



**Universidad**  
Zaragoza



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza

# Proyecto Fin de Carrera

Ingeniería Técnica Industrial: Electrónica

## Control distribuido de redes inteligentes mediante sistemas multiagente

Autor

Rafael Tallada Souto

Director

José Antonio Domínguez Navarro

Escuela de Ingeniería y Arquitectura – Universidad de Zaragoza

Septiembre 2013



# Agradecimientos

Quisiera agradecer a todas las personas que de una u otra forma me han acompañado durante estos cuatro años de estudios universitarios su cariño y su apoyo:

A mis compañeros de clase, por los buenos momentos que hemos pasado juntos y por haber tenido la paciencia suficiente para aguantarme.

A mi familia, por haberme dado la oportunidad de continuar formándome.

A Hans Bludszuweit, Logan López y Aitor Alcrudo; por su inestimable ayuda en los momentos más críticos del proyecto.

Y por supuesto a Ana, porque gracias a ella los años más duros de mi vida han resultado ser también los mejores.



## Resumen

La sociedad actual es testigo de una paulatina transición en la forma de generar y distribuir la energía eléctrica. El sistema tradicional cuya generación de electricidad se concentra en grandes instalaciones centralizadas está dando paso a otro en el que la generación se integra de forma distribuida en la red eléctrica.

Uno de los principales motores de este cambio es la progresiva implantación de las energías renovables. Las ventajas de este tipo de energías son de sobra conocidas: se trata de energías respetuosas con el medio ambiente y virtualmente inagotables. Sin embargo, su integración en la red eléctrica entraña nuevos desafíos de control.

Para afrontar estos desafíos, las redes eléctricas deberán incorporar elementos de monitorización, comunicación y control; que consigan un balance óptimo entre generación, demanda y almacenamiento energético y, además, reaccionen con rapidez ante los cambios. Nacen así los conceptos de “Red Inteligente” y de “Smart City”.

Este PFC ha sido realizado gracias a una beca de iniciación a la investigación de la fundación CIRCE y aborda el control distribuido de una microrred inteligente por medio de un sistema multiagente. Para ello se ha implementado un emulador mediante la tecnología de agentes JADE (desarrollada en Java) que reproduce los flujos de potencia en la microrred, y se han ideado diferentes estrategias para su control.

La validación del emulador se ha llevado a cabo con numerosos ensayos y simulaciones, obteniéndose resultados de gran utilidad a la hora de dimensionar de forma óptima los sistemas de almacenamiento eléctrico con los que cuenta la microrred.

Finalmente, se ha estudiado la posibilidad de interconectar la tecnología de agentes con el entorno de simulación MATLAB-Simulink. En concreto, se ha diseñado un sistema multiagente que optimiza el despacho económico de potencia de una microrred, modelada de forma gráfica en Simulink.



# Índice

Índice .....	1
Índice de figuras .....	5
Índice de tablas .....	7
Lista de abreviaturas .....	9
<b>1 Introducción .....</b>	<b>11</b>
1.1 Contexto .....	11
1.2 Objetivos y alcance del proyecto .....	13
1.3 Ámbito y motivación .....	13
1.4 Organización del documento .....	14
<b>2 Conceptos previos .....</b>	<b>15</b>
2.1 Microrredes .....	15
2.2 Técnicas heurísticas y optimización clásica .....	17
2.3 Agentes software y sistemas multiagente .....	18
2.3.1 Qué es un agente .....	18
2.3.2 Sistemas multiagente .....	21
2.3.3 Comunicación entre agentes .....	21
2.4 Framework JADE .....	22
2.4.1 Containers y Plataformas .....	22
2.4.2 Los agentes AMS, DF y RMA .....	23
2.4.3 Agentes en JADE: la clase Agent .....	26
2.4.4 Comportamientos .....	27
2.5 Antecedentes .....	29
<b>3 Optimización mediante técnicas heurísticas .....</b>	<b>31</b>
3.1 Propuesta de trabajo .....	31
3.2 Descripción del área sujeto de estudio .....	32
3.2.1 Recurso renovable .....	33
3.2.2 Demanda y balance de potencia en PCC .....	35
3.3 Descripción del modelo .....	37
3.3.1 Datos de entrada y de salida .....	40
3.3.1.1 Inputs .....	40
3.3.1.2 Outputs .....	41
3.3.2 Casos de estudio .....	43
3.3.3 Estrategias de control de P (potencia activa) .....	43
3.3.3.1 Peak Shaving .....	44
3.3.3.2 Smoothing (alisamiento de la curva de demanda) .....	48
3.3.4 Control de Q (potencia reactiva) .....	51
3.4 Implementación informática .....	54
3.4.1 Relación de clases .....	55
3.4.2 Protocolo de interacción FIPA Contract Net .....	56

3.4.2.1	Funcionamiento .....	56
3.4.2.2	Clases ContractNetInitiator y ContractNetResponder .....	58
3.4.3	Agentes.....	59
3.4.3.1	Agente Power Manager .....	59
3.4.3.2	Agente DG .....	61
3.4.3.3	Agente Storage .....	62
3.4.3.4	Agente Load .....	63
3.4.4	Elementos físicos.....	63
3.4.4.1	Modelo del convertidor. Clase Converter.....	63
3.4.4.2	Modelo de la batería. Clase Battery.....	64
3.4.5	Otras clases.....	66
3.4.6	Modo monitorización.....	67
3.5	Resultados .....	70
3.5.1	Microrred sin DG.....	70
3.5.1.1	Control de P .....	70
3.5.1.2	Control de Q .....	72
3.5.1.3	Actuación de las baterías.....	72
3.5.2	Microrred con DG.....	74
3.5.2.1	Control de P .....	74
3.5.2.2	Control de Q .....	78
3.5.2.3	Actuación de las baterías.....	80
3.6	Conclusiones y líneas futuras.....	84
3.6.1	Conclusiones.....	84
3.6.1.1	Emulador .....	84
3.6.1.2	Modelo simulado .....	84
3.6.2	Líneas futuras .....	85
3.6.2.1	Emulador .....	85
3.6.2.2	Modelo simulado .....	88
<b>4</b>	<b>Optimización clásica .....</b>	<b>89</b>
4.1	Caracterización del modelo .....	89
4.1.1	Reguladores Inteligentes .....	90
4.1.2	Agente Optimizador .....	91
4.1.2.1	Bloque de previsión de demanda .....	91
4.1.3	Otros elementos de la microrred.....	92
4.1.3.1	Generador eólico (recurso renovable) .....	92
4.1.3.2	Grupo electrógeno diésel (recurso no renovable).....	92
4.1.3.3	Sistemas de almacenamiento.....	92
4.1.3.4	Cargas .....	93
4.1.3.5	Línea de transmisión.....	93
4.1.3.6	Red .....	93
4.2	Algoritmo de optimización para la gestión de la energía.....	94
4.3	Implementación informática.....	96
4.3.1	Software utilizado .....	96
4.3.1.1	MATLAB-Simulink.....	96
4.3.1.2	MACSimJX .....	97
4.3.1.3	A Modeling Language for Mathematical Programming (AMPL).....	99
4.3.2	Esquema conceptual del modelo de simulación.....	99
4.4	Casos de estudio y resultados.....	101



4.4.1	Resultados de una optimización para cuatro periodos mediante AMPL.....	101
4.4.2	Verificación del seguimiento de las consignas.....	103
4.4.3	Respuesta a un cambio brusco de carga .....	105
4.4.4	Funcionamiento en isla .....	108
4.5	Conclusiones y líneas futuras.....	110
4.5.1	Conclusiones.....	110
4.5.2	Líneas futuras .....	111
<b>5</b>	<b>Conclusiones personales.....</b>	<b>113</b>
<b>6</b>	<b>Anexos.....</b>	<b>115</b>
6.1	Feedback modo <i>Debug</i> (Iteration 1 of 17520, <i>Peak Shaving</i> ) .....	115
<b>7</b>	<b>Bibliografía.....</b>	<b>125</b>



## Índice de figuras

Fig. 1.	Ejemplo de Smart Grid [2].....	11
Fig. 2.	Evolución del consumo y de la oferta de energía (1949 = 1) .....	12
Fig. 3.	Logo de la fundación CIRCE.....	13
Fig. 4.	Optimización de la ecuación (1) .....	18
Fig. 5.	Agente como entidad autónoma que interacciona con su entorno.....	19
Fig. 6.	SMA distribuido entre varios ordenadores conectados en red [12] .....	23
Fig. 7.	Funcionamiento del servicio de páginas amarillas del agente DF [12].....	24
Fig. 8.	Creación de agentes y notificaciones a AMS y DF.....	25
Fig. 9.	Interfaz gráfica del agente RMA .....	25
Fig. 10.	Ciclo de vida de un agente [12].....	27
Fig. 11.	Logos de OptimaGrid, FEDER y SUDOE.....	31
Fig. 12.	Mapa del sudoeste europeo, SUDOE .....	33
Fig. 13.	Recurso eólico anual .....	34
Fig. 14.	Recurso fotovoltaico anual .....	34
Fig. 15.	Recurso renovable (eólico + fotovoltaico).....	35
Fig. 16.	Demanda de potencia activa anual.....	36
Fig. 17.	$P_{cc}$ anual, con generación eólica y fotovoltaica .....	37
Fig. 18.	Esquema de la microrred emulada .....	38
Fig. 19.	Diagrama de funcionamiento del modelo.....	39
Fig. 20.	Ontología de control P y Q .....	43
Fig. 21.	Rutina de carga automática.....	46
Fig. 22.	Peak Shaving: 8ª semana. ....	47
Fig. 23.	Evolución del SOC de las baterías: Peak Shaving.....	47
Fig. 24.	Smoothing: 8ª semana ( $\omega=4$ , 1MWh) .....	48
Fig. 25.	Evolución del SOC de las baterías: Smoothing ( $\omega=4$ , 1MWh) .....	49
Fig. 26.	Smoothing: 8ª semana ( $\omega=8$ , 2MWh) .....	50
Fig. 27.	Evolución del SOC de las baterías: Smoothing ( $\omega=8$ , 2MWh) .....	50
Fig. 28.	Oferta de Q por parte de los agentes DG (obtención de $Q_{max}$ ) .....	52
Fig. 29.	Mecánica del control de Q .....	53
Fig. 30.	SMA de la microrred implementada.....	54
Fig. 31.	Tipos de agente según su rol.....	55
Fig. 32.	Diagrama de protocolo: FIPA-ContractNet .....	57
Fig. 33.	Modelo del convertidor.....	64
Fig. 34.	Comprobación del envío datos a través del puerto COM8 .....	69
Fig. 35.	Distribución de frecuencias de $P_{cc}$ : no DG (kW) .....	71
Fig. 36.	Distribución acumulada de $P_{cc}$ : no DG (kW) .....	71
Fig. 37.	Distribución acumulada del SOC (%) de las baterías: no DG .....	73
Fig. 38.	Distribución de frecuencias de $P_{cc}$ : DG eólica (kW).....	75
Fig. 39.	Distribución acumulada de $P_{cc}$ : DG eólica (kW).....	75
Fig. 40.	Distribución de frecuencias de $P_{cc}$ : DG fotovoltaica (kW).....	76
Fig. 41.	Distribución acumulada de $P_{cc}$ : DG fotovoltaica (kW).....	76
Fig. 42.	Distribución de frecuencias de $P_{cc}$ : DG eólica + fotovoltaica (kW).....	77
Fig. 43.	Distribución acumulada de $P_{cc}$ : DG eólica + fotovoltaica (kW).....	77
Fig. 44.	Distribución acumulada del f.d.p: DG eólica.....	79
Fig. 45.	Distribución acumulada del f.d.p: DG fotovoltaica .....	79
Fig. 46.	Distribución acumulada del f.d.p: DG eólica + fotovoltaica.....	80
Fig. 47.	Distribución acumulada del SOC (%): DG eólica.....	82
Fig. 48.	Distribución acumulada del SOC (%): DG fotovoltaica .....	82

