



**COMILLAS**  
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS  
INDUSTRIALES

TRABAJO FIN DE GRADO

*IMPLEMENTATION OF VOLTAGE REGULATION CONTROL SCHEME IN  
DISTRIBUTED COMPUTING PLATFORM*

*Autor: Diego Begara Bretón*

*Director: Alejandro Dominguez-García*

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
*Implementation of voltage regulation control scheme in distributed computing  
platform*

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico cuarto es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni  
total ni parcialmente y la información que ha sido tomada

de otros documentos está debidamente referenciada.

Fdo.: Diego Begara Breton Fecha: 09/06/2020



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Alejandro Domínguez-García Fecha: 09/06/2020



**COMILLAS**  
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS  
INDUSTRIALES

TRABAJO FIN DE GRADO

*IMPLEMENTATION OF VOLTAGE REGULATION CONTROL SCHEME IN  
DISTRIBUTED COMPUTING PLATFORM*

*Autor: Diego Begara Bretón*

*Director: Alejandro Dominguez-García*

Madrid

# **Acknowledgements**

I would like to thank Professor Dominguez and the University of Illinois at Urbana-Champaign (UIUC) very much for giving me the opportunity to work in this project that will continue and will make an impact on the way we control microgrids. After this project I will be able to say how much I learned and how I was able to help.

I would also like to thank Olaoluwapo Ajala for his help throughout this project, since without his help this would have not been possible. With his knowledge and patience I have been able to be here today presenting this paper.

Finally, I would like to thank the rest of team involved in the project: A. Mayoral, P. Samper, N. Siddartha and M. Zholbaryssov.

# IMPLEMENTACIÓN DEL ESQUEMA DE CONTROL DE REGULACIÓN DE TENSIÓN EN PLATAFORMAS INFORMÁTICAS DISTRIBUIDAS

**Autor: Begara Bretón, Diego.**

Director: Domínguez-García, Alejandro.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

## RESUMEN DEL PROYECTO

Las redes modernas se construyen siguiendo una arquitectura centralizada que se basa en una comunicación constante entre cada nodo con el controlador. El controlador regulará la red con la información obtenida de los nudos. El enfoque que estamos tomando a lo largo de este proyecto es implementar una arquitectura distribuida. En esta arquitectura eliminamos el nodo controlador. Lo sustituiremos por controladores más simples, que en este caso serán placas de Arduino, en cada uno de los nodos. Estos Arduinos se encargarán de la computación y la comunicación unos con otros. Los algoritmos que regulan la red se ejecutarán en los Arduinos. Funcionarán como una red neuronal en una estructura de colaboración.

Esta parte del proyecto se basa en la implementación del algoritmo de control de tensión en el código de Arduino que se implementará en la red. Este código será probado con el simulador de redes Typhoon HIL en uno de los laboratorios de la Universidad de Illinois en Urbana-Champaign. El objetivo es crear el código para que pueda funcionar con cualquier red posible que queramos simular. Como objetivos complementarios intentaré mejorar la robustez de los algoritmos y mejorar las funciones para acortar el tiempo de cálculo.

Para las pruebas hemos implementado el código en un ejemplo de 6 nodos. Comparando los resultados obtenidos podemos asegurar que el mejor enfoque para este caso es implementar una arquitectura distribuida que dará un tiempo de recuperación más rápido y fiable.

**Palabras clave:** Arduino, Centralizado, Control, Distribuido, Microrred, Typhoon, Tensión, Programación

## 1. Introducción

Según el Departamento de Energía de los Estados Unidos y su Iniciativa de Red Inteligente y las Redes Eléctricas del Futuro, se está produciendo una gran transformación en el departamento de distribución de energía [1]. Esto viene como resultado de un aumento de los Recursos Energéticos Distribuidos (DER) que se caracterizan por plantear dos retos. El primero son las curvas variables de potencia, ya que como podemos ver en Figura 1 y Figura 2, tanto la generación como el consumo plantean un reto a controlar debido a esta no linealidad. Como resultado del desafío anterior, la necesidad de controlar estos componentes es imperativa. Aquí es donde se introduce el segundo reto, estos recursos son demasiado pequeños para que la red los regule, como lo haría con elementos de consumo y generación tradicionales. Por lo tanto, necesitamos una arquitectura de control específica que sea capaz de controlarla. Para ello necesitamos introducir el concepto de una microrred.

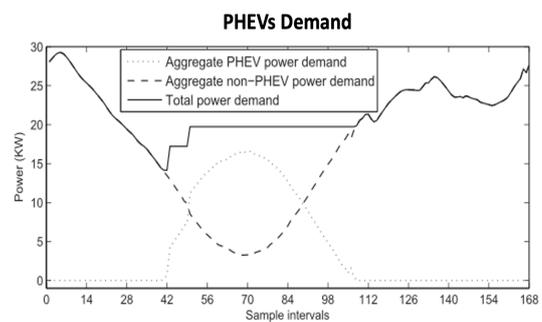
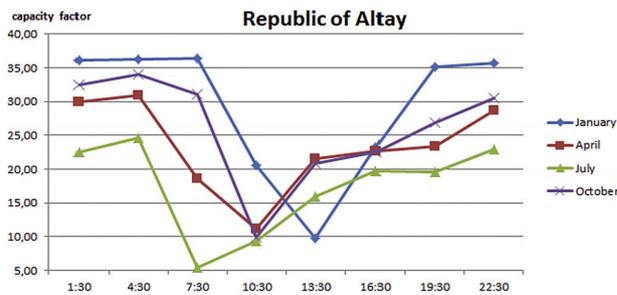


Figura 1: Curva de Generación de una Instalación PV[2]

Figura 2: Curva de Carga de Vehículo Eléctrico[3]

Las microrredes son grupos de cargas y microgeneradores que funcionan como un sistema singular controlable que proporciona energía a la zona. A lo largo de este proyecto vamos a estudiar la importancia y la viabilidad económica que supone su implementación. Para controlar las microrredes tenemos que elegir una arquitectura de control, que puede ser centralizada o distribuida. Por un lado, tenemos las arquitecturas de control centralizadas, que son las más utilizadas. Esto se debe a su facilidad de instalación y a los algoritmos más simples que se utilizan para regular la red. La principal desventaja de este enfoque es la vulnerabilidad frente a fallos de controlador, lo que implica que, si el controlador falla, la red va a funcionar sin regulación hasta que se repare. Por otro lado, tenemos las redes distribuidas. La principal diferencia con las arquitecturas centralizadas es el concepto de control. Reemplazan al controlador central e implementan controladores más pequeños en cada uno de los DER. Estos controladores actuarán como una malla, computando algoritmos iterativos para regular la red.

## **2. Definición del Proyecto**

Este proyecto surge de la búsqueda de desarrollar una microrred distribuida. Este es un proyecto en colaboración con uno de los equipos de investigación de la Universidad de Illinois en Urbana Champaign (UIUC). El proyecto aboga por el desarrollo de una serie de algoritmos de control que tienen como objetivo la simulación y el control de una microrred, en este caso una simulada. Estos algoritmos no sólo incluyen los algoritmos utilizados para el control de la red, como el control de tensión, el control de la frecuencia o el flujo óptimo de energía. Pero también incluyen todos los pasos requeridos para la regulación de las redes, desde la comunicación inalámbrica entre los nodos hasta la sincronización y la inicialización de los procedimientos.

Este trabajo se basa en la implementación del algoritmo para el control de tensión definido en [1]. La implementación se desarrollará en código programado en C++. Este código será utilizado más tarde por Arduinos para controlar una red simulada por el simulador virtual HIL.

A lo largo de este documento vamos a justificar la implantación de microrredes en el paradigma moderno de los sistemas de energía. Esto va a ser apoyado por un análisis económico que determinará la viabilidad de la implementación de las microrredes en la red eléctrica existente que tenemos hoy en día. Una vez decidida la implantación, vamos a explicar las dos posibles arquitecturas de control que se pueden implementar en las microrredes, la centralizada y la distribuida. Ambos enfoques se van a comparar desde el punto de vista de la fiabilidad y de la calidad de respuesta.

## **3. Descripción del modelo**

Para probar ambas arquitecturas de control a través de la implementación del algoritmo de control de tensión y definir el mejor enfoque para los sistemas de control de microrredes, necesitamos proporcionar un escenario de prueba en el que llevarlo a cabo. Hemos desarrollado una arquitectura radial de 6 nodos, la cual puede verse en la Figura 3. A partir de las ecuaciones de flujo de potencia hemos visto cómo las desviaciones de tensión son soportadas por la red al introducir energía reactiva. Los algoritmos que controlarán la salida de la potencia reactiva serán diferentes para cada uno de los enfoques.

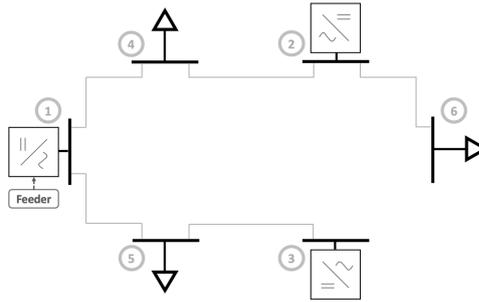


Figura 3: Diseño del Modelo de 6 Nodos

Los dos algoritmos, centralizados y distribuidos, comparten una primera etapa que es proporcional al factor de participación, la sensibilidad del nodo y el tamaño de la violación de tensión como sigue[1]:

$$\rho_k[r] = \frac{\alpha}{S_{jj}} (V_j - V_j[r])$$

$$\rho_k[r] = \frac{\alpha}{S_{jj}} (\bar{V}_j - V_j[r])$$

Para la segunda etapa, el enfoque centralizado es más simple puesto que se dispone de toda la información de los nodos. Por lo tanto, la 2ª etapa centralizada se divide según la reserva de potencia reactiva de los nodos. Sin embargo, para la arquitectura distribuida, la segunda etapa se calcula con un algoritmo iterativo que se llama ratio consensus {8.1} para obtener la cantidad de potencia reactiva que los DER vecinos tienen que suministrar

#### 4. Resultados

Una vez calculadas las simulaciones obtenemos los resultados en el instante de tiempo  $t=280$ , que va a representar el momento en el que todos los elementos de la red están conectados y con el primer cambio de carga en la Carga-1. El gráfico 1 y el gráfico 2 muestran los resultados para el caso centralizado. El gráfico 3 y el gráfico 4 muestran los resultados para el caso distribuido.

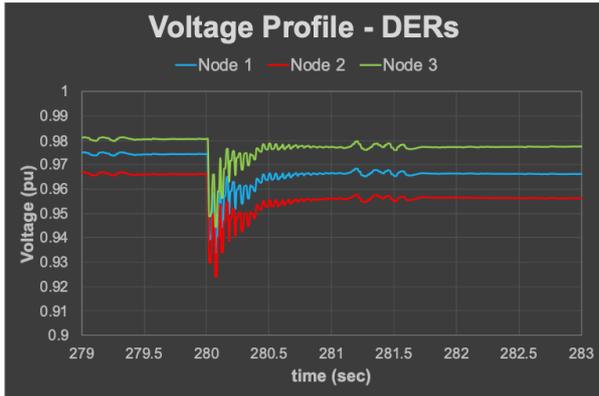


Gráfico 1: Perfil de Tensión en los DER (centralizado)

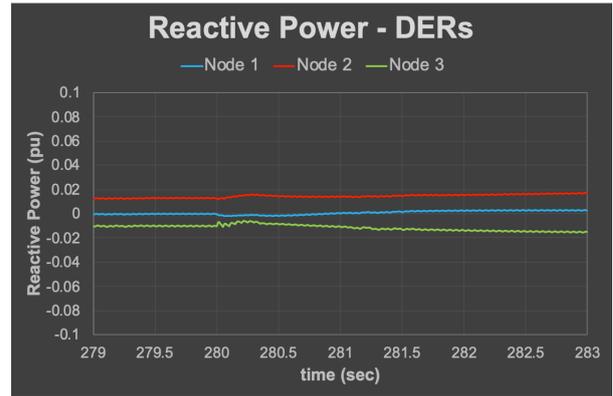


Gráfico 2: Potencia Reactiva en los DER (centralizado)

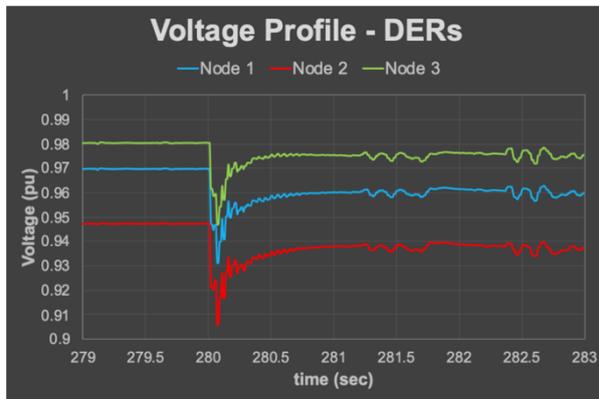


Gráfico 3: : Perfil de Tensión en los DER (distribuido)

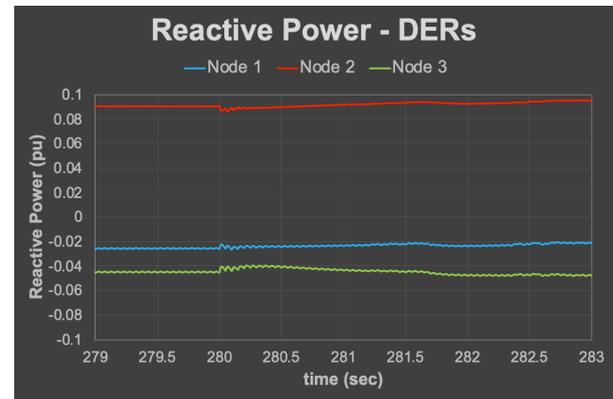


Gráfico 4: Potencia Reactiva en los DER (distribuido)

Vamos a analizar los resultados de acuerdo con cuatro características:

1. Velocidad de respuesta: Debido a sus cálculos más rápidos, el control centralizado tendrá una respuesta más rápida cada iteración, pero la velocidad es tan rápida que no deja al sistema el tiempo suficiente para responder al cambio en la potencia reactiva, lo que se traduce en un sistema oscilante más lento. Por otro lado, los algoritmos iterativos en el control distribuido lo hacen más lento en cada iteración, pero deja tiempo para que el sistema responda. Lo que resulta en una respuesta más irregular pero más rápida.
2. Esfuerzo del sistema: Si consideramos las formas de ambas respuestas, podemos ver cómo el distribuido tiene un mayor impacto inicial en el sistema debido a una gran variación de tensión, pero es seguido por pasos de tensión más pequeños que crearán

una curva de recuperación más suave. Por el contrario, el perfil de tensión centralizado tiene un impacto inicial menor, pero en cada iteración, la desviación de tensión es mayor que la de la distribuida

3. Respuesta general: Si consideramos el enfoque centralizado, podemos ver cómo la forma general y la desviación es constante a través de los tres nodos mientras que, en el sistema distribuido, el nodo 2 está más afectado por la perturbación. Esto tiene que ver con la distancia más cercana del nodo 2 a la carga 1 y el enfoque distribuido, que depende más de los controles locales.
4. Precisión: Debido a las arquitecturas de control, el enfoque centralizado tiene una mayor precisión sobre el distribuido debido a que tiene toda la información disponible. Esto se puede ver en los gráficos del perfil de tensión. El enfoque centralizado tiene una desviación mucho menor en la tensión considerando antes y después del evento.

Para concluir, el enfoque distribuido proporciona una respuesta, que, a pesar de ser menos precisa, es más rápida, más sutil y suave. Estas son las razones por las que pensamos que este es el mejor enfoque que seguir en este tipo de microrredes.

## **5. Conclusiones**

Las microrredes simuladas son el futuro de los proyectos de investigación de energía y de las pruebas de microrredes. Cada vez hay más programas de investigación que utilizan este método, lo que dará lugar a un mayor número de modelos normalizados que acelerarán las pruebas e investigaciones manteniendo los costes al mínimo. A lo largo de este trabajo hemos estudiado la implementación del algoritmo de control de tensión para ambas arquitecturas de control. Esto nos ha permitido comparar ambas para sacar conclusiones sobre la viabilidad de implementar microrredes distribuidas en lugar de centralizadas.

Como hemos visto a lo largo del análisis de resultados para la respuesta de control de tensión, el mejor enfoque es implementar una microrred cuya arquitectura de control sea distribuida. Esto se debe a que el control va a tener una respuesta más estable, menos oscilante y con un tiempo de recuperación más rápido. El análisis de resultados, aunque no es demasiado diferenciador, ha indicado la ligera superioridad del enfoque distribuido frente a la

arquitectura centralizada que se está utilizando actualmente en las microrredes de todo el mundo. Hemos visto cómo la mayor precisión de los modelos centralizados se ha visto reducida debido a su escasa estabilidad y a la velocidad de respuesta general.

Desde un punto de vista económico, este análisis ha demostrado la viabilidad del cambio de la configuración tradicional de la red a un enfoque de microrredes. Para el caso hemos estudiado el cambio para una pequeña zona industrial que se define en la figura 24. Una vez establecidas las hipótesis y realizado el desglose económico, hemos obtenido un VAN positivo, lo que significa que la inversión para el escenario de prueba es rentable. A partir de este análisis hemos visto cómo la implantación de las microrredes no sólo es el enfoque que nos conducirá a un futuro renovable según los cambios en la política y las iniciativas verdes, pero el que generará más riqueza.

Desde el punto de vista de la fiabilidad, debemos tener en cuenta el fallo de los controladores. Un enfoque distribuido va a proporcionar una fiabilidad superior. Esto se debe al hecho de que un enfoque centralizado tiene una vulnerabilidad de controlador de punto único, lo que hace que el sistema sea muy susceptible a los fallos. Si el controlador falla, la microrred no estará regulada hasta que se repare el mismo. Sin embargo, si implementamos una arquitectura distribuida, debido a la arquitectura de control distribuido, que requiere implementar un controlador en cada uno de los DER, vamos a tener una malla de control que puede dar cuenta de múltiples fallos simultáneos gracias a los algoritmos iterativos y las predicciones que pueden dar cuenta del mal funcionamiento de los nodos o de los problemas de comunicación. Este enfoque va a proporcionar una micro malla más fiable y segura.

Teniendo en cuenta las declaraciones anteriores, el enfoque que hemos encontrado que es el más adecuado para nuestro mundo moderno y siempre cambiante es la implementación de microrredes. Esas microrredes tienen que ser probadas usando la simulación de "Hardware In the Loop" para minimizar los costos y el tiempo de simulación. La arquitectura de control que esta microrred debe implementar es un enfoque distribuido que va a ser el más fiable. A su vez proporcionará una respuesta con mayor rapidez, estabilidad y suavidad a las violaciones de tensión.

## **6. Referencias**

[1] B. A. Robbins, C. N. Hadjicostis, and A. D. Domínguez-García, "A two-stage distributed architecture for voltage control in power distribution systems," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1470–1482, 2013.

- [2] B. V. Ermolenko, G. V. Ermolenko, Y. A. Fetisova, and L. N. Proskuryakova, "Wind and solar PV technical potentials: Measurement methodology and assessments for Russia," *Energy*, vol. 137, no. April, pp. 1001–1012, 2017.
- [3] Y. Mou, H. Xing, Z. Lin, and M. Fu, "Decentralized optimal demand-side management for PHEV charging in a smart grid," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 726–736, 2015.

# IMPLEMENTATION OF VOLTAGE REGULATION CONTROL SCHEME IN DISTRIBUTED COMPUTING PLATFORM

**Author: Begara Breton, Diego.**

Supervisor: Domínguez-García, Alejandro.

Collaborating Entity: ICAI-Pontifical University of Comillas

## ABSTRACT

Modern networks are built following a centralized architecture that relies on a constant communication between each node and the controller node. That node will then regulate the network. The approach we are taking throughout this project is implementing a distributed architecture. In this architecture we eliminate the controller node. We will substitute it by simpler controllers, Arduino boards, in each of the nodes. These Arduinos will be in charge of computation and communication with one another. The algorithms that regulate the network will be run in the Arduinos. They will work as a neuronal network in a collaborative structure.

This part of the project is based on the implementation of the voltage control algorithm into the Arduino code that will be implemented in the network. This code is tested with the Typhoon HIL simulator in one of the ECE laboratories. The goal is to code the algorithm so that it can work with any possible network we want to simulate. As complementary goals I will try to improve the robustness of the algorithms and improve functions in order to shorten the computational time.

For testing we have implemented the code into a 6-node example. Comparing the results obtained we can assure that the best approach for this case is implementing a distributed architecture which will give a more reliable faster recovery time.

**Keywords:** Arduino, Centralized, Control, Distributed, Microgrid, Typhoon, Voltage, Programming.

## 1. Introduction

According to the US DOE and their Smart Grid Initiative and the Electricity Networks of the Future, a large transformation is taking place within the power distribution department [1]. We have seen this as a result of an increase in Distributed Energy Resources (DERs) which are characterized by two main challenges. The first one is the power variation, since as we can see on Figure 1 and Figure 2, both generation and consumption set a challenge to control since due to this non-linearity, the network will have to account for them. As a result of the previous challenge the need to control these components is imperative. Here is where the second challenge is introduced, these resources are too small for the network to regulate them, as it would do with any of the traditional ones. Therefore, we need a specific control architecture that would be able to control it. In order to do so we need to introduce the concept of a microgrid.

Microgrids are clusters of loads and micro-sources operating as a singular controllable system that provides power to the area. And throughout this project we are going to study the importance of their implementation and the economic viability they provide. In order to control microgrids we need to choose a control architecture, which can be centralized or distributed. On one side we have centralized control architectures, which are the most commonly used. This is due to their ease of installations and simpler algorithms to regulate the network. The main disadvantage of this approach is the vulnerability to single-point controller failure, which implies that if the controller fails, the grid is going to run unregulated until it is repaired. On the other side we have distributed networks. The main difference with centralized architectures is the control concept. They replace the central controller and implement smaller controllers in each of the DERs. These controllers will act as a mesh, computing iterative algorithms in order to regulate the network.

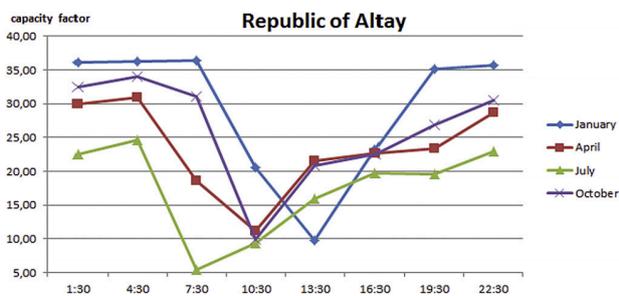


Figure 1: Daily Wind Power Output [2]

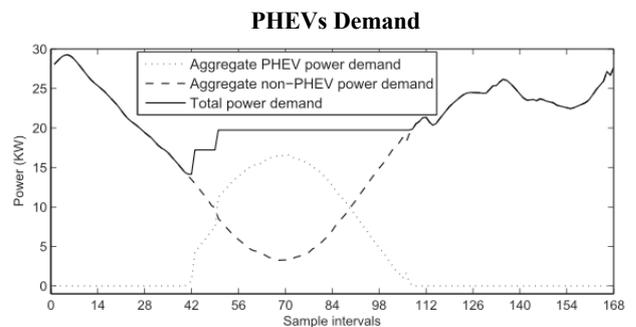


Figure 2: Power Demand from PHEVs [3]

## 2. Project Definition

This project comes from the desire to develop a distributed microgrid. This is a project from one of the research teams in the University of Illinois at Urbana Champaign (UIUC). The project advocates on the development of a number of control algorithms that aim towards the simulation and control of a microgrid, in this case a simulated one. These algorithms not only include control paths, such as voltage control, frequency control or optimal power flow. But also include every step in the regulation of grids, from wireless communication in between nodes to synchronization and initialization of procedures.

This paper is based on the implementation of the algorithm for voltage control defined in [1]. The implementation will be developed in programming code, in C++. This code will later be used by Arduinos in order to control a simulated grid by the virtual HIL simulator.

Throughout this paper we are going to justify the implantation of microgrid into the modern paradigm of power systems. This is going to be supported by an economic analysis that will determine the viability of implementing microgrids into the existing power network we have nowadays. Once decided on the implementation, we are going to explain the two possible control architecture that can be implemented into microgrids, centralized and distributed. Both approaches are going to be compared from a reliability and from a response quality point of view.

## 3. Description of the model

In order to test both control architectures with the implementation of the voltage control algorithm and define the best approach to microgrid control systems we need to provide a test scenario in which to do so. We have developed a 6-node radial architecture, which can be seen in Figure 3. From the power flow equations we have seen how voltage deviations are supported by the network by outputting reactive power. The algorithms that will control the reactive power output will be different for both approaches.

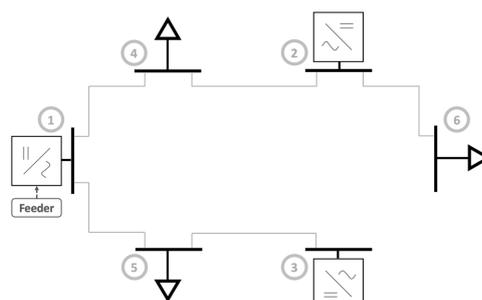


Figure 3: 6-Node Model Diagram

The two algorithms, centralized and distributed, share a 1<sup>st</sup> stage which is proportional to the participation factor, the sensitivity of the node and the size of the voltage violation as follows[1]:

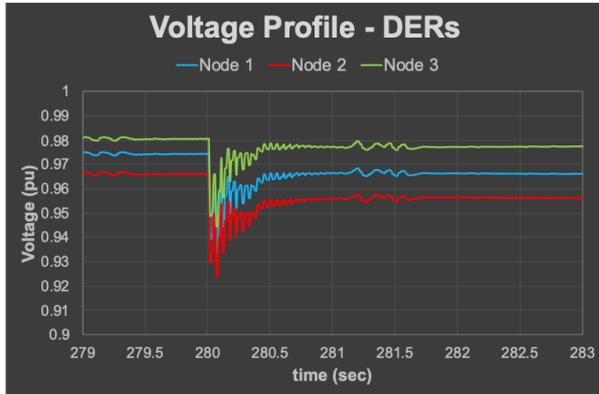
$$\rho_k[r] = \frac{\alpha}{S_{jj}} (\underline{V}_j - V_j[r])$$

$$\rho_k[r] = \frac{\alpha}{S_{jj}} (\bar{V}_j - V_j[r])$$

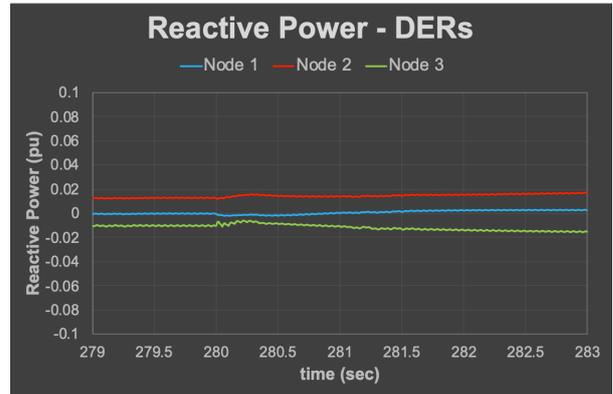
For the second stage, the centralized approach is simpler due to having all the information from the nodes available. Hence, the centralized 2<sup>nd</sup> stage is divided according to the reactive power reserve in the nodes. On the other side, for the distributed architecture, the second stage is computed with an iterative algorithm that is called ratio consensus {8.1} in order to obtain the amount of reactive power that the neighboring DERs have to supply

#### 4. Results

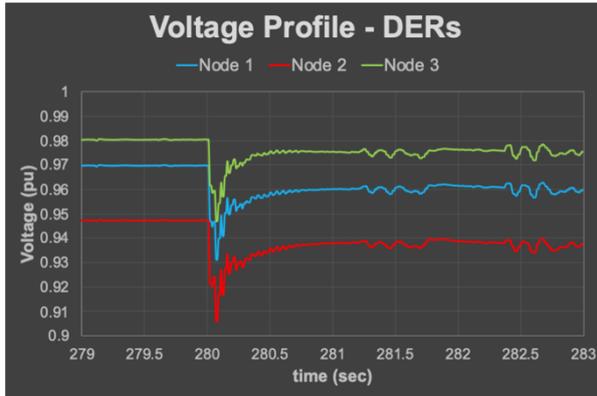
Once the simulations are computed we obtain the results at time t=280, which is going to account for the time instant at the which all the elements in the grid are connected and with the first load change in Load-1. Graph 1 and Graph 2 show the results for the centralized case. Graph 3 and Graph 4 show the results for the distributed case.



