

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

Automatización del proceso de decisión de un problema multicriterio en redes de energía distribuida

Autor: Álvaro Pajares Barroso

Tutor: Ascensión Zafra Cabeza

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Automatización del proceso de decisión de un problema multicriterio en redes de energía distribuida

Autor:

Álvaro Pajares Barroso

Tutor:

Ascensión Zafra Cabeza

Profesora titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado: Automatización del proceso de decisión de un problema multicriterio en redes de energía distribuida

Autor: Álvaro Pajares Barroso

Tutor: Ascensión Zafra Cabeza

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

Agradecimientos

A mis padres, hermanos y familia.

A todos los que me han apoyado durante esta etapa y me han ayudado a llegar hasta aquí.

A mis amigos.

A mi tutora, Ascensión Zafra Cabeza, que me ha ayudado en todo momento con este trabajo y me ha permitido ajustarlo a mis requerimientos.

La generación de energía actualmente implantada en nuestra sociedad se basa en un modelo que genera una gran cantidad de contaminación que, acompañada del uso prácticamente personal y diario de vehículos contaminantes, incita a la modificación de dicho modelo de generación.

En este trabajo se analizará la viabilidad de un sistema de generación de energía distribuida, donde la generación, en lugar de estar basada en centrales nucleares, turbinas de vapor o cualquier otro tipo de generación contaminante, se basa en energías renovables. Además, el uso de vehículos contaminantes es reemplazado por el uso de vehículos cero emisiones, como son los vehículos eléctricos PEV o los vehículos que funcionan con pila de combustible.

Estos modelos de generación distribuida generan ciertas incertidumbres, como pueden ser el coste de la generación o las dificultades que presentan las energías renovables para su almacenamiento y para satisfacer la demanda doméstica.

Con el fin de evaluar estas incertidumbres, se simulará una red doméstica a partir de programas informáticos y se automatizará el método Promethee, ejecutando una interfaz gráfica que facilita la toma y la modificación de los parámetros más importantes.

Por último, se controlará la alternativa más adecuada mediante un controlador predictivo generalizado.

Abstract

The generation of energy currently implanted in our society is based on a model that generates a large amount of pollution that, accompanied by the personal and daily use of polluting vehicles, encourages the modification of such generation model.

In this work the viability of a distributed energy generation system will be analyzed, where the generation, instead of being based on nuclear power plants, steam turbines or any other type of polluting generation, is based on renewable energies. In addition, the use of polluting vehicles is replaced by the use of zero emission vehicles, such as PEV electric vehicles or vehicles that work with fuel cells.

These models of distributed generation generate certain uncertainties, such as the cost of the generation or the difficulties that renewable energy presents for its storage and to satisfy domestic demand.

In order to evaluate these uncertainties, a Smart Grid will be simulated from computer programs and the Promethee method will be automated, executing a graphical interface that facilitates the taking and modification of the most important parameters.

Finally, the most appropriate alternative will be controlled by a generalized predictive controller.

Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice	xiii
Índice de Figuras	xv
1 Introducción	1
1.1 <i>Motivación</i>	1
1.1.1 Energías renovables	1
1.1.2 Vehículos libres de contaminación	3
1.2 <i>Microgrid resuelta</i>	3
1.3 <i>Objetivos del presente trabajo</i>	4
2 Descripción de la Microgrid	7
2.1 <i>Descripción física de la Microgrid</i>	7
2.1.1 Generador de potencia	7
2.1.2 Electrolizador y pila de combustible	13
2.1.3 Batería	13
2.1.4 Carga electrónica	13
2.2 <i>Descripción de la Microgrid en OpenDSS</i>	15
2.2.1 Herramienta informática OpenDSS	16
2.2.2 Interfaz COM OpenDSS-Matlab	17
2.2.3 Implementación de la Microgrid en OpenDSS	18
3 Método de Decisión Multicriterio	27
3.1 <i>Método Promethee</i>	27
3.2 <i>Obtención de los pesos a partir del método AHP</i>	29
3.2.1 Particularización a nuestro caso	29
4 Implementación de Interfaz Gráfica en Matlab (GUI)	31
4.1 <i>Interfaz gráfica propuesta</i>	31
4.1.1 Generación de alternativas	31
4.1.2 Modificación de los parámetros de evaluación	34
4.1.3 Evaluación y representación de las alternativas	36
4.1.4 Ejecución del modelo predictivo (GPC)	37
5 Control Predictivo Basado en Modelo (MPC)	39
5.1 <i>Estrategia de control</i>	39
5.1.1 Modelo de predicción	40
5.1.2 Función objetivo	43
5.1.3 Obtención de la ley de control	44
5.2 <i>Control Predictivo Generalizado (GPC)</i>	44
5.2.1 Formulación del Control Predictivo Generalizado	44
6 Evaluación Multicriterio de la Microgrid	47
6.1 <i>Evaluación para cada localidad modificando los pesos y las funciones de preferencia</i>	47

6.1.1	Evaluación para la localidad de Écija	47
6.1.2	Evaluación para la localidad de Tarifa	53
6.2	<i>Evaluación para cada localidad del estado inicial de la batería</i>	54
6.2.1	Evaluación para la localidad de Écija	55
6.2.2	Evaluación para la localidad de Tarifa	58
6.3	<i>Evaluación para cada localidad del circuito de hidrógeno</i>	60
6.3.1	Evaluación para la localidad de Écija	60
6.3.2	Evaluación para la localidad de Tarifa	61
6.4	<i>Evaluación conjunta del estado inicial SOC y del circuito de hidrógeno</i>	62
6.4.1	Evaluación para la localidad de Écija	63
6.4.2	Evaluación para la localidad de Tarifa	65
6.5	<i>Evaluación de la potencia nominal de la generación</i>	66
6.5.1	Evaluación para la localidad de Écija	67
6.5.2	Evaluación para la localidad de Tarifa	68
6.6	<i>Aplicación de control predictivo en la Microgrid</i>	70
6.6.1	Control GPC localidad de Écija	72
6.6.2	Control GPC localidad de Tarifa	74
6.7	<i>Comparación de los resultados obtenidos para el control por histéresis y el control predictivo</i>	75
7	Conclusiones	77
7.1	<i>Modelado en OpenDSS y control a partir de Matlab</i>	77
7.2	<i>Método de evaluación Promethee y obtención de pesos AHP</i>	77
7.3	<i>Interfaz gráfica y automatización de la evaluación</i>	77
7.4	<i>Control predictivo GPC</i>	78
Anexo A:	Manejo de la Interfaz Gráfica de Matlab (GUI)	79
	<i>Manejo de una interfaz gráfica</i>	81
	<i>Ejemplo de una interfaz gráfica</i>	82
Anexo B:	Códigos utilizados en Matlab	89
Referencias		135

ÍNDICE DE FIGURAS

Figura 1-1 Potencia eléctrica instalada	1
Figura 1-2 Demanda energética peninsular	2
Figura 1-3 Evolución de precios Energías Renovables	2
Figura 1-4 Microgrid del laboratorio	4
Figura 1-5 Esquema de la Microgrid	4
Figura 2-1 Relación Potencia-Velocidad del viento para el generador eólico	7
Figura 2-2 Datos del viento para Tarifa	8
Figura 2-3 Datos del viento para Écija	8
Figura 2-4 Velocidad del viento en Tarifa	10
Figura 2-5 Velocidad del viento en Écija	10
Figura 2-6 Distribución frecuencial de velocidades para Tarifa y Écija	11
Figura 2-7 Distribución horaria de velocidades para Tarifa y Écija	11
Figura 2-8 Generación eólica horaria para Tarifa y Écija	11
Figura 2-9 Representación de la generación eólica	11
Figura 2-10 Irradiancia horaria para Tarifa y Écija	12
Figura 2-11 Distribución de temperaturas para Tarifa y Écija	13
Figura 2-12 Esquema del circuito de hidrógeno	13
Figura 2-13 Demanda doméstica	14
Figura 2-14 Distribución generación y demanda de energía para distintos casos	15
Figura 2-15 Interfaz gráfica de OpenDSS	16
Figura 2-16 Estructura OpenDSS	17
Figura 2-17 Esquema ortogonal Microgrid para OpenDSS	18
Figura 2-18 Estado de la batería y del tanque de hidrógeno para distintas situaciones	26
Figura 3-1 Funciones de preferencia método Promethee	27
Figura 3-2 Valores de preferencia para el método AHP	29
Figura 3-3 Matriz de comparación AHP	30
Figura 3-4 Matriz normalizada AHP	30
Figura 3-5 Pesos para cada criterio	30
Figura 4-1 Interfaz gráfica propuesta	31
Figura 4-2 Representación generación de las alternativas	33
Figura 4-3 Escritura de datos con éxito	34
Figura 4-4 a) Ventana de edición matriz AHP. b) Valores predeterminados matriz AHP	35
Figura 4-5 Funciones de preferencia predeterminadas	36
Figura 4-6 Ventana emergente criterio SOC	36

Figura 4-7 Ordenación método Promethee con éxito	37
Figura 5-1 Evolución controladores MPC	39
Figura 5-2 Trayectoria de referencia	40
Figura 5-3 Método de control MPC	40
Figura 5-4 a) Modelo por respuesta impulsional. b) Modelo por respuesta ante escalón	41
Figura 5-5 Respuestas libre y forzada	43
Figura 5-6 Trayectorias de referencia para distintos valores de α	44
Figura 6-1 Escritura de datos para Écija para 2018	48
Figura 6-2 Matriz de comparación AHP	48
Figura 6-3 Evaluación de Écija para 2018	49
Figura 6-4 Evaluación de Écija para 2025	49
Figura 6-5 Importancia relativa de los criterios	50
Figura 6-6 Evaluación de Écija con pesos insertados manualmente para 2018	50
Figura 6-7 Evaluación de Écija con pesos insertados manualmente para 2025	50
Figura 6-8 Evaluación de Écija con funciones de preferencia insertadas manualmente para 2018	51
Figura 6-9 Evaluación de Écija con funciones de preferencia insertadas manualmente para 2025	52
Figura 6-10 Modificación valores matriz AHP	52
Figura 6-11 Evaluación de Écija con pesos obtenidos mediante el método AHP modificados para 2018	53
Figura 6-12 Evaluación de Écija con pesos obtenidos mediante método AHP modificados para 2025	53
Figura 6-13 Evaluación de Tarifa para 2018	54
Figura 6-14 Evaluación de Tarifa para 2025	54
Figura 6-15 Evaluación de Écija para distintos estados iniciales de la batería	56
Figura 6-16 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 75%	56
Figura 6-17 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 80%	57
Figura 6-18 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 85%	57
Figura 6-19 Valores de los distintos criterios para distintos estados iniciales de la batería en Écija	57
Figura 6-20 Evaluación de Tarifa para distintos estados iniciales de la batería	58
Figura 6-21 Evolución niveles SOC y MHL en Tarifa para un estado inicial de la batería del 65%	59
Figura 6-22 Evolución niveles SOC y MHL en Tarifa para un estado inicial de la batería del 70%	59
Figura 6-23 Evolución niveles SOC y MHL en Tarifa para un estado inicial de la batería del 75%	60
Figura 6-24 Valores de los distintos criterios para distintos estados iniciales de la batería en Tarifa	60
Figura 6-25 Evaluación de Écija con modificaciones en el circuito de hidrógeno	61
Figura 6-26 Evaluación de Tarifa con modificaciones en el circuito de hidrógeno	62
Figura 6-27 Evaluación general de Écija	63
Figura 6-28 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 70%, un electrolizador de 1 kW y una pila de combustible de 1,5 kW	64
Figura 6-29 Valores de los distintos criterios para la alternativa general más adecuada en Écija	64
Figura 6-30 Evaluación general de Écija con los pesos modificados	65
Figura 6-31 Evaluación general de Tarifa	66

Figura 6-32 Evaluación general de Tarifa con pesos modificados	66
Figura 6-33 Evaluación de Écija para distintas potencias nominales de la generación para 2018	68
Figura 6-34 Evaluación de Écija para distintas potencias nominales de la generación para 2025	68
Figura 6-35 Evaluación de Tarifa para distintas potencias nominales de la generación para 2018	69
Figura 6-36 Evaluación de Tarifa para distintas potencias nominales de la generación para 2018	70
Figura 6-37 Demanda y generación de energía	72
Figura 6-38 a) Estado de la batería (%) y del tanque de hidrógeno (%) al usar el control predictivo.	b) 72
Representación de los flujos de potencia al usar el control predictivo	72
Figura 6-39 Demanda y generación de energía para la localidad de Écija	73
Figura 6-40 a) Estado de la batería (%) y del tanque de hidrógeno (%) para Écija al usar el control predictivo.	
b) Representación de los flujos de potencia para Écija al usar el control predictivo	73
Figura 6-41 Demanda y generación de energía para la localidad de Tarifa	74
Figura 6-42 a) Estado de la batería (%) y del tanque de hidrógeno (%) para Tarifa al usar el control predictivo.	
b) Representación de los flujos de potencia para Tarifa al usar el control predictivo	74
Figura 6-43 Evolución niveles SOC y MHL con control por histéresis para las alternativas óptimas en Écija y Tarifa	75
Figura AA-0-1 Inicio rápido de la Interfaz Gráfica	79
Figura AA-0-2 Ventana de creación de Interfaz Gráfica	80
Figura AA-0-3 Creación ejemplo de Interfaz Gráfica	82
Figura AA-0-4 Ejemplo de Interfaz Gráfica	87

1 INTRODUCCIÓN

Actualmente, pese a la concienciación de cada vez más personas acerca de los efectos positivos de las energías renovables y de vehículos libres de contaminación, las energías contaminantes, como, por ejemplo, las provenientes de combustibles fósiles, siguen siendo la principal fuente de generación de nuestro país, y los vehículos Diesel y gasolina no dan prácticamente cabida a vehículos cero emisiones como pueden ser los eléctricos o vehículos con pila de combustible.

1.1 Motivación

1.1.1 Energías renovables

En primer lugar, se van a visualizar una serie de datos que ponen de manifiesto el excesivo uso de las energías contaminantes.

En cuanto a la generación de energía, en el siguiente gráfico se puede ver cómo la potencia eléctrica instalada en España se compone de un 51% proveniente de energías renovables (principalmente de la energía eólica y de la hidráulica).



Figura 1-1 Potencia eléctrica instalada

Pese a este dato, la producción de energía proveniente de energías renovables indica que, en 2017, tan solo el 33,3% de la energía fue producida a partir de fuentes renovables, habiendo sufrido un descenso del 7,5% respecto al año anterior. Cabe decir que este dato evidencia que la generación hidráulica ha proporcionado mucha menos energía de la que es capaz, debido principalmente a la sequía que tuvo lugar en este año.



Figura 1-2 Demanda energética peninsular

Estos datos ponen de manifiesto la dependencia de nuestra sociedad de las fuentes generadoras contaminantes, y por tanto de las grandes compañías de electricidad.

Por un lado, es obvio que no se puede depender del hecho de que haga sol, viento o que llueva en nuestro país para poder hacer llegar la electricidad a nuestros hogares.

Sin embargo, también es cierto que existen otras alternativas desconocidas para la gran mayoría de la sociedad, como por ejemplo la generación de energía distribuida, que se basa en la descentralización de la generación, generando la energía en el mismo lugar donde va a ser consumida.

El problema de la generación de energía distribuida está principalmente en la poca conciencia acerca de su viabilidad, la dificultad en el almacenamiento de energías renovables y el coste de tales energías. Tanto la viabilidad como el almacenamiento de estas energías serán tratados en apartados posteriores, mientras que, a continuación, se pueden ver ciertos datos acerca del coste de estas energías.

La siguiente tabla muestra una visión global de los costes de la producción de la energía solar y la eólica en dólares/kWh producido.

TABLE ES 1: GLOBAL WEIGHTED AVERAGE SOLAR AND WIND POWER INVESTMENT COSTS, CAPACITY FACTORS AND LCOEs, 2015 AND 2025

	Global weighted average data								
	Investment costs (2015 USD/kW)		Percent change	Capacity factor		Percent change ²	LCOE (2015 USD/kWh)		Percent change
	2015	2025		2015	2025		2015	2025	
Solar PV	1 810	790	-57%	18%	19%	8%	0.13	0.06	-59%
CSP (PTC: parabolic trough collector)	5 550	3 700	-33%	41%	45%	8.4%	0.15	0.09	-37%
CSP (ST: solar tower)	5 700	3 600	-37%	46%	49%	7.6%	0.15	0.08	-43%
Onshore wind	1 560	1 370	-12%	27%	30%	11%	0.07	0.05	-26%
Offshore wind	4 650	3 950	-15%	43%	45%	4%	0.18	0.12	-35%

Figura 1-3 Evolución de precios Energías Renovables

Se puede observar cómo se prevé un gran descenso en los próximos años en el coste de ambas energías. La energía solar fotovoltaica pasará a costar alrededor de 0,06\$/kWh y la eólica terrestre 0,05\$/kWh para 2025. Ambas energías tienen previsto un gran descenso en este aspecto, pero llama la atención la gran disminución del coste de la energía solar fotovoltaica (de un 59%, frente a un 26% en el caso de la eólica).

Estos datos son meramente intuitivos y recogidos de plantas de generación cuya capacidad es cercana o superior a 1MW, pero hacen ver que este coste es incluso inferior al que llega a nuestras casas, donde el precio que se paga por el kWh ronda 0,13€/kWh. Lógicamente, para una supuesta instalación de una planta fotovoltaica o eólica para el abastecimiento de un hogar, el precio por kWh será superior.

1.1.2 Vehículos libres de contaminación

El otro gran causante de la contaminación presente es como se ha dicho antes el uso de vehículos no limpios, tanto Diesel como gasolina.

Este hecho se debe principalmente al desconocimiento por parte del usuario y a la dificultad de encontrar gasolineras eléctricas o puntos de recarga para coches eléctricos. En cuanto a los coches por pila de combustible, este hecho es todavía más acentuado, ya que es muy poco corriente encontrar puntos de recarga de este tipo de vehículos. De todos modos, la sociedad todavía tiene en mente que el precio de este tipo de vehículos es demasiado elevado y que no son asequibles para gran parte de los ciudadanos. Sin embargo, este tipo de vehículos han experimentado un elevado descenso en su precio lo que, sumado a la disminución del precio por consumo y mantenimiento, hacen que la compra de un vehículo de este tipo sea una inversión a medio-largo plazo.

1.2 Microgrid resuelta

En este TFG se ha trabajado con la Microgrid presente en el laboratorio del departamento de Ingeniería de Sistemas y Automática de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla. Con ella se busca demostrar la viabilidad del uso de hidrógeno y baterías como recursos para el almacenamiento de energías renovables, donde la generación de energía no es constante y se pueden producir momentos de exceso de energía y de defecto de energía. A partir del almacenamiento en tanques de hidrógeno se trata por tanto de intentar hacer lo más estable posible el funcionamiento de tales energías.

La Microgrid se compone de:

- Fuente de potencia electrónica de 6kW. A partir de este generador vamos a recrear el funcionamiento de una planta fotovoltaica y de una eólica.
- Electrolizador de 3kW.
- Tanque de hidrógeno de 7Nm³ a 5bar de presión.
- Pila de combustible de 1,5kW.
- Carga electrónica de 2,5kW. A partir de esta carga vamos a simular la demanda de energía que corresponda, suponiendo una demanda doméstica y la demanda correspondiente a la carga de un coche eléctrico, híbrido o por pila de combustible.

A continuación, se muestra la distribución real de la Microgrid y un esquema de la misma:

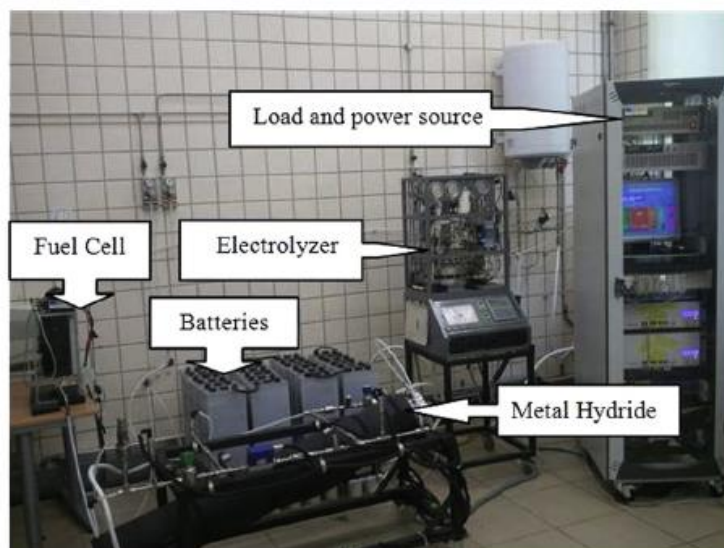


Figura 1-4 Microgrid del laboratorio

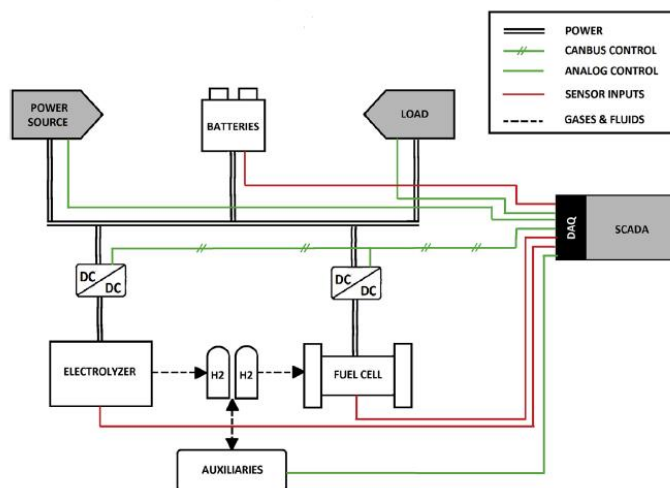


Figura 1-5 Esquema de la Microgrid

1.3 Objetivos del presente trabajo

En este trabajo se busca evaluar las distintas alternativas que puede presentar la Microgrid a partir de unos criterios determinados: satisfacción de la demanda, pérdidas, coste de la generación e impacto medioambiental.

Para ello se establecen los siguientes objetivos:

1. Descripción de la Microgrid ajustándose a nuestras condiciones en el programa OpenDSS.
2. Posibilidad de modificar los valores de potencia nominal de los distintos equipos de la Microgrid y la evaluación directa a partir del método de decisión Promethee.
3. Creación de una interfaz gráfica que permita la interacción sencilla del usuario con los distintos parámetros que definen la toma de decisiones.
4. Implementación del control predictivo.

Con la idea de cumplir nuestros objetivos se van a seguir los siguientes capítulos:

1. Introducción. Es el capítulo actual y se introduce la temática del trabajo.

2. Descripción de la Microgrid. Se describe la Microgrid resuelta y se explica el procedimiento seguido para la obtención de los datos que la definen. Además, se define en OpenDSS para evaluar su funcionamiento.
3. Método de decisión multicriterio. Se introduce brevemente el método de decisión Promethee y la obtención de los pesos por el método AHP.
4. Control predictivo basado en modelo (MPC). Se explica el enfoque y el modelo usado en este tipo de control, particularizando el control predictivo generalizado.
5. Uso de la interfaz gráfica de Matlab. Se explica cómo crear y manejar una interfaz gráfica en Matlab. Se recrea un ejemplo sencillo de creación de una interfaz y se expone la interfaz gráfica propuesta.
6. Evaluación de las alternativas y conclusiones.

2 DESCRIPCIÓN DE LA MICROGRID

En este apartado se describirán los componentes de la Microgrid y la fuente de obtención de los datos necesarios para su modelado. En primer lugar, se explicará cada componente y su función en la Microgrid, para luego explicar su descripción en el programa OpenDSS para su simulación.

2.1. Descripción física de la Microgrid

Como se ha comentado antes, la Microgrid se compone de un generador de potencia electrónico, un electrolizador, un depósito de hidrógeno, una pila de combustible, una batería y una carga electrónica.

El generador de potencia electrónico va a representar tanto la generación fotovoltaica como la generación eólica, y la carga electrónica representará la demanda doméstica que se debe satisfacer y distintos tipos de vehículos recargables (vehículo eléctrico PEV, vehículo híbrido y vehículo por pila de combustible).

Como dicha Microgrid es de algún modo ficticia, es decir, se encuentra en un laboratorio y no dispone de una situación geográfica para evaluar los datos meteorológicos, se evaluarán dos localidades: Écija y Tarifa. Se han elegido estas localidades ya que, en principio, en Tarifa la generación eólica debe de ser más favorable y en Écija la más favorable debería ser la fotovoltaica.

En el presente apartado se va a explicar el funcionamiento de cada uno de los componentes mencionados.

2.1.1 Generador de potencia

2.1.1.1 Generador eólico

A la hora de recrear el generador eólico, la idea es simular un generador que tenga una potencia máxima cercana a la máxima recreable por el generador electrónico, es decir, cercana a 6kW. Para ello, se van a tomar datos del generador eólico presente en el INTA del Arenosillo, en Huelva, que dispone de una potencia nominal de 5kW. La relación Potencia (kW)-Velocidad del viento (m/s) de dicho generador se ve representada por la siguiente curva, la cual se extrapolará para generalizarla a una potencia nominal cualquiera:

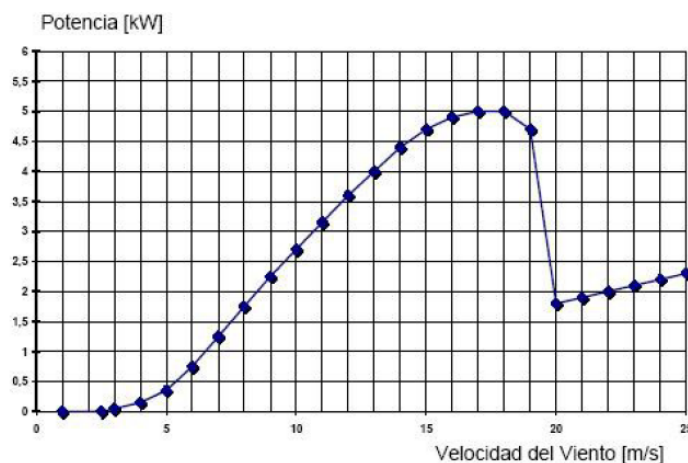


Figura 2-1 Relación Potencia-Velocidad del viento para el generador eólico

Esta curva queda representada en Matlab a partir de la siguiente función para cualquier potencia nominal:

```
function mult_out=Potencia_Eolica(v_hora)
```

```
v_ref=0:20;
```

```
mult_outref=[0,0,0,0,0.2,0.4,0.75,1.25,1.75,2.25,2.7,3.15,3.6,4,4.4,4.75,4.9,5,5
```

```
,4.65,1.8]/5;
p=polyfit(v_ref,mult_outref,4);
mult_out=polyval(p,v_hora);
end
```

Esta función tomará como parámetro de entrada la velocidad del viento y devolverá un multiplicador que relaciona la potencia nominal con la potencia real de salida.

$$mult = \frac{P_{out}}{P_{nominal}}$$

La duda que se presenta ahora es qué velocidad del viento se toma para cada hora del día. Para ello, se va a seguir el procedimiento detallado en el documento de referencia [4].

En primer lugar, se recogen los datos proporcionados por el IDEA para las dos localidades a estudiar, Tarifa y Écija:

Distribución por direcciones a 80m.

Coordenadas UTM(m): 255942,4000760

Dirección	Frecuencia (%)	Velocidad (m/s)	Potencia (%)	Weibull C (m/s)	Weibull K
N	2.21	6.335	0.86	7.013	2.348
NNE	0.91	4.832	0.18	5.35	1.988
NE	0.86	5.058	0.2	5.562	1.902
ENE	4.07	8.387	3.43	9.268	2.585
E	29.04	11.199	63.3	12.945	2.843
ESE	6.02	7.918	5.12	9	2.247
SE	1.72	5.066	0.42	5.969	2.309
SSE	1.42	4.565	0.24	5.387	2.44
S	1.73	6.262	1	7.179	1.726
SSW	3.81	7.038	2.4	7.974	2.098
SW	6.6	7.24	4.55	8.39	2.248
WSW	9.16	6.634	4.72	7.872	2.606
W	13.26	6.795	5.91	7.715	3.095
WNW	8.19	6.763	3.79	7.641	2.679
NW	5.99	6.464	2.04	7.096	3.217
NNW	5.01	6.488	1.82	7.087	2.771

Figura 2-2 Datos del viento para Tarifa

Distribución por direcciones a 80m.

Coordenadas UTM(m): 298443,4163262

Dirección	Frecuencia (%)	Velocidad (m/s)	Potencia (%)	Weibull C (m/s)	Weibull K
N	3.95	5.559	4.04	6.108	1.637
NNE	4.6	5.348	3.28	5.768	1.877
NE	7.66	5.5	5.2	5.941	2.144
ENE	14.97	5.876	13	6.578	2.293
E	6.53	4.379	2.31	4.797	2.159
ESE	3.59	3.956	1.12	4.392	1.889
SE	3.71	5.441	3	5.909	1.798
SSE	5.21	6.768	8.02	7.401	1.844
S	3.67	6.04	4.08	6.488	1.746
SSW	4.46	6.868	7.56	7.465	1.747
SW	11.18	7.197	17.22	7.812	2.148
WSW	15.45	6.786	21.43	7.51	2.117
W	6.31	5.318	4.82	5.826	1.819
WNW	2.95	4.549	1.56	4.938	1.659
NW	2.42	4.097	0.89	4.372	1.661
NNW	3.34	5.136	2.48	5.504	1.647

Figura 2-3 Datos del viento para Écija

En ambas tablas se representa la frecuencia con que ciertos valores de la velocidad del viento tienen lugar y su dirección. Por tanto, podemos llegar a la velocidad más frecuente: 11,199m/s para Tarifa y 6,786m/s para Écija. Este va a ser el valor promedio ρ . Cabe mencionar que, para el caso del generador eólico que se está evaluando, no se tendrá en cuenta la dirección del viento, ya que se cuenta con auto posicionamiento mediante veleta.

El siguiente paso será imponer una distribución que represente la frecuencia con que la velocidad con que el viento aparece. En situaciones de generación eólica, se suele hacer uso de la distribución Weibull, para lo que necesitamos el valor promedio de la velocidad, el factor de asimetría K y el factor de escala C.

Los datos presentes en el IDEA son para una altura de medición de 80m. Como nuestro generador tiene una altura de 20m, hay que ajustar la velocidad promedio a esta altura. De acuerdo con el documento previamente citado, la relación que mantiene la velocidad del viento con la altura es la siguiente:

$$v(z) = v_a \cdot \left[\frac{(z_R - z_0)}{z_a} \right]^\alpha$$

donde,

- $v(z)$: velocidad media del viento a la altura deseada.
- z_R : altura objeto de estudio.
- v_a : velocidad a la altura de referencia (a 80m en nuestro caso).
- z_a : altura de referencia (80m en nuestro caso).
- z_0 : corrección por obstáculos en el terreno.
- α : parámetro dependiente de la rugosidad superficial y de la estabilidad atmosférica media.

Dado que este caso de estudio no determina un lugar concreto dentro de la localidad, voy a suponer que no existen obstáculos en el terreno y, por tanto, se puede decir que $z_0 = 0$. En cuanto al parámetro α , la siguiente correlación facilita su cálculo.

$$\alpha = \frac{0.37 - 0.088 \cdot \ln(v_a)}{1 - 0.088 \cdot \ln(0.1z_a)}$$

Para poder hacer uso de esta correlación, se deben cumplir las siguientes hipótesis:

- 1) La rugosidad del terreno corresponde a un obstáculo densamente distribuido sobre el mismo.
- 2) Se debe cumplir que $h_c/z_R \leq 0.75$.

Al tratarse de un lugar sin determinar y haber hecho la suposición de que no existen obstáculos cercanos, ambas hipótesis se dan por cumplidas.

Al usar dicha correlación, llegamos a un valor de $\alpha = 0.1927$ para Tarifa y de $\alpha = 0.2466$ para Écija.

Ahora sí, al usar la ecuación anterior, obtenemos una velocidad promedio de 8.5743m/s para Tarifa y de 5.1755m/s para Écija para la altura de estudio de 20m.

Para el cálculo de los parámetros de Weibull, se llega a un valor del factor de escala de $C=8.5743/0.88623=9.675$ m/s para Tarifa y $C=5.1755/0.88623=5.84$ para Écija. En cuanto al factor de asimetría k, se toma el valor $k=2$, considerado óptimo en este tipo de distribuciones.

A partir de ambos parámetros, la distribución de Weibull se dispone a partir de la siguiente función:

$$f(v) = \frac{k}{C} \cdot \left[\frac{v}{C} \right]^{k-1} \cdot \exp\left[-\left(\frac{v}{C} \right)^k \right]$$

Al representar dicha función en Matlab, podremos sacar la frecuencia para cada valor de velocidad de viento.

Las distribuciones de Weibull de frecuencia para ambas ciudades son las siguientes:

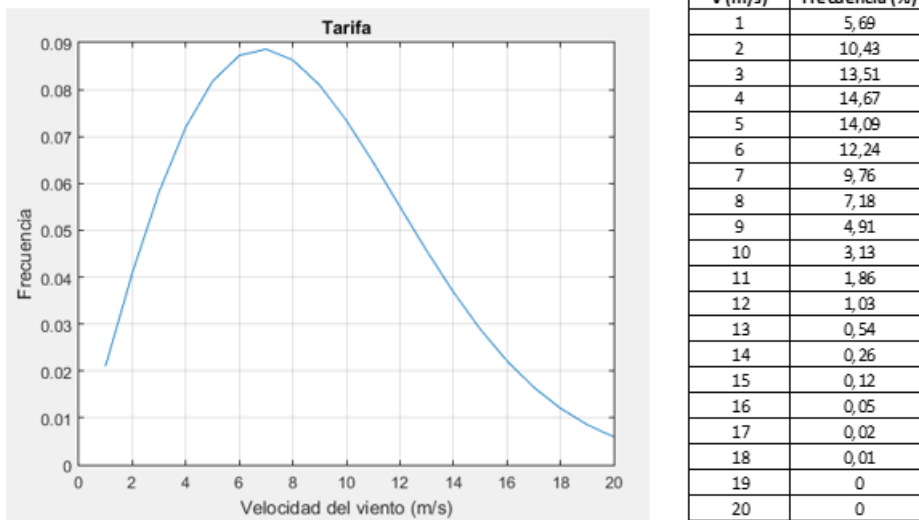


Figura 2-4 Velocidad del viento en Tarifa

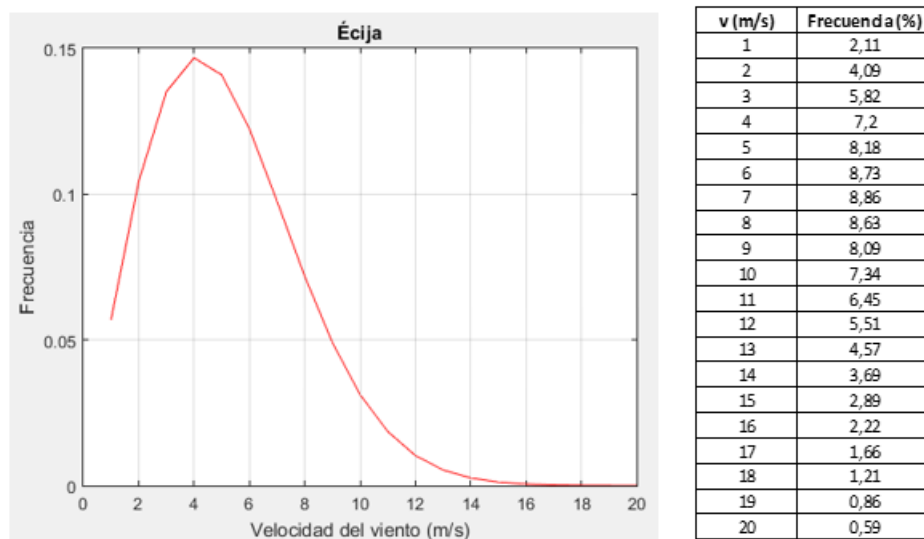


Figura 2-5 Velocidad del viento en Écija

Se puede observar como la distribución en Tarifa abarca un mayor rango de velocidades. Esto es algo que se intuía, ya que es bien sabido que en dicha localidad el viento está muy presente.

El último paso será reflejar esta distribución a una situación real diaria. Para ello, se multiplica la frecuencia con que cada velocidad aparece por 24 y se redondea, llegando así a un valor orientativo de cuantas horas al día se llegará a cada velocidad.

TARIFA																				
v(m/s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Frecuencia horaria	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	0	0	0	0

ÉCIJA																				
v (m/s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Frecuencia horaria	2	3	3	4	3	3	2	2	1	1	0	0	0	0	0	0	0	0	0	0

Figura 2-6 Distriución frecuencial de velocidades para Tarifa y Écija

Para distribuir las velocidades a lo largo de un día, se sigue la curva de la generación eólica típica en España y se busca llegar a una curva similar. En ambas localidades, la distribución quedaría como sigue:

TARIFA																								
v(m/s)	1	2	3	4	4	5	6	7	8	9	10	11	13	15	16	14	12	11	10	9	8	7	6	5
h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

ÉCIJA																								
v (m/s)	1	1	2	2	2	3	3	4	4	5	6	7	8	9	10	8	7	6	6	5	5	4	4	3
h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figura 2-7 Distribución horaria de velocidades para Tarifa y Écija

Por tanto, la generación eólica queda así resuelta, y a partir de dichos valores horarios se llega al valor más representativo posible de la potencia generada. Por ejemplo, para el caso en que la potencia nominal del generador sea de 5kW, como el caso del INTA, haciendo uso de la función previamente expuesta, la potencia de salida sería la siguiente para cada localidad:

TARIFA																								
h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Mult.	0	0,01	0,03	0,06	0,06	0,1	0,15	0,22	0,31	0,4	0,51	0,62	0,84	0,99	1	0,93	0,73	0,62	0,51	0,4	0,31	0,22	0,15	0,1
Pot. (kW)	0	0,05	0,15	0,3	0,3	0,5	0,75	1,1	1,55	2	2,55	3,1	4,2	4,95	5	4,65	3,65	3,1	2,55	2	1,55	1,1	0,75	0,5

ÉCIJA																								
h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Mult.	0	0	0,01	0,01	0,03	0,03	0,03	0,06	0,06	0,1	0,15	0,22	0,31	0,4	0,51	0,31	0,22	0,15	0,15	0,1	0,1	0,06	0,06	0,03
Pot. (kW)	0	0	0,05	0,05	0,15	0,15	0,15	0,3	0,3	0,5	0,75	1,1	1,55	2	2,55	1,55	1,1	0,75	0,75	0,5	0,5	0,3	0,3	0,15

Figura 2-8 Generación eólica horaria para Tarifa y Écija

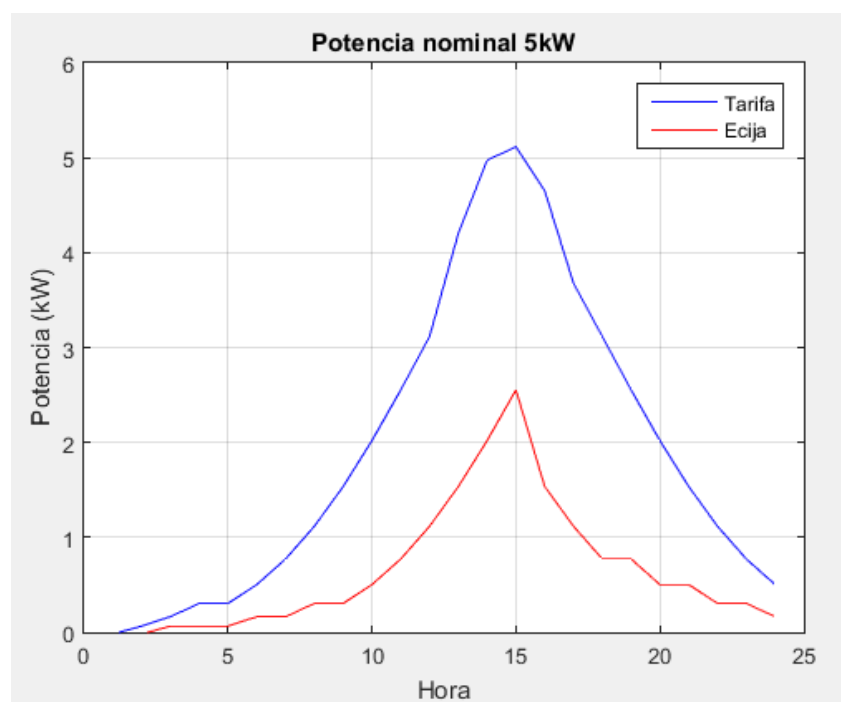


Figura 2-9 Representaión de la generación eólica

Aunque esta situación se corresponde con la más probable, también se tendrán en cuenta situaciones más desfavorables y situaciones más favorables en cuanto a las situaciones meteorológicas se refiere.

2.1.1.2 Generador fotovoltaico

Para el caso del generador fotovoltaico, en primer lugar, vamos a especificar los parámetros necesarios para determinar el funcionamiento fotovoltaico:

- Potencia pico: potencia máxima que se puede alcanzar.
- Irradiancia: potencia incidente por metro cuadrado de superficie.
- Temperatura ambiente.
- Curva P-T: relaciona la potencia con la temperatura. A mayor temperatura, el módulo funciona peor y produce menor potencia.
- Curva de eficiencia del inversor: relaciona la potencia con la eficiencia del inversor. A mayor potencia, mayor eficiencia tiene el inversor.

En este caso, el generador que se va a representar se corresponde con 40 paneles ISOFOTON 150, cuya potencia pico es de 6kW (para una intensidad de 8,12A y un voltaje de 18,5V) y se ajusta al requerimiento de la Microgrid.

Las curvas P-T y de eficiencia del inversor se obtienen del documento de referencia [6] y quedan representadas con las parejas de puntos que siguen:

- Para la curva P-T: $(P, T) = (1.2, 0); (1, 25); (0.8, 75); (0.6, 100)$
- Para la curva de eficiencia del inversor: $(P, \eta) = (0.1, 0.86); (0.2, 0.9); (0.4, 0.93); (1, 0.97)$

A la hora de calcular la irradiancia y la temperatura para poder evaluar la generación fotovoltaica, se va a recoger información anual de 2017 sobre la irradiación global (datos de energía, kWh/m²) y la temperatura tanto para Tarifa como para Écija, a partir de la Agencia Andaluza de la Energía. A partir de estos valores se obtendrá un valor promedio diario y se representará un día cuya irradiancia sea tal.

Para el caso de Écija, la irradiación media diaria es 5kWh/m², mientras que para Tarifa es de 4,9kWh/m². Para representar ambos casos, el día escogido coincide, el 24 de marzo. La curva de irradiancia a representar será la siguiente:

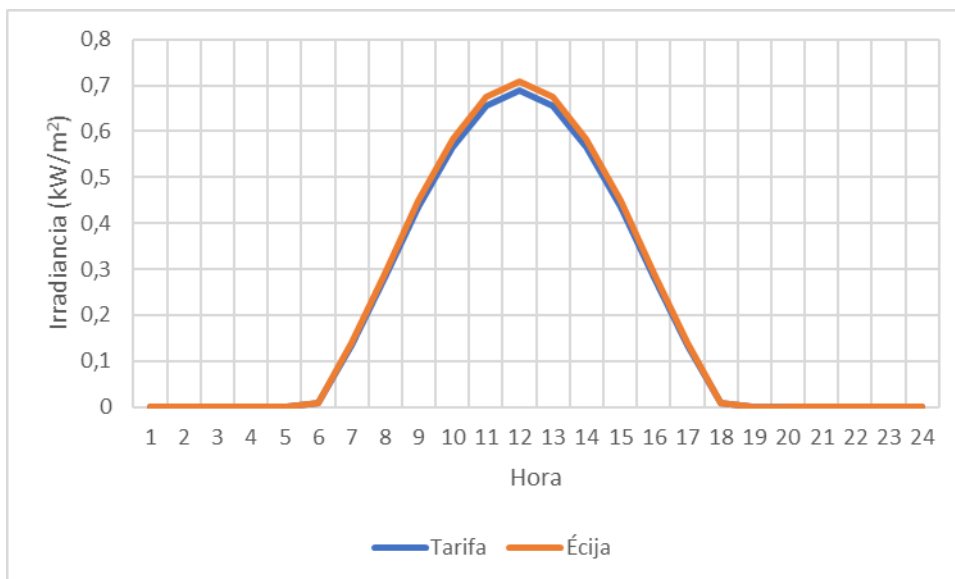


Figura 2-10 Irradiancia horaria para Tarifa y Écija

Como se puede observar, la distribución horaria de la irradiancia es muy similar en ambas localidades. En cuanto a la temperatura, se va a seguir la distribución típica normal de un día sabiendo cual es la temperatura máxima, la media y la mínima. La distribución queda como sigue:

TARIFA																								
Hora	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
T (°C)	12	12	12	12	11	11	13	14	15	17	18	19	21	22	22	22	21	21	18	15	14	13	12	12

ÉCUIA																								
Hora	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
T (°C)	11	11	10	10	9	9	11	12	13	15	17	19	21	22	22	22	21	20	18	15	14	13	12	12

Figura 2-11 Distribución de temperaturas para Tarifa y Écija

Con estos datos quedan por tanto definidos todos los datos necesarios para ver el funcionamiento diario más probable del generador fotovoltaico.

