



ESCUELA TÉCNICA SUPERIOR DE INGENIERIA (ICAICA)

GRADO EN INGENIERÍA ELECTROMECÁNICA

PV Agent

Desing and implementation in Labview

Autor: Jacobo Cerdeiras Megias

Director: Lorenzo Reyes

Asistant: Enrica Scolari

Madrid

Julio 2015



AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN ACCESO ABIERTO (RESTRINGIDO) DE DOCUMENTACIÓN

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Jacobo Cerdeiras Megias, como alumno de la UNIVERSIDAD PONTIFICIA COMILLAS (COMILLAS), **DECLARA**

que es el titular de los derechos de propiedad intelectual, objeto de la presente cesión, en relación con la obra Implementación del PV Agent en Labview¹, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual como titular único o cotitular de la obra.

En caso de ser cotitular, el autor (firmante) declara asimismo que cuenta con el consentimiento de los restantes titulares para hacer la presente cesión. En caso de previa cesión a terceros de derechos de explotación de la obra, el autor declara que tiene la oportuna autorización de dichos titulares de derechos a los fines de esta cesión o bien que retiene la facultad de ceder estos derechos en la forma prevista en la presente cesión y así lo acredita.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad y hacer posible su utilización de *forma libre y gratuita (con las limitaciones que más adelante se detallan)* por todos los usuarios del repositorio y del portal e-ciencia, el autor **CEDE** a la Universidad Pontificia Comillas de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra (a) del apartado siguiente.

3º. Condiciones de la cesión.

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia, el repositorio institucional podrá:

(a) Transformarla para adaptarla a cualquier tecnología susceptible de incorporarla a internet; realizar adaptaciones para hacer posible la utilización de la obra en formatos electrónicos, así

¹ Especificar si es una tesis doctoral, proyecto fin de carrera, proyecto fin de Máster o cualquier otro trabajo que deba ser objeto de evaluación académica



como incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.

(b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato. .

(c) Comunicarla y ponerla a disposición del público a través de un archivo abierto institucional, accesible de modo libre y gratuito a través de internet.²

(d) Distribuir copias electrónicas de la obra a los usuarios en un soporte digital.³

4º. Derechos del autor.

El autor, en tanto que titular de una obra que cede con carácter no exclusivo a la Universidad por medio de su registro en el Repositorio Institucional tiene derecho a:

a) A que la Universidad identifique claramente su nombre como el autor o propietario de los derechos del documento.

b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

c) Solicitar la retirada de la obra del repositorio por causa justificada. A tal fin deberá ponerse en contacto con el vicerrector/a de investigación (curiarte@rec.upcomillas.es).

d) Autorizar expresamente a COMILLAS para, en su caso, realizar los trámites necesarios para la obtención del ISBN.

d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

² En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría redactado en los siguientes términos:

(c) Comunicarla y ponerla a disposición del público a través de un archivo institucional, accesible de modo restringido, en los términos previstos en el Reglamento del Repositorio Institucional

³ En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría eliminado.



5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- a) Deberes del repositorio Institucional:
 - La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
 - La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
 - La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- b) Derechos que se reserva el Repositorio institucional respecto de las obras en él registradas:



UNIVERSIDAD PONTIFICIA COMILLAS

INGENIERO ELECTROMECÁNICO

- retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a de de

ACEPTA

Fdo.....



UNIVERSIDAD PONTIFICIA COMILLAS

INGENIERO ELECTROMECÁNICO

Proyecto realizado por el alumno/a:

Jacobo Cerdeiras

Fdo.: Fecha://

Autorizada la entrega del proyecto cuya información no es de carácter confidencial

EL DIRECTOR DEL PROYECTO

Lorenzo Reyes

Fdo.: Fecha://

Vº Bº del Coordinador de Proyectos

Fernando de Cuadra

Fdo.: Fecha://



UNIVERSIDAD PONTIFICIA COMILLAS

INGENIERO ELECTROMECÁNICO



ESCUELA TÉCNICA SUPERIOR DE INGENIERIA (ICAI)

GRADO EN INGENIERÍA ELECTROMECÁNICA

PV Agent

Desing and implementation in Labview

Autor: Jacobo Cerdeiras Megias

Director: Lorenzo Reyes

Asistant: Enrica Scolari

Madrid

Julio 2015



RESUMEN

El objetivo del proyecto ha sido el diseño y la implantación del *PV Agent* (Agente de paneles solares) en *Labview*. El *PV Agent* forma parte de los softwares utilizados para el control de COMMELEC. Por tanto este proyecto es solo una parte de un proyecto más grande llamado COMMELEC.

El objetivo de COMMELEC es desarrollar un control en tiempo real para manejar los distintos recursos de la red, ajustándose a las restricciones de esta. El tiempo de actuación del controlador debe ser inferior a 100ms. El objetivo principal es que permita la inclusión de generadores de energía renovable (en nuestro caso los paneles solares) de cara al presente y al futuro de la sociedad.

El *PV Agent* consta de distintas que se van a ir presentando en el orden secuencial en el que funcionarán. La primera parte es la toma de medidas en tiempo real para predecir los límites de actuación del convertidor en cuanto a potencia se refiere. Las medidas utilizadas son la radiación solar y las temperaturas del panel y ambiente, que se considerarán constantes de cara al siguiente instante.

Utilizando un código desarrollado en *Matlab* y que ha sido implementado en *Labview* somos capaces de predecir un intervalo de la radiación solar (W/m^2) para el siguiente instante. Con esta predicción y un modelo de los paneles solar más convertidor transformamos el intervalo de radiación en uno de potencia (W). El convertidor ha sido modelado como una eficiencia en función del punto de funcionamiento.

Una vez tenemos el intervalo lo pasamos por una función de transferencia para ver la respuesta temporal a un escalón (un cambio en el set point) y poder determinar en qué punto de funcionamiento se encontrara en el siguiente instante.

Una vez tenemos los márgenes hay que leer la comunicación con el *Grid Agent*. El *Grid Agent* es un agente intermedio (superior) que es el que se encarga de analizar la carga de la red y ajustar la producción de los distintos generadores. El *Grid Agent* también puede ser dependiente de un órgano más grande. La comunicación con el *Grid Agent* consiste en la lectura del set point requerido y la transmisión de los futuros márgenes en los que va poder trabajar el convertidor. El set point requerido ha de ser proyectado dentro de los límites actuales de funcionamiento. Estos son en el plano PQ: trabajar en la parte P positiva, con una S inferior o igual al límite impuesto por el convertidor y con una P inferior o igual a la máxima que permita la radiación en ese instante. Por último existe el límite de mínimo coseno (ϕ) admisible. A esta área se le llama *PQt Profile*.

Con el set point dentro del *PQt* se envía el nuevo punto de funcionamiento deseado al convertidor y se prepara y calcula el nuevo aviso que se debe mandar al *Grid Agent*. Este mensaje contiene el



punto de funcionamiento, el *PQt Profile*, la *Belief Function* y el *Virtual Cost*. La *Belief Function* es el conjunto de puntos (intervalo de P) en el cual podría operar el convertidor en caso de una disminución de la radiación solar en el siguiente instante. Este margen es el intervalo que nos proporciona el modelo del sistema sin la función de transferencia. Por último el *Virtual Cost* es el interés o lo proclive que es el sistema a encontrarse en determinado punto de operación (P, Q). En nuestro caso se intenta maximizar la P y focalizar la Q para que sea próxima a 0.

Con la visión general de cómo debe ser el *PV Agent*, se procede a diseñar y testar las distintas partes por separado para más tarde ponerlas en conjunto. Debido a una serie de limitaciones del convertidor con el cual se ha realizado el proyecto se han diseñado dos agentes. Uno como si el convertidor fuese incontrolable, en el cual pasamos a predecir directamente la futura Pmax a través de la P actual en vez de la radiación solar y donde no hay comunicación con el convertidor, y el agente normal que se ha explicado en líneas generales.

El agente de la carga incontrolable se ha implantado y testado en tiempo real únicamente faltando la comunicación con el *Grid Agent* que no estaba en funcionamiento. Del agente controlable se han implantado la mayoría del código en *Labview*, únicamente faltando el modelo de los paneles solares.

ABSTRACT

The goal of the project has been the design and implementation of the *PV Agent* in *Labview*. The *PV Agent* forms part of the softwares used in COMMELEC. So indeed the project is only a small part of a bigger one called COMMELEC, which is being developed at EPFL.

The goal of the COMMELEC is the development of a real time controller to manage the different renewable resources of the grid. This controller has to adjust itself to the restrictions of the grid. The *PV Agent* has to respond with an ultra-short term constraint (<100ms). The main objective is the inclusion of the solar panels resource for the present and future society.

The different parts of the *PV Agent* are going to be presented in the sequential order they take place. The first part is the acquisition of the irradiance and temperature measurements. This measures are used in order to predict the power limits where the converter will be able to work in the next time step. The temperatures are consider constant between the different time steps.

Using a *Matlab* code that has been traduced into *Labview*, an irradiance interval (W/m^2) is predicted for the next time step. With this interval and a model of the solar panels plus converter, the irradiance interval is traduced into power (W). The model of the converter is an efficiency depend on the operating point.



Once the limits of the interval are found, the time response of step in the converter set point is checked.

With the margins what the *PV Agent* has to do is read the communication with the *Grid Agent*. The *Grid Agent* is higher agent in charge to analyze the charge of the grid and command the production of the different generators. The *Grid Agent* can be dependent of bigger *Grid Agent*. The communication consist in reading the required power set point and the transmission of the future available power margins of the converter. The required set point has to be projected inside the actual limits. These limits are in the PQ plane: work in the $P > 0$, with an S equal or less of the converter limit and a P equal or less of what the actual irradiance can provide. At last there is a limit for minimum $\cos(\phi)$. The area limited by these constraints is called *PQt Profile*.

With the set point inside of the *PQt* the new set point request is send to the converter and the new advertisement to the *Grid Agenti* is send. This advertisement contains the working point, the *PQt Profile*, the *Belief Function* and the *Virtual Cost*. The *Belief Function* is the set of P points where the converter could operate in case of a decrease of the irradiance in the next time step. At last the *Virtual Cost* is the willing of the resource to be in determinate working point. In this case, the goal is to maximize the P and neither consume or generate Q.

With the general overview of how the *PV Agent* should work, it's been proceed to design and test the different parts separately. And afterwards to implement them all together. Due to several limitations in the converter, two converters have been design. One as if the resource was uncontrollable, where the actual future Pmax is predicted instead of the solar irradiance and there is no communication with converter. And a second one following the structure already explained.

The uncontrollable agent has been already implemented and is running in real time, simulating a request from the *Grid Agent*. While for the controllable agent most parts have been implemented and only the PV model is missing.



Contents

<i>Acknowledgment</i>	14
<i>Introduction</i>	15
1. <i>Data Management</i>	17
1.1 Data acquisition.....	17
1.2 Data analysis.....	18
2. <i>Converter Modeling</i>	22
2.1 Communication with converter	22
2.2 Experiment to characterize the dynamics.....	23
2.3 Efficiency of the converter	29
3. <i>Implementation of the agent in Labview</i>	30
3.1 Uncontrollable agent.....	31
3.2 Controllable PV Agent	32
Conclusion	36
Annex A	36
MAaverage.m.....	37
TFfor1order.m.....	38
TFfor2order.m.....	39
Test.m.....	40
Annex B	41
Communication and experimental codes	41
Codes for real time computation of data.....	41
Communication code	45
VIs of the Uncontrollable agent	48
Main for Uncontrolable agent rev3.vi.....	48
Initialization rev2.vi.....	50
Matrix initiation.vi.....	51



Matrix computation.vi.....	52
Agent process Uncontrolable agent.vi.....	54
Controllable Agent codes.....	54
Pmax Calculation.vi.....	54
TF(RT).vi.....	55
Projection 2.0.vi.....	55
_sencommand.vi.....	57
Agent Process for controllable agent.vi.....	59
PQ profile CA draw.vi.....	60



Table of Figures

Figure 1: PQt profile and belief function of the PV Agent	16
Figure 2: Diagram of the initial settlement	17
Figure 3: Example of a data sheet	18
Figure 4: Active Power introduce for 1h in the grid	19
Figure 5: Irradiance for the same time period as Fig. 4	19
Figure 6: Short time dynamics of the active power	20
Figure 7: General Characteristic of a PV Panel.....	20
Figure 8: a 100% step	24
Figure 9: example of moving average with different amount of points for the initialization. Lags of 20, 70, 120 and 200 points	24
Figure 10: the 1ºplot is with 2 M.A. of 70 points as lag. 2º one with 140. Then they are plot together.....	25
Figure 11: real response against the moving averages approximations.....	26
Figure 12: real response vs M.A. vs 1º order system	27
Figure 13:M.A. vs 2ºorder system.....	28
Figure 14: Converter efficiency computed with the recorded measurements	29
Figure 15: Efficiency of the converter from the vendor.....	30
Figure 16: Front panel of the 'Agent process Uncontrollable agent'	32
Figure 17: One diode model of PV array.....	33
Figure 18: Different PQt profiles are presented with the correspondent projection.....	35



Acknowledgment

I will like to thanks professor *Mario Paolone* for giving me the opportunity to develop this project that has cover most of the things I have learnt during the bachelor.

I will like to thanks *Lorenzo Reyes* for his help and explanations of *Labview* during the semester and to *Enrica Scolari* who had to work hand to hand with me during the semester.

I dedicate this Project as a finalization of my bachelor degree to my family that has always supported me, and especial to *Luis & Tota* that have always been in my mind and serve me as motivation during these four years.



Introduction

The project that is presented in this report forms part of a wider one that is being developed here at EPFL. This bigger project is called COMMELECⁱ. Its main goal is to create a control that manages in real time the renewable resources so they provide the energy following the restrictions of the grid. This new control method has to satisfy two main requirements of present and future electrical grids: (i) provide reliable and secure supply and (ii) introduce the rational power from renewable resources.

The main approaches so far have been (i) maintaining the power balance and (ii) keep voltages close to the rated valuesⁱⁱ.

In other words, the main goal is to define a scalable method for the direct and explicit control of real-time nodal power injections/absorptions. This is done through software agents, which are the responsible for subsystems and resources. To make it scalable, three features have been use:

- a. *Abstract framework*: it applies to all electrical subsystems and specifies their capabilities, internal state and expected behavior. A subsystem advertises all of this through *PQ profiles*, *virtual costs* and *belief functions*. This common abstract framework is fundamental for scalability.
- b. *Composition of Subsystems*: So a local grid with several resources can be seen as a single resource.
- c. *Separation concern*: Grid agents manipulate only data expressed by means of the abstract framework and don't need know the nature of the resource that is providing it.

There is one software for the grid agent, while for each resource agent in the grid is specific. They have to deal with the internal state of the resource and express it in the abstract framework. A second thing they have to deal with is the implementation of the received set points in the device.

In this general framework, the purpose of the project is the development and implementation of the PV Agent in the infrastructure of the COMMELEC. In detail the resource consist of a 20Kw PV panel plant connected to the grid through 2x10Kw converters. The panels are a 250W mono-crystalline PV modules from ECSolar Company. The two converters are from SolarMax and the model is 10MT. The Belief function in order to advertise the uncertainty of the solar resource, it's assume that the active power production might decrease from the requested set point with a predicted maximum variation.

$$P' = P - \Delta P_{pv}^{max};$$



The production of Q is restricted only by its relation with P and the constraints of the PQt profile.

$$Q' = \left\{ \begin{array}{l} \max \left\{ -P' \sqrt{\frac{1 - \cos_{\min}^2(\phi)}{\cos_{\min}(\phi)}}, Q \right\}, \text{ if } Q < 0, \\ \min \left\{ P' \sqrt{\frac{1 - \cos_{\min}^2(\phi)}{\cos_{\min}(\phi)}}, Q \right\}, \text{ if } Q \geq 0, \end{array} \right\}$$

The BF can be represented as a line that starts in $u=(P,Q)$ and ends in $u'=(P',Q')$. An example is presented in the next figure with the PQt profile⁴.

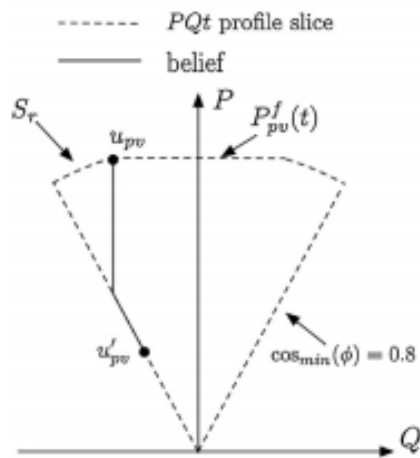


Figure 1: PQt profile and belief function of the PV Agent

The PQt profile it's define by the maximum available power at time t that can be maintained for a period T, the S_{\max} from the resource and the minimum $\cos(\varphi)$. The virtual cost seeks to maximize the P and minimize the Q.

The structure of the project will follow the same steps that have been used during the semester. The first thing will be to see what type of data it has to be deal with. These includes

⁴ From the "A composable method for real-time control of active distribution networks with explicit power set points. Part II: Implementation and validation, in Electric Power Systems Research" paper.



the acquisition and the analysis. The second step is to model the convertor and study how it works. And at last to implement everything in *Labview*.

1. Data Management

1.1 Data acquisition

The first thing needed was the improvement of the data acquisition. The measurements were computed and saved by using an FPGA and a LabView VI installed in a sBRIO located in the roof of the building. At first the voltage and current at the entrance (DC) and at the exit (AC) were computed and saved for a single PV module, together with the Irradiance (taken from a pyrometer) and three different temperatures, measured with specific sensors. The pyrometer is able to measure the global irradiance perpendicular to the panel. These were T1, the temperature inside the box, T2 and T3, the temperature in the back part of the panel. All these measurements were computed through a sBrio and saved in USB port. This way of saving the data was not efficient enough for our purpose. A diagram of the settlement is included.