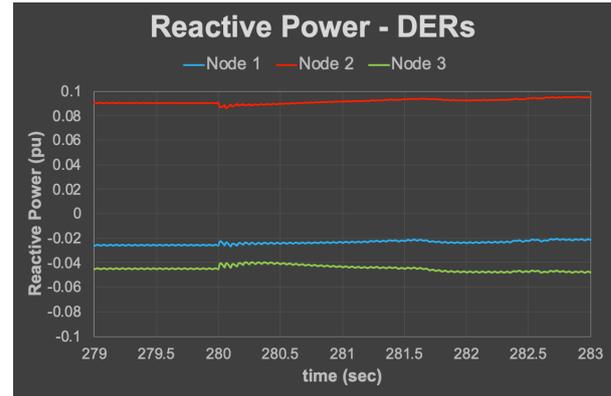
Graph 1: Voltage Profile in the DERs (centralized)



Graph 2: Reactive Power in the DERs (centralized)



Graph 3: Voltage Profile in the DERs (distributed)



Graph 4: Reactive Power in the DERs (distributed)

We are going to analyze the results according to four characteristics:

1. Response Speed: Due to its faster computations, the centralized control will have a faster response each iteration, but the speed is so fast that it does not let the system enough time to respond to the change in reactive power, which translates in a slower oscillating system. On the other side, the iterative algorithms in the distributed control makes it slower are each iteration but leaves time for the system to respond. Which results in a more irregular but faster response.
2. System Strain: If we consider the shapes form both responses, we can see how the distributed has a higher initial strain on the system due to a large voltage variation but is followed by smaller voltage steps that will create a smoother recovery curve. On the contrary, the centralized voltage profile has a lower initial strain, but at each iteration, the voltage deviation is larger than the one from the distributed
3. Overall Response: If we consider the centralized approach, we can see how the overall shape and deviation is constant across the three nodes while on the distributed system, node 2 is more affected by the perturbation. This has to do with the closer distance form node 2 to load 1 and the distributed approach, that relies more on local controls.
4. Precision: Due to the control architectures, the centralized approach has a higher precision over the distributed due to having all the information available. This can be seen on the voltage profile Graphs. The centralized approach has a much smaller deviation in voltage considering before and after the event.

To conclude, the distributed approach provides a faster, more subtle and smoother response while being less precise. These are the reasons why we think this is the better approach to take in these types of microgrids.

## **5. Conclusions**

Simulated microgrids are the future of power research projects and microgrid testing. There is a growing number of research programs that use this method, which will result in a larger number of normalized models that will speed up testing and research while keeping the costs at its minimum. Throughout this paper we have studied the implementation of the voltage control algorithm for both control architectures. This has allowed us to compare both of them in order to draw conclusions regarding the viability of implementing distributed rather than centralized microgrids.

As we have seen throughout the result analysis for the voltage control response, the best approach is implementing a microgrid whose control architecture distributed. This comes from the fact that the control is going to have a more stable, less oscillating and with faster recover time response. The result analysis, while not too differentiating, has indicated the slight superiority of the distributed approach over the centralized architecture that is currently being used in microgrids all around the world. We have seen how the higher accuracy from centralized models has been drawn back due to its poor stability and overall response speed.

From an economic point of view, this analysis has shown the viability of the change from traditional grid configuration to a microgrid approach. For the case we have studied the change for a small industrial area that is defined in Figure 27. Once the assumptions are set and the economic breakdown has been performed, we have obtained a positive NPV, which means that the investment for the test scenario is profitable. From this analysis we have seen how the implementation of microgrids not only is the approach that will lead us to a renewable future, but according to changes in politics and green initiatives, but the one that will generate the most wealth.

From a reliability point of view, we need to take into account controller failure. A distributed approach is going to provide a superior reliability. This is due to the fact that a centralized approach has a single-point vulnerability, which makes the system very susceptible to failure. If the controller fails, the microgrid is unregulated until the controller is repaired. However, if we implement a distributed architecture, due to the distributed control

architecture, that requires to implement a controller in each of the DERs, we are going to have a control mesh that can account for multiple simultaneous failures thanks to iterative algorithms and predictions that can account for node malfunction or communication problems. This approach is going to provide a more reliable and secure microgrid.

Taking into consideration the previous statements, the approach that we have found to be the most adequate for our modern, ever-changing world is the implementation of microgrids. Those microgrids have to be tested using Hardware In the Loop simulation in order to minimize costs and simulation time. The control architecture that this microgrid should implement is a distributed approach that is going to be the most reliable while having a fast, stable and smooth response to voltage violations.

## 6. References

- [1] B. A. Robbins, C. N. Hadjicostis, and A. D. Domínguez-García, “A two-stage distributed architecture for voltage control in power distribution systems,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1470–1482, 2013.
- [2] B. V. Ermolenko, G. V. Ermolenko, Y. A. Fetisova, and L. N. Proskuryakova, “Wind and solar PV technical potentials: Measurement methodology and assessments for Russia,” *Energy*, vol. 137, no. April, pp. 1001–1012, 2017.
- [3] Y. Mou, H. Xing, Z. Lin, and M. Fu, “Decentralized optimal demand-side management for PHEV charging in a smart grid,” *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 726–736, 2015.

# Index

<b>Chapter 1. Introduction.....</b>	<b>9</b>
1.1 Description of the problem .....	9
1.2 Microgrids.....	13
1.3 Centralized Approach .....	15
1.4 Distributed approach.....	17
<b>Chapter 2. Description of the Technology.....</b>	<b>21</b>
2.1 The Team .....	21
2.2 The Laboratory.....	22
2.2.1 Typhoon HIL .....	24
2.2.2 Lower-Level Controller Hardware .....	26
2.2.3 Software .....	28
<b>Chapter 3. Current Standpoint .....</b>	<b>31</b>
3.1 Literature Review.....	31
3.2 Previous Research on the matter.....	31
<b>Chapter 4. Depiction of the Project .....</b>	<b>33</b>
4.1 Justification .....	33
4.2 Objectives .....	34
4.3 Methodology .....	34
<b>Chapter 5. Model Developed.....</b>	<b>37</b>
5.1 Voltage Control Implementation .....	37
5.1.1 Flowchart .....	41
5.2 Distributed Approach.....	42
5.2.1 State Diagram .....	42
5.2.2 Algorithm .....	42
5.2.3 Description of the Code .....	45
5.2.4 Flowchart .....	47
5.3 Centralized Approach .....	48

5.3.1 State Diagram .....	48
5.3.2 Algorithm .....	48
5.3.3 Description of the Code .....	50
5.3.4 Flowchart .....	51
<b>Chapter 6. Results Analysis for 6-Node.....</b>	<b>53</b>
6.1 Layout .....	53
6.2 Simulation .....	56
6.3 Distributed and Centralized Control Results .....	57
6.3.1 Distributed Layout .....	57
6.3.2 Centralized Layout .....	58
6.3.3 Results analysis .....	59
6.4 Centralized VS Distributed Results .....	62
6.5 Conclusions from 6-Node Result Analysis .....	67
<b>Chapter 7. Economic Analysis.....</b>	<b>69</b>
7.1 Introduction.....	69
7.2 Example .....	70
7.2.1 Case Description.....	70
7.2.2 Economic base .....	72
7.2.3 Reliability Analysis.....	75
7.2.4 Investment evaluation.....	77
7.3 Conclusion of the Economic Analysis.....	81
<b>Chapter 8. Algorithms.....</b>	<b>83</b>
8.1 Ratio Consensus .....	83
8.1.1 Description.....	83
8.1.2 Algorithm Test.....	85
8.1.3 Code .....	87
8.2 Maxmin Consensus .....	88
8.2.1 Description.....	89
8.2.2 Algorithm Test.....	90
8.2.3 Conclusion for the Choice of $\varepsilon$ .....	93
8.2.4 Code .....	94

---

<b>Chapter 9. Conclusions and Further Steps .....</b>	<b>95</b>
9.1 Future steps .....	95
9.2 Conclusion .....	96
<b>Chapter 10. References .....</b>	<b>99</b>
<b>ANNEX I. Sustainable Development Goals .....</b>	<b>103</b>
<b>ANNEX II. Distributed Voltage Control .....</b>	<b>107</b>
Classes .....	107
Functions .....	108
<b>ANNEX III. Centralized Voltage Control .....</b>	<b>115</b>
<b>ANNEX IV. Electric Bill Evolution .....</b>	<b>117</b>
<b>ANNEX V. Ratio Consensus Code .....</b>	<b>119</b>
<b>ANNEX VI. Maxmin Consensus Code .....</b>	<b>123</b>

## *Figure Index*

Figure 1: Daily Wind Power Output [2] .....	14
Figure 2: Power Demand from PHEVs [3] .....	14
Figure 3: 6-Node Model Diagram.....	15
Figure 4: Daily Wind Power Output [2] .....	10
Figure 5: Power Demand from PHEVs [3] .....	10
Figure 6: Traditional vs Modern Network [7].....	12
Figure 7: Centralized Network Diagram.....	16
Figure 8: Distributed Network Diagram .....	19
Figure 9: Arduino Layout in the Laboratory .....	23
Figure 10: Typhoon HIL Device.....	24
Figure 11: Schematic Editor for 6-Node.....	25
Figure 12: HIL SCADA for 6-Node .....	26
Figure 13: System-Level Controller in the Laboratory.....	27
Figure 14: Arduino Components[9].....	28
Figure 15: Voltage limits .....	39
Figure 16: Voltage Control Flowchart .....	41
Figure 17: Distributed Actuation diagram .....	42
Figure 18: Distributed Voltage Control Code Flowchart.....	47
Figure 19: Centralized Actuation diagram.....	48
Figure 20: Centralized Voltage Control Code Flowchart .....	51
Figure 21: 6 Node Radial Model.....	54
Figure 22: P-Q Diagram.....	55
Figure 23: 6-Node Sensitivity Matrix .....	56
Figure 24: Typhoon Model of a Variable Load .....	56
Figure 25: Distributed Model.....	58
Figure 26: Centralized model.....	59

---

Figure 27: Economic Model Architecture.....	71
Figure 28: Traditional Reliability Distribution .....	76
Figure 29: Microgrid Reliability Distribution.....	76
Figure 30: Cash Flow Diagram .....	80
Figure 31: Architecture for RC test.....	85
Figure 32: Architecture for maxmin test.....	91
Figure 33: Banshee Network Model .....	96
Figure 34: Goal-7 [22] .....	103
Figure 35: Goal-9 [22] .....	103
Figure 36: Goal-11[22] .....	103
Figure 37: Global Access to Electricity [24].....	104
Figure 38: Class Diagram.....	107

## *Table Index*

Table 1: Line Parameters.....	54
Table 2: Bus Parameters.....	54
Table 3: Timeline of the Simulation .....	57
Table 4: Average Microgrid Cost [13].....	73
Table 5:Initial Investment .....	74
Table 6: Estimated Electric Bill [15]–[17].....	75
Table 7: Yearly NDE Savings.....	78
Table 8: Yearly Income Variations .....	79
Table 9: Yearly Income.....	80
Table 10: Ratio Consensus Test Results .....	86
Table 11: Electric Bill Evolution for Microgrid Scenario.....	117

## *Graph Index*

Graph 1:Voltage Profile in the DERs (centralized) .....	16
Graph 2: Reactive Power in the DERs (centralized).....	16
Graph 3: Voltage Profile in the DERs (distributed).....	17
Graph 4: Reactive Power in the DERs (distributed) .....	17
Graph 5: Active Power Variation in the loads (distributed).....	60
Graph 6: Active Power Variation in the loads (centralized) .....	60
Graph 7: Voltage Profile in the DERs (distributed).....	61
Graph 8: Voltage Profile in the DERs (centralized) .....	61
Graph 9: Reactive Power in the DERs (distributed) .....	61
Graph 10: Reactive Power in the DERs (centralized).....	61
Graph 11: Active Power Variation in Load-1 .....	63
Graph 12: Voltage Profile in the DERs (distributed).....	64
Graph 13: Voltage Profile in the DERs (centralized) .....	64

---

Graph 14: Reactive Power in the DERs (distributed) .....	64
Graph 15: Reactive Power in the DERs (centralized).....	64
Graph 16: Investment Distribution.....	74
Graph 17:Ratio Consensus Test.....	86
Graph 18:maxmin results for $\varepsilon =0.0001$ .....	92
Graph 19:maxmin results for $\varepsilon =0.01$ .....	92
Graph 20:Precision results for $\varepsilon =0.0001$ .....	92
Graph 21:Precision results for $\varepsilon =0.01$ .....	92



## **Chapter 1. INTRODUCTION**

### ***1.1 DESCRIPTION OF THE PROBLEM***

The US Energy Information Administration states that the power system consists of more than 7,300 power plants, nearly 160,000 miles of high-voltage power lines, and millions of low-voltage power lines and distribution transformers, which connect 145 million customers [4]. These numbers indicate the magnitude of the power delivery and consumption network we have nowadays. However, in order to understand the overall size of the grid, we need to dive into the economics. To set the grounds, the average expenditure in electricity per month on American houses is around \$117.65 [5]. That makes a total profit of \$15.75 billion per month just considering housing. With money comes improvement. The electrical market should focus their resources into providing the most reliable and affordable form of electricity. The main goal is to create a network as robust as possible so as to be able to withstand all the possible variations in consumption and generation our modern-electrical society can place. The second goal is to make the network more affordable, not only maximizing cash input, like electric bills, but also minimize losses, that come from blackouts and replacement of hardware.

Currently, we have an extremely complex and secure network, but there is plenty of room for improvement. On average, as shown by S. Massoud Amin [6], there is 92 minutes of power outage per year in the Midwest, and 214 minutes in the Northeast. This data shows how reliable the grid is, since this quantity accounts for a mere 0.03% of the year. This number might seem neglectable, but if you take into consideration the data introduced for the average customer bill introduced previously, the amount of money that is not being input is around thirty million dollars every month. This implies that there is plenty of room for improvement. From the premise that more reliable means more advanced and more money,

there needs to be a desire for improvement towards the most reliable and economic electric system possible.

According to the US DOE Smart Grid Initiative and the Electricity Networks of the future, a large transformation is taking place in the power distribution systems department. This trend is triggered by the rise in variable generation. The variable generation is referred to the implementation of renewable-based power plants like solar or wind farms. These plants are characterized by variable power function curves. An example of this type of curves is represented in Figure 4, shows the power output according to the time of the day in a region in the south of Russia. The second element is the inclusion of Variable Loads or VL. This concept is closely related to devices with the ability to store power, which is the case of plug-in vehicles (PHEVs) or battery storage. As can be seen in Figure 5, these devices demand variable power from the network for an interval of time. This will put more stress in the network and will make it more susceptible to incur in failure conditions. These variable generation and consumption devices are referred to as distributed energy resources or DER [1]. The use of DERs has known advantages such as a constant power factor and disconnection when a fault takes place.

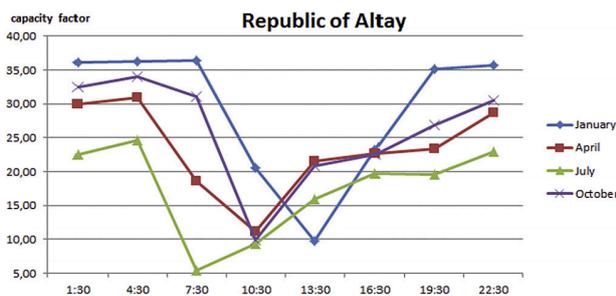


Figure 4: Daily Wind Power Output [2]

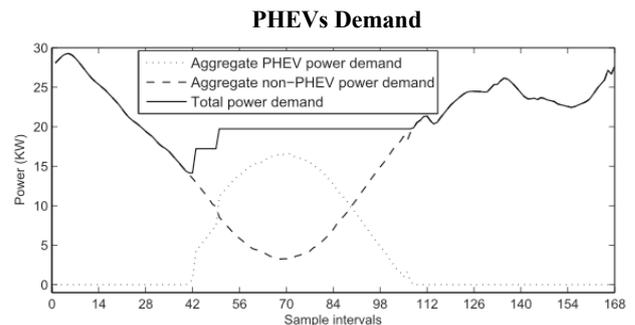


Figure 5: Power Demand from PHEVs [3]

Due to these new concepts in the electrical field, there is a need for change. The traditional network concept is based on a linear structure, as can be seen in Figure 6. It is based on the concept of generation vs load. The process begins by taking into account generation from

power plants, such as coal power plants or wind farms. After that, the power is transported to the costumers. And finally, on the other side, there are loads in which the power will be consumed. These cover from residential areas to hospitals or factories. It is safe to say that this has proven to be a very useful and reliable take on the depiction of networks in the past. Nevertheless, as our technology advances, the way we arrange and use also has to do so too.

The electrical arrange we have nowadays is based on a non-linear structure. The concept of generation and consumption is no longer a direct partition. We can see this concept in the right half of Figure 6. The use of DERs has turned the network into a radial bidirectional scheme. For example, the implementation of DERs has led to the use of batteries that are considered as VLs that take power for future injection into the network. Another use in which VLs are considered is in the charging of PHEVs which, as seen in Figure 5, is characterized by a variable power curve. Second, there are houses with domestic CHP systems (Combined Heat & Power). These can be modeled as a load that will draw not only depending on the time of the day, but also on the power input from the sun. This raises the possibility for homes to draw fewer power from the grid as well as generating and selling power to the grid. This will introduce “new generator elements” into the grid’s scheme.

This prompts us to consider the grid as a neuronal network where each cell has similar functions and responsibilities. But the key conclusion is that they need to work with each other so as to execute their purpose and keep the mesh working in the optimal operation conditions. We have described the behavior of the system and the reason why there needs to be a change. Now a control architecture has to be defined. This control scheme will be in charge of the monitoring and mending of the system. There are two possible control architectures: centralized or distributed.

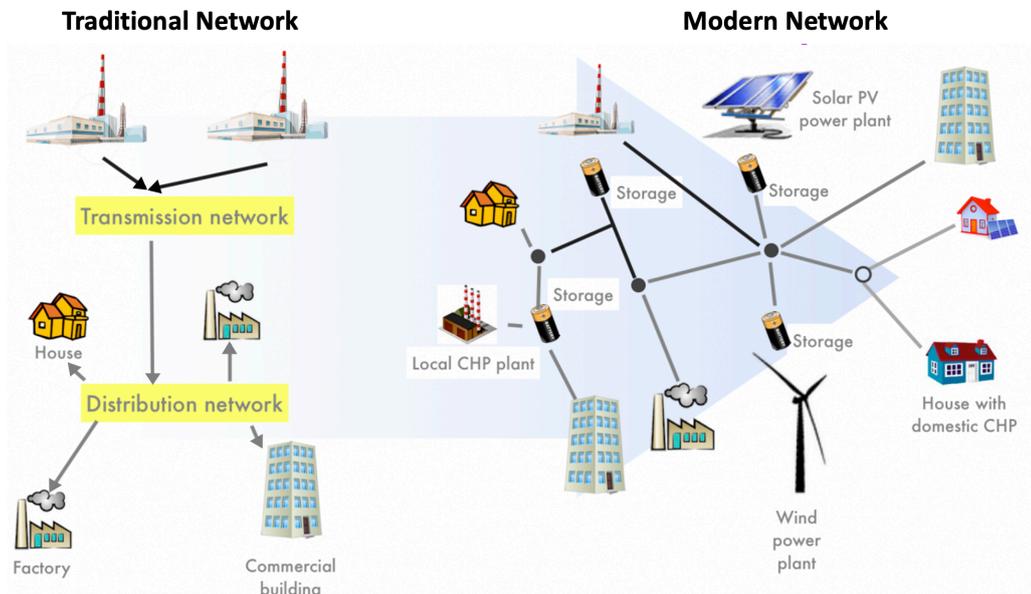


Figure 6: Traditional vs Modern Network [7]

DERs are the future that comes from the injection of renewable energy into the grid, but with them, comes some challenges. The main one is the inability to self-regulate themselves. DERs are too small for the network to be able to regulate them, therefore in order to implement them into existing electric grids we need to be able to regulate them. In order to control these modern technologies, an architectural model has to be chosen. We are going to be studying microgrids. We are choosing the study of microgrids over a traditional generation-transportation-consumption network due to the concepts we are going to be looking at. This is due to the inability of a traditional network to incorporate DERs in the calculations that are processed.

## 1.2 MICROGRIDS

The microgrid concept is established around the concept that modern electrical installations are not linearly built. Microgrids are defined as a cluster of loads and micro-sources operating as a singular controllable system that provides power as well as heat to its local area [8]. We can picture a microgrid as a fragment of the system that takes into account a small area. One example of the same could be a small town that counts with PV installations, wind farms, PHEVs and other installations which can be relevant to the study of electrical networks. The implementation of microgrids has some objectives, such as improving reliability, minimizing losses or avoiding blackouts.

Microgrids have two operating conditions, islanded and grid-connected:

- Grid-Connected: When the system is working with all its DERs and resources connected.
- Islanded: Once there is a failure in the system that leads to a disconnection of a part of the microgrid, the isolated area enters Islanded mode. In this mode, there needs to be at least one DER in order to allow for the system to control itself, confronting voltage and frequency deviation.

Regardless of the mode the microgrid is operating in, it consists of two layers, physical and electronic:

- Physical-Layer: The physical layer is referred to the microgrid, which in the case that we are going to be studying is going to be carried out by the Virtual HIL Device from Typhoon (C-HIL Testbed) {0} and the Texas Instruments MSP-EXP432e401. These two devices will work collaboratively so as to simulate the behavior of the network since testing these algorithms in a real-life microgrid can be very expensive. Apart from the economic, the implementation of experimental algorithms into the microgrid might result in a major failure that can infer in the microgrid not

performing as expected in the future.

- Cyber-Layer: The cyber layer of a microgrid is based on the computational states. It is the layer in charge of computing the algorithms that need to be run in order to evaluate the state of the grid and compute the changes that need to be done in the DERs so that the grid goes back to the rated values. In our case, the cyber layer is described by an Arduino {2.2.2} and the codes that are implemented {2.2.3}.

The implementation of microgrids for research purposes is not the best approach for several reasons. The first reason is the lack of normalized models, since the number of physical microgrids built for research is scarce. Therefore, the implementation of new control software would have rigorous previous testing of the microgrid in order to normalize it. Secondly, after the network is normalized and ready to be tested, the implementation of erroneous algorithms or conditions could result in a grid malfunction and even a condition violation that could damage the measuring instruments and devices. From these statements we can infer how the safest way to do research with microgrids is the utilization of simulated microgrids. The network simulator that is going to be used is the Typhoon HIL device and it is described in {0}.

In order to implement a microgrid we need to implement a control architecture than can be centralized or distributed. In this paper we are going to be comparing both approaches in order to determine the viability of implementing one over the other. The implementation of HIL simulations are going to allow for this comparison thanks to the scalability of the models.

### 1.3 *CENTRALIZED APPROACH*

This is the approach that is being used in microgrid nowadays. It is based on having all the information from the different nodes available at all times. The control architecture is going to be simpler but at the same time more expensive. The centralized architecture is based on three major phases that can be seen in Figure 7.

1. Communication: The first phase is the communication between nodes and controller. It will have as purpose the broadcast of state variables that will describe the operating conditions of the network. Some of the information that is transmitted can be the voltage level, frequency deviation or operational state (to check if there if any node is down).
  2. Processing: The second phase is the interpretation of the information received from the nodes. Once the information is obtained, it will be compared with the reference values. In case any irregularity is detected, the computational stage starts. In this stage several algorithms are conducted in order to determine the optimal conditions that will make the network go back to an operational state.
  3. Broadcast: Once the computational phase is over, we obtain as results the changes that need to be made to the DERs' setpoints in order to obtain the desired conditions in the microgrid. These can be from a change in the DER's reactive or active power input, to the decision to put one of the loads down. These changes are broadcasted to the nodes, who will proceed to follow the directions from the controller.
- Advantages: The main advantage of centralized over distributed systems is the decrease in difficulty of implementation of algorithms in the control architectures. In a centralized architecture every node communicates with the controller device in order to compute the system state. This results in a smaller computation scheme. The

algorithms that need to be implemented in order to restore normality are going to be easier than those of the distributed architecture. Another advantage is the ease of optimization procedures. All the information required for these calculations is available at the controlling node. On the contrary, distributed architectures do not have that information at every controller, therefore making it a more complex calculation. This results in an easier and faster control.

- **Disadvantages:** Due to the implementation of DERs, the probability of node failures increases, and with that increase comes the vulnerability of the controlling node. If the controlling node falls due to a failure in the network, the system will remain uncontrolled until that node is repaired. That is a very expensive operation and makes the network vulnerable to minor failures like overvoltage or frequency deviation. These faults can result in very costly outages or power blackouts when the failure affects the controller node. Centralized approaches are vulnerable to single-point vulnerabilities that would result in an unregulated grid until the controller is repaired.

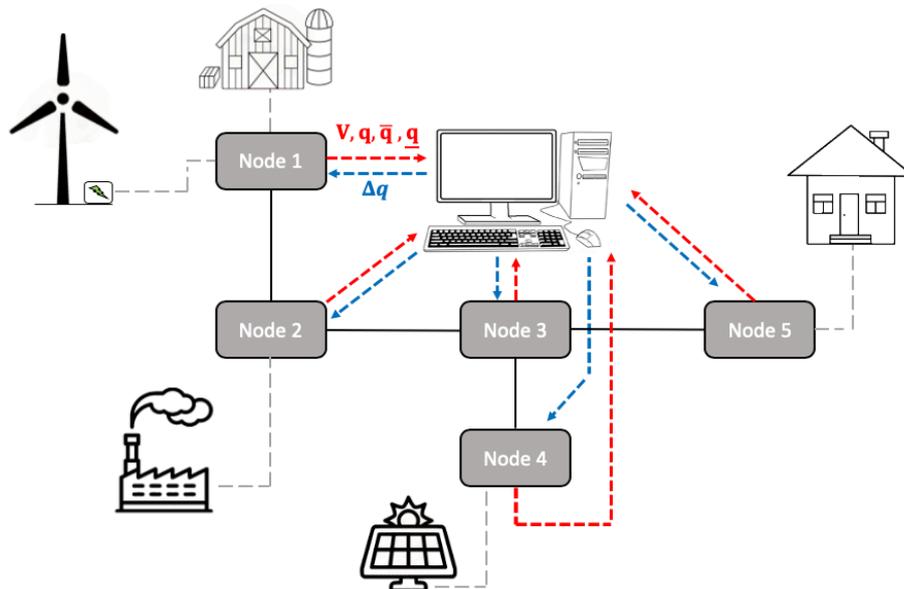


Figure 7: Centralized Network Diagram

## ***1.4 DISTRIBUTED APPROACH***

This approach is based on the collaboration between nodes. We get rid of the controller node that is in charge of the computation. In contrary to the centralized approach, there will be a control module on each of the nodes. This might result in a more expensive overall installation, but in the future, the cost and reliability of the network will make up for the initial investment. This architecture is based on these three phases:

1. Node check: This is the recursive phase in which the node will stay until there is the need to proceed to phase 2, the controller in the node will evaluate the information about the behavior of the same in order to determine whether it operating according to its rated values or if it is experiencing any form of malfunction. If the node finds a violation of the rated values (overvoltage, frequency violation...) it will enter the computational phase of the control.
2. Processing: In this phase the node's controller will implement the algorithms in order to determine the alterations that need to be done. These changes will be analyzed in regard to the available resources in the DERs. If the node is able to fulfil the request by itself, it will proceed to do so. In the case that the amount required of reactive or active power, or any measure required, is superior to the amount available, the node will send a flag to its neighbors. This flag will initiate phase 3 in which the nodes will collaborate to supply the resources needed.
3. Collaborative processing: When one of the nodes cannot provide the required amount of reactive power, the remaining has to be delivered by the rest of the nodes in the network. If the calculation was through as for a centralized network in which one node had all of the information, this would be a very easy computation. In a distributed network, the node will only have its own information and that of its neighbors. In order to obtain the amount each of the nodes has to provide, the nodes

will perform the ratio consensus algorithm, which is described in {8.1}. Once the algorithm is finished and the controllers set the nodes to the appropriate conditions, the system will go back to phase 1.