2.1.2 Electrolizador y pila de combustible

El circuito de hidrógeno que compone la Microgrid se basa en el siguiente esquema:

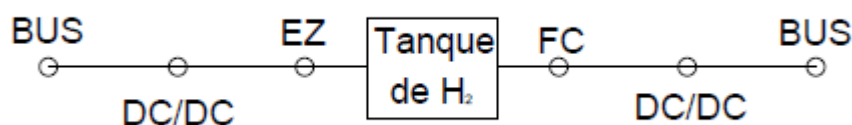


Figura 2-12 Esquema del circuito de hidrógeno

Se van a usar los resultados experimentales obtenidos, para luego usar la relación Potencia-Flujo de Hidrógeno (generado o consumido) para cualquier potencia que queramos evaluar:

- Electrolizador: se tiene una potencia nominal de 3kW capaz de generar hasta 0,5Nm³/h.
- Pila de combustible: se tiene una potencia nominal de 1,5kW consumiendo 1,2534Nm³/h.

En el esquema vemos que existe un convertidor de voltaje para cada componente, ya que ninguno trabaja a 48V. Para representarlo, experimentalmente se observa que para el electrolizador anteriormente instalado cuya potencia era de 1kW, al imponer una potencia de 1,009kW en el bus, el Electrolizador recibía 0,9 kW, y por tanto el rendimiento DC/DC es $\eta_{DCDC} = 0,89$. Esta relación es aplicable a la pila de combustible y, por tanto, la potencia que debe imponerse en el bus ha de ser:

$$P_{bus,EZ} = \frac{P_{EZ}}{\eta_{DCDC}} ; P_{bus,FC} = \eta_{DCDC} \cdot P_{FC}$$

En cuanto al depósito de hidruros, tiene una capacidad de almacenar 7Nm³ a 5bar, que es similar a 35Nm³ a 1 bar, que es el dato que se va a tomar.

2.1.3 Batería

La batería implementada se trata de una batería Plomo-Ácido, de capacidad $C_{120,br}=367Ah$, con un voltaje de 51,58V. Por tanto, en términos de energía, se trata de una batería cuya capacidad máxima es de 18,92kWh. Este componente tendrá el objetivo de estabilizar el voltaje de la planta en 48V.

2.1.4 Carga electrónica

La carga electrónica va a representar la demanda doméstica y los distintos tipos de vehículos.

2.1.4.1 Demanda doméstica

La demanda doméstica a satisfacer será la extrapolada del documento de referencia [3]. Principalmente se seguirá la distribución de la potencia, ya que la potencia en sí puede variar con el fin de llegar a una demanda de potencia que pueda ser satisfecha por la generación. La demanda que se obtiene de dicho documento es la siguiente:

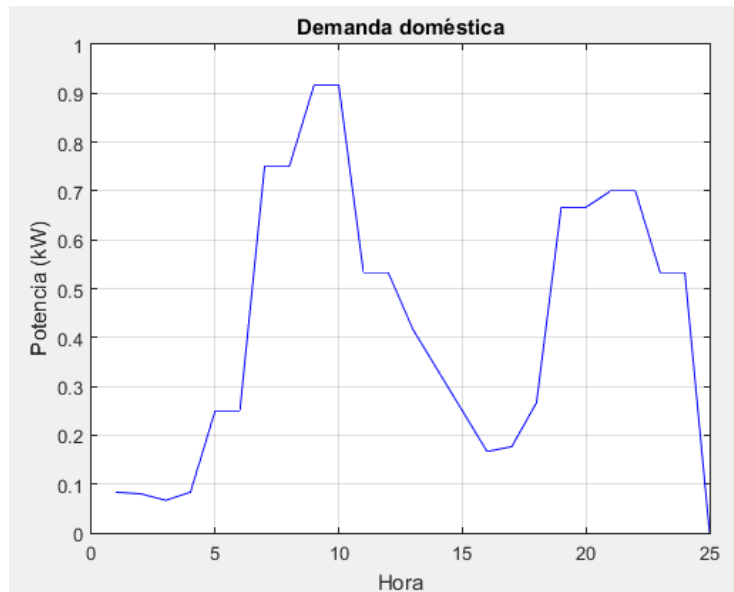


Figura 2-13 Demanda doméstica

2.1.4.2 Vehículos

Los vehículos que se van a tener en cuenta serán un vehículo eléctrico PEV, un vehículo híbrido y un vehículo por pila de combustible. Se buscará que estos vehículos se carguen durante 8 horas por la noche (de 23:00 a 6:59), suponiendo que realizan 50km al día. Los datos de cada uno de ellos han sido obtenidos como sigue:

- Vehículo eléctrico PEV: se ha representado el Volkswagen e-Golf. Las especificaciones técnicas indican que dicho vehículo tiene una autonomía de 300km y una batería de 24,2kWh. Por tanto, al realizar 50km la batería se descargaría alrededor de 4kWh, lo que supone que, para cargar el coche durante 8 horas se necesita una potencia de 0,5kW.
- Vehículo con pila de combustible: se ha representado el Honda Clarity Fuel Cell 2018. Las especificaciones técnicas indican que dicho vehículo tiene un tanque de 33L a 70MPa, que equivale a 23,1Nm³, y una autonomía de 366 millas (589,02km). Por tanto, al realizar 50km se produce un consumo de alrededor de 1,96Nm³ de hidrógeno, lo que supone que, para cargar el coche durante 8 horas se necesita un flujo de 0,2451Nm³/h. Como en nuestra Microgrid la carga está representada electrónicamente, vamos a suponer que el funcionamiento del coche es similar al de la pila de combustible de la Microgrid. Por tanto, si la pila de combustible genera 1,33kW con un flujo de 1,2534Nm³/h, para el flujo de necesario de 0,2451Nm³/h se va a suponer que es necesaria una potencia de 0,2601kW.
- Vehículo híbrido: debido a que un vehículo híbrido puede funcionar tanto mediante un motor libre de emisiones como con uno diesel o gasolina, de las especificaciones técnicas no se puede sacar un dato objetivo de la descarga de la batería. Por tanto, se ha representado el vehículo híbrido estudiado en el documento de referencia [4], donde se ha realizado un seguimiento a un vehículo híbrido con un uso diario. Se llegó a la conclusión de que consumían 12kWh recargando durante todo un día cada 21 días. Esto es equivalente a cargar 0,6kWh cada día, que repartidos en 8 horas resulta una potencia de 0,075kW.

Por tanto, se puede representar la generación y la demanda de energía para cada caso, suponiendo que todos los componentes trabajasen a potencia nominal:

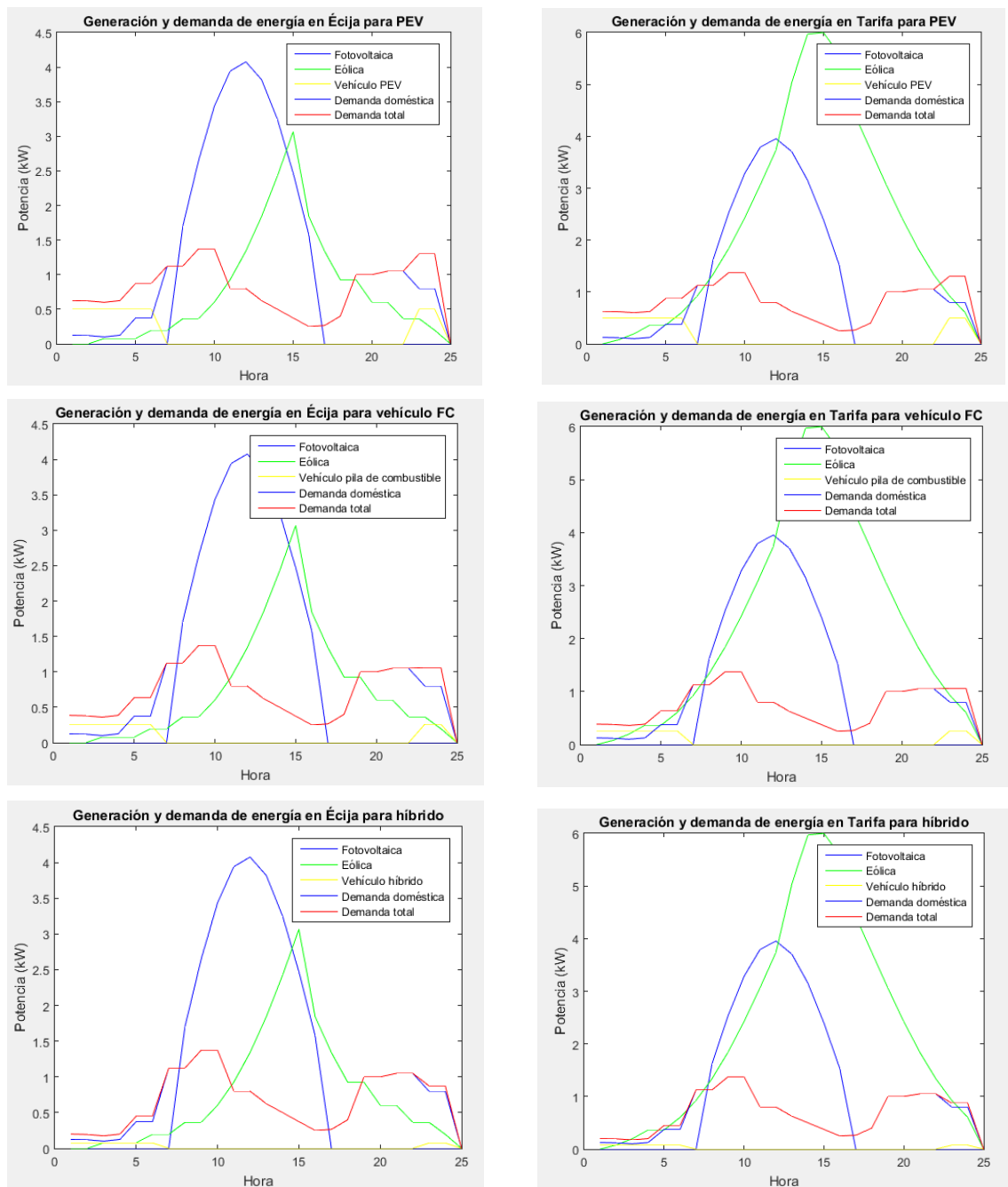


Figura 2-14 Distribución generación y demanda de energía para distintos casos

Más adelante se evaluarán situaciones donde los equipos no trabajen a potencia nominal e incluso posibles cambios que se pudiesen realizar, como, por ejemplo, la modificación de algún equipo del circuito de hidrógeno.

2.2. Descripción de la Microgrid en OpenDSS

En este apartado se presentará la herramienta informática OpenDSS y se verá cómo se ha hecho uso de Matlab para la descripción de la Microgrid, mediante la interfaz COM que relaciona el programa OpenDSS con Matlab.

2.2.1 Herramienta informática OpenDSS

En este apartado se explicará brevemente la herramienta de simulación Open Distribution System Simulator (OpenDSS).

Este programa se trata de una herramienta creada por el ‘Electric Power Research Institute’, cuyo objetivo es la simulación en tiempo secuencial de sistemas de distribución eléctrica en el dominio de la frecuencia. Esta herramienta está implementada en los tres modos que siguen:

1. Un programa ejecutable autónomo (OpenDSS.exe). Permite la simulación de una red a partir de la aplicación independiente.
2. Un proceso COM (Component Object Model) con servidor DLL diseñado para ser ejecutado desde gran variedad de plataformas de software existentes (OpenDSSEngine.DLL)
3. Una convención de llamada Stdcall DLL que proporciona todas las funciones del servidor COM, pero que puede ser usado mediante lenguajes que no soportan COM (OpenDSSDirect.DSS)

La figura 2-15 muestra la interfaz de la herramienta OpenDSS, donde, como se puede observar, se podrá simular una red directamente en forma de texto. La simulación de una red directamente ejecutada desde la interfaz de OpenDSS se define con claridad en el documento de texto [6].

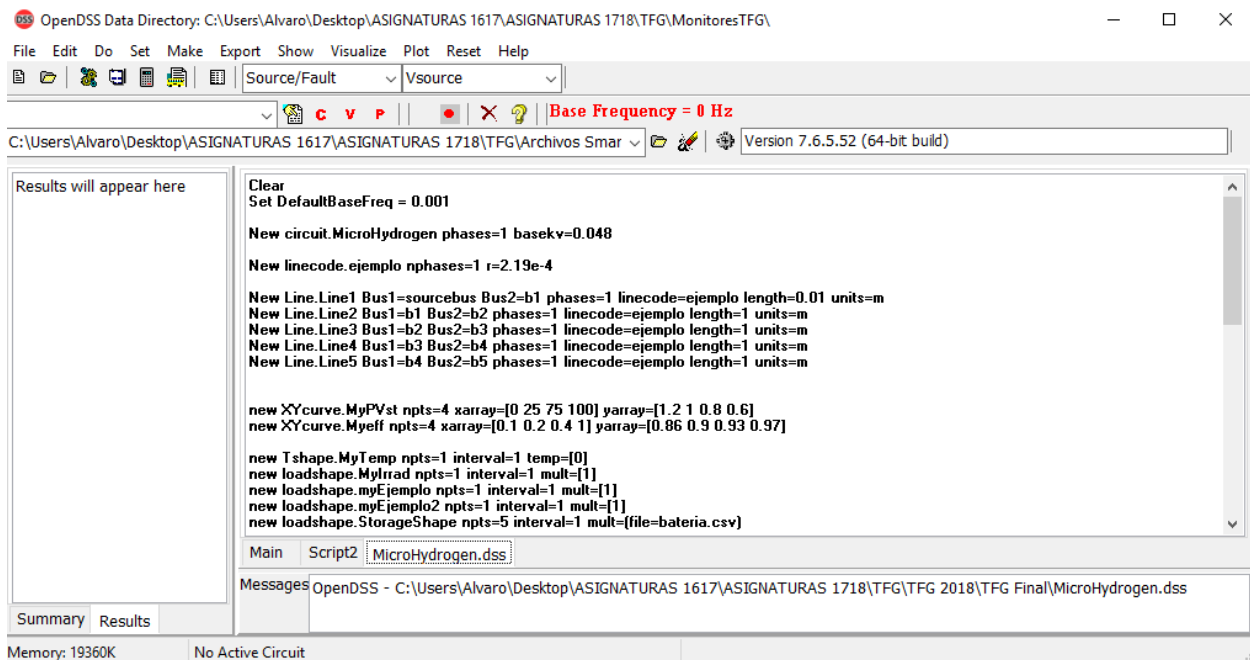


Figura 2-15 Interfaz gráfica de OpenDSS

OpenDSS permite la implantación los siguientes elementos básicos: de suministro de potencia (líneas, transformadores y condensadores), elementos de conversión de potencia (generadores, cargas disipadoras de energía, PVSystem y elementos de almacenamiento) y medidores (EnergyMeter, monitores y sensores). Se puede observar como permite la implantación de generadores, cargas, sistemas fotovoltaicos y elementos de almacenamiento, por lo que es perfecto para la simulación de la Microgrid estudiada.

Además, OpenDSS permite la obtención de información importante de cada uno de los elementos que sean incluidos en la red, como puedan ser las pérdidas, el estado de la batería, la intensidad o la tensión. Esto será de gran importancia debido a que las pérdidas y el estado de la batería serán criterios que se van a evaluar a la hora de determinar una alternativa óptima para la Microgrid.

Por otro lado, OpenDSS permite diversos modos de solución, como pueden ser el diario o el anual, indicando los multiplicadores que ponderan la potencia del objeto especificado para la secuencia de tiempo indicada (24 para el modo diario y 365 para el anual) (*LoadShapes*), y diversos modelos de descripción de las cargas o los generadores, indicando por ejemplo si trabajan a voltaje o intensidad constante.

La figura 2-15 muestra la estructura que sigue la herramienta. Se observa como existen distintas formas de

modelar el código. Esto puede hacerse desde la interfaz COM o incluyendo el texto directamente desde la interfaz de 'OpenDSS.exe', como se ha comentado. Además, los resultados también pueden ser visualizados desde otros programas, y por tanto no tiene que ser realizada la simulación desde el mismo programa. Este caso va a ser el que va a ser implementado en este trabajo, ya que se definirá la red a partir de Matlab.

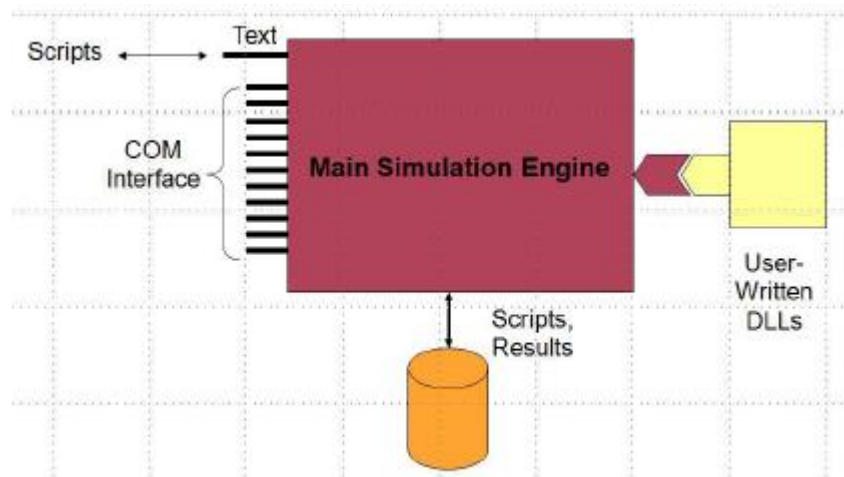


Figura 2-16 Estructura OpenDSS

En el siguiente apartado se explicará el uso de la interfaz COM que relaciona el programa OpenDSS con Matlab y permite el control de una planta a partir de un programa ajeno a OpenDSS

2.2.2 Interfaz COM OpenDSS-Matlab

En este apartado se va a explicar simplemente cómo llamar al programa OpenDSS desde Matlab y cómo obtener información del mismo.

Para la iniciación del programa OpenDSS y hacer que el programa Matlab interactúe con OpenDSS se ejecuta el código que se muestra a continuación. Se observa como se activa el servidor de OpenDSS que permite la interacción Matlab-OpenDSS. Este comando permite la escritura de texto en OpenDSS, la selección de un Circuito Activo y la obtención de soluciones del circuito.

```
DSSObj=actxserver('OpenDSSEngine.DSS');
DSSObj.Start(0);
DSSText=DSSObj.Text;
DSSCircuit=DSSObj.ActiveCircuit;
DSSSolution=DSSCircuit.Solution;
```

Una vez que se ha inicializado el programa, para indicar que un comando quiere ser ejecutado en el programa openDSS, debe estar precedido de '*DSSText.Command=*'. A partir de este comando se puede escribir el código que se quiera ejecutar en OpenDSS, como, por ejemplo, para limpiar los datos que puedan estar almacenados en el programa, el comando que se debe escribir en Matlab es el siguiente: '*DSSText.Command='Clear'*'.

Por tanto, siempre que se quiera escribir un comando para crear un objeto o modificarlo, se podrá hacer mediante este comando.

Otro modo de obtener información del sistema o de algún objeto desde Matlab será mediante el comando '*DSSCircuit.*' ó '*DSSObject.*' seguido de la información que se desea conocer. Por ejemplo, para obtener las pérdidas del circuito se hace uso del comando '*DSSCircuit.Losses*', sin necesidad de usar el comando '*DSSText.Command=*'.

A la hora de especificar un objeto, se debe tener en cuenta que se deben indicar los parámetros que puedan definirlo, y que este objeto debe ser especificado en su totalidad. Si se especifica únicamente el nombre del

objeto y no el tipo de objeto al que se refiere, se tomará el tipo de objeto anterior que haya sido especificado. Los parámetros deben ser indicados a partir del tipo de parámetro que se quiera especificar o según el orden que sigue cada objeto para la definición de los parámetros.

A partir de los comandos *'first'* y *'next'* se podrá pasar de un objeto a otro del mismo tipo, como puede ser para pasar de una línea a la siguiente.

El comando *'DSSCircuit.SetActiveElement'* permite la selección de un elemento en particular, y la obtención o la modificación de algún parámetro del elemento activo. Por ejemplo, a partir del siguiente comando se obtendrían los buses que engloba la línea descrita: *'DSSCircuit.SetActiveElement ([line.Linea1]); 'lineBusNames=DSSCircuit.ActiveElement.BusNames'*.

En este documento se va a utilizar principalmente la interfaz de texto de OpenDSS a partir de Matlab, realizando de control a partir de esta posibilidad.

Al igual que para el apartado anterior, en el documento de referencia [6] se explica detalladamente la interfaz COM OpenDSS-Matlab.

2.2.3 Implementación de la Microgrid en OpenDSS

Como se acaba de mencionar, se va a seguir un control por histéresis, controlado desde Matlab, donde se establece que el electrolizador se activará siempre que la batería se encuentre en un valor superior al 60% de su capacidad y que la pila de combustible generará corriente sólo cuando la capacidad de la batería sea inferior al 40%. Si el estado de la batería está entre el 40% y el 60% de la capacidad, la Microgrid actuará sin control, es decir, la batería se cargará si existe exceso de potencia entrante y se descargará si existe exceso de carga, sin hacer uso del electrolizador ni de la pila de combustible.

Buscando la similitud con la Microgrid real, el esquema ortogonal que se va a seguir es el mostrado en la figura 2-17. En dicho esquema, el generador representa la generación fotovoltaica o eólica y la carga la demanda doméstica y de los vehículos. El tanque de hidrógeno no se muestra debido a que se deberá controlar mediante Matlab de manera manual, ya que OpenDSS no contempla la simulación de elementos de generación o consumo de hidrógeno.

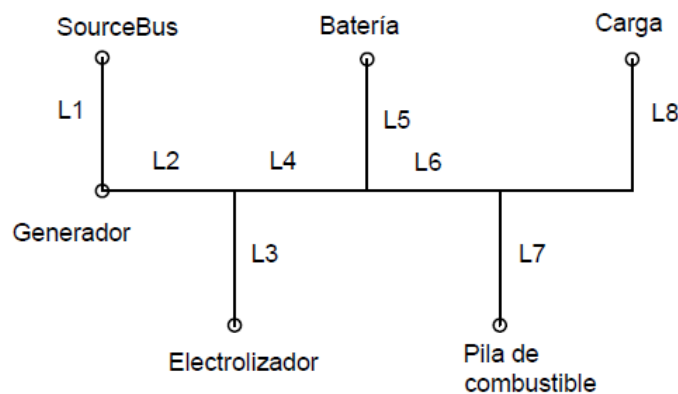


Figura 2-17 Esquema ortogonal Microgrid para OpenDSS

A continuación, se va a mostrar el código descrito en Matlab detallándolo punto por punto. Este código se corresponde con una función que devolverá las pérdidas de la Microgrid, el coste operativo de la planta generadora, el nivel final de hidrógeno generado y la diferencia entre el estado inicial de la batería y el final. Para ello, es necesario aportar las potencias nominales de vehículo, demanda doméstica, generación fotovoltaica/eólica, del electrolizador y de la pila de combustible, además del coste por kWh de la energía (que dependerá del año en que se haga la evaluación) y la ciudad (Écija se representa con un 0 y Tarifa con un 1). El código mostrado se corresponde con la generación fotovoltaica. No se muestra también el

correspondiente a la generación eólica porque simplemente cambia el apartado de la generación, y se puede visualizar en el Anexo.

```
function [Perd, Coste, H, SOC]=Fotovoltaica_Multicriterio(SOC, kWVeh, kWDom,
kWPV, kWFC, kW EZ, Coste_PV, Ciudad)
```

Se define la generación y la demanda:

```
%Ciudad=0 --> Ecija ; Ciudad=1 --> Tarifa
if Ciudad==0
irradnu=[0 0 0 0 0 0.0081 0.1382 0.2915 0.4478 0.5833 0.6753 0.7079
0.6753 0.5833 0.4478 0.2915 0.1382 0.0081 0 0 0 0 0 0];
temp=[11 11 10 10 9 9 11 12 13 15 17 19 21 22 22 22 21 20 18 15 14 13 12 12];
end
if Ciudad==1
irradnu=[0 0 0 0 0 0.0075 0.1339 0.2828 0.4348 0.5664 0.6559 0.6876
0.6559 0.5664 0.4348 0.2828 0.1339 0.0075 0 0 0 0 0 0 0];
temp=[11 12 12 12 11 11 13 14 15 17 18 19 21 22 22 22 21 21 18 15 14 13 12
12];
end
PCar=[1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0]; %El último 0 es
para el cálculo de la Diferencia Oferta-Demanda
```

Se calcula la potencia total capaz de suministrar la planta y el coste asociado. Se supone que el depósito de hidruros parte con 10Nm³ (de 35Nm³ de capacidad máxima) y que el estado inicial de la batería lo impone el usuario:

```
PV=Fotovoltaica(kWPV, Ciudad);
Pot=sum(PV);
Coste=Pot*Coste_PV;

%Condicion inicial.
EZ=1;
FC=0;
MHL_inicial=10;
MHL=MHL_inicial;
nivelMHL(1)=MHL;
SOC=SOC_inicial;
nivelSOC(1)=SOC_inicial;

%Rendimiento del convertidor DCDC(experimental)
rend_dcdc=0.89;

%Demanda doméstica
Dom=[125 120 100 125 375 375 1125 1125 1375 1375 800 800 625 500 375 250 265
400 1000 1000 1050 1050 800 800 0]./1375; % Multiplicadores demanda doméstica

%Valor inicial de la batería
D=Dom(1)*kWDom;
B=-(PV(1)-EZ*kWEZ/rend_dcdc+FC*kWFC*rend_dcdc-PCar(1)*kWVeh-D);
```

Se inicia el programa OpenDSS y se especifica el directorio donde recoger información:

```

DSSObj=actxserver('OpenDSSEngine.DSS');
DSSObj.Start(0);
DSSText=DSSObj.Text;
DSSCircuit=DSSObj.ActiveCircuit;
DSSSolution=DSSCircuit.Solution;

DSSText.Command='Clear';
DSSText.Command='set datapath= "C:\Users\Alvaro\Desktop\ASIGNATURAS
1617\ASIGNATURAS 1718\TFG\MonitoresTFG"';
DSSText.Command='Set DefaultBaseFreq = 0.001';

DSSText.Command='New circuit.MicroHydrogen phases=1 basekv=0.048';

```

Se definen las líneas del circuito:

```

DSSText.Command='New linecode.ejemplo nphases=1 r=2.19e-4';

DSSText.Command='New Line.Line1 Bus1=sourcebus Bus2=b1 phases=1
linecode=ejemplo length=0.01 units=m';
DSSText.Command='New Line.Line2 Bus1=b1 Bus2=b2 phases=1 linecode=ejemplo
length=1 units=m';
DSSText.Command='New Line.Line3 Bus1=b2 Bus2=b2.1 phases=1 linecode=ejemplo
length=1 units=m';
DSSText.Command='New Line.Line4 Bus1=b2 Bus2=b3 phases=1 linecode=ejemplo
length=1 units=m';
DSSText.Command='New Line.Line5 Bus1=b3 Bus2=b3.1 phases=1 linecode=ejemplo
length=1 units=m';
DSSText.Command='New Line.Line6 Bus1=b3 Bus2=b4 phases=1 linecode=ejemplo
length=1 units=m';
DSSText.Command='New Line.Line7 Bus1=b4 Bus2=b4.1 phases=1 linecode=ejemplo
length=1 units=m';
DSSText.Command='New Line.Line8 Bus1=b4 Bus2=b5 phases=1 linecode=ejemplo
length=1 units=m';

```

Se definen los multiplicadores que van a imponer el estado de cada componente. A la hora de definirlos se hace imponiendo el estado inicial:

```

DSSText.Command='new XYcurve.MyPVst npts=4 xarray=[0 25 75 100] yarray=[1.2 1
0.8 0.6]';
DSSText.Command='new XYcurve.Myeff npts=4 xarray=[0.1 0.2 0.4 1] yarray=[0.86
0.9 0.93 0.97]';

DSSText.Command='new Tshape.MyTemp npts=1 interval=1 temp=[11]';
DSSText.Command='new loadshape.MyIrrad npts=1 interval=1 mult=[0]';
DSSText.Command='new loadshape.electrolizador npts=1 interval=1 mult=[1]';
DSSText.Command='new loadshape.StorageShape npts=1 interval=1 mult=[1]';
DSSText.Command='new loadshape.fcell npts=1 interval=1 mult=[0]';
DSSText.Command='new loadshape.Pcar npts=1 interval=1 mult=[1]';
DSSText.Command='new loadshape.Domestica npts=1 interval=1 mult=[0.125]';

```

Se definen todos los componentes:

```
DSSText.Command='new PVSystem.PV phases=1 Bus1=b1 kV=0.048 irrads=0.963
temperature=25 basefreq=0.001';
DSSText.Command='pf=1 effcurve=Myeff P-TCurve=MyPVst Duty=MyIrrad
TDuty=MyTemp';
DSSText.Command=['PVSystem.PV.Pmpp=' num2str(kWPV)];
DSSText.Command=['PVSystem.PV.kVA=' num2str(kWPV)];

DSSText.Command='new Load.Car phases=1 Bus1=b5 kV=0.048 pf=1 model=1
basefreq=0.001 Duty=Pcar';
DSSText.Command=['Load.Car.kW=' num2str(kWVeh)];
%Demanda doméstica
DSSText.Command='new Load.Dom phases=1 Bus1=b5 kV=0.048 pf=1 model=1
basefreq=0.001 Duty=Domestica';
DSSText.Command=['Load.Dom.kW=' num2str(kWDom)];

DSSText.Command='new Load.Electrolizador phases=1 Bus1=b2.1 kV=0.048 pf=1
model=1 basefreq=0.001 Duty=electrolizador';
DSSText.Command=['Load.Electrolizador.kW=' num2str(kWEZ/rend_dc);];

DSSText.Command='new Generator.FuelCell phases=1 Bus1=b4.1 kV=0.048 pf=1
model=1 basefreq=0.001 Duty=fcell';
DSSText.Command=['Generator.FuelCell.kW=' num2str(kWFC*rend_dc);];

DSSText.Command='new Storage.Bateria phases=1 Bus1=b3.1 kV=0.048 kWrated=1.5
kW=1 kWhrated=18.92';
DSSText.Command=['Storage.Bateria.kWhstored=' num2str(SOC)];
DSSText.Command='basefreq=0.001 DispMode=follow Duty=StorageShape';

%Impongo el valor inicial de la batería (si está en carga o descarga y con
qué potencia
DSSText.Command=['LoadShape.StorageShape.mult=' num2str(B)];
```

Se definen los monitores para saber los estados cada hora del sistema fotovoltaico y de la batería:

```
DSSText.Command='new monitor.pvS PVSystem.PV mode=1 ppolar=no';
DSSText.Command='new monitor.bateria Storage.Bateria mode=3';

DSSText.Command='set voltagebases=[0.048]';
DSSText.Command='calcv';
for hour=1:24
```

Se impone el modo de solución (que evalúa horariamente) y se exportan los datos de los monitores y las pérdidas:

```
DSSText.Command='solve mode=duty stepsize=3600 number=1';

DSSText.Command='export mon pvS';
DSSText.Command='export mon bateria';
DSSText.Command='export losses';
```

Se evalúan las pérdidas en la batería y en el circuito:

```
%Perdidas en la batería
```

```

datosBAT(:, :)=textread('MicroHydrogen_Mon_bateria.csv', '', 'delimiter', ',', 'headerlines', 1, 'emptyvalue', NaN);
perdidasBAT(hour)=datosBAT(:, 7);

%Perdidas en el circuito (líneas)
SystemLosses=(DSSCircuit.Losses);
perdidasCircuito(hour)=SystemLosses(1)/1000;

%Valor de kWhStored en la bateria
kWhStored(hour)=datosBAT(:, 3);

```

Se evalúan las pérdidas de electrolizador y pila de combustible:

```

%Perdidas en EZ y FC
if EZ==1
perdidasEZ(hour)=kWEZ*(1-rend_dc/dc)/rend_dc/dc;
else
perdidasEZ(hour)=0;
end

if FC==1
perdidasFC(hour)=kWFC*(1-rend_dc/dc);
else
perdidasFC(hour)=0;
end

if B<0 %Bateria cargando
kWBATERIA(hour)=datosBAT(:, 6);
end

if B>0 %Bateria descargando
kWBATERIA(hour)=datosBAT(:, 5);
end

if B==0
kWBATERIA(hour)=0;
end

MHL=MHL+EZ*0.5/3*kWEZ-FC*1.2534/1.5*kWFC;
SOC=kWhStored(hour)+kWBATERIA(hour)-perdidasBAT(hour);

if SOC>18.92
SOC=18.92;
End

%Variacion de los niveles de hidrógeno y de la bateria
%con el tiempo
nivelMHL(hour)=MHL;
nivelSOC(hour)=SOC;

```

Se establece el control por histéresis, dependiente del nivel de carga de la batería:

```

if SOC < 18.92
if SOC > 0.60*18.92
disp('Electrolizador ON')

```



```
EZ=1;
FC=0;
MHL=MHL+0.5/3*kWEZ;%Experimental
end
end
if SOC < 0.60*18.92
if SOC > 0.4*18.92
disp('SIN CONTROL')
EZ=0;
FC=0;
end
end
if SOC < 0.4*18.92
disp('Pila de combustible ON')
EZ=0;
FC=1;
MHL=MHL-1.2534/1.33*kWFC;%Experimental.
end

SOC=kWhStored(hour)+kWBATERIA(hour)-perdidasBAT(hour);
```

Se calcula la diferencia entre Oferta y Demanda de energía para saber qué potencia debe suministrar la batería:

```
Dif_OfertaDemanda=PV(hour+1)+FC*kWFC/rend_dc-dc-EZ*kWEZ*rend_dc-dc-
PCar(hour+1)*kWVeh-D*kWDom;
B=-Dif_OfertaDemanda;
```

Se actualizan los multiplicadores a la hora siguiente:

```
T=temp(hour+1);
DSSText.Command=['TShape.MyTemp.temp=' num2str(T)];

I=irradnu(hour+1);
DSSText.Command=['LoadShape.MyIrrad.mult=' num2str(I)];

C=PCar(hour+1);
DSSText.Command=['LoadShape.Pcar.mult=' num2str(C)];

DSSText.Command=['LoadShape.electrolizador.mult=' num2str(EZ)];

DSSText.Command=['LoadShape.fcell.mult=' num2str(FC)];

DSSText.Command=['LoadShape.StorageShape.mult=' num2str(B)];

D=Dom(hour+1);
DSSText.Command=['LoadShape.Domestica.mult=' num2str(D)];

end
nivelSOC;
nivelMHL;
```

```
% Para la representación de los niveles de SOC y MHL
```

```

% figure
% hour=1:25;
% plot(hour,nivelSOC,'g',hour,nivelMHL,'b')
% xlabel('Hora')
% ylabel('kWh / Nm3')
% legend('SOC','MHL')
% grid

```

```

PerdidasBATdia=sum(perdidasBAT);
PerdidasCircuitodia=sum(perdidasCircuito);
PerdidasFCdia=sum(perdidasFC);
PerdidasEZdia=sum(perdidasEZ);

```

Se resumen los parámetros de salida:

```

%Asignamos el valor de las perdidas totales a Perd
Perd=PerdidasBATdia+PerdidasCircuitodia+PerdidasFCdia+PerdidasEZdia;

H=nivelMHL(25)-MHL_inicial;
SOC=nivelSOC(25)-SOC_inicial;
end

```

El objetivo será, a partir de esta función, evaluar los resultados obtenidos a partir del método Promethee para obtener una alternativa óptima.

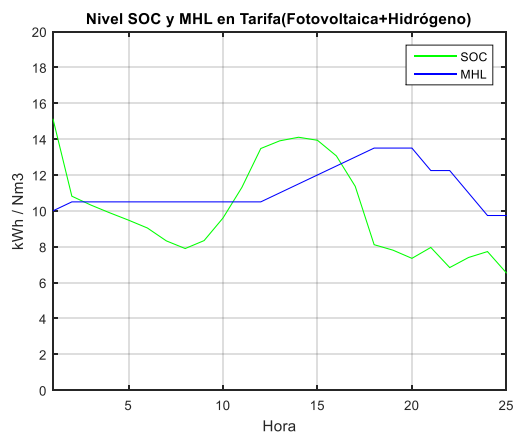
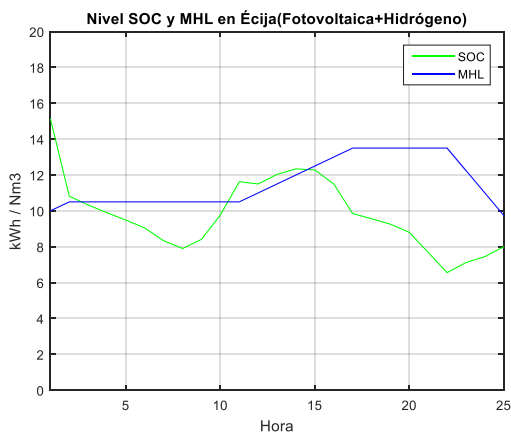
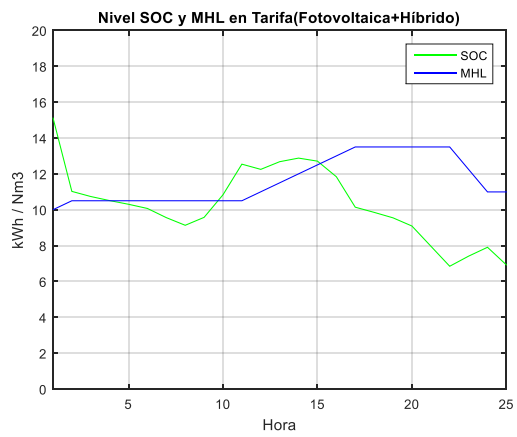
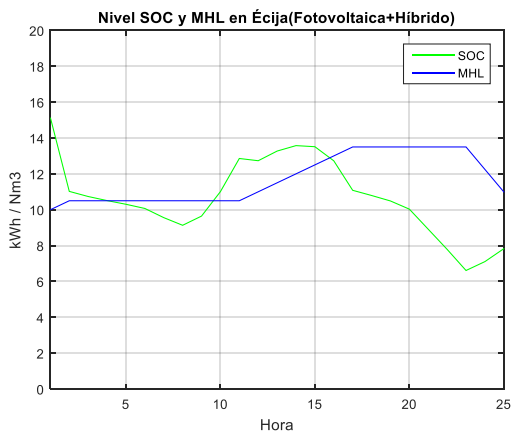
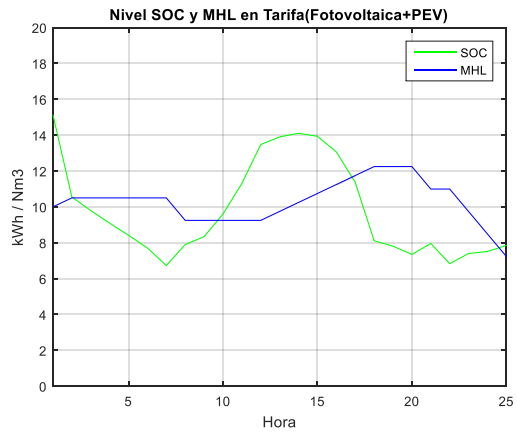
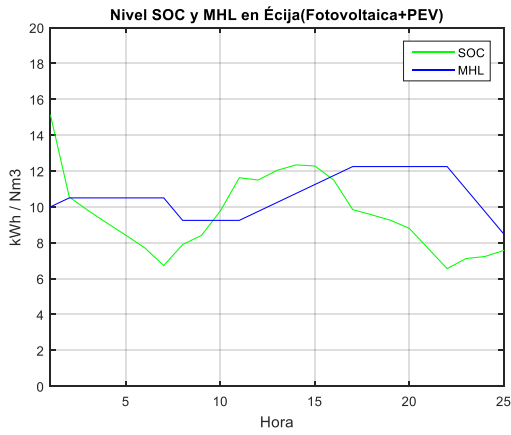
La figura 2-16 engloba varias representaciones de distintas posibles situaciones de la Microgrid, y se muestra la evaluación del estado de la batería y del depósito de hidruros. Se puede ver la diferencia para ambas fuentes de energía y localidades para la situación actual de la planta:

- Generador electrónico: suponemos un generador eólico de potencia nominal 5kW y uno fotovoltaico de potencia nominal de 6kW.
- Carga electrónica: se establece la curva de potencia previamente establecida (demanda doméstica y vehículos recargables).
- Electrolizador: de potencia nominal 3kW.
- Pila de combustible: de potencia nominal 1.5kW.

En esta figura se representa el valor de la energía total almacenada en la batería en kWh y la cantidad de hidrógeno almacenado en el tanque en volumen (Nm³). Por tanto, pese a que estas representaciones se encuentran en la misma gráfica, no tienen las mismas unidades.

Cabe mencionar que la simulación se realiza para un periodo de 24 horas, siendo controlada mediante un control por histéresis a partir de los datos obtenidos cada hora.

En las gráficas que se muestran en la figura 2-16, se puede observar como, en función del tipo de generación y del vehículo que se esté evaluando, es posible llegar a una cantidad superior de hidrógeno almacenada o a una mayor cantidad de energía almacenada en la batería. También se observa a primera vista que el uso de energía fotovoltaica puede ser más adecuado para Écija y la energía eólica para Tarifa, ya que se obtiene un mayor nivel de hidrógeno y de energía almacenada, así como que, con el uso del vehículo híbrido, se consigue llegar a una mejor situación final.



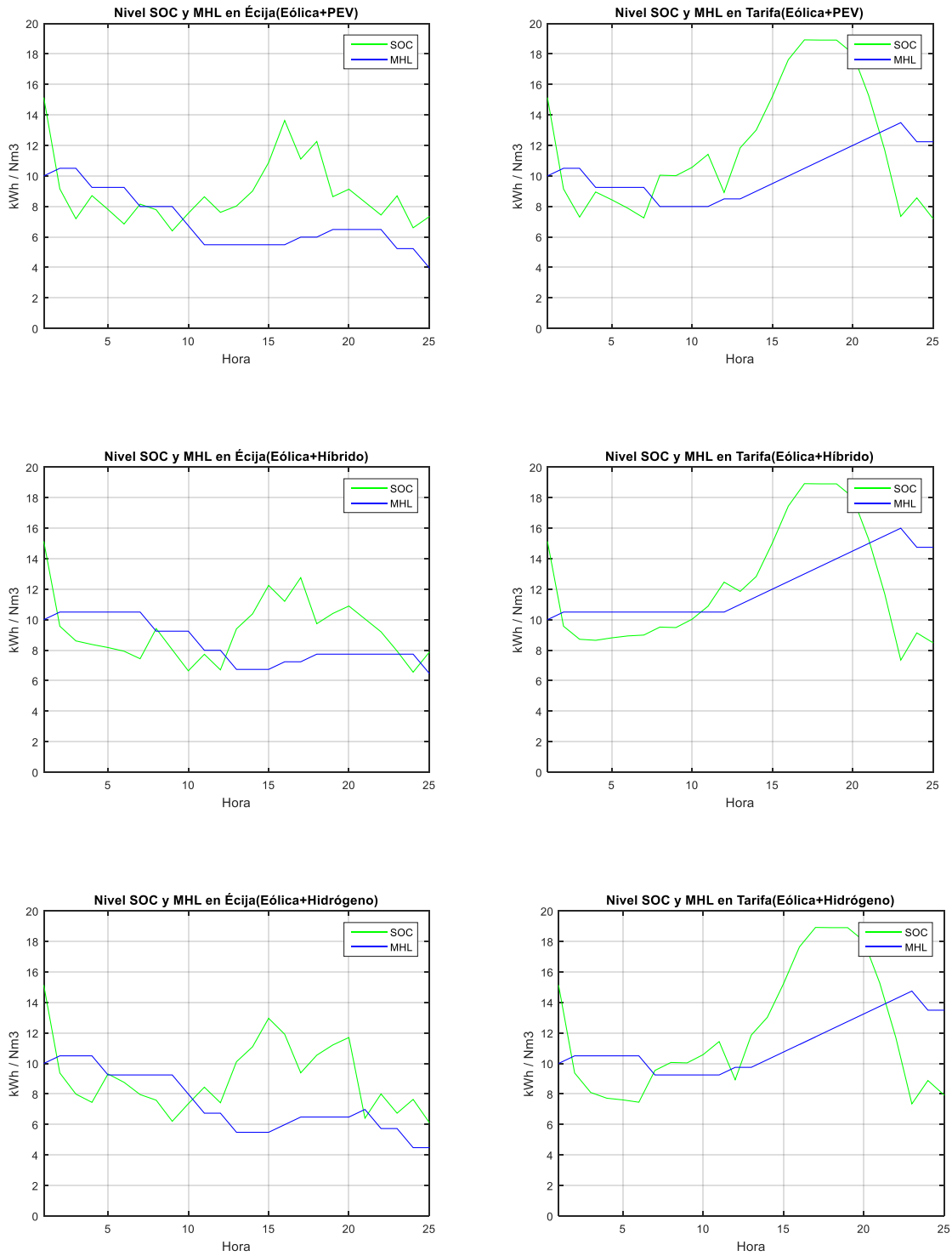


Figura 2-18 Estado de la batería y del tanque de hidrógeno para distintas situaciones

3 MÉTODO DE DECISIÓN MULTICRITERIO

En este apartado se describirá brevemente el método de evaluación seguido, el método PROMETHEE, Preference Ranking Organization METHod for Enrichment of Evaluations.