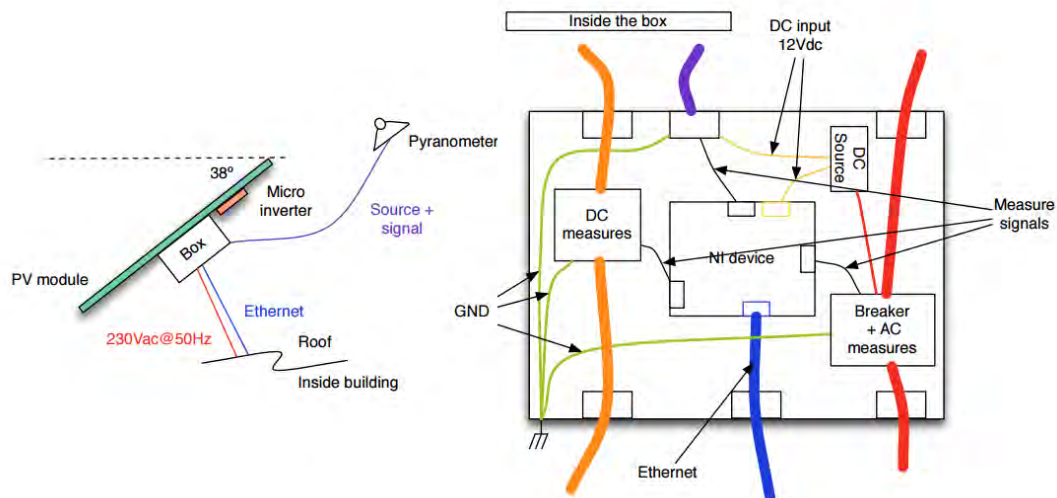


Figure 2: Diagram of the initial settlement

It has been taken the advantage of the ftp server installed in the sBRIO to save the TDMs files in the computer. The ftp server is linked to the IP address of the device and the measurements are saved through internet. Using this way it's been possible to access the measurements directly from the computer. A real time application was built to generate a TDMs file each hour and save all the measurements every 50 ms. This sBrio was connect just to one panel



which is a different kind from the others of the installation. Some problems were encountered with the panel and it was decided to set a new device for measuring the power, irradiance and different temperatures of our real installation.

This new device is a cRIO9068 and it's been installed a Legacy ftp server, since it doesn't comes from the fabric settings. Doing this the TDMs files are accessible in the same way thought is different type of device. The computation of the measurements it's done through a FPGA that takes the signals of three phase voltage plus the neutral (V₁, V₂, V₃, V_n) and the three currents plus the neutral plus the ground one (I₁, I₂, I₃, I_n, I_{pe}). The pyrometer and the temperature sensors were also moved and the values are still returned by the sbRIO in the roof but as shared variables. For the temperature sensors it's measured the T (°C) on the panel and the T (°C) of the ambient.

Here follows a brief explanation of the main code used to collect all the measurements and extract the main AC components needed by the PV agent. This code was previously developed by Lorenzo Reyes and only a few changes have been done to implement it in our case. From each signal taken by the FPGA the phase (°), amplitude and frequency (Hz) is computed. With these data we are able to compute the effective values of the currents (A) and voltages (V). To end up we compute the active power (W), reactive power (Var) and the power factor of each phase. Then power of each phase is sum up to get the active and reactive power that is been delivered to the grid. At last the total power with its power factor is computed. Each time the measures are saved with a time stamp so it's possible to know the actual time⁵. Figure 3 shows how the measurements are saved in the TDMs file is presented. The example it's just a data from 250ms period. Each file of one hour contains 72000 rows which is plenty of data. The data is presented in the next order: Time, Voltages, Currents, Power measurements, Irradiance measure and Temperatures. The code is presented in the Annex B.

Time stamp [UTC]	V _{an} [V]	V _{bn} [V]	V _{cn} [V]	V _{npe} [V]	I _a [A]	I _b [A]	I _c [A]	I _n [A]	I _{pe} [A]	S [VA]	P [W]	Q[Var]	Cos (phi)	E _e [W/m ²]	T _{mod1} [°C]	T _{mod2}	T _{amb} [°C]
3510756000	235.96	236.53	236.74	0.0801	12.859	12.966	12.907	0.0064	0.0287	9157.2	9157.1	42.375	1.0000	819.20	17.042	17.292	26.775
3510756000	96612	34864	35741	37092	3478	592	76231	16447	70946	38468	40422	14063	10707	45127	84109	2456	78373
3510756000	235.84	236.36	236.60	0.0722	12.858	12.961	12.907	0.0056	0.0286	9150.3	9150.2	39.344	1.0000	818.50	17.045	17.388	26.773
3510756000	79706	56247	83592	61342	64464	25429	73335	46333	97423	02812	18223	77792	09244	90612	95407	56713	49807
3510756000	235.69	236.22	236.44	0.0610	12.858	12.949	12.902	0.0046	0.0301	9140.2	9140.1	39.130	1.0000	818.65	17.041	17.389	26.772
3510756000	48094	93955	31076	39024	52847	08172	00697	11283	18928	06951	23189	57672	09164	80787	34932	64459	61875
3510756000	235.59	236.06	236.28	0.0428	12.828	12.937	12.882	0.0053	0.0297	9120.4	9120.4	17.580	1.0000	817.57	17.046	17.388	26.772
3510756000	05323	94772	64889	63632	81911	87155	23211	12357	20944	56452	39508	89579	01858	48089	67705	57093	45761
3510756000	235.42	235.97	236.14	0.0285	12.829	12.925	12.878	0.0053	0.0273	9111.6	9111.6	17.466	1.0000	817.22	17.057	17.389	26.773
3510756000	47179	74376	28909	82602	46819	08571	90375	09974	25288	50078	33338	15192	01837	90794	63119	9861	97574

Figure 3: Example of a data sheet

1.2 Data analysis

As a first thing I analyzed the power production of the installation. Figure 4 shows an example over a 1 hour period.

⁵ This time stamp is just the number of seconds elapsed since the 1 of January of the 1904 at mid-day

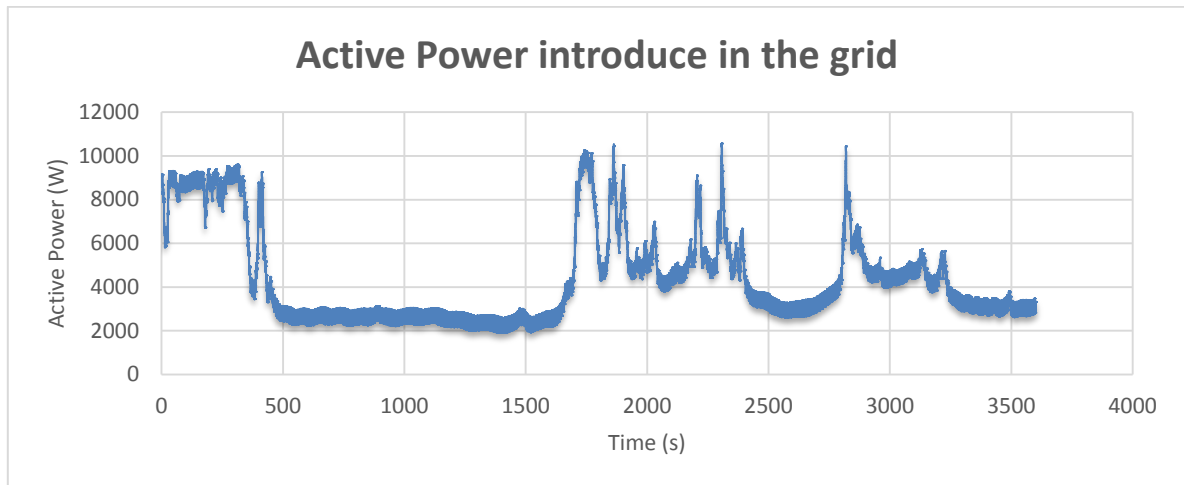


Figure 4: Active Power introduce for 1h in the grid

In this case I selected a cloudy day, the power peaks correspond to an increasing of the irradiance. This correspondence can be also seen in figure 4 where I plotted the irradiance for the same 1 hour period.

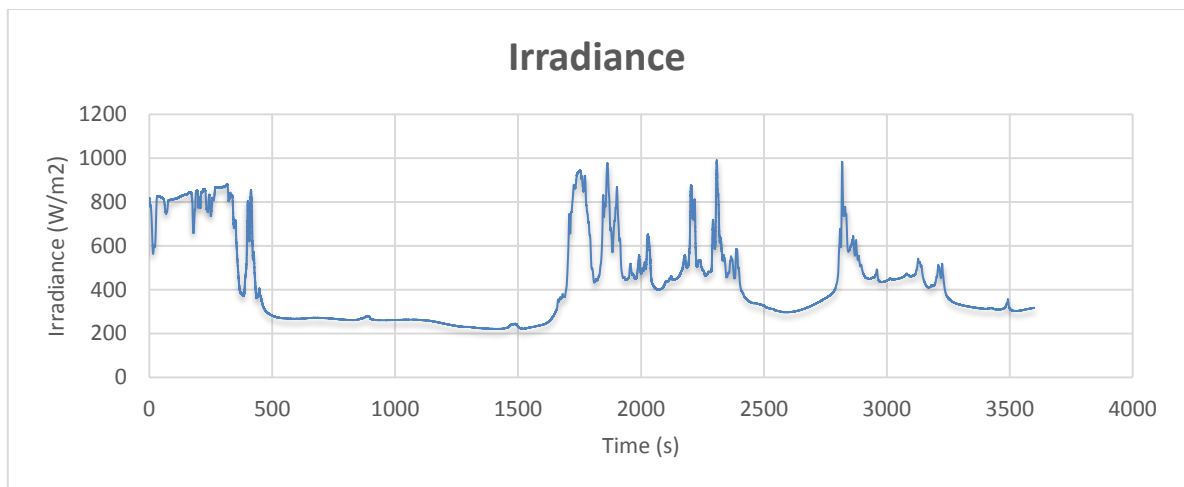


Figure 5: Irradiance for the same time period as Fig. 4

From the Fig.4 and Fig.5 it can be seen a proportional relation between both but also that the power line (Fig.4) is much thicker, in the moments the irradiance is stabilize, than the irradiance line (Fig.5). So we got into detail of the Fig.3 to see the dynamics of the active power in a short period of time. The next figure shows these short time dynamics.

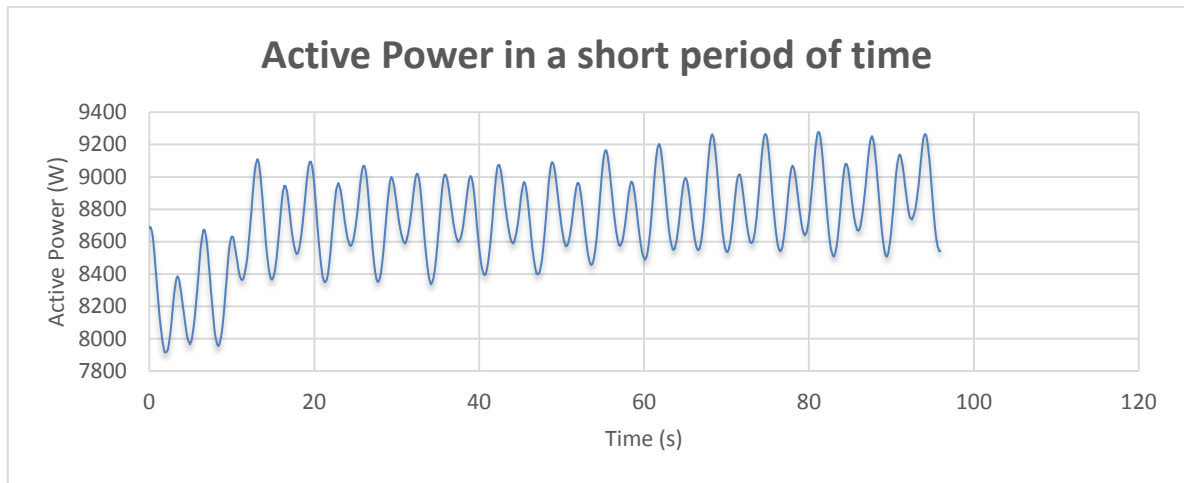


Figure 6: Short time dynamics of the active power

As we can see there is a strong oscillation in the power as it goes ± 400 W from what could be call the mean value. Also it's been check the amplitude of this oscillation doesn't depends on the power generated since it has the same amplitude when you are producing 2kW or 8kW. It is also clear that the oscillations are characterized by different amplitudes with different periods. These oscillations can be explained by the action of the MPPT of the converter. These oscillations has been attribute to the interaction of the MPPT controller inside the convertor that it's been used. These MPPT controller has an internal algorithm that tries to follow point of maximum output power along the curve V-I. An example of a general characteristic is presented.

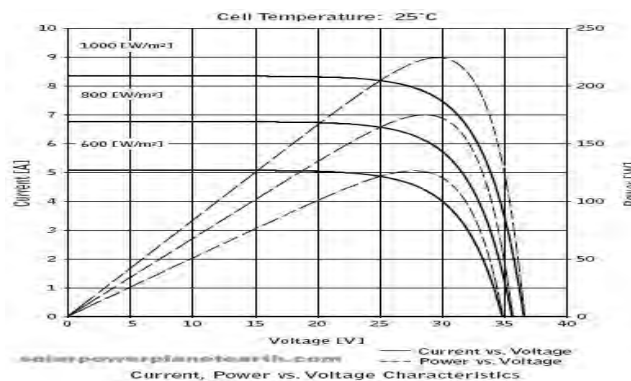


Figure 7: General Characteristic of a PV Panel

It has been impossible to find a function simple enough that could describe this oscillations (it's been seen that with a Fourier series is possible but too complicated for our scope). So I have focus on the long term dynamics of the converter by averaging these oscillations and considering the mean value. Following our intuition of linking the irradiance with active power



by a proportional constant, my first try after some research was to multiply the irradiance by a coefficient proportional to the surface of the panel, the amount of panels, the efficiency of the panels and of the converter. All the data related to the panel are in the data-sheet from the seller. For the efficiency of the converter is also a data sheet value. However it's been compute for being more precise as it's seen afterwards.

$$P = K * I;$$

$$K = N^{\circ}panels * A(m^2) * \eta;$$

This approximation calculates the output power by considering only its relation with the irradiance. The main limit is that the PV output power strongly depends on the cell temperature which is here neglected and it doesn't differ between direct and diffuse irradiance. A comparison of the error of the approximation for the different hours of the day and different days is presented in *Tables 1, 2*.

Hour	8h	10h	11h	13h	15h	17h	19h
Error	-17.3%	-18.6%	-12.45%	-6.37%	-6.98%	8.31%	-9.67%

Tableau 1: Error comparison between the predicted and real active power output on the 26/04/2015.

Hour	8h	10h	11h	13h	15h	17h	19h
Error	3.22%	1.13%	3.61%	7.1%	6.1%	2.85%	14.67%

Tableau 2: Error comparison between the predicted and real active power output on the 6/06/2015.

It's possible to see that the approximation is highly dependent on the month of the year and on the hour of the day. After this first approach it's been *Enrica Scolari* who worked on the prediction since it was more linked with his work. The idea was to use an available code, previously developed in the lab by *Dimitri Torregrossa*, to predict the solar irradiance on an ultra-short term time window (100 ms). The code was validated to forecast the irradiance and to predict the associated prediction interval (accounting for the uncertainty of the forecast). She has been working in the implementation of the code in Labview. The code has been validated to predict the AC power output when the converter works with the MPPT. One of the methods of point forecasting that is discussed in the paperⁱⁱⁱ and is been implemented is the *Holt Winter Method*.

In this method a first initiation of the matrix has to be done. For each forecasted point the error with real point and the derivative of the errors is calculated, and the matrix is uploaded. This matrix is compose of the probabilities of a having a different error per each derivative. This means in each row there are all the different probabilities of having an error for given derivative. Each element is the result of dividing the number of times you have had the same error for a given derivative by the number of times you have had that derivative. So for our



predicted point an interval of confidence is being set. This will give us the limits of where future irradiance or active power is going to be.

2. Converter Modeling

In order to success designing the PV agent a model for the converter has to be done. It's been considered as an efficiency between the DC power entering and the AC power in output with certain. Also the communications with it has been develop so implementing a set point can be possible. At last a study of the dynamics for a change in the setpoint has been done. The structure will follow:

- Communication with the converter
- Dynamics characterization
- Efficiency computation

2.1 Communication with converter

We first tried to communicate with it is been done through an Ethernet cable and using a Labview Library that was provided by an engineer that used to work in the company. This codes have been a great help since it gave us a point from where to start and not from scratch.

The first thing to do is to settle the converter accessible from remote control. Then a direction has to be assigned to each specific converter. This direction has to be a number between [1,249] and is done in the settings interface of the device. The next step is to turn on the Ethernet, it is done also in settings interface. Last thing is to settle an IP Direction in the same place. So now once you have done that you are able to start communicating.

In the Labview code you will need some different inputs. Regarding the device it's needed to introduce the IP address and the direction of the device. Depending if want to read from or to write on it, a different port has to be use. For reading the CP Abfrage is used while for writing is the CP Behefl. The last input is the key, this one come from the protocol of the converter. An example of the format of the message is the following: {FB;37;1A|64:XXX;XXX|055C}. In the XXX is where the keys are placed. It has to be taken into account the values it returns are in hexadecimal systems and have to be turn into decimal for a better comprehension. Also it gives each value with a certain scale that has to be transform to see the real measurement. These two transformation have been done in Labview with a few basic blocks. In the case it's wanted to implement a fix active power the following key is use: "PLR=" to the percentage of the total power the converter can give. This percentage is just a number in hexadecimal scale. A table to understand this in a better way is provided.



Reading	CP Abfrage
Writing	CP Behefl

Tableau 3: Ports selection

One thing it's been notice is that the buffer is only refresh every second and that it's only possible to set a new commission every second. This is going to be a great problem for our real time agent since the goal is to interact every 100 ms with the grid agent and the converter. A communication through a RS485 has been also tried since it could have been faster but it hasn't been manage and no improvement was found.

2.2 Experiment to characterize the dynamics

An experiment to see the dynamic response to a change on the set point has been run. It changes in the set points can be seen as different steps. The experiment has been done with different step size and run many times to gather as much data as possible.

Before running the experiments a few modifications and adding have been done to the initial code we received. First of all the initial code has been duplicated to so it can read from and write in the converter at the same time. Firstly the reading loop is presented and afterwards the explanation of the writing one will follow. The code is include in the annex B for a batter comprehension and is called 'mhaLib_ETHComm_rev3.vi'.