- Advantages: The main advantage of distributed over centralized systems is the low communication overhead and fast control response. This is related to the speed of the system. Distributed algorithms will have a faster recovery time after a fault. This can be seen in {6.4}. The second advantage distributed algorithms have over centralized architectures is the increased reliability, which can be measured in two manners. The first one is the fewer rebounds the signal gives after a fault is detected compared to the oscillatory response from a centralized system. On second stance, the system is less vulnerable to single failures. When the controller of a centralized network fails, the network stays unregulated. However, in a distributed network, when a node fails, the neighboring nodes will account for it regulating the network.
- Disadvantages: The increased complexity over centralized controls makes this approach more time consuming and computationally difficult.

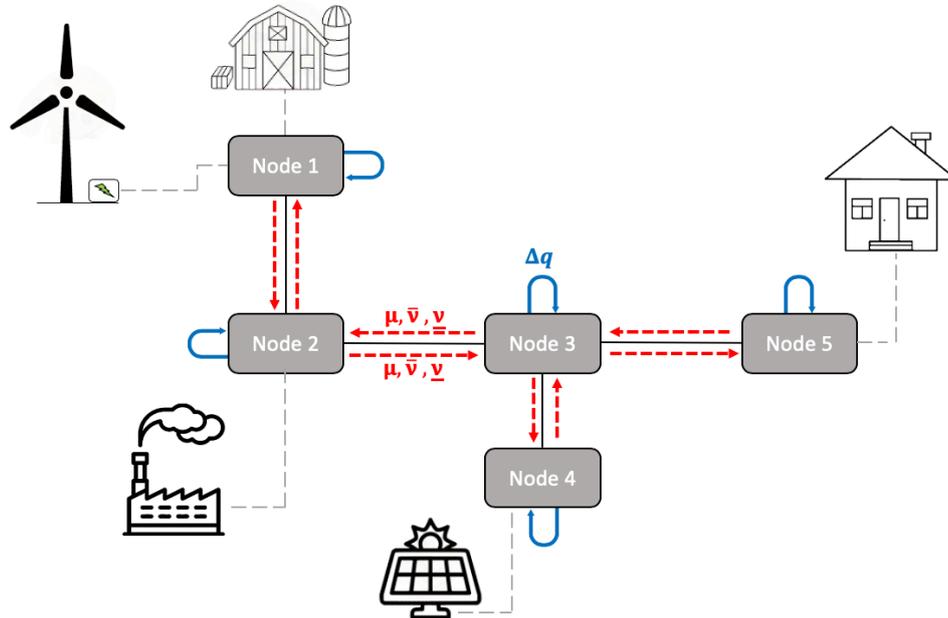


Figure 8: Distributed Network Diagram



## **Chapter 2. DESCRIPTION OF THE TECHNOLOGY**

### ***2.1 THE TEAM***

The team that is working in the ongoing project is composed of a teacher, three post-doc students, two graduate students and three under-grad students.

The director of the project is professor Alejandro Dominguez-García. He is a teacher in the university of Illinois at Urbana-Champaign. He has based his research in the power and Energy system area, especially in microgrids, grid data analytics and Reliability analysis.

Working along professor Dominguez are the post-docs Olaolu Ajala, Madi and Siddartha. They are in charge of creating and implementing algorithms towards the creation of a distributed microgrid.

In this project two other under-graduate students carrying out our senior thesis in this project. They are Patricia Samper, Alejandro Mayoral. Patricia Samper is developing the Feasible Flow algorithm for the network in which she is addressing the frequency control.

Alejandro is developing the model for the Typhoon HIL simulator. Particularly the individual high and low order models for each of the elements.

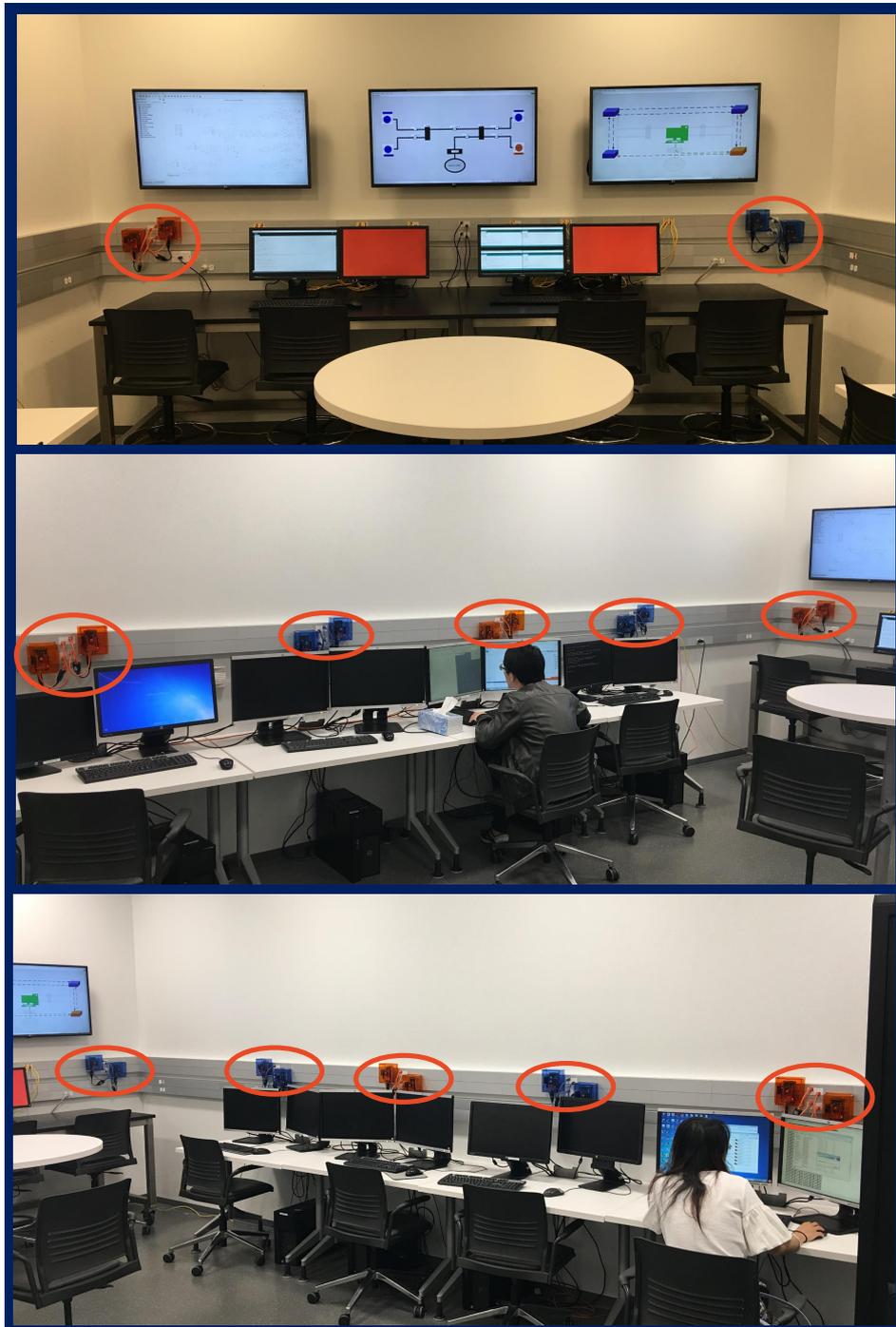
The major portion of my research project is based on the implementation of the voltage control algorithm into the microgrid simulator. This voltage control will determine the behavior of the network when working on under or over voltage conditions. This situation can be the output from many circumstances. The most common nowadays is the failure in generation, consumption or transportation of power. But the most important is the application of this control so as to regulate the implementation of non-constant generation.

This is referred to renewable generation in the more sustainable modern world in which we live.

## ***2.2 THE LABORATORY***

The project is taking place in one of the laboratories of the University of Illinois at Urbana-Champaign. The material in this room can be divided into three categories.

1. Network simulation: The first step towards creating a virtual network is the definition of the topology that the system will have. This process is developed by one of the team members, A. Mayoral. The implementation involves the creation of a model of the network in a program named Typhoon that will be uploaded into the network first simulator, the Typhoon HIL.
2. Algorithm creation and testing: The second step into simulating the network is writing and testing the code that will implement the algorithms that the controllers will have to apply. This is done in two high power computers which are used by the team.
3. Code storage and communication: After the code is written and thoroughly tested for errors, it is stored into the memory of the Arduinos. These Arduinos will act as nodes. Each Arduino can simulate multiple nodes. These Arduinos will communicate one with each other according to the structure and behavior set by the grid simulator. The Arduino layout from the system is shown in Figure 9.

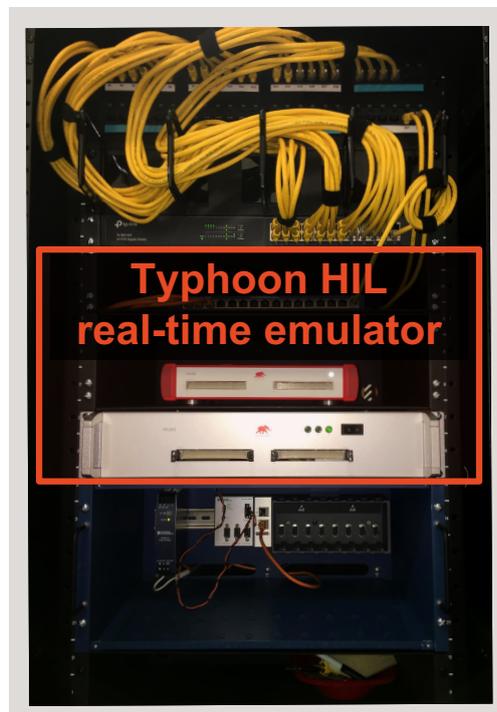


*Figure 9: Arduino Layout in the Laboratory*

## 2.2.1 TYPHOON HIL

### 2.2.1.1 Hardware

The Brain of the project is the C-HIL testbed. It will simulate the physical behavior of the network. It consists of three ultra-high-fidelity real-time simulation devices. One Typhoon HIL 402 and two Typhoon HIL 603s [9]. These will allow us to simulate full-scale networks in order to analyze the behavior. It will simulate responses at step sizes as low as  $0.5\mu\text{s}$  PWM sampling of 20 ns [9]. This allows for very high levels of fidelity. The computational power available through these network simulators gives the team the ability to emulate a large number of nodes. The Typhoon HIL device, which is shown in Figure 10 is responsible for the high-end simulation and behavior of the network where the system evolves over time. They also store pre-defined information about each node. This information will later be transmitted to the system-level controllers.



*Figure 10: Typhoon HIL Device*

### 2.2.1.2 Typhoon HIL Software

In order to simulate microgrids we have to use the Typhoon HIL software. It is going to be the most important part in the simulation. The main purposes it is going to serve are:

- Create models: The Schematic Editor is the tool used to design the microgrid models. It is going to be responsible for all the architecture, from the measurements to the behavior definition for droop controllers and loads. The interface is similar to that of Simulink but falls behind it in terms of simulation tools and mathematical resources. It is good enough for linear operations and some non-linearities but runs into problems when computing some non-linearities such as constant power loads. The main advantage this system has is the versatility since it has a rich library of controllers, from wind power plants to storage solutions. Figure 11 shows the Schematic Editor for the 6-node network that is going to be implemented in this project.

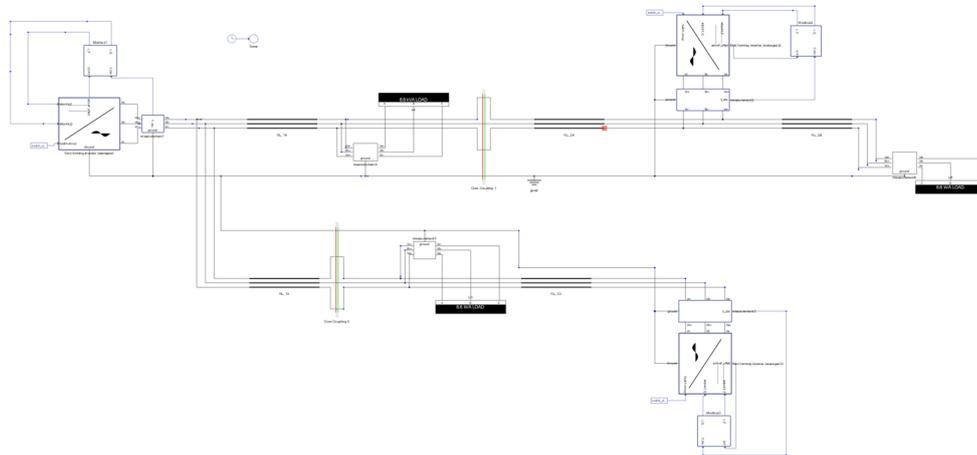


Figure 11: Schematic Editor for 6-Node

- Test models: The HIL SCADA is the tool used to run the simulations and monitor the results. Once the model is created, it is compiled and if it does not have any major failure, the HIL SCADA will open the model in order to start the simulation. By

starting the simulation together with the communication that has to be set with the Arduinos that will perform the control algorithms we will obtain the proper results. This tool will allow you to study how the model evolves over time, giving you the ability to add programable widgets and capturing devices. This will allow for you to check from voltage level to stress in the lines. Figure 12 shows the HIL SCADA for the 6-node example that is going to be studied throughout this project.

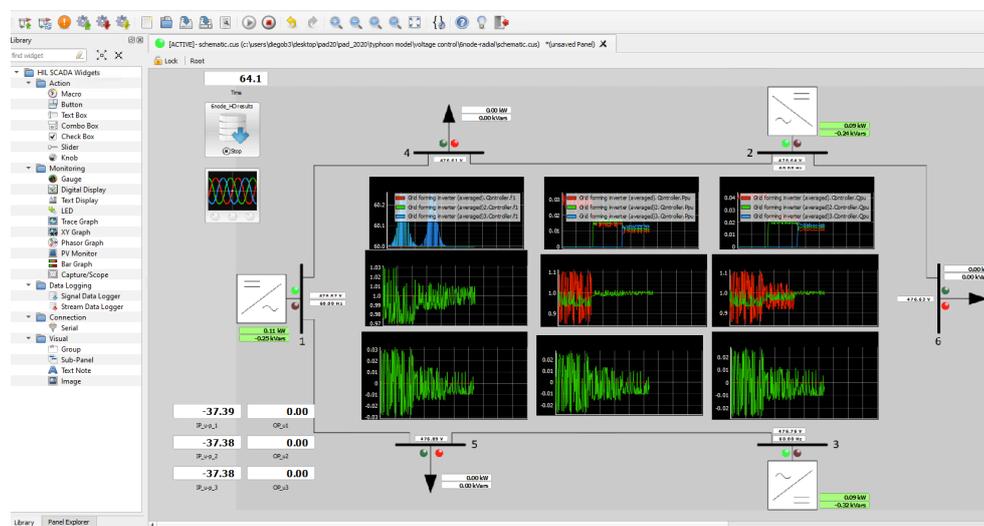
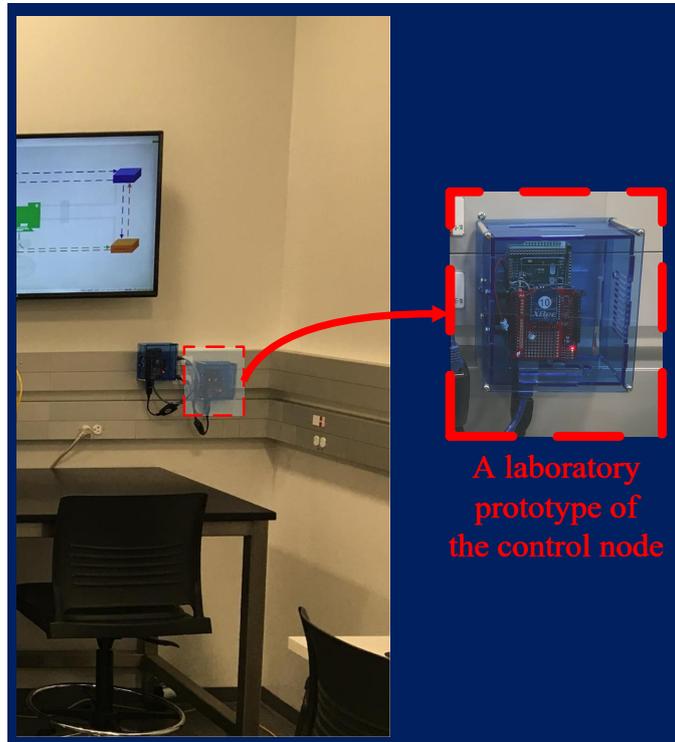


Figure 12: HIL SCADA for 6-Node

## 2.2.2 LOWER-LEVEL CONTROLLER HARDWARE

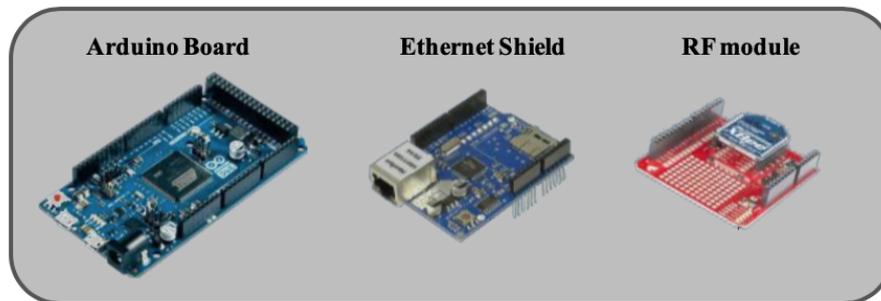
On the hierarch of the network. The first piece is the C-HIL bench.. The second layer is composed of the Lower-level Controllers. The role of this controller is being performed by Arduinos. The models. that will be used in this project are shown in Figure 13.



*Figure 13: System-Level Controller in the Laboratory*

This layer of control is in charge of performing system level DER control schemes. These are the ones that focus on issues like voltage and frequency violation; as well as oscillator control. The structure of the nodes, which is depicted in Figure 14, is divided as follows:

- Topology: The Arduinos modules, described in, are composed of three layers. The first one is the board that will be the main component. The board is in charge of storing the code and implementing it with the inputs received from the C\_HIL devices. The second layer is the Ethernet shield. The third layer is the XBee (RF) module.



*Figure 14: Arduino Components[9]*

- **Behavior:** The first and most important task is the communication in between one another. It is carried out by the XBee module and the ethernet. The XBee module is in charge of the wireless communication in between the Arduinos themselves. The Arduinos need to have bidirectional communication with one another in order to implement a distributed architecture. The Ethernet Shield is responsible for wired communications with the computers. Its purpose is to upload the code from the computers into the Arduinos and the display of information back in the computers after the computations are completed in the board.

### 2.2.3 SOFTWARE

The Arduinos are programmed in a language that resembles the architecture of C++. The aim of this project is to implement a series of algorithms into the libraries that the Arduino will further use. These algorithms will be the ones used to manage all the tasks that the controller will need to develop. From voltage control to node loss, all of the protocols and operations are stored in the Arduino's memory. The Arduinos are going to use Three main Libraries in which we will dive deeper further in the paper.

- **OAgent:** This is the library that is going to be of greater importance for this paper. The focus on this is to implement node synchronization and the definition of the

control algorithms. Development of functions that the Arduino will have to run as a response to a problem or a request (voltage control, node sync...).

- OGraph: This library is responsible for defining the nodes and the control architecture for the computation, setting the links between nodes and their properties.
- XBee: This library is responsible for communication between the Arduinos.



## **Chapter 3. CURRENT STANDPOINT**

### ***3.1 LITERATURE REVIEW***

The most important source of information for this project has been the paper by Professor Dominguez [1]. Thanks to it I have learned about distributed controls in microgrids and overall basic knowledge in the modern paradigm of electric grids and how to manage them. Apart from that, this paper is the one in which this thesis is based on. This is because the algorithms that are being implemented are extracted from this paper.

### ***3.2 PREVIOUS RESEARCH ON THE MATTER***

Some research has been done on the area, especially on the matter of the impact of distributed generation like the one in [11]. In this paper the increase of DGs is studied, and with it, the increase in voltage vulnerabilities that they propose. Once identified the faults, the control architecture is studied and even though the centralized approach is the one that is more common nowadays, they propose the alternative implementation of distributed generation that improves reliability while allowing for the connection of larger generators to the power network. Throughout this paper, they perform optimal flow computations in order to compare the two approaches.

Other approaches on the matter are brought by research projects like the one proposed in [12]. This study shows the benefits of frequency control in demand response in order to minimize peak power impact or load shifting and how the implementation of control architecture can help do so. In order to perform the control architecture, they need to compare the two approaches, which will be compared according to their ability to maximize the impact of the control algorithms in the demand response.



## **Chapter 4. DEPICTION OF THE PROJECT**

### ***4.1 JUSTIFICATION***

This project is a result of the voltage control algorithms described in [1]. In this paper the authors introduce a set of algorithms for the voltage control in distribute microgrid. This comes from the desire to change the control architecture of microgrids from centralized to distributed.

Our team is from the University of Illinois at Urbana Champaign (UIUC). And we are developing a distributed microgrid control architecture. This scheme is composed of a number of control algorithms that aim towards the simulation and control of a microgrid, in this case a simulated one. These algorithms not only include control paths, such as voltage control, frequency control or optimal power flow. But also include every step in the regulation of grids, from wireless communication in between nodes to synchronization and initialization of procedures.

This project is based on the implementation of the algorithm for voltage control into C++. This is due to the need to create a library with all the control algorithms a network will need to implement in order to ensure the proper performance of the microgrid. This code will later be used by Arduinos in order to control a simulated grid by the virtual HIL simulator.

Throughout this paper we are going to justify the implantation of microgrid into the modern paradigm of power systems. This is going to be supported by an economic analysis that will determine the viability of implementing microgrids into the existing power network we have nowadays. Once decided on the implementation, we are going to explain the two possible control architecture that can be implemented into microgrids, centralized and distributed.

Both approaches are going to be compared from a reliability and from a response quality point of view.

## **4.2 OBJECTIVES**

There are three objectives for this project, organized from top to bottom according to priority and importance.

1. Construction of the Arduino code: this step is the coding of the algorithms in C++ following object-oriented programming techniques. The code is defined in {ANNEX II} and consists of several layers of code.
2. Reinforce the ratio consensus Algorithm: The ratio consensus code has been written several times during the life of the project, each of the iterations has had the objective of increasing its efficiency and minimizing the processing time. It is described in {8.1}.
3. Improve iteration precision: In the ratio consensus algorithm, the precision of the same is affected by the number of iterations that are run. The adequate number of iterations can be set experimentally, but a way to automatize this project in order to stop the number of iterations at the optimal point, the maximum consensus, which is described in {8.2}.

## **4.3 METHODOLOGY**

The methodology of the research project is divided into steps:

1. Review previous work about centralized and distributed voltage control algorithms.

2. Understand the concept of microgrid, the voltage control algorithms and the instruments.
3. Create flowcharts and class diagrams for the centralized and distributed codes.
4. Code the algorithms into C++ and test them in typhoon.
5. Study the results and come up with the conclusion of the project.



## Chapter 5. MODEL DEVELOPED

### 5.1 VOLTAGE CONTROL IMPLEMENTATION

For a grid to work in optimal conditions there needs to be a limit to the deviation in voltage. Voltage is set to a Voltage Reference in the DERs and the limits will be set at  $\pm 5\%$ . This reference value is considered to be 1pu since the  $V_{ref}$  that will be chosen will be the base value. Voltage will tend to match the reference value, but this value can be altered. This deviation in voltage comes from variations in reactive and active power input and consumption. The most common scenario in which we encounter this form of failure is the failure of a node. This can be given as result of a generator plant falling or the disconnection of load. It can lead to a variation in the power scheme in the grid and will result in a variation in the voltage profile, therefore incurring in a violation of the limits. Another scenario in which we can come across a variation in voltage is the implementation of DERs in microgrids. These resources are characterized for their variable input/output power curves as can be seen in Figure 4 and Figure 5.

The variation of active and reactive power as a function of time  $r$  in PQ buses is defined in [1] as  $\Delta S[r+1] = S[r+1] - S[r]$ . From this expression we can define the expression for the computation of power flow in:

$$\begin{bmatrix} \Delta P[r] \\ \Delta Q[r] \end{bmatrix} = \begin{bmatrix} H & N \\ K & L \end{bmatrix} * \begin{bmatrix} \Delta \theta[r] \\ \Delta V[r] \end{bmatrix}$$

From where we obtain:

$$E. 1. \quad H = \left[ \frac{\partial P_i}{\partial \theta_j} \right], N = \left[ \frac{\partial P_i}{\partial V_j} \right], K = \left[ \frac{\partial Q_i}{\partial \theta_j} \right], L = \left[ \frac{\partial Q_i}{\partial V_j} \right]$$

These equations describe the variation of active and reactive power in the system due to change in  $V$  and  $\theta$ . We will be operating under the Decoupled assumption, in which it is

stated that the variations in voltage due to a change in active power injection is neglectable, we will be studying the system on the changes in reactive power as a means to change the voltage profile.

For the computation of the first stage control, we need to obtain the system's behavior, which is described by the sensitivity matrix.

We will start from the power flow equations, where  $i$  &  $k$  are the nodes at both ends of the line,  $\mathcal{N}_i$  is the set of neighbors to node  $i$  and  $b$  and  $g$  come from the  $Y_{bus}$  matrix as  $Y_{ik}=G_{ik}+jB_{ik}$ .

$$P_i = \sum_{k \in \mathcal{N}_i} V_i V_k [G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k)]$$

$$Q_i = \sum_{k \in \mathcal{N}_i} V_i V_k [G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k)]$$

And from the infinitesimal assumption that all of the angles will be near to zero, we can approximate  $\cos(\Delta\theta) \approx 1$  and  $\sin(\Delta\theta) \approx \Delta\theta$ , obtaining the simplified power flow equations:

$$E. 2 \quad P_i = \sum_{k \in \mathcal{N}_i} V_i V_k [G_{ik} + B_{ik}(\theta_i - \theta_k)]$$

$$E. 3 \quad Q_i = \sum_{k \in \mathcal{N}_i} V_i V_k [G_{ik}(\theta_i - \theta_k) - B_{ik}]$$

From the use of E. 1, differentiating the equations E. 2 and E. 3, we obtain the equation that will give us the system parameters H, N, K and L.