Fig. 49.	Distribución acumulada del SOC (%): DG eólica + fotovoltaica.....	82
Fig. 50.	Distribución acumulada del SOC (%): Peak Shaving.....	83
Fig. 51.	Distribución acumulada del SOC (%): Smoothing.....	83
Fig. 52.	Servidores adquiridos por CIRCE para el emulador .....	86
Fig. 53.	Emulador de la microrred: Power Hardware in the Loop (PHIL).....	87
Fig. 54.	Jerarquía del control.....	89
Fig. 55.	Droops de frecuencia y voltaje [25] .....	90
Fig. 56.	Esquema de conexión a la microrred.....	90
Fig. 57.	Panorámica del modelo Simulink utilizado.....	97
Fig. 58.	Estructura de MACsim .....	98
Fig. 59.	Canales de comunicación establecidos por MACSimJX [27] .....	98
Fig. 60.	SMA en el que se ejecuta el agente Optimizador.....	99
Fig. 61.	Esquema conceptual del modelo de simulación.....	100
Fig. 62.	Seguimiento de consignas de P: transición del periodo 1 al periodo 2 .....	104
Fig. 63.	Seguimiento de consignas de Q: transición del periodo 1 al periodo 2 .....	105
Fig. 64.	Respuesta a cambio brusco en la carga durante el periodo 3: P .....	106
Fig. 65.	Respuesta a cambio brusco de la carga óhmica durante el periodo 3: Q.....	107
Fig. 66.	Respuesta a cambio brusco de la carga inductiva durante el periodo 3: Q.....	108
Fig. 67.	Funcionamiento en isla: P .....	109

## Índice de tablas

Tabla 1. Actos comunicativos FIPA-ACL.....	22
Tabla 2. Parámetros estadísticos de las series temporales de P renovable (kW) .....	35
Tabla 3. Archivo de inputs: generador .....	41
Tabla 4. Archivo de inputs: carga .....	41
Tabla 5. Archivo de outputs: control .....	41
Tabla 6. Archivo de outputs: generador.....	42
Tabla 7. Archivo de outputs: batería .....	42
Tabla 8. Parámetros de control: Peak Shaving.....	44
Tabla 9. Parámetros de carga/descarga automática: Peak Shaving.....	45
Tabla 10. Principales clases Java del SMA.....	56
Tabla 11. Campos comunes a todos los mensajes ACL del proceso de subasta.....	59
Tabla 12. Vectores utilizados por MailBoxPQControl.....	60
Tabla 13. Mensaje ACL de <i>ACCEPT-PROPOSAL</i> enviado por el agente PowerManager .....	60
Tabla 14. Mensaje ACL de <i>PROPOSE</i> enviado por el agente DG.....	61
Tabla 15. Mensaje ACL de <i>INFORM</i> enviado por el agente DG .....	61
Tabla 16. Mensaje ACL de <i>PROPOSE</i> enviado por el agente Storage.....	62
Tabla 17. Mensaje ACL de <i>PROPOSE</i> enviado por el agente Load .....	63
Tabla 18. Clases secundarias.....	66
Tabla 19. Parámetros que definen la comunicación serie.....	67
Tabla 20. Identificadores de dato: comunicación serie.....	68
Tabla 21. Valores enviados a través del puerto serie.....	69
Tabla 22. Resumen y codificación de las opciones de simulación.....	70
Tabla 23. Parámetros estadísticos de las estrategias de P: no DG (kW) .....	70
Tabla 24. Reducción de picos con Peak Shaving: no DG .....	72
Tabla 25. Consumo de potencia reactiva (kVAR) .....	72
Tabla 26. Parámetros estadísticos del SOC (%) de las baterías: no DG.....	72
Tabla 27. Parámetros estadísticos de las estrategias de P: DG eólica (kW) .....	74
Tabla 28. Parámetros estadísticos de las estrategias de P: DG FV (kW) .....	75
Tabla 29. Parámetros estadísticos de las estrategias de P: DG eólica + FV (kW).....	76
Tabla 30. Reducción de picos con Peak Shaving: microrred con DG (kW) .....	78
Tabla 31. Parámetros estadísticos del control de Q (f.d.p.): Eólica / FV .....	78
Tabla 32. Parámetros estadísticos del control de Q (f.d.p.): Eólica + FV.....	79
Tabla 33. Parámetros estadísticos del SOC (%) de las baterías: microrred con DG.....	81
Tabla 34. Principios de funcionamiento: VSC conectado a red .....	91
Tabla 35. Parámetros de las baterías.....	93
Tabla 36. Datos de entrada del algoritmo de optimización .....	94
Tabla 37. Variables a optimizar .....	95
Tabla 38. Datos de entrada para el algoritmo de optimización.....	101
Tabla 39. Consignas optimizadas de P y Q.....	102
Tabla 40. Consignas de P para los periodos 1 y 2 .....	103
Tabla 41. Consignas de Q para los periodos 1 y 2.....	104
Tabla 42. Consignas de P para el periodo 3.....	105
Tabla 43. Consignas de Q para el periodo 3 .....	106
Tabla 44. Consignas de potencia activa .....	109



## Lista de abreviaturas

**AC:** Alternating Current  
**ACL:** Agent Communication Language  
**AE:** Agent Environment  
**AID:** Agent IDentifier  
**AMPL:** A Mathematical Programming Language  
**AMS:** Agent Management System  
**ASCII:** American Standard Code for Information Interchange  
**ATF:** Agent Task Force  
**CFP:** Call For Proposal  
**CSV:** Comma-Separated Values  
**DAI:** Distributed Artificial Intelligence  
**DC:** Direct Current  
**DF:** Directory Facilitator  
**DG:** Distributed Generation  
**DSP:** Digital Signal Processing  
**fdp:** factor de potencia  
**FEDER:** Fondo Europeo de Desarrollo REgional  
**FIPA:** Foundation for Intelligent Physical Agent  
**FSM:** Finite State Machine  
**FV:** FotoVoltaica  
**JADE:** Java Agent DEvelopment framework  
**PCC:** Point of Common Coupling/Connection  
**PI:** Protocolo de Interacción  
**PHIL:** Power Hardware In the Loop  
**PSH:** Pumped-Storage Hydroelectricity  
**PV:** PhotoVoltaics  
**RMA:** Remote Agent Management  
**SAE:** Sistema de Almacenamiento Eléctrico  
**SMA:** Sistema MultiAgente  
**SOC:** State Of Charge  
**SUDOE:** SUDOeste Europeo  
**TIC:** Tecnologías de la Información y la Comunicación  
**VSC:** Voltage-Source Converter





# 1 Introducción

A continuación se presenta la memoria del Proyecto Fin de Carrera (PFC) “Control distribuido de redes inteligentes mediante sistemas multiagente”, que ha tenido como principal objetivo la implementación de un emulador de microrredes basado en tecnología de agentes con el que realizar simulaciones para extraer información de interés.

En esta memoria se va a exponer el trabajo realizado, los resultados obtenidos en las simulaciones y las conclusiones finales.

## 1.1 Contexto

Resulta innegable que la sociedad actual es testigo de una paulatina transición en la forma de generar y distribuir la energía eléctrica. El sistema tradicional cuya generación de electricidad se concentra en grandes instalaciones centralizadas está dando paso a otro en el que la generación se integra de forma distribuida en la red eléctrica [1]. La causa del cambio está íntimamente relacionada con el auge imparable de las energías renovables.

Las ventajas que este auge supone son bien conocidas: se trata de energías respetuosas con el medio ambiente y virtualmente inagotables. Sin embargo, su integración en la red actual entraña nuevos desafíos. Uno de los principales retos proviene de su naturaleza intrínsecamente aleatoria, que dificulta lograr un equilibrio en tiempo real entre generación y demanda. El uso de sistemas de almacenamiento eléctrico (SAE) ayuda en esta tarea, pero no es suficiente. Para salvar este obstáculo, las redes eléctricas actuales deberán incorporar elementos de monitorización, comunicación y control; que consigan un balance óptimo entre generación, demanda y almacenamiento y, además, reaccionen con rapidez ante los cambios.