For the reading loop a set of blocks to disaggregate the message we receive is been set. This disaggregation has been especially design for the case where the active power (AC and DC) and the voltage and current of one of the phases are asked. After transforming the measurements into decimal base are safe in TDMs file. These TDMs is generated each time the program is run and can be saved in any folder of the computer. At last a time loop is over the whole structure so it runs continuously till the stop button is pressed.

The writing loop runs in parallel with a time loop that shares the stop button with the reading loop. The whole code to communicate is inside a case loop which is only true when the button is pressed. At last in order to know when the step occurs and be able to search each step a time stamp has been put.

So each step has two different files which records the data, one from the measurements board and another from the measures given by the converter. The cleaning and workout of the data has been done in first place with the one from the measurements board as we believe is more realistic and has been less filter. So the first thing it's been done is to cut and plot to have a first view of how the steps where. For the specific analysis it's been done in Matlab. Now an example of the plotting in excel are presented.

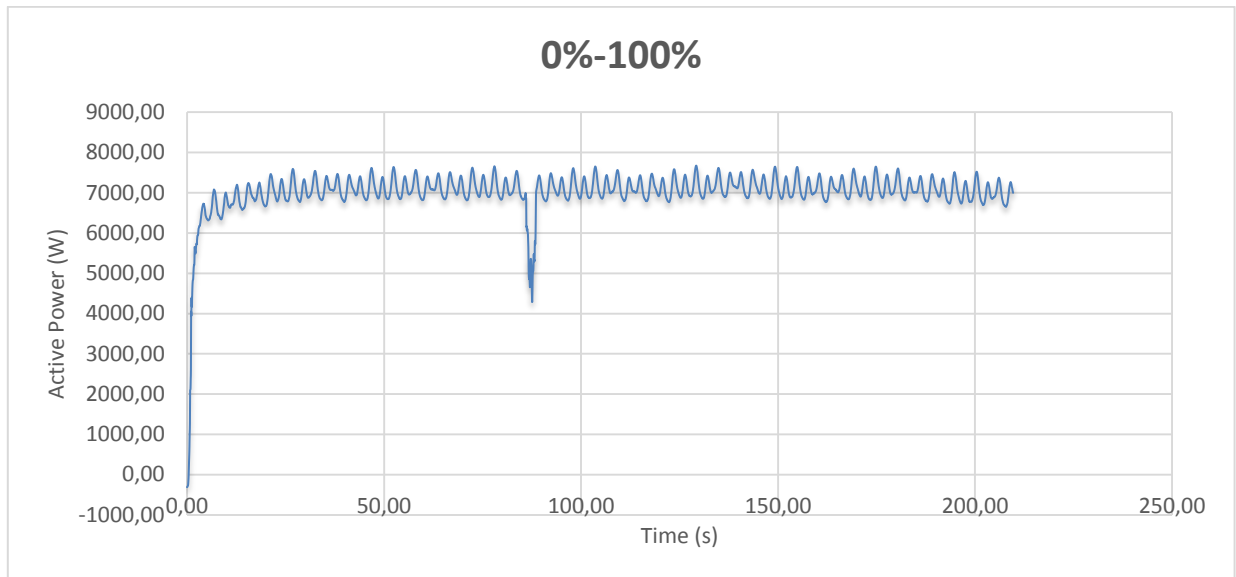


Figure 8: a 100% step

As we can see the response looks as the one of a first order system or as a second order system with a damping factor close to 1 which means an overshoot close to zero. Now an explanation of the Matlab code it's used will come.

Since we have such a noise in the response the first thing we have done is to calculate moving average of the response, so it's easier to analyze it. To initialize the moving average a set of previous points is needed. Depending on how many points are taken the response gets more fitted or not. An example is now shown.

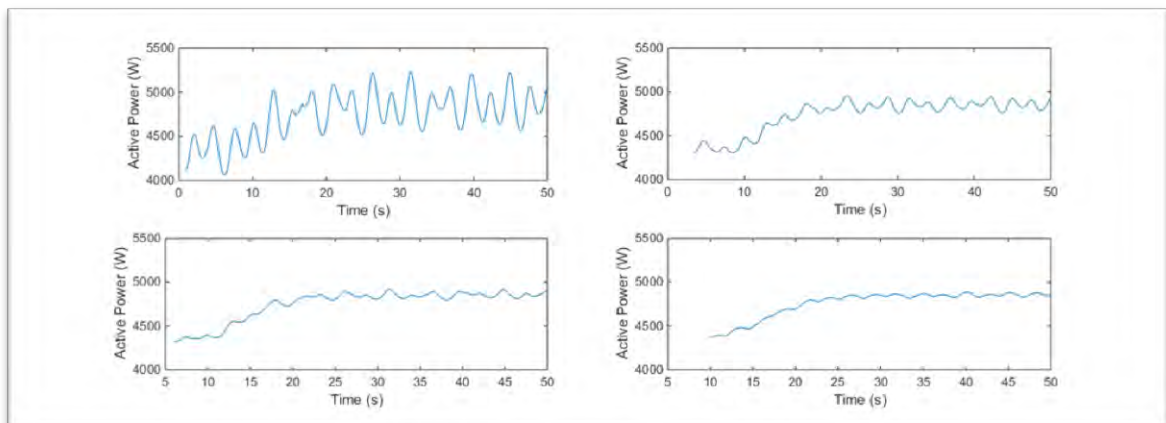


Figure 9: example of moving average with different amount of points for the initialization. Lags of 20, 70, 120 and 200 points.



It's shown that with a lag of points below 100 there is still too much noise and if a big set is taken too much information is lost for the small gain in resolution. So it's been estimated that optimal lag to balance the resolution and the information is 140. Since each point is every 0.05s this means the response has to be taken 7s before the step is done. Another analysis it's been done is if it's possible to get a better resolution doing a moving average of a moving average with a lag that is half of the bigger one. This has been done with two lags of 70 and with one of 140 and it's presented in the following figure.

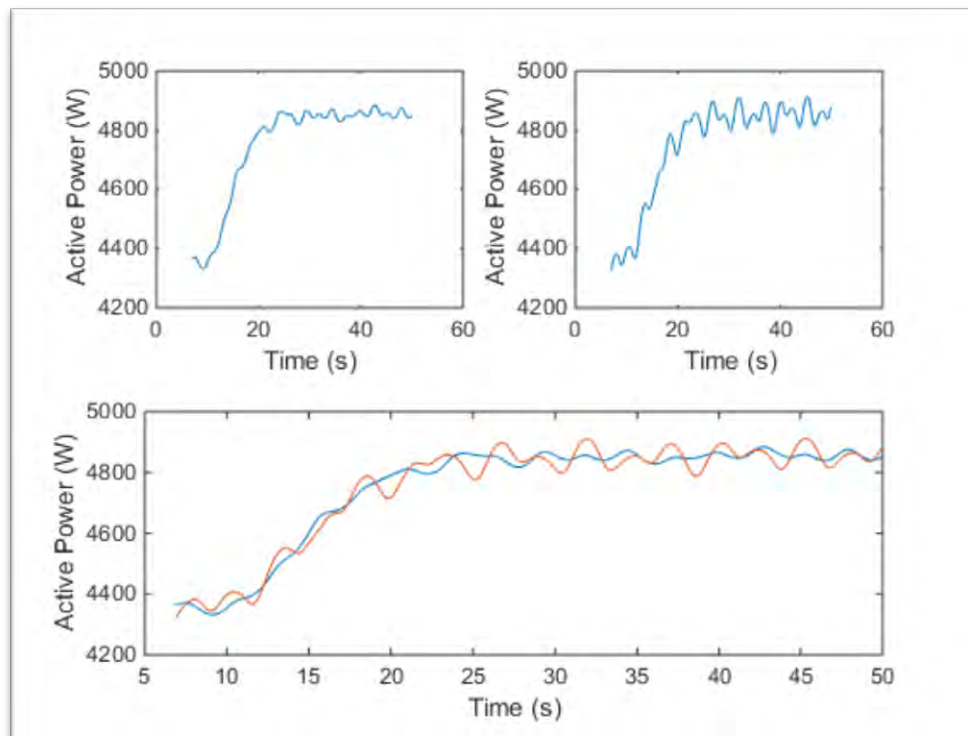


Figure 10: the 1^o plot is with 2 M.A. of 70 points as lag. 2^o one with 140. Then they are plot together.

It can be seen a lower oscillation amplitude is achieved taking the same amount of points and doing it twice, so for the following analysis this option has been chosen. The next step it's to set the 1^o order and 2^o order. Another figure of the moving average plots compare to our initial data is presented.

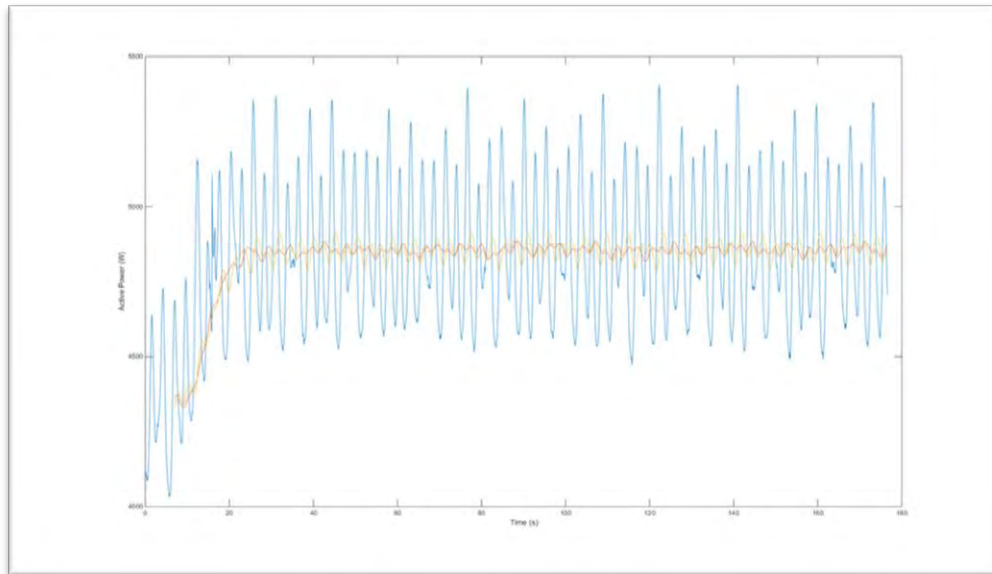


Figure 11: real response against the moving averages approximations

To define the 1^o order function the time constant of the exponential has to be found. This time constant is defined as the time to reach 63% of the final value. In our case it's been calculate that is $\tau=7s$. The comparison between real response vs M.A. vs 1^o order system is presented. The equation of 1^o order system in time domain is:

$$y(t) = (f(\infty) - f(0)) * \left(1 - e^{-\frac{t}{\tau}}\right) + f(0)$$

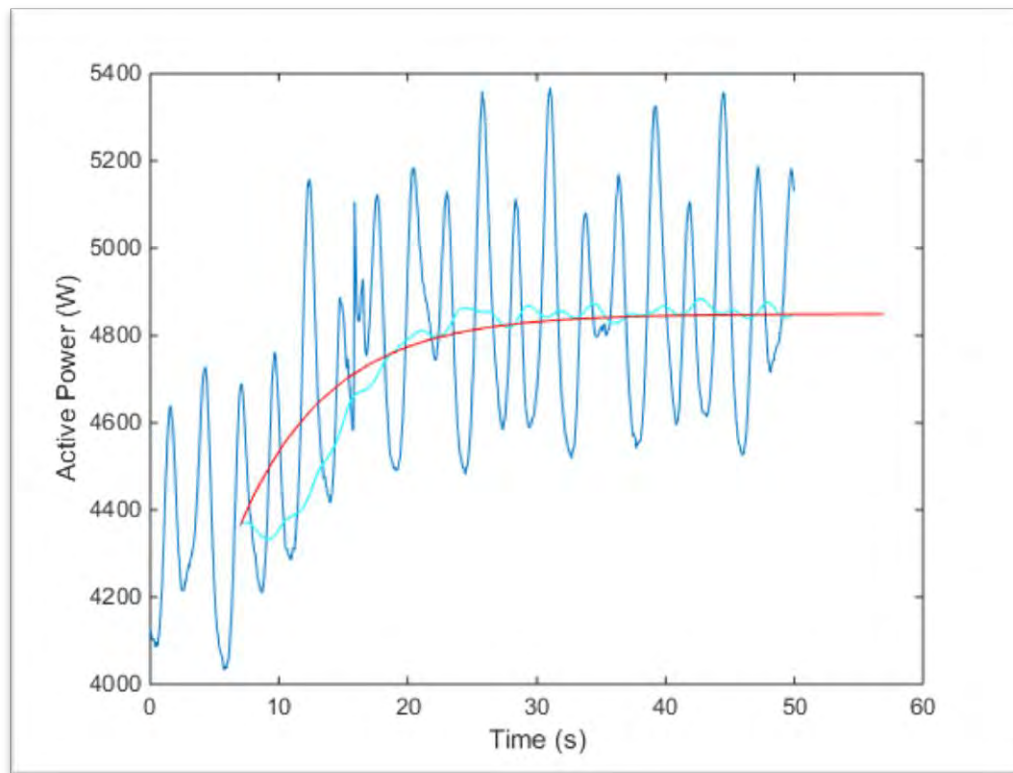


Figure 12: real response vs M.A. vs 1^o order system

For defining the 2^o order two parameters are needed the peak time and the overshoot. The overshoot is defined as the difference between the peak and the final value divided by the difference between final and initial value. For doing that we have calculate the mean value once the response is stabilized as the final value. For the initial value the first point is been taken. The peak time is also applied in the 1^o order system. For the peak time it's been taken the one of the highest point in the original response and for the peak point is the maximum of the moving average set. Overall the different responses a mean value of each parameter is been calculated, which lead us to an $M_p=0.04$ p.u. and $T_p=22$ s. In order to be able to implement it as temporal function the poles have to be calculated. These ones are $P=-0.1463\pm 0.1428i$. Now a representation of the 2^oorder function compare with the moving average of a general step done in the experiment is included.

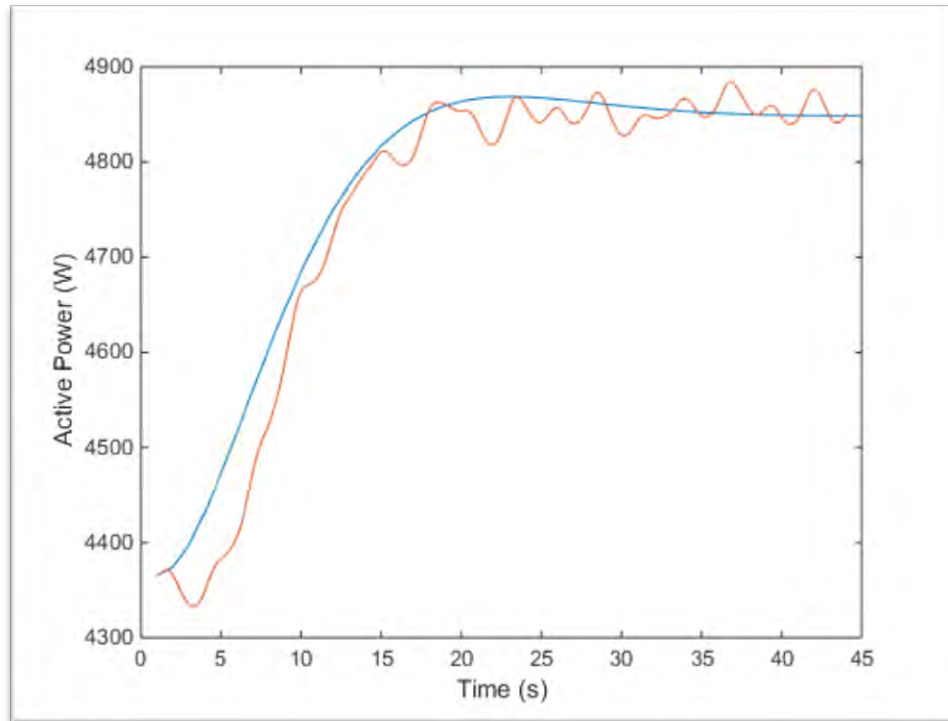


Figure 13: M.A. vs 2^o order system

As it can be seen is a good approximation of the dynamics, so now it has to be implemented in Labview. The 2^o order study has been proceeded since in some response it was clear there was an overshoot. So between the two approximations it's been chosen the 2^o order system as it's believed to be more accurate (it has given almost the same error) and is more real than 1^o system. Other type of approximations have been done in Matlab with the least square method with different models, and no good enough results were found.

For the labview implementation a subVI has been design with basic blocks as a time function. The general equation of a 2^o order system in time domain is:

$$y(t) = 1 - \frac{1}{\cos(\alpha)} * e^{-a*t} * \cos(b * t - \alpha) \text{ with } p = a \pm bi$$

Which p are the poles of the second order transfer function:

$$G(S) = K * \frac{Wn^2}{(S^2 + 2 * \zeta * Wn * S + Wn^2)}$$



2.3 Efficiency of the converter

Currently, the measurement board installed in the lab is able to return only the AC values. However, in order to compute the efficiency the DC measurements are also needed. For this reason, it was decided to use the values returned by the converter protocol itself and use them to calculate the efficiency. So in order to compute the table, the converter has been left running at full available power and recording data from the buffer for one whole day. The data have been reorganized in increasing order and the efficiency have been computed.

$$\eta = \frac{P_{ac}}{P_{dc}}$$

It's been seen as it's shown below that for values below 1000 W the measurements don't make any sense reaching an efficiency over 250% which is impossible. These errors have been attributed to the interference of the filter and the fact the converter is running at a really low power. It's been seen the converter does not run below 700 W even if is commanded. The next graph presents the efficiencies at different power rate.