Como hemos comentado antes, el objetivo de este proyecto será evaluar distintas alternativas en cuanto a ciertos criterios, que serán: estado final del tanque de hidrógeno, estado final de la batería, pérdidas en el circuito, coste de la generación energética e impacto medioambiental.

Además del uso del método PROMETHEE se calcularán los pesos de cada criterio a partir del método AHP.

3.1 Método Promethee

Este método de evaluación se basa en evaluar las diferencias para cada criterio en cada pareja de alternativas. A partir de estas diferencias en los distintos valores se establecerá una preferencia o indiferencia de una alternativa respecto a otra para cada criterio. Los pasos seguidos para esta técnica son los siguientes:

1. Se recoge los valores de cada alternativa para cada criterio.
2. Para cada criterio, se comparan los valores de las alternativas. Para ello se especifica una matriz por criterio con las diferencias existentes entre las alternativas $d_j(a, b) = g_j(a) - g_j(b)$, donde a y b son distintas alternativas, g se refiere a los criterios a evaluar y j el criterio que se está evaluando.
3. Se evalúa la matriz de diferencias mediante funciones de preferencia $P_j(a, b)$. Estas funciones se determinan para cada criterio y establecen la preferencia y la indiferencia de una alternativa respecto a otra. Se devuelve un valor entre 0 y 1, siendo la alternativa a tan preferente respecto a b como cercano a 1 sea este valor. Las distintas funciones de preferencia son las siguientes:

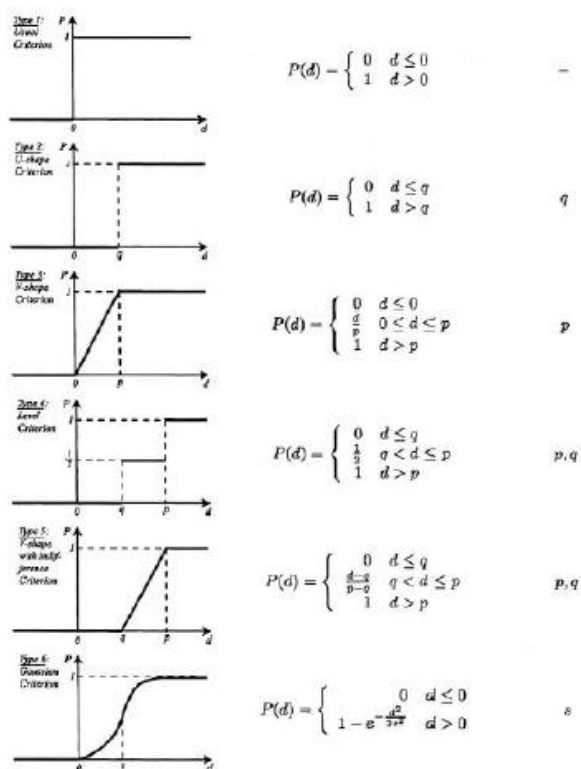


Figura 3-1 Funciones de preferencia método Promethee

Para la definición completa de las funciones de preferencia se tienen que especificar el umbral de indiferencia q (indica que, si la diferencia d es menor que este valor, la diferencia se considera insignificante y una alternativa es indiferente a otra, y por tanto se asigna un valor de $P=0$) y el umbral de preferencia estricta p (indica que, si la diferencia d es mayor que este valor, la diferencia se considera suficiente para indicar que una alternativa es estrictamente preferente a otra, y por tanto se asigna un valor de $P=1$).

4. Se establecen los pesos de los criterios w_j .
5. Se obtienen los índices de preferencia agregados. Expresan el grado de preferencia que tiene a sobre b . Estos índices son globales, y tienen en cuenta todos los criterios indicados. Cuanto más cercano a 1 sea este valor, mayor preferencia tendrá la alternativa a respecto a la alternativa b . $\pi(a, b) = \sum_{j=1}^k P_j(a, b) \cdot w_j$.
6. Como último parámetro se obtienen los flujos de superación. Engloban los índices de preferencia agregados para las n alternativas existentes. Con este valor se sabrá la preferencia de una alternativa respecto al resto (flujo de superación positivo) y cuantas alternativas son preferentes a la alternativa que se está evaluando (flujo de superación negativo).

$$\phi^+(a) = \frac{1}{n-1} \sum_{x \in A} \pi(a, x); \quad \phi^-(a) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a)$$

, donde A se refiere a todas las alternativas existentes.

El objetivo ahora es buscar la alternativa cuyo flujo positivo sea elevado y el negativo sea pequeño.

7. Se evalúan los flujos de superación obtenidos. Para ello se tienen dos alternativas: PROMETHEE I y PROMETHEE II:
 - PROMETHEE I: se hace la siguiente evaluación con los flujos de superación negativos y positivos, donde P indica preferencia, I indiferencia y R incomparabilidad:

$$aP^I b \Leftrightarrow \begin{cases} \phi^+(a) > \phi^+(b) \text{ y } \phi^-(a) < \phi^-(b) \\ \phi^+(a) = \phi^+(b) \text{ y } \phi^-(a) < \phi^-(b) \\ \phi^+(a) > \phi^+(b) \text{ y } \phi^-(a) = \phi^-(b) \end{cases}$$

$$aI^I b \Leftrightarrow \phi^+(a) = \phi^+(b) \text{ y } \phi^-(a) = \phi^-(b)$$

$$aR^I b \Leftrightarrow \begin{cases} \phi^+(a) > \phi^-(b) \text{ y } \phi^-(a) > \phi^-(b) \\ \phi^+(a) < \phi^+(b) \text{ y } \phi^-(a) < \phi^-(b) \end{cases}$$

La alternativa escogida será aquella que sea preferente a un mayor número de alternativas.

- PROMETHEE II: la evaluación se realiza con los flujos de superación netos, para eliminarse la incomparabilidad. Esta evaluación da menos información, pero puede servir para ciertos casos donde no se pueda definir la preferencia global con los flujos positivos y negativos:

$$\phi(a) = \phi^+(a) - \phi^-(a)$$

$$aP^{II} b \Leftrightarrow \phi(a) > \phi(b)$$

$$aI^{II} b \Leftrightarrow \phi(a) = \phi(b)$$

Para el caso de la Microgrid resuelta, las funciones de preferencia que se van a imponer como predeterminadas son las siguientes:

- Estado de la batería (SOC): función Linear, con un valor de $p=5$ y $q=0,2$.
- Estado del depósito de hidruros (MHL): función Linear, con un valor de $p=5$ y $q=0,2$.
- Pérdidas: función Linar, con un valor de $p=6$ y $q=0,2$.
- Coste: función V-Shape, con un valor de $p=5$.
- Impacto medioambiental: función Usual.

3.2 Obtención de los pesos a partir del método AHP

Otro método de decisión que sirve para la evaluación de problemas multicriterio es el AHP, Analytic Hierarchy Process.

En el caso que nos compete, no se va a realizar la evaluación por este método, pero sí se van a obtener los pesos de cada criterio, por lo que únicamente se va a explicar su obtención. Para un entendimiento más específico de este método, el documento de referencia <<TFG DE LOS MÉTODOS>> lo explica detalladamente y expone un ejemplo.

Para la obtención de los pesos, en primer lugar se debe de imponer la importancia relativa de un criterio respecto a otro, creando una matriz de comparación A nxn (siendo n el número de criterios) donde el valor a(j,k) hace referencia a la importancia del criterio j sobre el criterio k, siendo j más importante que k si este valor es mayor que 1. Los valores de la matriz serán de 0 a 9, y el valor a(k,j)=1/a(j,k), por lo que sólo es necesario imponer los valores de la diagonal inferior o superior de la matriz. Los valores de preferencia se establecerán como sigue:

Valor de a_{jk}	Interpretación
1	j y k son igual de importantes
3	j es ligeramente más importante que k
5	j es más importante que k
7	j es considerablemente más importante que k
9	j es absolutamente más importante que k

Figura 3-2 Valores de preferencia para el método AHP

Una vez se haya constituido la matriz A, se debe normalizar la matriz, dividiendo cada valor por la suma de su columna.

$$\hat{a}_{jk} = \frac{a_{jk}}{\sum_{i=1}^n a_{ik}}$$

Por último, para obtener los pesos de cada criterio, se calcula el valor medio de cada fila de la matriz A normalizada. Como cada columna está asociada a un criterio, el valor medio de la columna que corresponda será el peso que se debe asignar para ese criterio.

$$w_j = \frac{\sum_{i=1}^n \hat{a}_{ik}}{n}$$

3.2.1 Particularización a nuestro caso

En nuestro caso, el orden de preferencia que se va a seguir es el siguiente:

$$\text{Satisfacción de la demanda} > \text{Impacto Medioambiental} > \text{Coste} > \text{Pérdidas}$$

La satisfacción de la demanda se ve representada con la diferencia entre el nivel de hidrógeno inicial y final y con la diferencia entre el estado inicial y final de la batería. Para no hacer que la satisfacción de la demanda adquiera un peso excesivamente elevado, se va a suponer para la obtención de los pesos que el criterio es únicamente satisfacción de la demanda y no los dos correspondientes al nivel de hidrógeno y al estado de la batería, para luego ponderar este peso entre el estado del tanque de hidruros y el estado de la batería.

En este aspecto, se va a suponer que el estado del tanque tiene una mayor importancia que el de la batería, ya que la batería es fácilmente recargable desde la red. Por tanto, vamos a suponer que el 70% del peso de la satisfacción de la demanda se corresponde con el estado del tanque por el 30% que corresponde al estado de la

batería.

La matriz de comparación que responde a las importancias relativas es la siguiente:

	<i>Satisfs. Demanda</i>	<i>Pérdidas</i>	<i>Coste</i>	<i>Impacto MA</i>
<i>Satisfs. Demanda</i>	1	5	4	3
<i>Pérdidas</i>	0,2	1	0.5	0.25
<i>Coste</i>	0,25	2	1	0.3333
<i>Impacto MA</i>	0,3333	4	3	1

Figura 3-3 Matriz de comparación AHP

Una vez se calcula la matriz normalizada queda:

	<i>Satisfs. Demanda</i>	<i>Pérdidas</i>	<i>Coste</i>	<i>Impacto MA</i>
<i>Satisfs. Demanda</i>	0,5607	0,4167	0,4706	0,6545
<i>Pérdidas</i>	0,1121	0,0833	0,0588	0,0545
<i>Coste</i>	0,1402	0,1667	0,1176	0,0727
<i>Impacto MA</i>	0,1869	0,3333	0,3529	0,2182

Figura 3-4 Matriz normalizada AHP

Ahora por último se calcula el valor medio de cada fila y se obtienen los pesos para cada criterio, teniendo en cuenta que satisfacción de la demanda engloba dos criterios:

<i>Satisfs. Demanda</i>	<i>Pérdidas</i>	<i>Coste</i>	<i>Impacto MA</i>
0,525637	0,0772	0,1243	0,2728

Figura 3-5 Pesos para cada criterio

Como el criterio satisfacción de la demanda está dividido en dos subcriterios, queda como sigue:

$$w_1 = 0,1577; w_2 = 0,3679; w_3 = 0,0772; w_4 = 0,1243; w_5 = 0,2728$$

El siguiente paso será el estudio de las alternativas por el método Promethee brevemente descrito y mediante una interfaz gráfica que establezca la posibilidad de interactuar con el método de decisión intuitivamente. Para la implantación del código Promethee se usará el código creado en el documento << **TFG DE IMPLM. PROMETHEE**>> efectuando las modificaciones necesarias para ajustarlo a nuestro caso.

El próximo apartado explicará el uso de una interfaz gráfica en Matlab y la interfaz gráfica propuesta en este trabajo. Esta interfaz dará la posibilidad de modificar estas condiciones de partida impuestas.

4 IMPLEMENTACIÓN DE INTERFAZ GRÁFICA EN MATLAB (GUI)

La interfaz gráfica en Matlab permite la interacción del usuario con una expresión gráfica a partir de la cual se pueden modificar datos, representar gráficas o hacer más sencillo el modo de modificar parámetros o variables de un problema.

En este apartado se mostrará la interfaz gráfica propuesta para poder modificar valores fácilmente de un problema de decisión multicriterio y obtener una solución adecuada.

En el Anexo I se explica cómo crear y manejar la interfaz gráfica de Matlab, además de un ejemplo.

4.1 Interfaz gráfica propuesta

La interfaz gráfica propuesta va a permitir la escritura de los valores de cada criterio en Excel para las distintas alternativas, la modificación de los parámetros de evaluación para el método Promethee y la evaluación y representación de las alternativas para este método.

La interfaz queda como sigue:

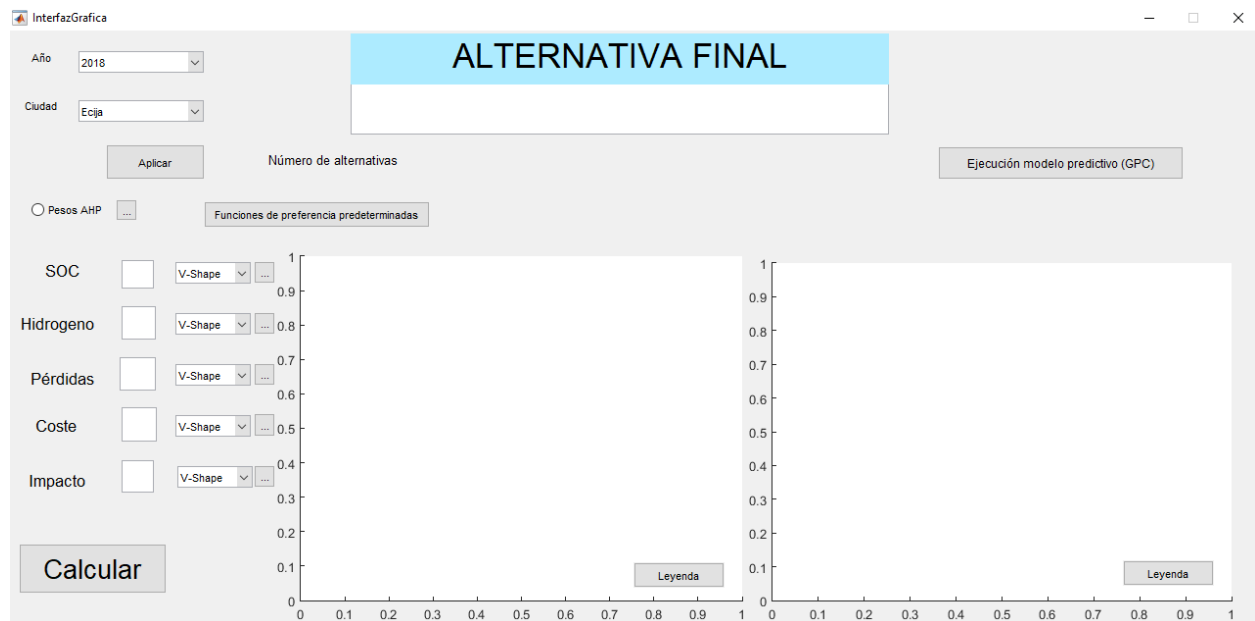


Figura 4-1 Interfaz gráfica propuesta

Se va a explicar las distintas opciones que se tienen para los cuatro grandes grupos en que se divide la interfaz.

4.1.1 Generación de alternativas

En la esquina superior izquierda se da la posibilidad de modificar la localidad y el año de evaluación. La localidad determinará los datos climatológicos promedio, de viento y de irradiación, que serán usados para el cálculo de la curva de generación, mientras que el año de evaluación va a modificar el precio unitario de la generación.

Los valores de irradiación y de viento promedios se encuentran en los archivos '*Fotovoltaica_multicriterio.m*' y '*Eolica_multicriterio.m*' y en '*Fotovoltaica.m*' y '*Eolica.m*'. Los archivos correspondientes a la fotovoltaica contienen los datos de irradiación y los de la eólica los datos de viento. Si se desea imponer una curva de

irradiación o de velocidad del viento diferente referente a un periodo de tiempo de un día, se debe modificar el vector de 24 valores que se correspondan, o bien el vector 'wind' o bien 'irradnu'. El caso de evaluación que se está realizando en este trabajo es aquel que evalúa el caso promedio. Otra posibilidad sería, por ejemplo, ver cual sería el resultado si se impone el caso en que tanto irradiación como velocidad del viento son máximas.

Por otro lado, en el archivo 'Escritura_datos_excel.m' se da la posibilidad de modificar las potencias a evaluar para los distintos elementos de la microgrid (generación, electrolizador, pila de combustible y demanda doméstica).

A continuación, se muestra, detalladamente, el código de dicho archivo que se usará para la imposición de las distintas alternativas que se quieran evaluar. Cabe aclarar que se pueden imponer tantas alternativas como sean necesarias en cada uno de los equipos.

La modificación del estado inicial se realiza a partir del siguiente código. En este caso, se evalúan un estado inicial del 80%, del 70% y del 60%.

```
%SOC inicial
Porcentaje_SOC=[80 70 60];
NumSOC=length(Porcentaje_SOC);
```

Para la modificación de la potencia nominal de la generación, se hará uso del siguiente código. La modificación de este parámetro modifica la curva de la generación, ya que esta estará de algún modo ponderada al caso inicial, donde la generación fotovoltaica tiene una potencia nominal de 6 kW y la eólica de 5 kW. En el caso establecido como ejemplo, se evalúa para el caso de la fotovoltaica una potencia nominal de 6 kW y otra de 3 kW y en el caso de la eólica una potencia nominal de 5 kW y otra de 3 kW.

```
%Potencia de la generación
kWPVTot=[6 3];
NumPotPV=length(kWPVTot);
kWEolTot=[5 3];
NumPotEol=length(kWEolTot);
```

El siguiente código especifica la demanda doméstica. Aquí se podrá imponer un porcentaje de la demanda a satisfacer. Se supone que la satisfacción del 100% de la demanda doméstica cubriría con todas las necesidades domésticas. En este código se evalúa una demanda doméstica del 100% y otra del 80%. Esta modificación no modifica la forma de la curva de demanda doméstica, ya que los multiplicadores no varían.

```
%Potencia máxima de la demanda
Porcentaje_demanda=[100 80];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
```

Por último, para la modificación de los parámetros del circuito de hidrógeno, con el siguiente código se impone la potencia nominal de electrolizador y pila de combustible. Esta variación influirá lógicamente en el flujo de hidrógeno que es generado o consumido. Además, también influirá en las pérdidas generadas por ambos elementos. En este caso se evalúan un electrolizador con potencia nominal de 3 kW y otro de 1,5 kW, así como una pila de combustible de 1,5 kW y otra de 1 kW.

```
%Potencia del EZ
kWEZTot=[3 1.5];
NumEZ=length(kWEZTot);
```

```

%Potencia de la FC
kWFCTot=[1.5 1];
NumFC=length(kWFCTot);

```

Una vez se escoge la localidad y el año a evaluar, y al haber impuesto ya las combinaciones que se desean evaluar a partir del código anteriormente comentado, al pulsar sobre el botón 'Aplicar', se realizará la escritura de los datos en el documento Excel 'Evaluacion Microgrid.xls' de todos los valores para cada criterio y alternativa, mostrando en pantalla el número de alternativas que se van a evaluar. Estos valores serán los correspondientes a la localidad escogida, con un coste de la generación establecido a partir del año escogido y con las distintas potencias impuestas para cada uno de los elementos. A partir de un bucle que evalúa, para cada energía renovable, todas las combinaciones posibles entre las distintas potencias de los elementos y los estados iniciales de la batería, se hace un recuento que dará el número de alternativas que se están evaluando.

Para el código descrito en la página anterior como ejemplo, la generación de las distintas alternativas viene ejemplificada en el siguiente esquema, la figura 4-2, donde se representa cómo se han generado las alternativas correspondientes a ambas generaciones.

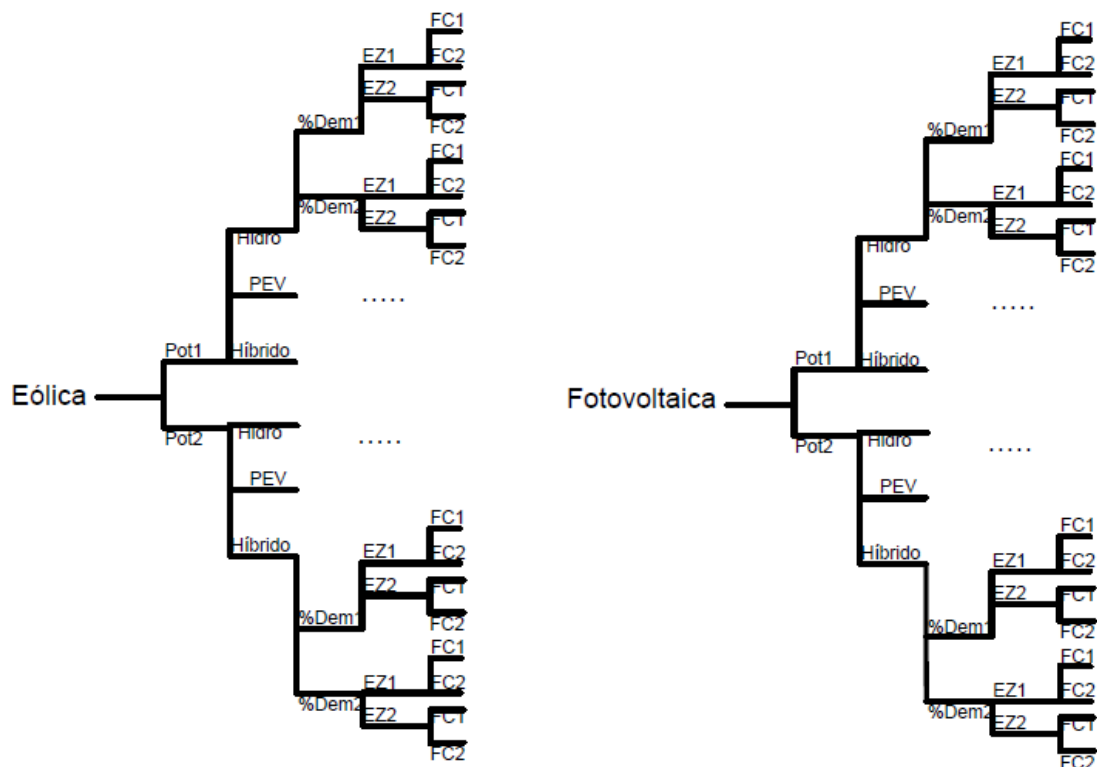


Figura 4-2 Representación generación de las alternativas

Aunque el archivo Excel es necesario para la evaluación de las distintas alternativas, no es necesario abrir dicho archivo manualmente para nada, ya que la interfaz recoge los datos directamente de dicho archivo. Una vez se haya realizado la escritura de datos, se mostrará por pantalla la ventana representada en la figura 4-3.

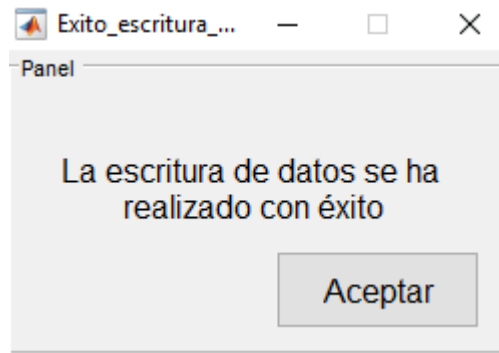


Figura 4-3 Escritura de datos con éxito

Cabe mencionar que se pueden introducir tantas alternativas como se quieran, teniendo en cuenta que el tiempo de respuesta será mayor y que en la representación de los métodos Promethee I y II será más difícil reconocer las distintas alternativas.

Una vez se imponen los datos de la Microgrid tal como se quiere evaluar, se procede a la imposición de los parámetros de evaluación.

4.1.2 Modificación de los parámetros de evaluación

Los criterios que van a determinar la evaluación son fijos, y para excluir alguno de los criterios que se están evaluando, se debe de imponer un peso de 0 a ese criterio. Como se comentó en el apartado 3, los criterios que se han seguido para la evaluación de las distintas alternativas son los siguientes: estado de la batería (kWh), nivel de hidruros (Nm^3), coste de la generación (\$), pérdidas en el circuito (kWh) e impacto medioambiental (cualitativo).

En cuanto a los pesos, se dan dos opciones: la imposición manual de los pesos o el uso del método AHP. En este último caso, se debe activar la opción de '*Pesos AHP*' y pulsar el botón que le acompaña. Se abrirá la ventana mostrada a continuación, que da la opción de introducir la matriz AHP que impone la importancia relativa de un criterio respecto a otro, pudiendo imponer estas importancias relativas bien manualmente o bien recogiendo unas condiciones predeterminadas que han sido ya impuestas y que se muestran en la figura 4-4.

Como se indicó anteriormente, los criterios que recogen el estado de la batería y el nivel de hidruros serán tomados como uno, ya que ambos evalúan la satisfacción de la demanda. Por tanto, el peso que se obtenga para el criterio "SOC/Hidro" será la suma del criterio que evalúa el estado de la batería y del criterio que evalúa el nivel de hidruros (la cantidad de hidrógeno será el 70% de este peso, mientras que el estado de la batería supondrá el 30%, debido a la facilidad de recargar la batería a partir de la red).

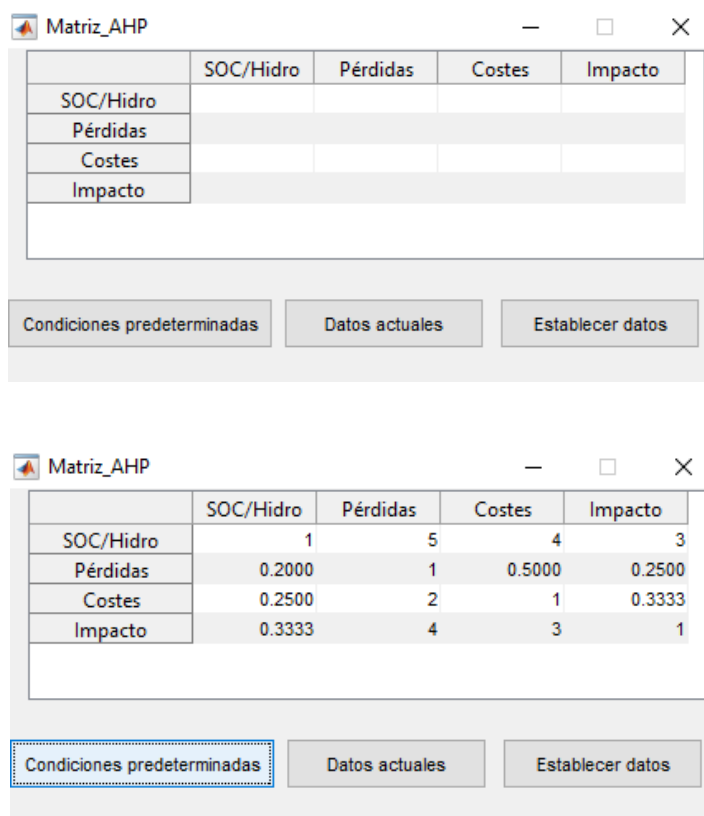


Figura 4-4 a) Ventana de edición matriz AHP. b) Valores predeterminados matriz AHP

Las condiciones predeterminadas establecen una importancia global entre criterios como sigue:

$$\text{Satisfacción demanda} > \text{Impacto} > \text{Costes} > \text{Pérdidas}$$

Además, al pulsar el botón 'Datos actuales', se muestra los valores que se encuentran impuestos. Hay que tener en cuenta que, la primera vez que se pulse la opción 'Pesos AHP', esta matriz se encontrará vacía y habrá que establecer los datos obligatoriamente.

En cuanto a las funciones de preferencia para cada criterio, también existen unas funciones de preferencia predeterminadas, que se muestran e imponen una vez se pulsa sobre el botón 'Funciones de preferencia predeterminadas'. Una vez impuestas estas funciones de preferencia imponiendo la función de preferencia que se desee a partir del menú desplegable y del botón que le acompaña, que mostrará una ventana para recoger los valores de p , q o s según corresponda y permitirá ver los valores que se encuentran en ese momento impuestos. Hay que tener en cuenta que en primera estancia las funciones de preferencia no se encuentran determinadas hasta que se hace pulsado el botón previamente dicho o se establecen manualmente. Además, al modificar la función de preferencia para un criterio, se resetean los valores de p , q y s que estuviesen determinados. Las funciones de preferencia impuestas serán las establecidas en el apartado 3.

Como puede observarse en la figura 4-5, la modificación de cada una de las funciones de preferencia se realiza mediante el menú desplegable. Una vez seleccionada la función deseada, el botón '...' muestra los valores que es necesario insertar y la posibilidad de mostrar el valor que se encuentra impuesto. Por ejemplo, en la figura 4-6 se muestra la ventana emergente que aparecerá si se hace click en el botón del criterio 'SOC', que muestra tanto la necesidad de insertar los valores de p y q .

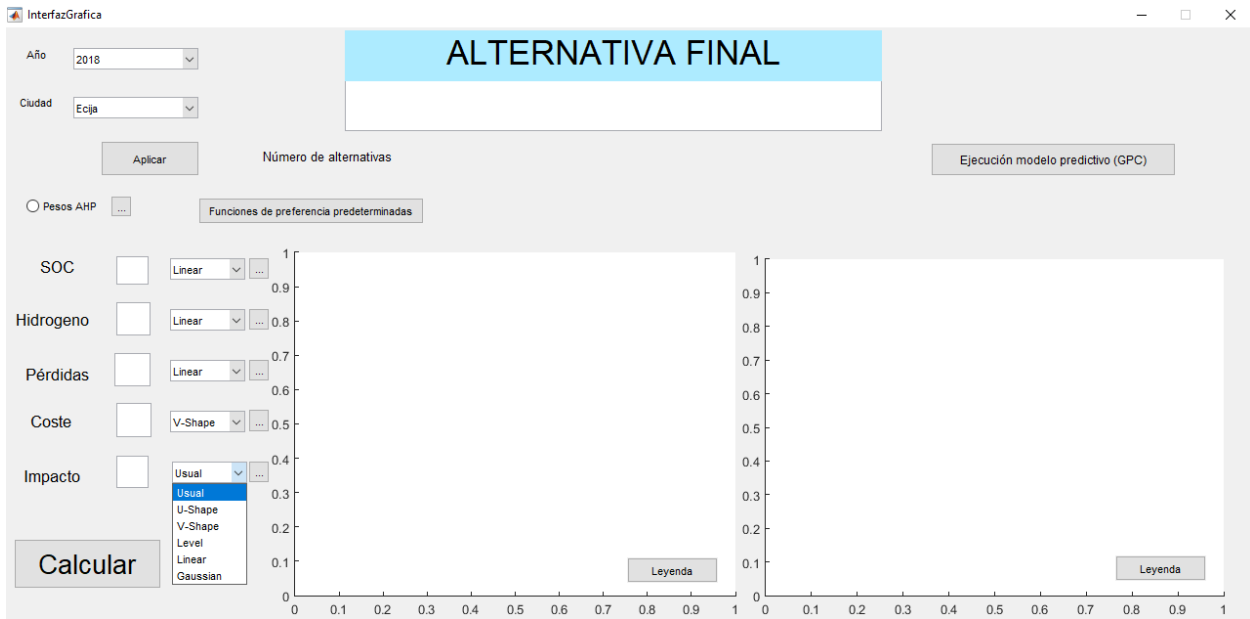


Figura 4-5 Funciones de preferencia predeterminadas

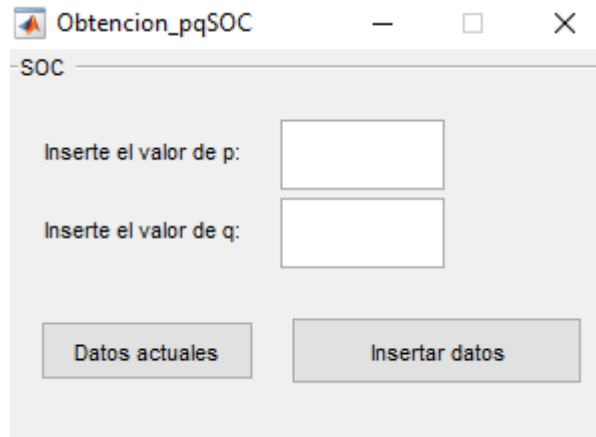


Figura 4-6 Ventana emergente criterio SOC

4.1.3 Evaluación y representación de las alternativas

Por último, una vez se ha realizado la escritura de los datos y que se han impuesto los parámetros de evaluación, al pulsar en el botón ‘Calcular’, se mostrará una representación de los flujos positivos y negativos para el método Promethee I (figura 1) y de los flujos netos para el método Promethee II (figura 2). Además, se indicará la alternativa que presenta mejores resultados para las condiciones impuestas, indicando la potencia de generación y de los distintos componentes, además del vehículo escogido, como, por ejemplo, ‘Fotovoltaica 6+PEV+FC 1.5+EZ 3+Dom 100%’ indica que la generación escogida es la fotovoltaica con potencia nominal de 6kW, el vehículo es eléctrico, la pila de combustible tiene una potencia nominal de 1,5kW y el electrolizador de 3kW y la demanda doméstica está al 100% de su potencia máxima.

A la hora de la elección de la mejor alternativa, en primer lugar, se comprueba la alternativa que es mejor que un mayor número de alternativas tanto para el flujo positivo como para el negativo (método Promethee I). En el caso de que existan varias alternativas que sean más adecuadas que un mismo número de alternativas, se observará la opción con un flujo neto superior. Una vez se ha realizado la ordenación y se ha llegado a una alternativa óptima, se muestra por pantalla la ventana que se muestra en la figura 4-7.

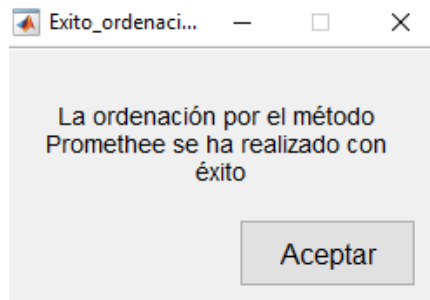


Figura 4-7 Ordenación método Promethee con éxito

4.1.4 Ejecución del modelo predictivo (GPC)

Al pulsar el botón '*Ejecución modelo predictivo (GPC)*' se mostrará por pantalla dos figuras, representando el modelo predictivo explicado en el apartado 5.

El código de esta interfaz ha sido incluidos en el Anexo.

5 CONTROL PREDICTIVO BASADO EN MODELO (MPC)

El control predictivo basado en modelo se trata de un enfoque de control basado en ordenador que usa un modelo matemático para simular la evolución del sistema y calcula las entradas óptimas de acuerdo con una función objetivo, para un horizonte de predicción N . Permite lidiar con situaciones complejas como restricciones, perturbaciones, retrasos, etc.

Los controladores predictivos se basan en las siguientes ideas:

- Uso explícito de un modelo para predecir la salida del proceso en futuros instantes de tiempo.
- Cálculo de las señales de control minimizando una cierta función objetivo.
- Estrategia deslizante, de forma que en cada instante el horizonte se va desplazando hacia el futuro, lo que implica la aplicación de la primera señal de control en cada instante y desechar el resto, repitiendo el cálculo en cada instante de muestreo.

Actualmente, el control predictivo cuenta con numerosas aplicaciones tanto en la industria de procesos como en control de motores o en Robótica, consiguiendo sistemas de control de elevadas prestaciones operando prácticamente sin intervención durante largos periodos de tiempo.

5.1 Estrategia de control

Los controladores pertenecientes al MPC siguen el siguiente método:

1. Predicción de las futuras salidas en cada instante t para un horizonte temporal N , haciendo uso del modelo del proceso. Como muestra la siguiente figura, las salidas predichas $\hat{y}(t+k | t)$ para $k=1 \dots N$ dependen de los valores ya conocidos para el instante t y de señales de control futuras que ya han sido calculadas: $u(t+k | t)$, $k=0 \dots N-1$.

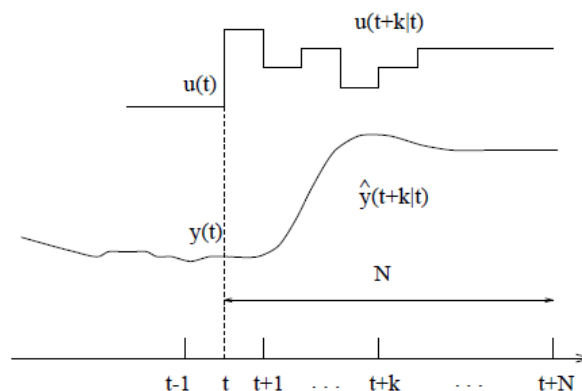


Figura 5-1 Evolución controladores MPC

2. El cálculo de las señales de control futuras optimizará un criterio que se pretende mantener lo más cerca posible de una trayectoria de referencia $w(t+k)$. Dicho criterio suele ser una función cuadrática de los errores entre la salida predicha y la trayectoria de referencia, también predicha. En la siguiente figura se puede ver la tendencia que sigue el control predictivo siguiendo la trayectoria de referencia.

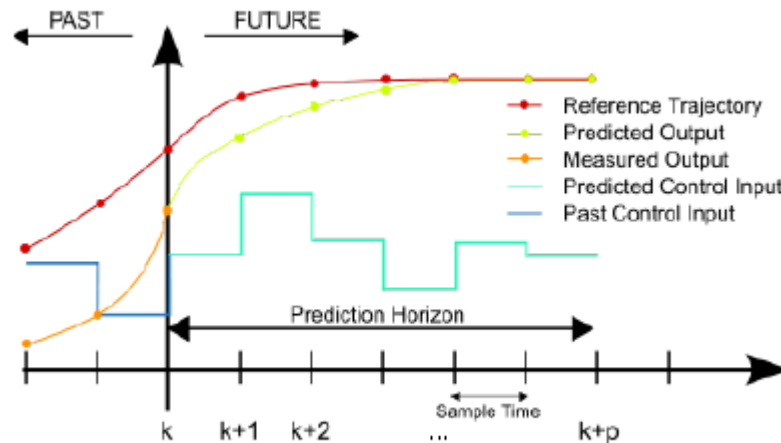


Figura 5-2 Trayectoria de referencia

- Una vez enviada la señal de control $u(t | t)$, se desechan las siguientes señales de control calculadas, puesto que en el siguiente instante ya se conoce $y(t+1)$ y se puede repetir el paso 1, ahora se calcula para $u(t+1 | t+1)$, que será más exacto que $u(t+1 | t)$. Este concepto se conoce como horizonte deslizante, ya que para cada instante t del horizonte temporal, se calculan un número N de señales de control, de las cuales se descartan todas menos la primera.

Para llevar a cabo esta metodología, se sigue la siguiente estructura:

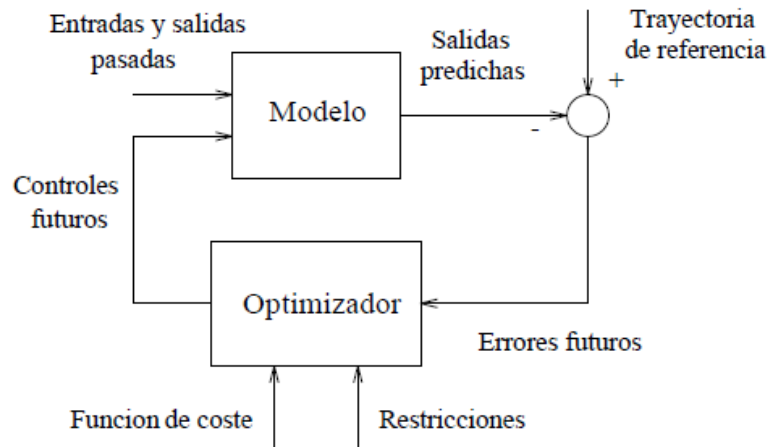


Figura 5-3 Método de control MPC

Como puede observarse, los elementos más importantes son el modelo, que representa el sistema, y el optimizador, que calcula las señales de control futuras de acuerdo con una función objetivo o de coste y a ciertas restricciones impuestas.

Se van a comentar los distintos elementos comunes que están presentes en los controladores predictivos. Estos son: modelo de predicción, función objetivo y obtención de la ley de control.

5.1.1 Modelo de predicción

Debe ser capaz de determinar la dinámica del proceso y de permitir el cálculo de las predicciones. Este modelo

puede ser determinado de diversas maneras, siempre observando la relación de las salidas con las entradas medibles, ya sean variables manipuladas o perturbaciones medibles.

Este modelo de predicción se divide en dos partes: el modelo del proceso, que establece una relación salidas-entradas medibles, y el modelo de las perturbaciones, que intentará evaluar las entradas no medibles, el ruido y posibles errores de modelado.

5.1.1.1 Modelo del proceso

Las formas más usadas para modelar un proceso son las siguientes:

- Modelo por respuesta impulsional: se obtienen los valores de salida al imponer un impulso como valor de entrada, considerándose N valores.

$$y(t) = \sum_{i=1}^N h_i u(t - i) = H(z^{-1})u(t)$$

, h_i son los valores obtenidos al someter al proceso a un impulso y $H(z^{-1})=h_1z^{-1}+h_2z^{-2}+\dots+h_Nz^{-N}$ representa el retardo temporal. La salida predicha vendrá dada por:

$$\hat{y}(t + k|t) = \sum_{i=1}^N h_i u(t + k - i|t) = H(z^{-1})u(t + k|t)$$

Este método no requiere de información adicional del proceso, pero sí de un gran número de parámetros ya que N suele ser 40 o 50.

- Modelo por respuesta ante escalón: ahora la señal es un escalón.

$$y(t) = y_0 + \sum_{i=1}^N g_i \Delta u(t - i) = y_0 + G(z^{-1})(1 - z^{-1})u(t)$$

, g_i son los valores obtenidos al someter al proceso a un escalón y $\Delta u(t)=u(t)-u(t-1)$. La salida predicha vendrá dada por:

$$\hat{y}(t + k|t) = \sum_{i=1}^N g_i \Delta u(t + k - i|t)$$

La siguiente figura muestra ambos métodos:

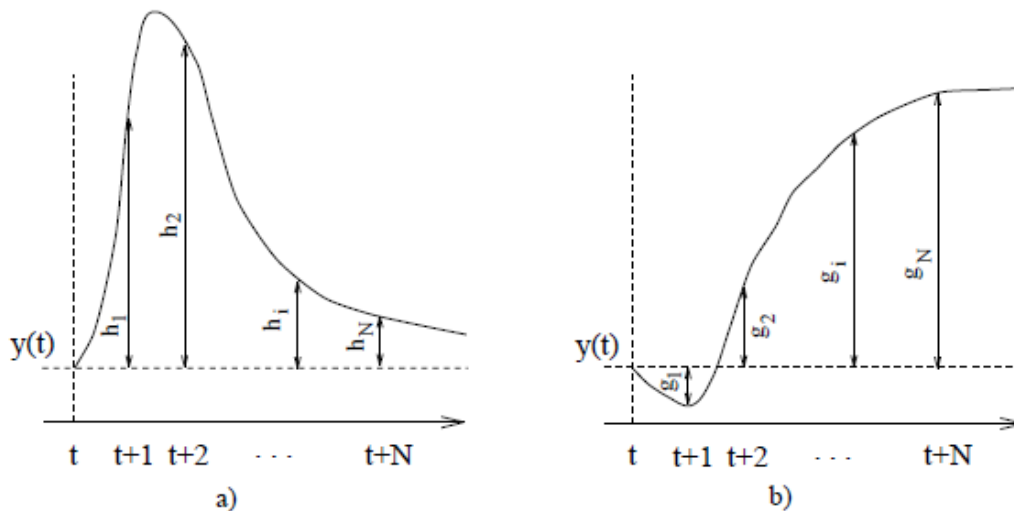


Figura 5-4 a) Modelo por respuesta impulsional. b) Modelo por respuesta ante escalón

- Modelo por función de transferencia: la función de transferencia $G=B/A$ vendrá dada por:

$$A(z^{-1})y(t) = B(z^{-1})u(t)$$

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_naz^{-na}$$

$$B(z^{-1}) = b_1z^{-1} + b_2z^{-2} + \dots + b_nbz^{-nb}$$

, y la predicción vendrá dada por:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})}u(t+k|t)$$

En principio, es necesario un conocimiento previo del proceso sobre todo en cuanto al orden de los polinomios A y B .