For  $i \neq k$

$$E. 4 \quad H = \frac{\partial P_i}{\partial \theta_k} = -V_i V_k B_{ik}$$

$$N = \frac{\partial P_i}{\partial V_k} = V_i [G_{ik} + B_{ik}(\theta_i - \theta_k)]$$

$$K = \frac{\partial Q_i}{\partial \theta_k} = -V_i V_k G_{ik}$$

$$L = \frac{\partial Q_i}{\partial V_k} = V_i [G_{ik}(\theta_i - \theta_k) - B_{ik}]$$

For  $i=k$ :

$$E. 5 \quad H = \frac{\partial P_i}{\partial \theta_k} = \sum_{i \neq k} V_i V_k B_{ik} \quad N = \frac{\partial P_i}{\partial V_k} = \sum_{k \in N_i} V_i [G_{ik} + B_{ik}(\theta_i - \theta_k)]$$

$$K = \frac{\partial Q_i}{\partial \theta_k} = \sum_{i \neq k} V_i V_k G_{ik} \quad L = \frac{\partial Q_i}{\partial V_k} = \sum_{k \in N_i} V_i [G_{ik}(\theta_i - \theta_k) - B_{ik}]$$

For the computation of the sensitivity matrix we will need to compute the matrices H, K, N, L. In order to obtain them we will use E. 5 for the elements in the diagonal, and E. 4 for the out-diagonal elements.

Once we have H, K, N, L, the computation of the Sensitivity matrix will be obtained from the expression:

$$E. 6 \quad S = (L - KH^{-1}N)^{-1}$$

The voltage levels will be monitored by the controller. The reference values for the control will be established in accordance with the base voltage and a percentage of security chosen beforehand to ensure the proper functioning of the system. The voltage limits will be established as  $V = V_{ref} \pm V_{ref} * (\%/100)$ , being % the precision limit for the voltage control, as can be explained in Figure 15.

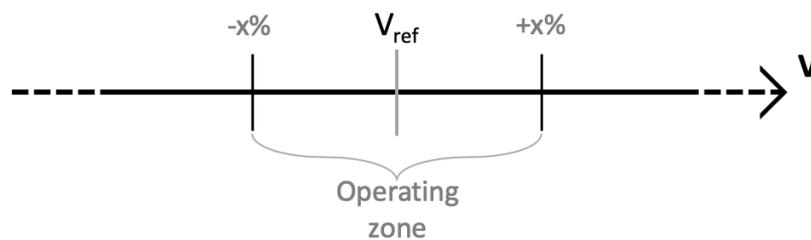


Figure 15: Voltage limits

The controller is going to be recurrently evaluating the voltage level in order to prevent voltage violations. If the controller detects a violation, it will start the implementation of the voltage control algorithm. The voltage control algorithm consists of two stages:

- 1st Stage: The first stage is the self-adjustment step where the local control tries to mitigate the fault its own DER. The controller will compute the algorithms that will obtain as a result the target value for the reactive power in the DER. The controller will evaluate the quantity obtained through its calculation with the level of available resources from the node.
  - If the node can supply the change in reactive power needed for the implementation of the first stage, it will do so, mitigating the failure and restoring the normal operation conditions. This will bring the voltage profile back to the operating zone shown in Figure 15.
  - If the node cannot supply the change in reactive power needed, the system will initiate the second stage control. First the faulty node will set its reactive power to the limit ( $\bar{q}$  or  $\underline{q}$ ) depending on the nature of the failure (overvoltage or undervoltage). Second, the computation of the second stage algorithms will commence.
- 2nd Stage: The second stage is the collaborative step in the voltage control algorithm. It is described as a request for help from the faulty node to its neighbors. When the faulty node cannot provide the required amount of reactive power that the network needs in order to mitigate the problem, the system starts the collaborative stage. In it all the neighboring nodes will help injecting/consuming reactive power in order to restore the voltage profile. Once the controller asks for help from the neighboring nodes, the second stage takes off.

### 5.1.1 FLOWCHART

Figure 16 shows the flowchart that describes the steps that take place in the voltage control algorithm is:

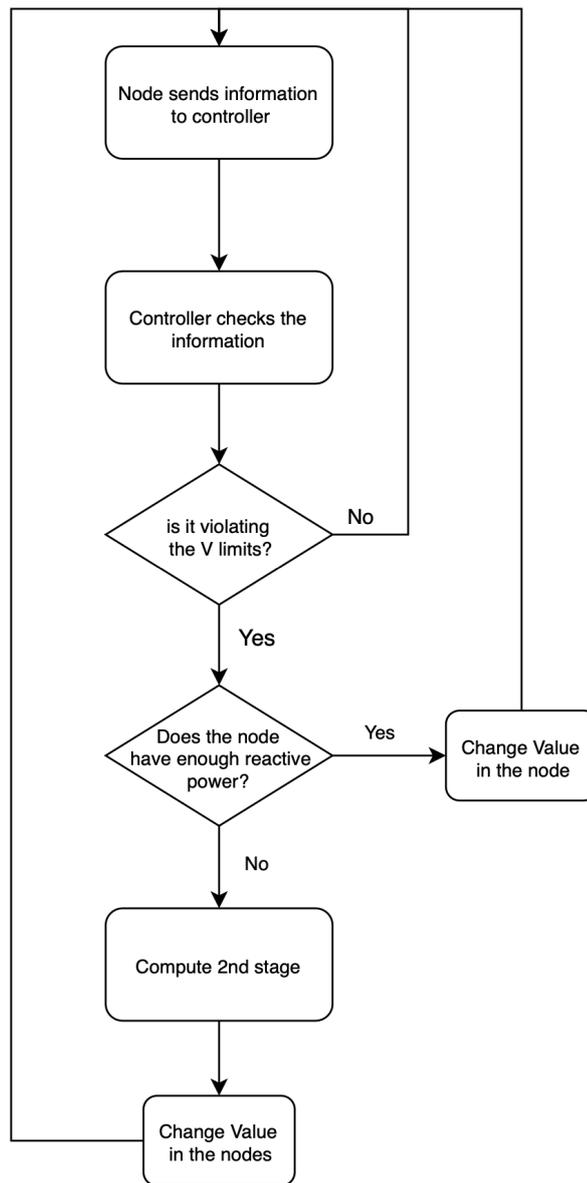


Figure 16: Voltage Control Flowchart

## 5.2 DISTRIBUTED APPROACH

### 5.2.1 STATE DIAGRAM

Figure 17 shows the state diagram for the distributed approach. We can observe how the system goes from normal operating mode to the first stage once it spots a voltage violation. Once the first stage is computed, the system can either go back to normal operating mode if the node has enough reactive power to mitigate the fault, or it can initiate the iterative algorithm of the second stage where all the nodes will work together in order to reach the normal operating mode.

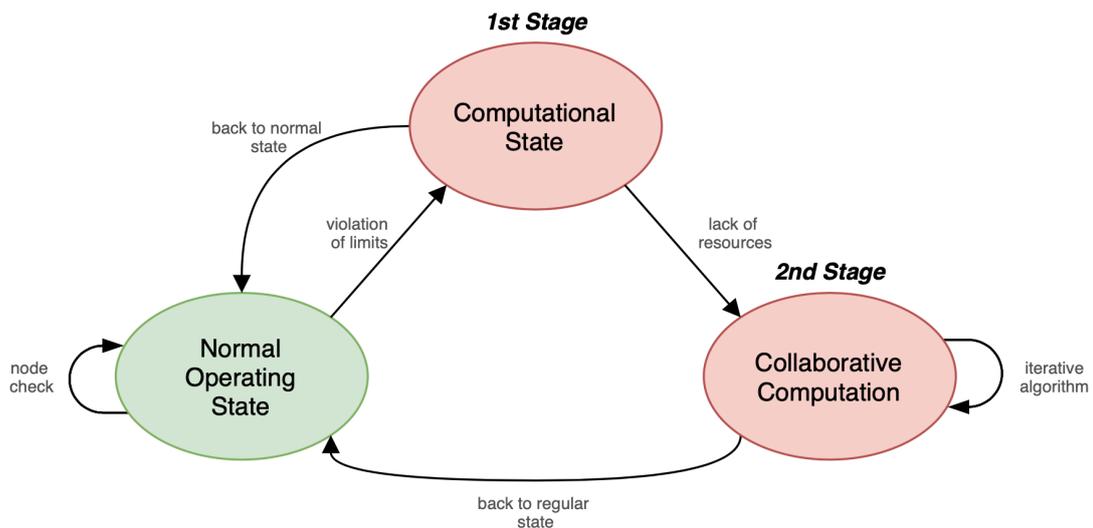


Figure 17: Distributed Actuation diagram

### 5.2.2 ALGORITHM

#### 5.2.2.1 First stage

Once the controller detects a voltage violation, the system will start computing the first stage controlling algorithm. In the first stage the algorithms will try do an auto-reparation strategy.

The required reactive power needed by the node will be supplied by itself in order to minimize distributed approaches and increase efficiency and increase the speed of response.

First, the reactive power needed is calculated. It is a function of the difference between the voltage level and the limit in the DER ; the participation factor ( $\alpha$ ) and the node's sensitivity ( $s_{jj}$ ) computed with E. 6 and shown in Figure 23.

- Over-Voltage:

$$E. 7 \quad \Delta q_k[r] = \frac{\alpha}{s_{jj}} (\underline{V}_j - V_j[r])$$

- Under-Voltage:

$$E. 8 \quad \Delta q_k[r] = \frac{\alpha}{s_{jj}} (\bar{V}_j - V_j[r])$$

Once the required reactive power is computed, the amount of remaining resources has to be taken into account. If there is enough the system will change the node's reactive power target level and the control algorithm will have finished. If one of the nodes doesn't have enough reactive power, the faulty node will be categorized as a saturated node and all the nodes in the system will enter the Second Stage part of the algorithm.

### ***5.2.2.2 Second Stage***

This is the collaborative part of the algorithm. In contrast to the centralized voltage control algorithm which is characterized by having all the information from all the nodes available, this algorithm is built following a distributed architecture which makes this step more complex.

Before the start of the algorithm, there needs to be an initialization. We have to initialize  $\nu$  and  $\mu$ , which are going to be the auxiliary variables for the implementation of the ratio consensus in the nodes so as to obtain the required amount to rise or decrease in reactive power in order to mitigate the fault. For the explanation of the ratio consensus we need to define the concept of the reactive power target as [1]  $\hat{q}_j[r + 1] = q_j[r] + \Delta q_j[r]$ .

The first stage is the initialization of the auxiliary variables. This initialization is characterized by the state of the node.

- Over-saturated:

$$\begin{cases} \mu[0] = \hat{q}_j[r + 1] - \bar{q}_j \\ \bar{v}[0] = 0 \\ \underline{v}[0] = \underline{q}_j - \hat{q}_j[r + 1] \end{cases}$$

- Under-saturated:

$$\begin{cases} \mu[0] = \hat{q}_j[r + 1] - \underline{q}_j \\ \bar{v}[0] = \bar{q}_j - \hat{q}_j[r + 1] \\ \underline{v}[0] = 0 \end{cases}$$

- Between q limits:

$$\begin{cases} \mu[0] = 0 \\ \bar{v}[0] = \bar{q}_j - \hat{q}_j[r + 1] \\ \underline{v}[0] = \underline{q}_j - \hat{q}_j[r + 1] \end{cases}$$

Once the variables are initialized, the system will compute ratio consensus in order to find the quantity that the DER have to change their q level in order to help the faulty node. The ratio consensus algorithm is explained in {8.1}. We will compute two ratio consensus algorithms. One with the upper nu and one with the lower nu. Once we reach consensus, depending on the value of  $\mu$  one of the results will be chosen. The reactive power variation due to the second stage algorithm is defined by  $\eta$ . The ratio consensus function will be denoted as  $RC(y,z)$ . The formulation for the ratio consensus is:

$$\bar{\eta} = RC(\mu, \bar{v})$$

$$\underline{\eta} = RC(\mu, \underline{v})$$

The criteria for the election of  $\eta$  is based on the sign of  $\mu$  with  $k$  being the final iteration number:

$$\eta = \begin{cases} \bar{\eta} & \mu_j[k] > 0 \\ \underline{\eta} & \mu_j[k] < 0 \end{cases}$$

On the last step of the voltage control algorithm, the limits need to be evaluated. The target values for the nodes will be decided as follows[1]:

$$q_j[r + 1] \begin{cases} \bar{q}_j, & \hat{q}_j[r + 1] + \eta_j > \bar{q}_j \\ q_j, & \hat{q}_j[r + 1] + \eta_j < \underline{q}_j \\ \hat{q}_j[r + 1] + \eta_j, & \text{otherwise} \end{cases}$$

### 5.2.3 DESCRIPTION OF THE CODE

The code has been written in C++ in the Sublime Text editor and the content is divided in two libraries, OGraph and OAgent. OGraph is in charge of defining the image of the network, assigning neighbors and node parameters. The objects that are defined in OGraph are the OLocalVertex, which stores the local parameters from the node, and the ORemoteVertex, which stores the parameters obtained from the broadcasted information packages that are transmitted from one Arduino to the order in order to implement ratio consensus. The class diagrams that define ORemoteVertex and OLocalVertex are in {ANNEX II}.

The code is divided in several function inside an Arduino library, which are shown in {ANNEX II} and are divided as follows:

- Voltagecontrol: The core of the algorithm. From it we will be calling the rest of the functions.
- Initializevoltagecontrol: The initialization of the parameters in the OLocalVertex for the voltage control algorithm.

- IsOver/Undervoltage: These two functions will check if the node is experiencing voltage violation over or under the limit.
- Firststagecontrol: The implementation of the First Stage of the algorithm. The operation that are computed are specified in 5.2.2{5.2.2.1}.
- Shareflag: This function will broadcast whether the nodes need to implement the Second Stage voltage control or not.
- Initializevariablessecstage: The initialization of the values for the auxiliary variables for the second stage ratio consensus computation.
- Secondstagecontrol: The implementation of the Second Stage algorithm. The collaborative algorithm that is computed in {5.2.2.2}.

Finally, the code returns the new reactive power setpoint for the node. This value will later be passed to the Typhoon HIL device.

## 5.2.4 FLOWCHART

Figure 17 shows the flowchart that describes the steps that the code takes in order to implement the Distributed voltage control algorithm.

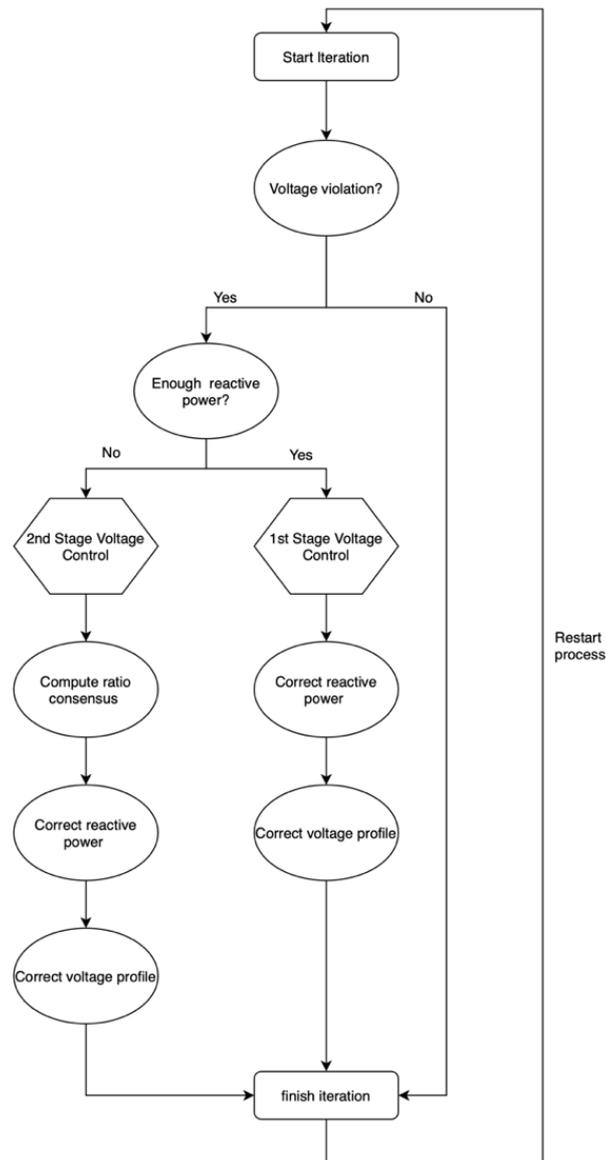


Figure 18: Distributed Voltage Control Code Flowchart

## 5.3 CENTRALIZED APPROACH

### 5.3.1 STATE DIAGRAM

Figure 19 shows the state diagram from the centralized algorithm for voltage control. Once the system is operational, the system will infinitely be checking the variables obtained. Once it detects a voltage violation. It will use the available information, such as reactive power levels and limits to fulfill the voltage deviation. There is no need to compute an iterative algorithm like in the distributed approach.

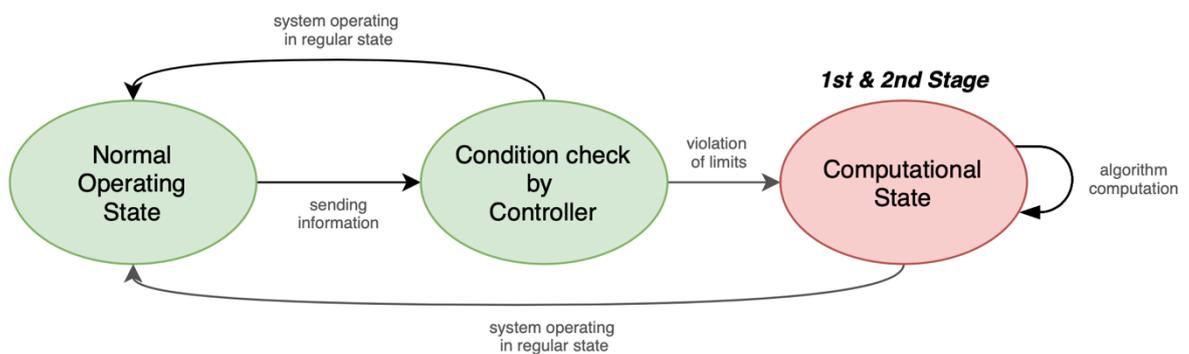


Figure 19: Centralized Actuation diagram

### 5.3.2 ALGORITHM

The information from the nodes is going to be introduced in the Arduino, which will compare it with the limits until it detects a voltage violation. Once the violation is found, the Arduino will implement the voltage control algorithm in the Centralized approach.

Since there is only one Arduino in charge of the computation, the iteration times will be shorter since the communication in between controllers is eliminated. The control will start by the implementation of the first stage.

### 5.3.2.1 First Stage

The implementation of the first stage of the algorithm is the one where the failure will try to be fixed within the faulty node. The algorithm that is going to be implemented for the first stage is the same as for the distributed approach. The equations that are going to be implemented are the same as with the distributed network. We will use E. 7 for over-voltage cases and E. 8 for under-voltage. If the node is not undergoing any fault, the reactive power variation in this node will be zero ( $\Delta q_j=0$ ). After obtaining the  $q_{\text{target}}$  for each of the nodes, the algorithm will compare the value with the reactive power reserve. In case the reactive power needed exceeds the limits of the DER, the remaining power will be added to the reactive power that needs to be supplied by the second stage.

$$\hat{q}_j[r + 1] = q_j[r] + \rho_j[r]$$

$$q_{\text{sec}} = \sum_{\hat{q} > \bar{q}} (\hat{q} - \bar{q}) + \sum_{\hat{q} < \underline{q}} (\hat{q} - \underline{q})$$

### 5.3.2.2 Second Stage

This is the collaborative part of the algorithm where the neighboring nodes will vary their reactive power level in order to fix the failure that the first stage could not fix.

The lack of need of communication makes the code simpler. There is no need to implement ratio consensus. The required reactive power variation in the second stage is computed proportionally to the reserve the node has:

$$\Delta q_{\text{sec}}^j = q_{\text{sec}} * \frac{q_{\text{reserve}}^j}{q_{\text{reserve}}}$$

Once the variation of reactive power in the second stage is computed, there needs to be another check of reactive power limit violation for the final  $q_{\text{target}}$ . Once the second stage is

over, the reactive power level is communicated to the typhoon HIL device which will change the setpoints in order to get the system back to its normal operating state.

### **5.3.3 DESCRIPTION OF THE CODE**

The code is implemented directly Into the Arduino code due to the simplicity of the algorithm. The code is shown in {ANNEX III}.

The code is divided in:

- First Stage: The node tries to fix the failure by itself.
- Reserve Q check: The Reserve of Q is computed.
- Second Stage: the collaborative algorithm is computed with all the information that has already been recovered by the Arduino.

### 5.3.4 FLOWCHART

Figure 20 shows the flowchart that describes the steps that the Arduino code takes in order to implement the Centralized voltage control algorithm.

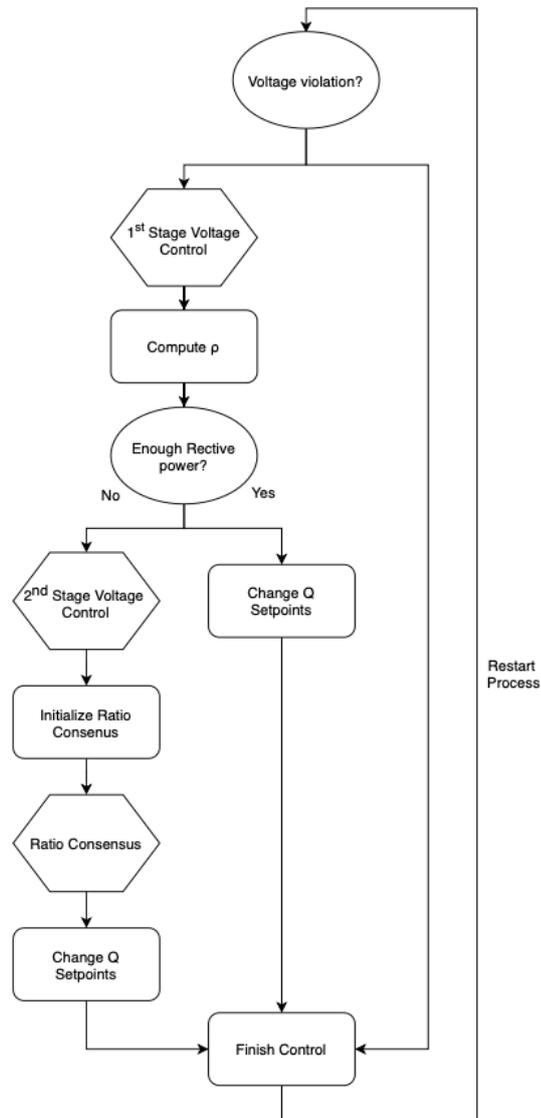


Figure 20: Centralized Voltage Control Code Flowchart



## **Chapter 6. RESULTS ANALYSIS FOR 6-NODE**

Throughout this project we focus on developing control algorithms that can help to control a microgrid. The aim is to develop a distributed control architecture that can regulate any microgrid in which it is implemented. In order to achieve that, the algorithms that are designed need to be tested. In this chapter we will describe the test conditions for the algorithms and the results that we have obtained, particularly from the voltage control algorithm.

First, we will describe the layout of the microgrid that is going to be used as testbench, describing the position of the components and the characteristics of the network, which includes the number of generators as well as the power limits and sensitivity of the system. Secondly, we will lay out and comment the results obtained for both the distributed and centralized architectures. Third, we will compare the results obtained for both architectures whilst drawing the conclusions of this analysis. Finally, we will present the conclusions of the test scenario.

### **6.1 LAYOUT**

The example that is going to be studied is a six-node radial configuration. Figure 21 shows the layout of the microgrid. The system consists of three generators (1,2,3) and three load buses (4,5,6). This model will be implemented into the virtual HIL in order to compute the simulation. Each of the nodes in the system is modeled by an Arduino board. Such board has access to the libraries that are uploaded from the Computer.

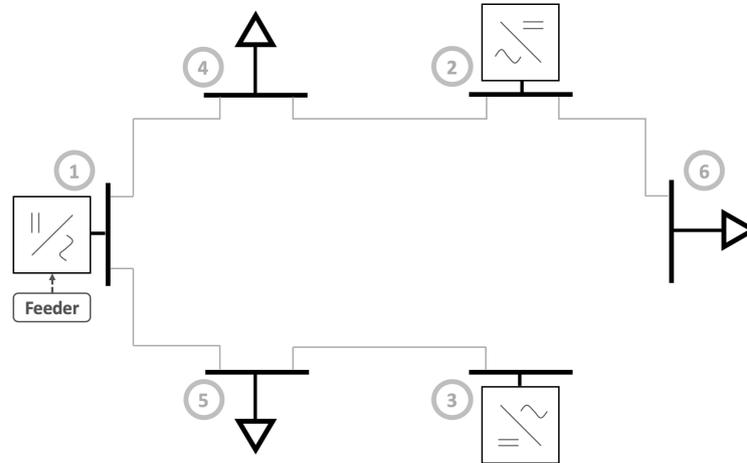


Figure 21: 6 Node Radial Model

The system is characterized by its line parameters, which are shown in Table 1 , which describes the resistance and impedance in between lines as well as the rated power at the which they operate. The nodes are characterized by their rated voltage which is the same for all of them and will be the voltage reference ( $V_{ref}$ ) and the rated power at the which they operate. These values are shown in Table 2.

Table 1: Line Parameters

Bus		Parameters		
from	to	R(pu)	X(pu)	S (KVA)
1	4	2.34592	3.04623	200
1	5	2.73691	2.35838	200
4	2	2.22862	3.04623	200
2	6	2.34592	2.35838	200
5	3	2.22862	4.42195	200

Table 2: Bus Parameters

Bus	$V_{rated}$ (V)	$S_{rated}$ (KVA)
1	277	200
2	277	200
3	277	200
4	277	6.6
5	277	6.6
6	277	6.6

From the values described in the previous tables we can define the power limits for the DERs. The values for the Reactive and Active power limits will be inferred from the P-Q diagram shown in Figure 22, where P is the active power, Q the reactive power and S the total power. From this diagram and using E. 9, we can infer the boundaries. There are many different criteria so as to arrive at a viable solution for the values. The criteria for the shape of the limits should be chosen accordingly to the role that the DER plays in the grid. Using these equation as well as the grid's experimental data we could make a decision in accordance with the reactive and active power that the DER is going to require. This technic maximizes the performance and minimizes the shortages in resources. Since for these simulations we do not have experimental data or operational data from the DERs, we will be choosing the least restricting criteria. The least restricting case is obtained following the square paradigm. In this scenario, the limits will be the same for active and reactive power.

$$E. 9 \quad S^2 = \sqrt{P^2 + Q^2}$$

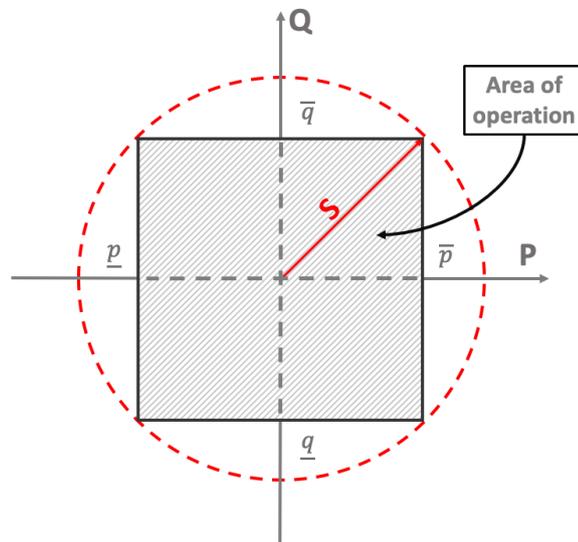


Figure 22: P-Q Diagram

From the parameters in Table 1 and Table 2; and the equation E. 6 we obtain the sensitivity matrix that is shown in Figure 23.

$$S_{ij} = \begin{bmatrix} -0.225736 & -0.005353 & 0.018581 & -2.96672 & 0.008358 & 3.51653 \\ -0.033996 & -0.224693 & 0.021049 & 0.001518 & 0.000087 & -2.96511 \\ 0.003489 & 0.003292 & -0.258852 & 1.84111 & -2.51039 & -2.17775 \\ -2.96534 & 0.00153 & 1.84085 & -0.225909 & 0.007286 & -0.032548 \\ 0.007543 & 0.000079 & -2.51089 & 0.043151 & -0.258583 & -0.051291 \\ 3.51514 & -2.96532 & -2.17847 & -0.003869 & -0.008873 & -0.227252 \end{bmatrix}$$

Figure 23: 6-Node Sensitivity Matrix

## 6.2 SIMULATION

In order to perform the simulation for the comparison between the centralized and the distributed voltage control we are going to perform the tests under the below characteristics:

1. **Black-start:** The microgrid will implement a black-start. This process consists in starting with all of the elements disconnected. Each of the elements will be connected in 30 seconds intervals until the grid is fully connected.
2. **Variable loads:** the simulation is going to put the voltage control to test once the system is operating in fully connected mode. The loads are going to be characterized by a change in value. Once the load is connected, it is going to change its value at every minute until the end of the simulation. The variation in active power is going to be of 10 kW. The load design that is going to be implemented in the HIL device is shown in Figure 24.