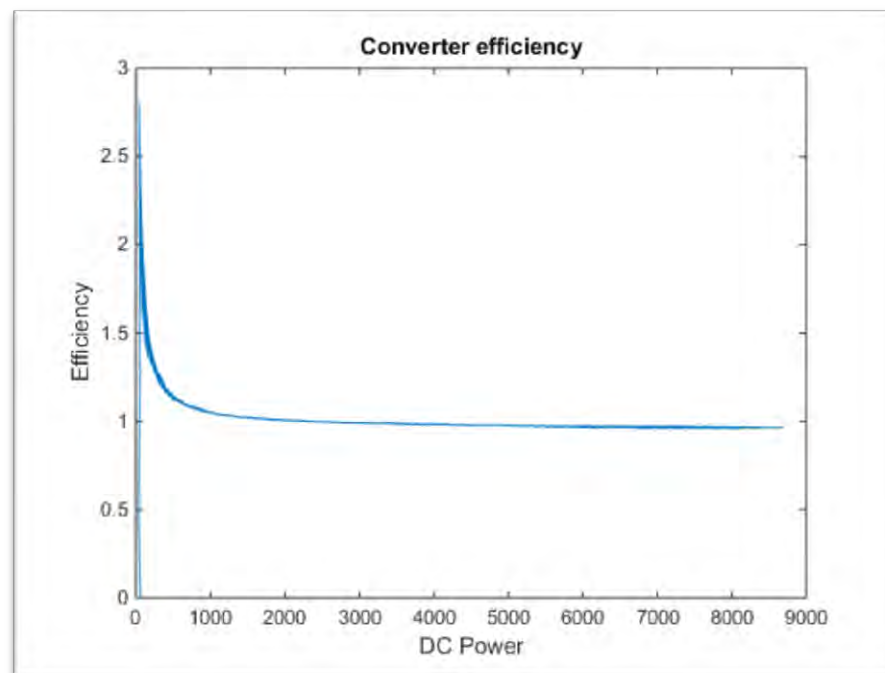


Figure 14: Converter efficiency computed with the recorded measurements

For this reason the converter efficiency is for now considered as defined by the vendor and it will be computed when the DC measurements will be available. The curve is now presented.

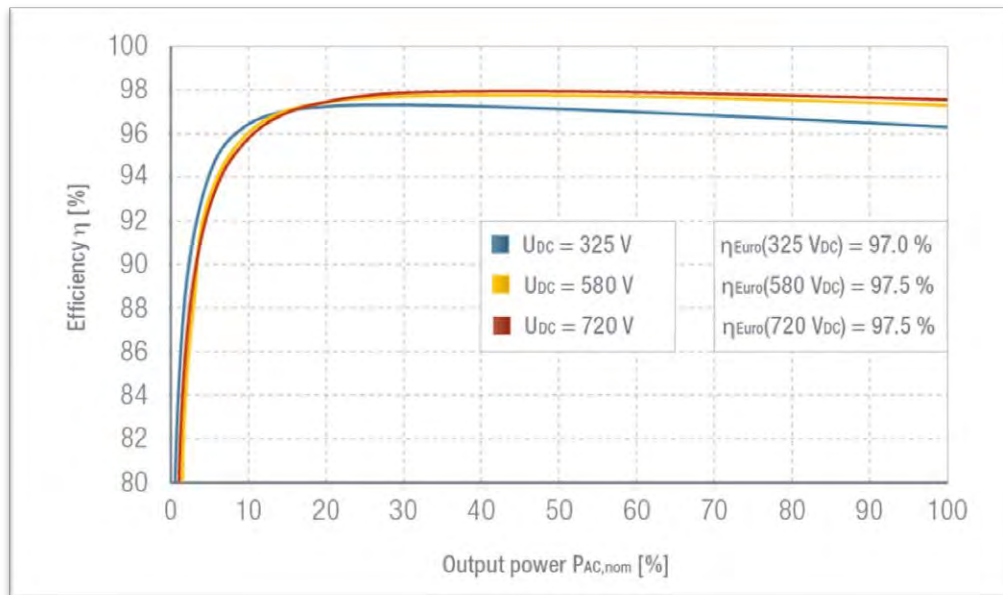


Figure 15: Efficiency of the converter from the vendor.

3. Implementation of the agent in Labview

Now the explanation of the implementation in *Labview* will follow. The agent should be able to read the measurements in real time, to forecast the future available power, to command a set point to the converter and to send the advertisement to the grid agent. For the RT measurements the code used is the one explained in chapter of data acquisition.

As mentioned, during the development of the agent we encountered an experimental problem. The converter is able to implement a requested command every 1 second and it needs some second to stabilize to the requested set point. For this reason an alternative solution is needed; in fact the COMMELEC framework would need an answer from the converter in the ms scale. The alternative solution is to consider the PV agent as an uncontrollable resource.

However the initial idea has been carried on and the controllable agent will be used and tested in the future, when a faster converter is available.



3.1 Uncontrollable agent

As it's consider uncontrollable it makes no sense to try to send a command so in these case there will be no communication with the converter. For prediction will consist on predict the following available power using the Holt Winter Method as point predictor and the dynamic predictor for the prediction interval. In this case it is possible to directly predict the output power and there is no need of a physical model that returns the power starting from the predicted irradiance, since no set point is asked. The output power is related to the solar dynamic. The prediction code is initialized with past historical power data and it's recomputed with the real-time power data as input shared variable from the outputs of the measurements. The prediction code has been separated in three parts (VI) since it was unhandled. Those three VIs correspond to the three frames it was composed and are call 'Initializationrev2', 'Matrix initiation' and 'Matrix computation'. The last VI has inside a time loop which it could have been put it outside but since most of the variables are feeding backwards it would have mean having a considerable number of outputs add it to the already big number of inputs it has. For that reason the two outputs values are needed (limits of the prediction interval are written in a global variable.

The PQt profile has been settle as the actual point the converter is running at. So instead of being an area in the PQ space, it will be the point that comes from the measurements board. The belief function has been define as the prediction interval of active power available with the same reactive power it's operating. This will lead us to a vertical line where the actual (P, Q) should be place in. The last thing it has to be advertised to the grid agent is the cost function. Since there is no willing to change, either to produce more or not the cost function is zero. Every cost function is define as polynomial of second order and only the coefficients have to be set. So for our case all these coefficients are define as zeros as the $C_{ui}(P,Q)=0^{iv}$.

A code to transmit all these has been design. The P and Q (for the PQt profile) are inputs of the VI as local variables from the outputs of the measurement VI. The prediction limits are inputs as global as they are generate inside of VI that has a time loop that doesn't end till the agent is stopped. The last two inputs are the maximum electric power of the converter (VA) and a cluster with the information of the requested point (Preq, Qreq) which is to plot it with the PQt and the BF. The information of the cost function is define as constants inside the code. All these information is put together in a cluster and transmitted as an output. In these code there is also a graphical representation of the request point, the performance point and of the belief function. These representations are done through arrays with different (P, Q) points. The arrays are built through *For* loops. The name of the code is 'Agent process Uncontrollable agent' and it will be include in the annex B with the rest of the labviews codes. Figure 14 shows the front panel is and an example of a graphical representation.

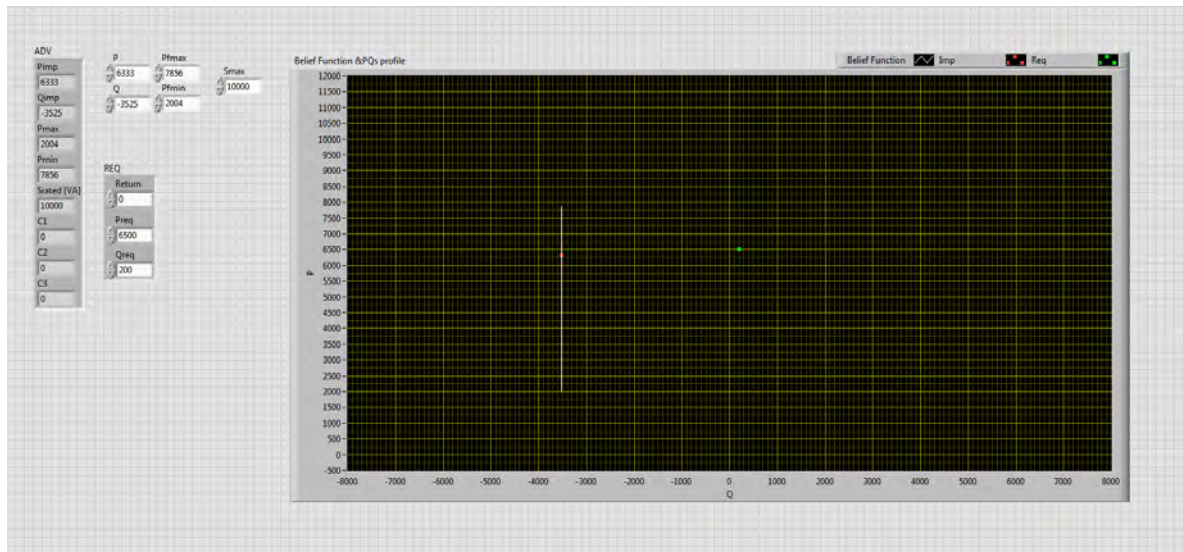
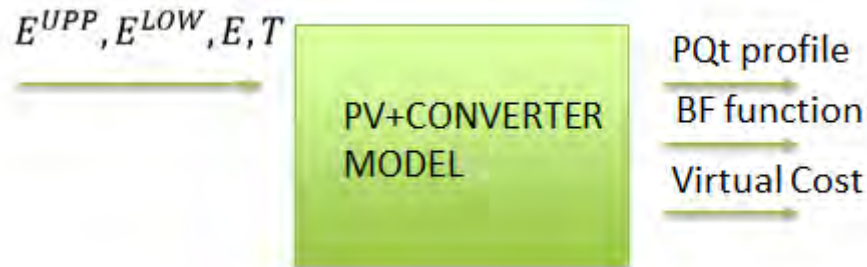


Figure 16: Front panel of the 'Agent process Uncontrollable agent'

To conclude the main code is formed by one time loop to compute the FPGA in parallel with a time loop that computes the measurements, the prediction code of the point and interval and a time loop that simulates the interaction with the agent providing a request and computing the advertisement. All the timed loop except the one of the FPGA are computed every 50 ms. There is only one main stop controller that is written in a global variable in order to stop the inner loop of the prediction code. This global variable is first written as false. The other loops share main stop as local variables.

3.2 Controllable PV Agent

In these case the model will predict the maximum power from the irradiance and the temperature. The temperature is consider it will be the actual since no significance variance may occur in such small period of time (50ms). The irradiance point and interval is predicted with the code it has already been talked about. The idea is with this four values have it as an input of the PV + Converter model to get the output power and at the end the PQt, BF and VC.



The model of the PV array it's been provided consist in five parameters. The 5 parameters allows to find the I-V curves (which the MPPT is based on) of the PV array based on datasheet values from the vendor. The 5 parameters are computed in STC and then updated based on the actual irradiance and temperature. The model is based on the one diode model for PV arrays and the set of equations can only be solved numerically. In Figure 17 the model is presented.

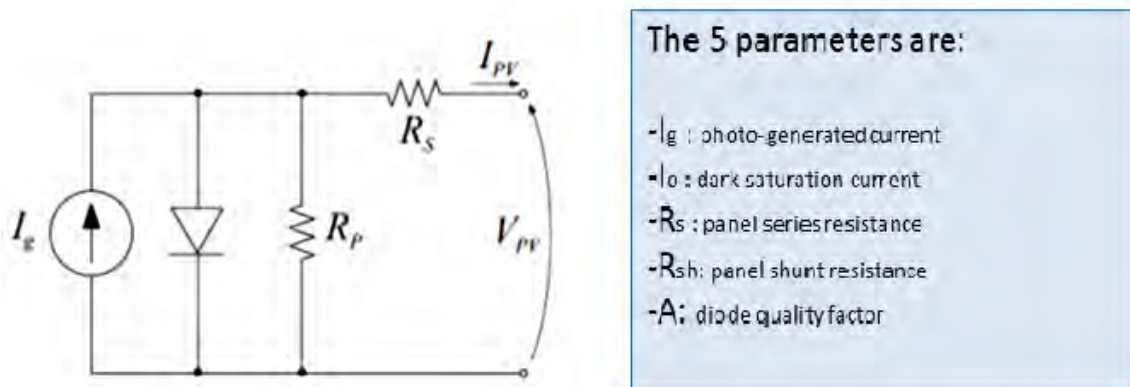


Figure 17: One diode model of PV array.

Once the maximum power is obtained, it's computed with the efficiency of the convertor. With the transfer function that is been calculated the agent is able to predict the response of the convertor at given instant t when a certain set point is asked.

Before implementing the power request, it has to be project in the PQt profile so it's an accessible point. These PQt profile is define by three parameters: the S_{max} from the convertor, the minimum Cosine "Phi" it would like to have and the maximum P at that moment from the irradiance and the convertor dynamics. With these three parameters, the intersections can give us three different figures where it possible to project. These projection of the PQ request can be done in many different ways such as the closest point inside the area



to the PQ request. In our case the projection has been done trying to collect the maximum active power at first and then optimizing the reactive one. In our case, and in order to simplify the projection method, it's assumed that the agent wants to maximize its active power production, as it talk in the paper for the VC, according to its constraints. The way to do so is comparing the request with the limits of the PQt. It has been taken the advantage of projecting and fixing firstly the active power to afterwards do it with the reactive one. So firstly the point is moved in the scalar coordinates and afterwards in the polar ones. The name of the VI is "Projection2.0". The comparison equations are presented now and all numbers are scalar:

$$1. P' = \min(Smax, Pmax);$$

$$2. P' = \min(P', Preq);$$

$$3. Q' = \min\left(\sqrt{Smax^2 - P'^2}, Qreq\right);$$

$$4. Q' = \min\left(Q', P' \frac{\sqrt{1 - \cos_{min}^2(\phi)}}{\cos_{min}(\phi)}\right);$$

These last two equations are valid for the case that $Qreq > 0$, in the case it was $Qreq < 0$ these other two comparisons are done. So in order to differ each case structure has been use.

$$3. Q' = \max\left(-\sqrt{Smax^2 - P'^2}, Qreq\right);$$

$$4. Q' = \max\left(-P' \frac{\sqrt{1 - \cos_{min}^2(\phi)}}{\cos_{min}(\phi)}, Qreq\right);$$

Some graphical representations presenting the different forms of PQt profile with request point already projected are presented in Figure 18.

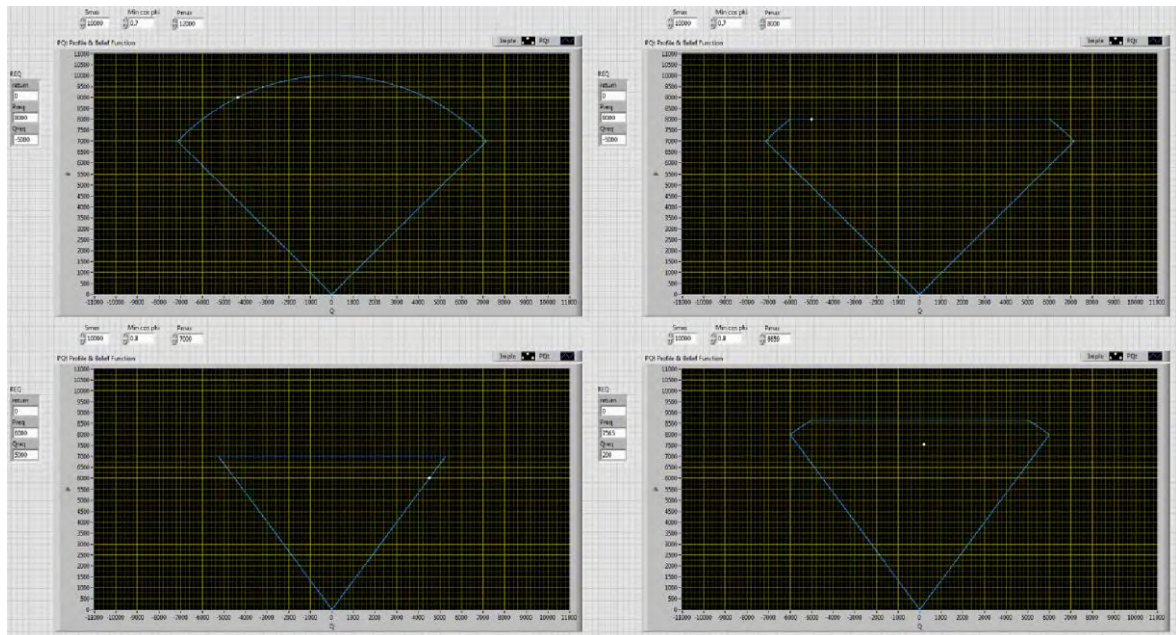


Figure 18: Different PQt profiles are presented with the correspondent projection.

Once it's been project and check is inside the PQt, the converter has to be commanded. The protocol that is been explained is been used. It has few modifications so the power request input is transform directly to hexadecimal system and the PLR key is not necessary to write it. In this code there is only the writing loop. The name of the VI is "Send command" and belongs to the mhaLibrary.

Now it has to communicate the advertisement to the grid agent. The cost function is going to be $C=-P$ as it's tried to produce always as maximum power as possible. So C_2 is defined -1 and the other two coefficients are zero. The PQt profile has already been explain and the belief function are the possible values of Q for an active power equal to $[P_{imp} \pm 400 \text{ W}]$. In the code both are plot together, the PQt and BF with the request form the GA, the Commanded point and the implemented point. The inputs of these VI are the Smax, the implemented point (P_{imp} , Q_{imp}), the predicted interval that comes as local variable and Min $\cos(\phi)$. The name of the VI is "Agent process controllable agent". Inside the VI a subVI is call to produce all the arrays that are plotted and is call "PQprofile draw".

In the "PQprofile draw" VI is plotted the PQt and the Belief in parallel. The PQt profile is built by adding arrays that start and finish in the intersection points between the curves. With these 2 functions, the commanded point is plotted so the function to project the request is also called inside the VI.



Conclusion

This report presents how the PV Agent has been implemented in Labview. Although the Controllable Agent has not been tested in real time due to some physical constraints, a first approximation of how it should be developed for a faster converter is presented. Some of the VIs realized in the current project, can also be used further in time for its implementation in the COMMELEC framework. A further research on implementing the Q req should be done.

On the other hand an Uncontrollable Agent has been finalized and can be implemented as an alternative solution to include the PV resource in the COMMELEC.

It has served me to apply many different concepts from many different fields in the industrial engineering acquired during my bachelor.

Annex A

This annex will include the different codes used in Matlab for the in order to compute and find the TF.