- Modelo por espacio de estados: se representa como sigue:

$$x(t) = Ax(t-1) + Bu(t-1)$$

$$y(t) = Cx(t)$$

, siendo x el estado del sistema y A , B y C las matrices de entrada (A , B) y de salida (C) del sistema. La predicción viene dada por:

$$\hat{y}(t+k|t) = C\hat{x}(t+k|t) = C[A^kx(t) + \sum_{i=1}^k A^{i-1}Bu(t+k-i|t)]$$

5.1.1.2 Modelo de las perturbaciones

El modelo mayormente usado es el Autorregresivo Integrado de Media Móvil (ARIMA). En él, las perturbaciones (diferencias entre la salida medida y la calculada por el modelo) se calculan como sigue:

$$n(t) = \frac{C(Z^{-1})e(t)}{D(Z^{-1})}$$

, donde $D(z^{-1})$ incluye el integrador $\Delta = 1 - z^{-1}$, $e(t)$ es un ruido de media cero y el polinomio C suele considerarse 1. Cabe mencionar que el uso de un integrador hace que el error en régimen permanente sea nulo.

5.1.1.3 Respuestas libre y forzada

La idea del uso de los términos libre y forzado es el uso de la secuencia de acciones de control como la suma de dos señales: $u(t) = u_f(t) + u_c(t)$.

El término $u_f(t)$ se refiere a las entradas pasadas, y en el futuro se mantiene constante e igual al último valor de la variable manipulada. Así pues:

$$u_f(t-j) = u(t-j), \text{ para } j = 1, 2, \dots$$

$$u_f(t+j) = u(t-1), \text{ para } j = 0, 1, 2, \dots$$

Por otro lado, $u_c(t)$ se refiere a las señales de control en instantes futuros:

$$u_c(t-j) = 0, \text{ para } j = 1, 2, \dots$$

$$u_c(t+j) = u(t+j) - u(t-1), \text{ para } j = 0, 1, 2, \dots$$

A partir de ambas señales de control se puede observar una respuesta libre y otra forzada del sistema. La primera, $y_f(t)$ es la respuesta que corresponde al momento en que la variable manipulada obtiene el valor de $u_f(t)$. En cuanto a la respuesta forzada, $y_c(t)$, se corresponde con la señal de control $u_c(t)$. La respuesta libre es aquella que llevaría el proceso debido a la situación actual de las señales de control y la forzada es debida a señales de control futuras. En la siguiente figura se muestra un esquema de estos conceptos.

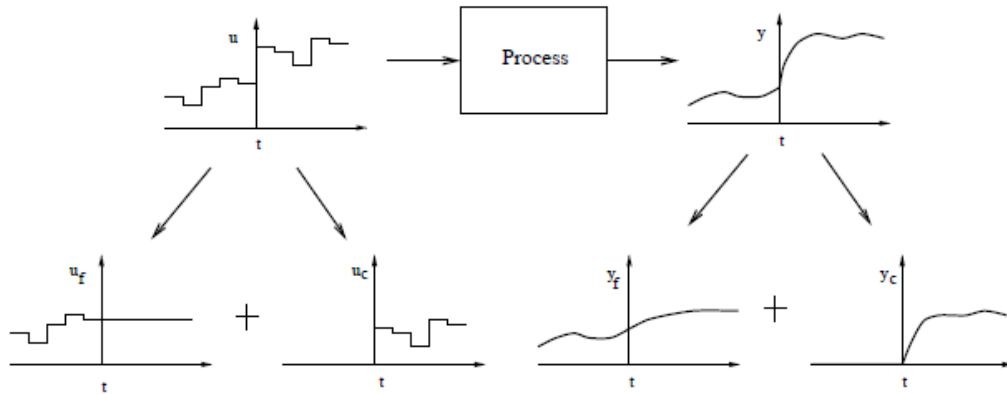


Figura 5-5 Respuestas libre y forzada

5.1.2 Función objetivo

Los valores futuros de control se calculan para que la predicción de la salida siga a la trayectoria de referencia. Desde un punto de vista matemático, esto quiere decir que los errores de predicción (trayectoria de referencia – predicción salida) sean mínimos. La expresión general cuadrática de esta función suele ser la siguiente:

$$J(N_1, N_2, Nu) = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{Nu} \lambda(j) [\Delta u(t+j-1)]^2$$

Se pueden considerar:

- Parámetros: N_1 y N_2 son los horizontes mínimo y máximo de coste y Nu es el horizonte de control. Los parámetros N_1 y N_2 marcan los límites en que se quiere que la salida siga a la referencia. Si se tiene un valor de N_1 grande, no importa que se obtengan errores en los primeros instantes y se obtendrá una respuesta suave del proceso.

Tanto δ como λ ponderan el comportamiento futuro. Suelen ser valores constantes o secuencias exponenciales. Por ejemplo, es común usar un puso exponencial $\delta(j)$ como sigue:

$$\delta(j) = \alpha^{N_2-j}$$

Si $0 < \alpha < 1$, el control será suave y con menor esfuerzo. Si $\alpha > 1$, se penalizan más los primeros errores y el control es más brusco.

- Trayectoria de referencia: con el fin de reaccionar antes de un cambio, el conocimiento de la evolución futura de la referencia es una gran ventaja presente en el control predictivo, evitando los efectos del retardo en la respuesta del proceso. Con el fin de llegar al estado de referencia futura, se suele usar la trayectoria de referencia $w(t+k)$ que se aproximará suavemente desde el valor actual de la salida $y(t)$ al valor de referencia real.

$$w(t) = y(t) ; w(t+k) = \alpha w(t+k-1) + (1-\alpha)r(t+k), k = 1 \dots N$$

Como se ha comentado antes, dependiendo del valor de α , la respuesta será más suave o menos. Puede verse en esta ecuación que la trayectoria de referencia dependerá del valor de referencia actual más la referencia a la que se quiere llegar a partir de la elección del parámetro α que impondrá la trayectoria de referencia. En la siguiente figura se muestran dos trayectorias de referencia, donde la trayectoria w_2 tiene un valor de α mayor que la trayectoria w_1 .

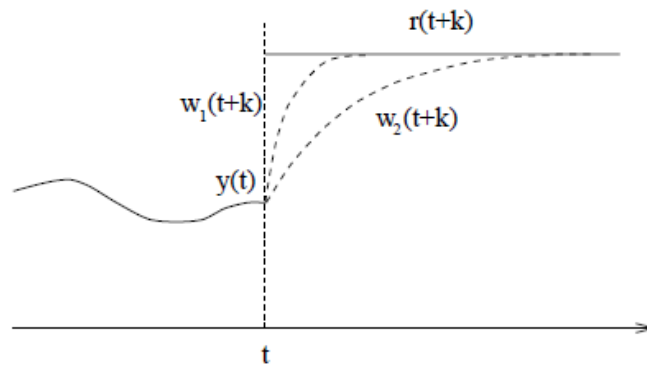


Figura 5-6 Trayectorias de referencia para distintos valores de α

- Restricciones: por lo general, se considerarán límites en la señal de control y en las salidas:

$$u_{min} \leq u(t) \leq u_{max}, \forall t$$

$$du_{min} \leq u(t) - u(t-1) \leq du_{max}, \forall t$$

$$y_{min} \leq y(t) \leq y_{max}, \forall t$$

5.1.3 Obtención de la ley de control

Para la obtención de la ley de control, $u(t+k | t)$, se necesitará minimizar la función de coste. Para ello, se deben calcular las salidas predichas $\hat{y}(t+k | t)$ a partir de valores pasados y salidas y señales de control futuras, usando el modelo elegido y se sustituyen en la función de coste. Se obtendrá, por tanto, una expresión, cuya minimización dará los valores buscados. Si el modelo es lineal, puede obtenerse una solución analítica, mientras que en otro caso, se debe usar un método iterativo de optimización.

Existirán $N_2 - N_1 + 1$ variables independientes. Para reducir el número de grados de libertad, se propone una cierta ley de control estructurada. Esta ley de control estructurada evita la libre evolución de variables manipuladas (sin estructurar) puede llevar a señales de control de alta frecuencia indeseadas que pueden conducir a la inestabilidad.

Esta estructura de la ley de control se basa en el uso del parámetro Nu , horizonte de predicción, a partir de cuyo valor no se plasma ninguna variación en las señales de control hasta el momento N_2 .

$$\Delta u(t+j-1) = 0 \quad j > Nu$$

5.2 Control Predictivo Generalizado (GPC)

Se basa en minimizar una función cuadrática que mide tanto la distancia entre la salida predicha del sistema y una cierta trayectoria de referencia hasta el horizonte de predicción como el esfuerzo de control necesario para obtener dicha salida.

Este control predictivo va a ser el que se implante en la Microgrid propuesta una vez se escoja una alternativa óptima.

El modelado ha sido implantado por la profesora tutora de este proyecto, Ascensión Zafra Cabeza.

5.2.1 Formulación del Control Predictivo Generalizado

Este algoritmo se basa en el modelo de proceso de función de transferencia y en el modelo de las perturbaciones ARIMA. El modelo que va a modelar la dinámica del proceso (al ser un proceso de una entrada y una salida,

SISO), es conocido como el modelo Autorregresivo de Media Móvil Integrado (CARIMA):

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})\frac{e(t)}{\Delta}$$

, donde $\Delta=1-z^{-1}$, $u(t)$ e $y(t)$ son la señal de control y salida del proceso, $e(t)$ es un ruido blanco de media cero y d es el tiempo muerto del sistema. A , B y C son los siguientes polinomios de desplazamiento hacia atrás:

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na}$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb}$$

$$C(z^{-1}) = 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{nc}z^{-nc}$$

Por simplicidad, se va a tomar el polinomio C igual a 1.

La idea de este control es minimizar la función objetivo propuesta en el apartado 4.1.2, buscando un conjunto de señales de control $u(t)$, $u(t+1)$... que hagan que la salida futura del proceso $y(t+j)$ sea lo más próxima posible a la trayectoria de referencia $w(t+j)$.

5.2.1.1 Predicción óptima

En este punto se va a buscar la predicción óptima de $y(t+j)$ para $j \geq N_1$ y $j \leq N_2$. Considerando la siguiente ecuación:

$$1 = E_j(z^{-1})\Delta A + z^{-j}F_j(z^{-1})$$

, y combinándola con la ecuación del modelo CARIMA, operando según el texto de referencia << CONTROL PREDICTIVO >>, se llega a la siguiente predicción óptima:

$$\hat{y}(t+j|t) = G_j(z^{-1})\Delta u(t+j-d-1) + F_j(z^{-1})y(t)$$

, donde $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$. Para la obtención de los (N_2-N_1) polinomios E_j y F_j , de grado $(j-1)$ y n_a respectivamente, para $j=N_1 \dots N_2$ se sigue el siguiente proceso iterativo:

1. $E_1 = 1$; $F_1 = z(1 - \tilde{A})$, con $\tilde{A} = (1 - z^{-1}) \cdot A$
2. $e_{j+1,j} = f_{j0}$
3. $f_{j+1,i} = f_{j,i+1} - f_{j0}\tilde{a}_{i+1}$, para $i = 0 \dots n_a$ ($f_{j,na+a} = 0$)

La demostración que establece que los polinomios E_j y F_j se pueden calcular según tal proceso, de forma que los valores del polinomio $j+1$ dependan del polinomio j se encuentra en el documento de referencia <<CONTROL PREDICTIVO >>.

Una vez calculados los polinomios E_j y F_j , los polinomios G_j también pueden ser calculados recursivamente, es decir, a partir del polinomio j :

$$G_{j+1} = E_{j+1}B = (E_j + f_{j,0}z^{-j})B = G_j + f_{j,0}z^{-j}B$$

$$g_{j+1,j+i} = g_{j,j+i} + f_{j,0}b_i, \text{ para } i = 0 \dots nb$$

Para el periodo de retardo d , la señal de control sólo tendrá repercusión a partir del instante $d+1$, y los horizontes pueden ser determinados como $N_1=d+1$, $N_2=d+N$ y $Nu=N$.

Las j predicciones óptimas pueden ser escritas de forma matricial como:

$$y = \mathbf{G}u + \mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t) = \mathbf{G}u + \mathbf{f}$$

Donde \mathbf{f} solo depende del pasado.

5.2.1.2 Obtención de la ley de control

La ecuación de la función de costes queda, por tanto:

$$J = (\mathbf{G}u + \mathbf{f} - \mathbf{w})^T (\mathbf{G}u + \mathbf{f} - \mathbf{w}) + \lambda u^T u$$

, donde \mathbf{w} es un vector que recoge las N trayectorias de referencia.

La función de costes puede escribirse como:

$$J = \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{b} \mathbf{u} + \mathbf{f}_0$$

, donde

$$\mathbf{H} = 2(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})$$

$$\mathbf{b} = 2(\mathbf{f} - \mathbf{w})^T \mathbf{G}$$

$$\mathbf{f}_0 = (\mathbf{f} - \mathbf{w})^T (\mathbf{f} - \mathbf{w})$$

Si no existen restricciones, el mínimo de J puede ser determinado igualando su gradiente a 0, llegando a:

$$\mathbf{u} = -\mathbf{H}^{-1} \mathbf{b}^T$$

5.2.1.3 Caso multivariable

Para el caso en que se tengan m entradas y n salidas, los polinomios $A(z^{-1})$ y $C(z^{-1})$ son matrices polinomiales de dimensión $n \times n$ y $B(z^{-1})$ de dimensión $n \times m$.

La minimización se realiza como en el caso monovariable, pero dando como resultado un vector de señales de control $u_1(t) \dots u_m(t)$.

Para el caso de la Microgrid de laboratorio, se hace uso del código '*modelo grid.m*' generado, como se ha dicho anteriormente, por la profesora tutora de este documento. Dicho código recibirá la demanda y la generación de la planta para la alternativa más adecuada obtenida a partir de la interfaz y hará uso del control predictivo para controlar la planta, en lugar del control por histéresis implantado inicialmente.

6 EVALUACIÓN MULTICRITERIO DE LA MICROGRID

En este capítulo se van a mostrar distintas alternativas que se van a evaluar a través de la interfaz gráfica creada y a partir de las cuales se pretende encontrar una alternativa óptima para cada una de las localidades y posibles mejoras o adecuaciones que se puedan llevar a cabo en la planta.

Se verá en primer lugar la opción más adecuada para cada localidad tanto en 2018 como en 2025 para las condiciones predeterminadas.

Como se explicó en el apartado 4, al imponer una localidad y un año en concreto, se especifica la curva de generación y el coste unitario de la generación. Así mismo, las potencias nominales de los elementos se podrán modificar a partir del archivo *'Escritura_datos_excel.m'*.

En el resto de los apartados se verá la alternativa más adecuada modificando las potencias nominales de generación, de electrolizador y pila de combustible y otras modificaciones como el estado inicial de la batería.

6.1 Evaluación para cada localidad modificando los pesos y las funciones de preferencia

En este apartado se va a evaluar a partir de la interfaz gráfica la mejor alternativa para cada localidad. Para ello, en el archivo *'Escritura_datos_excel.m'*, el código que define las características de cada elemento es el siguiente:

```
%SOC inicial
Porcentaje_SOC=[80];
SOC_inicial=Porcentaje_SOC/100*18.92;% 80% del máximo
%Potencia de la generación
kWPVTot=[6];
NumPotPV=length(kWPVTot);
kWEolTot=[5];
NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[3];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFCTot=[1.5];
NumFC=length(kWFCTot);
```

Dicho código es general para ambas localidades y para las distintas opciones de evaluación.

Ahora se verá cada localidad por separado y se impondrán distintas opciones de evaluación (funciones de preferencia y pesos).

6.1.1 Evaluación para la localidad de Écija

En primer lugar, se pulsa el botón de aplicar para la localidad de Écija y se impone la situación temporal que se va a evaluar. Se verá en un primer momento la situación actual y después la situación para 2025.

En el momento en que se haya realizado la escritura de todos los datos necesarios, se mostrará por pantalla el número de alternativas que se están evaluando. En este caso han resultado un total de seis combinaciones posibles: 3 vehículos para cada una de las energías renovables (eólica y fotovoltaica).

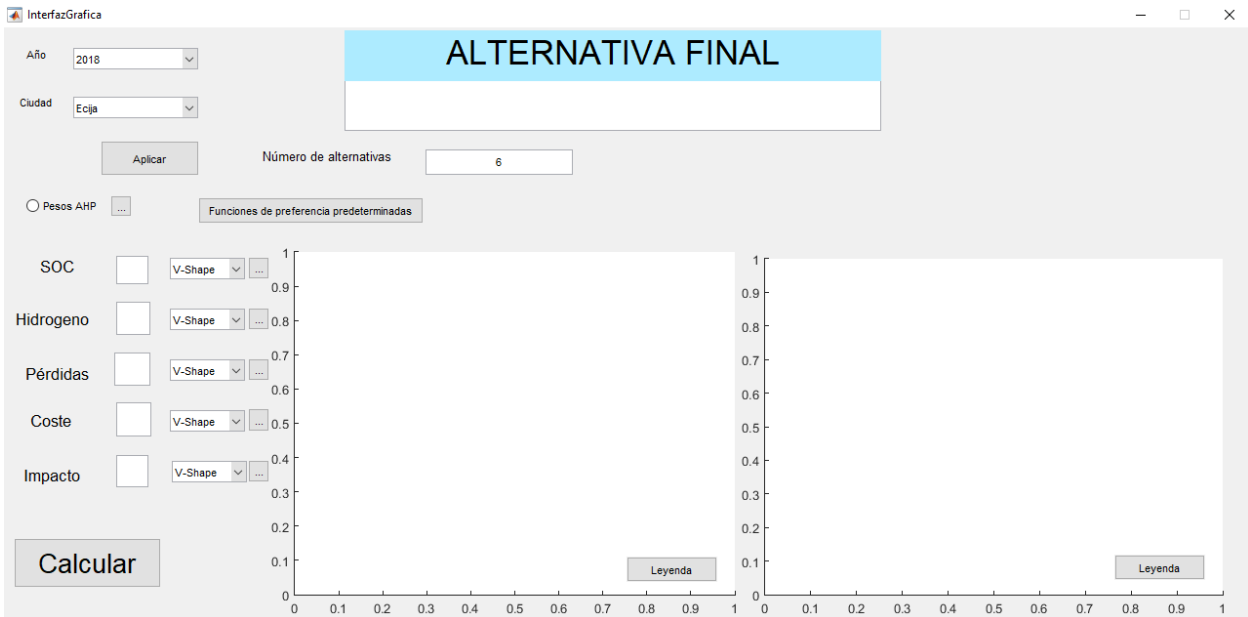


Figura 6-1 Escritura de datos para Écija para 2018

Una vez realizada la escritura, se procede a la imposición de las distintas condiciones de evaluación.

6.1.1.1 Funciones de preferencia y pesos predeterminados.

En este subapartado, la evaluación se verá para los pesos especificados en el apartado 3 obtenidos a partir del método AHP, y las funciones de preferencia serán también las predeterminadas.

Como se comentó en el apartado 3, al imponer los pesos mediante el método AHP, se establecen unas importancias relativas entre los distintos criterios. En la figura 6-2 se muestran dichas importancias relativas.

	<i>Satisfs. Demanda</i>	<i>Pérdidas</i>	<i>Coste</i>	<i>Impacto MA</i>
<i>Satisfs. Demanda</i>	1	5	4	3
<i>Pérdidas</i>	0,2	1	0.5	0.25
<i>Coste</i>	0,25	2	1	0.3333
<i>Impacto MA</i>	0,3333	4	3	1

Figura 6-2 Matriz de comparación AHP

Los pesos que corresponden a esta matriz de comparación son los siguientes:

$$w_1 = 0,1577; w_2 = 0,3679; w_3 = 0,0772; w_4 = 0,1243; w_5 = 0,2728$$

Al evaluar las alternativas correspondientes, se obtienen los siguientes resultados para ambas situaciones temporales. En la figura 6-2, se pueden observar los resultados para el año 2018 y en la figura 6-3 los resultados obtenidos para el año 2025.

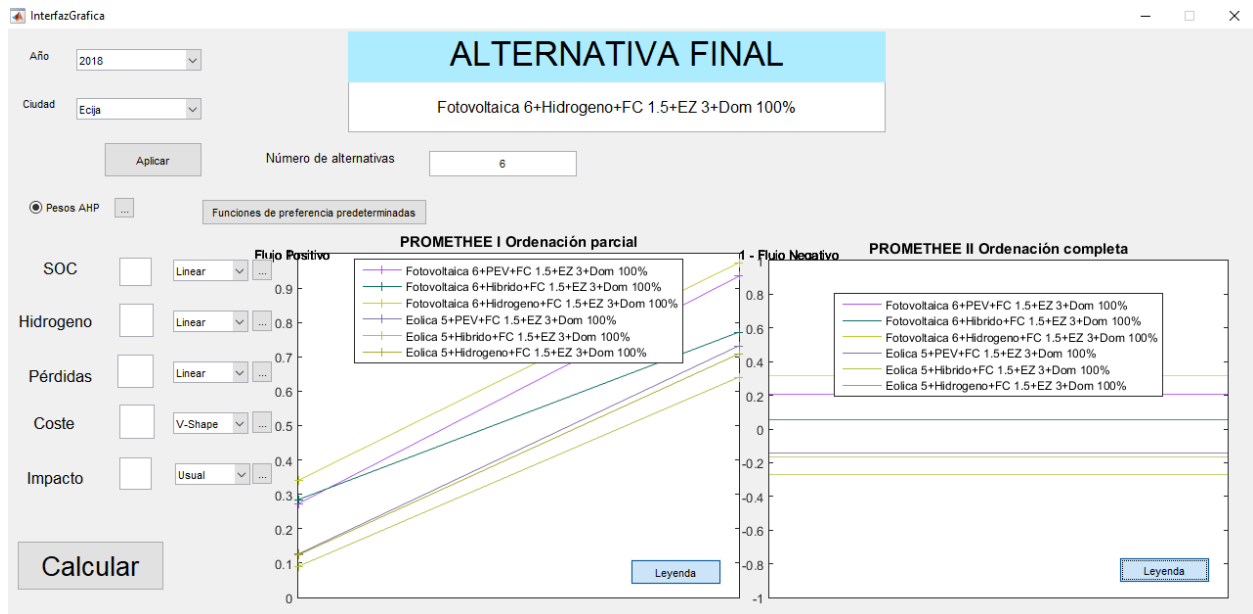


Figura 6-3 Evaluación de Écija para 2018

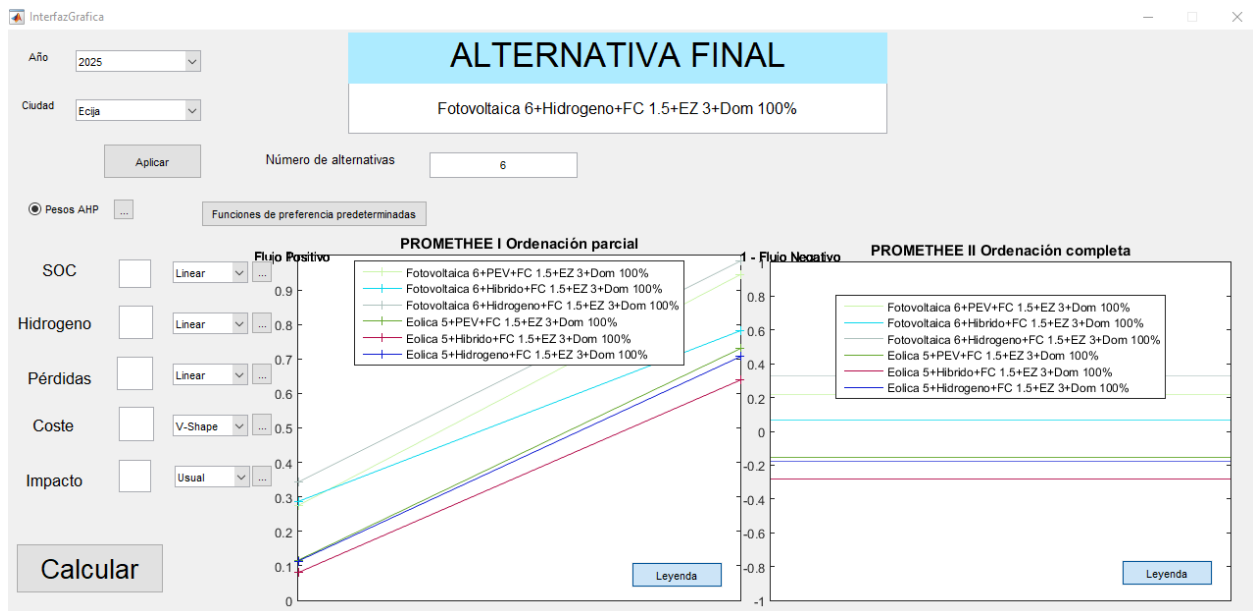


Figura 6-4 Evaluación de Écija para 2025

En ambas figuras, como se comentó en el apartado 4, se muestran dos gráficas que representan el método Promethee I y II, respectivamente. En la primera gráfica se representa una línea recta para cada alternativa, que une el flujo positivo con (1-Flujo negativo). Un flujo positivo elevado indicará que esa alternativa presenta mejores características para un mayor número de criterios, mientras que un valor bajo del flujo negativo será indicativo de que existe un mayor número de alternativas que superan a esa alternativa en distintos criterios. Por tanto, aquella alternativa que se encuentre en un lugar más elevado dentro de la gráfica será aquella que sea más adecuada. En el caso en que se produzca cruzamiento entre varias alternativas en la gráfica correspondiente al método Promethee I, la gráfica del método Promethee II será la que determinará la alternativa óptima, siendo una alternativa más adecuada si presenta un flujo neto mayor.

En este caso, no se produce cruzamiento, y el método Promethee I será el que establezca la mejor alternativa. Se puede observar como la alternativa óptima es aquella cuya generación es fotovoltaica y el coche con pila de combustible, tanto para el año 2018 como para 2025.

Este resultado era de esperar ya que, en Écija, la energía fotovoltaica es capaz de producir más energía y, aunque el vehículo híbrido tiene un menor consumo, el vehículo basado en hidrógeno no tiene ningún tipo de impacto medioambiental.

6.1.1.2 Funciones predeterminadas e imposición de pesos manualmente.

En este subapartado, las funciones de preferencia serán las predeterminadas y la importancia que va a tener cada peso es la siguiente, en una escala de 0 a 10:

Estado de la batería (SOC)	Estado del depósito de H (MHL)	Pérdidas	Coste	Impacto MA
5	6	2	9	0

Figura 6-5 Importancia relativa de los criterios

Como no es necesario normalizar los pesos, se van a introducir dichos valores en la interfaz. La introducción de estos valores se realiza directamente por pantalla, desactivando la opción 'Pesos AHP', como se puede ver en la figura 6-5 y en la figura 6-6. Los resultados se muestran en las figuras 6-5 y 6-6, representando el mismo número de alternativas que en el apartado anterior, 6. Estas 6 alternativas corresponden a los 3 tipos de vehículos posibles para los dos tipos de energía renovable.

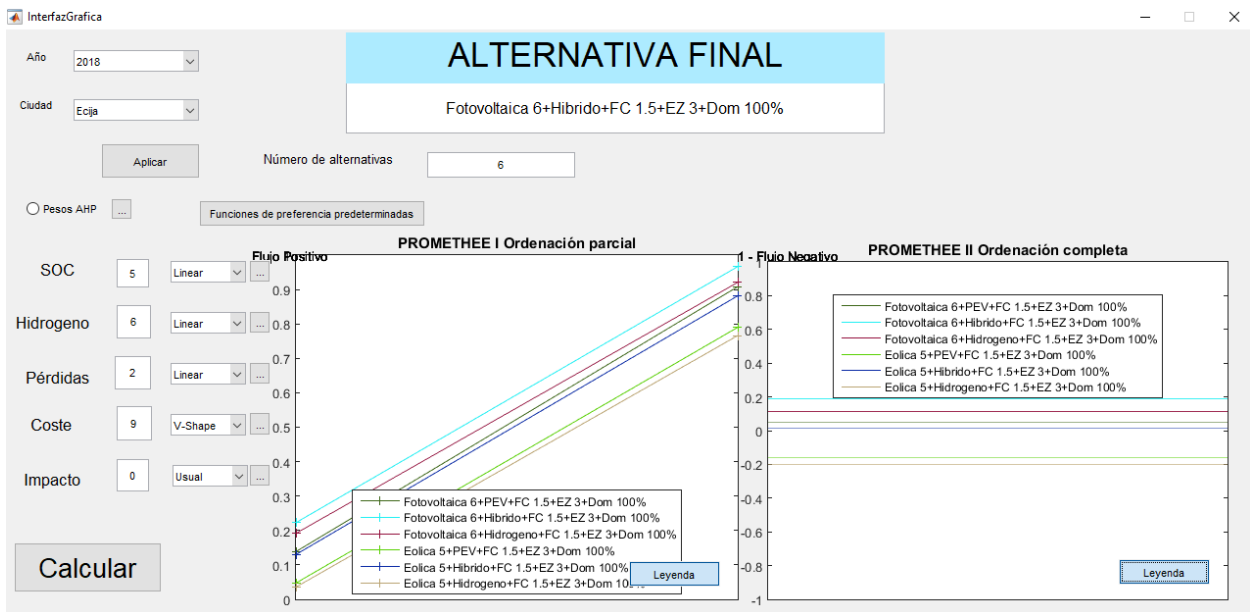


Figura 6-6 Evaluación de Écija con pesos insertados manualmente para 2018

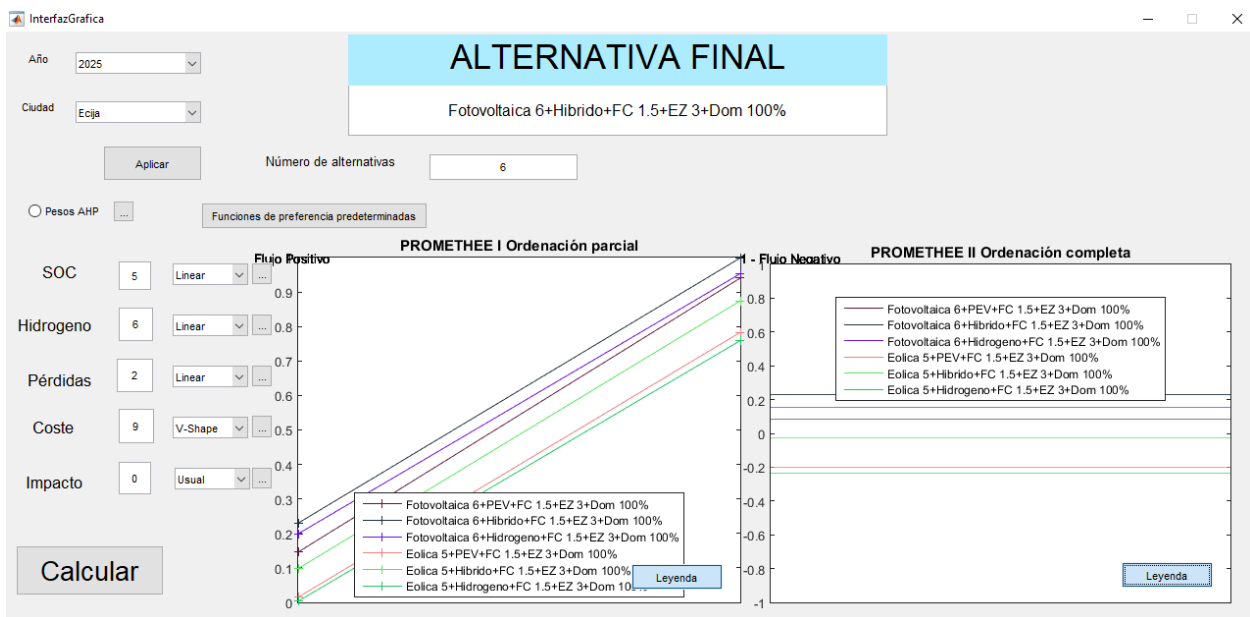


Figura 6-7 Evaluación de Écija con pesos insertados manualmente para 2025

Se puede ver que, pese al aumento de la importancia del coste de la generación y la disminución del resto de

parámetros, la generación óptima sigue siendo la fotovoltaica. En cuanto al vehículo, el híbrido es el óptimo, algo que era previsible, ya que hemos supuesto que el impacto medioambiental no tiene relevancia alguna en la evaluación. Además, en la figura 6-7, se puede observar la superioridad de esta alternativa con respecto a las otras con el flujo negativo, que es prácticamente 0. Esto indica que no existe alternativa que sea predominante a ella en ninguno de los criterios. Esto no quiere decir que sea superior a ella cuantitativamente en todos los criterios, pero sí que, con las funciones de preferencia establecidas, no existe ninguna alternativa que sea mejor que ella para tales criterios.

6.1.1.3 Pesos predeterminados (AHP) e imposición de las funciones de preferencia manualmente.

En este subapartado se van a escoger los pesos predeterminados obtenidos mediante el método AHP en el apartado 3, mientras que se modificarán las funciones de preferencia manualmente.

Las funciones de preferencia elegidas son las siguientes:

- Estado de la batería (SOC): función Linear, con un valor de $p=6$ y $q=1,5$.
- Estado del depósito de hidruros (MHL): función Linear, con un valor de $p=4$ y $q=0,75$.
- Pérdidas: función V-Shape, con un valor de $p=3$.
- Coste de la generación: función Linear, con un valor de $p=4$ y $q=1$.
- Impacto MA: función Usual.

Los resultados se muestran en las figuras 6-8 y 6-9. En este caso también se puede observar como, para ambos casos, la alternativa óptima es la misma: generación fotovoltaica y vehículo por hidrógeno. Esto indica que la modificación de las funciones de preferencia no tiene porqué tener demasiada importancia. Esto es debido a la similitud entre las distintas funciones de preferencia escogidas y debido a que la mayoría de las alternativas no exceden los valores de preferencia absoluta ni los valores de indiferencia establecidos.

Una mala imposición de las funciones de preferencia puede llevar a escoger una alternativa errónea. Por ejemplo, si se establece un valor de indiferencia elevado, la mayoría de las alternativas serán indiferentes entre ellas y no se podrá llegar a una solución lógica.

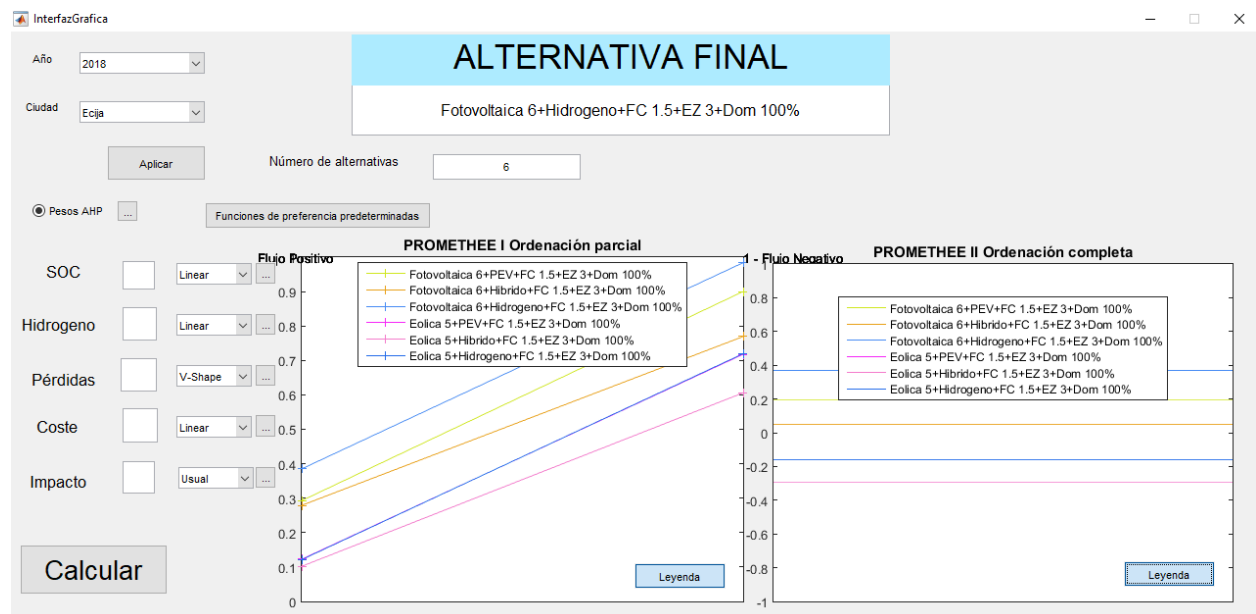


Figura 6-8 Evaluación de Écija con funciones de preferencia insertadas manualmente para 2018

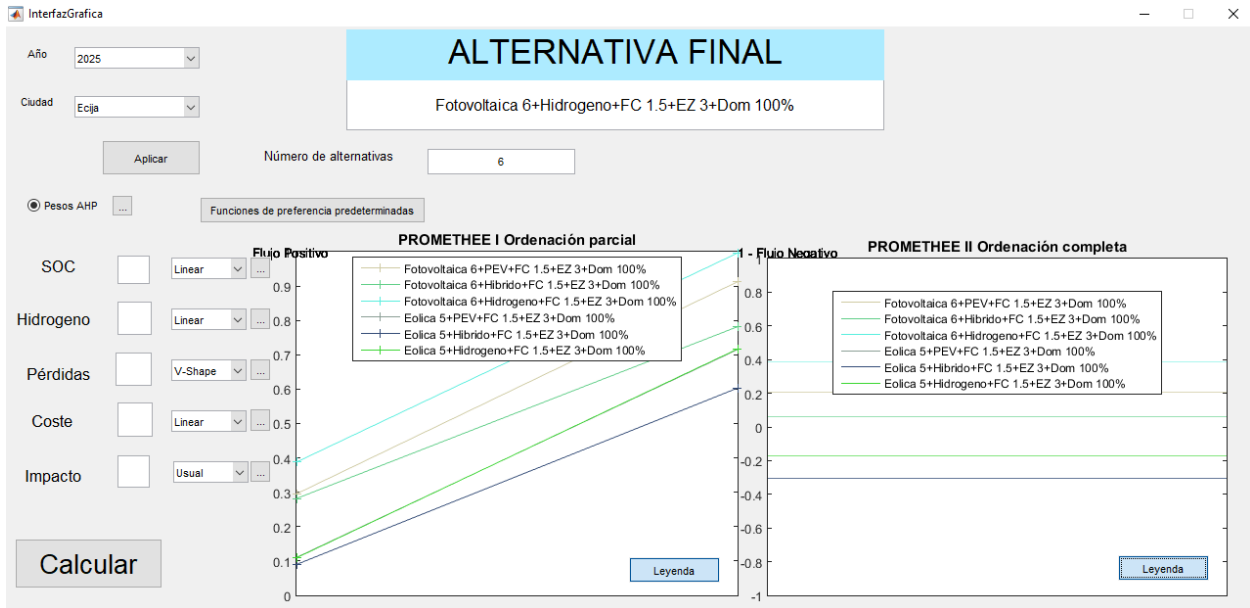


Figura 6-9 Evaluación de Ecija con funciones de preferencia insertadas manualmente para 2025

6.1.1.4 Pesos obtenidos por el método AHP modificando matriz y funciones de preferencia predeterminadas.

En este subapartado se va a ver la opción de obtener los pesos a partir del método AHP, pero a partir de una matriz distinta a la predeterminada. Hay que tener en cuenta que los valores de la matriz superior son la inversa de sus semejantes en la matriz inferior. En cuanto a las funciones de preferencia, se tomarán las predeterminadas.

A la hora de modificar los valores de la matriz de combinación AHP, aparecerá la ventana para su modificación (ver figura 6-10), donde se van a imponer los siguientes valores:

	SOC/Hidro	Pérdidas	Costes	Impacto
SOC/Hidro	1	4	4	2
Pérdidas	0.2500	1	1	0.2500
Costes	0.2500	1	1	0.2500
Impacto	0.5000	4	4	1

Figura 6-10 Modificación valores matriz AHP

Una vez establecida la matriz AHP, se evalúan las alternativas. Esta evaluación se muestra en las figuras 6-11 y 6-12. Estas figuras muestran un resultado similar a las anteriores evaluaciones. En ambos casos, la modificación de las importancias relativas no implica un cambio en la selección de la alternativa más adecuada. Aunque se puede observar una aproximación de la alternativa que contempla el uso del vehículo eléctrico PEV, la alternativa más adecuada para esta localidad sigue siendo la generación fotovoltaica y el uso del vehículo con pila de combustible.

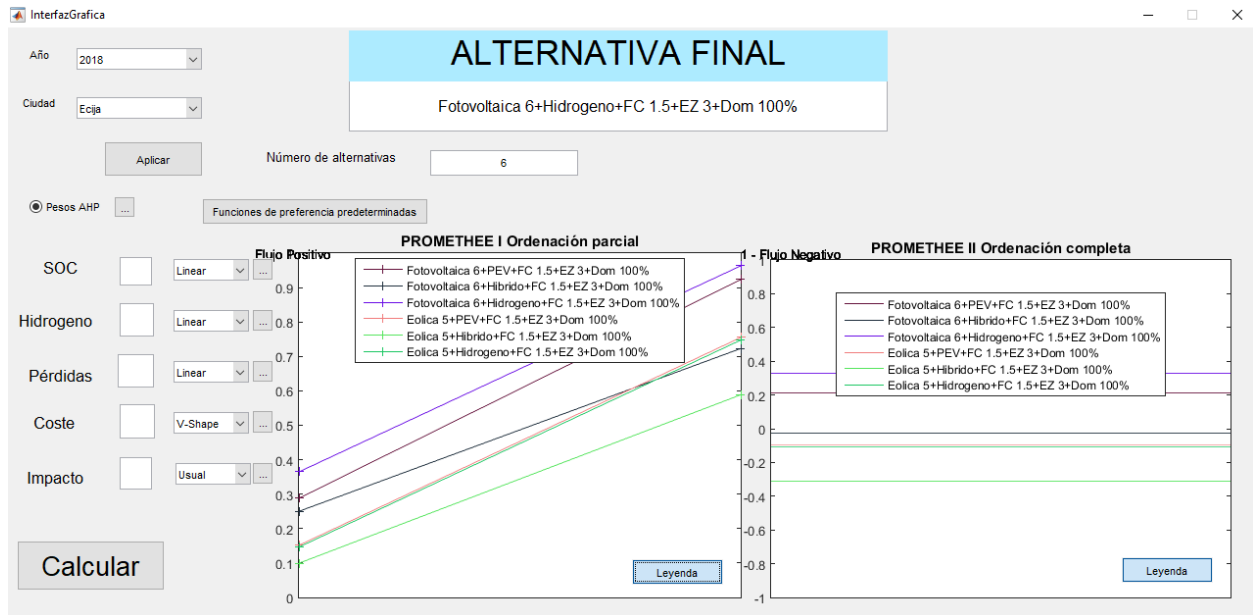


Figura 6-11 Evaluación de Écija con pesos obtenidos mediante el método AHP modificados para 2018

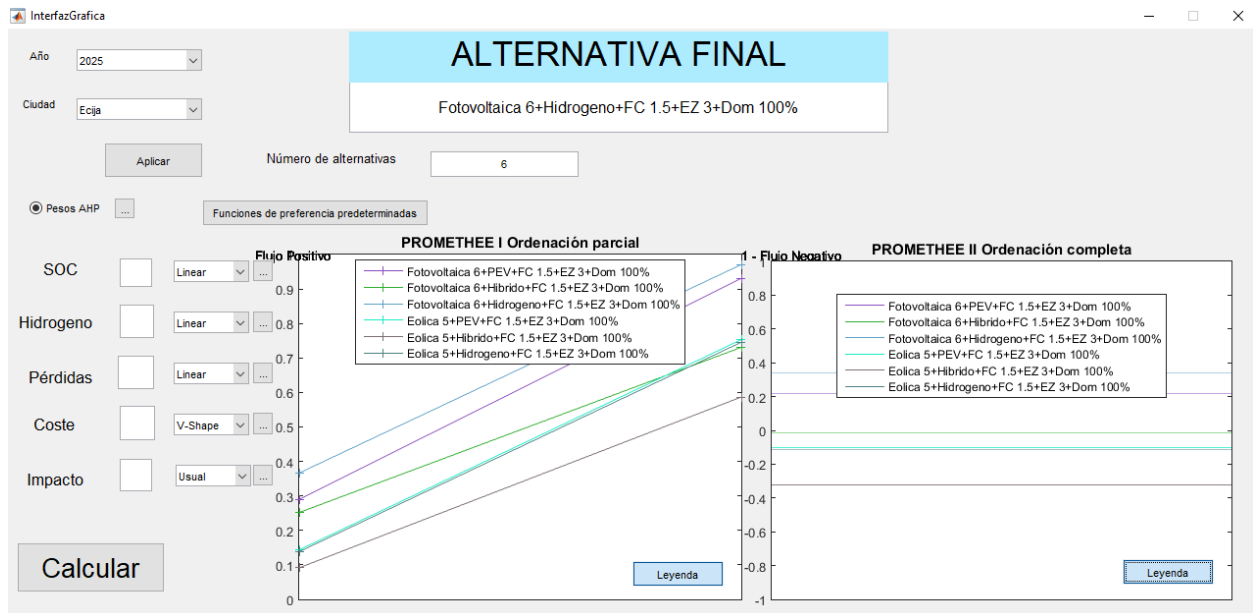


Figura 6-12 Evaluación de Écija con pesos obtenidos mediante método AHP modificados para 2025

6.1.2 Evaluación para la localidad de Tarifa

Para el caso de Tarifa, se evaluará simplemente para las funciones de preferencia y pesos predeterminados, con el fin de no usar las mismas funcionalidades que para el caso de Écija.