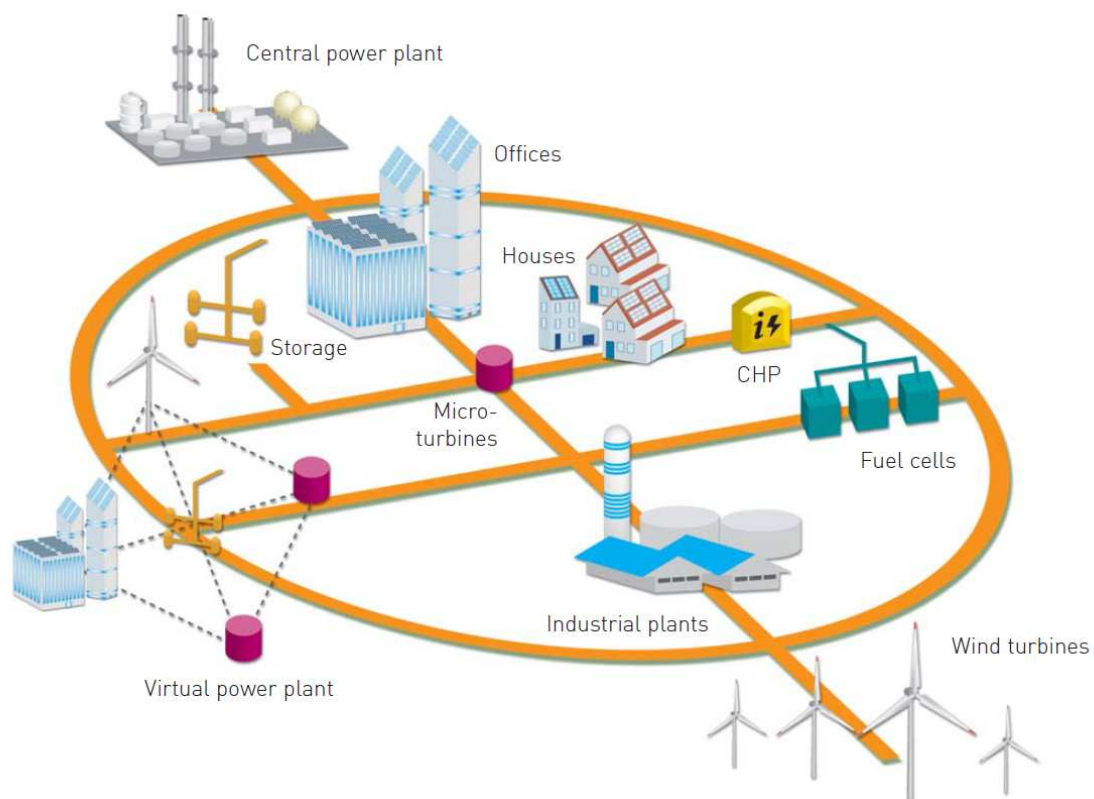


Fig. 1. Ejemplo de Smart Grid [2]

Nace así el concepto de “Smart Grid” o red inteligente [3]. La definición de Smart Grid varía según el autor. A un primer nivel, se podría definir como la forma de proveer energía, desde la generación a la demanda, que se vale de las tecnologías TIC. Un sistema de esta naturaleza tiene como objetivo prioritario asegurar el suministro. A un segundo nivel, la comunicación establecida entre los distintos elementos de la Smart Grid trae consigo ahorro energético, reducción de costes y una optimización de la generación proveniente de recursos no renovables.

Además de la problemática que presenta la ausencia de control sobre el recurso natural, la red eléctrica deberá ser capaz de hacer frente al aumento sostenido de la demanda de energía. Aun teniendo en cuenta el aumento de la eficiencia energética que se presupone en el ámbito de la edificación, el transporte, etc.; se estima que el consumo actual de energía eléctrica va a verse multiplicado por cuatro en 2040.

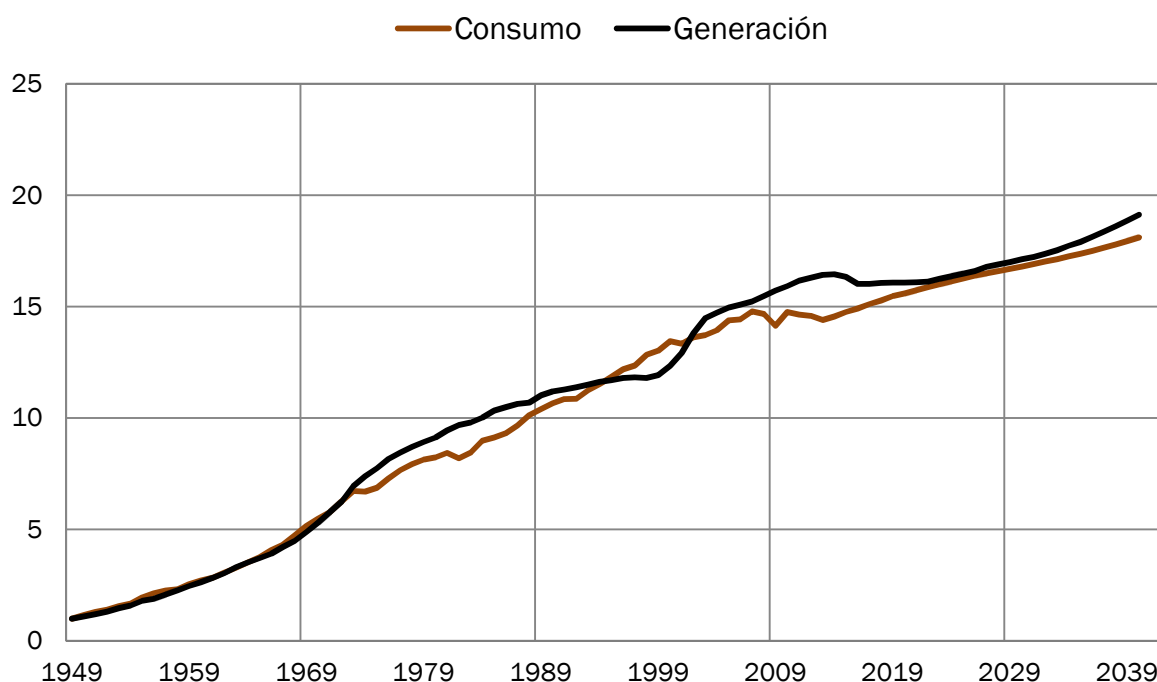


Fig. 2. Evolución del consumo y de la oferta de energía (1949 = 1)

Por si esto no fuera suficiente, se prevén incrementos considerables en el consumo eléctrico de carácter puntual debido a la progresiva implantación que va a experimentar el vehículo eléctrico. Estos picos de demanda obligarán al desarrollo de SAE inteligentes que respondan de forma flexible ante esta complejidad creciente.

Se plantea también la posibilidad de modificar la curva de demanda actuando directamente sobre el comportamiento del consumidor, incentivando el uso de la energía en periodos con menor demanda global y “penalizándolo” durante dichos picos. Existen modalidades de contrato en algunos países como EE.UU. en los que el consumidor otorga cierto control a la empresa suministradora sobre algún aspecto concreto de su consumo, como el sistema de refrigeración, a fin de que la compañía eléctrica pueda modificar el patrón de demanda en caso de ser necesario. El consumidor recibe como contrapartida descuentos en la factura o tarifas más ventajosas.

## 1.2 Objetivos y alcance del proyecto

El principal objetivo de este proyecto ha consistido en implementar un emulador de microrredes basado en el “paradigma de agentes”. Este paradigma se centra en el desarrollo de entidades software que pueden actuar de forma autónoma y razonada. Para ello se ha realizado un sistema multiagente (SMA) en el que cada elemento de la microrred cuenta con un agente asociado encargado de su gestión.

La efectividad del SMA a la hora de replicar el funcionamiento real de una microrred ha sido evaluada por medio de distintas simulaciones, bajo distintos supuestos de funcionamiento, y con diferentes estrategias de control. La arquitectura y los protocolos utilizados han sido escogidos cuidadosamente para dejar la puerta abierta a futuros desarrollos y mejoras funcionales.

El lenguaje de programación elegido ha sido Java debido a dos razones. La primera es que se trata de un lenguaje independiente de la plataforma, lo que ofrece una gran libertad a la hora de ejecutar la SMA en ordenadores con distintos sistemas operativos. La segunda razón es la existencia del *framework* JADE, basado por completo en lenguaje Java, que facilita sobremedida la implementación de sistemas multiagente, evitando que el programador tenga que preocuparse de desarrollar las funcionalidades de más bajo nivel.

Por último, se ha estudiado la posibilidad de emplear un modelo ya existente implementado en el entorno de simulación Simulink para reproducir los flujos de potencia en una microrred. El control de dicha microrred se lleva a cabo de forma externa por agentes en JADE. La comunicación entre ambas plataformas se ha realizado por medio del *middleware* MACSimJX, mientras que para optimizar el despacho de potencia se ha empleado el lenguaje de modelado algebraico AMPL.

## 1.3 Ámbito y motivación

El presente documento recoge parte del trabajo llevado a cabo para CIRCE durante el curso 2012/2013, en el marco de una beca de iniciación a la investigación.

La principal motivación para realizar este PFC fue la posibilidad de integrar en un mismo proyecto programación, cuestiones de control y energías renovables; temas por los que el autor ha ido desarrollando un gran interés a lo largo de su etapa como estudiante de ing. técnica electrónica.



Fig. 3. Logo de la fundación CIRCE

El PFC se enmarca dentro de una de las líneas de investigación abiertas en el área de I+D+i de Integración de Energías Renovables (IER), de la fundación CIRCE, consistente en el diseño óptimo de sistemas integrados de generación y almacenamiento (bombeo hidráulico reversible, baterías, condensadores y supercondensadores), incluyendo la integración en microrredes, tanto conectadas a red como aisladas.

Se abarca también la gestión óptima de flujos energéticos de fuentes renovables como mini y micro hidroeléctrica, eólica, biomasa y fotovoltaica, así como el diseño de controles que aseguren la estabilidad del sistema y minimicen el impacto en red.

## 1.4 Organización del documento

El presente documento consta de memoria y anexos. La memoria consta de cinco capítulos en los que se ofrece un completo resumen trabajo realizado, así como las pertinentes justificaciones, resultados y conclusiones. La información recogida en la memoria es ampliada en los anexos.

El primer capítulo de la memoria, en el que está incluido este apartado, es de carácter introductorio; y sitúa al PFC en su contexto, presenta los objetivos a cumplir y el ámbito y motivación para su realización.

El capítulo 2 expone algunos conceptos previos necesarios para la comprensión del documento. Este capítulo explicará en qué consiste una microrred y algunas de sus opciones de control, describirá las diferencias entre optimización clásica y optimización mediante técnicas heurísticas, desarrollará el concepto de SMA, realizará una pequeña introducción al *framework* de agentes JADE y terminará con algunos antecedentes.

El contenido del capítulo 3 abarca el grueso del proyecto: el emulador de microrredes mediante SMA, correspondiente a la parte de optimización heurística. El capítulo incluye explicaciones sobre su funcionamiento y sobre el modelo de microrred implementado, realiza un análisis de los datos que se utilizarán para su validación, presenta los resultados obtenidos y termina con las conclusiones y líneas de continuidad.