MAaverage.m

```
%% Code to start calculating a transfer function for a step reaction%%
%% 1° Step Calculate the moving average, since there are lots of
oscilations%%
%% First thing we call the Excel data, the script structure is linked
with the specific structure of the Excel, so changes may have to be
done%%
filename='DESL_PVs_MNTRNG_2015_04_24_13h';
sheet='Step Simulations';
Data=xlsread(filename,sheet); %% Remember the file has to be in the
same folder as the script or either put the path

d=1; %%so it reads a column not a row
Time=Data(:,31); %%just need to change the digit depending on the
column
PW=Data(:,32); %%same same

F=tsmovavg(PW,'s',20,d);
G=tsmovavg(PW,'s',70,d);
H=tsmovavg(PW,'s',120,d);
J=tsmovavg(PW,'s',250,d);

figure(1)
subplot(2,2,1)
plot(Time,F)

subplot(2,2,2)
plot(Time,G)

subplot(2,2,3)
plot(Time,H)

subplot(2,2,4)
plot(Time,J)

G=tsmovavg(PW,'s',70,d);
G2=tsmovavg(G,'s',70,d);
H=tsmovavg(PW,'s',140,d);

figure(2)
subplot(2,2,1)
plot(Time,G2)

subplot(2,2,2)
plot(Time,H)

subplot(2,2,[3,4])
plot(Time,G2,Time,H)

figure(3)
```



```
subplot(2,1,1)  
plot(Time,PW,'r',Time,G2,'y')
```

```
subplot(2,1,2)  
plot(Time,PW,'r',Time,H,'g')
```

TFfor1order.m

```
%% Code to calculate a 1°order transfer function for a step reaction%%
```

```
%% First thing we call the Excel data, the script structure is linked  
with the specific structure of the Excel, so changes may have to be  
done%%
```

```
filename='DESL_PVs_MNTRNG_2015_04_24_13h';  
sheet='Step Simulations';  
Data=xlsread(filename,sheet); %% Remember the file has to be in the  
same folder as the script or either put the path
```

```
d=1; %%so it reads a column not a row  
Time=Data(:,31); %%just need to change the digit depending on the  
column  
PW=Data(:,32); %%same same  
syms t s;
```

```
%% 1° Step Calculate the moving average, since there are lots of  
oscilations%%
```

```
l=140; % Lag use for the moving average  
n=140/2; % Has to be an integer  
G=tsmovavg(PW,'s',n,d);  
G2=tsmovavg(G,'s',n,d);  
G=tsmovavg(PW,'s',l,d);
```

```
figure (1)  
plot(Time,PW,Time,G2,Time,G)
```

```
figure (2)  
plot(Time,PW,Time,G2)
```

```
R=G2(600:800);  
L=length(R);  
A=ones(1,L);  
Q=A*R;  
F=Q/L; %%is Y in the infinite  
U=G2(1); % the lag that has been taken for the moving average and is  
the value in t(0+)  
P=max(G2); % maximum value of the funcation
```



```
a=0.63*F;
a=round(a);
T=find(G2==a);
T=(T-1)*0.05; % 0.05 is the period between each time stamp
T=7;

%% Now we define the function of the model%%
D=(F-U)*(1-exp(-Time/T))+U;
ft=(F-U)*(1-exp(-t/T))+U
k=F-U
T
p=U
Fs=laplace(ft)

figure (3)
plot(Time,PW,Time,G2,'c',Time+7,D,'r')
```

TFfor2order.m

```
%% Code to calculate a 2°order transfer function for a step reaction%%

%% First thing we call the Excel data, the script structure is linked
with the specific structure of the Excel, so changes may have to be
done%%
filename='DESL_PVs_MNTRNG_2015_04_24_13h';
sheet='Step Simulations';
Data=xlsread(filename,sheet); %% Remember the file has to be in the
same folder as the script or either put the path

d=1; %%so it reads a column not a row
Time=Data(:,10); %just need to change the digit depending on the
column
PW=Data(:,11); %%same same
syms t s;

%% 1° Step Calculate the moving average, since there are lots of
oscilations%%
l=140; % Lag use for the moving average
n=l/2; % Has to be an integer
G=tsmovavg(PW,'s',n,d);
G2=tsmovavg(G,'s',n,d);
G=tsmovavg(PW,'s',l,d);

figure (1)
plot(Time,PW,Time-3.5,G2,Time-3.5,G)

G2=G2(1:800);
Time=Time(1:800);
PW=PW(1:800);
```



```
figure (2)
plot (Time, PW, Time-3.5, G2);

R=G2 (600:800);
L=length (R);
A=ones (1, L);
Q=A*R;
F=Q/L; %is Y in the infinite
U=G2 (1); % the lag that has been taken for the moving average and is
the value in t(0+)
P=max (G2); % maximum value of the funcation

%% Now we define the function of the model%%

Mp= (P-F) / (F-U);
alpha=atan (log (Mp) / (-pi));
zeta=sin (alpha);

tp=find (PW==max (PW));
tp=(tp-1)*0.05; % same same
Wn=pi/ (cos (alpha)*tp);

num=Wn^2;
den=[1 2*zeta*Wn Wn^2];
Mp
tp
Fs=tf (num, den)
[A, B, C, D] = tf2ss (num, den);
```

Test.m

The code presented is just a try of how the resulting TF of second order is in time domain.

```
tp=22;
Mp=0.04;
alpha=atan (log (Mp) / (-pi));
zeta=sin (alpha);
Wn=pi/ (cos (alpha)*tp);
num=Wn^2;
den=[1 2*zeta*Wn Wn^2];
Mp
tp
Fs=tf (num, den)
[z, p, k] = zpndata (Fs)

syms t;
a=0.1463;
b=0.1428;
```

```
Y (t)=1-1/cos (alpha)*exp (-a*t)*cos (b*t-alpha)
```




```
figure(1)
plot(2+2*Y(0:0.1:40))
figure(2)
plot(2-1*Y(0:0.1:40))
```

Annex B

This annex will include the codes used and implemented in *Labview*. There will be presented in three parts: Communication and experimental codes used during the project, code of the Uncontrollable Agent with the subVI's and the Controllable Agent his subVI's.

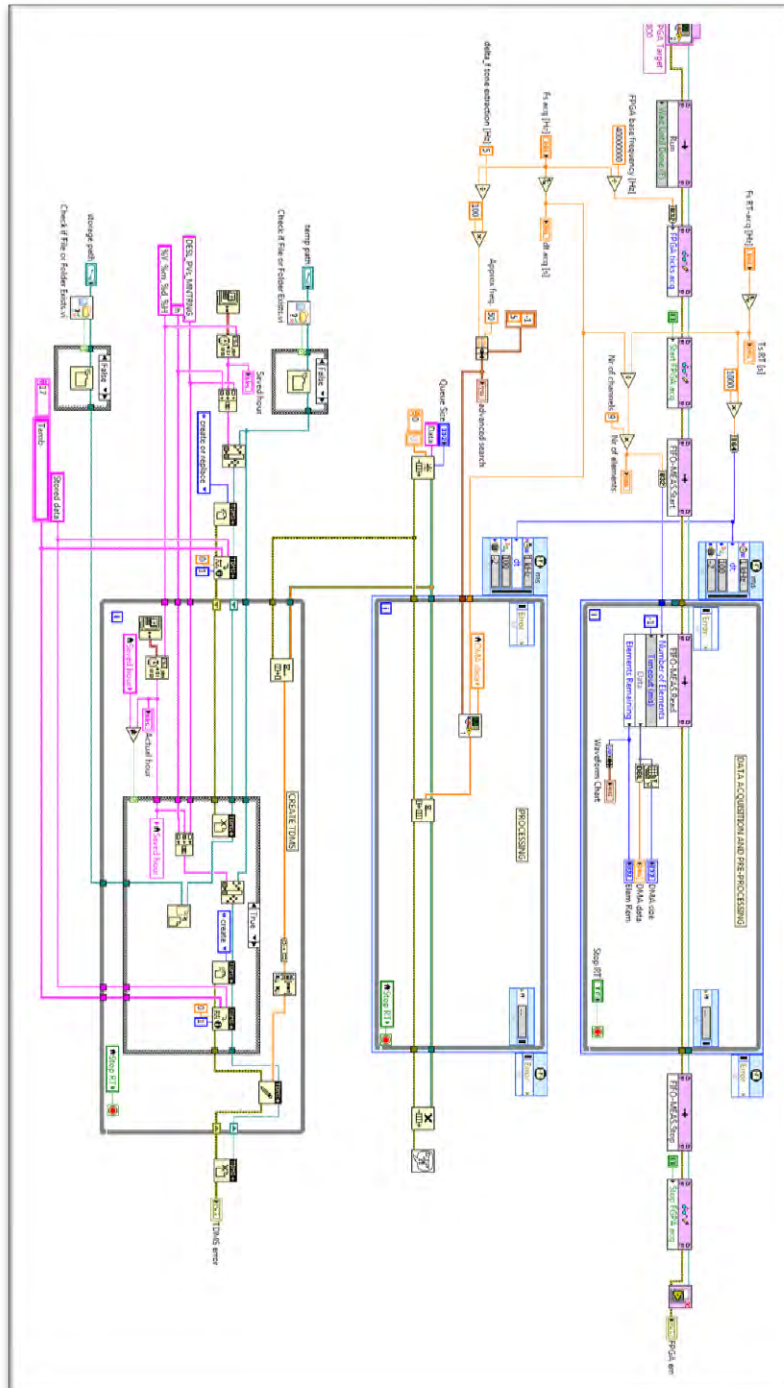
Communication and experimental codes

Codes for real time computation of data

This is the code that has been used for recording the data. It's called 'RT rev8'.



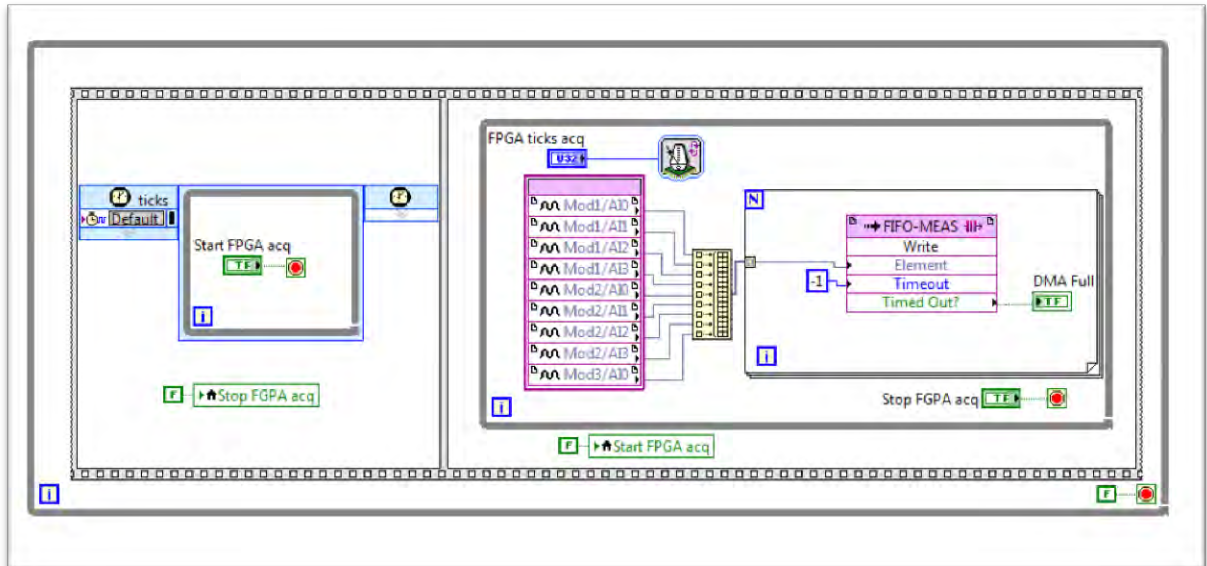
RT rev8.vi



There are two sub Vis dependent of this code: the FPGA computation and the 'Measurements' code.

