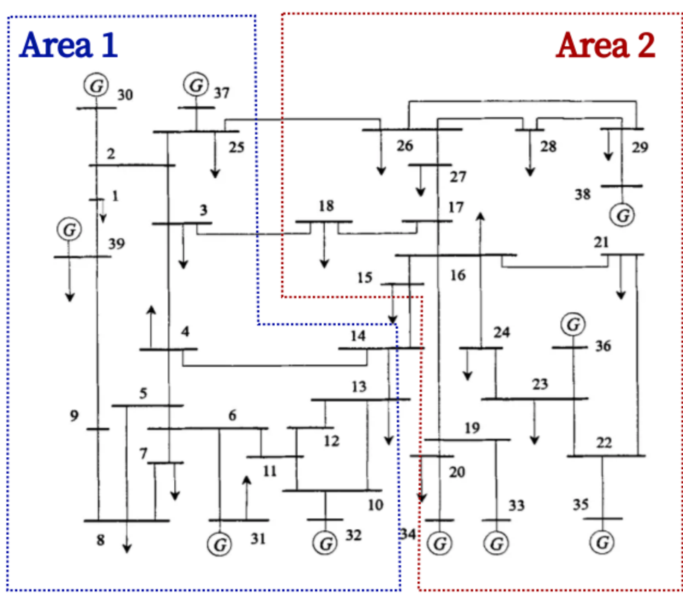

Figure 1: IEEE39 2 area diagram.

From the COLMENA side, we deploy multiple agents as docker containers deployed in a single computer. We deploy at 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 ('generator', 'solver')
- Agent 2: Agent with requirement ('converter')

On the simulation side, we start the simulation with a grid in steady state conditions. Then, we define a perturbation at $t = 2s$ that consists in a line tripping. This perturbation provokes a change in frequency. We set up multiple scenarios with the objective of seeing how the different roles act in the grid to counteract this perturbation.

5.1 Results

6 Milestone Analysis

Milestone 1

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 [3].

Milestone 2

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.

7 Future Work

References

[1] 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.

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

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