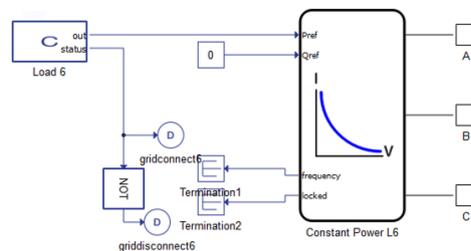


Figure 24: Typhoon Model of a Variable Load

The events for the simulation are described in the timeline shown in Table 3.

*Table 3: Timeline of the Simulation*

RELEVANT EVENTS	
Ocurrance	Time(seconds)
Gen-1 Conection	0
Gen-2 Conection	26
Gen-3 Conection	46
Load-1 Conection	100
Load-2 Conection	160
Load-3 Conection	220
Load-1 Update (+10kW)	$100+x*180$
Load-2 Update (+10kW)	$160+x*180$
Load-3 Update (+10kW)	$220+x*180$

## **6.3 DISTRIBUTED AND CENTRALIZED CONTROL RESULTS**

### **6.3.1 DISTRIBUTED LAYOUT**

The main characteristic of this approach is the control architecture. This architecture is characterized by a distributed control algorithm where all of the controllers have to work collaboratively in order to regulate the network as we saw in {1.4}.

Throughout this analysis the system level control architecture consists of Arduinos performing the function of controllers. In this architecture, the HIL device would be individually connected to each of the Arduinos, just like a traditional microgrid would. The Arduinos need to be arranged in a particular distributed architecture and would have to exchange information with their neighboring Arduino or Arduinos in order to perform the distributed algorithm. The architecture that has been chosen for this example is a linear layout. The controller from Node-2 would have all the information from both neighbors but Node-1 and Node -2 would only have information from Node-2. This will affect the performance of the control algorithm but will also show the ability that distributed networks

have to regulate themselves even under the worst conditions. Figure 25 shows the communication architecture that is implemented in the distributed control algorithm.

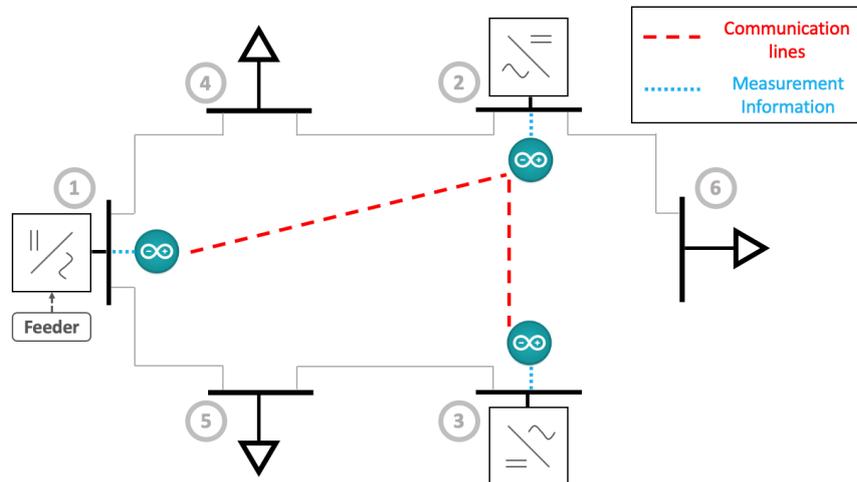


Figure 25: Distributed Model

### 6.3.2 CENTRALIZED LAYOUT

The layout of this model is based on the concept of the controller node. All the information from the measuring devices is going to be delivered to the controller node. The controller will compute the calculations and broadcast the new setpoints to the DERs. The main advantage of this method will be speed. This system will be faster than the distributed system at each iteration due to the lack of iterative algorithms. This will result in a faster response time to failures.

The system level control architecture consists of a single Arduino that is going to receive the voltage deviation from Typhoon device and the reactive power setpoints from all the DERs.

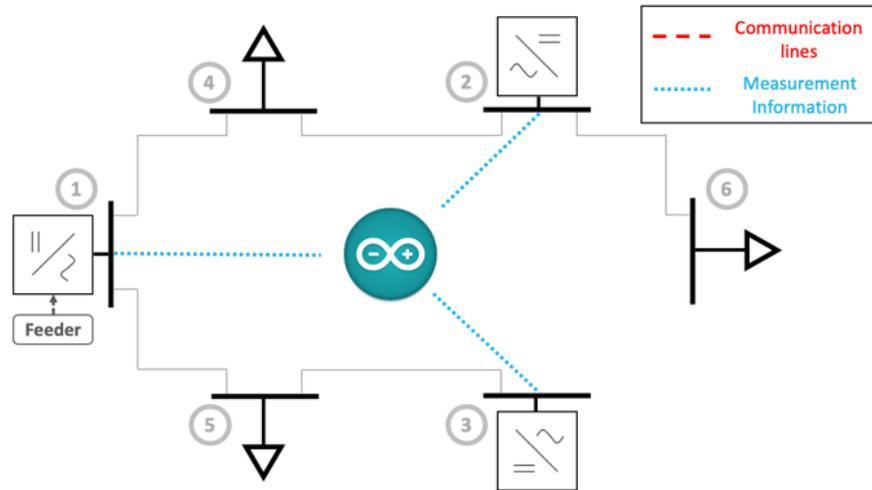
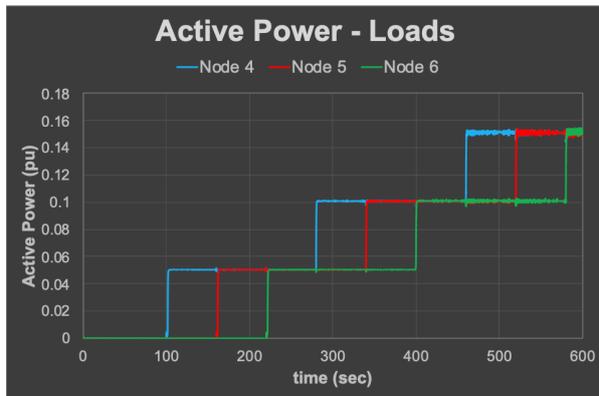


Figure 26: Centralized model

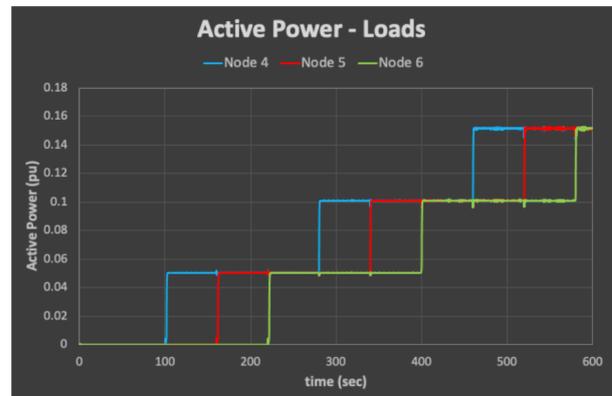
### 6.3.3 RESULTS ANALYSIS

In this section of the project we will present the response that the system will have to a specific operating state. We are going to analyze both approached, distributed and centralized architectures at the same time since the overall response is going to be similar. In {6.4} we will compare both responses in order to decide on the best approach for the specific project we are developing.

In order to draw conclusions from the voltage control algorithm we will study stress events in the simulation. We will analyze how the system reacts to the variable loads. There are two main events that need to be studied, which are defined in {6.2}. The first one is the load connection during the black start. The second one is the load size variation. Graph 5 and Graph 6 show the evolution of active power demand in the loads for the time period.



Graph 5: Active Power Variation in the loads  
(distributed)

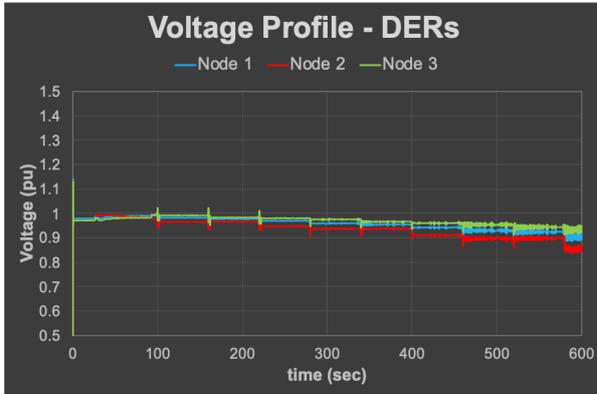


Graph 6: Active Power Variation in the loads  
(centralized)

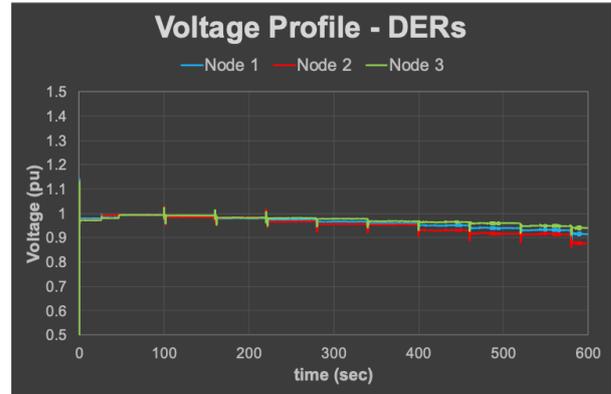
Throughout this simulation we will study the response of the DERs to the load variation shown in Graph 5 and Graph 6 above. In order to obtain relevant data we will compute the simulations in the HIL SCADA tool, which is defined in {2.2.1.2}.

The load connection events take place between  $t = 100$  and  $t = 220$  seconds and the load updates take place every minute. In order to obtain a meaningful reading, we will capture six hundred seconds, which will include the connection of the three loads and two updates for each of them.

In Graph 7 and Graph 8 we can observe that the voltage profile starts in 1 per unit, the reference value. We can see how over time the voltage goes through minor deviations caused by the connection of the generators and loads. These deviations are handled by the voltage control algorithm that we have implemented. While the voltage seems to tend to 1, we can see a slight curve in the voltage profile. This voltage variation comes as a result of the change in the system's consumption due to the variable loads. As the power consumption goes up, the voltage profile goes down.

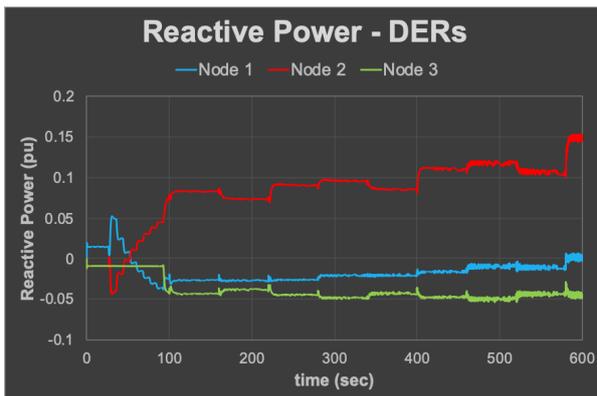


Graph 7: Voltage Profile in the DERs (distributed)

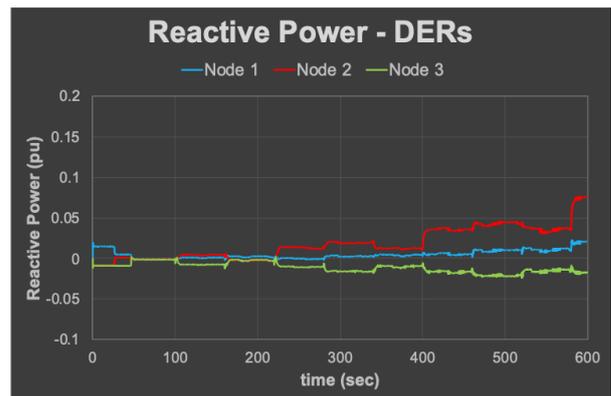


Graph 8: Voltage Profile in the DERs (centralized)

In order to understand the voltage profile, we need to study the DERs' response to the voltage violation that comes from the load increase. As we have seen in {5.2.2}, the response from the system will be the variation in the DERs' reactive power setpoints. This can be seen in Graph 9 and Graph 10. We can observe how with the reactive power in the DERs changes each of the load changes. This is going to result in an increased reactive power profile due to the increase in the overall consumption in the system.



Graph 9: Reactive Power in the DERs (distributed)



Graph 10: Reactive Power in the DERs (centralized)

From the results in the Graphs 7 to 9 we observe how the variation in the system's consumption is going to result in a voltage drop. This will be avoided through to the implementation of the voltage control algorithm, which will make the DERs increase their reactive power output to account for the voltage violation. This reactive power increase will be relieved with the implementation of secondary and tertiary voltage control, which will alleviate the reactive power in the DERs.

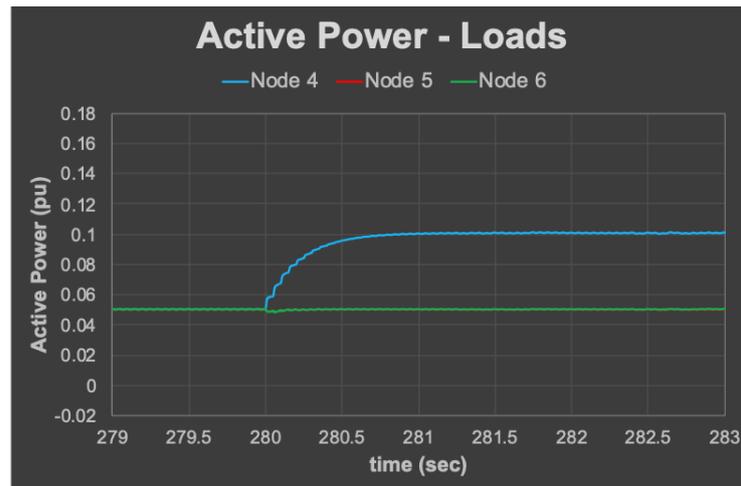
## **6.4 CENTRALIZED VS DISTRIBUTED RESULTS**

The main difference when comparing these two approaches is the communication scheme they use. The centralized architecture is based on broadcasting the results from all the nodes through the typhoon device to the Arduino, which will implement the algorithms by itself and, after the results are obtained, the new setpoints are broadcasted to the nodes. This can be seen in Figure 25. On the other side, the distributed algorithm is based on the communication in between Arduinos, especially through the ratio consensus algorithm {8.1}. The structure is based on three Arduinos that communicate with each other in a linear configuration. Each of the Arduinos obtains the information from one of the nodes, then computes the singular and iterative algorithms. Once the Arduino obtains the new setpoints for the nodes, they broadcast it and proceed to the next iteration. This can be seen in Figure 26.

### **6.4.1.1 Results Comparison**

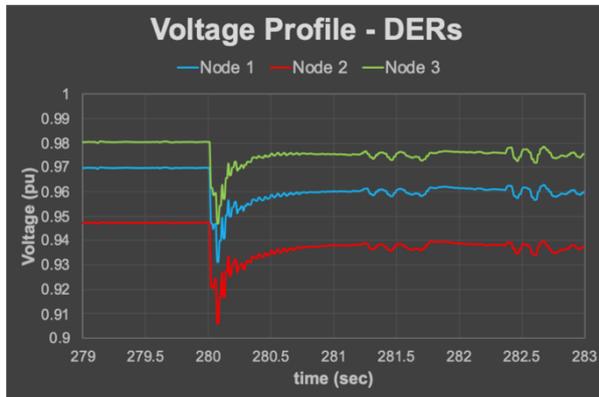
In order to study the differences between the centralized and distributed approaches we need to replicate similar conditions for both. For such purpose we will study the response to a load variation. The specific event that is going to be studied is the one that takes place at  $t = 280$  seconds. This event is the consumption increase in Load-1. This element has been chosen since it takes place once all of the elements are in grid connected mode. This will allow us to evaluate the voltage and reactive power response to load variations. Graph 11: Active Power Variation in Load-1 shows the active power level in Node 4 for the 10-kW variation

in Load-1. We can observe how the change is progressive, which will result in a more contained voltage profile than if it was an immediate change.

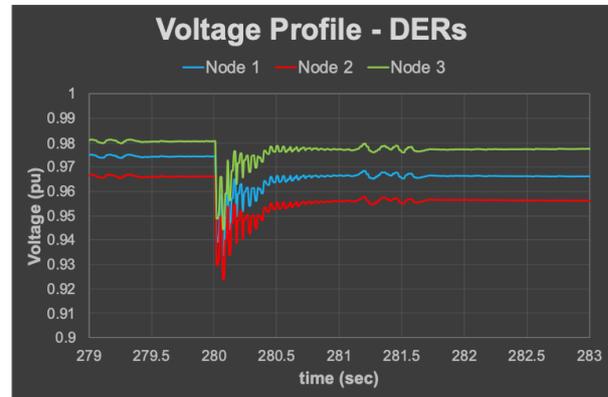


Graph 11: Active Power Variation in Load-1

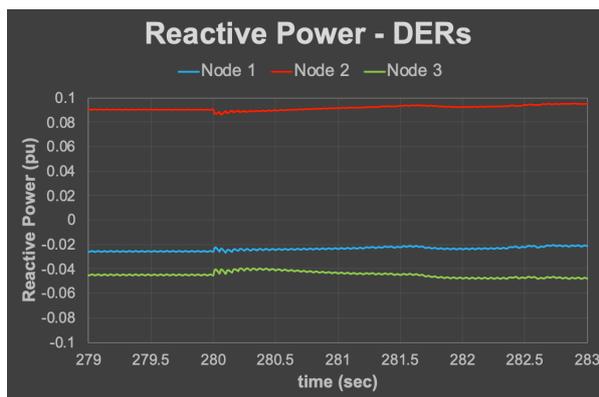
From Graphs 3 to 6, zooming in the event that is going to be studied ( $t = 280$  seconds), we obtain the voltage response to the system's load variation. Graph 12 and Graph 13 show the voltage response for the increase of 10 kW in Load-1. We will also study how reactive power changes in order to perform the voltage control. Graph 14 and Graph 15 show the reactive power levels in the DERs during the load variation.



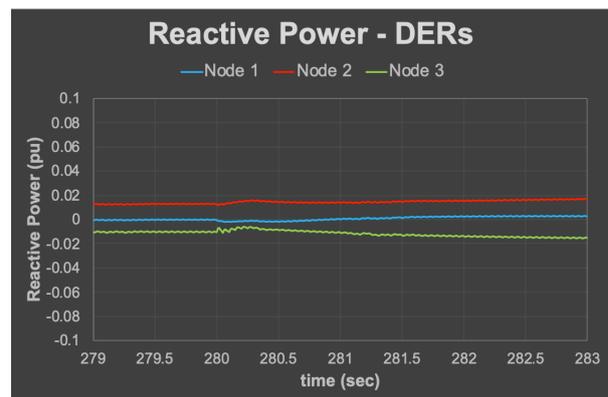
Graph 12: Voltage Profile in the DERs (distributed)



Graph 13: Voltage Profile in the DERs (centralized)



Graph 14: Reactive Power in the DERs (distributed)



Graph 15: Reactive Power in the DERs (centralized)

We will divide the comparison into four categories; overall response to voltage limits violation, signal shape, speed and precision.

1. Overall Response: Studying the general shape of the response, we can notice one main difference which is related to the separation between nodes. In the distributed simulation, shown in Graph 12, we can observe how the voltage profile has a greater distance between voltage lines between the DERs. We can see that this has to do with

the architecture of the distributed approach. As we can see on Figure 25, the system is designed in a linear instead of a radial configuration. This issue, together with the understanding of the distributed algorithm explained in {5.2.2} will lead us to infer that the reason why the separation between nodes happens is the different distances to the failure for each of the nodes. We can observe how the most affected is Node-2. This comes as a result of being 1.5 miles away from Load-1 and 2miles from Load-3. This issue can also be seen in Graph 14 above, where we can observe how Node 2 has the highest reactive power output level. The difference between the two models is that the centralized approach has all the information from the nodes available. This results in more precise results that will make targeting the failure easier. This can be seen in Graph 15 above, where the difference in the reactive power outputs is smaller than in the distributed case.

2. Signal Shape: Considering the shape of the voltage profile in the response to the load variation, we can infer two of the three main decision variables.
  - *Stability*: As regards to the stability of the response, in Graph 12 and Graph 13 we can observe the differences between both approaches. On one side, we can see how the distributed approach develops a steeper curve with fewer oscillation than the centralized one. This is due to the calculations that are required to process the iterative algorithm. Due to this slower response, the system is able to reach a steadier state than what is achieved with the centralized algorithm. On the other side, in the centralized approach we can observe how the higher speed of the algorithm plays against itself. The system is given the command to change the reactive power setpoint, that produces a drastic voltage peak and, before it has the opportunity to accommodate to that voltage level, it receives the next order to change the reactive power level. This results in a faster but more unstable control.

- *System Strain*: The strain on the system is a very severe issue. In order to evaluate it, we need to study the initial variation in the voltage profile at the time instant following the variation in load. From Graph 12 and Graph 13 above we can extract how the distributed approach puts a higher strain in the system. The difference is small at first, due to the more precise response he centralized approach offers. The peak voltage variation is around 20 V for both cases, but during the iterations, the strain the centralized voltage puts on the network is greater than the one from the distributed. This is because of the greater oscillation that was mentioned on the previous point.
  
- 3. Response Speed: We have already introduced how the centralized control has a faster computation speed, which results in shorter change periods and should translate in a faster response. This is not true in this case. As we saw on the previous category, the oscillatory shape of the centralized approach means a faster instantaneous response. But if we take a look at Graph 12 above, we can observe how the distributed approach has a steeper response curve, which in this case means there is going to be a smoother voltage change and the voltage will change faster. The time to stationary in the distributed approach can be estimated as  $t_{\text{dist}} = 280.38 - 280 = 0.38$  seconds, while in the centralized approach, the time is  $t_{\text{cent}} = 280.49 - 280 = 0.49$  seconds. The difference, while small, can make a significant impact in not putting too much stress on the network.
  
- 4. Precision: The parameter that is going to be evaluated to study the precision of the response is the voltage deviation between the instant before the load change and once it has reached steady state and the voltage is constant. From Graph 12 and Graph 13 above we can observe how the voltage deviation in the centralized case is 1.44V and in the distributed case 2.256 V. The difference is 0.8 V, which is a small number but, over an extended period of time can cause an undervoltage fault in the whole grid,

which would have terrible consequences. As we saw previously, this is due to the inability to implement tertiary voltage control, and therefore could not be avoided in this case. If we analyze Graph 7 and Graph 8 above we can observe how the voltage profile in the distributed case has a steeper arc and therefore could incur in further faults produced by this flaw.

## **6.5 CONCLUSIONS FROM 6-NODE RESULT ANALYSIS**

From the result analysis performed in {6.3} and {6.4} we can infer that both approaches, the distributed and the centralized approaches have certain advantages and disadvantages.

- Centralized: the main advantage of this approach is the superior precision compared to the distributed approach: the voltage precision is better than the distributed approach due to the centralized architecture that allows nodes to have all the information available. This will result in a steadier and more reliable network as long as we do not face controller failure, in which case the network would remain unregulated. The major disadvantage about implementing a centralized network is the overall speed and strain in the network. The control architecture provides faster instantaneous response, but since the speed of control is faster than the ability of the system to react to the change in setpoints, the speed is slowed down by the system. This results in an overall slower system. Likewise, we can observe in Graph 13 above how the oscillatory voltage response to the load change can damage the network by placing a higher strain on it compared to the more progressive voltage curve in the distributed case.
- Distributed: the main advantage that this approach has is the quick and stable response to the load variation as compared to the centralized approach. In Graph 12 above we can observe how, while being more irregular, the voltage response to the increase in consumption has a steeper and more stable curve that makes this

approach the most suitable for quick response applications. We can see how the changes in reactive power output are slower due to the iterative algorithms that need to be implemented. This would be an advantage for the approach: while the Arduino is computing the new setpoints for that iteration, the system has time to stabilize. This will make the response more stable, and therefore better than in the centralized approach, which is characterized by an oscillatory and unstable response. The main disadvantage that this model has is the small precision arising from the distributed algorithm. This is due to the inaccuracies that come from computing iterative algorithms like the ratio consensus, defined in {8.1}. The distributed architecture is going to result in iterative algorithms that need to compute approximations and a finite number of iterations in order to compute operations that are much simpler in a centralized approach.

## Chapter 7. ECONOMIC ANALYSIS

### 7.1 INTRODUCTION

In {1.1} we introduced the limitations that the traditional architecture for the implementation of electric network has. The main ones we found were:

- Inability to safely implement DERs into the grid: These networks were conceived with linear architecture without the need to control the resources available. The implementation of DERs with variable generation or consumption as can be seen in Figure 4 and Figure 5, makes for the need to control the grid. Therefore, the chances of the grid undergoing sudden power input variations or even shutdown requires the implementation of a control architecture.
- Overall reliability: Traditional networks are known to have been very reliable, but there is an average of 92 minutes of power outage per year in the Midwest, and 214 minutes in the Northeast [6]. This allows us to improve.
- Economical losses: the cost of reparation and blackouts is making an impact on the benefits that can be obtained from the grid. Beyond that, the possible implementation of DERs will influence the power generation and therefore maximize the economical input that can be extracted from the network.

From these topics we can develop a solution, which as seen in {1.2} consists on the implementation of microgrids, which allow for the reliable implementation of DERs. They introduce many benefits, the main ones being the control architecture and the ability to transform the grid into a non-linear bidirectional mesh. This comes from the fact that by allowing the use of microgrids we are enabling the implementation of consumer level DERs that will help control the network while inputting a greater amount of power into the grid.

## **7.2 EXAMPLE**

### **7.2.1 CASE DESCRIPTION**

We are going to study a synthetic case which consist on the implementation of microgrids in a commercial suburban area in Illinois, United States. This area has had the same grid architecture for a long time and the requirements of the modern electric field has made them doubt their standpoint. In the past they had only a coal power plant, which was the most usual scenario for the United States up until the rise of renewable energies. Recently they felt the need to enlarge their power consumption due to new industries and more people moving into the plants. They thought that it was time for change, so they decided to install four wind turbines and a solar farm. Soon enough they encountered reliability issues. Their outdated power transportation architecture and lack of control functionalities made them be doubtful. The question to be answered is whether the implementation of a new control architecture will have a beneficial impact on their electric bills and reliability issues.

In order to perform this test, we will need to define the conditions under which we are going to be comparing their existing solution with the new proposal. We are going to implement the analysis on an industrialized area from the south of Chicago. As we have already defined, this area can be pictured as an industrial complex which is characterized by a superstore, an automotive factory and residential areas for the factory and superstore workers to live. We are going to translate this in a 2/3 industry 1/3 residential area. This division is going to affect both the electric consumption as well as the expenses for the new installation, as can be seen in E. 11.

Figure 27 shows the architecture that is going to be studied. It is based on the 6-node example defined in Figure 21. We are going to make some assumptions in order to be able to predict the material and soft costs.