El capítulo 4 enfoca el control desde el punto de vista de la optimización clásica, y presenta los resultados obtenidos al integrar la tecnología de agentes con software de simulación gráfica y optimización lineal. Al igual que en el capítulo 3, se va a proporcionar información acerca de la implementación informática y acerca del modelo y los escenarios de simulación, para finalizar con los resultados y conclusiones obtenidos.

Por último, el capítulo 5 recoge conclusiones y valoraciones de carácter personal acerca del trabajo realizado.

## 2 Conceptos previos

En el presente capítulo se explican varios conceptos que se han creído necesarios para entender el contenido de la memoria.

### 2.1 Microrredes

No existe una definición de microrred universalmente aceptada [4]. En [5] se define microrred como una “interconexión de fuentes de energía distribuida con elementos de almacenamiento integrados”. En el año 2002, el CERTS (Consortium for Electric Reliability Technology Solutions) establece la siguiente definición para el concepto de microrred: “Agregación de cargas y microgeneradores que se opera como un sistema único y que provee tanto de energía eléctrica como térmica”.

Pese a que día de hoy todavía no se ha acuñado una definición formal de microrred, la bibliografía consultada suele coincidir a la hora de señalar los elementos que pueden formar parte de ella. La enumeración de elementos que viene a continuación sigue la clasificación que hace [1]:

1) **Generación:**

- Renovable o no renovable, en función del recurso energético.
- Controlables o intermitentes (modo de operación).

2) **Almacenamiento:** eléctrico / térmico.

3) **Cargas:**

- Eléctricas / térmicas.
- Cargas críticas: requieren mayor calidad de suministro (i.e. procesos industriales).
- Controlables o no controlables.

4) **Interruptor:** punto de interconexión con la red eléctrica (PCC, *Point of Common Coupling*).

5) **Protecciones:**

- Transiciones conectado-aislado-conectado (detección de condición en isla).
- Potencias de cortocircuito para detección de faltas.

6) **Sistemas de control:**

- Sistemas de control central de la microrred.
- Controladores locales (generadores controlados, cargas).
- Comunicaciones.
- Distribución de la lógica de control (control primario, regulación secundaria y servicios auxiliares).

El carácter distribuido de la generación es una consecuencia del tamaño de los generadores, mucho más pequeños que los de una planta convencional, así como de la disponibilidad física del recurso. Los microgeneradores se instalarán, en la medida de lo posible, próximos a los puntos de consumo para minimizar las pérdidas asociadas al transporte. Lo mismo sucede con las baterías y otros sistemas de almacenamiento.

Una microrred deberá ser capaz de llevar a cabo ciertas tareas que garanticen un suministro de calidad. En [1] y [5] se especifican algunas de las funciones con las que es deseable que cualquier microrred cuente:

- 1) Operación autónoma: que no se requiera de ninguna intervención externa para operar correctamente.
- 2) Gestión de la operación: despacho de potencia y control de tensiones.
- 3) Posibilidad de aislarse de la red principal (funcionamiento “en isla”) y capacidad para realizar las conexiones y desconexiones de forma controlada.
- 4) Seguimiento de carga (*load following*).
- 5) Alisamiento de picos de consumo.

Las microrredes estudiadas en este PFC disponen además de elementos adicionales para su monitorización, comunicación y control: se pueden considerar redes “inteligentes”. La gestión de las redes inteligentes debe de poder, no sólo de mantener el balance energético entre los generadores, las demandas de la red inteligente y el intercambio con la red principal, sino también optimizar la forma en la que esto se lleva a cabo con el objetivo de minimizar los costes.

Existen tres posibles enfoques o arquitecturas de control a la hora de llevar a cabo dichas labores de gestión: control distribuido, centralizado o semidistribuido.

### **Control centralizado**

Es el enfoque de control más clásico y extendido. Un único ente recibe todos los datos disponibles y los procesa para devolver posteriormente instrucciones y consignas. Al tratarse del sistema más implantado, existe una gran cantidad de información disponible al respecto. Sus puntos fuertes son la estabilidad, robustez y sencillez a la hora de implementarlo. Como debilidades cabría citar su lenta reacción bajo ciertos supuestos de funcionamiento, ya que su potencia de tratamiento de la información se concentra en un solo punto.

### **Control distribuido**

En el lado opuesto al control centralizado se encuentra la arquitectura totalmente descentralizada. Ya no se dispone de un único ente responsable de toda la red, sino que son varios entes o agentes los que se encargan de controlar su entorno más próximo. Cada agente recoge datos en un ámbito local, sin tomar en consideración lo que hay más allá de éste.

Este es el enfoque que más está siendo estudiado actualmente, y se postula como la línea de investigación más prometedora. La idea es explotar las ventajas que supone repartir la potencia de tratamiento de información en el espacio, haciendo un uso más eficiente de los recursos. Esto comporta normalmente unos tiempos de reacción más reducidos y, en ocasiones, la eliminación de los “cuellos de botella” que suelen aparecer en sistemas centralizados.

### **Control híbrido o semi-distribuido**

Como su nombre indica, esta arquitectura combina características de las dos anteriores. Si bien existe un notable grado de descentralización, cuenta con niveles jerárquicos definidos, los cuales tienen asociados distintos grados de responsabilidad y ámbitos de actuación. Aúna virtudes de ambas topologías al combinar la flexibilidad de un control distribuido con la sencillez y robustez del centralizado.

En base a esto, se ha decidido que la regulación de las dos microrredes objeto de estudio en este PFC siga una arquitectura semi-distribuida.

## **2.2 Técnicas heurísticas y optimización clásica**

La problemática de control abordada en este trabajo se ha enfocado de dos maneras distintas: mediante técnicas heurísticas de optimización y mediante optimización lineal.

### **Heurística**

Con el fin de resolver problemas complicados de manera eficiente, a veces es necesario comprometer algunos requisitos de optimalidad y construir una estructura de control que no garantice encontrar la mejor respuesta, pero que casi siempre encuentre una buena solución. De esta forma surge la idea de heurística. La palabra heurística viene de la palabra griega *heuriken* que significa “descubrir” (origen también del célebre eureka de Arquímedes).

La heurística es una técnica que aumenta la eficiencia de un proceso de búsqueda, aunque posiblemente sacrificando demandas de completitud. Una función heurística es una correspondencia entre las descripciones de estados del problema alguna media de deseabilidad, normalmente representada por números. En general, hay que hacer una ponderación entre el costo de evaluación de una función heurística y el ahorro de tiempo que ésta proporciona [6].

El enfoque mediante técnicas heurísticas del capítulo 3 va a establecer las consignas de control en base a un cierto número de directrices (estrategias). Un control de este tipo asegura buenos resultados, pero no garantiza la solución óptima.

### **Optimización clásica**

La optimización, en el sentido clásico del término, consiste en la selección de la mejor solución en algún aspecto concreto de entre todas las alternativas posibles. Los problemas de optimización suelen componerse de tres elementos [7]:

- **Función objetivo:** es la medida cuantitativa del funcionamiento del sistema que se desea optimizar (maximizar o minimizar). El ejemplo más ilustrativo sería una función de coste que interesa minimizar.
- **Variables:** representan las decisiones que se pueden tomar para afectar al valor de la función objetivo.
- **Restricciones:** representan el conjunto de relaciones (expresadas como ecuaciones o inecuaciones) que ciertas variables están obligadas a satisfacer. Los límites físicos de potencia que puede entregar un generador o el nivel de carga máximo de una batería serían ejemplos de restricciones.

La Fig. 4 constituye un ejemplo de optimización. En ella se muestra el gráfico de un paraboloide dado por la ecuación (1). La función objetivo es la propia ecuación y las variables son  $x$  e  $y$ . En este caso no se contemplan restricciones adicionales. El punto rojo situado en  $(0, 0, 4)$  es el que maximiza la función objetivo.

$$f(x, y) = -(x^2 + y^2) + 4 \quad (1)$$

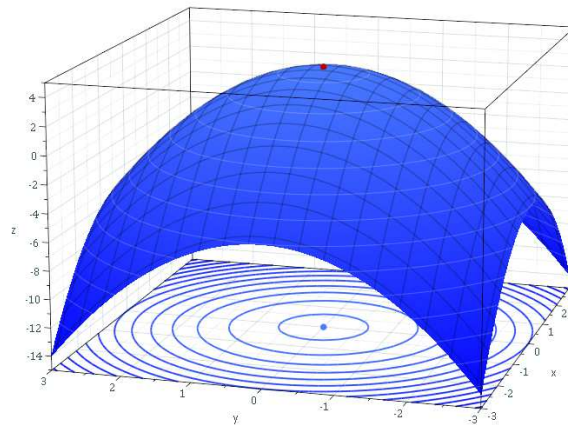


Fig. 4. Optimización de la ecuación (1)

## 2.3 Agentes software y sistemas multiagente

Los agentes software surgen dentro del campo de la Inteligencia Artificial, concretamente a partir de trabajos desarrollados en el área de DAI (Distributed Artificial Intelligence). Esta investigación inicial progresa hacia la madurez dando lugar a conceptos como la Programación Orientada a Agentes y los Lenguajes de comunicación de Agentes. Posteriormente, este concepto se extiende al resto de la Ingeniería de Software, planteándose hoy en día la Ingeniería de Software Orientada a Agentes [8].

### 2.3.1 Qué es un agente

Al igual que para microrred, no existe una definición universalmente aceptada de agente. No obstante, todas las definiciones coinciden al señalar cuatro características definitorias: los agentes software son entidades autónomas, inteligentes, poseen movilidad, y presentan un comportamiento social [4].

#### I. Agente como entidad **autónoma**.

Autonomía, reactividad e iniciativa son las tres propiedades que caracterizan al agente como entidad autónoma:

- Autonomía: es capaz de trabajar sin la intervención directa del usuario y tiene cierto control sobre sus acciones y estado interno.
- Reactividad: puede percibir su entorno (ya sea el mundo físico, a través de sensores, un usuario, por medio de una interfaz gráfica o vocal, aplicaciones



en la red, otros agentes; etc.) y responder oportunamente a cambios que se produzcan en el mismo.

- **Iniciativa:** es proactivo, puesto que tiene la capacidad de actuar incluso en ausencia de estímulos externos. Su actuación viene determinada por los objetivos a cumplir, y tratará de lograr su consecución [9].