6.1.2.1 Funciones de preferencia y pesos predeterminados.

Como en el apartado 6.1.1.1, se realizará la evolución para las funciones de preferencias predeterminadas y los pesos obtenidos mediante el método AHP de forma predeterminada. Así mismo, las alternativas que serán evaluadas serán un total de 6, representando los 3 posibles vehículos para cada una de las energías renovables.

En las figuras 6-13 y 6-14 se muestran los resultados obtenidos al realizar la evaluación.

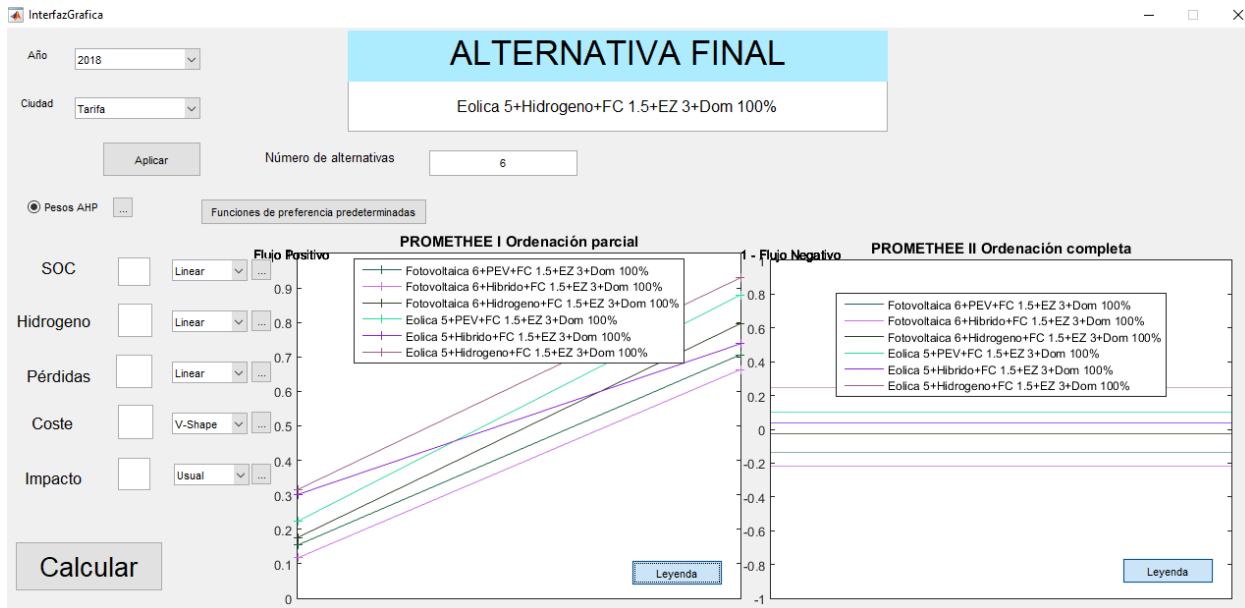


Figura 6-13 Evaluación de Tarifa para 2018

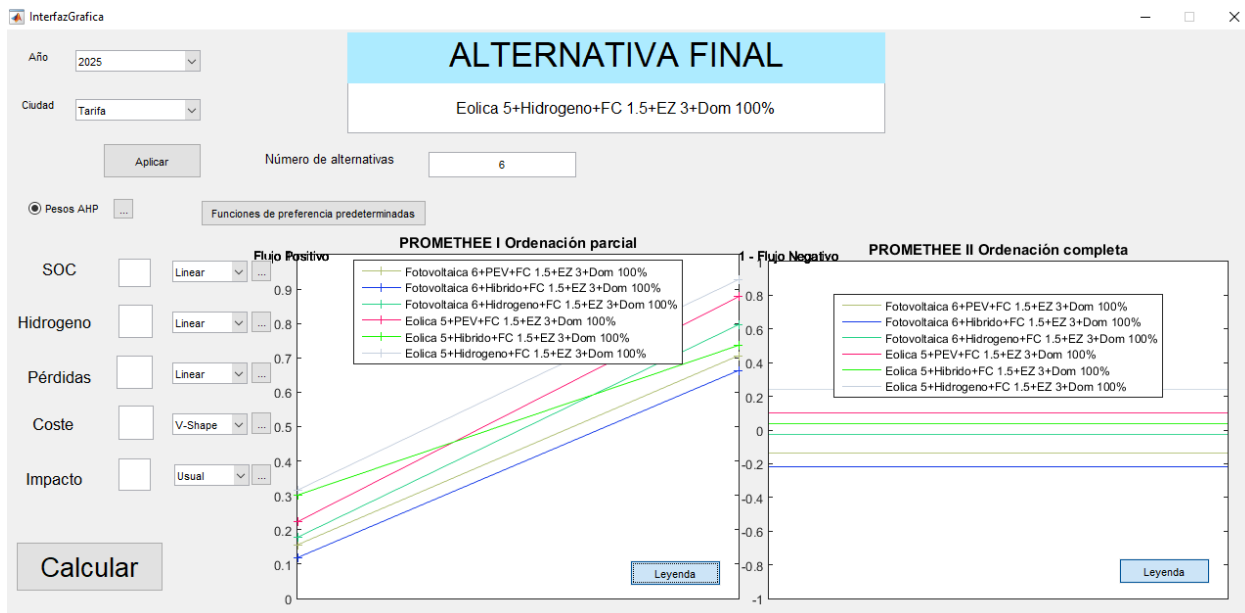


Figura 6-14 Evaluación de Tarifa para 2025

Se puede observar como, para el caso de Tarifa, la alternativa escogida sería la generación eólica, algo esperado, ya que la meteorología en dicha localidad refleja que el viento tiene una gran presencia. En cuanto al vehículo, se escoge el coche por hidrógeno. Se puede observar como no se produce cruzamiento entre las distintas alternativas y que la alternativa escogida es claramente más adecuada para las especificaciones indicadas.

En este apartado 6.1 se ha podido ver la coherencia de los resultados obtenidos, ya que las alternativas óptimas han sido aquellas que se podían prever, y una vez que se han modificado ciertos criterios como el impacto medioambiental, se ha visto cómo la alternativa óptima ha variado.

En los siguientes apartados se verán modificaciones en la propia Microgrid, por lo que podremos ver posibles modificaciones que permitan un funcionamiento más adecuado de la misma.

6.2 Evaluación para cada localidad del estado inicial de la batería

En este apartado se evaluará para cada ciudad el estado inicial de la batería que permitirá un mejor

funcionamiento de la Microgrid.

Como para la obtención de la alternativa óptima para esta variación el coste no es un criterio que vaya a modificar la elección de una alternativa u otra, se va a evaluar simplemente para el año 2018.

La evaluación se hará a partir de las funciones de preferencia predeterminadas y de los pesos obtenidos mediante el método AHP ya predeterminados.

Se van a evaluar los siguientes valores para el estado inicial de la batería: 50%, 55%, 60%, 65%, 70%, 75%, 80%, 85%, 90%, 95% y 100%. Por tanto, la parte del código que corresponde con la descripción de los parámetros modificables de la Microgrid que se va a introducir en el archivo *'Escritura_datos_excel.m'* es el siguiente:

```
%SOC inicial
Porcentaje_SOC=[50 55 60 65 70 75 80 85 90 95 100];
NumSOC=length(Porcentaje_SOC);
%Potencia de la generación
kWPVTot=[6];
NumPotPV=length(kWPVTot);
kWEolTot=[5];
NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[3];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFCTot=[1.5];
NumFC=length(kWFCTot);
```

6.2.1 Evaluación para la localidad de Écija

En la figura 6-15 se muestran los resultados. Se observa como, al existir un mayor número de alternativas a evaluar, es más difícil la visualización de las distintas alternativas. El número total de alternativas refleja las 11 diferentes situaciones iniciales de la batería para cada una de las distintas energías y para cada uno de los distintos vehículos evaluables ($11 \times 3 \times 2 = 66$).

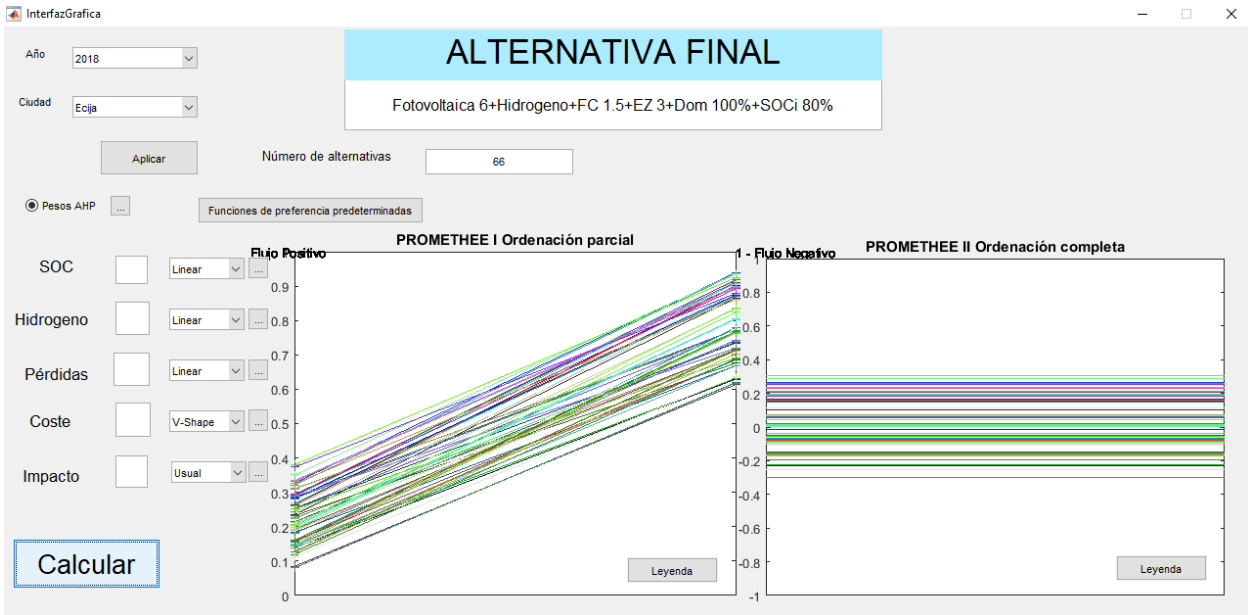


Figura 6-15 Evaluación de Écija para distintos estados iniciales de la batería

Seguidamente, en las figuras 6-16, 6-17 y 6-18, se detallan la evaluación del estado de la batería y del depósito de hidruros para el caso en que el estado inicial de la batería sea del 80%, así como para el caso en que el estado inicial es del 75% y aquella del 85%, por ser las más cercanas a la opción óptima.

En estas figuras se representa el valor de la energía total almacenada en la batería en kWh y la cantidad de hidrógeno almacenado en el tanque en volumen (Nm³). Por tanto, pese a que ambas representaciones se encuentran en la misma gráfica, no tienen las mismas unidades.

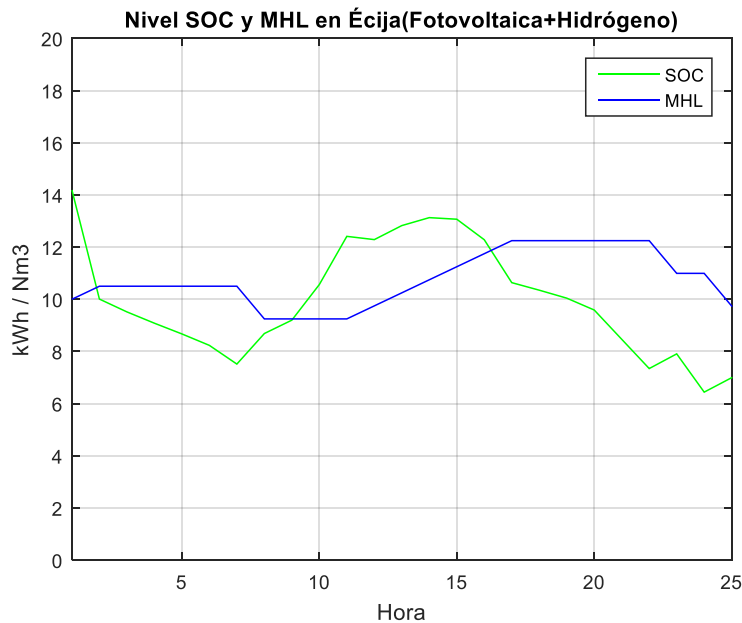


Figura 6-16 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 75%

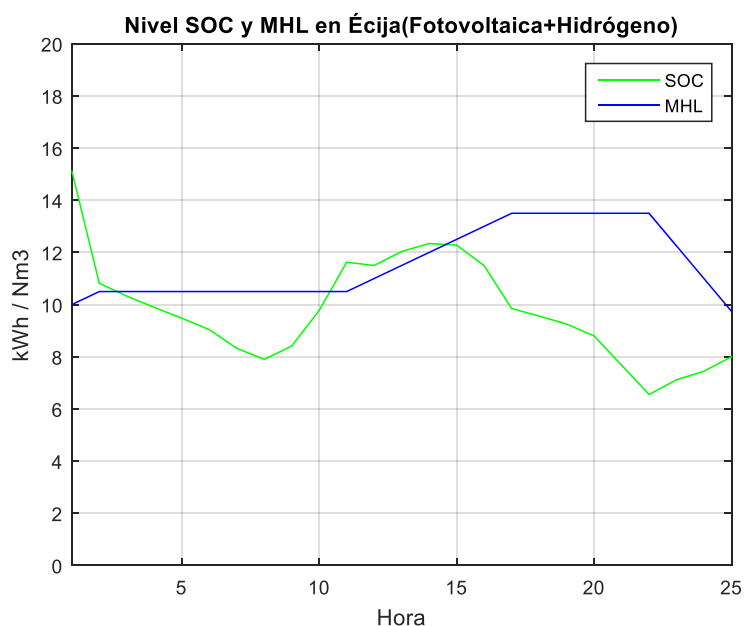


Figura 6-17 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 80%

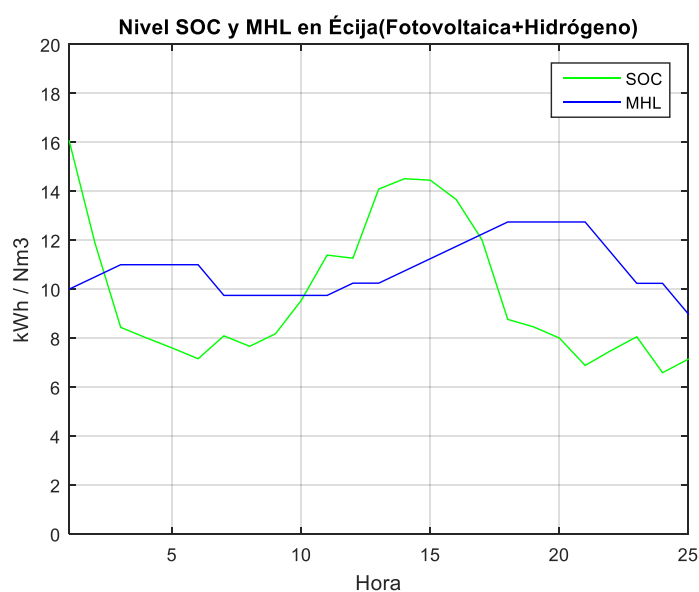


Figura 6-18 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 85%

A primera vista no se determina una gran diferencia en cuanto a diferencia entre el estado final e inicial tanto del estado de la batería como del depósito de hidruros. Por tanto, se revisa los datos escritos en el documento Excel 'Evaluacion Microgrid.xls' para ver estos valores:

Alternativas	NivelSOC	Hidrógeno	Pérdidas	Costes	Impacto MA
Fotovoltaica 6+Hidrogeno+FC 1.5+EZ 3+Dom 100%+SOCi 75%	-7,196019	-0,2602	5,50545111	3,4999133	Very good
Fotovoltaica 6+Hidrogeno+FC 1.5+EZ 3+Dom 100%+SOCi 80%	-8,741262	0,9932	5,20025596	3,4999133	Very good
Fotovoltaica 6+Hidrogeno+FC 1.5+EZ 3+Dom 100%+SOCi 85%	-8,938839	-1,0136	6,78878798	3,4999133	Very good

Figura 6-19 Valores de los distintos criterios para distintos estados iniciales de la batería en Écija

Se observa que la distribución para el caso en que la batería comienza con un 85% de carga no mejora en ningún criterio a las otras dos. Además, puede observarse que las pérdidas son superiores, lo que puede indicar que se hace un mayor uso del circuito de hidrógeno, donde las pérdidas de electrolizador y pila de combustible se hacen notar. En cuanto a los casos del 75% y del 80%, se puede ver como el hecho de que el peso impuesto al criterio que se corresponde con el estado del tanque de hidrógeno sea superior al del estado de la batería hace que la

alternativa del 80% sea la más adecuada (ya que, pese a que el nivel de la batería tenga un descenso mayor, se produce una mayor cantidad de hidrógeno).

6.2.2 Evaluación para la localidad de Tarifa

A la hora de evaluar la localidad de Tarifa, se partirá de los mismos datos que para Écija. Por tanto, se evaluarán un total de 66 alternativas, como se muestra en la figura 6-20. En esta figura, al igual que antes, no se establece una visualización correcta de las alternativas, al ser un número tan elevado. Pese a ello, se muestra por pantalla la alternativa más adecuada, que se corresponde con, además de, como ya era sabido, energía eólica y vehículo con pila de combustible, un estado inicial de la batería del 70%.

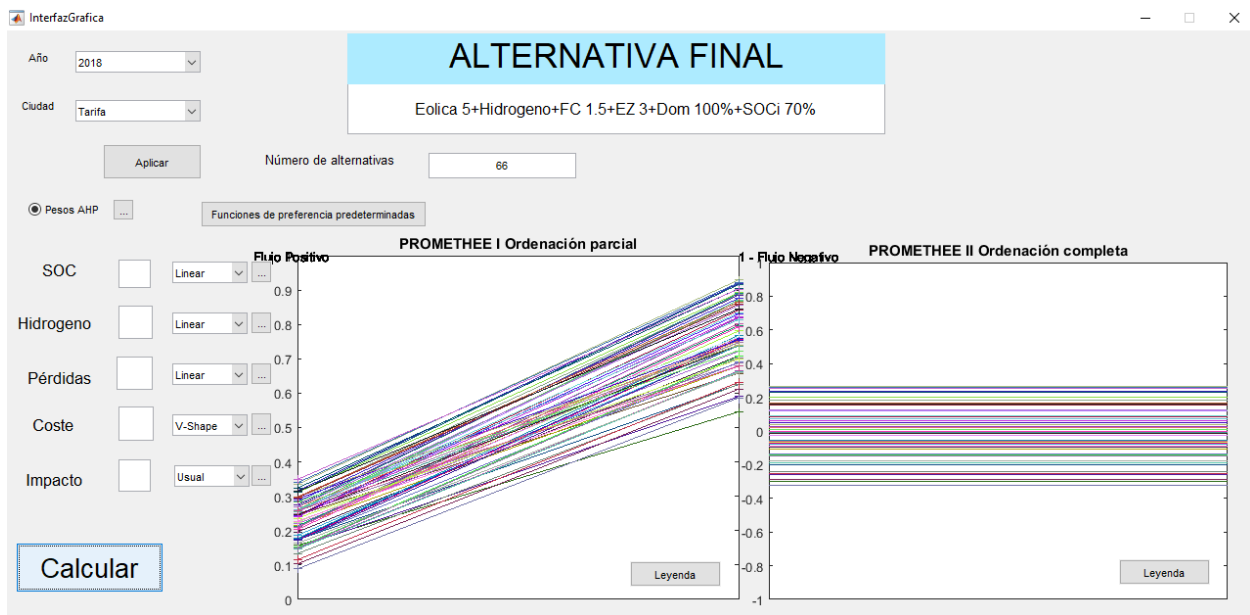


Figura 6-20 Evaluación de Tarifa para distintos estados iniciales de la batería

Como en el subapartado anterior, en las figuras 6-21, 6-22 y 6-23, se mostrará la evolución de la batería y del depósito de hidruros para el caso óptimo, batería inicialmente al 70%, y los dos casos más cercanos, 65% y 75%.

Cabe mencionar de nuevo que las unidades del nivel de la batería y de la cantidad de hidrógeno almacenada no son las mismas, siendo kWh y Nm³, respectivamente.

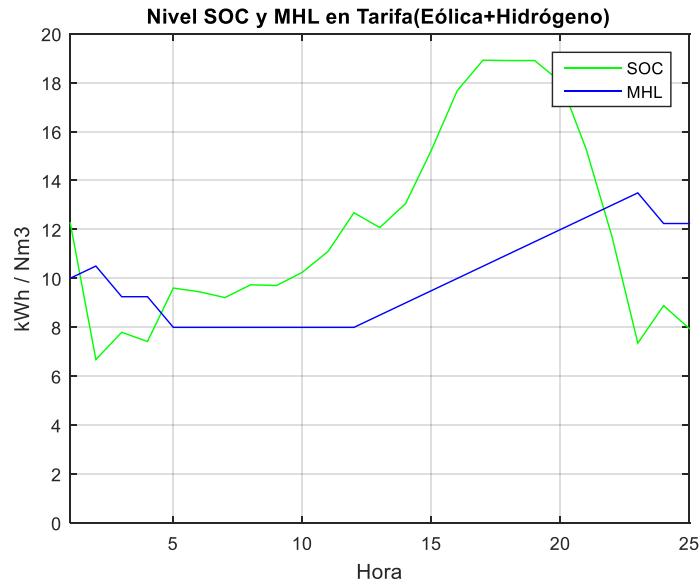


Figura 6-21 Evolución niveles SOC y MHL en Tarifa para un estado inicial de la batería del 65%

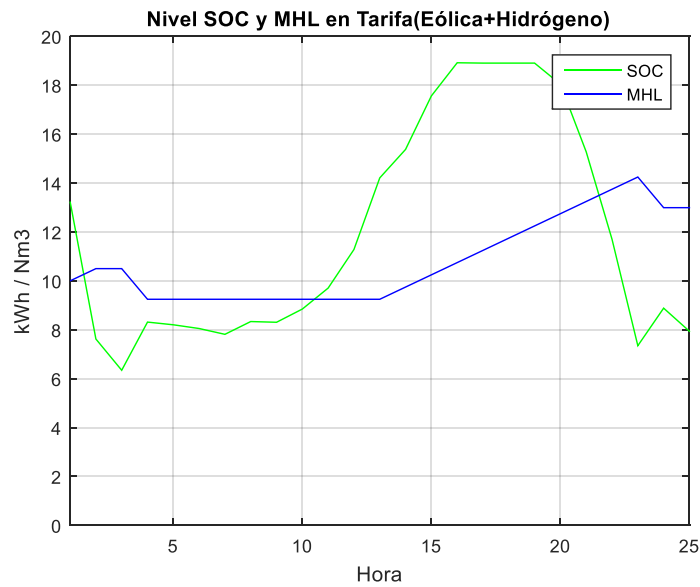


Figura 6-22 Evolución niveles SOC y MHL en Tarifa para un estado inicial de la batería del 70%

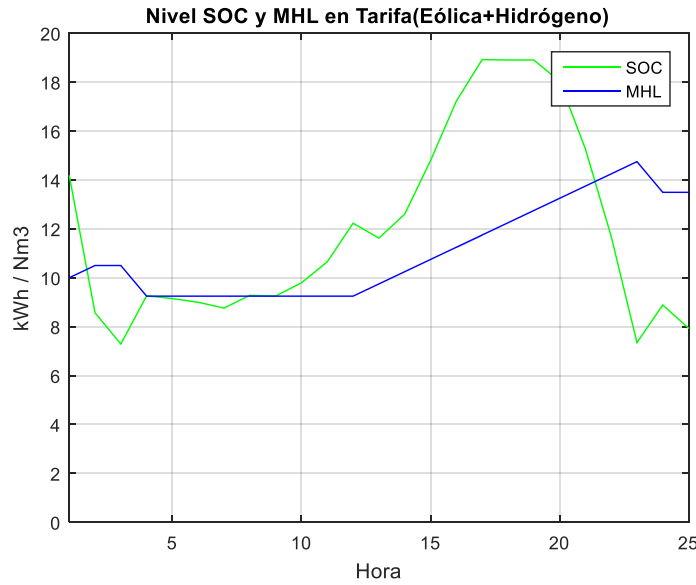


Figura 6-23 Evolución niveles SOC y MHL en Tarifa para un estado inicial de la batería del 75%

Como en el caso de Écija, se muestran los datos presentes en el Excel para poder observar de manera más cuantitativa los valores.

Alternativas	NivelSOC	Hidrógeno	Pérdidas	Costes	Impacto MA
Eolica 5+Hidrogeno+FC 1.5+EZ 3+Dom 100%+SOCi 65%	-4,360186	2,2398	9,11916754	5,00974046	Very good
Eolica 5+Hidrogeno+FC 1.5+EZ 3+Dom 100%+SOCi 70%	-5,306186	2,9932	8,42291579	5,00974046	Very good
Eolica 5+Hidrogeno+FC 1.5+EZ 3+Dom 100%+SOCi 75%	-6,252186	3,4932	8,77017537	5,00974046	Very good

Figura 6-24 Valores de los distintos criterios para distintos estados iniciales de la batería en Tarifa

En este caso sucede lo contrario que en Écija, ya que la alternativa más adecuada no será aquella que genera una mayor cantidad de hidrógeno, sino aquella que tenga un menor descenso de carga en la batería. Además, también es aquella en la que se obtienen menores pérdidas.

6.3 Evaluación para cada localidad del circuito de hidrógeno

Una vez se ha obtenido el valor óptimo para el estado inicial de carga de la batería, se van a introducir modificaciones para el valor de la potencia nominal del electrolizador y de la pila de combustible.

Se evaluarán ambas localidades, teniendo en cuenta el valor inicial obtenido en el apartado anterior. Tanto para Écija como para Tarifa, se evaluarán las siguientes potencias: para el electrolizador, 3 kW; 2,5 kW; 2 kW; 1,5 kW y 1 kW, y para la pila de combustible, 1,5 kW; 1 kW y 0,5 kW. La evaluación será conjunta, por tanto, se obtendrá una pareja de valores como resultado óptimo.

6.3.1 Evaluación para la localidad de Écija

A la hora de evaluar la localidad de Écija, el código impuesto en 'Escritura_datos_excel.m' queda como sigue:

```
%SOC inicial
Porcentaje_SOC=[80];
NumSOC=length(Porcentaje_SOC);
%Potencia de la generación
kWPVTot=[6];
NumPotPV=length(kWPVTot);
kWEolTot=[5];
```

```

NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[3 2.5 2 1.5 1];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFCTot=[1.5 1 0.5];
NumFC=length(kWFCTot);

```

Al ejecutar la interfaz, se obtiene el resultado que se muestra en la figura 6-25. Se puede observar un total de 90 alternativas a evaluar. Estas se corresponden con las 15 combinaciones posibles dentro del circuito de hidrógeno, para las 6 combinaciones iniciales ($15 \times 6 = 90$). La figura 6-25 muestra que la alternativa óptima no es aquella que hace uso de la mayor potencia nominal posible, sino que será la que se corresponde con una potencia del electrolizador de 1 kW y pila de combustible de 1,5 kW. Esto se debe a las elevadas pérdidas que se generan en ambos equipos, haciendo que, pese a tener una mayor potencia y mayor capacidad de generación de hidrógeno, sea más adecuada la disminución de las pérdidas obtenidas.

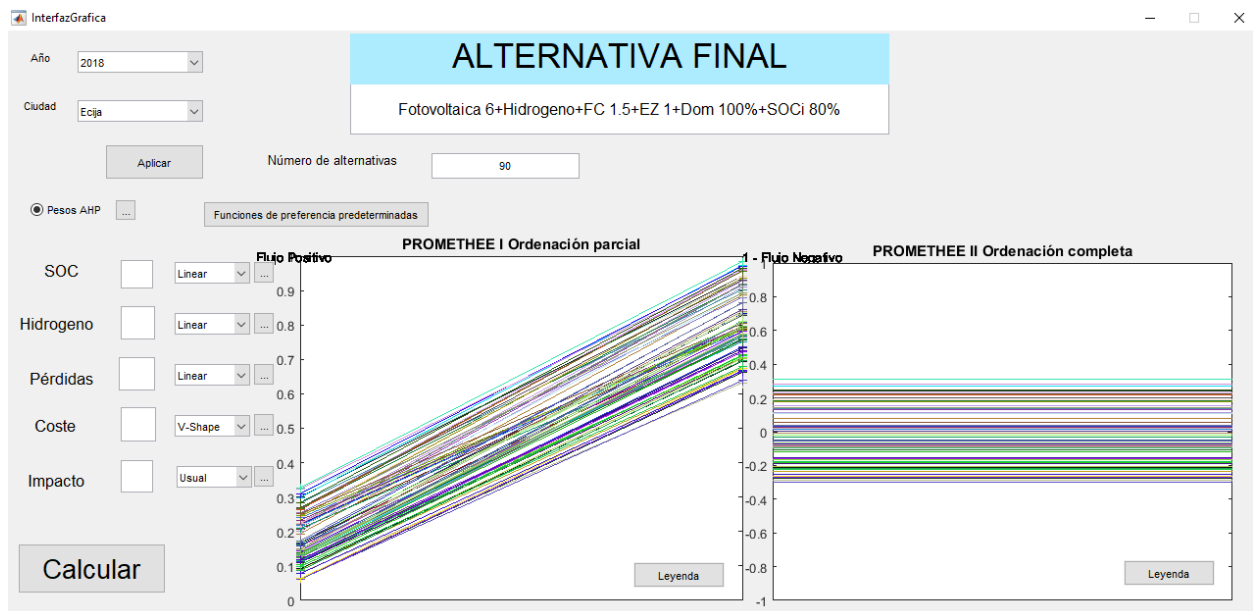


Figura 6-25 Evaluación de Écija con modificaciones en el circuito de hidrógeno

6.3.2 Evaluación para la localidad de Tarifa

A la hora de evaluar la localidad de Tarifa, el código impuesto en 'Escritura_datos_excel.m' queda como sigue:

```

%SOC inicial
Porcentaje_SOC=[70];
NumSOC=length(Porcentaje_SOC);
%Potencia de la generación
kWPVTTot=[6];
NumPotPV=length(kWPVTTot);
kWEolTot=[5];
NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];

```

```

kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[3 2.5 2 1.5 1];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFCTot=[1.5 1 0.5];
NumFC=length(kWFCTot);

```

Una vez ejecutado el programa, el resultado obtenido se muestra en la figura 6-26. Al igual que en el caso de Écija, la alternativa óptima no se corresponde con la potencia nominal máxima, y es aquella que representa a un electrolizador de potencia 1 kW y pila de combustible de 1,5 kW.

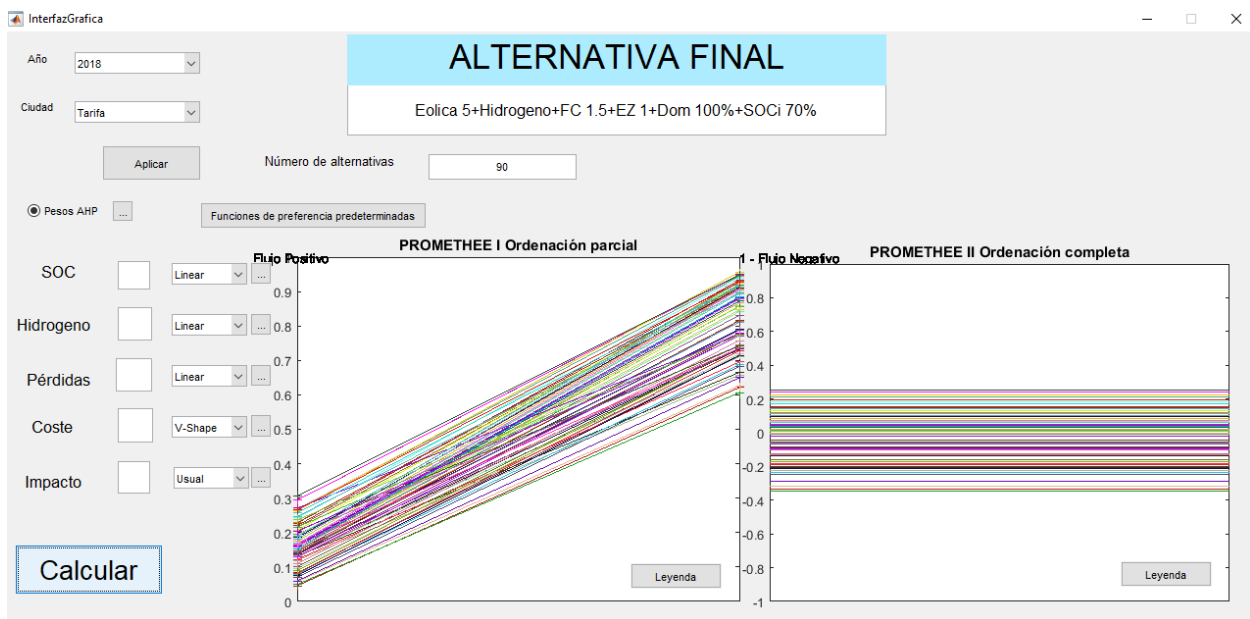


Figura 6-26 Evaluación de Tarifa con modificaciones en el circuito de hidrógeno

6.4 Evaluación conjunta del estado inicial SOC y del circuito de hidrógeno

En este apartado se va a realizar la evaluación que engloba a las dos evaluaciones anteriores. Por tanto, con esta evaluación se verá si el valor óptimo de estado inicial de la batería depende de la potencia del circuito de hidrógeno y viceversa.

La evaluación se hará en un principio a partir de la matriz AHP predeterminada y las funciones de preferencia predeterminadas.

Para evitar un número excesivo de alternativas, como ya se prevé que el vehículo más adecuado va a ser el de hidrógeno, se van a evaluar sólo las alternativas correspondientes a este tipo de vehículos.

Pese a que esta evaluación se podría haber efectuado directamente, las evaluaciones separadas ayudan a imponer un menor número de alternativas y por tanto facilitar la resolución del problema.

Tanto para Écija como para Tarifa, el código impuesto en el archivo '*Escritura_datos_excel.xls*' es el siguiente:

```

%SOC inicial
Porcentaje_SOC=[50 55 60 65 70 75 80 85 90 95 100];
NumSOC=length(Porcentaje_SOC);
%Potencia de la generación

```



```

kWPVTot=[ 6 ];
NumPotPV=length(kWPVTot);
kWEolTot=[5];
NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[3 2.5 2 1.5];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFACTot=[1.5 1];
NumFC=length(kWFACTot);

```

6.4.1 Evaluación para la localidad de Écija

Al establecer como localidad Écija, se obtienen los resultados que se muestran en la figura 6-27. Se observan un total de 176 alternativas, que se corresponden con los 11 posibles estados de la batería, las 2 opciones de energía renovable, las 2 potencias nominales de la pila de combustible y las 4 del electrolizador ($11 \times 2 \times 2 \times 4 = 176$). Se puede observar que la alternativa escogida es aquella con un electrolizador de 1 kW, pila de combustible de 1,5kW y un estado inicial de la batería al 70%, observándose un descenso considerable en el estado inicial de la batería.

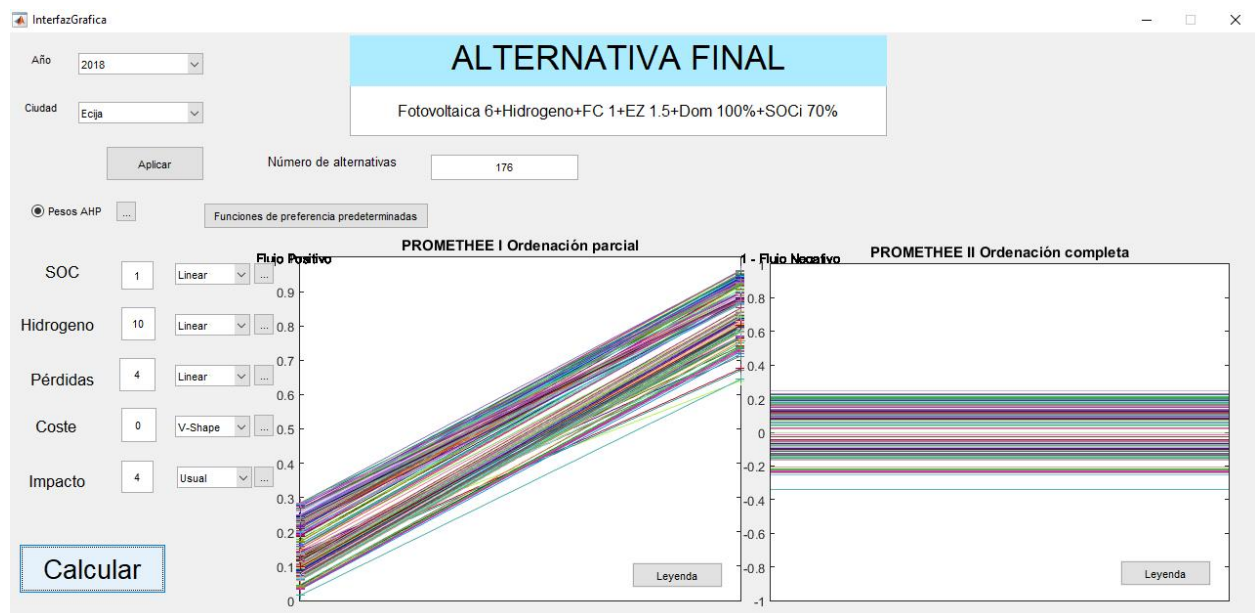


Figura 6-27 Evaluación general de Écija

Con esta evaluación se puede ver la capacidad de la interfaz, capaz de evaluar 176 alternativas simultáneamente. También cabría la posibilidad de evaluar cualquier número de alternativas, pero podrían aparecer errores al existir un número tan elevado de alternativas y flujos de valores cercanos. Además, se ha comprobado que, para una evaluación de alrededor de 1000 alternativas, el tiempo de ejecución es cercano a 30 minutos.

La evolución del estado de la batería y del tanque de hidrógeno se muestra en la figura 6-28. Se puede observar un leve incremento en el nivel de hidrógeno y un descenso pronunciado en el nivel de la batería. Además, se observa que las modificaciones en los estados de batería y tanque de hidrógeno son más suaves, debido a la disminución de la potencia nominal de electrolizador y pila de combustible, haciendo que las variaciones que se puedan producir sean menos drásticas.

Los valores obtenidos para cada uno de los criterios se muestran en la figura 6-28.

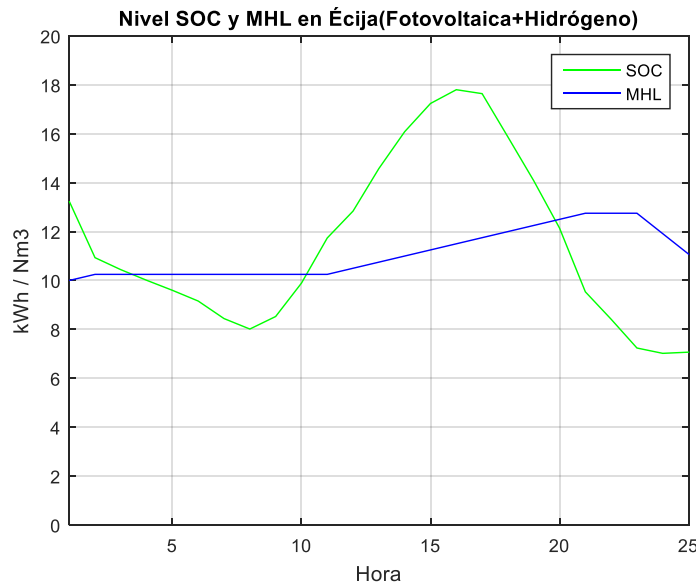


Figura 6-28 Evolución niveles SOC y MHL en Écija para un estado inicial de la batería del 70%, un electrolizador de 1 kW y una pila de combustible de 1,5 kW

Alternativas	NivelSOC	Hidrógeno	Pérdidas	Costes	Impacto MA
Fotovoltaica 6+Hidrogeno+FC 1+EZ 1.5+Dom 100%+SOCi 70%	-6,1763387	1,0788	5,20081994	3,4999133	Very good

Figura 6-29 Valores de los distintos criterios para la alternativa general más adecuada en Écija

Comparando esta tabla con la del subapartado 5.2.1, se observa que para este último caso se obtienen mejores resultados tanto para el nivel de la batería como para el nivel de hidrógeno.

Además de esta evaluación, se va a incrementar el peso del nivel del depósito de hidrógeno para ver cual sería la alternativa óptima ya que, como se dijo antes, la recarga de la batería es más sencilla que la obtención de Hidrógeno. Por ello, se van a modificar los pesos de los criterios, y se van a establecer manualmente. Se va a imponer un valor de 1 a 10 para cada criterio de acuerdo con la importancia global del mismo, obteniendo los pesos automáticamente. La importancia de cada criterio será la siguiente: SOC: 2; Hidrógeno: 10; Pérdidas: 4; Coste: 0; Impacto: 0.

Como se prevé que la generación será fotovoltaica, se impone un peso de 0 al criterio que representa el coste de la generación. Además, al ser la evaluación únicamente de vehículos por pila de combustible, el impacto tampoco se tendrá en cuenta.

La figura 6-30 muestra los resultados obtenidos para esta evaluación.

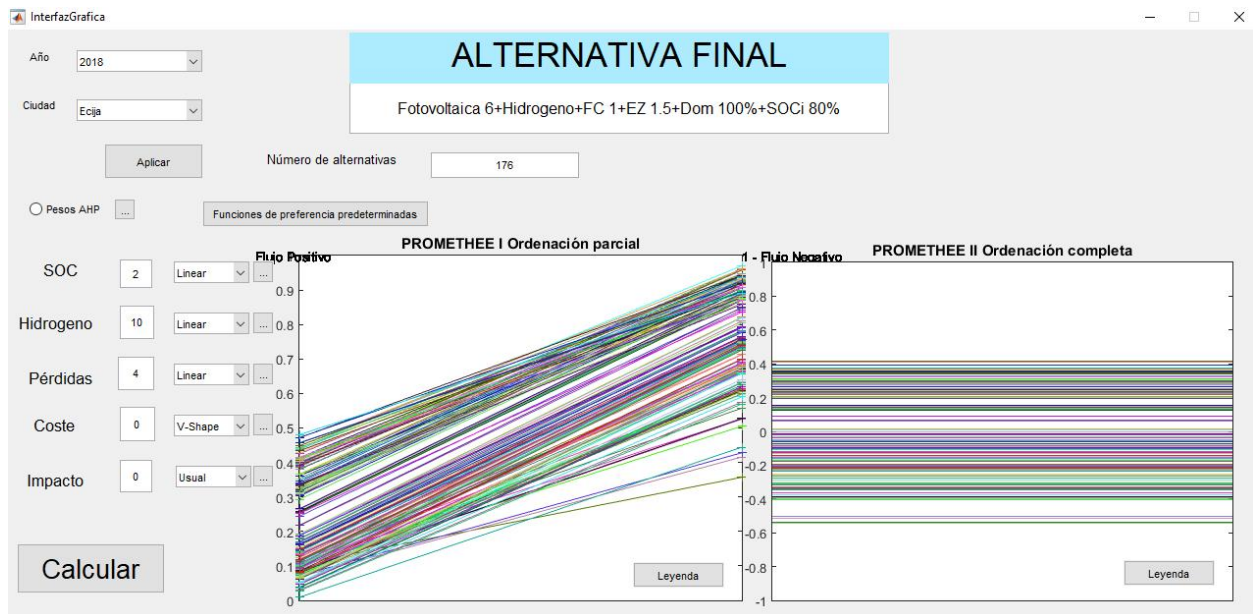


Figura 6-30 Evaluación general de Écija con los pesos modificados

Como se puede observar, al modificar los pesos la alternativa óptima será aquella que tiene un estado inicial de la batería del 80%.