- Generator 1 is a 3 MW coal plant.
- Generator 2 is a 1 Mw wind turbine.
- Generator 3 is a set of residential PV installations with a total of 900 kW.
- Load 1 is a residential area characterized by 1000 customers.
- Load 2 is a 50,000 ft<sup>2</sup> superstore (e.g. Walmart).
- Load 3 is a 40,000 ft<sup>2</sup> automotive factory (e.g. Ford Motors).

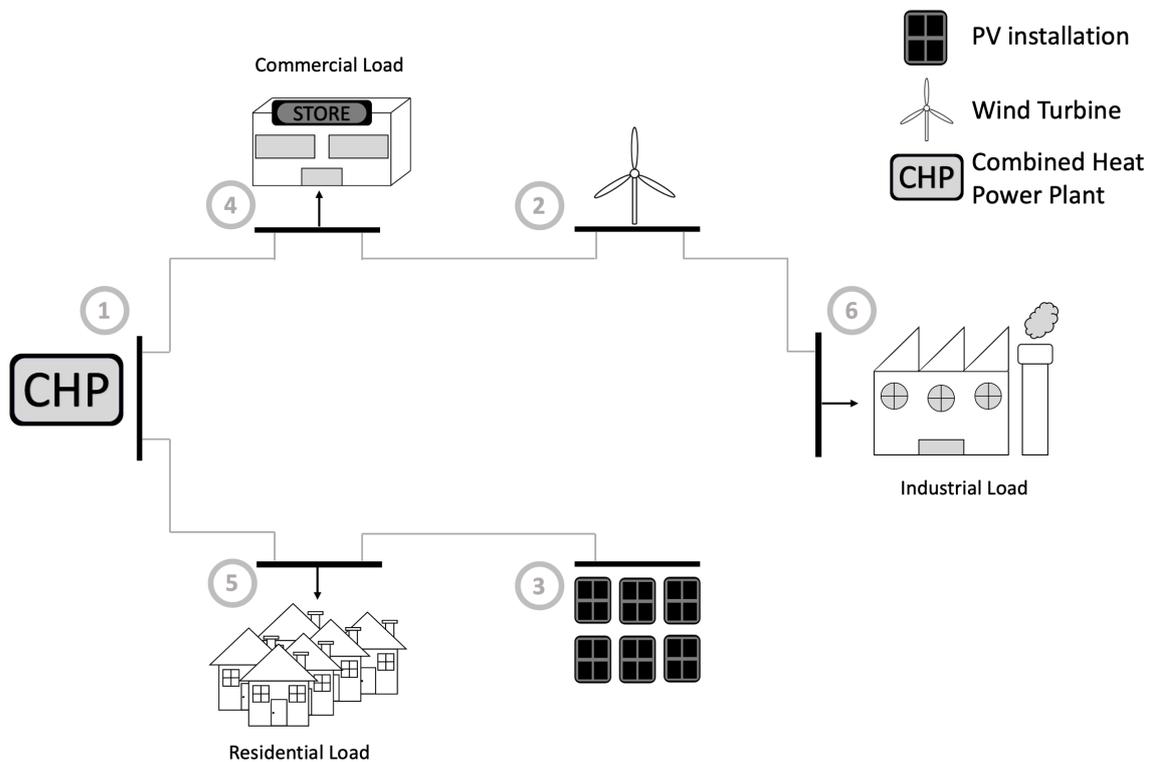


Figure 27: Economic Model Architecture

### 7.2.2 ECONOMIC BASE

In order to develop the economic analysis, we need to understand how the electric market is divided. The main sectors that require for the utilization of microgrids are institutional, community, commercial and utilities[13].

- The Institutional segment is represented by areas that cannot be unsupplied. They need to be able to provide electricity for themselves in case that the institution goes into islanded mode. Some examples of these types of microgrids can be governmental facilities or hospitals. They are mainly dominated by new CHP.
- The Commercial segment is depicted by production plants or industrial complexes. These also need to be able to supply the energy required within their microgrid in order to continue production and avoid major failures.
- The Community segment concerns the regulation of residential areas that have DERs integrated into their area. The importance of implementing microgrids into these areas rests on the inability to be self-regulated since the size of the DERs is usually insufficient to do so. By creating the microgrid all of the DERs are going to be working together in order to supply for the community while having the ability to collaboratively regulate the network.
- The Utility segment is represented mostly by the electric industry. They are related to the distribution of power, which mainly comes from solar farms.

From [13] we can provide the data referred to the average costs of a microgrid depending on the segment that it is going to be installed, which is shown in Table 4. The expected investment for each of the segments depending on the size of the grid, measured in \$/MW produced.

*Table 4: Average Microgrid Cost [13]*

Segment	Average Cost (\$/MW)
Institutional	3,338,666
Commercial	4,079,428
Community	2,119,908
Utility	2,548,080

For the implementation of this economic analysis we are going to consider a mixed microgrid which will be composed of two commercial and one community area. Therefore, taking into account the data obtained in Table 4 we can create an estimation for the costs for this particular microgrid assuming a size of the grid of 1.8MW.

$$E. 10 \quad I_o = \left[ \frac{2}{3} * C_{\text{commercial}} + \frac{1}{3} * C_{\text{community}} \right] P = \$ 6,092,705.64$$

From E. 10 we have obtained that the estimated initial investment is around \$1 million, which we will call  $I_o$ . In order to ensure the veracity of the data we have to break down the costs. These are divided into two categories, material and soft costs.

- Soft costs: the costs responsible of making the network operational. Some examples of these are engineering costs, construction or regulation, which will be required as the first part of the investment.
- Material costs: it is based on the hardware and software that is going to be required in order to operate the microgrid in an optimal state. This is a very wide field, including from generation facilities to control architecture.

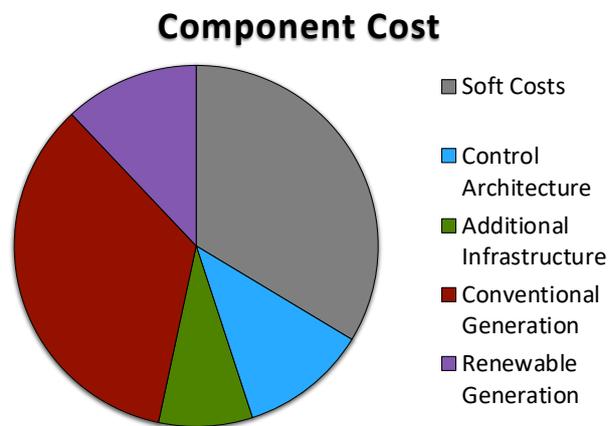
In order to evaluate the costs we have to implement the data obtained from [13] using the assumptions used to define E. 10. From those two we obtain the following equation:

$$E. 11 \quad C_{\text{Component}} = \left[ \frac{2}{3} * C_{\text{Commercial}} + \frac{1}{3} * C_{\text{Community}} \right]$$

From the use of E 11. and the information from [13] we obtain the estimated cost per component in \$ for the proposed example. Table 5 gives the cost divided per component for the test model. Graph 16 shows the distribution of the investment for the model proposed.

*Table 5: Initial Investment*

	<b>Test Example</b>
Soft Costs	2,051,211
Control Architecture	690,507
Additional Infrastructure	507,725
Conventional Generation	2,112,138
Renewable Generation	731,125



*Graph 16: Investment Distribution*

We are going to establish the base results. We will use this information in order to evaluate the investment the tariffs for the payment of electricity generated on microgenerators is divided as follows. From [14] we obtain that the average electric rate for Commercial, Industrial and residential customers is as follows. For commercial electricity the rate is 4.05 cents/kWh, for industrial 1.39 cents/kWh and for residential 10.44 cents/kWh.

According to the data described in the previous paragraph and consumption data obtained from [15], [16] and [17] we can provide a monthly electric bill for the industrial complex.

*Table 6: Estimated Electric Bill [15]–[17]*

Sector	Electric bill per month		
	\$/kWh	kWh	\$
Commercial	0.0405	208333	8437.5
Industrial	0.0139	317000	4406.3
Residential	0.1044	755000	78822

From the information obtained in Table 6 we obtain that the total annual income if there was not a microgrid would be of \$1.099 million.

In order to study the economic impact of implementing a microgrid we need to study the reliability.

### 7.2.3 RELIABILITY ANALYSIS

In order to study the reliability for this example we are going to have to make some assumptions.

- We are going to consider the case where there is no microgrid installed, the components are going to be arranged in series. This results in the components being vulnerable to a single point failure. If one of the components goes down for replacement or repair, the whole network is going to go into blackout mode.
- On the contrary, the components in the microgrid are going to be arranged in parallel. This comes from the idea that the microgrid has a control architecture that is going to intervene in case of failure. If one of the components goes down, the rest of them are going to try to account for it while it is being replaced or repaired. This gives a very considerable advantage to this approach.

- We are going to be assuming a repair time of 2 hours for each of the components in order to simplify the computations.

Figure 28 shows the reliability arrangement for a traditional network and Figure 29 for a network that has a microgrid implemented.

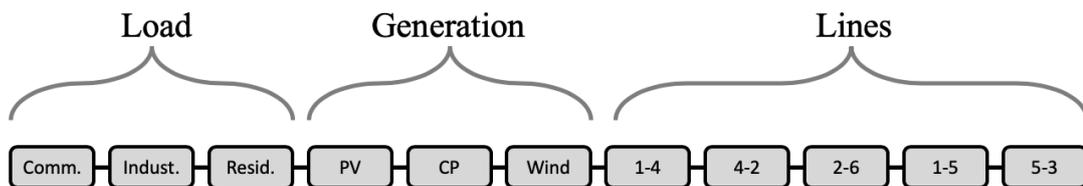


Figure 28: Traditional Reliability Distribution

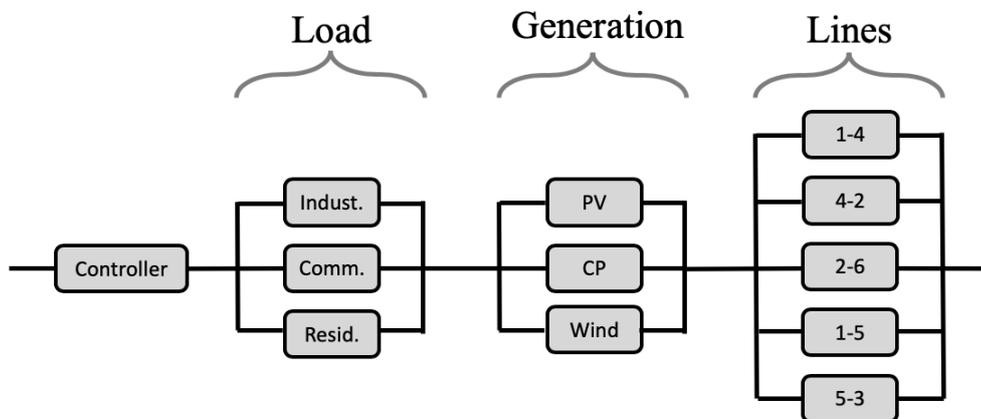


Figure 29: Microgrid Reliability Distribution

We are going to define the time to failure for both of the models. For the simplicity of the computations we are going to assume that for the elements in parallel, the failure rate is going to be  $1/n$  where  $n$  is the number of components in the parallel structure.

$$E. 12 \quad \lambda_{\text{Trad}} = 5\lambda_{\text{Line}} + 3\lambda_{\text{Gen}} + 3\lambda_{\text{Load}} \text{ [failures/yr]}$$

$$E. 13 \quad \lambda_{\text{Microgrid}} = \lambda_{\text{Controller}} + \frac{1}{5}\lambda_{\text{Line}} + \frac{1}{3}\lambda_{\text{Gen}} + \frac{1}{3}\lambda_{\text{Load}} \text{ [failures/yr]}$$

Where  $\lambda_{\text{line}}$ ,  $\lambda_{\text{gen}}$ ,  $\lambda_{\text{load}}$  and  $\lambda_{\text{controller}}$  are the failure rates for lines, generators, loads and controllers respectively. We are going to assume failure rates of 0.34 (f/yr) for generators and loads, 0.5(f/yr) for controllers and 0.28(f/yr) for the lines. Therefore, we obtain from E. 12 and E. 13 that the failure rates for the traditional and proposed architectures are  $\lambda_{\text{traditional}}=3.44$  (f/yr) and  $\lambda_{\text{microgrid}}=0.783$  (f/yr). With the failure rate we can obtain the non-delivered energy (NDE) that is going to be the differentiating factor for the investment study[18].

$$E. 14 \quad \text{NDE} = \lambda_i * r * P$$

Where  $r$  is the repair time and  $P$  is the power of the system which in this case is a 1.7782 MW network. From the implementation of E. 12 and E. 13 together with E. 14 we obtain that:

$$\text{NDE}_{\text{traditional}} = 12.234 \text{ MWh}$$

$$\text{NDE}_{\text{microgrid}} = 2.785 \text{ MWh}$$

$$E. 15 \quad \Delta\text{NDE} = 12.234 - 2.785 = 9449.57 \text{ MWh}$$

## 7.2.4 INVESTMENT EVALUATION

In order to evaluate the viability of the project we have to derive the yearly savings that could be obtained and compare them with the initial investment.

From the use of Table 11 and E. 15 we can obtain that the average savings in non-delivered energy per year as a function of the period. These savings are shown in Table 7. The total. Apart from this we are going to assume that, since by implementing a microgrid control, the

failure rate is going to decrease, the man-hour and repair is going to go be reduced as well. We are going to assume that value to be a 5% of the yearly electric bill from the traditional scenario. This leaves us with a material and human failure savings at the beginning of each period of:

$$E. 16 \quad \text{Savings}_{\text{faults}} = 1,099,989 * 0.05 = \$54,999.5$$

*Table 7: Yearly NDE Savings*

Period	Savings (\$/yr)
yr1 -- yr3	7,203
yr4 -- yr6	8,403
yr7 -- yr9	9,604
yr10 -- yr12	10,804

Throughout this project we have been explaining the importance of microgrid virtual test devices. These devices not only have a huge importance in versatility and scalability for testing microgrids before implementing them, but they also offer a grand cut in testing expenses. We have seen how the microgrid testing process, which usually takes weeks to develop has been reduced to days. This reduction has an impact not only on material expenses that come from not having to install the microgrid, but also on the human-hours that it takes to do the installation and testing. We have to take into account the reduction in development research. We are going to assume that this amount is a 50% of the investment in soft costs and Control Architecture. Using the values in Table 5 we obtain development costs.

$$C_{\text{development}} = [I_{\text{softs}} + I_{\text{Control}}] * 0.5 = \$1.37 \text{ million}$$

From the development costs we can infer the savings that come from utilizing a virtual HIL device from microgrid testing. The reduction in resources and men-hour gives us an estimated savings of 60% of the development costs.

$$E. 17 \quad Savings_{HLL} = C_{development} * 0.6 = \$ 822,515$$

Since this is a long-term project, we expect the microgrid to evolve, and with that will come more development expenses. Therefore, the savings due to virtual microgrid simulation as well as for faults as described in E. 16, will be applied every three years according to research and development in a fast-changing world.

We have seen how some European countries such as Germany are encouraging the race towards a renewable future by applying a surcharge of 6.405 cents per kilowatt hour in the electric bill [19]. This has encouraged change. We are walking into a period of instability in the which we need to seek a green future. We are going to translate this into a surcharge of 20% increase in the electric bill every three years in matter of green tax reduction and green initiative incentive for the microgrid perspective since it is the one that would be available for further modernization and installation of renewable resources. Table 11 shows the updated electrical monthly bills for each of the periods from the microgrid installed scenario. Finally, by comparing the updated electric bills with the traditional one we can compute the increase in income due to the implementation of microgrids. Shows the yearly variation in income by period.

*Table 8: Yearly Income Variations*

Period	$\Delta$ income (\$/yr)
yr1 -- yr3	219,998
yr4 -- yr6	439,996
yr7 -- yr9	659,994
yr10 -- yr12	879,992

From all of the assumption made above and the calculations that have been developed, we can compute the total savings breakdown that can be obtained by implementing this method. These are going to be divided into 3-year periods assuming this to be a 12-year project. From E. 16 and E. 17 at the beginning of every period we are going to have an income of \$877,515.

Shows the yearly income divided in periods due to green tax return and non-delivered energy.

Table 9: Yearly Income

Period	income (\$/yr)
yr1 -- yr3	227,201
yr4 -- yr6	448,399
yr7 -- yr9	669,598
yr10 -- yr12	890,796

In order to evaluate the investment, we have to consider the difference in both regarding a present value, which will be the Net Present Value (NPV) with an assumed discount rate of  $d=7\%$ , which is a common discount rate in electric projects. We are going to consider this project to have a life of 12 years. Figure 30 shows the cash flow diagram for the problem. From this we are going to derive the NPV equation.

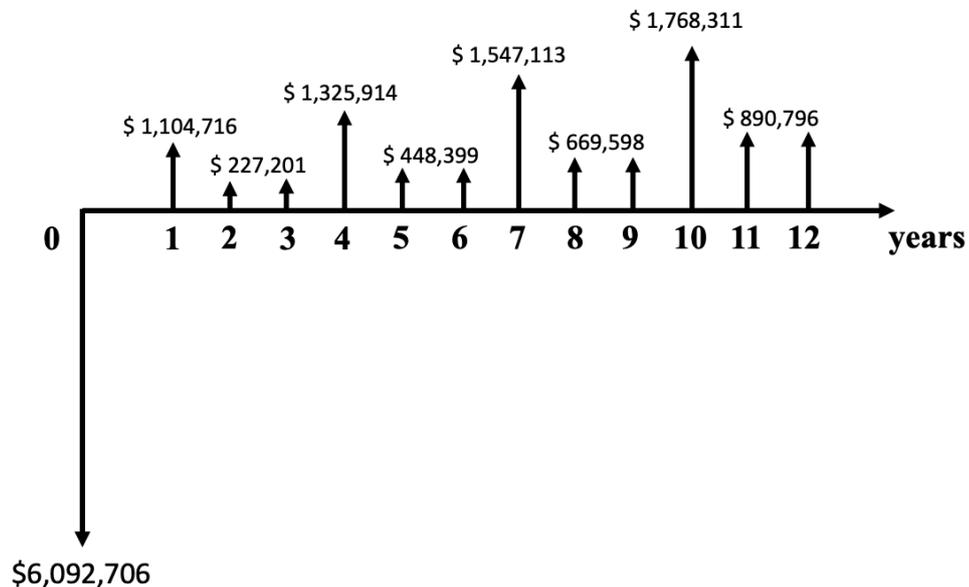


Figure 30: Cash Flow Diagram

$$NPV = -6,092,706 + \frac{1,104,716}{1+d} + \frac{227,201}{(1+d)^2} + \frac{227,201}{(1+d)^3} + \dots + \frac{890,796}{(1+d)^{12}}$$

$$NPV = \$ 388,719$$

### **7.3 CONCLUSION OF THE ECONOMIC ANALYSIS**

We have obtained a positive NPV for this example, which means that this is an overall worthwhile investment. This means that after a period of 12 years, we would be making profit from the investment we made.

To conclude, the implementation of microgrids has known advantages, not only economical but also environmental. This comes from the ability to control generation and loads through a control architecture. We need to be able to control the network since renewable-based generation is less stable, therefore being more prone to failure. Throughout this analysis we have seen how by implementing a microgrid we can enlarge the GDP of the country while advancing in the green race. We have supported this statement with the implementation of a microgrid in an industrial complex, which after 12 years would become profitable. We think that implementing microgrids into our network can help not only obtain more profit but also to leave a better world for the upcoming generations.



## Chapter 8. ALGORITHMS

### 8.1 *RATIO CONSENSUS*

#### 8.1.1 DESCRIPTION

For the implementation of the second stage in the voltage control algorithm there needs to be collaboration in between the neighboring nodes. This comes as a consequence that the target node does not have enough reactive power available so as to manage the fault by itself. For the system to go back to normal, the remaining reactive power needed is provided by the remaining neighbors.

The partitions of the reactive power remaining have to be chosen regarding the reactive power levels and limits. This process can be executed following a centralized approach. This process is based on the computing of the average of the raising or lowering potential of each DER. However, we want to develop a distributed algorithm. This algorithm is the fair split ratio consensus.

In the algorithm, the desired output for each operation is going to be the ratio in between the state variables  $\mu_j[k]$  and  $\nu_j[k]$ , which will be denoted by  $\gamma_j[k]=(y_j[k] / z_j[k])$ . After each iterations, the state variables need to be updated as follows from [20]:

$$y_j[k + 1] = \sum_{i \in \{\mathcal{N}_j^-\} \cup \{j\}} \frac{1}{1 + \mathcal{D}_i^+} y[k]$$

$$z_j[k + 1] = \sum_{i \in \{\mathcal{N}_j^-\} \cup \{j\}} \frac{1}{1 + \mathcal{D}_i^+} z_j[k]$$

This algorithm shows that the value at each iteration is going to be updated according to the value of  $\mu_j[k]$  and  $v_j[k]$  in iteration  $k$  from the neighboring nodes, denoted by  $\mathcal{N}_j^-$ , divided by the weight of the node, which is the outdegree of node  $i$   $\mathcal{D}_i^+ + 1$  since it is also accounting for itself. The outdegree is the number of nodes that are part of the set of neighbors of node  $i$ .

```

If(_G->isinneighbor(alsb,i))
{
    Float inmu = _getmufrompacket();
    Float insigma = _getsigmafrompacket();
    Uint8_t neighborid = _getneighboridfrompacket();

    Uint8_t nodeid = s->getid();

    Float Mudiff = inmu - s->getnumin(i);
    Float sigdiff = insigma - s->gettau(i);

    Iny += Mudiff;
    S->setnumin(i,inmu);
    Inz += sigdiff;
    S->settau(i,insigma);

    Node_check[neighborid -1] = 1;
}

S->setymin((s->getymin()/Dout) + iny);
S->setmumin(s->getmumin() + (s->getymin()/Dout));
S->setz((s->getz()/Dout) + inz);
S->addtosigma(s->getz()/Dout);

```

The values of  $\gamma_j[k]$  from each node will converge to a common value, since as we mentioned, we are computing the average of the values. The value of  $\gamma_j[k]$  at the last iteration will asymptotically approach to:

$$E. 18 \quad \gamma_j = \lim_{k \rightarrow \infty} \gamma_j[k] = \frac{\sum_{i=1}^n y_i[0]}{\sum_{i=1}^n z_i[0]}, \quad j \in \mathcal{V}_c$$

By computing this algorithm paired to a correct initialization of the state variables, we will obtain enough information about the system so that we can compute the weighted

distribution of reactive power addition or consumption needed so as to supply the required amount that will make the network go back to its operating conditions.

In the ideal case for the algorithm, the number of iterations would be infinite, but in a real simulation, the number of iterations that the algorithm runs are set by various criteria. The criterion that will be used in this project is the max-min consensus.

### 8.1.2 ALGORITHM TEST

In order to demonstrate the implementation of the algorithm we are going to be performing the mean of three numbers that are going to be inside each of the nodes. This is achieved by allocating the value 1 for the  $z$  in E. 18.

We are going to use a linear scheme as the one that is described in Figure 31 for the test scenario. The values that are going to be used are designed by the variable  $Y$ .

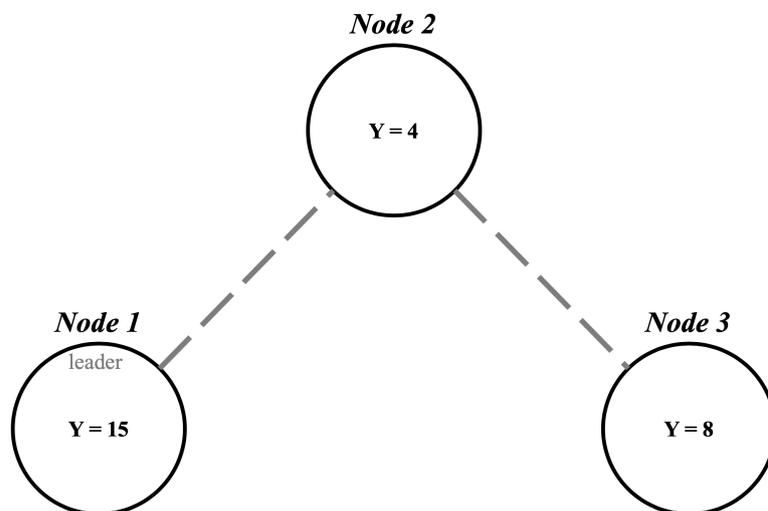


Figure 31: Architecture for RC test

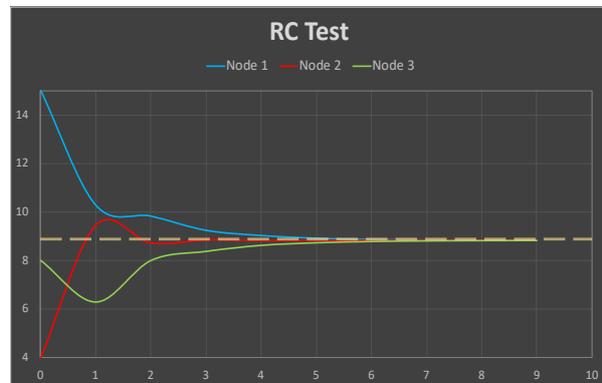
By implementing the ratio consensus with a centralized approach, we would only need to compute the average of the three values as follows:

$$\gamma = \frac{Y_1 + Y_2 + Y_3}{3} = 9$$

Since we are going to develop the algorithm with a distributed approach, the algorithms that we are going to implement are the ones described in {8.2.1}. Graph 17 and Table 10 show the results obtained for the test.

Table 10: Ratio Consensus Test Results

Iteration	Node 1	Node 2	Node 3
0	15	4	8
1	10.28571	9.454546	6.285714
2	9.836068	8.719101	8.000001
3	9.252444	8.847683	8.375731
4	9.039684	8.826069	8.623811
5	8.927517	8.829722	8.730023
6	8.876149	8.829119	8.78238
7	8.851441	8.829197	8.806918
8	8.83977	8.829209	8.81863
9	8.834206	8.829178	8.824168



Graph 17: Ratio Consensus Test

If we analyze Graph 17 and Table 10 we can observe how the value from each of the nodes tends to the average value. This implementation has two main drawbacks. The first one is the time needed to compute the RC algorithm, which is greater than that needed to implement the centralized operation. The second is the deviation from the target value due to packet drops in communication and lack of infinite iterations. As a result, the value to which the nodes tend is 8.83 instead of 9. This comes from the inability to have all the information from the DERs available at each node since it is a distributed approach.

However, the main advantage is the ability to perform operation in a distributed architecture, which allows for a distributed network control, which is the main objective of this project.

### 8.1.3 CODE

The code from the ratio consensus algorithm is allocated in {ANNEX V}.

The structure of the code is as follows. First, we have to define which of the nodes is going to be the leader. The leader node works the same way as the non-leader nodes. The only difference is that the leader node will be set as time Reference for the collaborative algorithm. Once the leader is set and the nodes have synced, the ratio consensus algorithm starts. The first stage of the code is the initialization of the nodes, which is done in the function “fairsplitratioconsensus\_RSL”. Once initialized, the algorithm will differentiate in between the leader node and the non-leaders, calling distinct functions (leaderfairsplitratioconsensus\_RSL or nonleaderfairsplitratioconsensus\_RSL). In the second stage, the nodes will start by sharing their time through the XBee protocol. The new time for all of the Arduinos will be the one that the leader has in order to sync all of the nodes. Once the nodes are synced, they will enter the body of the algorithm, that is the function “ratiomaxminconsensus”. Once in the ratio maxmin consensus, the Arduino will develop the algorithm describe previously. The result ( $y$ ) will be sent to the Arduino and the ratio consensus will be finished.