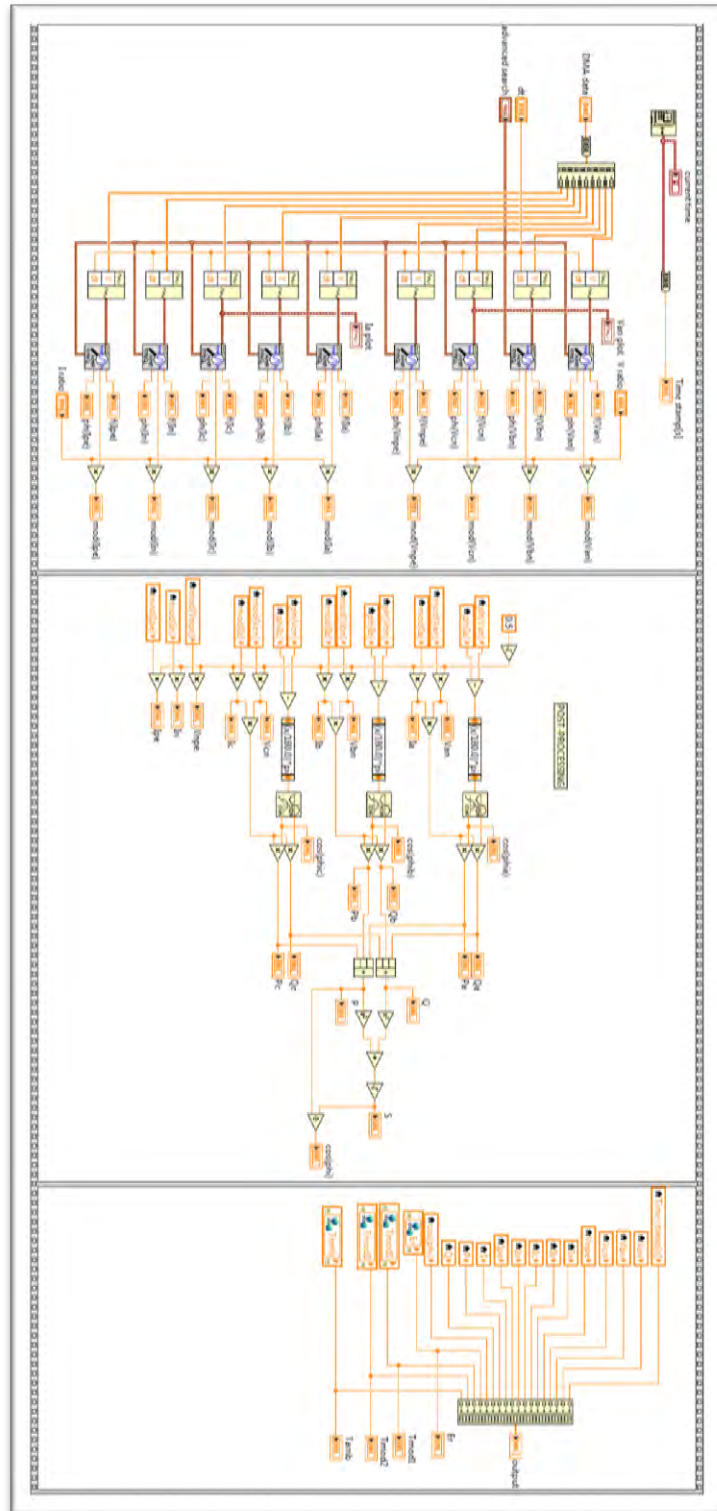


FPGA.vi





Measurements.vi

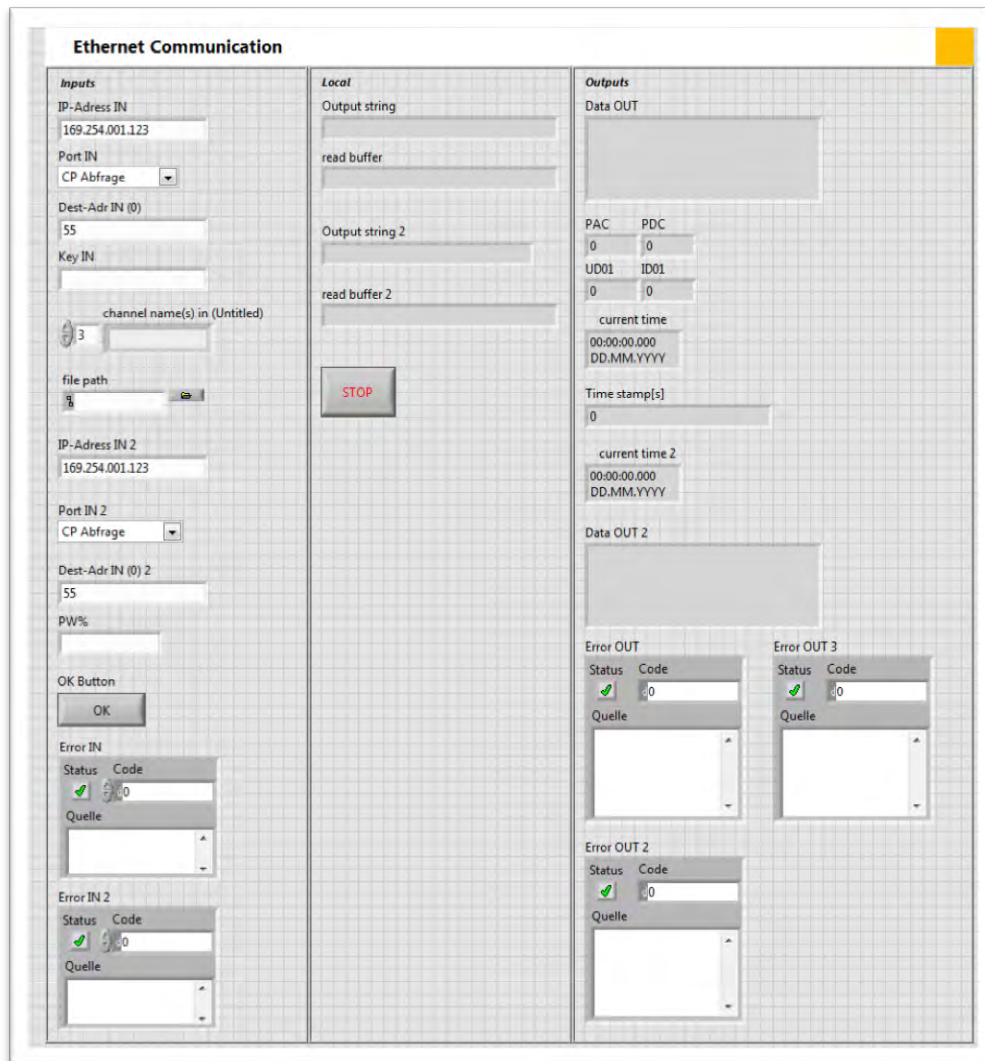




Communication code

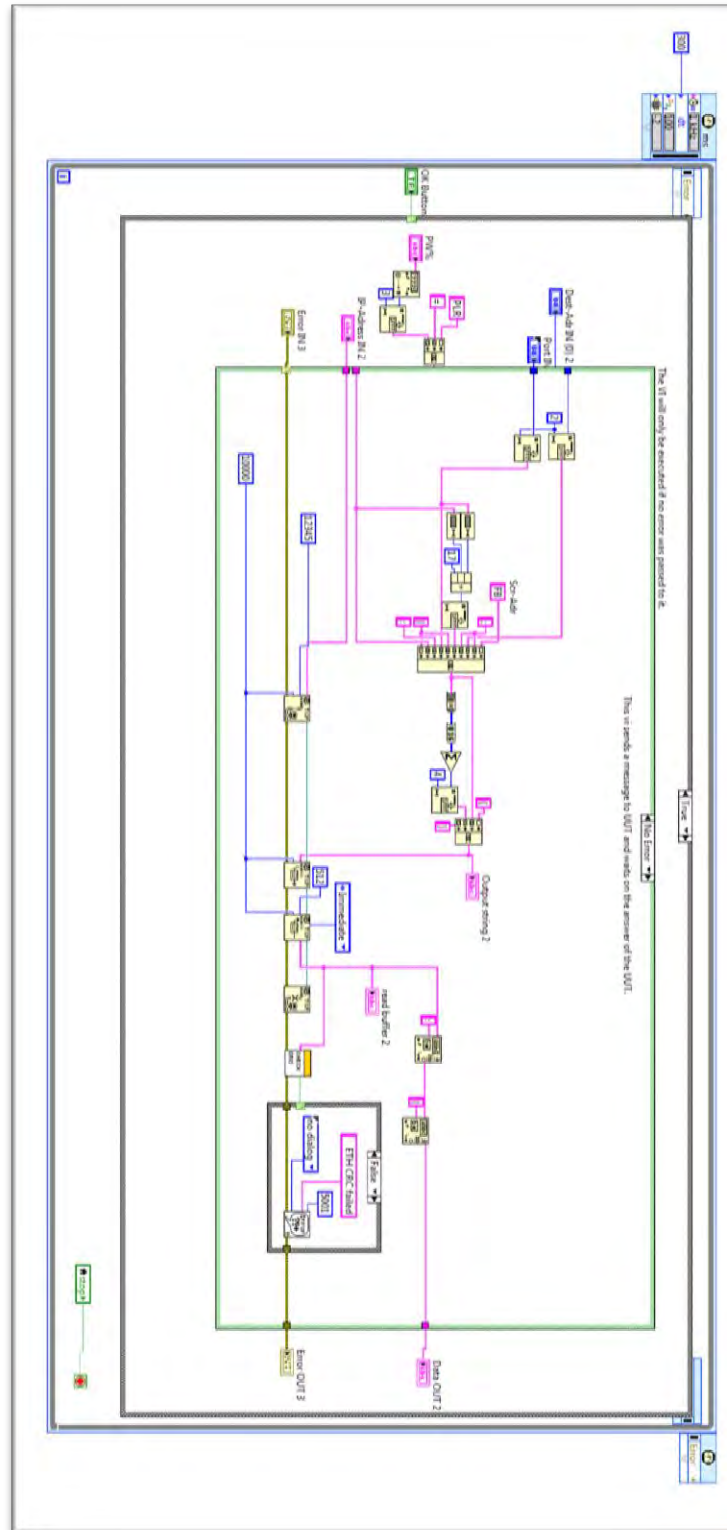
The front and back panel are both presented since it's interesting to see the interface of the VI. It was used to run the experiment of the set points.

mhaLib_ETHComm_rev3.vi Front panel



mhaLib_ETHComm_rev3.vi Back panel

The code is shown in two parts because it was impossible to fix it in one. The first one is the reading loop and the second one is the writing. It was used to run the experiment of the set points.





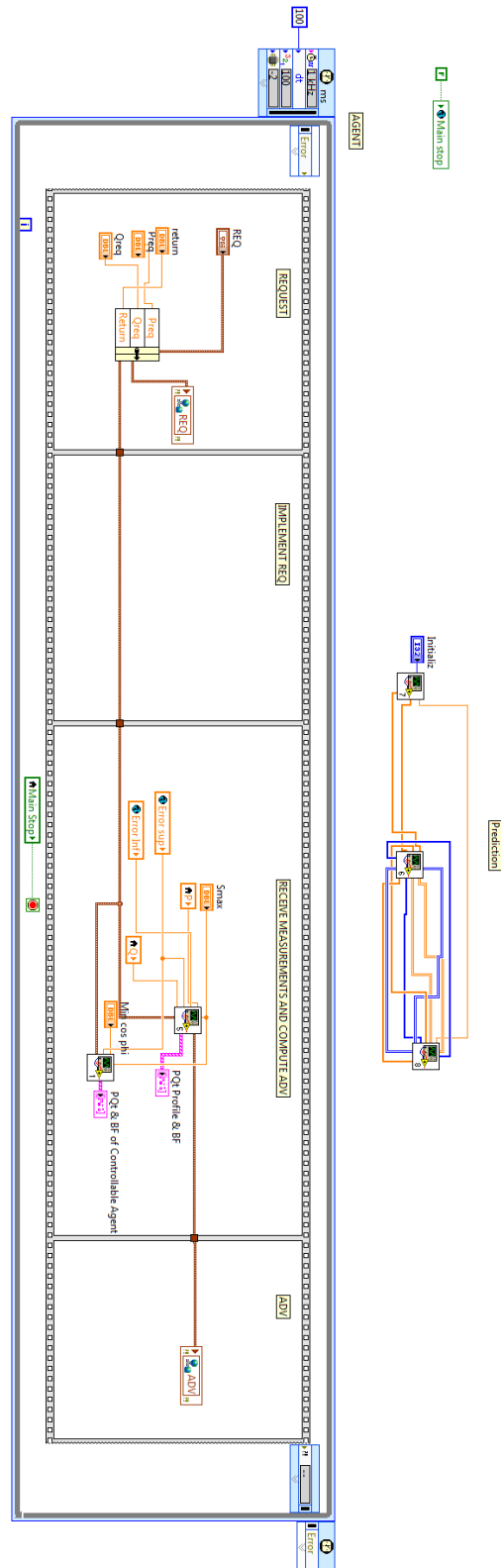
VIs of the Uncontrollable agent

The FPGA VI and Measurements code that have been already presented have been used here so they are not included.

Main for Uncontrolable agent rev3.vi

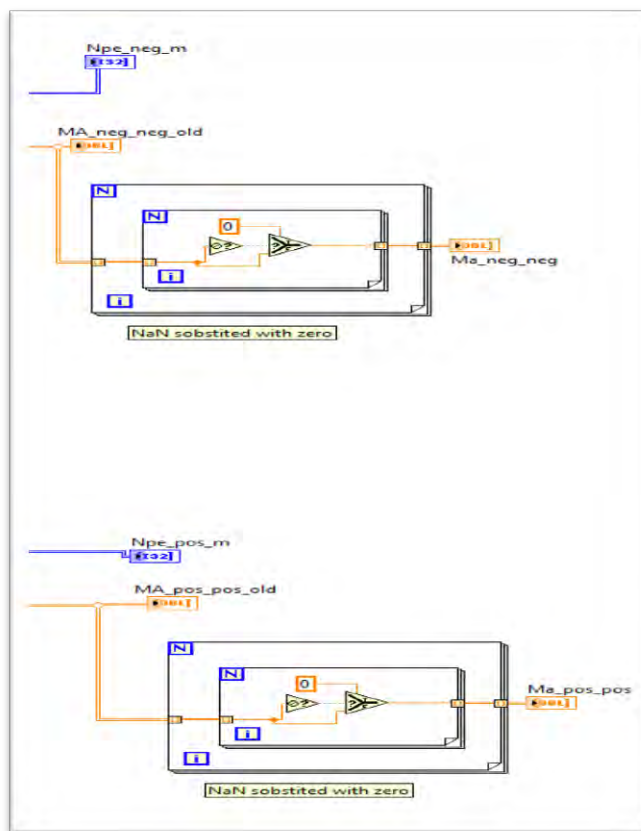
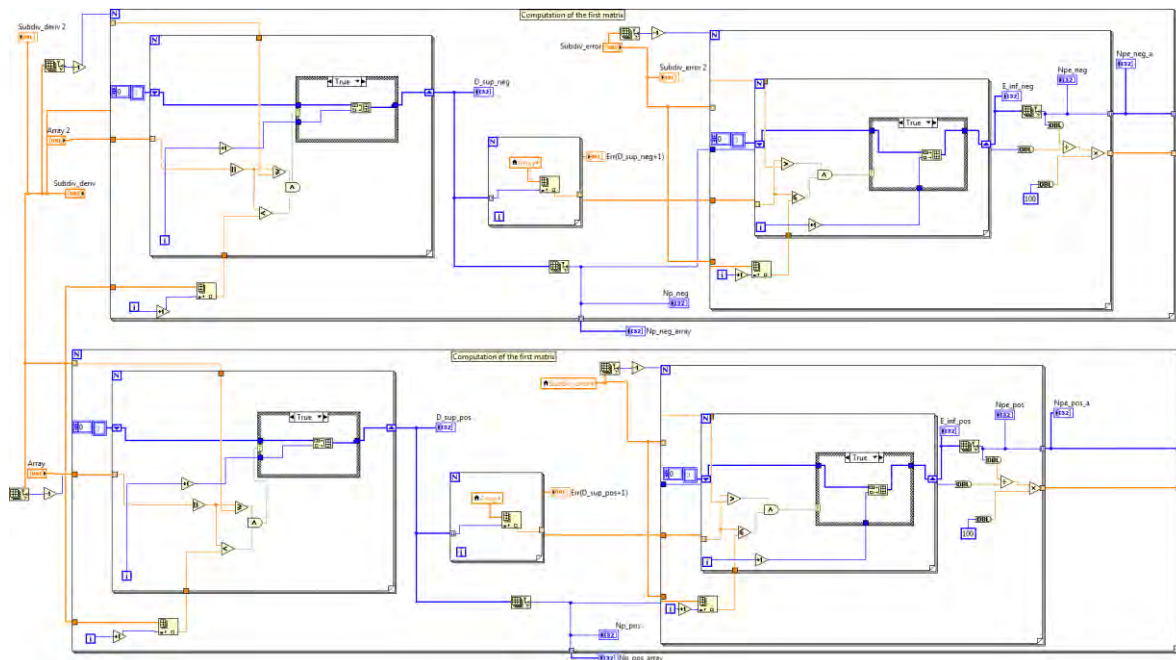
There is a similar part as in the RT rev8 over the code that is shown. That part is where the FPGA is initialize and the measurements are computed, so the only part missing from the RT rev8.vi is the where the TDMS file is created, written and save.

The three VI's that are outside from the time loop is the prediction code cut in pieces for a better implementation and are presented after this first code. Then comes the part where the communication with the converter takes part with the four different frames.



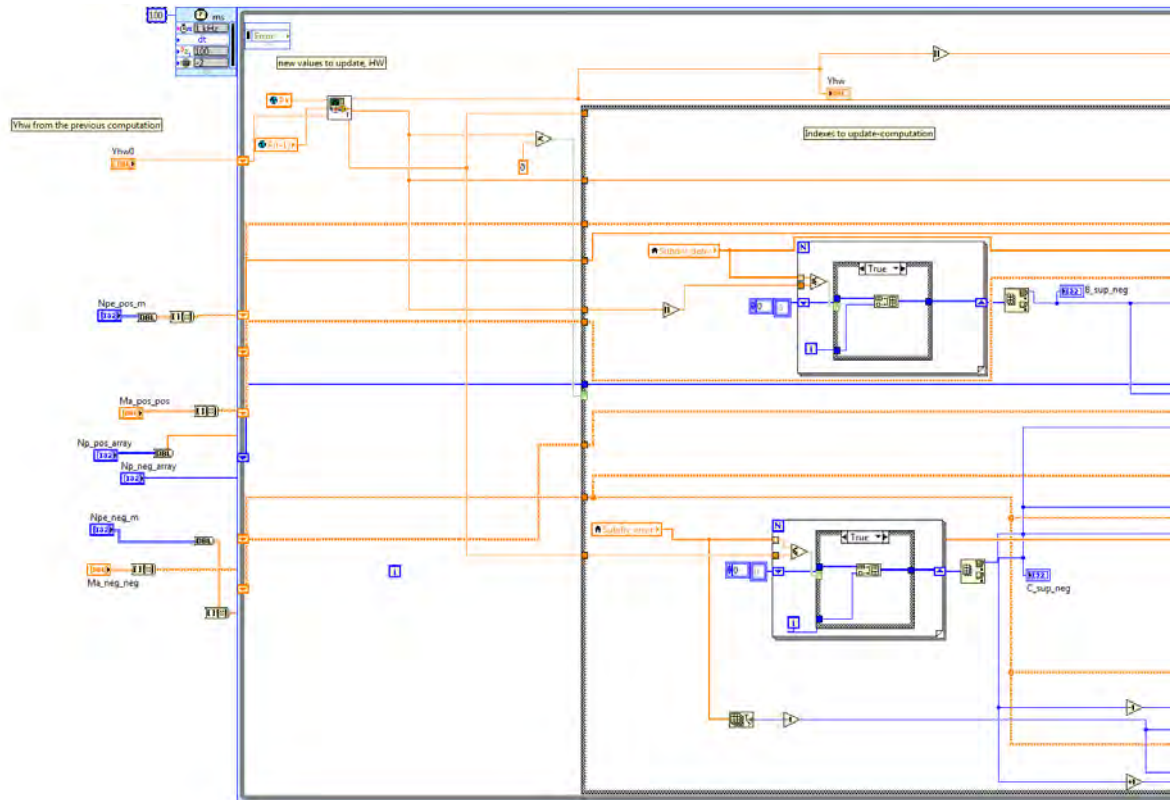


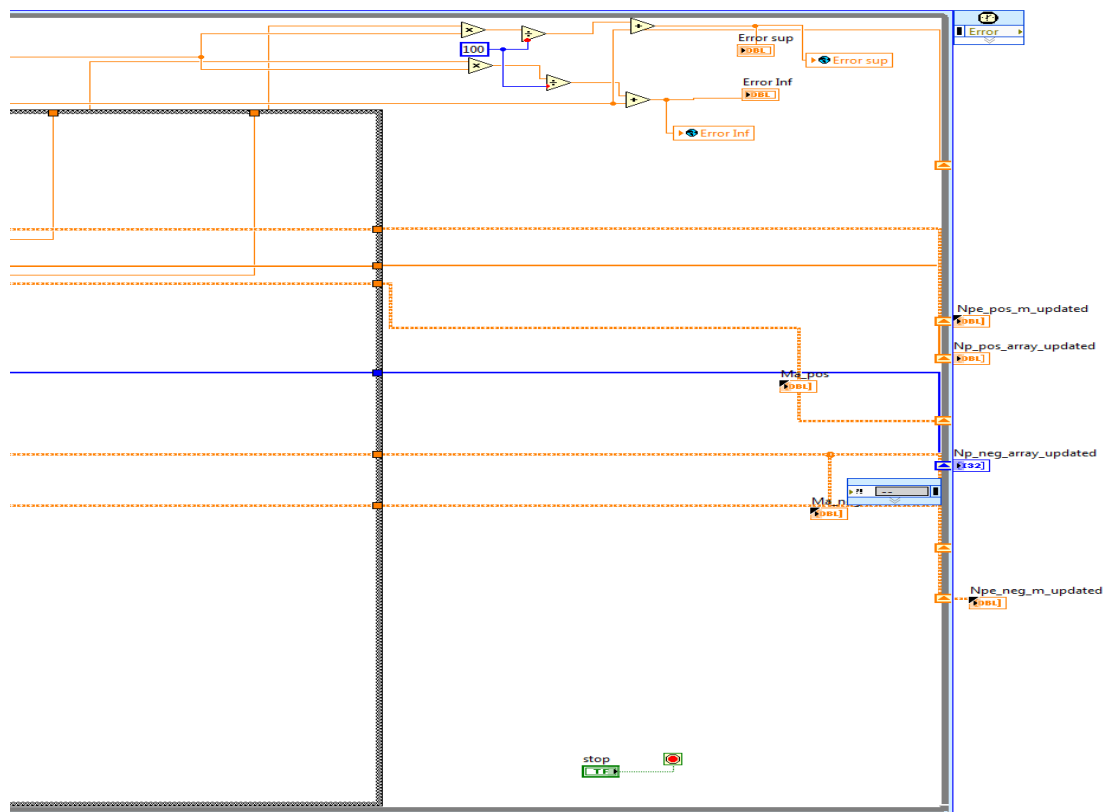
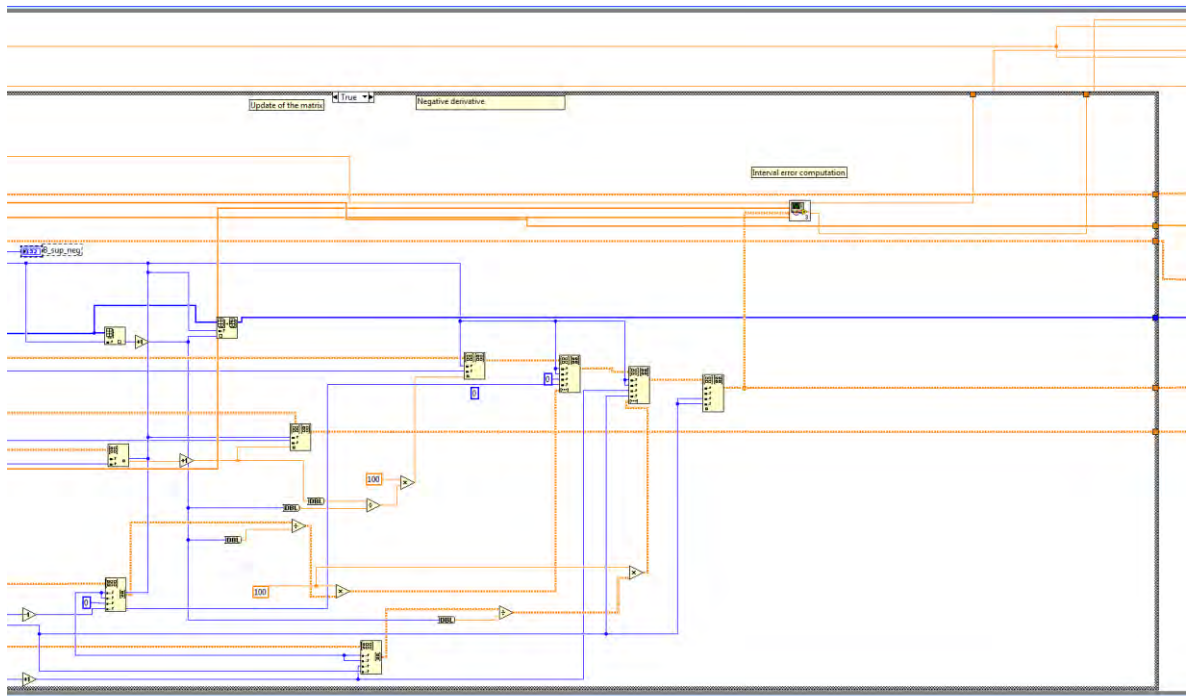
Matrix initiation.vi