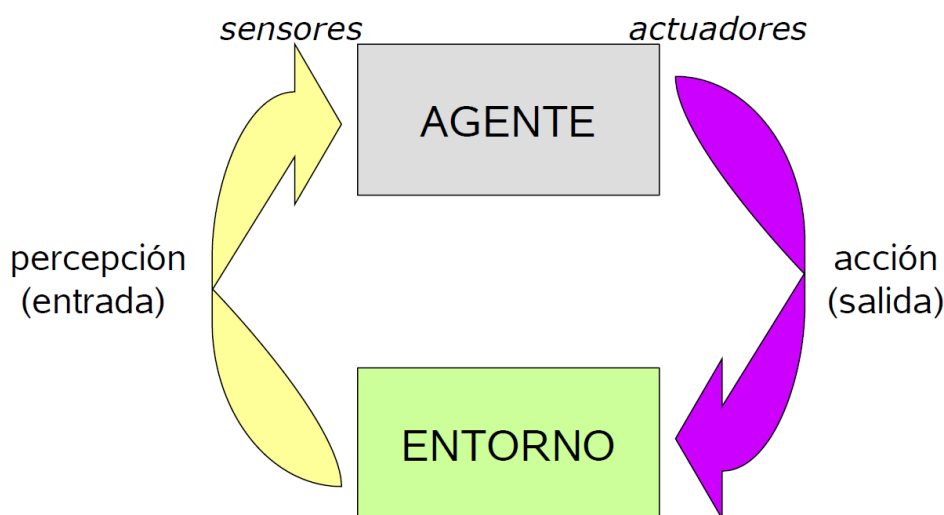


Fig. 5. Agente como entidad autónoma que interacciona con su entorno

## II. Agente como entidad **inteligente**.

Un agente es capaz de razonar y aprender:

- Razona, porque decide qué objetivos perseguir, a qué eventos reaccionar y cómo actuar para conseguir sus metas. También puede suspender o abandonar un objetivo para dedicarse a otro.
- Aprende, porque mediante técnicas de aprendizaje puede adaptarse mejor al entorno y a las variaciones dinámicas de éste. Es capaz de mejorar su proceso de toma de decisiones en base a su experiencia.

## III. Los agentes poseen **movilidad**.

Un agente puede migrar de un nodo a otro en una red preservando su estado interno. Este aspecto es de gran importancia por dos motivos:

- El agente puede adquirir un conocimiento global del sistema viajando y recopilando estímulos locales.
- Permite repartir la potencia de cálculo del sistema entre más de un procesador.

#### IV. Agente como ente **social**.

Los agentes tienen la capacidad, incluso la necesidad, de interactuar con otros agentes para llevar a cabo sus tareas y alcanzar sus objetivos [9]; poseen habilidades sociales:

- Interacción: pueden dialogar con otros agentes.
- Delegación: están habilitados para asignar tareas.
- Cooperación: son capaces de trabajar junto a otros agentes para lograr objetivos en común.
- Coordinación: pueden organizar el proceso conducente a solucionar un problema de forma que se eviten interacciones nocivas y se exploten al máximo las beneficiosas.
- Negociación: de producirse un conflicto de intereses, son capaces de llegar a un acuerdo que sea aceptable por todas las partes implicadas.

Para ilustrar el comportamiento social de los agentes se proponen los dos ejemplos que vienen a continuación [4]:

##### Ejemplo 1)

Un radiador eléctrico se encuentra controlado por dos agentes, en el que cada uno tiene objetivos distintos. Un agente intenta llegar a la temperatura deseada lo más rápido posible, mientras que el otro pretende controlar el uso de la resistencia para alargar la vida del radiador. Cada agente desea resultados a priori incompatibles: existe un conflicto de intereses y será necesario negociar una solución intermedia.

##### Ejemplo 2)

Dos agentes instalados en sendas bombas de agua quieren mantener el nivel de agua de una piscina a un mismo valor constante. En este caso los objetivos de cada agente coinciden y deberán comunicarse y cooperar entre ellos para decidir cuánto caudal vierte cada uno a la piscina, manteniendo así fijo el nivel de agua.

### 2.3.2 Sistemas multiagente

Un conjunto de agentes autónomos actuando en un mismo entorno constituye un sistema multiagente (SMA). Un SMA afronta los problemas siguiendo la estrategia del “divide y vencerás”. Los elementos que lo componen son [8]:

- 1) Un entorno.
- 2) Un conjunto de objetos.
  - Los objetos se encuentran integrados con el entorno. En un momento dado, es posible asociar uno de estos objetos con un lugar en el entorno.
  - Los objetos son pasivos, pueden ser percibidos, creados, destruidos y modificados por los agentes.
- 3) Un conjunto de agentes.
- 4) Un conjunto de relaciones que unen objetos y agentes.
- 5) Un conjunto de operaciones que hacen posible que los agentes perciban, produzcan, consuman, transformen y manipulen los objetos.
- 6) Operadores que representan la aplicación de operaciones sobre el mundo y la reacción de éste al ser alterado (las “leyes del universo”).

Las principales características de un SMA son:

- **Heterogeneidad:** cada agente se especializa en una tarea concreta, lo que permite un reparto de responsabilidades.
- **Escalabilidad:** los SMA son escalables. La escalabilidad es una propiedad muy deseable e indicativa de la capacidad del sistema para adaptarse a las circunstancias cambiantes. El SMA seguirá funcionando correctamente aunque se modifique su tamaño. Esto aporta gran flexibilidad, al permitir añadir o quitar agentes sobre la marcha sin tener que parar o reajustar el SMA.
- **Distribución:** se distribuye físicamente el conocimiento y la gestión de recursos. Un sistema distribuido exhibe una buena tolerancia a fallos, pues podrá seguir funcionando en caso de que alguna de las partes deje de funcionar.

### 2.3.3 Comunicación entre agentes

Los agentes carecen de una visión global del sistema, lo que implica que la información que recopilen por sí mismos nunca podrá ser completa. Necesitan interactuar con otros agentes para alcanzar sus objetivos: necesitan comunicarse. La comunicación entre agentes está basada en la teoría de actos del habla [10]. En concreto, sigue las especificaciones promovidas por la *Foundation for Intelligent Physical Agent* (FIPA).

La FIPA es una organización que se encarga de establecer estándares sobre la tecnología basada en agentes para asegurar la interoperabilidad (es decir, ser capaz de compartir información) entre SMA de distinto tipo. FIPA ha fijado también estándares para la comunicación entre agentes que se conocen como FIPA – ACL (Agent Communication Language). Estos estándares establecen una serie de mensajes que representan distintas acciones del habla o actos comunicativos. FIPA – ACL reconoce veintidós actos comunicativos [11], de los cuales se recogen algunos de los más relevantes en la Tabla 1.

Acto comunicativo	Descripción
Request	Solicita al receptor que realice alguna acción
Inform	Informa al receptor de que una proposición es cierta
CFP (Call For Proposal)	Solicita propuestas para realizar una acción dada
Propose	Envía una proposición para llevar a cabo alguna acción
Accept Proposal	Acepta la propuesta recibida
Request	Solicita al receptor que realice alguna acción
Cancel	Cancela alguna acción previamente solicitada
Subscribe	Anuncia la intención del emisor de ser informado de toda acción relevante que el receptor lleve a cabo

Tabla 1. Actos comunicativos FIPA-ACL.

## 2.4 Framework JADE

El *Java Agent Development Framework*, o JADE, es una plataforma software completamente escrita en lenguaje Java (requiere la versión 1.4, como mínimo) que simplifica sobremanera la implementación de sistemas multiagente gracias a un *middleware* que sigue las especificaciones promovidas por la FIPA, y a un set de herramientas gráficas que ayudan en el tedioso proceso de depurar fallos durante las distintas fases de desarrollo del código.

Una plataforma multiagente puesta en marcha con JADE puede distribuirse entre varios ordenadores, que ni siquiera tienen por qué tener instalado el mismo sistema operativo. La configuración puede ser controlada de forma sencilla por medio de una Interfaz Gráfica de Usuario y ser modificada sin que sea necesario detener el funcionamiento del SMA. Asimismo, los agentes disponen de total libertad para migrar de una plataforma a otra dinámicamente.

JADE es distribuido por Telecom Italia, dueño del *copyright*, bajo una licencia de código abierto. Desde Mayo de 2003, el proyecto es supervisado por cinco miembros: Telecom Italia, Motorola, Whitestein Technologies AG, Profactor GmbH y France Telecom R&D. La última versión de JADE fue lanzada el 26 de Junio de 2012 y puede descargarse de forma gratuita desde la página web oficial de la aplicación: <http://jade.tilab.com/>.

A modo de resumen, JADE proporciona:

- Un “entorno de tiempo” o *runtime environment* donde los agentes pueden desenvolverse e interactuar, y que debe ser arrancado en un host antes de que los agentes puedan ejecutarse.
- Una librería de clases que los programadores pueden utilizar para desarrollar sus agentes.
- Un kit de interfaces y herramientas gráficas con las que administrar y monitorizar la actividad de la plataforma multiagente.

### 2.4.1 Containers y Plataformas

Cada instancia del *JADE runtime environment* se llama Container, puesto que puede “contener” varios agentes. Al conjunto de containers activos se le denomina Plataforma. En cada plataforma, debe existir un *main container* activo en el que el resto de containers

debe registrarse nada más comenzar su ejecución. Para ello, es necesario indicarles dónde encontrar (host y puerto) al *main container*. Si otro *main container* arranca en la red que conforma el sistema multiagente, constituye una plataforma diferente en la que pueden registrarse nuevos containers.

La Fig. 6 ilustra con un ejemplo los conceptos que se acaban de explicar. El sistema multiagente del ejemplo está formado por dos plataformas JADE compuestas de uno y tres containers respectivamente. Los agentes JADE pueden comunicarse entre sí sin importar su localización gracias a que se identifican de manera inequívoca con un nombre único (Agent Identifier, AID): mismo container (por ejemplo los agentes A2 y A3), distintos containers dentro de una misma plataforma (agentes A1 y A2) o distintas plataformas (agentes A4 y A5).