## 8.2 *MAXMIN CONSENSUS*

When computing the ratio consensus algorithm required to perform the Second Stage voltage control algorithm, a number of iterations has to be performed. Through experimental observation we have studied how after a finite number of iterations, around thirty for a normal operation, the data type that is being used in the C++ structure (float) is too small. After each iteration, since the computations that are taking place are non-linear, the number of decimals grows. Therefore, the float datatype, which is 4 bytes big and has 7 digits of precision, is not adequate for an infinite precision that would be necessary so as to arrive at an exact ratio consensus value[21]. From this we can state that there need to be set a limit for the number of iterations that the algorithm will have to be computed for.

In. Order to determine the optimal number of iterations that the algorithm is going to run in order to arrive at the results, we have two approaches:

1. Basic Approach: this approach is based on the empirical method. We will compute a number of computations of the ratio consensus algorithm for a number of precisions and numbers. According to that database, each time we have to implement the RC algorithm we would have to choose a specific number of iterations that will give the best precision without incurring into data type issues.
2. Maxmin Consensus: this is the method that is going to be used in order to implement the RC algorithm in the voltage control. The key characteristic of this approach is that the number of iterations can be infinite. The length of the algorithm will be determined by a precision variable that will be chosen in accordance with the precision the user desires.

So as to implement this method we need the difference between the maximum and the minimum value that the ratio has at a particular iteration of the RC algorithm. The ratio that is going to be studied is the value of gamma, which is defined in E. 18.

The maxmin algorithm is going to be compute every number of iterations in order to require less computational waste. In this case we are going to implement it every three iterations. However, since the purpose of the algorithm is to improve precision, we will implement it after the 5<sup>th</sup> iteration, skipping the 3<sup>rd</sup> iteration. By implementing the algorithm with this delayed start, we will be getting rid of possible fake positives in the ratio consensus, therefore maximizing precision and resiliency.

### 8.2.1 DESCRIPTION

The algorithm is going to start by defining the local maximum and minimum values to the value that  $\gamma$  has at the specific iteration[21]:

$$\gamma_{max}^i[1] = \gamma^i[n - 1]$$

$$\gamma_{min}^i[1] = \gamma^i[n - 1]$$

Once the variables are initialized, the node is going to broadcast the maximum and minimum values to its neighboring nodes. Once the node receives the values from the neighbors, it will compare the values with its own, updating the local variables as follows[21]:

$$\gamma_{max}^i[k] = \max_{j \in \mathcal{N}[i]} [\gamma_{max}^j[k - 1]]$$

$$\gamma_{min}^i[k] = \min_{j \in \mathcal{N}[i]} [\gamma_{min}^j[k - 1]]$$

This process will be repeated a number of times in order to ensure that all of the nodes agree on the maximum and minimum values. First, we need to define the concept of the diameter of the network. The diameter of the network is the number of communication steps that the system has to conduct in order to transmit the information from one end of the network to the other. For example, for the network defined in Figure 25, with a diameter is 2. This is due to the fact that the network is arranged in a linear configuration. The largest distance the information has to travel is 2, from one end to the other according to this, we will perform

the information broadcast and comparison  $n = (d+3)$  times. This number is chosen in order to account for packet drops.

Once the algorithm has been run for  $n$  times, the program will compute the difference between the maximum and minimum values, that is denoted epsilon ( $\varepsilon$ ) and can be computed as in E. 19. Epsilon is the measure that is going to be implemented for precision. Throughout this project the values that are going to be evaluated are small. Since we are going to be operating with voltage errors that exceed over a 5% the rated value, the values we are working with are in the order of 0-1 since we are obtaining per-unit values. The epsilons that have been applied for this test are  $\varepsilon=0.01$  and  $\varepsilon=0.0001$ .

$$E. 19 \quad \varepsilon[k] = \gamma_{max}^i[k] - \gamma_{min}^i[k]$$

Once the maxmin Algorithm is computed, the value for  $\varepsilon$  will be compared with the target value that has been defined previously.

$$\begin{cases} \text{Continue the RC} & \varepsilon[k] > \varepsilon_{target} \\ \text{finish RC} & \varepsilon[k] < \varepsilon_{target} \end{cases}$$

If the ratio consensus algorithm has to finish, the value that is going to be given back is:

$$\gamma_{final}^i[n] = \gamma^i[n - 1]$$

For each of the nodes.

## 8.2.2 ALGORITHM TEST

For this example, we are going to run the same test that was implemented in {8.1} for testing the ratio consensus algorithm. The scheme that is going to be used is the same as in the RC. Figure 32 shows the architecture scheme and the values that the Y are going to take.

For this study case, the rule which states that the algorithm will start at the 6<sup>th</sup> iteration is ignored in order to better illustrate the behavior of the algorithm. We are going to be performing the mean of three numbers that are going to be inside each of the nodes.

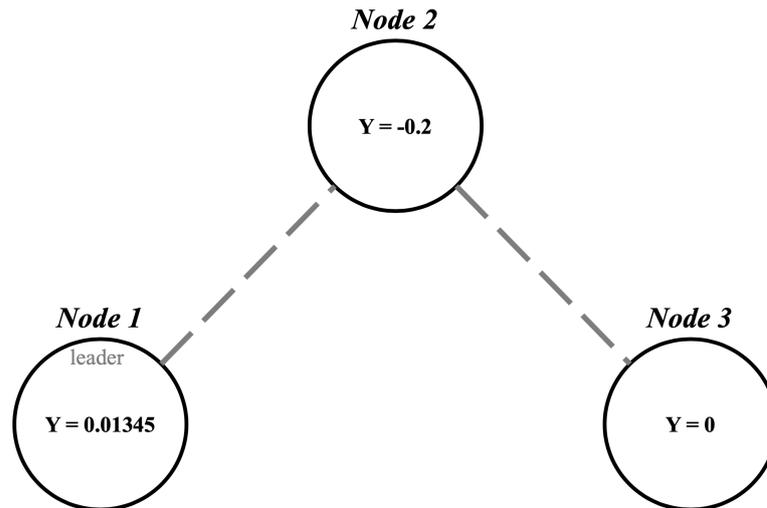
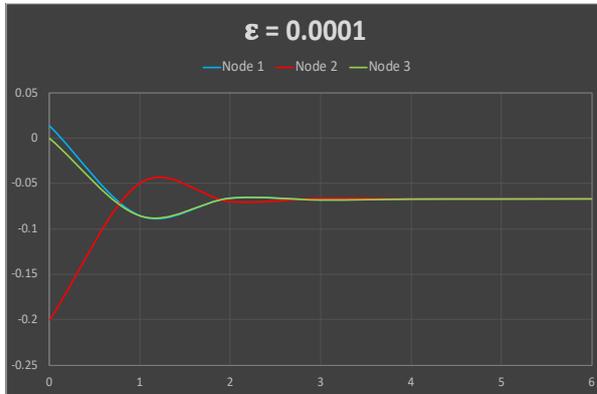
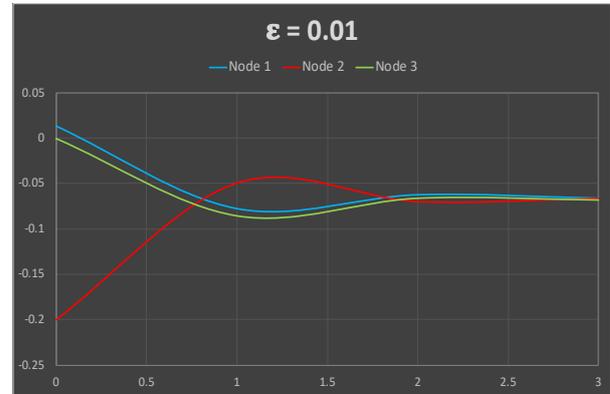


Figure 32: Architecture for maxmin test

By computing the ratio consensus algorithm together with the maxmin consensus we obtain the results in Graph 18 and Graph 19 . We have computed the results with two levels of precision. The first one with a greater precision ( $\varepsilon=0.0001$ ) and the second one with less precision ( $\varepsilon=0.01$ ) but as good as the first one.



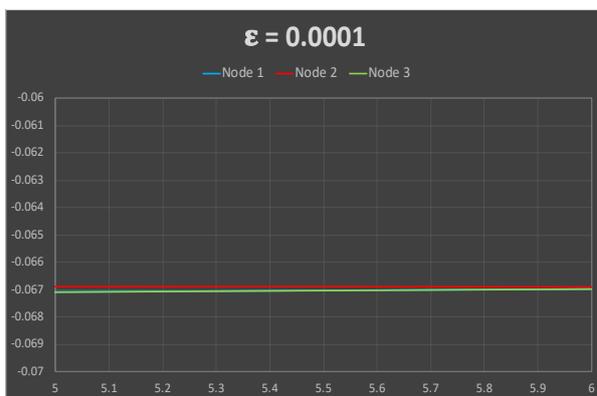
Graph 18: maxmin results for  $\varepsilon=0.0001$



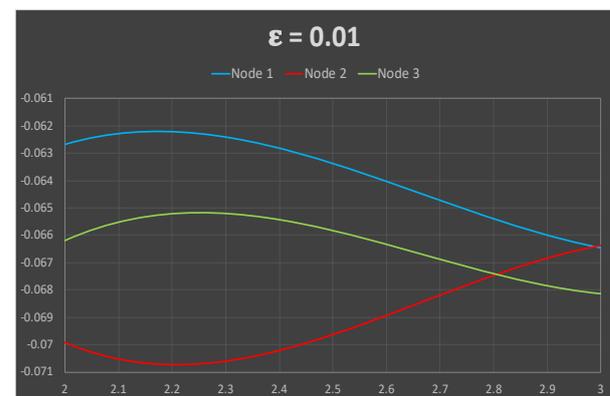
Graph 19: maxmin results for  $\varepsilon=0.01$

Examining the results, we can observe how Graph 19 converges faster to the desired result which is  $Y = -0.062183$ . However, from Graph 18 we can observe how the result converges in more iterations, which translates in more computing time, which is a drawback of choosing a higher precision.

But speed is not the only concern for the implementation of the algorithm. We have to evaluate the precision for the specific application. If we consider both graphs and evaluate the last iteration for the both of them, we can see the precision. In order to ensure the equity for both values, we are going to choose a window of  $\Delta=0.01$  for both examples.



Graph 20: Precision results for  $\varepsilon=0.0001$



Graph 21: Precision results for  $\varepsilon=0.01$

From Graph 20 we can observe how by using a smaller  $\varepsilon$ , we increase precision obtaining in the three nodes a closer result for the ratio consensus. If we analyze Graph 21 we can see how the results that have been obtained are very different for each of the nodes, oscillating around the desired value.

### **8.2.3 CONCLUSION FOR THE CHOICE OF $\varepsilon$**

Taking into account the results obtained in {8.2.2}, we can state that the choice of  $\varepsilon$  has to be done taking into account not only the time required to reach ratio consensus, but also the desired precision for the algorithm.

If the main objective of the algorithm is the speed, there would need to be some sacrifice in precision, since more precision means more iterations, and hence, the algorithms would require a greater time to compute the ratio consensus. The results for this sort of application can be seen in Graph 19 and Graph 21.

On the contrary, if the main purpose of the algorithm is to obtain a precise result across the nodes at expenses of the time being greater, the optimal  $\varepsilon$  that should be chosen would be smaller in order. The results for this sort of application can be seen in Graph 18 and Graph 20.

In conclusion, the optimal approach that is the one that fits the purpose of the algorithm for your specific needs. Throughout this project we are going to be implementing a resilient model of the ratio consensus algorithm. As proceeds, we are going to choose a smaller  $\varepsilon$ . This is going to allow us for a more precise result. This is the best approach due to the small numbers that are going to be studied in the voltage control algorithm.

## 8.2.4 CODE

The code from the ratio consensus algorithm is allocated in {ANNEX V}.

The structure of the code is as follows. The maxmin consensus code is going to be implemented every three iterations of the ratio consensus. Once the code starts by calling “maxminconsensus\_RSL”, each of the nodes is going to be divided into leader and non-leader and will proceed to its specific function (“leadermaxminconsensus\_RSL” or “nonleadermaxminconsensus\_RSL”). Once the nodes are synched for communication, the body of the algorithm starts with the function “maxmin\_RSL”. This function is going to start initializing *gammaMax* and *gammaMin* to the value of the gamma at the specific iteration of the RC code. Once initialized, the node will receive the Max and Min values from its neighboring nodes. It will compare the received values with its own ones and update them. Once the comparison is over, the node will broadcast the package with the information to the other nodes. This is done over “broadcastmaxminpacket\_RSL”. The process is repeated *n* times, being *n* the diameter of the network plus a constant, which in this case has value 3.

Once the code is iterated *n* times, the value of epsilon is stored into each of the nodes. The program will continue with the ratio consensus function and it will compare the epsilon obtained with the value entered by the user. If the one obtained is smaller than the value given, the algorithm will finish. Otherwise, the RC algorithm will continue its iterations.

## Chapter 9. CONCLUSIONS AND FURTHER STEPS

### 9.1 *FUTURE STEPS*

The future steps for this project can be divided into two subcategories:

- Short-term: The short-term goals are Three. The first one is the improvement in resiliency for the algorithms in order to narrow the gap with the centralized voltage control algorithm. The second one is the merge of all the codes that need to be implemented in order to regulate a network (voltage control, frequency deviation, feasible flow...). once all the algorithms are implemented and work properly, they will be tested into the same microgrid that will be the 6-node architecture that has been implemented throughout the paper. The third is the implementation of the voltage control code into higher order networks. The best example for this is the banshee network shown in figure. Compared to the 6-node example, shown in Figure 21, it poses a great step in complexity.
- Long-term: the long-term goals for this project is the creation of a scalable approach to distributed microgrids. From implementing synchronization between nodes to implementing all the algorithm like the voltage control or the optimal dispatchment problem. This proposition will change the way we manage microgrids. And with the help of Typhoon, this research project will be something that will change the industry.

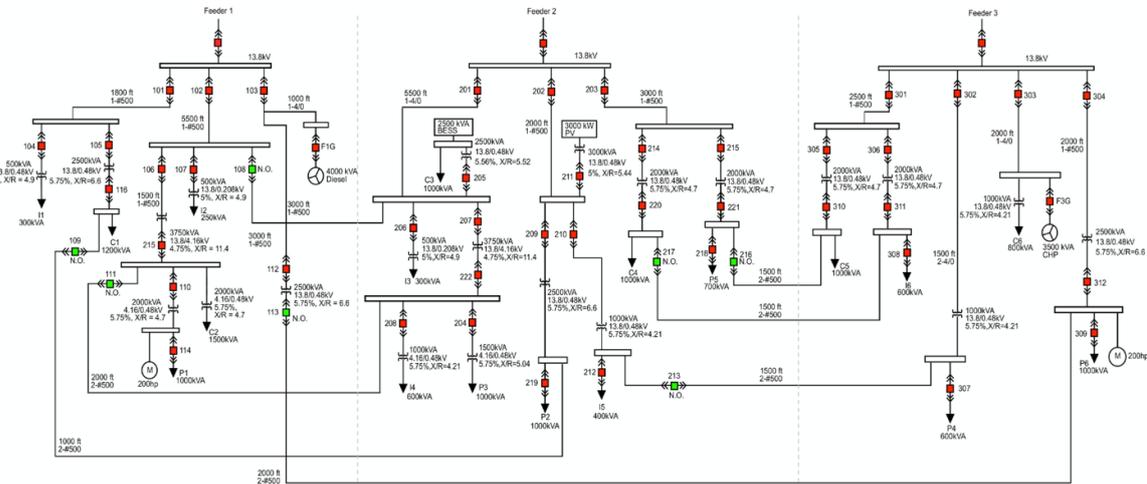


Figure 33: Banshee Network Model

## 9.2 CONCLUSION

We live in an energy hungry modern reality where power needs to be understood as a non-linear concept. This concept can be seen in Figure 6 and after studying the implication this approach has, we understand the changes we need to implement into our train of thought. As the concept of power generation and consumption has been reshaped, our way to manage it has to do so too. We need to be able to control the network, and the best method to do it is with the implementation of microgrids. Microgrids are clusters of DERs (Distributed Energy Resources). DERs are self-regulating energy resources that can maintain the system working even when the system is in islanded mode as long as it has one DER. The problem comes with the growth in size from the microgrid, which makes the system unable to regulate itself. At that point we need to define a control architecture, that as we have seen can be either distributed or centralized.

Once we have the microgrid defined, we need to be able to set the control architecture, create the control algorithms, test them and obtain results from it. In order to do so, there needs to

be a simulation. Using a physical microgrid has many disadvantages that makes it unsustainable expensive. There are not many normalized microgrids from the which we could obtain contrasted results for our tests. They are prone to failure under erroneous conditions. Therefore, if there was an error in the controller's code that led the network to critical operating condition, the network could be damage or even left operational. This could incur in an expensive repair process, defeating the purpose of testing the network before implementing it. For these reasons, for this project and all the ones that will study these algorithms and architectures, the team will be using simulated microgrids.

Simulated microgrids are the future of power research projects and microgrid testing. They can be designed to simulate either centralized or distributed control architectures. There is a growing number of research programs that use this method, which will result in a larger number of normalized models that will speed up testing and research while keeping the costs at its minimum. The tool that has been chosen for this project is the testing in Typhoon HIL simulators in order to create a distributed network that can be compared with the centralized counterpart in order to deliberate the viability of the change from centralized to distributed control architectures. Throughout this paper we have studied the implementation of the voltage control algorithm for both control architectures. This has allowed us to compare both of them in order to draw conclusions regarding the viability of implementing distributed rather than centralized microgrids.

We need to evaluate the change from an economic perspective. Is it worth changing the control architecture to accommodate for mother challenges? In order to answer this question, we have developed an economic analysis based on the 6-node example that has been defined throughout this paper. This analysis has shown the viability of the change from traditional grid configuration to a microgrid approach. For the case we have studied the change for a small industrial area that is defined in Figure 27. Once the assumptions are set and the economic breakdown has been performed, we have obtained a positive NPV, which means that the investment for the test scenario is profitable. From this analysis we have seen how

the implementation of microgrids not only is the approach that will lead us to a renewable future, but according to changes in politics and green initiatives, but the one that will generate the most wealth.

As we have seen throughout the result analysis for the voltage control response, the best approach is implementing a microgrid whose control architecture distributed. This comes from the fact that the control is going to have a more stable, less oscillating and with faster recover time response. The result analysis, while not too differentiating, has indicated the slight superiority of the distributed approach over the centralized architecture that is currently being used in microgrids all around the world. We have seen how the higher accuracy from centralized models has been drawn back due to its poor stability and overall response speed.

From a reliability point of view, we need to take into account controller failure. A distributed approach is going to provide a superior reliability. This is due to the fact that a centralized approach has a single-point vulnerability, which makes the system very susceptible to failure. If the controller fails, the microgrid is unregulated until the controller is repaired. However, if we implement a distributed architecture, due to the distributed control architecture, that requires to implement a controller in each of the DERs, we are going to have a control mesh that can account for multiple simultaneous failures thanks to iterative algorithms and predictions that can account for node malfunction or communication problems. This approach is going to provide a more reliable and secure microgrid.

Taking into consideration the previous statements, the approach that we have found to be the most adequate for our modern, ever-changing world is the implementation of microgrids. Those microgrids have to be tested using Hardware In the Loop simulation in order to minimize costs and simulation time. The control architecture that this microgrid should implement is a distributed approach that is going to be the most reliable while having a fast, stable and smooth response to voltage violations.

## Chapter 10. REFERENCES

- [1] B. A. Robbins, C. N. Hadjicostis, and A. D. Domínguez-García, “A two-stage distributed architecture for voltage control in power distribution systems,” *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1470–1482, 2013.
- [2] B. V. Ermolenko, G. V. Ermolenko, Y. A. Fetisova, and L. N. Proskuryakova, “Wind and solar PV technical potentials: Measurement methodology and assessments for Russia,” *Energy*, vol. 137, no. April, pp. 1001–1012, 2017.
- [3] Y. Mou, H. Xing, Z. Lin, and M. Fu, “Decentralized optimal demand-side management for PHEV charging in a smart grid,” *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 726–736, 2015.
- [4] S. Hoff, “Detail @ Www.Eia.Gov,” 2020. [Online]. Available: <http://www.eia.gov/todayinenergy/detail.cfm?id=25372>.
- [5] N. Carolina, “2018 Average Monthly Bill- Residential 2018 Average Monthly Bill- Residential,” pp. 101–102, 2018.
- [6] S. M. Amin, “U.S. Electrical Grid Gets Less Reliable,” *IEEE Spectr.*, 2011.
- [7] KSFR, “Retake Our Democracy Host Paul Gibson Weighs in on the Democratization of Energy | KSFR.” [Online]. Available: <https://www.ksfr.org/post/retake-our-democracy-host-paul-gibson-weighs-democratization-energy>. [Accessed: 17-Apr-2020].
- [8] R. H. Lasseter, “MicroGrids: A Conceptual Solution,” *2002 IEEE Power Eng. Soc. Winter Meet. Conf. Proc. (Cat. No.02CH37309)*, vol. 1, pp. 305–308, 2002.
- [9] O. Azofeifa *et al.*, “Controller Hardware-in-the-Loop Testbed for Distributed

- Coordination and Control Architectures,” 2020.
- [10] “Virtual HIL Device - Typhoon HIL.” [Online]. Available: <https://www.typhoon-hil.com/products/virtual-hil-device/>. [Accessed: 08-Apr-2020].
- [11] P. N. Vovos, A. E. Kiprakis, A. R. Wallace, and G. P. Harrison, “Centralized and distributed voltage control: Impact on distributed generation penetration,” *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 476–483, 2007.
- [12] S. Lu *et al.*, “Centralized and decentralized control for demand response,” *IEEE PES Innov. Smart Grid Technol. Conf. Eur. ISGT Eur.*, pp. 1–8, 2011.
- [13] J. Giraldez *et al.*, “Phase I Microgrid Cost Study : Data Collection and Analysis of Microgrid Costs in the United States Phase I Microgrid Cost Study : Data Collection and Analysis of Microgrid Costs in the United States,” no. October, 2018.
- [14] Electricity Local, “Chicago, IL Electricity Rates | Electricity Local,” 2020. [Online]. Available: <https://www.electricitylocal.com/states/illinois/chicago/>. [Accessed: 27-May-2020].
- [15] OUC, “Business Energy Advisor | Manufacturing Facilities,” 2017. [Online]. Available: <https://ouc.bizenergyadvisor.com/article/manufacturing-facilities>. [Accessed: 27-May-2020].
- [16] Energy/star, “Supermarkets: An Overview of Energy Use and Energy Efficiency Opportunities Energy Use in Supermarkets How to Talk to Supermarket Owners and Operators About Energy Efficiency.”
- [17] Electric/Coice, “How Much Electricity Do Homes in Your State Use?,” 2017. [Online]. Available: <https://www.electricchoice.com/blog/electricity-on-average-do-homes/>. [Accessed: 27-May-2020].
- [18] P. M. Costa and M. A. Matos, “Assessing the contribution of microgrids to the

- reliability of distribution networks,” *Electr. Power Syst. Res.*, 2009.
- [19] “German power grid firms cut renewable energy surcharge for 2019 - Reuters,” 2019. [Online]. Available: <https://www.reuters.com/article/germany-renewables-fee/german-power-grid-firms-cut-renewable-energy-surcharge-for-2019-idUSL8N1WV1HI>. [Accessed: 28-May-2020].
- [20] S. T. Cady, S. Member, A. D. Domínguez-garcía, C. N. Hadjicostis, and S. Member, “A Distributed Frequency Regulation Architecture for Islanded Inertialess AC Microgrids,” vol. 23, no. 5, pp. 1717–1735, 2015.
- [21] S. T. Cady, S. Member, A. D. Domínguez-garcía, C. N. Hadjicostis, and S. Member, “Finite-time approximate consensus and its application to distributed frequency regulation in islanded AC microgrids,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2015, vol. 23, no. 5, p. 7.
- [22] “SDGs .. Sustainable Development Knowledge Platform,” 2019. [Online]. Available: <https://sustainabledevelopment.un.org/sdgs>. [Accessed: 21-May-2020].
- [23] “Goal 7 .. Sustainable Development Knowledge Platform,” 2018. [Online]. Available: <https://sustainabledevelopment.un.org/sdg7>. [Accessed: 21-May-2020].
- [24] N. Routley, “Mapped: The 1.2 Billion People Without Access to Electricity - Visual Capitalist,” 2019. [Online]. Available: <https://www.visualcapitalist.com/mapped-billion-people-without-access-to-electricity/>. [Accessed: 21-May-2020].
- [25] “Goal 9 .. Sustainable Development Knowledge Platform,” 2017. [Online]. Available: <https://sustainabledevelopment.un.org/sdg9>. [Accessed: 21-May-2020].
- [26] “Goal 11 .. Sustainable Development Knowledge Platform,” 2018. [Online]. Available: <https://sustainabledevelopment.un.org/sdg11>. [Accessed: 21-May-2020].



## ANNEX I. SUSTAINABLE DEVELOPMENT GOALS

The SDG's or Sustainable Development Goals are a set of 17 premises that the UN has created in order to ensure a more sustainable future for all. They are based on the premise of pursuing an inclusive, clean, economic and advanced future throughout development. There are three core dimensions that have to rule the process of thinking while developing the project. The first one is the economic growth that affects both society and the entity. The second one is the Social inclusion, allowing for everyone to contribute to development. The third aspect is the environmental protection needs to be provided in order to produce the least harm. The main concern that the UN is the inability to look at the three of them as individual goals. They have a common are in the which sustainable development is allocated. These three core elements must be driven by two critical dimensions, partnership and peace. Partnership is based on the opportunity to strengthen the cooperation in between entities. On the other side, peace, justice and strong institutions are a key to ensure that the three core elements of development work properly.



*Figure 34: Goal-7 [22]*



*Figure 35: Goal-9 [22]*



*Figure 36: Goal-11[22]*

The goals that are more related to this project are those connected with accessible renewable generation. The ones that have been selected are Goals 7, 9 and 11.

- **Goal 7:** The aim of this goal is to grant access to an affordable, reliable, sustainable and modern electricity in poorer countries [23]. The way we can relate this goal to the project is by offering a viable alternative to big electric networks. Figure 37 shows the scarce access to electricity in central Africa. The main issue is the inability to set a grand-scale electric networks, since the average time to do so is around 25 years [24]. The solution is the implementation of Mini-Grids, which are based on solar powered generation in order to supply isolated cities and villages. The main concern with this model is reliability, which is where our project comes into place. A distribute architecture is more reliable and less vulnerable for single point failure, which makes it ideal for this use. The implementation of distributed mini grids is the perfect alternative to deliver electricity to the underdeveloped world.