Matrix computation.vi

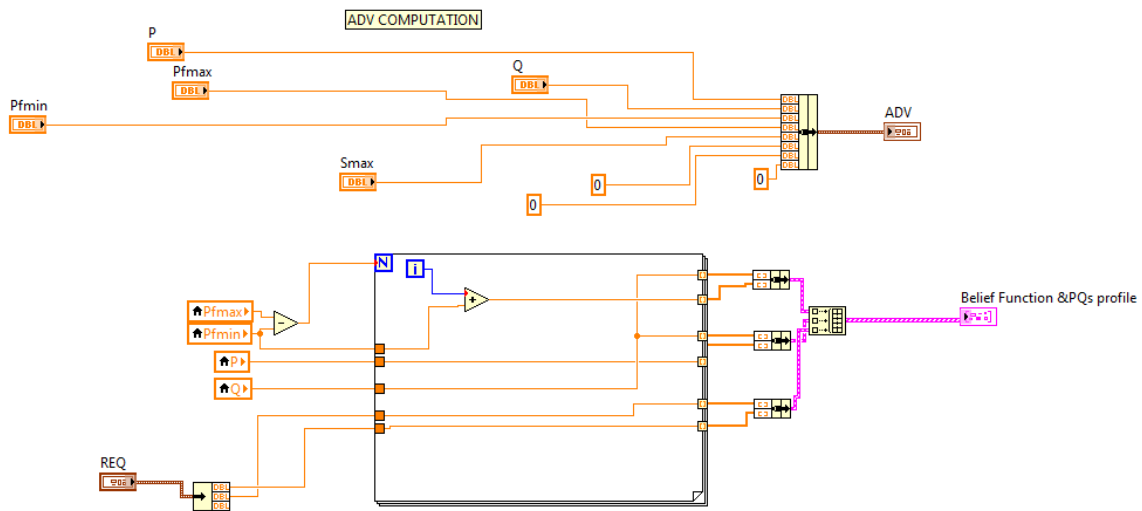






Agent process Uncontrolable agent.vi

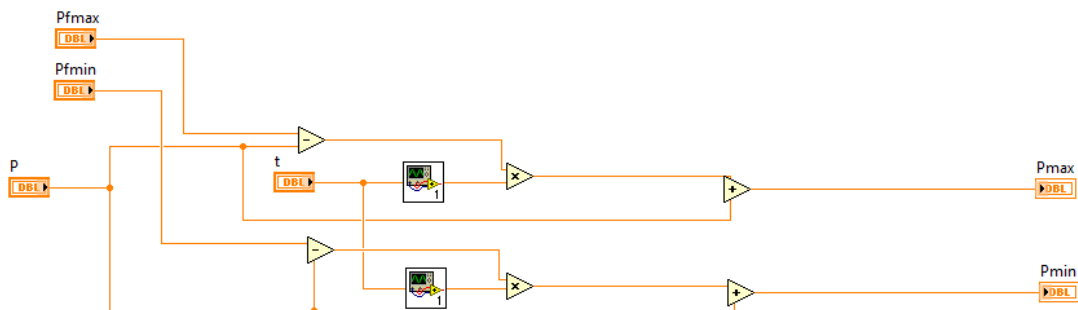
This the lastVI that appears inside the communication with the converter.



Controllable Agent codes

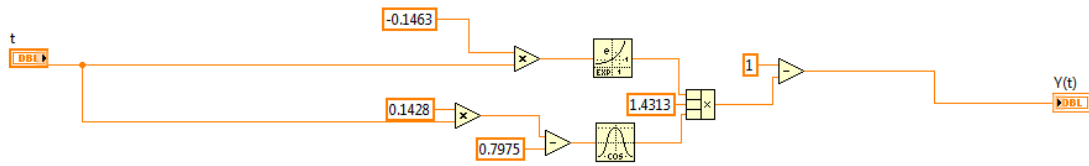
The main panel is not going to be included since it wasn't ready to run and some changes might had to be done.