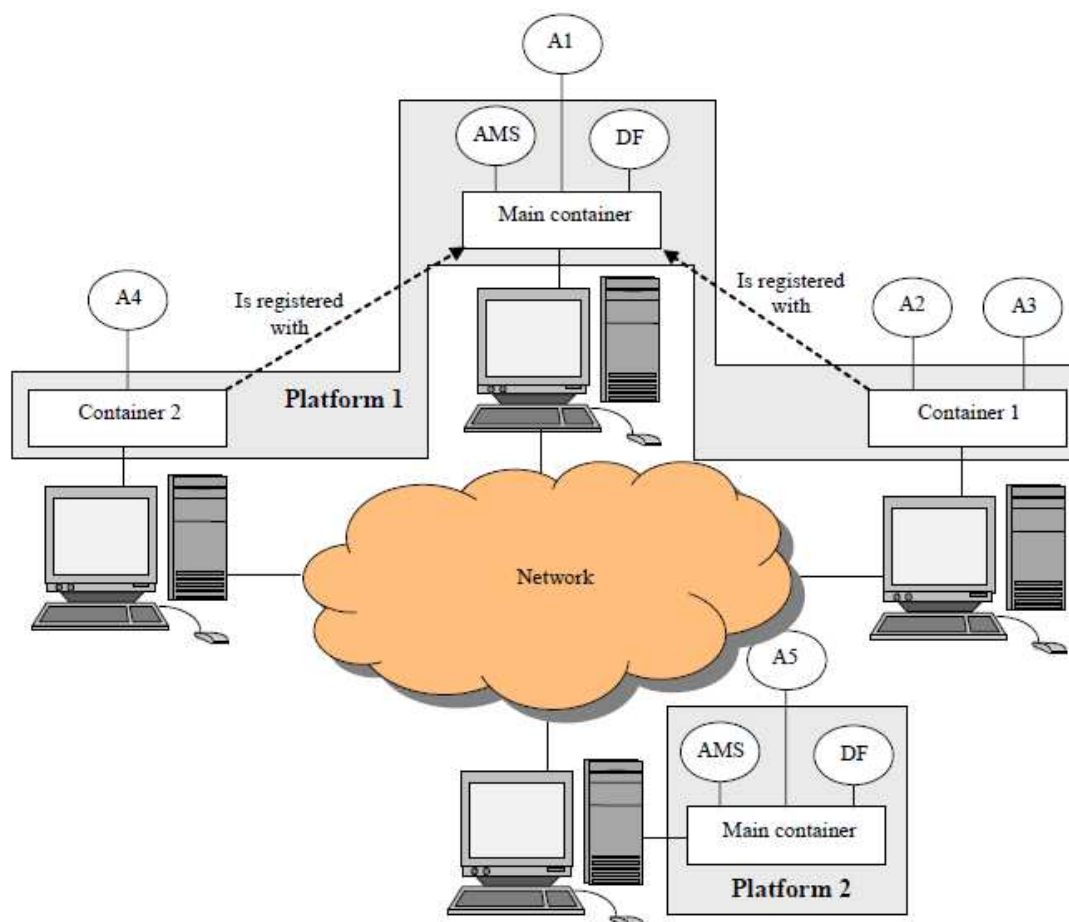


Fig. 6. SMA distribuido entre varios ordenadores conectados en red [12]

Para trabajar y programar sistemas multiagente usando JADE no es necesario conocer el funcionamiento subyacente del *JADE runtime environment*, tan sólo iniciarlo antes de ejecutar los agentes.

#### 2.4.2 Los agentes AMS, DF y RMA

Además de la posibilidad de permitir a otros containers registrarse en él, el *main container* difiere de un container normal al contener tres agentes especiales que se inician de forma automática en el arranque: AMS, DF y RMA.

El agente **AMS** (*Agent Management System*) es la máxima autoridad del SMA y el que controla la plataforma. Es el único que puede crear y destruir a otros agentes, containers, o incluso parar la ejecución de la plataforma. También es el encargado de asegurar que todos los agentes del sistema posean nombres únicos, de forma que sea imposible que surjan malentendidos en la comunicación. Si un agente es eliminado, el AMS es el responsable de notificárselo al resto de agentes, para que no lo tenga en consideración cuando lleven a cabo nuevas operaciones.

El agente **DF** (*Directory Facilitator*) ejerce la función de “Páginas Amarillas”. Los agentes registran en él los servicios que ofrecen y buscan a su vez servicios que necesitan de otros agentes tal y como se ilustra en la Fig. 7.

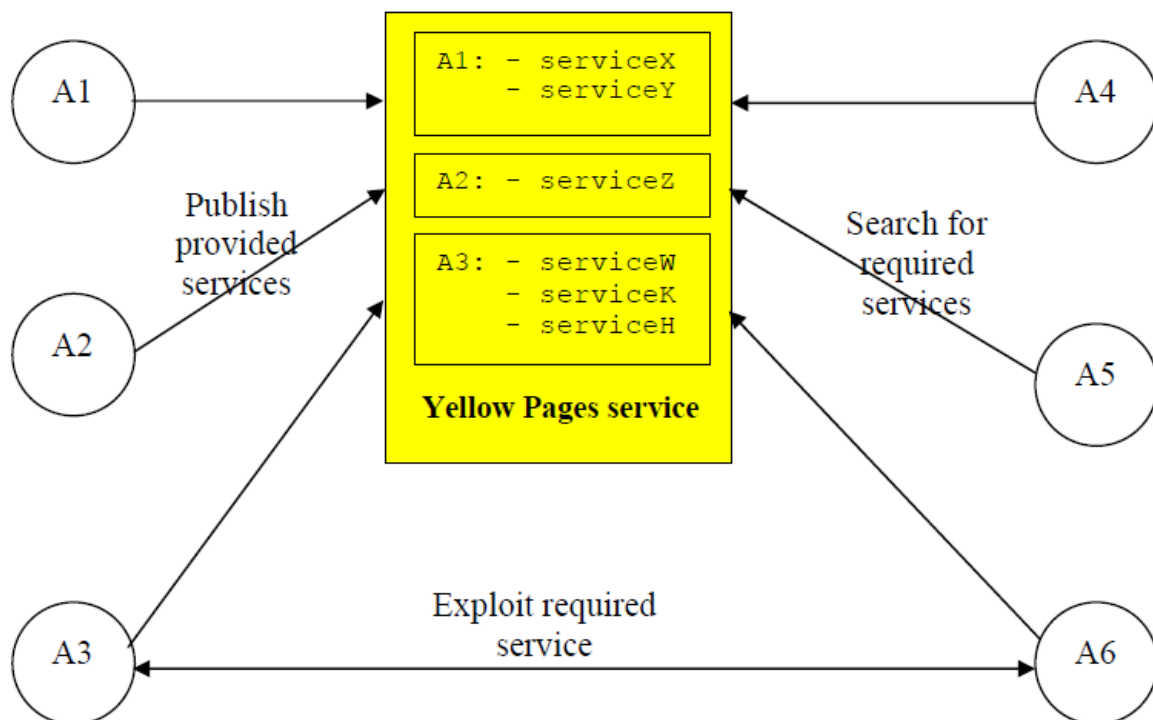


Fig. 7. Funcionamiento del servicio de páginas amarillas del agente DF [12]

Los agentes A1, A2 y A3 son los únicos que ofrecen servicios en este ejemplo de la Fig. 7. Los agentes A4, A5 y A6 van a demandar servicios. El proceso que se sigue es simple. Primeramente, los agentes registran los servicios ofertados en las páginas amarillas del agente DF. A partir de este momento, cuando un agente requiera un servicio, buscará en las páginas amarillas si se oferta y, en caso afirmativo, preguntará quién lo ofrece y se pondrá en contacto con él de forma directa mediante el envío de un mensaje ACL.

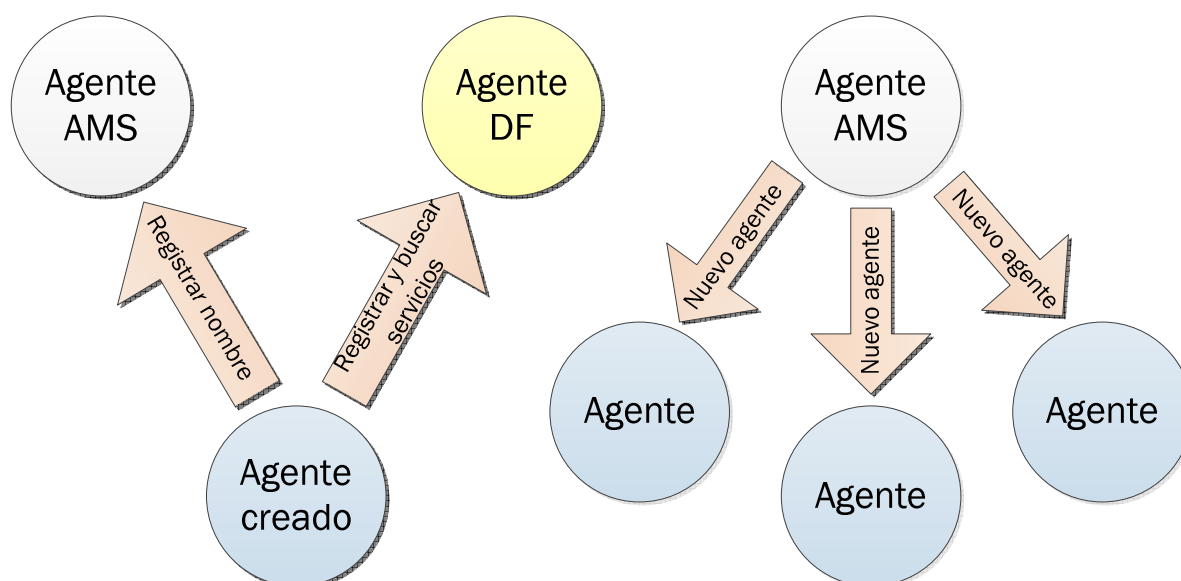


Fig. 8. Creación de agentes y notificaciones a AMS y DF

El agente **RMA** (*Remote Agent Management*) proporciona al usuario una interfaz gráfica desde la que se pueden llevar a cabo numerosas acciones: añadir, eliminar o mover agentes; crear nuevos *containers*; utilizar herramientas de depuración de fallos; etc.

La Fig. 9 es una captura de pantalla de la interfaz que proporciona el agente RMA, realizada durante la ejecución de un hipotético SMA distribuido entre distintos ordenadores de varios edificios del Campus Río Ebro.