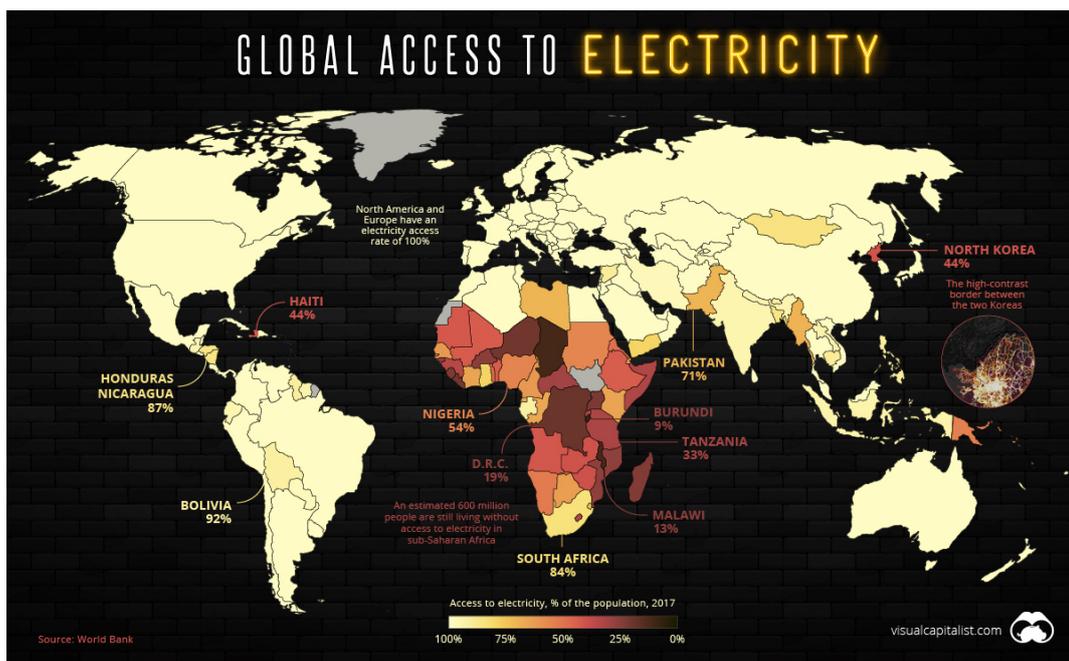


Figure 37: Global Access to Electricity [24]

- Goal 9: The aim of this goal is to build a resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation [25]. This concept is closely related to the idea that we need to spend our countries' money in the best way possible without wasting any. Therefore, the optimal approach to the implementation of electric networks is the one that will ensure the least failure with the longest life. We can relate this goal with the project by stating that one of the objectives cited in {4.2} is to create a more reliable ratio consensus algorithm. Also, in {1.4} we stated how distributed microgrids offer a more reliable operation due to the lack of one-point vulnerability due to the multiplicity of controllers.
- Goal 11: The aim of this goal is to make cities safer, more inclusive, resilient and sustainable [26]. This is related to Goal 7 and the accessibility to electricity, since the more people living in slums, the more people without electricity. This will result in less opportunities to achieve a better, more modern life. As a result, the economical gap will enlarge hence making it worse.



## ANNEX II. DISTRIBUTED VOLTAGE CONTROL

### CLASSES

Figure 38 shows the class diagrams of OLocalVertex and ORemoteVertex.

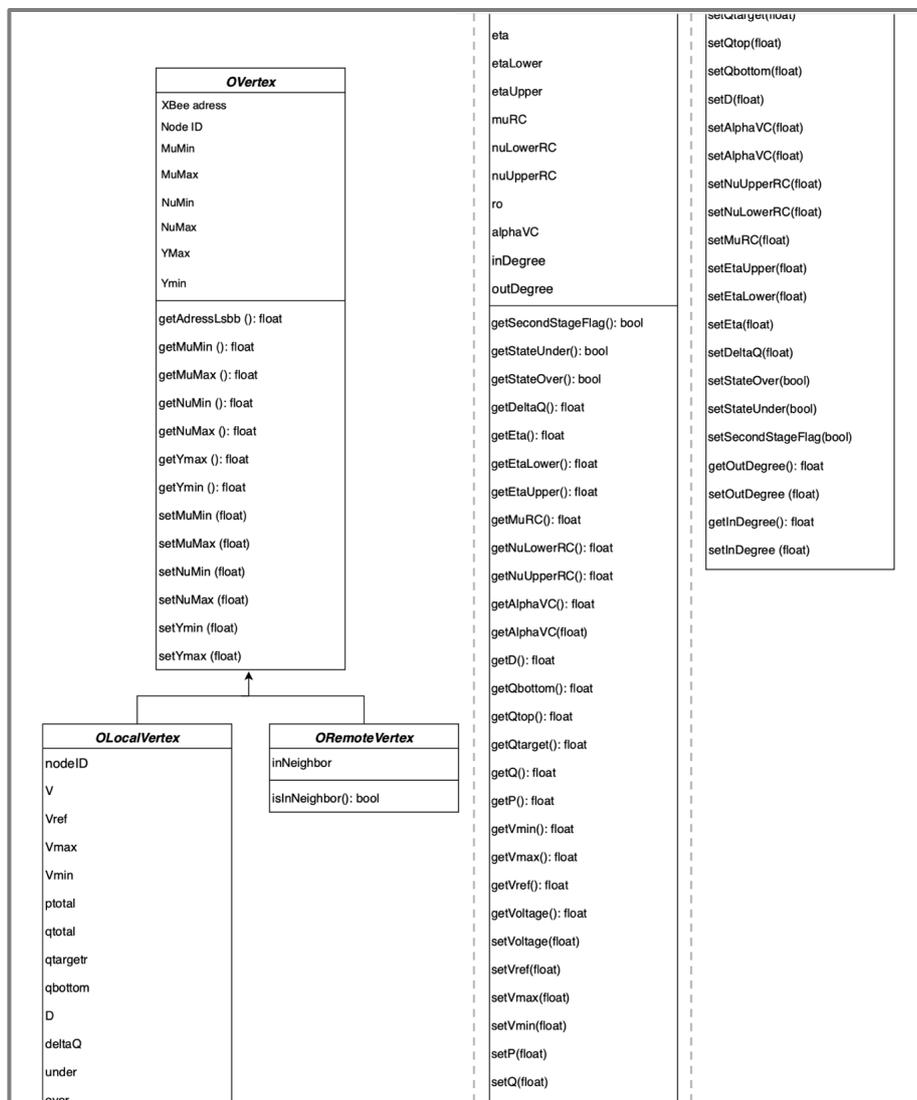


Figure 38: Class Diagram

## ***FUNCTIONS***

- Voltage Control Body

```
float OAgent::voltageControl_dist( float diffV, float Vref, float secPercentage,
float p, float q, float qtop, float qbottom, float Sij , float alphaVC, uint8_t
iterations, uint16_t period, uint8_t diameter, float eps {

    OLocalVertex * s = _G->getLocalVertex();
    _initializeVoltageControl(s,diffV,Vref,secPercentage,p,q,qtop,qbottom,Sij,
alphaVC );

    isUnderVoltage(s);
    isOverVoltage(s);
    firstStageControl(s);
    shareFlag( s, (diameter + 6), period );

    if(s->getSecondStageFlag())
    {
        secondStageControl( s, iterations, period, diameter, eps );
        if((s->getQtarget()+s->getEta()) > s->getQtop())
        {
            s->setQ(s->getQtop());
        }else if((s->getQtarget()+s->getEta()) < s->getQbottom())
        {
            s->setQ(s->getQbottom());
        }else
        {
            s->setQ( s->getQtarget() + s->getEta())
        }
    }else{
        if(s->getQtarget() > s->getQtop())
        {
            s->setQ(s->getQtop());

        }else if(s->getQtarget() < s->getQbottom())
        {
            s->setQ(s->getQbottom());

        }else
        {
            s->setQ( s->getQtarget() );
        }
    }
    return ( s->getQ());
}
```

- Voltage Control Initialization

```
void OAgent::_initializeVoltageControl( OLocalVertex * s, float diffV, float
Vref, float secPercentage, float p, float q, float qtop, float qbottom, float
Sij, float alphaVC )
{
    _G->clearAllStates();
    s->setVoltage(Vref + diffV);
    s->setVref(Vref);
    s->setVmax(1.05);
    s->setVmin(0.95);
    s->setP(p);
    s->setQ(q);
    s->setQtop(qtop);
    s->setQbottom(qbottom);
    s->setD(Sij);
    s->setAlphaVC(alphaVC);
    s->setSecondStageFlag(false);
}
```

- Over/Under Voltage Check

```
void OAgent::isOverVoltage(OLocalVertex * s)
{
    if(s->getVoltage() > s->getVmax())
    {
        uint8_t ID = s->getID();
        s->setStateOver(true);
    }
    if(s->getVoltage() <= s->getVmax())
    {
        uint8_t ID = s->getID();
        s->setStateOver(false);
    }
}

void OAgent::isUnderVoltage(OLocalVertex * s)
{
    if(s->getVoltage() < s->getVmin())
    {
        uint8_t ID = s->getID();
        s->setStateUnder(true);
    }
    if(s->getVoltage() >= s->getVmin())
    {
        uint8_t ID = s->getID();
        s->setStateUnder(false);
    }
}
```

- First Stage

```

void OAgent::firstStageControl( OLocalVertex * s )
{
    Serial<<"*****1st Stage*****"<<endl;

    if(s->getStateOver())
    {
        Serial<<"1st stage---Over"<<endl;
        s->setRo( (s->getAlphaVC() / s->getD()) *(s->getVmax() - s-
>getVoltage()));
        s->setQtarget(s->getQ() + s->getRo());
        Serial<<"ΔQ in 1st stage is: ";
        Serial.println(s->getRo());
        Serial<<"Q_target for node "<<s->getID()<<" after 1st stage is: "<<s-
>getQtarget()<<endl;

        if(s->getQtarget() < s->getQbottom())
        {
            s->setSecondStageFlag(true);
        }else if(s->getQtarget() > s->getQtop())
        {
            s->setSecondStageFlag(true);
        }else{
            s->setSecondStageFlag(false);
        }
    } else if(s->getStateUnder())
    {
        Serial<<"1st stage---Under"<<endl;
        s->setRo( (s->getAlphaVC() / s->getD()) *( s->getVmin() - s->getVoltage()
));
        s->setQtarget(s->getQ()+s->getRo());
        Serial<<"ΔQ in 1st stage is: "<<s->getRo()<<endl;
        Serial<<"Q_target for node "<<s->getID()<<" after 1st stage is: "<<s-
>getQtarget()<<endl;

        if(s->getQtarget() > s->getQtop())
        {
            s->setSecondStageFlag(true);
        } else if(s->getQtarget() < s->getQbottom())
        {
            s->setSecondStageFlag(true);
        }else{
            s->setSecondStageFlag(false);
        }
    }else
    {
        s->setSecondStageFlag(false);
        s->setRo( float(0));
    }
}

```

```

s->setDeltaQ(float (0));
s->setQtarget(s->getQ());
Serial<<"Q_target for node "<<s->getID()<<" after 1st stage is: "<<s-
>getQtarget()<<endl;
}
}

```

- **Broadcast Second Stage Flag**

```

void OAgent::shareFlag( OLocalVertex * s, uint8_t iterations, uint16_t period)
{
    float Dout = float(s->getOutDegree() + 1);
    _initializeFairSplitting_RSL(s,0,0,0);
    unsigned long start;
    bool txDone;
    uint16_t txTime;
    int iter=0;
    int node_check[NUM_REMOTE_VERTICES];
    uint32_t aLsb;
    for(int i=0; i < NUM_REMOTE_VERTICES; i++)
    {
        node_check[i] = 0;
    }
    int frame = 30;
    do
    {
        srand(analogRead(0));
        txTime = (rand() % (period - 2*frame)) + frame;
        txDone = false;
        start = millis();
        uint8_t i;
        while(uint16_t(millis()-start) < period)
        {
            if(_fairSplitPacketAvailable())
            {
                aLsb = _rx->getRemoteAddress64().getLsb();
                if(_G->isInNeighbor(aLsb,i))
                {
                    if(getSecondStageFlagfromPackage(s))
                    {
                        s->setSecondStageFlag(true);
                    }
                    uint8_t neighborID = _getNeighborIDFromPacket();
                    uint8_t nodeID = s->getID();

                    node_check[neighborID -1] = 1;
                }
            }
        }
    }
}

```

```

        if((int((millis() - start)) >= txTime) && !txDone) {
            txDone = true;
            _broadcastFairSplitPacket_RSL(s);
        }
    }
    if(!_quiet) {
        delay(10);
    } else {
        delay(25);
    }
    for(int j=0;j < NUM_REMOTE_VERTICES; j++)
    {
        if(node_check[j] == 0 && node_counter[j] >= 0)
            node_counter[j] += 1;
        else if(node_check[j] == 1 )
            node_counter[j] = 0;

        if(node_counter[j] >= int(iterations/2) )
        {
            s->setStatus(j+1, 1);
            s->decrementInDegree();
            uint8_t dout = s->getOutDegree();
            s->setOutDegree(dout - 1);
            node_counter[j] = -1;
        }
        node_check[j] = 0;
    }
    iter++;
}while(iter < iterations);
_buffer[1] = s->getOutDegree();
_buffer[2] = _G->getN() - 1;
}

```

- Second Stage Variable Initialization

```

void OAgent::_initializeVariablesSecStage(OLocalVertex * s)
{
    if((s->getQtarget()) > (s->getQtop()))
    {
        s->setMuRC( s->getQtarget() - s->getQtop());
        s->setNuUpperRC(float(0));
        s->setNuLowerRC(s->getQbottom() - s->getQtarget());
    }else if ((s->getQtarget()) < (s->getQbottom()))
    {
        s->setMuRC( s->getQtarget() - s->getQbottom());
        s->setNuUpperRC(s->getQtop() - s->getQtarget());
        s->setNuLowerRC(float(0));
    }else
    {
        s->setMuRC(float(0));
    }
}

```

```
s->setNuUpperRC(s->getQtop()-s->getQtarget());  
s->setNuLowerRC(s->getQbottom()-s->getQtarget());  
}  
tNuLowerRC(-s->getQsecondary());  
}
```

- Second Stage

```
void OAgent::secondStageControl( OLocalVertex * s, uint8_t iterations, uint16_t  
period, uint8_t diameter, float eps)  
{  
    Serial<<"*****2nd Stage*****"<<endl;  
    _initializeVariablesSecStage(s);  
    s->setEtaLower(fairSplitRatioConsensus_RSL( s->getMuRC(), s-  
>getNuLowerRC(), iterations, period, diameter, eps));  
    s->setEtaUpper(fairSplitRatioConsensus_RSL( s->getMuRC(), s-  
>getNuUpperRC(), iterations, period, diameter, eps));  
  
    if( s->getMuRC() < 0 )  
    {  
        s->setEta(s->getEtaLower());  
    }else if( s->getMuRC() >= 0 )  
    {  
        s->setEta(s->getEtaUpper());  
    }  
}
```



## ANNEX III. CENTRALIZED VOLTAGE CONTROL

- First Stage

```
for(i=0; i<3; i++)
{
    if(V[i] > Vmax)
    {
        ro[i] = alphaVC/Sij[i]*(Vmax - V[i]);

        if((Q[i]+ ro[i]) < qbottom)
        {
            Q_secondary=Q_secondary + ( Q[i]+ ro[i] - qbottom );
            ro[i] = ro[i] - Q_secondary;
        }

        if((Q[i]+ ro[i]) > qtop)
        {
            Q_secondary=Q_secondary + ( Q[i]+ ro[i] - qtop );
            ro[i] = ro[i] - Q_secondary;
        }
    }
    }else if(V[i] < Vmin)
    {
        ro[i] = alphaVC/Sij[i]*(Vmin - V[i]);

        if((Q[i]+ ro[i]) < qbottom)
        {
            Q_secondary=Q_secondary + ( Q[i]+ ro[i] - qbottom );
            ro[i] = ro[i] - Q_secondary;
        }

        if((Q[i]+ ro[i]) > qtop)
        {
            Q_secondary=Q_secondary + ( Q[i]+ ro[i] - qtop );
            ro[i] = ro[i] - Q_secondary;
        }
    }
    }else
    {
        ro[i] = 0;
    }
}
```

- Reserve of Reactive Power Computation

```
float QreserveRise[3] ;
float QreserveLower[3] ;
float QreserveUp = 0;
float QreserveDown = 0;

for(i=0;i<3;i++)
{
    QreserveRise[i]= qtop - (Q[i] + ro[i]);
    QreserveLower[i]= qbottom - (Q[i] + ro[i]);

    QreserveUp = QreserveUp + QreserveRise[i];
    QreserveDown = QreserveDown + QreserveLower[i];
}
```

- Second Stage

```
if (Q_secondary != 0)//second stage
{
    if(Q_secondary > 0)
    {
        for(i=0;i<3;i++)
        {
            deltaQsec[i] = Q_secondary * (QreserveRise[i]/QreserveUp);
        }
    }else
    {
        for(i=0;i<3;i++)
        {
            deltaQsec[i] = Q_secondary * (QreserveLower[i]/QreserveDown);
        }
    }
}
```

- New Reactive Power Setpoints

```
for(i=0;i<3;i++)
{
    Qnew[i] = Q[i] + ro[i] + deltaQsec[i];
    if(Qnew[i] > qtop)
    {
        Qnew[i]=qtop;
    }else if(Qnew[i] < qbottom)
    {
        Qnew[i]=qbottom;
    }
}
```

## ANNEX IV. ELECTRIC BILL EVOLUTION

Table 11 show the evolution of the monthly electric bill for the economic analysis in {7.2} according to the increased estimation due to green initiatives.

*Table 11: Electric Bill Evolution for Microgrid Scenario*

Year 1 -- Year 3			
Electric bill per month			
Sector	\$/kWh	kWh	\$
Commercial	0.0486	208333	10125
Industrial	0.01668	317000	5287.56
Residential	0.12528	755000	94586.4

Year 4 -- Year6			
Electric bill per month			
Sector	\$/kWh	kWh	\$
Commercial	0.0567	208333	11812.5
Industrial	0.01946	317000	6168.82
Residential	0.14616	755000	110350.8

Year 7 -- Year 9			
Electric bill per month			
Sector	\$/kWh	kWh	\$
Commercial	0.0648	208333	13500
Industrial	0.02224	317000	7050.08
Residential	0.16704	755000	126115.2

Year 10 -- Year 12			
Electric bill per month			
Sector	\$/kWh	kWh	\$
Commercial	0.0729	208333	15187.5
Industrial	0.02502	317000	7931.34
Residential	0.18792	755000	141879.6



## ANNEX V. RATIO CONSENSUS CODE

- Initialization

```
float OAgent::fairSplitRatioConsensus_RSL(float y, float z, uint8_t iterations,
uint16_t period, uint8_t diameter, float eps) {
    srand(analogRead(7));
    float gamma = 0;

    if(isLeader())
    {
        gamma = leaderFairSplitRatioConsensus_RSL(y, z,
iterations,period,diameter,eps);
    }
    else
    {
        gamma = nonleaderFairSplitRatioConsensus_RSL(y, z,
iterations,period,diameter,eps);
    }
    return gamma;
}
```

- Leader Function

```
float OAgent::leaderFairSplitRatioConsensus_RSL(float y, float z, uint8_t
iterations, uint16_t period, uint8_t diameter, float eps) {
    unsigned long t0 = myMillis();
    unsigned long startTime = t0 + RC_DELAY;
    OLocalVertex * s = _G->getLocalVertex();
    float gamma = 0;
    bool scheduled
= _waitForChildSchedulePacketRC(SCHEDULE_FAIR_SPLIT_HEADER,SCHEDULE_TIMEOUT,
startTime, iterations, period);

    if (!scheduled)
    {
        Serial<<"RC scheduling was a FAIL!"<<endl;
        delay(5);
        gamma = -1;
    }
    else
    {
        Serial<<"RC scheduling was a SUCCESS!"<<endl;
        delay(5);
        if(_waitToStart(startTime,true,10000))
        {
            delay(5);
        }
    }
}
```

```

        gamma = ratiomaxminConsensus(y, z, iterations,period,diameter,eps);
    }
}
return gamma;
}

```

- Non-Leader Function

```

float OAgent::nonleaderFairSplitRatioConsensus_RSL(float y, float z, uint8_t
iterations, uint16_t period, uint8_t diameter, float eps) {
    unsigned long startTime = 0;
    float gamma = 0;
    bool scheduled = _waitForParentSchedulePacketRC(startTime,iterations,period,-
1);

    if(scheduled)
    {
        Serial<<"RC scheduling was a SUCCESS!"<<endl;
        delay(5);
        if(_waitToStart(startTime,true,10000)) {
            delay(5);
            gamma = ratiomaxminConsensus(y, z, iterations,period,diameter,eps);
        }
    }
    else
    {
        Serial<<"RC scheduling was a FAIL!"<<endl;
        delay(5);
        gamma = -1;
    }
    return gamma;
}

```

- Body

```

float OAgent::ratiomaxminConsensus(float y, float z, uint8_t iterations,
uint16_t period, uint8_t diameter, float eps)
{
    OLocalVertex * s = _G->getLocalVertex();
    float Dout = float(s->getOutDegree() + 1);
    _initializeFairSplitting_RSL(s,y,z,eps);
    unsigned long start;
    bool txDone;
    bool mucheck = 0;
    bool sigmacheck = 0;
    uint16_t txTime;
    float inY;
    float inZ;
    float endY;
    float endZ;
    int count = 3;
    int iter=0;
}

```

```

int node_check[NUM_REMOTE_VERTICES];
uint32_t aLsb;

for(int i=0; i < NUM_REMOTE_VERTICES; i++)
{
    node_check[i] = 0;
}
int frame = 30;

do
{
    srand(analogRead(0));
    txTime = (rand() % (period - 2*frame)) + frame;
    txDone = false;
    start = millis();
    inY = 0;
    inZ = 0;
    delay(5);
    endY=s->getYMin();
    endZ=s->getZ();
    uint8_t i;

    while(uint16_t(millis()-start) < period)
    {
        if(_fairSplitPacketAvailable())
        {
            aLsb = _rx->getRemoteAddress64().getLsb();
            if(_G->isInNeighbor(aLsb,i))
            {
                float inMu = _getMuFromPacket();
                float inSigma = _getSigmaFromPacket();
                uint8_t neighborID = _getNeighborIDFromPacket();

                uint8_t nodeID = s->getID();

                float Mudiff = inMu - s->getNuMin(i);
                float sigdiff = inSigma - s->getTau(i);
                inY += Mudiff;
                s->setNuMin(i,inMu);
                inZ += sigdiff;
                s->setTau(i,inSigma);
                node_check[neighborID -1] = 1;
            }
        }
        if((int)(millis() - start) >= txTime) && !txDone) {
            txDone = true;
            _broadcastFairSplitPacket_RSL(s);
        }
    }
}
if(!_quiet) {

```

```

        delay(10);
    } else {
        delay(25);
    }
    s->setYMin((s->getYMin()/Dout) + inY);
    s->setMuMin(s->getMuMin() + (s->getYMin()/Dout));
    s->setZ((s->getZ()/Dout) + inZ);
    s->addToSigma(s->getZ()/Dout);
    s->setGammaRSL(s->getYMin()/s->getZ());

    _buffer[count] = (s->getYMin())/s->getZ();
    _bufferY[count] = s->getYMin();
    _bufferZ[count] = s->getZ();
    count++;
    for(int j=0;j < NUM_REMOTE_VERTICES; j++)
    {
        if(node_check[j] == 0 && node_counter[j] >= 0)
            node_counter[j] += 1;
        else if(node_check[j] == 1)
            node_counter[j] = 0;
        if(node_counter[j] >= int(iterations/2) )
        {
            s->setStatus(j+1, 1);
            s->decrementInDegree();
            uint8_t dout = s->getOutDegree();
            s->setOutDegree(dout - 1);
            node_counter[j] = -1;
        }
        node_check[j] = 0;
    }
    Serial<<"gamma: ";
    Serial.println(s->getGammaRSL(),6);

    if(((iter % 3) == 0) && (iter != 0) && (iter != 3))
        maxMinConsensus_RSL(s,eps,diameter,period);
        iter++;
    }while((s->getFlagMaxMin() == false ));

    if(s->getZ() != 0)
        _buffer[0] = (s->getYMin()/s->getZ());

    _buffer[1] = s->getOutDegree();
    _buffer[2] = _G->getN() - 1;

    s->setMuRC(s->getMuMin());
    return (s->getGammaRSL());
}

```

## ANNEX VI. MAXMIN CONSENSUS CODE

- Initialization

```
void OAgent::maxMinConsensus_RSL(OLocalVertex * s, float Epsilon, uint8_t
diameter, uint16_t period)
{
    srand(analogRead(7));

    if(isLeader())
    {
        leaderMaxMinConsensus_RSL(s,Epsilon,diameter,period);
    }else
    {
        nonleaderMaxMinConsensus_RSL(s,Epsilon,diameter,period);
    }
}
```

- Leader Function

```
void OAgent::leaderMaxMinConsensus_RSL(OLocalVertex * s, float Epsilon, uint8_t
diameter, uint16_t period)
{
    unsigned long t0 = myMillis();
    unsigned long startTime = t0 + MC_DELAY;
    bool scheduled
=waitForChildSchedulePacketMaxMin(SCHEDULE_MAXMIN_HEADER,SCHEDULE_TIMEOUT,
startTime, 20, period);

    if (!scheduled)
    {
        Serial << "maxmin scheduling was a FAIL"<<endl;
        delay(5);
    }
    else
    {
        Serial << "maxmin scheduling was a SUCCESS!"<<endl;
        delay(5);
        if(_waitToStart(startTime,true,10000))
            maxMin_RSL(s,Epsilon,diameter,period);
    }
}
```

- Non-Leader Function

```
void OAgent::nonleaderMaxMinConsensus_RSL(OLocalVertex * s, float Epsilon,
uint8_t diameter, uint16_t period)
{
```

```

unsigned long startTime = 0;
uint8_t iterations =20;
bool scheduled =
_waitForParentSchedulePacketMaxMin(startTime,iterations,period,-1);

if(scheduled)
{
    Serial << "maxmin scheduling was a SUCCESS!"<<endl;
    delay(5);
    if(_waitToStart(startTime,true,10000))
        maxMin_RSL(s,Epsilon,diameter,period);
}
else
{
    Serial << "maxmin scheduling was a FAIL"<<endl;
    delay(5);
}
}

```

- Broadcast Information

- *Sending*

```

void OAgent::_broadcastMaxMinPacket(long max, long min) {
    uint16_t payload[5];
    payload[0] = MAXMIN_HEADER;
    payload[1] = max;
    payload[2] = max >> 16;
    payload[3] = min;
    payload[4] = min >> 16;

    _zbTx = ZBTxRequest(_broadcastAddress, ((uint8_t * )(&payload)),
sizeof(payload));
    unsigned long txTime = _xbee->sendTwo(_zbTx,false,true);
#ifdef VERBOSE
    Serial << _MEM(PSTR("Transmit time: ")) << txTime << endl;
#endif
}

```

- *Receiving*

```

float OAgent::_getMaxFromPacket_RSL() {
    uint8_t ptr = 2;
    long Max = _getUint32_tFromPacket(ptr);
    float max = float (Max);
    return max/BASE;
}

float OAgent::_getMinFromPacket_RSL() {
    uint8_t ptr = 6;
    long Min = _getUint32_tFromPacket(ptr);
}

```

```
float min = float (Min);
return min/BASE;
}
```

- **Body**

```
void OAgent::maxMin_RSL(OLocalVertex * s, float Epsilon, uint8_t diameter,
uint16_t period)
{
    float diff = 0;
    s->setFlagMaxMin(false);
    unsigned long start;
    bool txDone;
    uint16_t txTime;
    int count = 3;
    uint32_t aLsb;
    int frame = 20;
    s->setGammaMax(s->getGammaRSL());
    s->setGammaMin(s->getGammaRSL());

    for(uint8_t k = 0; k < (diameter+3); k++) {
        srand(analogRead(0));
        txTime = (rand() % (period - 2*frame)) + frame;
        txDone = false;
        start = millis();
        uint8_t i;

        while(uint16_t(millis()-start) < period) {
            if(_maxminPacketAvailable_RSL()) {

                aLsb = _rx->getRemoteAddress64().getLsb();
                if(_G->isInNeighbor(aLsb,i))
                {
                    float inMax = _getMaxFromPacket_RSL();
                    float inMin = _getMinFromPacket_RSL();
                    uint8_t neighborID = _getNeighborIDFromPacket();
                    if(inMax > s->getGammaMax())
                        s->setGammaMax(inMax);
                    if(inMin < s->getGammaMax())
                        s->setGammaMin(inMin);
                }
            }
            if((int((millis() - start)) >= txTime) && !txDone) {
                txDone = true;
                _broadcastMaxMinPacket_RSL(s);
            }
        }
        diff = (s->getGammaMax()) - (s->getGammaMin());
        Serial<<"difference: ";
        Serial.println(diff,6);
    }
}
```

```
if(diff <= Epsilon)
{
    s->setFlagMaxMin(true);
}else
{
    s->setFlagMaxMin(false);
}
}
```