Pmax Calculation.vi

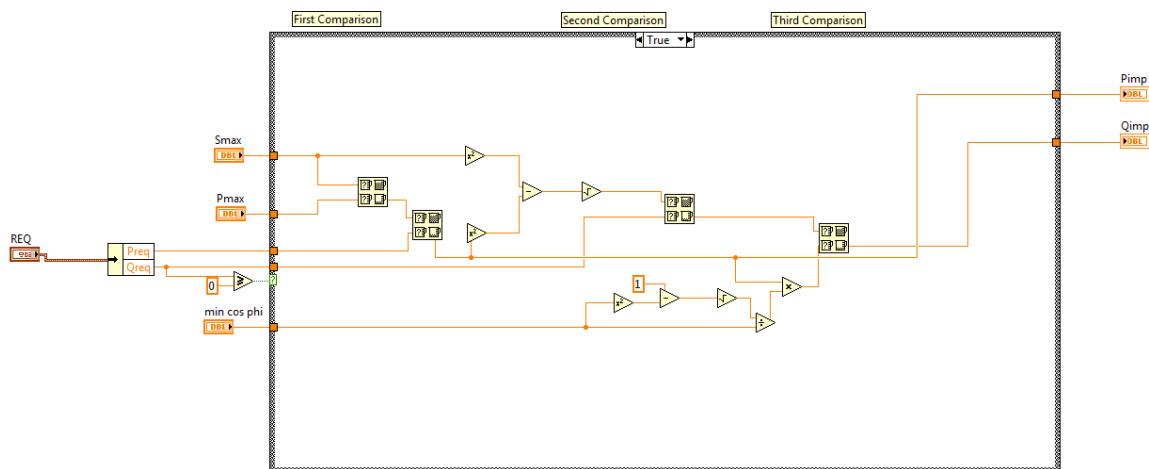


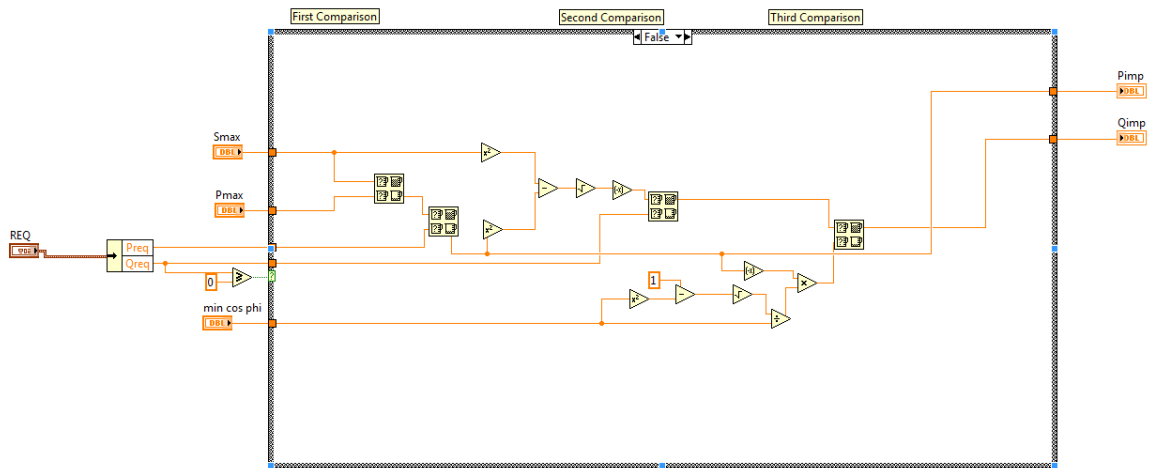


TF(RT).vi



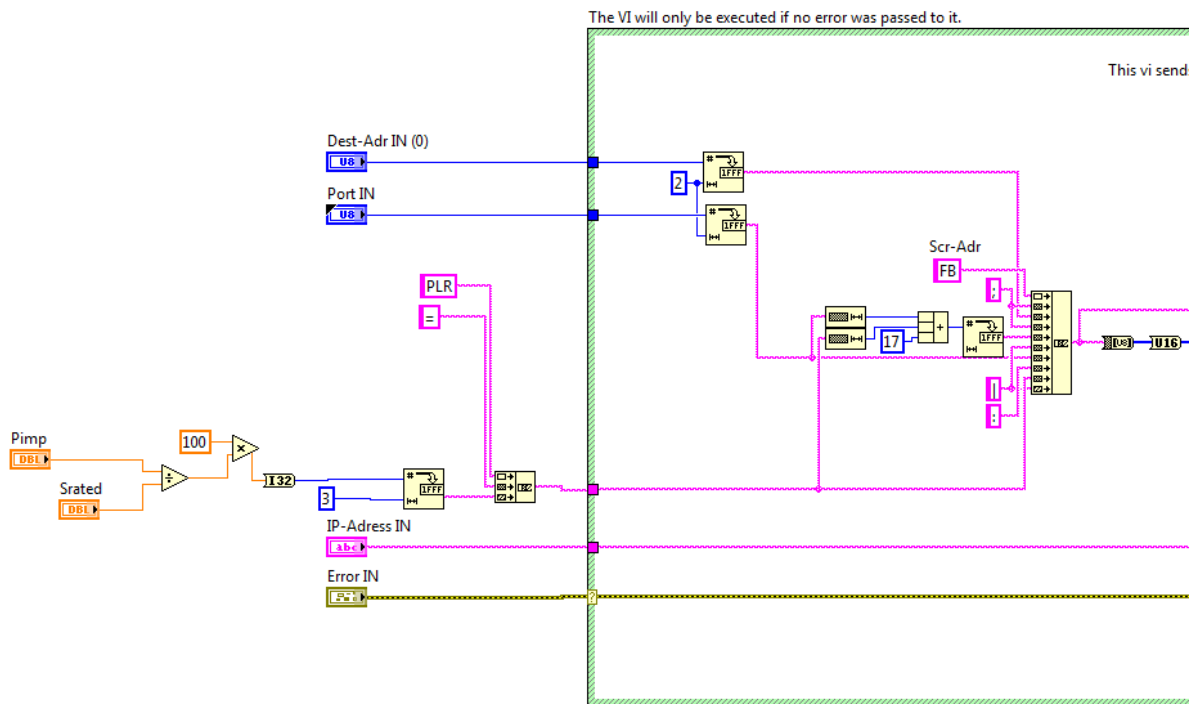
Projection 2.0.vi

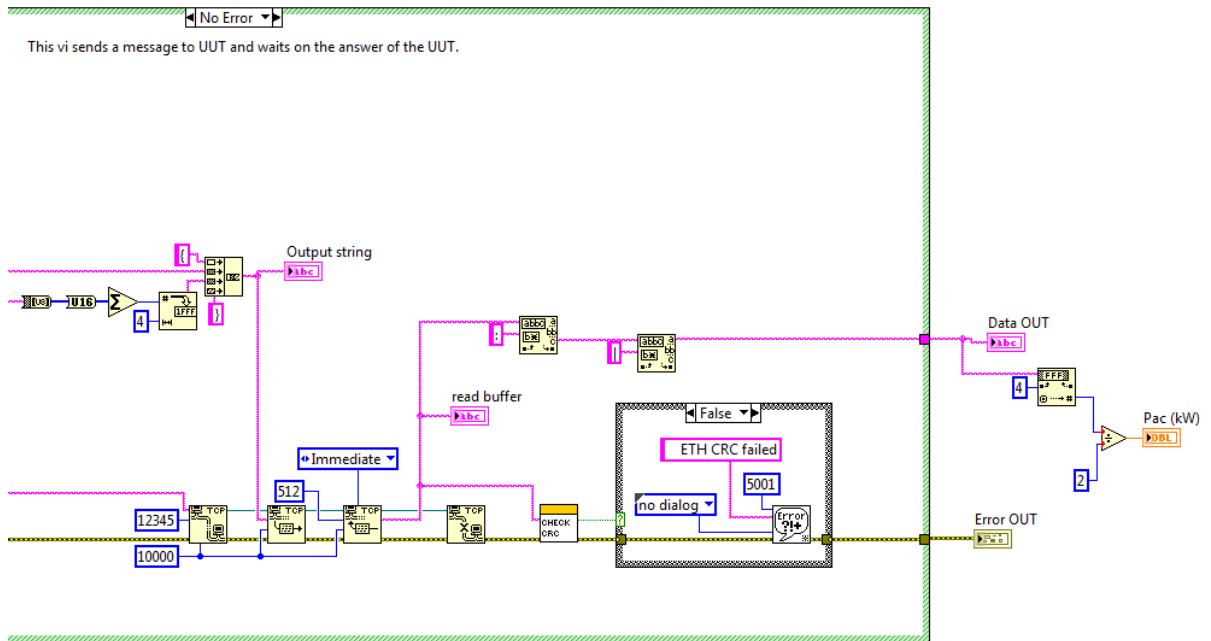






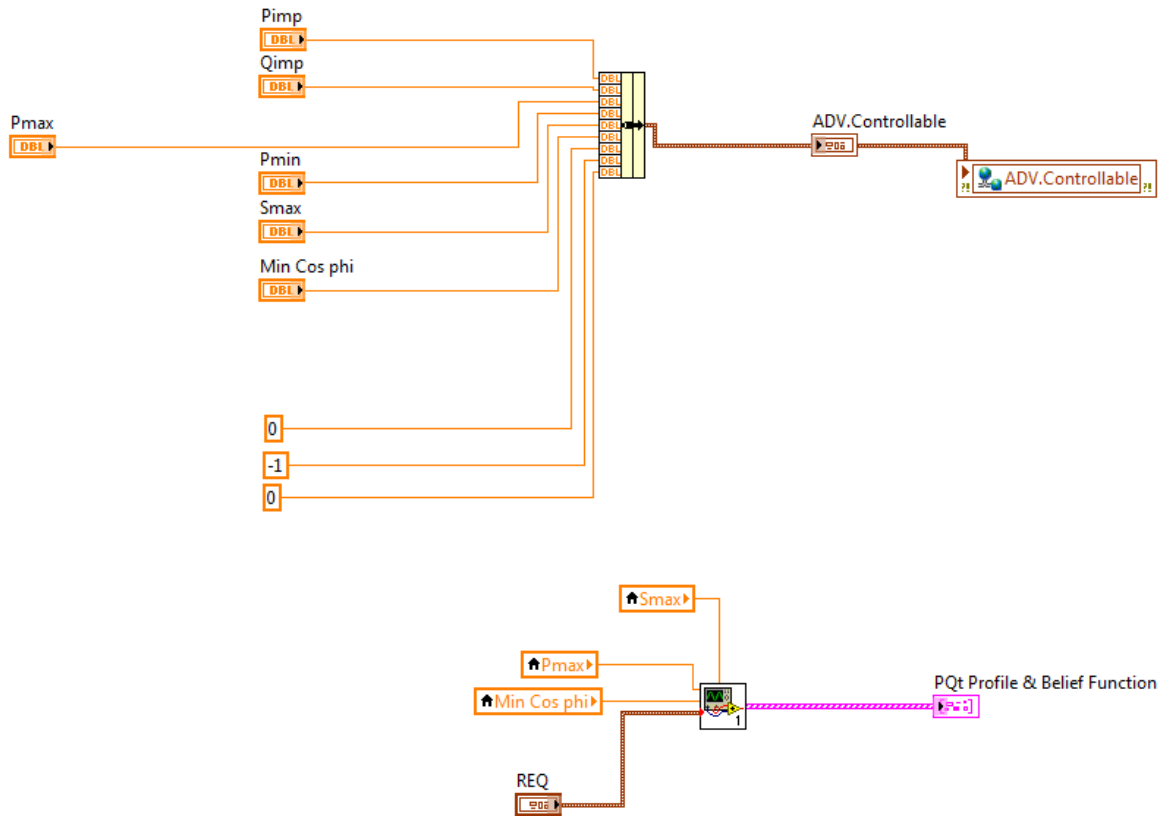
_sencommand.vi





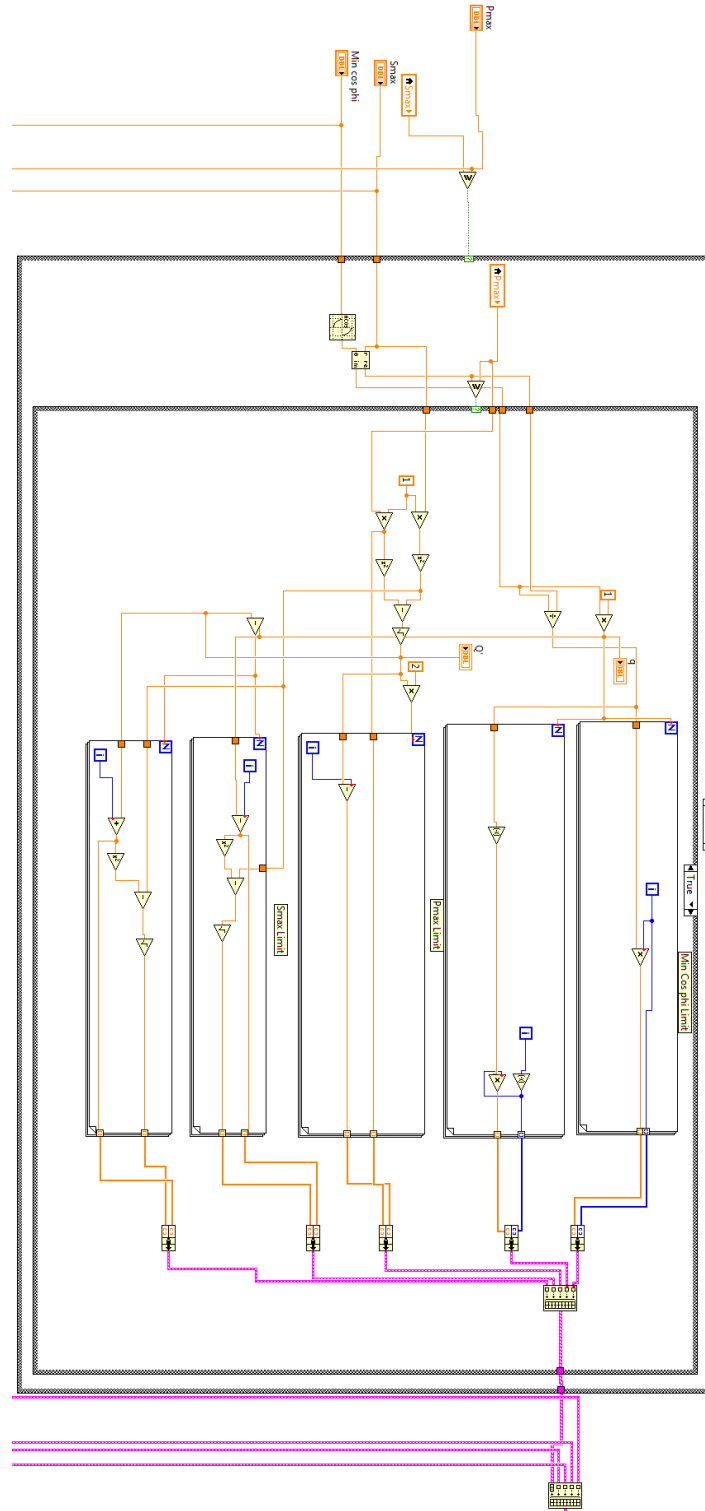


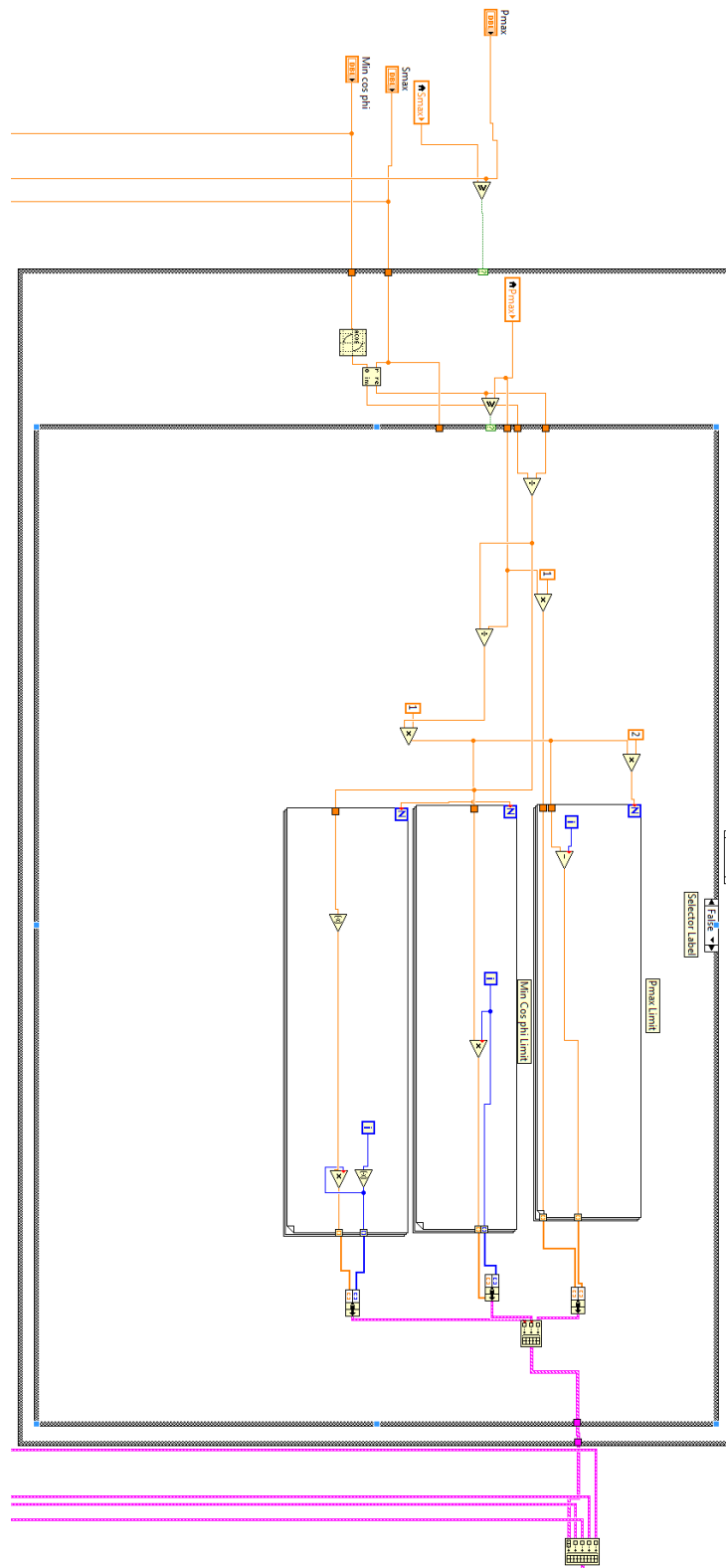
Agent Process for controllable agent.vi

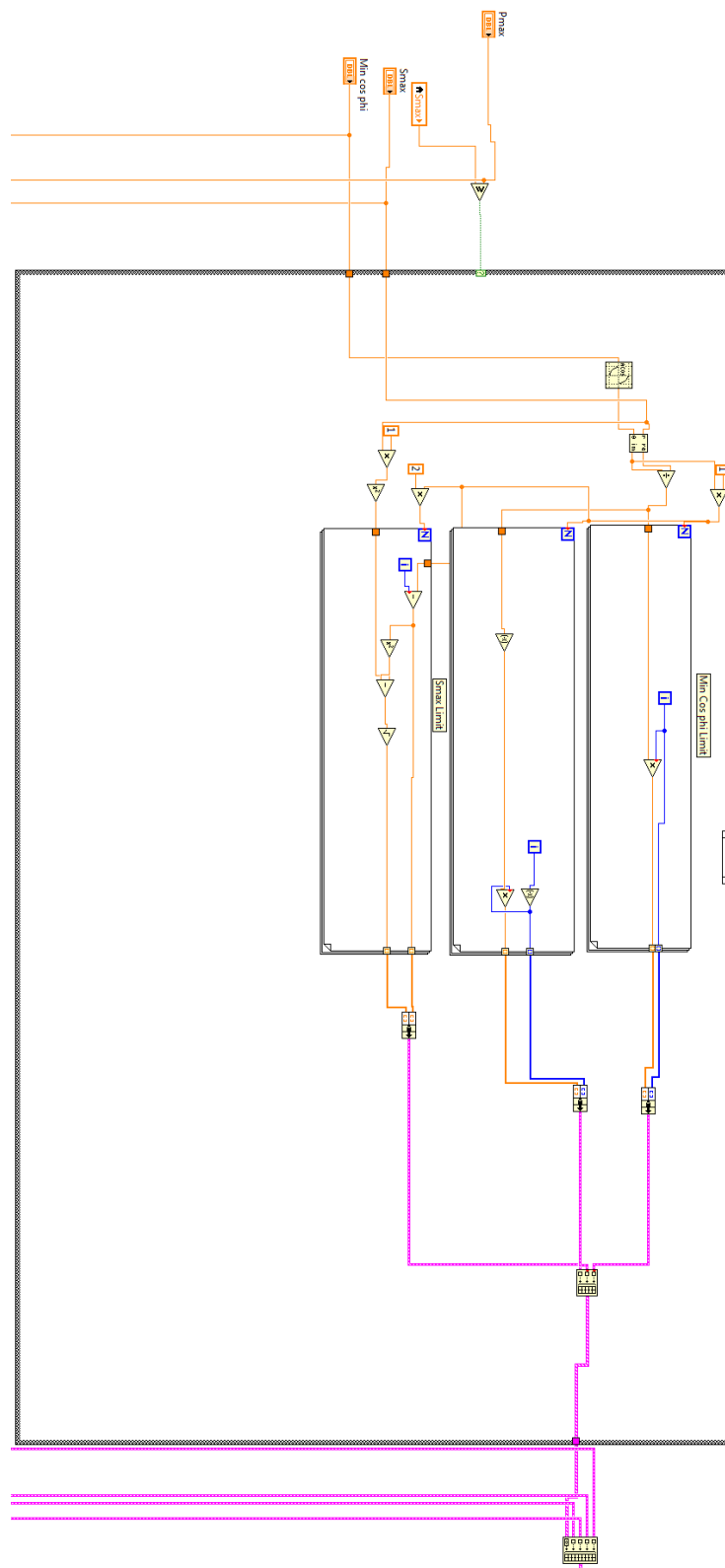


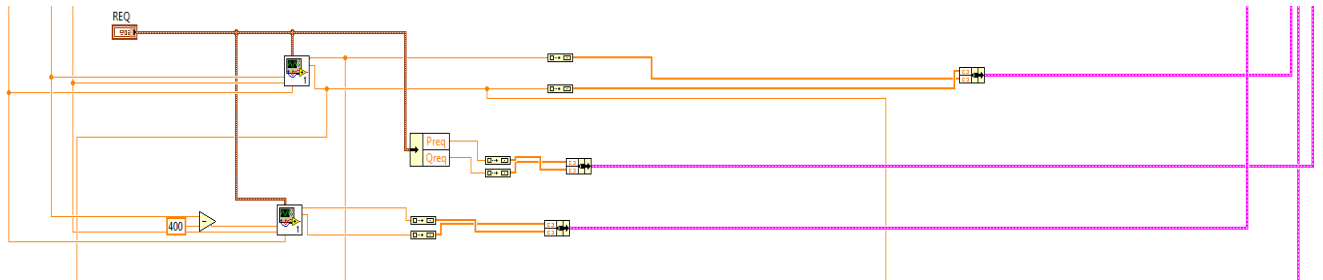


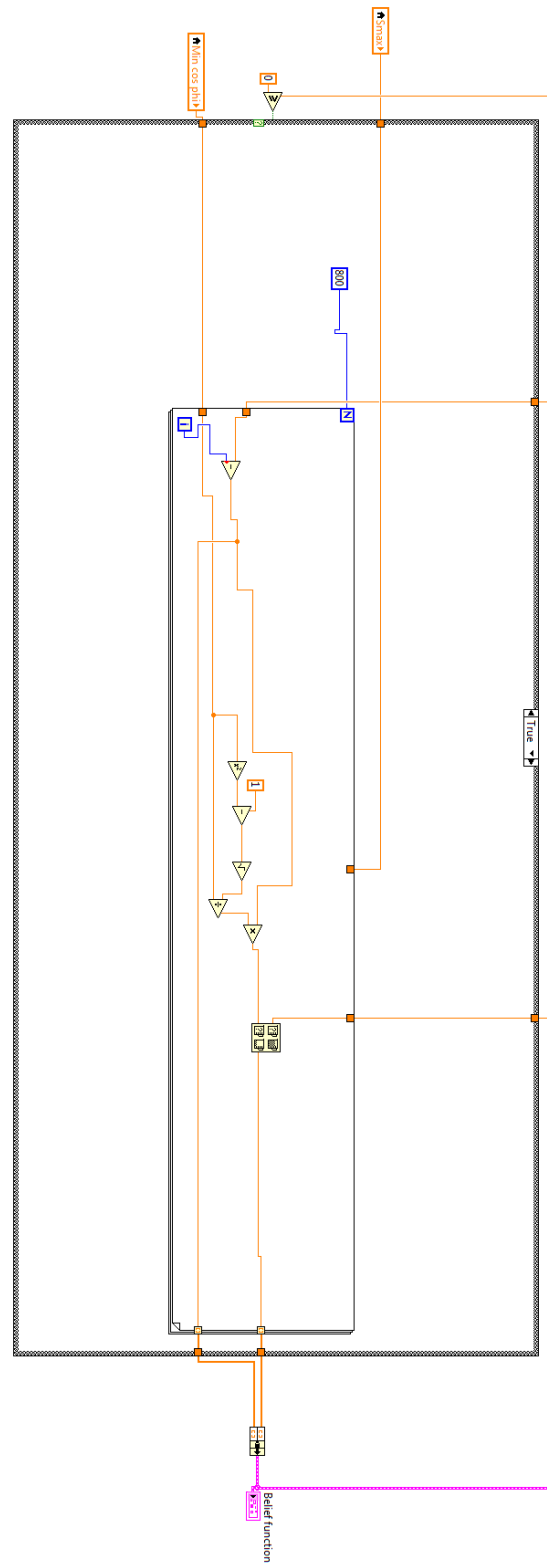
PQ profile CA draw.vi

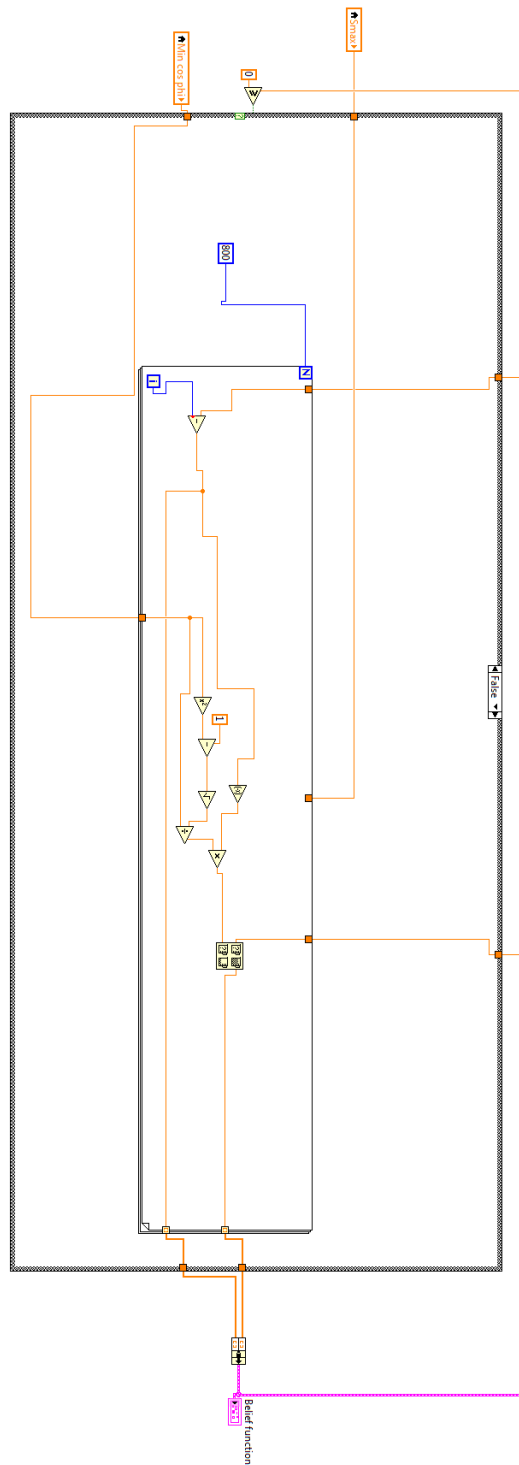








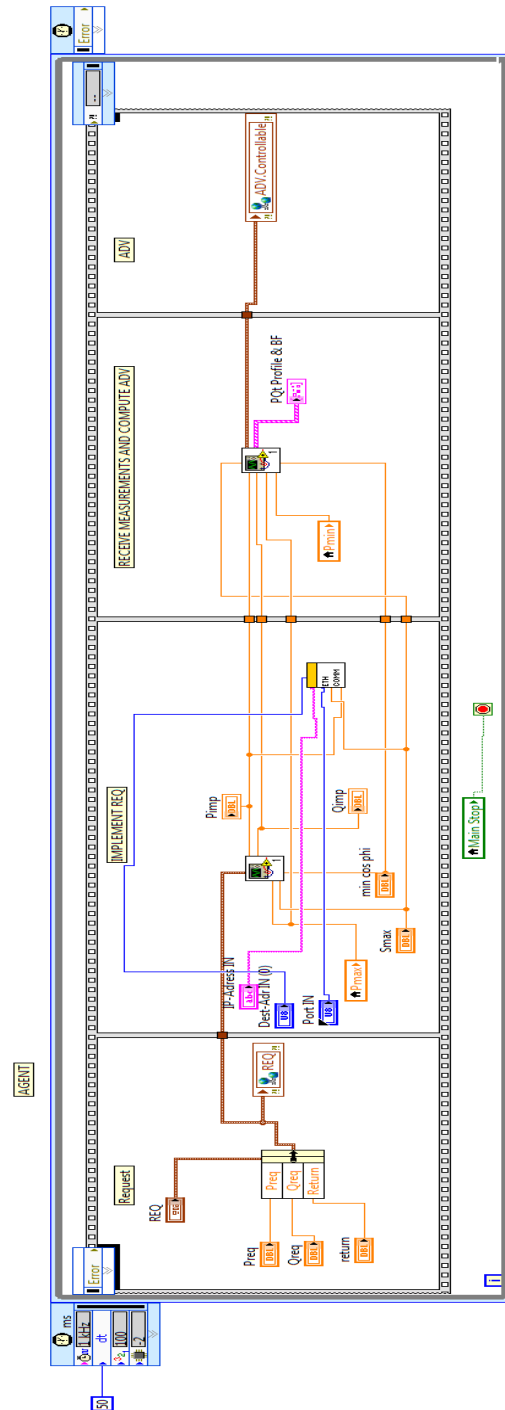




The projection2.0 vi is call inside.



Agent representation code inside the main and simulating the communication with the grid agent.





ⁱ Exploded in the paper: "A compasable method for real-time control of active distribution networks with explicit power setpoints. Part I: method".

ⁱⁱ Exploded in the paper: "A compasable method for real-time control of active distribution networks with explicit power setpoints. Part I: method".

ⁱⁱⁱ The name of the paper is « Model-Free Computation of Ultra Short-Term Prediction Intervals of Solar Irradiance" and it hasn't been published yet.

^{iv} Explained in the paper: " A compasable method for real-time control of active distribution networks with explicit power setpoints. Part I: method".