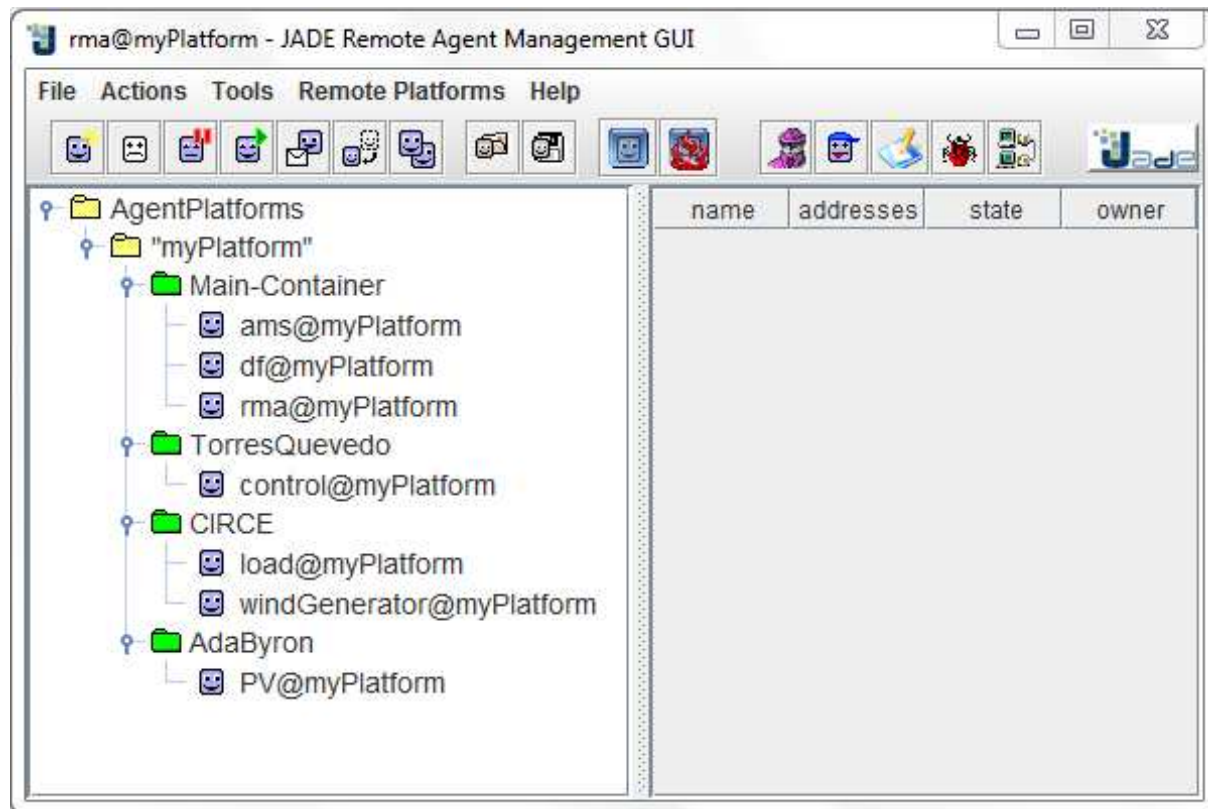


Fig. 9. Interfaz gráfica del agente RMA

El SMA cuenta con una única plataforma llamada “myPlatform”. Dentro de la plataforma existen cuatro containers, representados por carpetas de color verde. Los containers no tienen por qué estar ejecutándose en el mismo ordenador, sino que pueden estar situados en servidores que estén a kilómetros de distancia y seguir comunicándose vía Ethernet.

En el ejemplo de la Fig. 9, el agente “control” se ejecutaría en el edificio Torres Quevedo, mientras que en el Ada Byron se podría encontrar el agente “PV”, encargado de gestionar unas placas fotovoltaicas. En el edificio CIRCE podría estar ejecutándose en el mismo servidor los agentes “load” y “windGenerator”, responsables de la gestión de una carga y de un generador eólico respectivamente. El Main-Container podría haberse arrancado en cualquiera de los servidores mencionados, o en algún otro dispuesto para tal función.

### 2.4.3 Agentes en JADE: la clase Agent

Crear un agente en JADE es tan sencillo como extender la clase `jade.core.Agent` e implementar el método `setup()`, tal y como se muestra en el siguiente fragmento de código:

```
import jade.core.Agent;
public class MyAgent extends Agent {

    protected void setup() {
        // Comandos a ejecutar durante la inicialización.
    }
}
```

El agente continúa ejecutándose aunque no lleve a cabo ninguna tarea más allá de la propia inicialización. Para poner fin a su ejecución es necesario llamar al método `doDelete()`. Similar al método `setup()`, que es invocado por el *JADE runtime* tan pronto como un agente es iniciado y que lleva a cabo las inicializaciones antes mencionadas, el método `takeDown()` se invoca justo antes de eliminar el agente y está ideado para llevar a cabo operaciones de limpieza.

```
public class Emergency_Agent extends Agent {

    protected void setup() {
        // Comandos a ejecutar durante la inicialización.
        if(emergency_mode == true) {
            // El agente continuará ejecutándose.
        }
        else {
            // Si no es necesario se procede a su eliminación.
            System.out.println("System running properly. No extra " +
                               + "action required.");
            doDelete();
        }
    }

    protected void takeDown() {
        System.out.println("Emergency-agent " + getAID().getName()+
                           + " terminating.");
    }
}
```



## 2.4.4 Comportamientos

Un agente realiza su función por medio de sus comportamientos o *behaviours*. Un comportamiento representa una tarea que un agente es capaz de llevar a cabo y es implementado en una clase que extiende la clase `jade.core.behaviours.Behaviour`.

Para que un agente ejecute una tarea definida en un comportamiento, basta con llamar al método `addBehaviour()` de la clase `Agent`. Un comportamiento puede ser añadido en cualquier momento, ya sea en el inicio, cuando el agente ejecuta su método `setup()`; o desde cualquiera de sus otros comportamientos en ejecución.

Una clase que extiende `Behaviour` debe implementar obligatoriamente dos métodos: `action()` y `done()`. El método `action()` define las operaciones a realizar mientras el comportamiento esté en ejecución, mientras que el `done()` retorna una variable booleana que especifica si el comportamiento ha sido completado, y tiene por tanto que ser retirado del *pool* de comportamientos activos.

El gráfico de flujo de la Fig. 10 muestra el ciclo de vida de un agente. Los métodos que debe implementar el programador aparecen en color rojo.

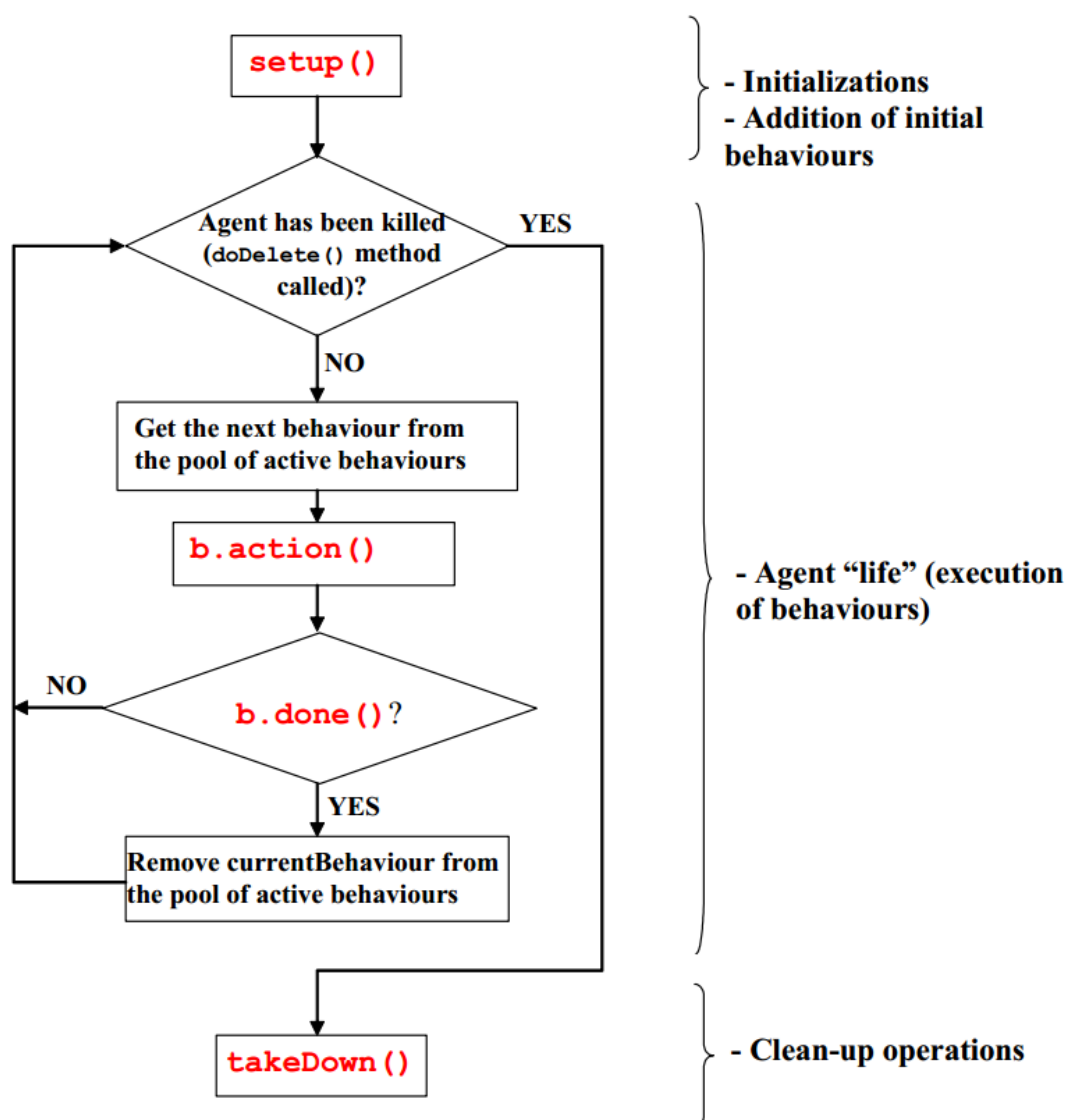


Fig. 10. Ciclo de vida de un agente [12]

En el ejemplo que viene a continuación se ha programado un comportamiento llamado `MyBehaviour` mediante una extensión de la clase `Behaviour`. Las tareas del comportamiento se incluirían dentro del método `action()`.

```
public class MyBehaviour extends Behaviour {

    public void action() {
        if (something == true) {
            // Lista de comandos a ejecutar.
        }
    }

    public boolean done() {
        return true;
    }
}
```

JADE ofrece clases para varios tipos de *behaviours*. Los tres más importantes son *one-shot*, *cyclic* y *generic*.

- **One-shot behavior:** este comportamiento se ejecuta una única vez. La clase `jade.core.behaviours.OneShotBehaviour` implementa el método `done()` de forma que retorne siempre *true*, y puede extenderse para implementar comportamientos de esta naturaleza.

```
public class MyOneShotBehaviour extends OneShotBehaviour {

    public void action() {
        // Realizar operación X.
    }
}
```

La operación X se realiza una única vez.