Por tanto, en este punto, el usuario debe hacer una elección subjetiva de qué alternativa es la más adecuada. Para nuestro caso se va a suponer que la recarga de la batería se puede llevar a cabo sin problema alguno y se va a elegir esta última alternativa, siendo la alternativa óptima aquella que trabaja con **generación fotovoltaica, un electrolizador de 1,5 kW, una pila de combustible de 1 kW y un estado inicial de la batería del 80%**.

6.4.2 Evaluación para la localidad de Tarifa

Al igual que para el caso de Écija, en primer lugar, se ejecuta la interfaz para las condiciones predeterminadas. El número de alternativas a evaluar será el mismo, es decir, 176 alternativas. La figura 6-31 muestra los resultados obtenidos, donde se puede observar que la alternativa más adecuada es aquella que contempla la generación eólica, una pila de combustible de 1 kW, un electrolizador de 2,5 kW y un estado inicial de la batería del 70%. Se observa que, en este caso, el electrolizador tiene una potencia nominal superior al caso de Écija, y más cercana a la potencia nominal máxima que se está evaluando.

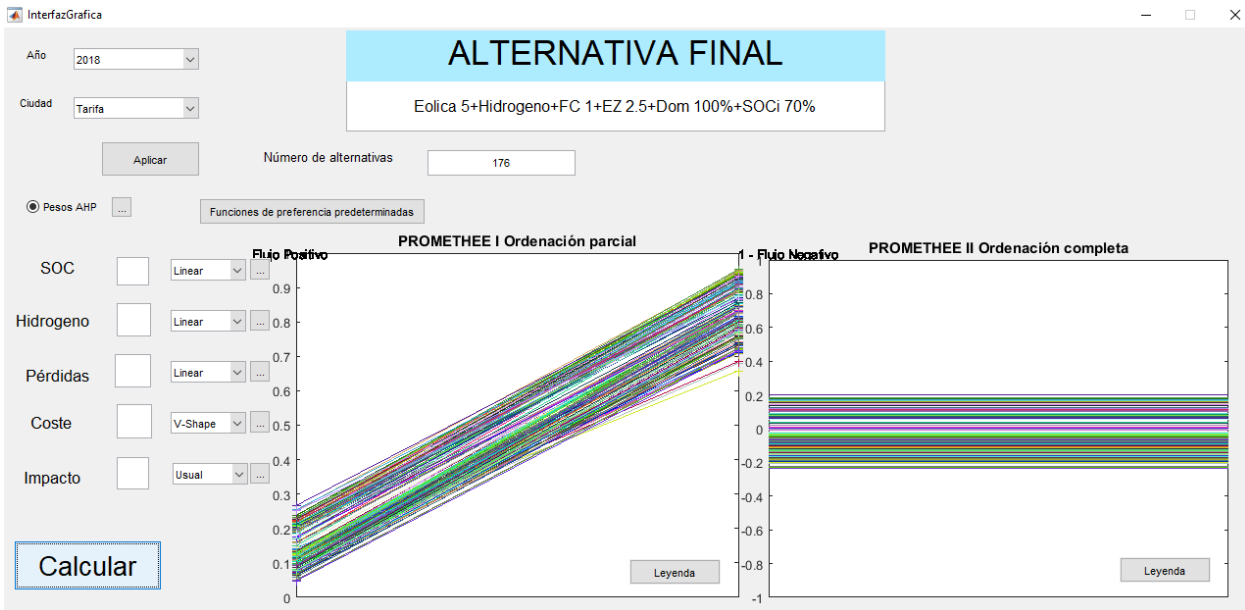


Figura 6-31 Evaluación general de Tarifa

Al igual que en el caso anterior, se va a evaluar también la opción de insertar pesos manualmente, imponiendo una mayor importancia a la cantidad de hidrógeno almacenada e ignorando la repercusión del coste y del impacto medioambiental. La figura 6-32 muestra los resultados obtenidos.

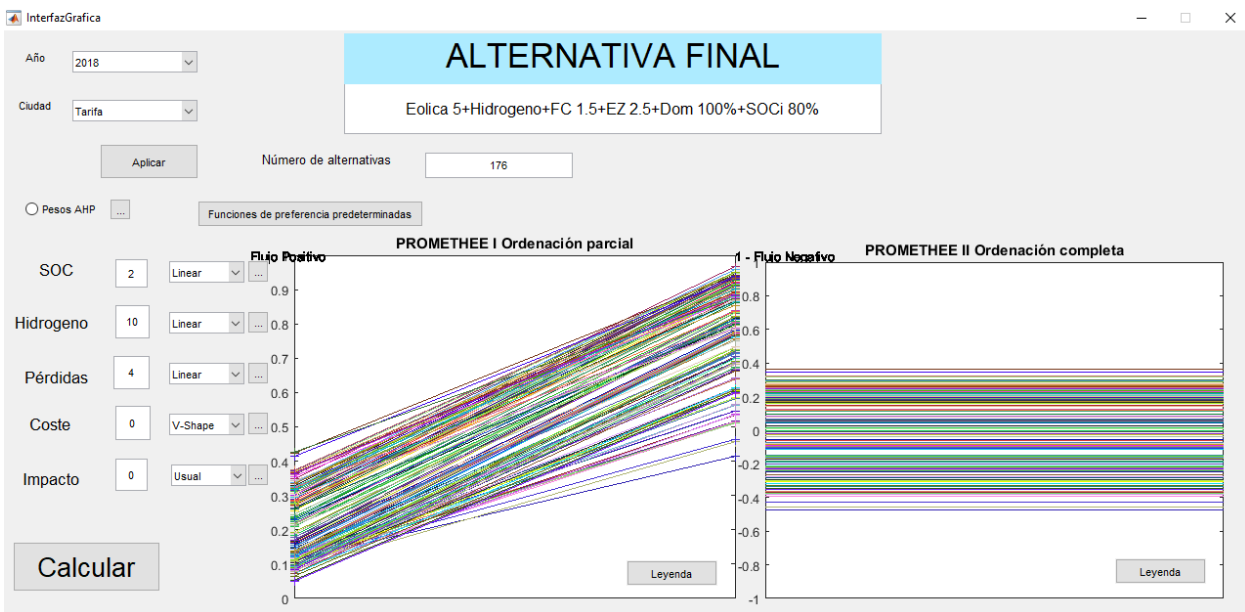


Figura 6-32 Evaluación general de Tarifa con pesos modificados

Al igual que en el caso anterior, se debe de hacer una evaluación subjetiva que determine cual de las dos alternativas se va a escoger, si esta última o la que se puede ver en la figura 6-31, ya que difieren en la potencia de la pila de combustible y en el estado inicial de la batería. Se va a escoger como alternativa óptima esta última, representada en la figura 6-32, que trabaja con **generación eólica, pila de combustible de 1,5 kW, electrolizador de 2,5 kW y un estado inicial de la batería del 80%**.

6.5 Evaluación de la potencia nominal de la generación

En esta evaluación se va a ver la posibilidad de modificar la potencia nominal de la generación. Para ello, no se va a modificar la distribución de la curva de generación, ya que se ha observado en el apartado 2.1 que ambas

curvas son aquellas que mejor pueden representar la generación.

Con el fin de establecer una evaluación que se centre mayormente en la generación y que haga el problema lo más sencillo posible para evitar posibles errores, se va a suponer que el coche escogido ha sido el coche por pila de combustible y que el impacto medioambiental tiene un peso de valor 0. Además, se aumentará el valor de los pesos de los criterios correspondientes al coste y a las pérdidas. Eso sí, cabe decir que en este trabajo no se tienen en cuenta las pérdidas que ocurren en la generación en sí, sino aquellas que se dan en el circuito en el que se mueve la energía.

Los pesos que se van a tener en cuenta son los siguientes:

$$w_1 = 0,2 ; w_2 = 0,3 ; w_3 = 0,25 ; w_4 = 0,25 ; w_5 = 0$$

Para ambas localidades se van a usar los valores recogidos en el apartado 6.4 para el estado inicial de la batería y para las potencias nominales del electrolizador y la pila de combustible.

Como en este caso el coste de la generación si puede suponer una variable que se deba tener en cuenta, se evaluarán los años 2018 y 2025.

6.5.1 Evaluación para la localidad de Écija

Para la localidad de Écija, el código correspondiente a la escritura de las distintas alternativas queda como sigue:

```
%SOC inicial
Porcentaje_SOC=[80];
NumSOC=length(Porcentaje_SOC);
%Potencia de la generación
kWPVTot=[6 5.5 5 4.5 4 3.5 3];
NumPotPV=length(kWPVTot);
kWEolTot=[5 4.5 4 3.5 3];
NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[1.5];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFCTot=[1];
NumFC=length(kWFCTot);
```

Los resultados obtenidos se muestran en las figuras 6-33 y 6-34. Se observan un total de 12 alternativas, que se corresponden con las 12 diferentes potencias que se van a evaluar para las energías renovables (7 para energía fotovoltaica y 5 para la energía eólica), ya que únicamente se está evaluando el vehículo con pila de combustible.

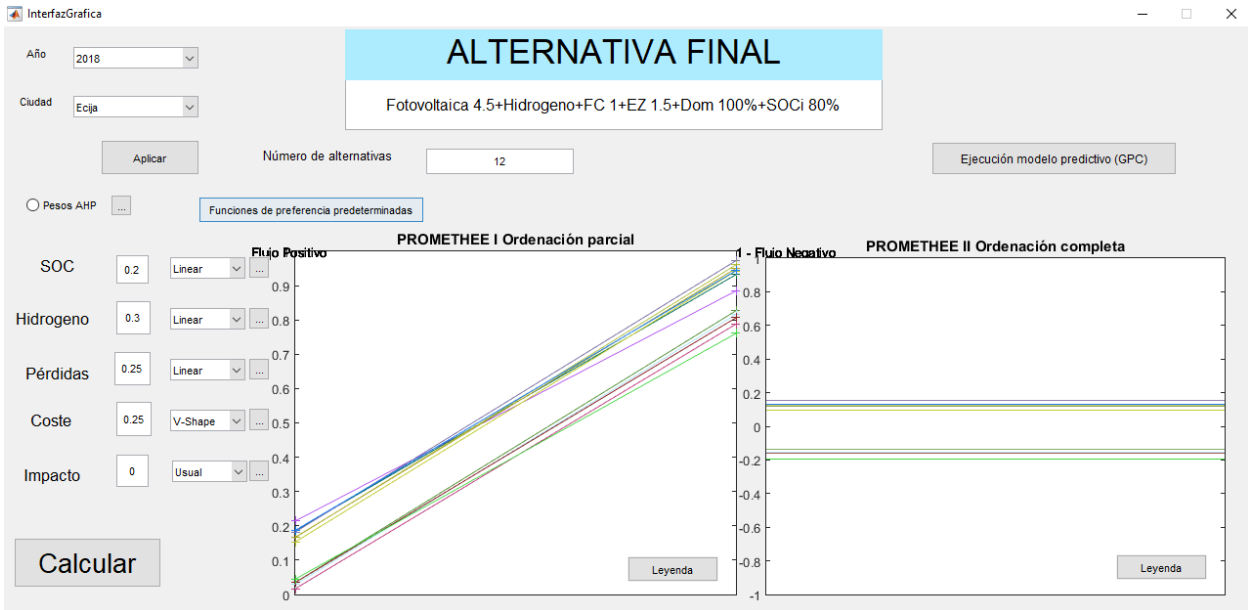


Figura 6-33 Evaluación de Écija para distintas potencias nominales de la generación para 2018

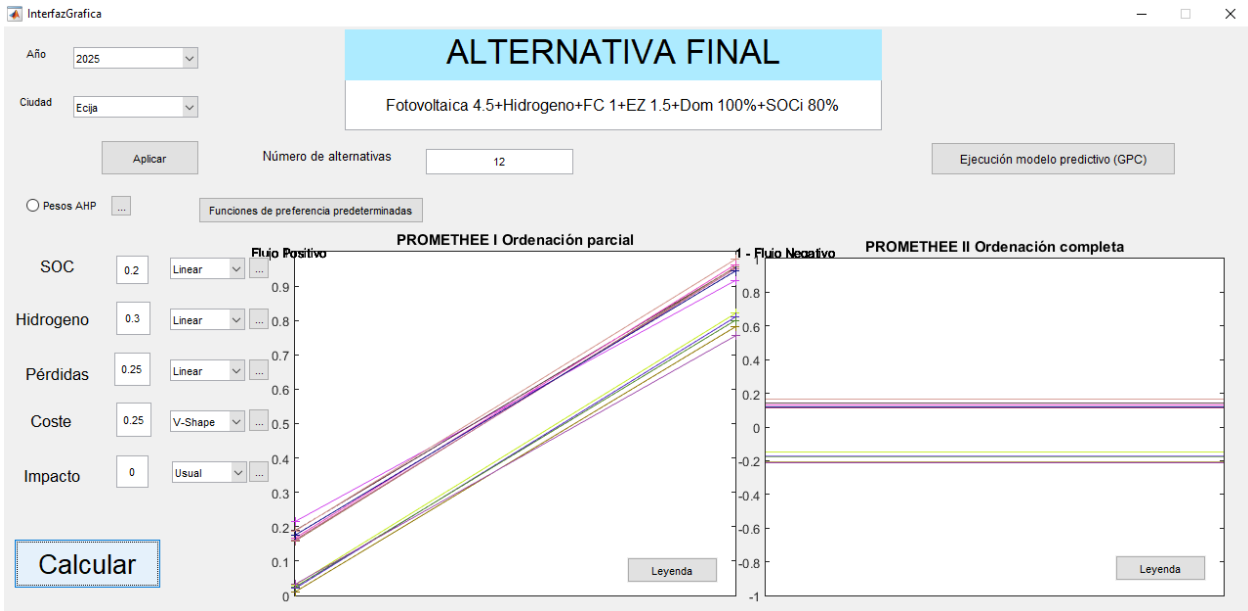


Figura 6-34 Evaluación de Écija para distintas potencias nominales de la generación para 2025

Se puede observar como la potencia nominal óptima no es la máxima que se ha considerado. Esto es debido a la importancia que ha adquirido en esta evaluación las pérdidas y el coste de la generación. Para estas condiciones, la alternativa óptima se basa en una generación fotovoltaica con una potencia nominal de 4,5 kW. Además, el año impuesto no altera la elección de la alternativa más adecuada, ya que se puede ver cómo para ambos años el resultado obtenido es el mismo.

Cabe destacar que se produce cruzamiento en las alternativas evaluadas. Por tanto, la alternativa es escogida por el método Promethee II, siendo aquella que tiene un flujo neto superior.

6.5.2 Evaluación para la localidad de Tarifa

Para la localidad de Tarifa, el código correspondiente a la escritura de las distintas alternativas queda como sigue:

```
%SOC inicial
```

```

Porcentaje_SOC=[80];
NumSOC=length(Porcentaje_SOC);
%Potencia de la generación
kWPVTot=[6 5.5 5 4.5 4 3.5 3];
NumPotPV=length(kWPVTot);
kWEolTot=[5 4.5 4 3.5 3];
NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[2.5];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFCTot=[1.5];
NumFC=length(kWFCTot);
    
```

Los resultados obtenidos se muestran en las figuras 6-35 y 6-36. Como se ha mencionado anteriormente para la localidad de Écija, se evalúan un total de 12 alternativas.

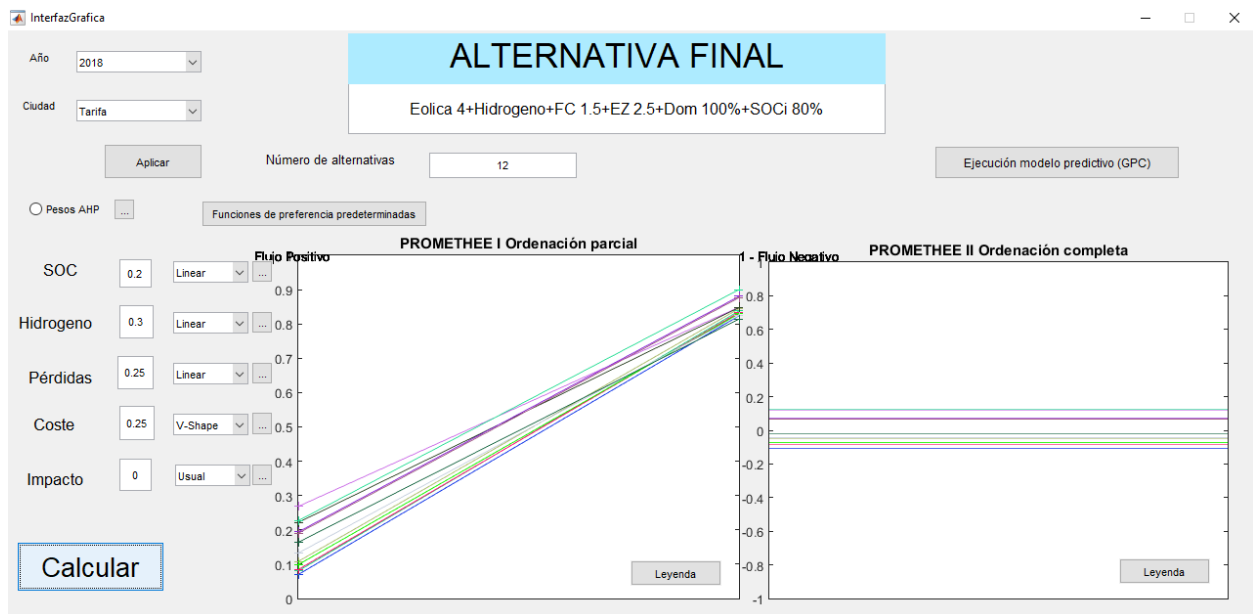


Figura 6-35 Evaluación de Tarifa para distintas potencias nominales de la generación para 2018

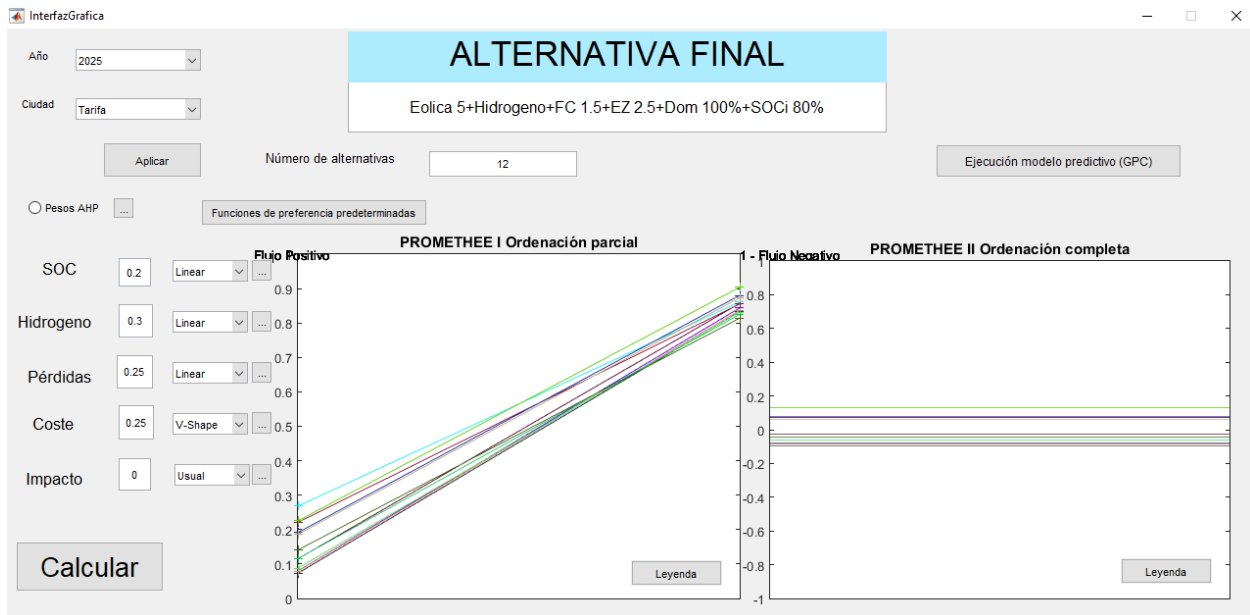


Figura 6-36 Evaluación de Tarifa para distintas potencias nominales de la generación para 2018

En este caso, obtenemos una potencia nominal de 4 kW para 2018 y una de 5 kW para 2025. Por tanto, en este caso es un parámetro fundamental el año de implantación de la planta. El hecho de que exista esta diferencia entre ambos años se debe a que, como en 2025 la energía eólica tendrá un menor coste, la implantación de una planta de mayor potencia supone un beneficio debido a que la capacidad de satisfacer una mayor demanda no se ve penalizada por el mayor coste.

También se puede ver como se produce cruzamiento en el método Promethee I con varias alternativas, haciendo que la elección de la alternativa se produzca a partir del método Promethee II.

6.6 Aplicación de control predictivo en la Microgrid

Una vez obtenido el caso óptimo de la planta en el último apartado, se hará uso de la opción de representar la evolución del sistema llevado a cabo por la estrategia de control predictivo, estableciendo un estado de referencia. Con este control, por tanto, se difiere de la estructura seguida hasta ahora, y en lugar del control por histéresis se llevará a cabo un control predictivo.

Este control será para cada 10 minutos, es decir, 144 puntos al día deben ser impuestos. A partir de esta estrategia de control predictivo, se busca que el estado de la batería sea del 60% excepto en las horas centrales del día, desde las 12:00h hasta las 18:20h, que será del 55% y el del tanque de hidrógeno del 45% excepto, al igual que la batería, un valor del 55% en las horas centrales del día, desde las 12:00h hasta las 18:20h.

A la hora de representar la evolución de la estrategia de control predictivo, se podrá hacer para cada evaluación multicriterio que se realice a partir de la opción incluida en la interfaz. Al pulsar el botón '*Ejecución Modelo Predictivo (GPC)*' se realizará la simulación diaria a partir de la estrategia de control predictivo, y se mostrarán dos figuras. La primera representará la demanda y la generación, mientras que la segunda muestra el estado de la batería y del tanque de hidrógeno junto con la potencia suministrada o absorbida tanto por la batería como por el circuito de hidrógeno y la potencia que debe ser suministrada por la red, además de la diferencia entre la generación y la demanda de la Microgrid, que indicará cuanta energía debe ser suministrada o absorbida, según el momento del día.

En este apartado, se compararán los resultados obtenidos a partir de la demanda y la generación establecida por el departamento de Ingeniería de Sistemas y Automática de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla y la evolución que sigue la planta con este control predictivo con la evolución de la planta a partir de los datos de demanda (generales para Tarifa y Écija) y los de generación que han sido obtenidos en el apartado 6.5.

La estrategia de control implantada por este departamento se define por las siguientes características:

- Variables de entrada: potencia absorbida por el electrolizador o generada por la pila de combustible (Ph2) y potencia inyectada o absorbida de la red (Pgrid).
- Variables de salida: estado de la batería (SOC (%)) y del tanque de hidrógeno (MHL (%)).
- Referencias de la salida: 60% para el estado de la batería y 45% para el tanque de hidrógeno. Ambas referencias se ven modificadas desde el punto 72 (12:00h) hasta el 110 (18:20h), tomando un valor del 55%.
- Esfuerzo de control cuadrático: $\lambda=1$.
- Horizontes de control y predicción: el horizonte de control para cada variable será de 24, así como el horizonte de predicción.
- Restricciones en las variables de entrada: el valor de Ph2 estará entre -0,9 kW y 0,9 kW y el valor de Pgrid estará entre -2,5 kW y 2,5 kW.
- Restricciones en las variables de salida: el valor de SOC estará entre el 40% y el 80% y el de MHL entre el 10% y el 70%.
- Optimizador: la optimización se realizará a partir de la función '*miqp.m*' de Matlab, que se corresponde con un '*Mixed Integer Quadratic Program*', un integrador cuadrático.

Los datos de generación y demanda que se van a evaluar serán, por tanto, la alternativa que se basa en la generación fotovoltaica de 4,5 kW, vehículo con pila de combustible, una pila de combustible de 1 kW, electrolizador de 1,5kW y un estado inicial de la batería del 80% en el caso de Écija, y la generación será eólica de 4 kW (al estar evaluando el año 2018), el vehículo será con pila de combustible, una pila de combustible de 1,5 kW, un electrolizador de 2,5 kW y un estado inicial de la batería del 80% para la localidad de Tarifa. Como la demanda y la generación que se obtiene a partir de la interfaz se corresponde con un periodo de 24 horas y se obtienen los datos para cada hora en lugar de los 10 minutos que están estipulados en esta estrategia de control predictivo, supondremos que los datos correspondientes a las 6 particiones de 10 minutos tienen el mismo valor, correspondiente al valor de la hora que corresponda.

Tanto la demanda como la generación de energía se escribirán en los archivos Excel '*GENI.xls*' y '*DEMI.xls*', respectivamente.

A la hora de imponer la demanda, se supone la suma de la demanda doméstica establecida en el documento de referencia [3] y la demanda requerida para la carga del vehículo escogido, distribuida durante las 8 horas que engloban el tramo horario de 23:00h a 6:59h, como se explica en el apartado 2.1.4.

Las figuras 6-37 y 6-38 muestran el resultado obtenido a partir de los datos impuestos por el departamento de Ingeniería de Sistemas y Automática de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla. En la figura 6-37 se puede ver la curva de generación y demanda de energía que se da, mientras que en la figura 6-38 muestra tanto la evolución del nivel de batería y del tanque de hidrógeno como los flujos de potencia necesarios en los distintos equipos (batería, circuito de hidrógeno y red). La figura 6-38b muestra para valores positivos de potencia si la batería o el tanque de hidrógeno se está descargando y negativos para la carga, al igual que para la potencia de la red, que será negativa si se están inyectando potencia a la red y positiva si la red está suministrando energía. Se puede observar como la Microgrid hace uso del circuito de hidrógeno y de la red para los momentos en que la trayectoria de referencia es modificada. La variable '*Pbat*' se corresponde con la suma de las variables '*disturbances*', '*Ph2*' y '*Pgrid*'.

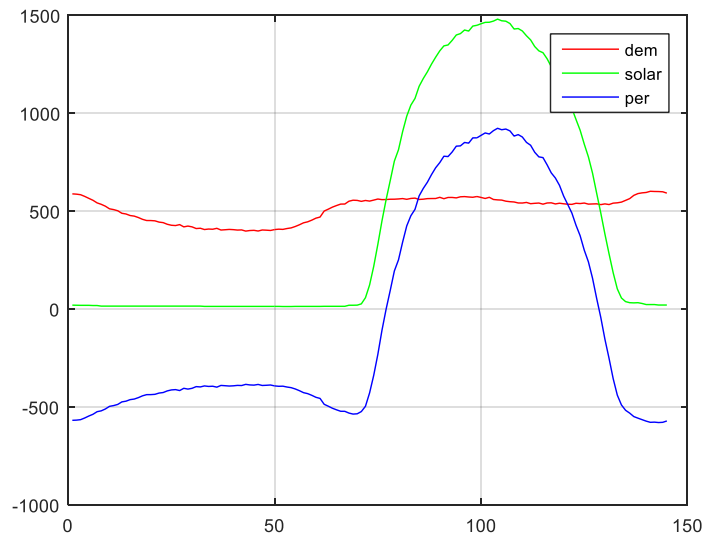


Figura 6-37 Demanda y generación de energía

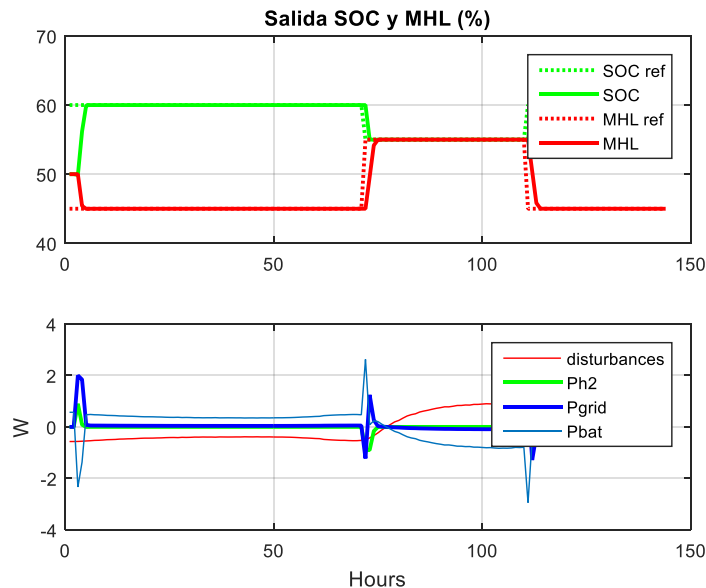


Figura 6-38 a) Estado de la batería (%) y del tanque de hidrógeno (%) al usar el control predictivo.
 b) Representación de los flujos de potencia al usar el control predictivo

6.6.1 Control GPC localidad de Écija

Como se ha comentado en el apartado anterior, la alternativa más adecuada para esta localidad se corresponde con la generación fotovoltaica de 4,5 kW, vehículo con pila de combustible, pila de combustible de 1 kW, electrolizador de 1,5kW y un estado inicial de la batería del 80%. La figura 6-39 muestra la demanda que se debe satisfacer y la generación que se da para este caso. La curva denominada 'per' se corresponde con la diferencia entre ambos. Para el caso en que esta curva se encuentre en valores positivos, indica que la generación es capaz de satisfacer a la demanda y que existe un exceso de potencia, mientras que, si es negativo, se tiene que hacer uso de los dispositivos de almacenamiento.

Se puede observar comparando esta figura con la figura 6-37 como se produce una mayor cantidad de energía, siendo el pico de energía generada para Écija cercano a los 3 kW y para el caso estudiado por el departamento de 1,5 kW. Así mismo, se puede observar como la curva de demanda no tiene la misma forma. La demanda presente en el caso estudiado por el departamento es prácticamente constante y cercana a 0,5 kW, mientras que

la impuesta en este trabajo es superior y tiene una curva que varía durante el día, obteniéndose el valor máximo de 1,375 kW en las horas centrales del día.

En la figura 6-40a se muestra la evolución de los estados tanto de la batería como del nivel del tanque de hidrógeno, donde se puede observar cómo se pretende mantener la batería al 60% y el tanque de hidrógeno al 50%, excepto para el rango horario de 12:00h a 18:20h. La figura 6-40b muestra con la curva 'disturbances' la potencia sobrante de la generación, donde se ve como en las horas centrales del día existe más potencia disponible. Además, se puede ver los momentos del día en que es necesario el uso de potencia ajena a la Microgrid y procedente de la red.

En la figura 6-40b se puede ver una cierta similitud a la figura 6-38b, exceptuando la distribución de la generación y de la demanda de energía, que hacen que el uso de la batería se haga en distintos momentos a lo largo del día. Se comprueba los esfuerzos que hace el control predictivo para mantener las variables de salida lo más cercano posible a la trayectoria de referencia, modificando las señales de control que se envían a las variables de entrada, y, al igual que para el caso estudiado por el departamento, se hace uso de la red y del circuito de hidrógeno cuando se modifica la trayectoria de referencia

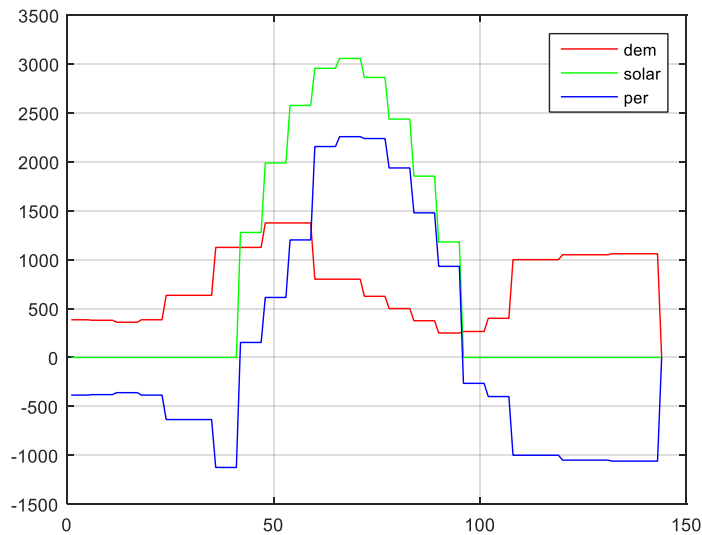


Figura 6-39 Demanda y generación de energía para la localidad de Écija

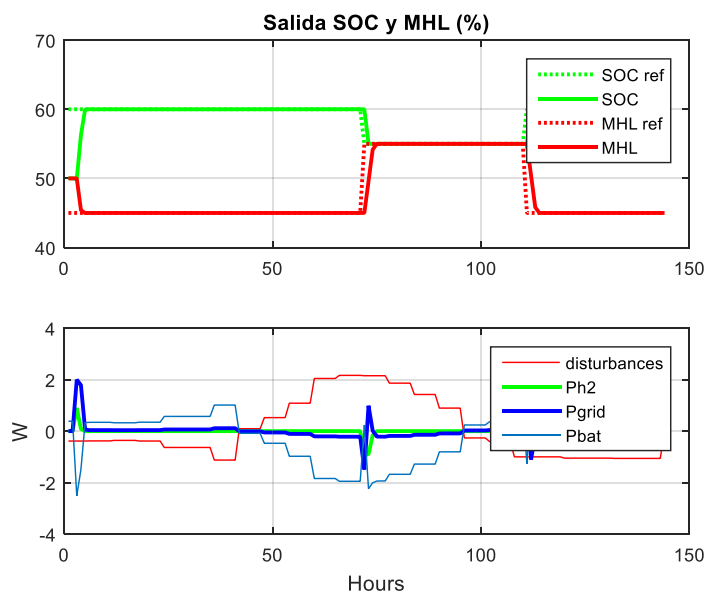


Figura 6-40 a) Estado de la batería (%) y del tanque de hidrógeno (%) para Écija al usar el control predictivo.
 b) Representación de los flujos de potencia para Écija al usar el control predictivo

6.6.2 Control GPC localidad de Tarifa

Como se ha comentado en el apartado anterior, la alternativa más adecuada para esta localidad se corresponde con generación eólica de 4 kW, vehículo con pila de combustible, una pila de combustible de 1,5 kW, un electrolizador de 2,5 kW y un estado inicial de la batería del 80%. La figura 6-41 muestra la demanda y generación de energía correspondiente a esta situación. En comparación con las figuras 6-37 y 6-39 correspondientes a la situación evaluada por el departamento y a la de Écija, se puede observar una generación más constante, generando mayor cantidad de energía en horas donde el Sol no es capaz de generar energía debido a que ya hayan pasado las horas de luz o a que la irradiancia existente no es necesaria para generar energía. Sin embargo, la energía eólica no cuenta con este problema.

La figura 6-42 muestra las ventajas de la generación eólica en un lugar donde se dispone de grandes recursos eólicos como es Tarifa, ya que se ve la gran cantidad de energía que excede a la demanda y que es usada para inyectarla a la red o almacenarla mediante el circuito de hidrógeno o la batería. Al igual que para el caso estudiado por el departamento y de la localidad de Écija, se hace uso de la red y del circuito de hidrógeno cuando se modifica la trayectoria de referencia.

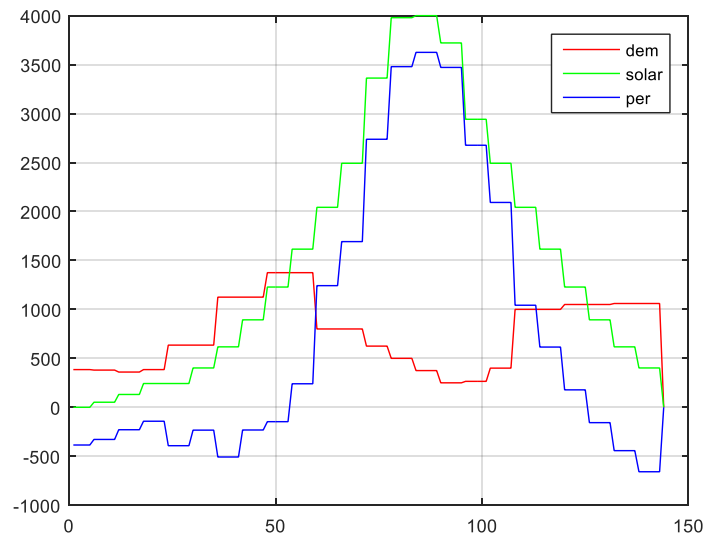


Figura 6-41 Demanda y generación de energía para la localidad de Tarifa

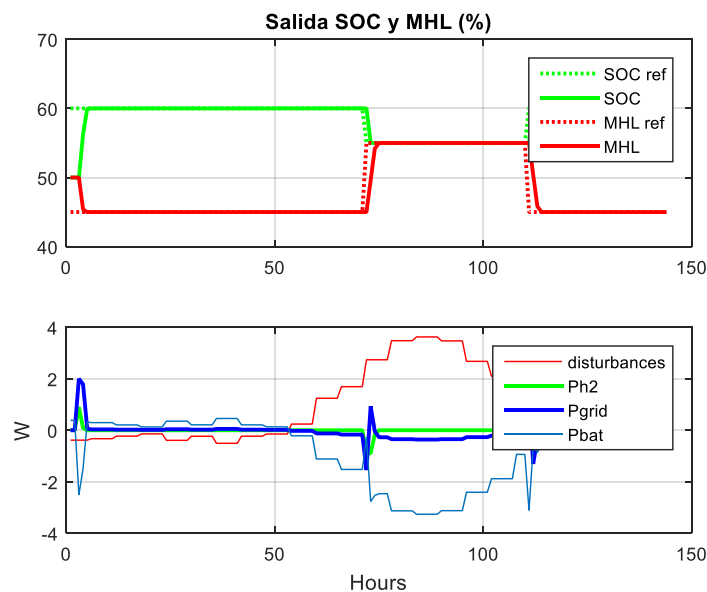


Figura 6-42 a) Estado de la batería (%) y del tanque de hidrógeno (%) para Tarifa al usar el control predictivo.

b) Representación de los flujos de potencia para Tarifa al usar el control predictivo

6.7 Comparación de los resultados obtenidos para el control por histéresis y el control predictivo

En este apartado, se van a comparar los resultados obtenidos a partir de las alternativas óptimas obtenidas en el apartado 6.5 para ambas estrategias de control: el control por histéresis y el control predictivo.

Para ello, en la figura 6-43 se muestra la evaluación en tanto por ciento del nivel de la batería y del tanque de hidrógeno para la alternativa más adecuada tanto en Tarifa como en Écija. A partir de los parámetros que generan la alternativa óptima para cada localidad, se observa como para Écija el nivel de hidrógeno permanece prácticamente constante, mientras que el estado de la batería disminuye del 80% a cerca del 40%. En el caso de Tarifa, el nivel de hidrógeno sube del 30% al 40%, mientras que la batería experimenta un descenso similar al caso de Écija, disminuyendo del 80% al 40%.

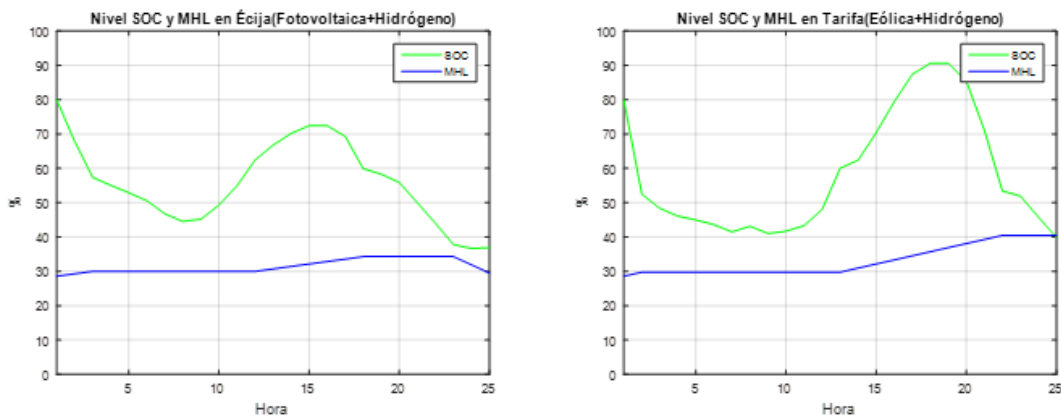


Figura 6-43 Evolución niveles SOC y MHL con control por histéresis para las alternativas óptimas en Écija y Tarifa

Al comparar esta figura con las figuras 6-42 y 6-40, se observa como la evolución del depósito de hidruros es similar para ambas estrategias de control, ya que dicho valor permanece prácticamente inalterado al comparar los valores iniciales y finales. Sin embargo, al imponer una estrategia de control predictivo, el estado de la batería permanece constante, mientras que para el caso del control por histéresis se produce un descenso pronunciado. Este hecho se debe principalmente a que en el control predictivo se ha considerado la posibilidad de usar energía proveniente de la red, mientras que para el caso del control por histéresis no se ha hecho tal consideración. Además, en el control por histéresis no se tiene un modelado de la planta tan preciso como en el control predictivo, ya que se han extrapolado valores experimentales en el circuito de hidrógeno que pueden incitar a errores que exijan una mayor demanda de energía para la generación de hidrógeno y una menor potencia generada a partir de un determinado flujo de hidrógeno.

Sin embargo, se comprueba que ambas estrategias de control son válidas para el funcionamiento de esta Microgrid.

7 CONCLUSIONES

A continuación, se muestran las conclusiones más importantes obtenidas en este trabajo.

7.1 Modelado en OpenDSS y control a partir de Matlab

A partir de ambos programas ha sido posible la monitorización del proceso que se da en la Microgrid. Se ha podido ver que OpenDSS es un software de sencilla programación y que puede dar una estimación de valores tan importantes como las pérdidas presentes o el modo de funcionamiento de la batería o de la generación fotovoltaica. La interacción de este programa con Matlab ha hecho posible efectuar un control por histéresis, de forma que horariamente se actualice el modo de funcionamiento de la planta.

Los resultados son suficientemente aceptables como para poder comparar unas alternativas con otras.

Además, a partir de este modelado, queda demostrada la viabilidad de la generación distribuida y del uso del circuito de hidrógeno como elemento de almacenamiento.

Una posible mejora para este apartado sería la introducción del uso del electrolizador y de la pila de combustible a carga parcial, haciendo uso de ellos cuando se tenga un exceso o un defecto de energía y por tanto funcionando a carga parcial.

7.2 Método de evaluación Promethee y obtención de pesos AHP

En este trabajo se ha podido ver como el método Promethee es capaz de establecer un orden relativo para tantas alternativas como sea requerido. A partir de las diferencias que tienen para cada criterio las distintas alternativas y con multitud de posibles combinaciones, es posible la obtención de una alternativa óptima para nuestros requerimientos.

Por tanto, este método agiliza y simplifica la elección de una alternativa que se ajuste lo máximo posible a las exigencias impuestas por el usuario.

7.3 Interfaz gráfica y automatización de la evaluación

A partir de la interfaz gráfica propuesta, se ha visto la facilidad para modificar los parámetros de evaluación que son requeridos para la misma.

Así mismo, se comprueba que la solución óptima para ambas localidades es el uso del coche por pila de combustible y que la generación energética óptima será la generación fotovoltaica para Écija y la eólica para Tarifa.

Por otro lado, se ha establecido un valor óptimo para el estado inicial de la batería para cada caso, y se ha visto que, para un uso óptimo de la Microgrid, la potencia nominal de los elementos del circuito de hidrógeno debe ser menores de los que actualmente están siendo utilizados.

Por último, observando los datos obtenidos se llega a la conclusión de que la localidad más apropiada para la implantación de una red de generación distribuida sería Tarifa. Esta conclusión era previsible, ya que es bien sabido que en dicha localidad existe una gran disponibilidad de recursos eólicos.

Como posible mejora de la interfaz gráfica propuesta, quedaría la implantación de los planos GAIA en dicha interfaz, con el fin de poder tener una representación visual de la evaluación.

7.4 Control predictivo GPC

El apartado 6.6 pone de manifiesto la viabilidad del uso de la estrategia del control predictivo para esta Microgrid. En este caso, se observa que la implantación de una planta doméstica que funciona a partir de energía eólica en Tarifa tiene una mayor cantidad de recursos que una que funcione a partir de energía fotovoltaica en Écija. Esto se ve reflejado en la mayor cantidad de energía disponible para inyectar a la red y también en la cantidad de energía que puede ser almacenada en las baterías.

Por otro lado, se observa un funcionamiento similar al que se da para el caso de la evaluación realizada por el departamento de Ingeniería de Sistemas y Automática de la Escuela Técnica de Ingeniería de la Universidad de Sevilla, aunque con una potencia instalada mayor y con una demanda superior.

Por último, al comparar la estrategia de control por histéresis y el control predictivo, se considera la posibilidad de mejorar la implantación de la Microgrid en el programa OpenDSS a partir de datos experimentales a partir de los cuales se pueda tener una simulación más precisa de la planta.

ANEXO A: MANEJO DE LA INTERFAZ GRÁFICA DE MATLAB (GUI)

En primer lugar, se escribirá el comando *guide* en el panel principal. Este comando abre una pestaña donde se da la posibilidad de modificar una interfaz gráfica ya existente o de crear una nueva, como vemos en la siguiente imagen.

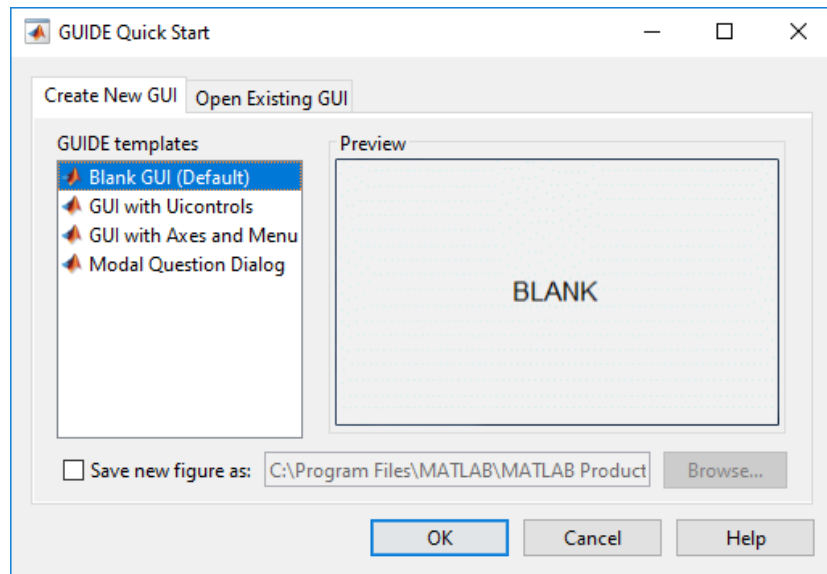


Figura AA-0-1 Inicio rápido de la Interfaz Gráfica

En este caso, se va a crear una nueva interfaz gráfica desde cero, aunque también se podría escoger una de las opciones donde aparecen varios modelos (*GUI with Uicontrols*, *GUI with Axes and Menu* y *Modal Question Dialog*). Al escoger la opción *Blank GUI(Default)*, se muestra el editor de interfaces gráficas, donde se podrán imponer objetos y situarlos donde se requiera, además de establecer las propiedades deseadas.

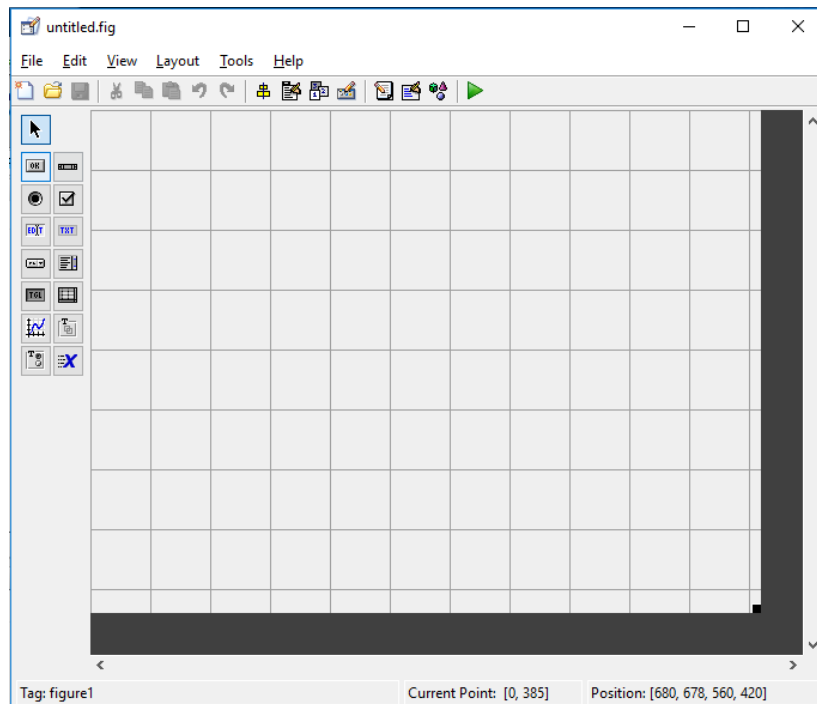












Figura AA-0-2 Ventana de creación de Interfaz Gráfica

En este punto, conviene explicar cada uno de los objetos que aparecen y que pueden ser insertados:

-  **Select:** se trata del cursor. Simplemente sirve para mover los distintos objetos que se incluyan en la interfaz y para visualizar o modificar las propiedades haciendo doble click sobre un objeto.
-  **Push Button:** objeto que actúa como botón. Suele estar acompañado de una subrutina asociada que realice alguna operación o alguna modificación cuando se pulsa sobre este objeto.
-  **Slider:** barra deslizadora. Barra que en función de su posición dentro de la interfaz tiene un valor u otro. A su vez, en función de un valor u otro se puede generar una subrutina asociada que muestre algún tipo de resultado.
-  **Radio Button:** botón con estado ON/OFF. Este estado se corresponde con un valor de 1 (ON) o de 0 (OFF). Este objeto suele tener asociado una subrutina asociada que indique la inclusión o no de algún parámetro o variable en alguna operación.
-  **Check box:** este objeto tiene una función similar al anterior. La diferencia con el objeto anterior es simplemente visual.
-  **Edit text:** texto editable. A partir de este objeto se podrá escribir dentro de la interfaz gráfica con el fin de modificar el valor de algún parámetro y llegar a otro resultado. Hay que tener en cuenta que el valor que se encuentra grabado aquí es del tipo string, por lo que, si queremos usar este objeto para incluir su valor numérico en una función, debemos hacer uso del comando “str2num” o “str2double”.
-  **Static text:** texto estático. Texto que permanecerá inalterado en la interfaz gráfica.
-  **Pop-up menu:** menú desplegable. A partir de la propiedad ‘String’ podemos incluir distintas opciones a elegir, separadas por párrafos. A la hora de seleccionar una alternativa, se modificará la propiedad ‘Value’, tomando un valor de 1 al número de opciones que se hayan puesto.
-  **List box:** lista de opciones. Tiene una función similar a la del objeto anterior, solo que en este caso podemos ver las opciones de la lista con una barra deslizadora situada en el lateral.
-  **Toggle button:** función similar al objeto push button, solo que en este caso al pulsar el botón este pasa a

tener un valor de 1 y adquiere un color azul y al volver a pulsarlo vuelve a su estado inicial (valor de 0 y color grisáceo).



Table: da la posibilidad de mostrar una tabla en la interfaz gráfica. Se puede editar la tabla desde el código de la interfaz gráfica o desde el editor de la interfaz, siempre que se imponga esta condición de edición desde el gestor de propiedades (haciendo doble click sobre el elemento). La propiedad que permite almacenar o cambiar los datos de la tabla es 'data'. Puede representar los valores de una matriz, teniendo en cuenta que los datos aquí almacenados no son del tipo numérico.



Axes: incluye la posibilidad de crear un gráfico. Para poder crear el gráfico en la interfaz, se tiene que llamar a este objeto mediante el código correspondiente a la interfaz gráfica (del cual se hablará más adelante) en primer lugar llamando al objeto con el código *axes(handles.axes1)* (si el objeto axes se llamase axes1) y a continuación mediante la función plot. Este objeto no contiene ninguna subrutina asociada. La función principal será representar alguna función o evolución de alguna variable.



Panel: es un panel sobre el cual se suelen posicionar el resto de los objetos para agrupar el conjunto de la interfaz gráfica.

Todos estos objetos podrán ser insertados en la interfaz gráfica, y para poder imponer el modo de funcionamiento de cada uno de ellos y las relaciones que puedan existir, se tiene que guardar la interfaz.

Una vez guardada la interfaz gráfica se generarán dos archivos: un archivo *.fig* y otro *.m*. El primero ejecutará la interfaz gráfica y será donde se podrá interactuar con ella, mientras que en el segundo se encuentra el código correspondiente a la interfaz gráfica creada, donde se incluirán todas las subrutinas y relaciones entre los objetos. Este código se explicará en el apartado que sigue.

Manejo de una interfaz gráfica

En primer lugar, se van a explicar algunos conceptos con los que el usuario debe familiarizarse antes de empezar con el uso de la interfaz. Estos conceptos son los siguientes:

- **Handles:** estructura general de una interfaz gráfica de usuario (GUI). A través de handles podremos llegar a los distintos objetos que coloquemos en la interfaz gráfica (una gráfica, un botón de actuación, etc.)
- **Callback:** subrutina que se ejecuta debido a la interacción directa del usuario con la interfaz gráfica. Por ejemplo, si la interfaz tiene un botón que al pulsarlo muestra un resultado, el callback de este objeto sería el hecho de mostrar el resultado. No todos los objetos ejecutan callbacks, ya que, por ejemplo, una gráfica no tiene ninguna interacción directa con el usuario, simplemente muestra una función o unos resultados.
- **Variables locales:** variables que se crean dentro de cada objeto. No se podrá hacer uso de estas variables fuera del objeto correspondiente. Hay que prestar mucha atención a estas variables, ya que para el uso de ciertas variables en varios objetos se tiene que buscar algún modo de relacionarlas.
- **Variables globales:** variables definidas dentro del handles, pero no en un objeto en particular, sino englobando toda la estructura general.

En el código generado en el archivo *.m* de la interfaz gráfica se verán distintos apartados, cada uno de ellos generando una función:

- La primera función será del tipo '*function varargout = Nombre_Interfaz(varargin)*'. Crea la interfaz gráfica.
- La segunda función será del tipo '*function Nombre_Interfaz_OpeningFcn(hObject, eventdata, handles, varargin)*'. Función para los parámetros de entrada, donde no aparece ningún parámetro de salida
- La tercera función será del tipo '*function varargout=Nombre_Interfaz_OutputFcn(hObject, eventdata, handles)*'. Esta función se crea para devolver los parámetros de salida.

- El resto de las funciones representarán los valores de entrada y los callbacks correspondientes para cada objeto, según tengan o no, y serán del tipo: *'function edit1_CreateFcn(hObject, eventdata, handles)'* para objetos que requieran valores de entrada (como en este caso, un texto editable) o *'function pushbutton1_Callback(hObject, eventdata, handles)'* para objetos que tengan callbacks (como en este caso, un push button).

Para la asignación de callbacks a distintos objetos y la asignación de valores o propiedades, se va a hacer uso de este código. Aquello que sea escrito entre el espacio existente entre la función que crea un callback y la siguiente función, va a ser ejecutado una vez se haga uso de ese callback al interactuar con la interfaz.

Para hacer uso de valores que se hayan escrito en la interfaz o que tengan ciertos objetos en alguna de sus propiedades o para, contrariamente, imponer valores de propiedades de objetos, se van a usar las funciones *get* y *set*, correspondientemente, seguido del objeto que se requiera *handles.edit1* y de la propiedad que se desea. Por ejemplo, para recoger la información que un texto editable tiene escrita, se usaría el comando *'get(handles.edit1, 'String')*, siempre teniendo en cuenta que este valor es del tipo string.

Con el fin de poder usar una misma variable para distintos objetos o interfaces, se pueden definir las variables globales, como sigue: *'global a'* (crearía la variable global *a*). Esta variable siempre que se quiera usar tendrá que estar precedida del mismo código que para su definición: *'global a'*.

Por último, para poder interactuar con la interfaz hay tres alternativas: ejecutar el código *.m*, ejecutar el código *.fig* o ejecutar la interfaz desde la ventana de edición.

En el siguiente apartado se mostrará un ejemplo para poder explicar con mayor claridad cómo manejar una interfaz gráfica.

Ejemplo de una interfaz gráfica

La interfaz gráfica que se va a ejemplificar se corresponde con una interfaz que muestra la gráfica de una función $f(x)$ elegida a partir de un pop-up menu que da varias opciones ($\sin(x)$, $\cos(x)$ o $\tan(x)$) para una frecuencia que se recoge desde la misma interfaz. Además, se dará la posibilidad de escribir una función $f(x)$ diferente para su representación y se podrán modificar los límites de la gráfica.

En la ventana de edición, esta interfaz queda como sigue:

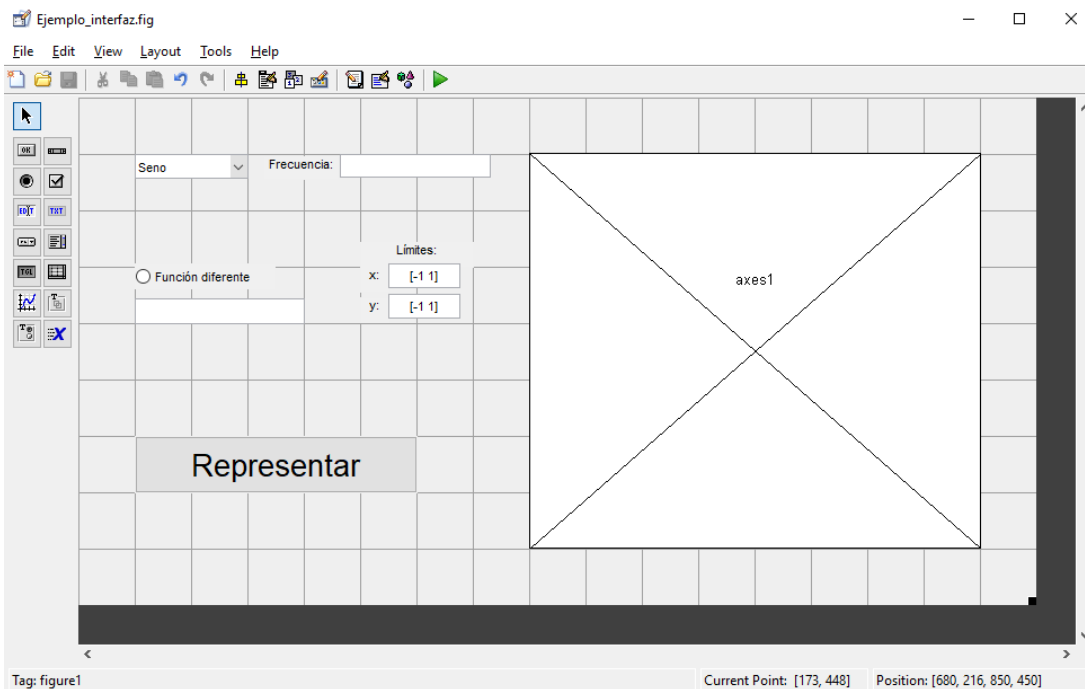


Figura AA-0-3 Creación ejemplo de Interfaz Gráfica

Este interfaz se compone de:

- Cuatro textos editables.
- Cuatro textos estáticos.
- Un pop-up menú.
- Un radiobutton.
- Un togglebutton.
- Un eje.

Al pulsar el botón '*Representar*' se mostrará por pantalla la representación de la función escogida. Para poder representar una función diferente a seno, coseno o tangente el botón '*Función diferente*' debe estar activado y la función escrita en el texto editable que aparece, escrita como si fuese a ser evaluada con un vector, como por ejemplo ' $x.^3+x.^2$ '.

El código *.m* que se ha creado se muestra a continuación, para poder ver el uso de distintas opciones que da el programa.

```
function varargout = Ejemplo_interfaz(varargin)

% EJEMPLO_INTERFAZ MATLAB code for Ejemplo_interfaz.fig
%     EJEMPLO_INTERFAZ, by itself, creates a new EJEMPLO_INTERFAZ or raises
the existing
%     singleton*.
%
%     H = EJEMPLO_INTERFAZ returns the handle to a new EJEMPLO_INTERFAZ or
the handle to
%     the existing singleton*.
%
%     EJEMPLO_INTERFAZ('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in EJEMPLO_INTERFAZ.M with the given input
arguments.
%
%     EJEMPLO_INTERFAZ('Property','Value',...) creates a new
EJEMPLO_INTERFAZ or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before Ejemplo_interfaz_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to Ejemplo_interfaz_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Ejemplo_interfaz

% Last Modified by GUIDE v2.5 08-Jun-2018 12:55:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Ejemplo_interfaz_OpeningFcn, ...
                  'gui_OutputFcn',  @Ejemplo_interfaz_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```

end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Ejemplo_interfaz is made visible.
function Ejemplo_interfaz_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Ejemplo_interfaz (see VARARGIN)

% Choose default command line output for Ejemplo_interfaz
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Ejemplo_interfaz wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Ejemplo_interfaz_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
% contents as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1
x=[-1000:0.001:1000];
lim_x=str2num(get(handles.edit3,'String'));
lim_y=str2num(get(handles.edit4,'String'));
freq=str2double(get(handles.edit1,'String'));
if get(handles.radiobutton1,'Value')==0
    if get(handles.popupmenu1,'Value')==1
        y=sin(freq*x);
    elseif get(handles.popupmenu1,'Value')==2
        y=cos(freq*x);
    elseif get(handles.popupmenu1,'Value')==3
        y=tan(freq*x);
    end
elseif get(handles.radiobutton1,'Value')==1
    a=strcat('@(x)',get(handles.edit2,'String'));
    func=str2func(a);
    y=func(x);
end
axes(handles.axes1)
plot(x,y)
xlim([lim_x])
ylim([lim_y])
grid

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a
double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a
double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a
double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Se puede observar como el único elemento que tiene asignado un callback es el togglebutton. Un ejemplo de uso de la interfaz se muestra a continuación, mostrando por pantalla la función x^3+x^2 , estableciendo los límites en $[-5\ 5]$ para ambos ejes.

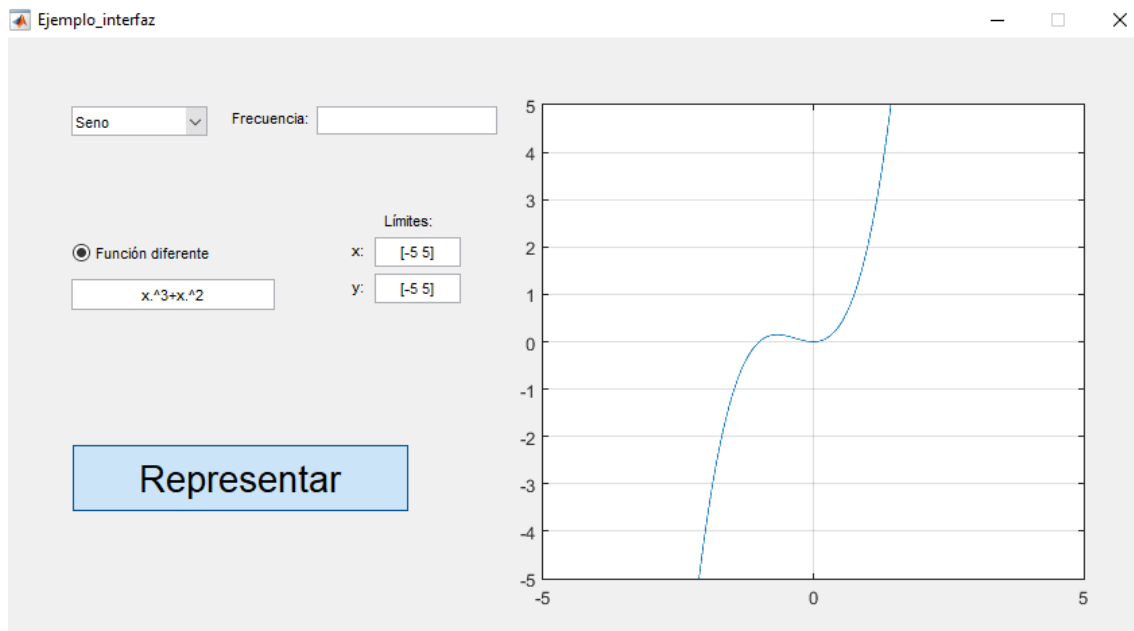


Figura AA-0-4 Ejemplo de Interfaz Gráfica

En el siguiente apartado se mostrará la interfaz creada para la evaluación multicriterio de las distintas alternativas que puede presentar la Microgrid propuesta.

ANEXO B: CÓDIGOS UTILIZADOS EN MATLAB

Código A1-1: Implementación de la Microgrid para generación fotovoltaica

```
function
[Perd, Coste, H, SOC]=Fotovoltaica_Multicriterio(SOC_inicial, kWVeh, kWDom, kWPV, kW
FC, kWEZ, Coste_PV, Ciudad)
    %Ciudad=0 --> Ecija ; Ciudad=1 --> Tarifa
    if Ciudad==0
        irradnu=[0 0 0 0 0 0.0081 0.1382 0.2915 0.4478 0.5833 0.6753
0.7079 0.6753 0.5833 0.4478 0.2915 0.1382 0.0081 0 0 0 0 0 0 0];
        temp=[11 11 10 10 9 9 11 12 13 15 17 19 21 22 22 22 21 20 18 15 14 13
12 12 0];
    end
    if Ciudad==1
        irradnu=[0 0 0 0 0 0.0075 0.1339 0.2828 0.4348 0.5664 0.6559
0.6876 0.6559 0.5664 0.4348 0.2828 0.1339 0.0075 0 0 0 0 0 0 0];
        temp=[11 12 12 12 11 11 13 14 15 17 18 19 21 22 22 22 21 21 18 15 14
13 12 12 0];
    end
    PCar=[1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0]; %El último 0 es
para el cálculo de la Diferencia Oferta-Demanda
    PV=Fotovoltaica(kWPV,Ciudad); %Para saber como se va a comportar la PV
    Pot=sum(PV);
    Coste=Pot*Coste_PV;

    %Condicion inicial. Se supone que se empieza con la batería a más del
    %60%
    EZ=1;
    FC=0;
    %B=1; Va a ser especificado a partir del valor inicial de demanda y de
    %generación.
    MHL_inicial=10;
    MHL=MHL_inicial;
    nivelMHL(1)=MHL;
    SOC=SOC_inicial;
    nivelSOC(1)=SOC_inicial;

    %Rendimiento del electrolizador(experimental)
    %    rend_ez=0.683435556;
    rend_dc dc=0.89;

    %Demanda doméstica
    Dom=[125 120 100 125 375 375 1125 1125 1375 1375 800 800 625 500 375 250
265 400 1000 1000 1050 1050 800 800 0]./1375; % Multiplicadores demanda
doméstica

    %Valor inicial de la batería (suponiendo que es mayor del 60%
    D=Dom(1)*kWDom;
    B=-(PV(1)-EZ*kWEZ/rend_dc dc+FC*kWFC*rend_dc dc-PCar(1)*kWVeh-D);

    DSSObj=actxserver('OpenDSSEngine.DSS');
    DSSObj.Start(0);
    DSSText=DSSObj.Text;
    DSSCircuit=DSSObj.ActiveCircuit;
    DSSSolution=DSSCircuit.Solution;

    DSSText.Command='Clear';
```

```

DSSText.Command='set datapath= "C:\Users\Alvaro\Desktop\ASIGNATURAS
1617\ASIGNATURAS 1718\TFG\MonitoresTFG"';
DSSText.Command='Set DefaultBaseFreq = 0.001';

DSSText.Command='New circuit.MicroHydrogen phases=1 basekv=0.048';

DSSText.Command='New linecode.ejemplo nphases=1 r=2.19e-4';

DSSText.Command='New Line.Line1 Bus1=sourcebus Bus2=b1 phases=1
linecode=ejemplo length=0.01 units=m';
DSSText.Command='New Line.Line2 Bus1=b1 Bus2=b2 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line3 Bus1=b2 Bus2=b2.1 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line4 Bus1=b2 Bus2=b3 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line5 Bus1=b3 Bus2=b3.1 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line6 Bus1=b3 Bus2=b4 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line7 Bus1=b4 Bus2=b4.1 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line8 Bus1=b4 Bus2=b5 phases=1
linecode=ejemplo length=1 units=m';

DSSText.Command='new XYcurve.MyPVst npts=4 xarray=[0 25 75 100]
yarray=[1.2 1 0.8 0.6]';
DSSText.Command='new XYcurve.Myeff npts=4 xarray=[0.1 0.2 0.4 1]
yarray=[0.86 0.9 0.93 0.97]';

DSSText.Command='new Tshape.MyTemp npts=1 interval=1 temp=[11]';
DSSText.Command='new loadshape.MyIrrad npts=1 interval=1 mult=[0]';
DSSText.Command='new loadshape.electrolizador npts=1 interval=1
mult=[1]';
DSSText.Command='new loadshape.StorageShape npts=1 interval=1 mult=[1]';
DSSText.Command='new loadshape.fcell npts=1 interval=1 mult=[0]';
DSSText.Command='new loadshape.Pcar npts=1 interval=1 mult=[1]';
DSSText.Command='new loadshape.Domestica npts=1 interval=1 mult=[0.09]';

DSSText.Command='new PVSystem.PV phases=1 Bus1=b1 kV=0.048 irrads=0.963
temperature=25 basefreq=0.001';
DSSText.Command='pf=1 effcurve=Myeff P-TCurve=MyPVst Duty=MyIrrad
TDuty=MyTemp';
DSSText.Command=['PVSystem.PV.Pmpp=' num2str(kWPV)];
DSSText.Command=['PVSystem.PV.kVA=' num2str(kWPV)];

DSSText.Command='new Load.Car phases=1 Bus1=b5 kV=0.048 pf=1 model=1
basefreq=0.001 Duty=Pcar';
DSSText.Command=['Load.Car.kW=' num2str(kWVeh)];

%Demanda doméstica
DSSText.Command='new Load.Dom phases=1 Bus1=b5 kV=0.048 pf=1 model=1
basefreq=0.001 Duty=Domestica';
DSSText.Command=['Load.Dom.kW=' num2str(kWDom)];

DSSText.Command='new Load.Electrolizador phases=1 Bus1=b2.1 kV=0.048
pf=1 model=1 basefreq=0.001 Duty=electrolizador';
DSSText.Command=['Load.Electrolizador.kW=' num2str(kWEZ/rend_dcdc)];

```

```

DSSText.Command='new Generator.FuelCell phases=1 Bus1=b4.1 kV=0.048 pf=1
model=1 basefreq=0.001 Duty=fcell';
DSSText.Command=['Generator.FuelCell.kW=' num2str(kWFC*rend_dcdc)];

DSSText.Command='new Storage.Bateria phases=1 Bus1=b3.1 kV=0.048
kWrated=1 kW=1 kWhrated=18.92';
DSSText.Command=['Storage.Bateria.kWhstored=' num2str(SOC_inicial)] ;
DSSText.Command='basefreq=0.001 DispMode=follow Duty=StorageShape';

%Impongo el valor inicial de la batería
DSSText.Command=['LoadShape.StorageShape.mult=' num2str(B)];

DSSText.Command='new monitor.pvS PVSsystem.PV mode=1 ppolar=no';
DSSText.Command='new monitor.Fuel Generator.FuelCell mode=1 ppolar=no';
DSSText.Command='new monitor.bateria Storage.Bateria mode=3';

DSSText.Command='set voltagebases=[0.048]';
DSSText.Command='calcv';
for hour=1:24
    DSSText.Command='solve mode=duty stepsize=3600 number=1';

    DSSText.Command='export mon pvS';
    DSSText.Command='export mon bateria';
    DSSText.Command='export mon Fuel';
    DSSText.Command='export losses';

    %Perdidas en la batería
    datosBAT(:, :)=textread('MicroHydrogen_Mon_bateria.csv',
'', 'delimiter', ',', 'headerlines', 1, 'emptyvalue', NaN);
    perdidasBAT(hour)=datosBAT(:, 7);

    %Perdidas en el circuito (líneas)
    SystemLosses=(DSSCircuit.Losses);
    perdidasCircuito(hour)=SystemLosses(1)/1000;

    %Valor de kWhStored en la batería
    kWhStored(hour)=datosBAT(:, 3);

    %Perdidas en EZ y FC
    if EZ==1
        perdidasEZ(hour)=kWEZ*(1-rend_dcdc)/rend_dcdc;
    else
        perdidasEZ(hour)=0;
    end

    if FC==1
        perdidasFC(hour)=kWFC*(1-rend_dcdc);
    else
        perdidasFC(hour)=0;
    end

    if B<0 %Batería cargando
        kWbATERIA(hour)=datosBAT(:, 6);
    end

    if B>0 %Batería descargando
        kWbATERIA(hour)=datosBAT(:, 5);
    end

    if B==0

```

```

        kWATERIA(hour)=0;
    end

    MHL=MHL+EZ*0.5/3*kWEZ-FC*1.2534/1.5*kWFC;
    SOC=kWhStored(hour)+kWATERIA(hour)-perdidasBAT(hour);

    if SOC>18.92
        SOC=18.92;
    end
    %Variacion de los niveles del hidrógeno y de la bateria
    %con el tiempo
    nivelMHL(hour+1)=MHL;
    nivelSOC(hour+1)=SOC;

    if SOC > 0.75*18.92
        disp('Electrolizador ON')
        EZ=1;
        FC=0;
    end

    if SOC < 0.75*18.92
        if SOC > 0.60*18.92
            disp('Electrolizador ON')
            EZ=1;
            FC=0;
        end
    end
    if SOC < 0.60*18.92
        if SOC > 0.4*18.92
            disp('SIN CONTROL')
            EZ=0;
            FC=0;
        end
    end
    if SOC < 0.4*18.92
        disp('Pila de combustible ON')
        EZ=0;
        FC=1;
    end

    Dif_OfertaDemanda=PV(hour+1)+FC*kWFC/rend_dc-dc-EZ*kWEZ*rend_dc-dc-
    PCar(hour+1)*kWVeh-D*kWDom;
    B=-Dif_OfertaDemanda;

    T=temp(hour+1);
    DSSText.Command=['TShape.MyTemp.temp=' num2str(T)];

    I=irradnu(hour+1);
    DSSText.Command=['LoadShape.MyIrrad.mult=' num2str(I)];

    C=PCar(hour+1);
    DSSText.Command=['LoadShape.Pcar.mult=' num2str(C)];

    DSSText.Command=['LoadShape.electrolizador.mult=' num2str(EZ)];

```



```

Coste=Pot*Coste_Eolica;

%Condicion inicial. Se supone que se empieza con la batería a más del
%60%
EZ=1;
FC=0;
MHL_inicial=10;
MHL=MHL_inicial;
nivelMHL(1)=MHL;
SOC=SOC_inicial;
nivelSOC(1)=SOC_inicial;
%Rendimiento del electrolizador(experimental)
rend_dcdc=0.89;

%Demanda doméstica
Dom=[125 120 100 125 375 375 1125 1125 1375 1375 800 800 625 500 375 250
265 400 1000 1000 1050 1050 800 800 0]./1375; % Multiplicadores demanda
doméstica

%Valor inicial de la batería (suponiendo que es mayor del 60%)
D=Dom(1)*kWDom;
B=-(PotEol(1)-EZ*kWEZ/rend_dcdc+FC*kWFC*rend_dcdc-PCar(1)*kWVeh-D);

DSSObj=actxserver('OpenDSSEngine.DSS');
DSSObj.Start(0);
DSSText=DSSObj.Text;
DSSCircuit=DSSObj.ActiveCircuit;
DSSSolution=DSSCircuit.Solution;

DSSText.Command='Clear';
DSSText.Command='set datapath= "C:\Users\Alvaro\Desktop\ASIGNATURAS
1617\ASIGNATURAS 1718\TFG\MonitoresTFG"';
DSSText.Command='Set DefaultBaseFreq = 0.001';

DSSText.Command='New circuit.MicroHydrogen phases=1 basekv=0.048';

DSSText.Command='New linecode.ejemplo nphases=1 r=2.19e-4';

DSSText.Command='New Line.Line1 Bus1=sourcebus Bus2=b1 phases=1
linecode=ejemplo length=0.01 units=m';
DSSText.Command='New Line.Line2 Bus1=b1 Bus2=b2 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line3 Bus1=b2 Bus2=b2.1 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line4 Bus1=b2 Bus2=b3 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line5 Bus1=b3 Bus2=b3.1 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line6 Bus1=b3 Bus2=b4 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line7 Bus1=b4 Bus2=b4.1 phases=1
linecode=ejemplo length=1 units=m';
DSSText.Command='New Line.Line8 Bus1=b4 Bus2=b5 phases=1
linecode=ejemplo length=1 units=m';

DSSText.Command='new XYcurve.MyPVst npts=4 xarray=[0 25 75 100]
yarray=[1.2 1 0.8 0.6]';
DSSText.Command='new XYcurve.Myeff npts=4 xarray=[0.1 0.2 0.4 1]
yarray=[0.86 0.9 0.93 0.97]';

```



```

DSSText.Command='new loadshape.MyWind npts=1 interval=1 mult=[0]';
DSSText.Command='new loadshape.electrolizador npts=1 interval=1
mult=[1]';
DSSText.Command='new loadshape.StorageShape npts=1 interval=1 mult=[1]';
DSSText.Command='new loadshape.fcell npts=1 interval=1 mult=[0]';
DSSText.Command='new loadshape.Pcar npts=1 interval=1 mult=[1]';
DSSText.Command='new loadshape.Domestica npts=1 interval=1
mult=[0.125]';

DSSText.Command='new generator.WIND phases=1 Bus1=b1 kV=0.048
basefreq=0.001';
DSSText.Command='pf=1 Duty=MyWind';
DSSText.Command=['generator.WIND.kW=' num2str(kWEol)];

DSSText.Command='new Load.Car phases=1 Bus1=b5 kV=0.048 pf=1 model=1
basefreq=0.001 Duty=Pcar';
DSSText.Command=['Load.Car.kW=' num2str(kWVeh)];

%Demanda doméstica
DSSText.Command='new Load.Dom phases=1 Bus1=b5 kV=0.048 pf=1 model=1
basefreq=0.001 Duty=Domestica';
DSSText.Command=['Load.Dom.kW=' num2str(kWDom)];

DSSText.Command='new Load.Electrolizador phases=1 Bus1=b2.1 kV=0.048
pf=1 model=1 basefreq=0.001 Duty=electrolizador';
DSSText.Command=['Load.Electrolizador.kW=' num2str(kWEZ/rend_dc dc)];

DSSText.Command='new Generator.FuelCell phases=1 Bus1=b4.1 kV=0.048 pf=1
model=1 basefreq=0.001 Duty=fcell';
DSSText.Command=['Generator.FuelCell.kW=' num2str(kWFC*rend_dc dc)];

DSSText.Command='new Storage.Bateria phases=1 Bus1=b3.1 kV=0.048
kWrated=1.5 kW=1 kWhrated=18.92';
DSSText.Command=['Storage.Bateria.kWhstored=' num2str(SOC_inicial)];
DSSText.Command='basefreq=0.001 DispMode=follow Duty=StorageShape';

%Impongo el valor inicial de la batería
DSSText.Command=['LoadShape.StorageShape.mult=' num2str(B)];

DSSText.Command='new monitor.wind Generator.WIND mode=1 ppolar=no';
DSSText.Command='new monitor.Fuel Generator.FuelCell mode=1 ppolar=no';
DSSText.Command='new monitor.bateria Storage.Bateria mode=3';

DSSText.Command='set voltagebases=[0.048]';
DSSText.Command='calcv';
for hour=1:24
    DSSText.Command='solve mode=duty stepsize=3600 number=1';

    DSSText.Command='export mon wind';
    DSSText.Command='export mon bateria';
    DSSText.Command='export mon Fuel';
    DSSText.Command='export losses';

    %Perdidas en la batería
    datosBAT(:,:)=textread('MicroHydrogen_Mon_bateria.csv',
'', 'delimiter', ',', 'headerlines', 1, 'emptyvalue', NaN);
    perdidasBAT(hour)=datosBAT(:,7);

    %Perdidas en el circuito (líneas)
    SystemLosses=(DSSCircuit.Losses);

```

```

perdidasCircuito(hour)=SystemLosses(1)/1000;

%Valor de kWhStored en la bateria
kWhStored(hour)=datosBAT(:,3);

%Perdidas en EZ y FC
if EZ==1
    perdidasEZ(hour)=kWEZ*(1-rend_dc/dc)/rend_dc/dc;
else
    perdidasEZ(hour)=0;
end

if FC==1
    perdidasFC(hour)=kWFC*(1-rend_dc/dc);
else
    perdidasFC(hour)=0;
end

if B<0 %Bateria cargando
    kWATERIA(hour)=datosBAT(:,6);
end

if B>0 %Bateria descargando
    kWATERIA(hour)=datosBAT(:,5);
end

if B==0
    kWATERIA(hour)=0;
end

MHL=MHL+EZ*0.5/3*kWEZ-FC*1.2534/1.5*kWFC;
SOC=kWhStored(hour)+kWATERIA(hour)-perdidasBAT(hour);

%Variacion de los niveles del hidrógeno y de la bateria
%con el tiempo
if SOC>18.92
    SOC=18.92;
end
nivelMHL(hour+1)=MHL;
nivelSOC(hour+1)=SOC;

if SOC > 0.75*18.92
    disp('Electrolizador ON')
    EZ=1;
    FC=0;
end

if SOC < 0.75*18.92
    if SOC > 0.6*18.92
        disp('Electrolizador ON')
        EZ=1;
        FC=0;
    end
end

if SOC < 0.60*18.92
    if SOC > 0.4*18.92
        disp('SIN CONTROL')
        EZ=0;
    end
end

```

```

        FC=0;
    end
end
if SOC < 0.4*18.92
    disp('Pila de combustible ON')
    EZ=0;
    FC=1;
end

    Dif_OfertaDemanda=PotEol(hour+1)+FC*kWFC/rend_dc dc-
EZ*kWEZ*rend_dc dc-PCar(hour+1)*kWVeh-D*kWDom;
    B=-Dif_OfertaDemanda;

    W=wind(hour+1);
    DSSText.Command=['LoadShape.MyWind.mult=' num2str(W)];

    C=PCar(hour+1);
    DSSText.Command=['LoadShape.Pcar.mult=' num2str(C)];

    DSSText.Command=['LoadShape.electrolizador.mult=' num2str(EZ)];

    DSSText.Command=['LoadShape.fcell.mult=' num2str(FC)];

    DSSText.Command=['LoadShape.StorageShape.mult=' num2str(B)];

    D=Dom(hour+1);
    DSSText.Command=['LoadShape.Domestica.mult=' num2str(D)];

end
% nivelSOC
% nivelMHL

% figure
% hour=1:25;
% plot(hour,nivelSOC,'g',hour,nivelMHL,'b')
% xlabel('Hora')
% ylabel('kWh / Nm3')
% legend('SOC','MHL')
% grid
% %
PerdidasBATdia=sum(perdidasBAT);
PerdidasCircuitodia=sum(perdidasCircuito);
PerdidasFCdia=sum(perdidasFC);
PerdidasEZdia=sum(perdidasEZ);

%Asignamos el valor de las perdidas totales a Perd
Perd=PerdidasBATdia+PerdidasCircuitodia+PerdidasFCdia+PerdidasEZdia;

H=nivelMHL(25)-MHL_inicial;
SOC=nivelSOC(25)-SOC_inicial;

end

```

Código A1-3: Obtención potencia fotovoltaica generada diaria

```

function PV=Fotovoltaica(kWPV,Ciudad)

DSSObj=actxserver('OpenDSSEngine.DSS');
DSSObj.Start(0);
DSSText=DSSObj.Text;
DSSCircuit=DSSObj.ActiveCircuit;
DSSSolution=DSSCircuit.Solution;

DSSText.Command='Clear';
DSSText.Command='set datapath= "C:\Users\Alvaro\Desktop\ASIGNATURAS
1617\ASIGNATURAS 1718\TFG\MonitoresTFG"';
DSSText.Command='Set DefaultBaseFreq = 0.001';

DSSText.Command='New circuit.Fotovoltaica phases=1 basekv=0.048';

DSSText.Command='New linecode.FV nphases=1 r=2.19e-4';
DSSText.Command='New Line.Line1 Bus1=sourcebus Bus2=b1 phases=1 linecode=FV
length=0.01 units=m';

DSSText.Command='new XYcurve.MyPVst npts=4 xarray=[0 25 75 100] yarray=[1.2 1
0.8 0.6]';
DSSText.Command='new XYcurve.Myeff npts=4 xarray=[0.1 0.2 0.4 1] yarray=[0.86
0.9 0.93 0.97]';
if Ciudad==0 %Ecija
    DSSText.Command='new Tshape.MyTemp npts=24 interval=1 temp=[11 11 10 10 9
9 11 12 13 15 17 19 21 22 22 22 21 20 18 15 14 13 12 12]';
    DSSText.Command='new loadshape.MyIrrad npts=24 interval=1 mult=[0 0 0 0 0
0.0081 0.1382 0.2915 0.4478 0.5833 0.6753 0.7079 0.6753 0.5833 0.4478
0.2915 0.1382 0.0081 0 0 0 0 0 0]';
else %Tarifa
    DSSText.Command='new Tshape.MyTemp npts=24 interval=1 temp=[12 12 12 12
11 11 13 14 15 17 18 19 21 22 22 22 21 21 18 15 14 13 12 12]';
    DSSText.Command='new loadshape.MyIrrad npts=24 interval=1 mult=[0 0 0 0 0
0.0075 0.1339 0.2828 0.4348 0.5664 0.6559 0.6876 0.6559 0.5664 0.4348
0.2828 0.1339 0.0075 0 0 0 0 0 0]';
end
DSSText.Command='new PVSystem.PV phases=1 Bus1=b1 kV=0.048 irrads=0.963
temperature=25 basefreq=0.001';
DSSText.Command='pf=1 effcurve=Myeff P-TCurve=MyPVst Duty=MyIrrad
TDuty=MyTemp';
% kWPV=6;
DSSText.Command=['PVSystem.PV.Pmpp=' num2str(kWPV)];
DSSText.Command=['PVSystem.PV.kVA=' num2str(kWPV)];

DSSText.Command='new monitor.pvS PVSystem.PV mode=1 ppolar=no';

DSSText.Command='set voltagebases=[0.048]';
DSSText.Command='calcv';

DSSText.Command='solve mode=duty stepsize=3600 number=24';
DSSText.Command='export mon pvS';

datosPV(:,:)=textread('Fotovoltaica_Mon_pvS.csv', '', 'delimiter', ',', '
headerlines', 1, 'emptyvalue', NaN);
for i=1:24
    PV(i)=-datosPV(i,3);
end

```

```
PV(25)=0;
end
```

Código A1-4: Obtención potencia eólica generada diaria

```
function WIND=Eolica(kWWIND,Ciudad)

DSSObj=actxserver('OpenDSSEngine.DSS');
DSSObj.Start(0);
DSSText=DSSObj.Text;
DSSCircuit=DSSObj.ActiveCircuit;
DSSSolution=DSSCircuit.Solution;

DSSText.Command='Clear';
DSSText.Command='set datapath= "C:\Users\Alvaro\Desktop\ASIGNATURAS
1617\ASIGNATURAS 1718\TFG\MonitoresTFG"';
DSSText.Command='Set DefaultBaseFreq = 0.001';

DSSText.Command='New circuit.Eolica phases=1 basekv=0.048';

DSSText.Command='New linecode.WIND nphases=1 r=2.19e-4';
DSSText.Command='New Line.Line1 Bus1=sourcebus Bus2=b1 phases=1 linecode=WIND
length=0.01 units=m';
if Ciudad==0
DSSText.Command='new loadshape.MyWind npts=24 interval=1 mult=[0 0
0.0129601672164600 0.0129601672164600 0.0129601672164600 0.0327692089248149
0.0327692089248149 0.0607198632868300 0.0607198632868300 0.100471820487435
0.154345718593935 0.223323143556015 0.307046629205736 0.403819657257537
0.510606657308233 0.307046629205736 0.223323143556015 0.154345718593935
0.154345718593935 0.100471820487435 0.100471820487435 0.0607198632868300
0.0607198632868300 0.0327692089248149]';
else
DSSText.Command='new loadshape.MyWind npts=24 interval=1 mult=[0
0.0129601672164600 0.0327692089248149 0.0607198632868300 0.0607198632868300
0.100471820487435 0.154345718593935 0.223323143556015 0.307046629205736
0.403819657257537 0.510606657308233 0.623033006837020 0.840610003657521
0.994772625200141 1 0.930316145319512 0.735385031205466 0.623033006837020
0.510606657308233 0.403819657257537 0.307046629205736 0.223323143556015
0.154345718593935 0.100471820487435]';
end

DSSText.Command='new generator.WIND phases=1 Bus1=b1 kV=0.048
basefreq=0.001';
DSSText.Command='pf=1 Duty=MyWind';
% kWPV=6;
DSSText.Command=['generator.WIND.kW=' num2str(kWWIND)];

DSSText.Command='new monitor.windS generator.WIND mode=1 ppolar=no';

DSSText.Command='set voltagebases=[0.048]';
DSSText.Command='calcv';

DSSText.Command='solve mode=duty stepsize=3600 number=24';
DSSText.Command='export mon windS';

datosWIND(:,:)=textread('Eolica_Mon_windS.csv', '', 'delimiter', ',', '
headerlines', 1, 'emptyvalue', NaN);
```

```

for i=1:24
    WIND(i)=-datosWIND(i,3);
end
WIND(25)=0;
end

```

Código A1-5: Escritura de los datos en Excel

```

function NumAlternativas=Escritura_datos_excel(Ciudad,ano)
i=1;
global i_PV
i_PV=1;

if ano==2018
    Coste_PV=0.13;
    Coste_Eol=0.07;
elseif ano==2019
    Coste_PV=0.11;
    Coste_Eol=0.07;
elseif ano==2020
    Coste_PV=0.10;
    Coste_Eol=0.07;
elseif ano==2021
    Coste_PV=0.09;
    Coste_Eol=0.06;
elseif ano==2022
    Coste_PV=0.08;
    Coste_Eol=0.06;
elseif ano==2023
    Coste_PV=0.08;
    Coste_Eol=0.06;
elseif ano==2024
    Coste_PV=0.07;
    Coste_Eol=0.05;
elseif ano==2025
    Coste_PV=0.07;
    Coste_Eol=0.05;
end

%% Posibles configuraciones

%SOC inicial
Porcentaje_SOC=[80];
NumSOC=length(Porcentaje_SOC);
%Potencia de la generación
kWPVTot=[6];
NumPotPV=length(kWPVTot);
kWEolTot=[5];
NumPotEol=length(kWEolTot);
%Potencia máxima de la demanda
Porcentaje_demanda=[100];
kWDomTot=Porcentaje_demanda/100*1.375;
NumDom=length(kWDomTot);
%Potencia del EZ
kWEZTot=[3];
NumEZ=length(kWEZTot);
%Potencia de la FC
kWFCTot=[1.5];
NumFC=length(kWFCTot);

```



```

for ij=1:NumSOC
    SOC_inicial=Porcentaje_SOC(ij)/100*18.92;% % del máximo
for k=1:NumPotEol
    kWEol=kWEolTot(k);
    Energy={'Eolica '};
    for jj=1:NumDom
        kWDom=kWDomTot(jj);
        Porc_dem=Porcentaje_demanda(jj);
        for jjj=1:NumEZ
            kWEZ=kWEZTot(jjj);
            for jjjj=1:NumFC
                kWFC=kWFCTot(jjjj);
                for Veh=1:NumVeh
                    if Veh==1
                        kWVeh=0.5;
                        VehStr={'PEV'};
                    end
                    if Veh==2
                        kWVeh=0.075;
                        VehStr={'Hibrido'};
                    end
                    if Veh==3
                        kWVeh=0.2601;
                        VehStr={'Hidrogeno'};
                    end
                end
            end
        end
    end
end
end

[Perd(i),Cost(i),H(i),SOC(i)]=Eolica_Multicriterio(SOC_inicial,kWVeh,kWDom,kW
Eol,kWFC,kWEZ,Coste_Eol,Ciudad);
    if H(i)>0 && SOC(i)>0
        Satisfac_Demanda(i)=1;
    else
        Satisfac_Demanda(i)=0;
    end
    if Veh==2
        Impacto(i,1)={'Very bad'};
    else
        Impacto(i,1)={'Very good'};
    end
    Fiabilidad(i,1)={'Very good'};

Alternativa(i)=strcat(Energy,{num2str(kWEol)},{'+'},VehStr,{'+'},{'FC
'},{num2str(kWFC)},{'+'},{'EZ '},{num2str(kWEZ)},{'+'},{'Dom
'},{num2str(Porc_dem)},{'%'},{'+'},{'SOCi
'},{num2str(Porcentaje_SOC(ij))},{'%'});
    datos_gpc(i,1)=i; %Número de alternativa
    datos_gpc(i,2)=kWEol; %kW nom de eólica
    datos_gpc(i,3)=0; %kW nom de fotovoltaica
    datos_gpc(i,4)=kWDom; %kW de doméstica (máx)
    i=i+1;
end
end
end
end
end
end

%El siguiente bucle se usa para eliminar la información de cualquier
escritura que hayamos hecho antes
for limp=1:200
    limpiado(limp,1)={' '};
end

```



```

limpiado(limp,2)={' '};
limpiado(limp,3)={' '};
limpiado(limp,4)={' '};
limpiado(limp,5)={' '};
limpiado(limp,6)={' '};
limpiado(limp,7)={' '};
end
%Escritura datos para gpc
xlswrite('Datos gpc.xls',limpiado,'Hojal','A1');
xlswrite('Datos gpc.xls',datos_gpc,'Hojal','A1');
%Escritura datos para Promethee
xlswrite('C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS 1718\TFG\TFG
2018\Evaluacion Microgrid.xls',limpiado,'Hojal','A6');
xlswrite('C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS 1718\TFG\TFG
2018\Evaluacion Microgrid.xls',Alternativa,'Hojal','A6');
xlswrite('C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS 1718\TFG\TFG
2018\Evaluacion Microgrid.xls',SOC,'Hojal','B6');
xlswrite('C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS 1718\TFG\TFG
2018\Evaluacion Microgrid.xls',H,'Hojal','C6');
xlswrite('C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS 1718\TFG\TFG
2018\Evaluacion Microgrid.xls',Perd,'Hojal','D6');
xlswrite('C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS 1718\TFG\TFG
2018\Evaluacion Microgrid.xls',Cost,'Hojal','E6');
xlswrite('C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS 1718\TFG\TFG
2018\Evaluacion Microgrid.xls',Impacto,'Hojal','F6');

NumAlternativas=i-1;
Exito_escritura_datos;
end

```

Código A1-6: Código Promethee modificado

```

%Nombres archivo de datos y nombres alternativas
Nombre_archivo_datos='C:\Users\Alvaro\Desktop\ASIGNATURAS 1617\ASIGNATURAS
1718\TFG\TFG 2018\Evaluacion Microgrid.xls';
%Ahora, tenemos ya el número de alternativas a partir de la
%función Escritura_datos_excel que hemos ejecutado antes desde la interfaz
FinalAlternativas=num2str(NumAlt+5);
NombreAlternativas=strcat('A6:A',FinalAlternativas);
[nada,Alternativas]=xlsread(Nombre_archivo_datos,NombreAlternativas);
%Señalar la fila donde se encuentran los nombres de los alternativas
disp(Alternativas)
M=size(Alternativas,1);

%Nombre de criterios, tipos y matriz con los valores para las alternativas
para cada criterio
[~,Nombre_criterios]=xlsread(Nombre_archivo_datos,'B1:F1'); %%Señalar también
la fila donde se encuentran los nombres de los criterios
N=size(Nombre_criterios,2); %%N=n° de columnas de Nombre_criterios-->n° de
criterios
%%Fila o columna (positivos o negativos)
tipo=xlsread(Nombre_archivo_datos,'B3:F3'); %: [-1 1 -1 1 1]
[~,tipo_cuant_cual]=xlsread(Nombre_archivo_datos,'B2:F2'); %% (Cuantitativo o
Cualitativo)
MatrizAlternativas=strcat('B6:F',FinalAlternativas);
[G_num,G_cual]=xlsread(Nombre_archivo_datos,MatrizAlternativas);
G_num_str=mat2str(G_num);

if AHP==1; %Los pesos deben ponerse por el método AHP
    global MatrizAHP
    size_AHP=size(MatrizAHP,2);
    for i=1:size_AHP

```

```

        for j=1:size_AHP
            MatrizAHP_num(i,j)=MatrizAHP{i,j};
        end
    end

    w_sin_normalizar=Obtencion_pesosAHP(MatrizAHP_num);
else
    w_sin_normalizar(1)=str2double(get(handles.edit9,'String'));
    w_sin_normalizar(2)=str2double(get(handles.edit2,'String'));
    w_sin_normalizar(3)=str2double(get(handles.edit3,'String'));
    w_sin_normalizar(4)=str2double(get(handles.edit4,'String'));
    w_sin_normalizar(5)=str2double(get(handles.edit5,'String'));
end

[w]=Pesos_distribuido(N,w_sin_normalizar)

%Pairwise Preference Matrix
FP_vector=zeros(1,N);
cont_num=0;
cont_cual=0;
dif=zeros(M);
dif1=zeros(M);
Pairwise_pref_matriz=zeros(M);
G_cual_vector=cell(M,1);
flujo_sup_netto_criterio_alt=zeros(M,N);
Matriz_datos=cell(M,N);
cuant_cual=zeros(1,N);
qps_matriz=zeros(3,N);

global p1 q1 s1 FP_vector
for i=1:N
    fprintf('\nCriterio: %s\n',Nombre_criterios{i})
    tipol=tipo(i);
    FP=FP_vector(i);
    q=q1(i);
    p=p1(i);
    s=s1(i);
    tf1=strcmp('Cuantitativo',tipo_cuant_cual{i});
    tf2=strcmp('Cualitativo',tipo_cuant_cual{i});
    qps_matriz(1,i)=q;
    qps_matriz(2,i)=p;
    qps_matriz(3,i)=s;
    if tf1==1 %Criterio j es cuantitativo
        cuant_cual(i)=1; %Para almacenar qué criterios son cuantitativos
        (asignándoles un 1) y los que son cualitativos (un 0)
        cont_num=cont_num+1;
        G_num_vector=G_num(:,cont_num);
        for j=1:M
            Matriz_datos{j,i}=G_num_vector(j);
        end
        for j=1:M
            sumatorio=0;
            for k=1:M
                dif(j,k)=G_num_vector(j)-G_num_vector(k);
                dif1(k,j)=G_num_vector(k)-G_num_vector(j);
                if tipol==1 %(criterio positivo)
                    d=dif(j,k);
                    d1=dif1(k,j);
                elseif tipol==-1 %(criterio negativo)
                    d=-dif(j,k);
                    d1=-dif1(k,j);
                end
            end
        end
    end
end

```

```

end
[P]=Funcion_preferencia_b(FP,d,q,p,s);
[P1]=Funcion_preferencia_b(FP,d1,q,p,s);
sumatorio=sumatorio+(P-P1);
Pairwise_pref_matriz(j,k)=Pairwise_pref_matriz(j,k)+P*w(i);
end
flujo_sup_netto_criterio_alt(j,i)=sumatorio/(M-1);
end
elseif tf2==1 %Criterio j es cualitativo
cont_cual=cont_cual+1;
for j=1:M
G_cual_vector{j,1}=G_cual{j,cont_cual};
Matriz_datos{j,i}=G_cual{j,cont_cual};
end
[G_cual_vector_numeros]=escala_ordenacion(M,G_cual_vector);
for j=1:M
sumatorio=0;
for k=1:M
dif(j,k)=G_cual_vector_numeros(j)-G_cual_vector_numeros(k);
dif1(k,j)=G_cual_vector_numeros(k)-G_cual_vector_numeros(j);
if tipol==1 %(criterio positivo)
d=dif(j,k);
d1=dif1(k,j);
elseif tipol== -1 %(criterio negativo)
d=-dif(j,k);
d1=-dif1(k,j);
end
[P]=Funcion_preferencia_b(FP,d,q,p,s);
[P1]=Funcion_preferencia_b(FP,d1,q,p,s);
sumatorio=sumatorio+(P-P1);
Pairwise_pref_matriz(j,k)=Pairwise_pref_matriz(j,k)+P*w(i);
end
flujo_sup_netto_criterio_alt(j,i)=sumatorio/(M-1);
end
end
end
disp(Pairwise_pref_matriz)

%Flujos de superación positivo, negativo y neto
flujo_sup_pos=zeros(M,1);
flujo_sup_neg=zeros(M,1);
for i=1:M
sumatorio_pos=0;
sumatorio_neg=0;
for j=1:M
sumatorio_pos=sumatorio_pos+Pairwise_pref_matriz(i,j);
sumatorio_neg=sumatorio_neg+Pairwise_pref_matriz(j,i);
end
flujo_sup_pos(i)=(1/(M-1))*sumatorio_pos; %Flujo de superación positivo
flujo_sup_neg(i)=(1/(M-1))*sumatorio_neg; %Flujo de superación negativo
end
flujo_sup=flujo_sup_pos-flujo_sup_neg; %Flujo de superación neto

%Método Promethee I (ordenación parcial)
a_b=cell(M);
a_b_aux=zeros(M);
for i=1:M
for j=1:M
if (flujo_sup_pos(i)>flujo_sup_pos(j)&&flujo_sup_neg(i)<flujo_sup_neg(j))
a_b{i,j}='P';
a_b_aux(i,j)=1;

```

```

elseif
    (flujo_sup_pos(i)==flujo_sup_pos(j)&&flujo_sup_neg(i)<flujo_sup_neg(j))
    a_b{i,j}='P';
    a_b_aux(i,j)=1;
elseif
    (flujo_sup_pos(i)>flujo_sup_pos(j)&&flujo_sup_neg(i)==flujo_sup_neg(j))
    a_b{i,j}='P';
    a_b_aux(i,j)=1;
elseif flujo_sup_pos(i)==flujo_sup_pos(j)&&flujo_sup_neg(i)==flujo_sup_neg(j)
    a_b{i,j}='=';
elseif (flujo_sup_pos(i)>flujo_sup_pos(j)&&flujo_sup_neg(i)>flujo_sup_neg(j))
    a_b{i,j}='NC';
elseif (flujo_sup_pos(i)<flujo_sup_pos(j)&&flujo_sup_neg(i)<flujo_sup_neg(j))
    a_b{i,j}='NC';
end
end
end
disp(a_b)
orden=cell(M);
for i=1:M
    cont_orden=0;
    for j=1:M
        if a_b_aux(j,i)==1
            cont_orden=cont_orden+1;
            orden{cont_orden,i}=Alternativas{j};
        end
    end
end
Preferible=cell(M);
a_b_fila=cell(1,M);
for i=1:M
    cont_P=0;
    cont_I=0;
    cont_R=0;
    for j=1:M
        a_b_fila{1,j}=a_b{i,j};
    end
    tf_P=strcmp('P',a_b_fila);
    tf_I=strcmp('=',a_b_fila);
    tf_R=strcmp('NC',a_b_fila);
    for j=1:M
        if tf_P(j)==1
            cont_P=cont_P+1;
            Preferible{cont_P,i}=Alternativas{j};
        elseif tf_I(j)==1
            cont_I=cont_I+1;
            Indiferentes{cont_I,i}=Alternativas{j};
        elseif tf_R(j)==1
            cont_R=cont_R+1;
            No_comparables{cont_R,i}=Alternativas{j};
        end
    end
end
disp(Alternativas')
disp(Preferible)

Preferible_columna=cell(M,1);
for i=1:M
    y_n1=isempty(Preferible_columna{1,1});
    if y_n1==0
        fprintf('\n%s preferible a: \n',Alternativas{i})
    end
end

```

```

for j=1:M
y_n2=isempty(Preferible_columna{j,1});
if y_n2==0
fprintf('%s\n',Preferible_columna{j})
end
end
end
end

%Representación Método Promethee I
axes(handles.axes2)
cla reset %resetea el axes2
axes(handles.axes1)
cla reset %resetea el axes1
axes(handles.axes1)
color_alt=zeros(M,3);
for i=1:M
p_x=[-1,1];
p_y=[flujo_sup_pos(i),1-flujo_sup_neg(i)];
y_pos=flujo_sup_pos(i);
y_neg=1-flujo_sup_neg(i);
plot1=plot(p_x,p_y,'-+', 'Color',[rand,rand,rand]); %Le asigna un color
aleatorio a cada alternativa
set(gca,'xtick',[]) %Para quitar los números del eje x
text_ylabel1=text(-1.2,1,'Flujo Positivo');
text_ylabel2=text(1,1,'1 - Flujo Negativo'); %Para poder poner título en
ambos extremos de la gráfica sin emplear dos ejes y
color_alt(i,:)=get(plot1,'Color'); %Para almacenar el color de cada
alternativa
axis([-1 1 0 1])
hold on
end
title('PROMETHEE I Ordenación parcial')
global leyenda_promI;
leyenda_promI=Alternativas;
%Método Promethee II (ordenación completa)
a_b_II=cell(M);
for i=1:M
for j=1:M
if flujo_sup(i)>flujo_sup(j)
a_b_II{i,j}='P';
elseif flujo_sup(i)==flujo_sup(j)
a_b_II{i,j}='I';
end
end
end
disp(a_b_II)
flujo_sup1=flujo_sup; %%Para no modificar el orden inicial de flujo_sup
Alternativas1=Alternativas; %%Para no modificar el orden inicial de
Alternativas

%Para imponer la alternativa final óptima
max_alt=0;
Preferible_columna=cell(M,1);
flujo_supAltfinal=0;
for i=1:M
for j=1:M
sum=0;
Preferible_columna{j,1}=Preferible{j,i};
Columna_esc=isempty(Preferible_columna{j,1});
if Columna_esc==0
sum=sum+1;

```

```

end
if sum>max_alt
    Alternativa_final=Alternativas{i};
    flujo_supAltfinal=flujo_sup(i);
    Fila_alt_fin=i;
    max_alt=sum;
elseif sum==max_alt
    if flujo_sup(i)>flujo_supAltfinal
        Alternativa_final=Alternativas{i};
        flujo_supAltfinal=flujo_sup(i);
        Fila_alt_fin=i;
    end
end
end
end
end

for i=1:M
for j=1:M-i

if flujo_sup1(j)<flujo_sup1(j+1)
aux=flujo_sup1(j);
aux1=Alternativas1(j);
flujo_sup1(j)=flujo_sup1(j+1);
Alternativas1(j)=Alternativas1(j+1);
flujo_sup1(j+1)=aux;
Alternativas1(j+1)=aux1;
end
end
end
for i=1:M
fprintf('\n%s\t',Alternativas1{i})
fprintf('%d', flujo_sup1(i))
end
fprintf('\n')
fprintf('\n')

%Representación Método Promethee II
axes(handles.axes2)
for i=1:M
p_x=[-1,1];
p_y=[flujo_sup(i), flujo_sup(i)];
y_texto=flujo_sup(i);
plot2=plot(p_x,p_y, '-', 'Color',color_alt(i,:));
set(gca, 'xtick', [])
axis([-1 1 -1 1])
hold on
end
title('PROMETHEE II Ordenación completa')
global leyenda_promII;
leyenda_promII=Alternativas;

%Representación perfil de una alternativa
disp(flujo_sup_netto_criterio_alt)

```

Código A1-7: Interfaz gráfica

```

function varargout = InterfazGrafica(varargin)
% INTERFAZGRAFICA MATLAB code for InterfazGrafica.fig

```

```

% INTERFAZGRAFICA, by itself, creates a new INTERFAZGRAFICA or raises
the existing
% singleton*.
%
% H = INTERFAZGRAFICA returns the handle to a new INTERFAZGRAFICA or the
handle to
% the existing singleton*.
%
% INTERFAZGRAFICA('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in INTERFAZGRAFICA.M with the given input
arguments.
%
% INTERFAZGRAFICA('Property','Value',...) creates a new INTERFAZGRAFICA
or raises the
% existing singleton*. Starting from the left, property value pairs are
applied to the GUI before InterfazGrafica_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to InterfazGrafica_OpeningFcn via
varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help InterfazGrafica

% Last Modified by GUIDE v2.5 18-Jun-2018 20:35:27

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @InterfazGrafica_OpeningFcn, ...
                  'gui_OutputFcn', @InterfazGrafica_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before InterfazGrafica is made visible.
function InterfazGrafica_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to InterfazGrafica (see VARARGIN)

% Choose default command line output for InterfazGrafica
handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes InterfazGrafica wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = InterfazGrafica_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%Dejamos al usuario la elección entre pesos por método AHP y pesos metidos
%manualmente
AHP = get(handles.radiobutton1, 'Value');
%Impongo la parte correspondiente al código Promethee
NumAlt=str2num(get(handles.edit10, 'String') );
Codigo_Promethee_modificado;
set(handles.edit11, 'String', Alternativa_final);
Exito_ordenacionPromethee;

%Obtención de los valores para el modelo gpc
Ciudad=get(handles.popupmenu1, 'Value');
global i_PV
if Fila_alt_fin < i_PV
    kWPV=xlsread('Datos gpc.xls', 'Hoja1', strcat('C', num2str(Fila_alt_fin)));
    gen1=Fotovoltaica(kWPV, (Ciudad-1));
else
    kWEol=xlsread('Datos gpc.xls', 'Hoja1', strcat('B', num2str(Fila_alt_fin)));
    gen1=Eolica(kWEol, (Ciudad-1));
end
mult_dom=[125 120 100 125 375 375 1125 1125 1375 1375 800 800 625 500 375 250
265 400 1000 1000 1050 1050 800 800 0]./1375;
kWDom=xlsread('Datos gpc.xls', 'Hoja1', strcat('D', num2str(Fila_alt_fin)));
PCar=[1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0];
dem1=kWDom*mult_dom+PCar*0.2601;
Escritura_gpc(dem1, gen1);

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit1 as text
% str2double(get(hObject, 'String')) returns contents of edit1 as a
double

```



```
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a
double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3 as a
double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%          str2double(get(hObject,'String')) returns contents of edit4 as a
double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%          str2double(get(hObject,'String')) returns contents of edit5 as a
double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

```

```
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%          str2double(get(hObject,'String')) returns contents of edit6 as a
double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%          str2double(get(hObject,'String')) returns contents of edit7 as a
double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%          str2double(get(hObject,'String')) returns contents of edit8 as a
double
```

```

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of edit9 as a
double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns popupmenu2
contents as cell array
%         contents{get(hObject, 'Value')} returns selected item from popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
AnoElegido=get(handles.popupmenu1, 'Value');
CiudadElegida=get(handles.popupmenu2, 'Value');
if AnoElegido==1
    ano=2018;
elseif AnoElegido==2
    ano=2019;
elseif AnoElegido==3
    ano=2020;
elseif AnoElegido==4
    ano=2021;
elseif AnoElegido==5
    ano=2022;
elseif AnoElegido==6
    ano=2023;
elseif AnoElegido==7
    ano=2024;
elseif AnoElegido==8
    ano=2025;
end
if CiudadElegida==1
    Ciudad=0;
elseif CiudadElegida==2
    Ciudad=1;
end
NumAlternativas=Escritura_datos_excel(Ciudad,ano);
set(handles.edit10, 'String', num2str(NumAlternativas));

```

```
function edit10_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%         str2double(get(hObject,'String')) returns contents of edit10 as a
double

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1

function edit11_Callback(hObject, eventdata, handles)
% hObject      handle to edit11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%         str2double(get(hObject,'String')) returns contents of edit11 as a
double

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```

    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu3
global p1 q1 s1
p1(1)=0;
q1(1)=0;
s1(1)=0;

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu4
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu4
global p1 q1 s1
p1(2)=0;
q1(2)=0;
s1(2)=0;

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu5
contents as cell array
% contents{get(hObject,'Value')} returns selected item from popupmenu5
global p1 q1 s1
p1(3)=0;
q1(3)=0;
s1(3)=0;

% --- Executes during object creation, after setting all properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu6.
function popupmenu6_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu6
contents as cell array
% contents{get(hObject,'Value')} returns selected item from popupmenu6
global p1 q1 s1
p1(4)=0;
q1(4)=0;
s1(4)=0;

% --- Executes during object creation, after setting all properties.
function popupmenu6_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.

```



```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu7.
function popupmenu7_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu7
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu7
global p1 q1 s1
p1(5)=0;
q1(5)=0;
s1(5)=0;

% --- Executes during object creation, after setting all properties.
function popupmenu7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
FP=get(handles.popupmenu3,'Value');
global p1 s1 q1
if FP==1
    p1(1)=0;
    q1(1)=0;
    s1(1)=0;
elseif FP==2
    p1(1)=0;
    s1(1)=0;
    Obtencion_qSOC;
    uiwait;
elseif FP==3
    q1(1)=0;
    s1(1)=0;
    Obtencion_pSOC;
    uiwait;
elseif FP==4
    s1(1)=0;
    Obtencion_pqSOC;
    uiwait;
    global p q
    p=p1(1);

```

```

    q=q1(1);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqSOC;
        uiwait;
        p=p1(1);
        q=q1(1);
    end
elseif FP==5
    s1(1)=0;
    Obtencion_pqSOC;
    uiwait;
    global p q
    p=p1(1);
    q=q1(1);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqSOC;
        uiwait;
        p=p1(1);
        q=q1(1);
    end
elseif FP==6
    p1(1)=0;
    q1(1)=0;
    Obtencion_sSOC;
    uiwait;
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
FP=get(handles.popupmenu4, 'Value');
global p1 s1 q1
if FP==1
    p1(2)=0;
    q1(2)=0;
    s1(2)=0;
elseif FP==2
    p1(2)=0;
    s1(2)=0;
    Obtencion_qHidrogeno;
    uiwait;
elseif FP==3
    q1(2)=0;
    s1(2)=0;
    Obtencion_pHidrogeno;
    uiwait;
elseif FP==4
    s1(2)=0;
    Obtencion_pqHidrogeno;
    uiwait;
    global p q
    p=p1(2);
    q=q1(2);
    while q>p
        Error_pq;

```

```

        uiwait;
        Obtencion_pqHidrogeno;
        uiwait;
        p=p1(2);
        q=q1(2);
    end
elseif FP==5
    s1(2)=0;
    Obtencion_pqHidrogeno;
    uiwait;
    global p q
    p=p1(2);
    q=q1(2);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqHidrogeno;
        uiwait;
        p=p1(2);
        q=q1(2);
    end
elseif FP==6
    p1(2)=0;
    q1(2)=0;
    Obtencion_sHidrogeno;
    uiwait;
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
FP=get(handles.popupmenu5, 'Value');
global p1 s1 q1
if FP==1
    p1(3)=0;
    q1(3)=0;
    s1(3)=0;
elseif FP==2
    p1(3)=0;
    s1(3)=0;
    Obtencion_qPerdidas;
    uiwait;
elseif FP==3
    q1(3)=0;
    s1(3)=0;
    Obtencion_pPerdidas;
    uiwait;
elseif FP==4
    s1(3)=0;
    Obtencion_pqPerdidas;
    uiwait;
    global p q
    p=p1(3);
    q=q1(3);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqPerdidas;
        uiwait;
        p=p1(3);

```

```

        q=q1(3);
    end
elseif FP==5
    s1(3)=0;
    Obtencion_pqPerdidas;
    uiwait;
    global p q
    p=p1(3);
    q=q1(3);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqPerdidas;
        uiwait;
        p=p1(3);
        q=q1(3);
    end
elseif FP==6
    p1(3)=0;
    q1(3)=0;
    Obtencion_sPerdidas;
    uiwait;
end

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
FP=get(handles.popupmenu6,'Value');
global p1 s1 q1
if FP==1
    p1(4)=0;
    q1(4)=0;
    s1(4)=0;
elseif FP==2
    p1(4)=0;
    s1(4)=0;
    Obtencion_qCostes;
    uiwait;
elseif FP==3
    q1(4)=0;
    s1(4)=0;
    Obtencion_pCostes;
    uiwait;
elseif FP==4
    s1(4)=0;
    Obtencion_pqCostes;
    uiwait;
    global p q
    p=p1(4);
    q=q1(4);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqCostes;
        uiwait;
        p=p1(4);
        q=q1(4);
    end
elseif FP==5

```

```

s1(4)=0;
Obtencion_pqCostes;
uiwait;
global p q
p=p1(4);
q=q1(4);
while q>p
    Error_pq;
    uiwait;
    Obtencionpq_Costes;
    uiwait;
    p=p1(4);
    q=q1(4);
end
elseif FP==6
    p1(4)=0;
    q1(4)=0;
    Obtencion_sCostes;
    uiwait;
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
FP=get(handles.popupmenu7,'Value');
global p1 s1 q1
if FP==1
    p1(5)=0;
    q1(5)=0;
    s1(5)=0;
elseif FP==2
    p1(5)=0;
    s1(5)=0;
    Obtencion_qImpacto;
    uiwait;
elseif FP==3
    q1(5)=0;
    s1(5)=0;
    Obtencion_pImpacto;
    uiwait;
elseif FP==4
    s1(5)=0;
    Obtencion_pqImpacto;
    uiwait;
    global p q
    p=p1(5);
    q=q1(5);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqImpacto;
        uiwait;
        p=p1(5);
        q=q1(5);
    end
elseif FP==5
    s1(5)=0;
    Obtencion_pqImpacto;
    uiwait;
    global p q

```

```

    p=p1(5);
    q=q1(5);
    while q>p
        Error_pq;
        uiwait;
        Obtencion_pqImpacto;
        uiwait;
        p=p1(5);
        q=q1(5);
    end
elseif FP==6
    p1(5)=0;
    q1(5)=0;
    Obtencion_sImpacto;
    uiwait;
end

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global p1 q1 s1 FP_vector
FP_vector(1)=5;%SOC
q1(1)=0.2;
p1(1)=5;
s1(1)=0;
set(handles.popupmenu3,'Value',5)
FP_vector(2)=5;%Hidrogeno
q1(2)=0.2;
p1(2)=5;
s1(2)=0;
set(handles.popupmenu4,'Value',5)
FP_vector(3)=5;%Perdidas
q1(3)=0.2;
p1(3)=6;
s1(3)=0;
set(handles.popupmenu5,'Value',5)
FP_vector(4)=3;%Coste
q1(4)=0;
p1(4)=5;
s1(4)=0;
set(handles.popupmenu6,'Value',3)
FP_vector(5)=1;%Impacto
q1(5)=0;
p1(5)=0;
s1(5)=0;
set(handles.popupmenu7,'Value',1)

% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
% hObject      handle to togglebutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1
if get(hObject,'Value')==1
    axes(handles.axes1)
    global leyenda_promI
    legend(leyenda_promI,'Location','Best');

```

```

elseif get(hObject,'Value')==0
    axes(handles.axes1)
    legend('off');
end

% --- Executes on button press in togglebutton2.
function togglebutton2_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton2
if get(hObject,'Value')==1
    axes(handles.axes2)
    global leyenda_promII
    legend(leyenda_promII,'Location','Best');
elseif get(hObject,'Value')==0
    axes(handles.axes2)
    legend('off');
end

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Matriz_AHP; %Interfaz con tabla donde aparece la matriz AHP

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
modelo_grid

```

Código A1-8: Preparación de los datos para el control predictivo

```

format long;

TT=24*6;
puntos=TT;

entradas=2; %Ph2 y Pgrid
salidas=2; % SOC y MHL
nacciones=entradas;
ruptura_ref=TT/2;

refy(1,:)=60*ones(1,TT); % salida 1
refy(2,:)=45*ones(1,TT); % salida 2

refy(1,ruptura_ref:110)=55; % salida 1
refy(2,ruptura_ref:110)=55; % salida 2

%Esfuerzo de control en cuadratico

```

```

lambda = [1*ones(entradas,1)];

%Ponderacion del error
beta1=1;
%ponderacion para cada una de las entradas
%despues se copia a beta para multiplicarlo por h (coste lineal)
betaaux=ones(1,entradas);

% Horizontes de control y predicción
N_grid =24*ones(salidas,1);
Nc=24;

% Optimizador
optimizer=0; %'1' cplex '0' miqp

d=zeros(salidas,1);
dm=d;
Nacum=[0; cumsum(N_grid)];
Nyd=N_grid+dm;
Nydacum=[0; cumsum(Nyd)];
for i=1:entradas
    Nu(i)=1;
end
Nuacum=[0; cumsum(Nu)'];

%RESTRICCIONES

restricfuel=0;

restry=1;
ymin=0*ones(1,salidas);
ymin=[40 10];

ymax=5*ones(1,salidas);
ymax=[80 70];

restrU=0; %U real
Umax= [0.9 2];

Umin=[-0.9 -2.5];

restru=0;%Incrementos de U
%umax=[100*ones(1,entradas)];
%umin=[4*ones(1,entradas)];
umax=[0.6 2];
umin=[-0.6 -2.5]

% %ust<=ugt
otrasr=0;
otrasa=ones(1,entradas);
otrasb=0; % hay que actualizarlo con Pnet en cada vuelta

for j=1:entradas

```



```

    for i=Nuacum(j)+1:Nuacum(j+1)
        L(i,i) = lambda(j);
        beta(i,i) = betaaux(j);
    end
end

restrsto=0;
restrchance=0;

%% Modelo en espacio de estados
%X=[SOC MHL]T
%u=[PH2 Pgrid]
% PH2=Pfc+Pe
% Pbat=ph2+Pgrid+pnet
% LA sperturbaciones es la diferencia entre la demanda y lo producido por
% las pv
Bc=[0.2712 0.1986;-0.5096 0];
Pc=[0.1986; 0]; % las perturbaciones solo afectan al SOC

%Tiempo de muestreo,
B=zeros(salidas,entradas);
tm=0.5; %30 segundos
tm=10; % ai igual que la demanda

B=tm*Bc;
P=tm*Pc;

A=[1 0;1 0];
Am=A;

% efecto_ac=zeros(nacciones,1);
% uanteriores=zeros(nacciones,1);
% auxanterior=zeros(nacciones,salidas);
% micoste=zeros(salidas,puntos);
demanda=xlsread('DEM1.xls'); %% demanda en MW desde las 21:00 24h, cada
10minutos. Se escala
demanda=demanda/56; % ahora en w para demanda doméstica
generacion=xlsread('GEN1.xls');
generacion=generacion/2;
perturbacion=generacion-demanda;
figure
plot(demanda,'r');
hold on;
plot(generacion,'g');
plot(perturbacion,'b');
legend('dem','solar','per')
grid on

perturbacion=perturbacion/1000; % paso a kw tal como están expresadas las
entradas
offsetgpc=zeros(salidas,puntos);
offsetgpc(1,:)=perturbacion(1:puntos)*Pc(1);
%offsetgpc(1,:)=10*ones(1,puntos);
s0(1)=50;
s0(2)=50;

Bm=B;

```

```
gpc_grid;
pinta_R;
```

Código A1-9: Estrategia de control predictiva

```
% ASCENSION ZAFRA CABEZA
% Octubre 2017

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Control predictivo generalizado con retraso mínimo diferente por filas%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Este gpc ha sido modificado para que en cada instante sea calculada una
% nueva G, ya que el modelo del proceso puede cambiar en cada instante de
% tiempo.

% Con las dimensiones de la matriz A se obtiene el número de salidas
% y el valor de Na
Na =size(A,2)-1;
Nam =size(Am,2)-1;

% Se calcula el valor para Nb
Nb =size(B,2)/entradas-1;
Nbm =size(Bm,2)/entradas-1;

% Se dimensionan los vectores y matrices necesarios para el control
G =zeros(sum(N_grid),sum(Nu));
yGPC =0*ones(salidas ,puntos);
yGPC(1,1:2)=s0(1);
yGPC(2,1:2)=s0(2);

ysinr=0*ones(salidas,puntos);

%ysinm=0*ones(salidas,puntos);
uGPC =zeros(entradas,puntos); %Incremento de U
ureal=0*ones(entradas,puntos); %u real
%ref=ones(salidas,puntos);
%opttime=zeros(1,puntos);

%Construcción de la matriz deltaA
z=zeros(salidas,1);
Avir=[A z]-[z A];
Amvir=[Am z]-[z Am];
wreal=0; %referencia real

%offsetcostemain=zeros(1,puntos);
%offsetcostemainacum=zeros(1,puntos+1);

%offsetcosteext=zeros(1,puntos);
%offsetcosteextacum=zeros(1,puntos+1);

%offsety=zeros(1,salidas);

% Comienzo del bucle del control GPC
```

```

for bucle = (max(max(Na,Nb),max(Nam,Nbm)))+1:puntos,

    bucle%B=[];
    %construye_modelo_bucle_IC_coste; % si el modelo cambiase
    Bm=B;

    % en offsetcoste tenemos el offset del bucle

    %Amvir=Avir;
    % Construcción de la matriz G.
    % la dimension de G es N*salidas,entradas
    % Se calcula como la respuesta a un impulso en incremento de u
    for j=1 : salidas,
        for k=1 : entradas,
            G(Nacum(j)+1:Nacum(j+1),Nuacum(k)+1:Nuacum(k+1)) =...
                bloqueq(Bm(j,(k-
1)*(Nbm+1)+1+dm(j):k*(Nbm+1)),Amvir(j,:),N_grid(j),Nam,Nu(k));
        end
    end
%
end

H=2*(beta1*(G'*G) + L);

% J=(Gu+f-w)'(Gu+f-w)+lambda*u'u
% J=0.5u'Hu+b'u+f0

% f0=(f-w)'(f-w)
% b'=2(f-w)'G factor lineal
% u=-H(-1)b
% H=2(G'G+lambdaI) %factor cuadratico
% H=(beta1*(Gaux'*Gaux) + L);

%calculo de la matriz de restricciones
%añado restriccion de coste fuel <x

cal_a2;

% Construcción del vector f
% En primer lugar se simula desde el instante t hasta
% el instante t+N(salida)+d(salida)

f=zeros(sum(N_grid),1);
facum= zeros(sum(Nyd),1); %facum=[ysim(t) ysim(t+1) ... ysim(t+d)
ysim(t+d+1) ... ysim(t+d+N)]

for j=1 : salidas,
    yaux=flipud(yGPC(j,bucle-Nam:bucle)'); %yaux=[y(t-1) y(t-2) ... y(t-
na+1)]
    IncU=fliplr(uGPC(:,bucle-Nbm:bucle)); %IncU=[0 u(t-1) ... u(t-nb)]

    for l=1 : Nyd(j),
        aux=0;
        for k=1 : entradas,

```

```

        aux = aux + Bm(j, (k-1) * (Nbm+1) + 1:k * (Nbm+1)) * IncU(k, :)';
    end

    facum(Nydacum(j)+1) = facum(Nydacum(j)+1) - Amvir(j, 2:Na+2) * yaux +
aux;
    yaux = [facum(Nydacum(j)+1); yaux(1:Nam)];
    IncU = [zeros(entradas, 1), IncU(:, 1:Nbm)];

    end
end

% Se toma sólo la parte correspondiente a los horizontes de control
% es decir se elimina la parte correspondiente al retraso

for j=1 : salidas,
    f(Nacum(j)+1:Nacum(j+1)) = ...

facum(Nydacum(j)+dm(j)+1:Nydacum(j)+dm(j)+N_grid(j)); %offsetgpc(j,bucle);
%modificacion
    end
    %Fin del cálculo de f
    f(1:N_grid(1)) = f(1:N_grid(1)) + offsetgpc(1,bucle);
    %f(1:N(1)) = f(1:N(1)) + 0;
    %f(N(1)+1:N(1)+N(2)) = f(N(1)+1:N(1)+N(2)) + 0;
    if bucle == 2
        f(1:N_grid(1)) = f(1:N_grid(1)) + s0(1);

f(N_grid(1)+1:N_grid(1)+N_grid(2)) = f(N_grid(1)+1:N_grid(1)+N_grid(2)) + s0(2);
    end
    w = [];
    for iw=1:salidas

        w = [w; refy(iw,bucle) * ones(N_grid(iw), 1)];

end

%calculo de la matriz de restricciones b
cal_b2;

h = 2 * G' * (f - w);
%para añadir ponderaciones formamos el vector: beta
%añado termino del cfuel, que depende de Pfuel(t)
betapi = beta;
%    betapi(1) = beta(1) + beta2 * Pfuel(bucle);
h = betapi * h;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
% PARAMETROS DE OPTIMIZACION MIXTA

u0 = H \ (G' * (w - f));

sense = 1; %1 para minimizar, -1 max

```

```

    if size(Ar,1)==0 %Ar<=br , el <= lo da ctype
        Ar=ones(1,entradas*Nu(1));
        br=10000000000;
    else
%       Ar=zeros(1,entradas*Nu(1));
%       Ar(6)=1;
%       br=1000;
        end
nr = size(Ar,1);
nvar = size(Ar,2);
ctype = [];
for i=1:nr,
    ctype=[ctype ; 'L'];
end

%tipo de variables C=real y B=booleana
vartype = [];
for i=1:entradas,
    vartype=[vartype ; 'C'];
end

opt = optimset('Display','off','LargeScale','off');
ub=[0.9; 2];
%lb(1:5)=[50 800 20 0.20 36500];
% lb(2)=0;
% lb(1)=0;
% % lb(3)=20;
% lb(4)=0.2;

lb=[ -0.9; -2.5];

%ub(1:2)=[100 100];
% ub(2)=37;
% if restriceje2==1
%     ub(1)=8;
% end
%ub(4)==20
for i=1:entradas,
    for j=1:Nu(i),
        ivar(Nu(i)*(i-1)+j)=vartype(i);
    end
end
% ivar=[ivar 'C'];
options = [];
options.solver = 'quadprog';

if optimizer==0
    [xopt,fsol,flag,efic]=miqp(H,h,Ar,br,Aeq, Beq,vartype,lb,ub);
    % xopt=qp(H,h,Ar,br,Aeq, Beq,lb,ub)
    % xopt=quadprog(H,h,Ar,br,Aeq,Beq,lb,ub);
else
    %buena cplex
    bucle
end

[xopt,obj,status,extra]=cplexmex(sense,H,h,Ar,br,ctype,lb,ub,vartype,u0);

```

```

end
    % end

aux=xopt;
%aux=x;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

uGPC(:,bucle)=aux(1:Nu:entradas*Nu); %incrementos
ureal(:,bucle)=ureal(:,bucle-1)+ uGPC(:,bucle);%ureal

% Simulación del sistema con Incrementos de u
for j=1 : salidas,
    yaux=flipud(yGPC(j,bucle-Na:bucle)');
    aux=0;
    IncUp=fliplr(uGPC(:,bucle-Nb:bucle));
    for k=1 : entradas,
        aux = aux + B(j, (k-1)*(Nb+1)+1:k*(Nb+1))*IncUp(k,:)';
    end

    yGPC(j,bucle+1) = A(j,:)*yaux + aux+offsetgpc(j,bucle);
    %yGPC(j,bucle+1) = A(j,:)*yaux + aux;

end

end %end del bucle

```


REFERENCIAS

- [1] *Diario de Renovables*, <http://www.diariorenovables.com>
- [2] IRENA (2016), *The Power to Change: Solar and Wind Cost Reduction Potential to 2025*
- [3] BORDONS, C., GARCÍA-TORRES, F. y VALVERDE, L. «*Gestión Óptima de la Energía en Microrredes con Generación Renovable*», *Revista Iberoamericana de Automatica e Informatica Industrial*, publicación 12 (2015) p. 117-132.
- [4] José Podadera Borrego, «*Análisis de Fallos, Simulación y Optimización en Control Predictivo sobre planta real de Generación Distribuida*», *Trabajo Fin de Grado*, Grado en Ingeniería de Tecnologías Industriales, Tutora Ascensión Zafra Cabeza, Escuela Técnica Superior de Ingeniería, Sevilla, 2017.
- [5] *Instituto para la Diversificación y Ahorro de la Energía*, <http://www.idae.es>
- [6] Juan Pablo Claro Báez, «*Modelado y Simulación de Smart Grid con OpenDSS y Matlab*», *Trabajo Fin de Grado*, Grado en Ingeniería en Tecnologías Industriales, Tutora Ascensión Zafra Cabeza, Escuela Técnica Superior de Ingeniería, Sevilla, 2016.
- [7] *Agencia Andaluza de la Energía*, <http://www.agenciaandaluzadelaenergia.es>
- [8] DUGAN, R., MONTENEGRO, D. «*The Open Distribution System Simulator (OpenDSS)*», Electric Power Research Institute, March 2018
- [9] María Siles Luna, «*Implementación de Métodos de Toma de Decisiones Multicriterio usando Matlab*», *Trabajo Fin de Grado*, Grado de Ingeniería de Tecnologías Industriales, Tutor Sebastián Lozano Segura, Escuela Técnica Superior de Ingeniería, Sevilla, 2016.