- **Cyclic behaviour:** como su nombre indica, este comportamiento nunca detiene su ejecución y el método `action()` realiza las mismas operaciones cada vez que es llamado. La clase `jade.core.behaviours.CyclicBehaviour` implementa el método `done()` de forma que devuelva siempre *false*, y puede extenderse para implementar comportamientos que encajen en esta definición.

```
public class MyCyclicBehaviour extends CyclicBehaviour {

    public void action() {
        // Realizar operación Y.
    }
}
```

La operación Y se realiza de manera cíclica.

- **Generic behaviours:** las operaciones de este tipo de comportamientos varían en función de un estado interno del que se tiene memoria. Se comporta como una Máquina de Estados Finitos (*Finite State Machine*, FSM), que se mueve de una situación a otra cuando determinadas condiciones son verificadas:

```
public class MyThreeStepBehaviour extends Behaviour {  
  
    private int step = 0;  
  
    public void action() {  
        switch (step) {  
            case 0:  
                // Realizar operación X.  
                step++;  
                break;  
            case 1:  
                // Realizar operación Y.  
                step++;  
                break;  
            case 2:  
                // Realizar operación Z.  
                step++;  
                break;  
        }  
    }  
  
    public boolean done() {  
        return step == 3;  
    }  
}
```

Las operaciones X, Y y Z se llevan a cabo una tras otra hasta que el comportamiento se completa.

JADE ofrece la posibilidad de combinar diferentes tipos de comportamientos simples para crear otros más con funciones más especializadas. Además de los tres *behaviours* antes mencionados, las librerías JADE cuentan con varias clases que contemplan situaciones más específicas. Por citar algunas: *SequentialBehaviour*, *ParallelBehaviour* y *FSMBehaviour*; cuyos nombres son lo suficientemente ilustrativos.

## 2.5 Antecedentes

El empleo de tecnología de agentes para el control de microrredes ha ido ganando popularidad durante los últimos años. En la revisión bibliográfica llevada a cabo para el proyecto se han encontrado multitud de trabajos que siguen esta línea. A continuación se hará referencia a los que se han considerado de mayor interés.

El artículo [13] ofrece una visión panorámica de las investigaciones llevadas a cabo por universidades e instituciones de Europa, EE.UU., Canadá y Japón en el ámbito de las microrredes. En algunos casos, estas investigaciones han conducido a la construcción de emuladores como el que se pretende realizar en CIRCE, y cuyo punto de partida es este PFC.

El artículo [14] presenta un emulador de microrredes para uso docente e investigador. Este emulador implementa una parte de la microrred con elementos físicos reales tales como cargas, generadores, baterías y líneas de transmisión, y otra parte mediante simulación con software. Los desarrolladores se valen de la tecnología de agentes y del

*framework* JADE para implementar el control, y del programa LabVIEW para establecer las comunicaciones con los elementos reales anteriormente mencionados.

En los artículos [15] y [16] se pueden encontrar otros dos ejemplos de microrredes emuladas en tiempo real mediante agentes JADE. Ambos hacen especial hincapié en la ontología del SMA desarrollado, especificando los tipos de agentes participantes y sus funciones. En [15], además, se contemplan dos modos de operación: microrred conectada a red y funcionamiento en isla.

El artículo [17] presenta el sistema de control de una microrred de tres nodos, estableciendo tres niveles jerárquicos de control que designa como RMU, iNode e iSocket respectivamente. Las siglas de RMU corresponden a *Remote Manager Unit*, mientras que la “i” que actúa de prefijo para los Nodes (nodos) y Sockets (tomas o enchufes) se refiere a “Intelligent”. El enfoque por optimización clásica realizado en este PFC (capítulo 4) reproducirá esta misma estructura jerárquica, aunque con distinta nomenclatura.

Por último, los artículos [18] y [19] combinan un control basado en SMA JADE con microrredes modeladas en Simulink. Ambos diseños se valen del *middleware* MACSimJX para establecer los canales de comunicación (bidireccionales) entre los agentes y la simulación.

### 3 Optimización mediante técnicas heurísticas

En este capítulo se recoge el trabajo llevado a cabo para el proyecto OptimaGrid. OptimaGrid se enmarca dentro de la segunda convocatoria de proyectos del Programa de Cooperación Territorial Espacio Sudeste Europeo (SUDOE) 2007-2013, que apoya el desarrollo regional a través de la cofinanciación de proyectos transnacionales por medio del FEDER.

OptimaGrid tiene como objetivos definir, diseñar e implementar sistemas de control inteligentes de la energía que permitan gestión en tiempo real de una micro-red de distribución de energía eléctrica, aplicada a un área industrial con elevado porcentaje de penetración de energías renovables.



Fig. 11. Logos de OptimaGrid, FEDER y SUDOE

Se pretende lograr que las áreas industriales sean autogestionables energéticamente, y tratar de cambiar el concepto contaminante asociado a ellas por otro concepto de áreas ecológicas. Para ello se profundiza de nuevo en el concepto “Smart Grid” o Red Inteligente. La red inteligente es la convergencia de la energía eléctrica y las tecnologías de la información y la comunicación (TIC).

En OptimaGrid se desarrollan procesos ubicuos en sistemas multiagente, que adoptan dinámicamente el balance generación-demanda, para conseguir un suministro de energía más eficiente, sostenible, económico y seguro, con la máxima prioridad en el uso de los recursos renovables locales.

#### 3.1 Propuesta de trabajo

El proyecto OptimaGrid ha sido llevado a cabo por varios socios, entre los que se encuentra la fundación CIRCE; ha tenido una duración de dos años, y ha seguido tres líneas principales de investigación:

1. Medidas de ahorro y eficiencia energética.
2. Sistemas de almacenamiento de energía.
3. Actuación sobre la curva de demanda.

El trabajo realizado en el presente capítulo busca optimizar el uso de los sistemas de almacenamiento de una microrred, situándose por lo tanto dentro de la segunda línea de investigación. Se pretende implementar un sistema distribuido de gestión, basado en una arquitectura multiagente, que controle diversos aspectos de una microrred de forma autónoma. El *framework* elegido para dicha misión es JADE, por los motivos mencionados en el apartado 2.4.

Este capítulo continúa el trabajo llevado a cabo por Hans Bludszuweit para OptimaGrid, recogido en un informe realizado para CIRCE [20], en el que se realizó un modelo en MATLAB-Simulink que llevaba a cabo la gestión del almacenamiento en función de varias estrategias de control. Los resultados obtenidos con este modelo serán usados como referencia para la validación del emulador de microrredes que se va a realizar.

El emulador constaría de dos partes: 1) electrónica de potencia y 2) un simulador de la microrred. Esta propuesta de trabajo encaja perfectamente en el paradigma de simulación “*Power Hardware in the Loop*” (PHIL) [21], puesto que combina dispositivos hardware que funcionan en tiempo real (cargas, baterías, generadores, etc.) con software de simulación.

La electrónica de potencia permitiría la emulación de cualquier generador o carga, así como la conexión con diversos dispositivos reales tales como generadores, baterías o cargas, mientras que el simulador de la microrred estaría formado por un clúster de ordenadores, y se encargaría de llevar a cabo los cálculos de flujos de potencia del resto de la microrred de una forma distribuida. De esta manera, el carácter distribuido de los generadores y cargas que componen una microrred podría ser reproducido en el simulador, al estar éste basado en una arquitectura multiagente.

Cada elemento de la microrred (generadores, cargas, etc.) contará con un agente asociado. Una vez inicializados, los agentes deberán de ser capaces de actuar de forma independiente según hayan sido programados. El emulador ofrecerá flexibilidad “plug-and-play” [22]. Una de las principales características de los SMA es precisamente ésta: los agentes pueden “ser enchufados” o ser eliminados de forma totalmente asíncrona en cualquier momento de la ejecución. También pueden replicarse en caso de fallo para asegurar su propia supervivencia y el consiguiente funcionamiento del SMA.

Cada agente asociado a un elemento de la microrred dispondrá de dos modos de funcionamiento: *simulación* y *monitorización*. En el modo *simulación* los inputs serán obtenidos de algún tipo de archivo, por ejemplo una hoja de cálculo Excel. De igual manera, los outputs quedarán recogidos en otro archivo con el que poder trabajar con posterioridad.

Cuando un agente se encuentre en modo *monitorización*, los inputs provendrán de lecturas reales del elemento hardware en cuestión, que serán realimentadas a través de tarjetas DPS al agente encargado de su gestión. Los outputs del agente, es decir, las consignas de P y Q a ejecutar por el dispositivo real, se enviarán también al elemento físico por medio de las mismas tarjetas DSP.

## 3.2 Descripción del área sujeto de estudio

Para validar el funcionamiento del emulador se ha optado por utilizar series temporales de datos reales de demanda y generación de un pequeño polígono industrial. Se trata de áreas en las que la actividad proviene principalmente de talleres, pequeñas factorías y almacenes. Este tipo de parques industriales es muy común en la zona del sudoeste europeo o SUDOE, que abarca Portugal, España y el sudoeste de Francia (Fig. 12). En la ciudad de Zaragoza existen varios polígonos y áreas que encajarían en este perfil, como El Portazgo o Mercazaragoza.

La lectura de datos se ha realizado durante todo un año y en intervalos de 30 minutos. En cada lectura quedan registrados los valores medios de la demanda de potencia activa y del factor de potencia del polígono durante la última media hora. En cuanto a la generación, se dispone de sendas series temporales de oferta eólica y fotovoltaica con pasos de tiempo idénticos a los de la demanda. Por lo tanto, se cuenta con un total de 17520 muestras para cada una de las variables de entrada.



**Fig. 12.** Mapa del sudoeste europeo, SUDOE

Ante tal cantidad de datos, resulta de gran utilidad poder analizarlos de forma visual mediante representaciones gráficas. Se ha optado por los diagramas de caja, que proporcionan información acerca de los valores mínimos y máximos que alcanza una variable, de los cuartiles (Q1, Q2 o mediana y Q3) y de la simetría de la distribución.

### 3.2.1 Recurso renovable

En la Fig. 13 y la Fig. 14 aparecen recogidos los datos de generación eólica y fotovoltaica respectivamente. La interpretación de este tipo de gráficos puede resultar engañosa si el lector no está familiarizado con ellos.

Cada figura cuenta con 13 cajas: 12 cajas correspondientes a los meses del año más una última caja en la que se presenta agrupado el año completo. Cada mes cuenta con un rectángulo (caja) de la que salen dos líneas acabadas en un punto (bigotes). El punto superior representa el valor máximo que ha alcanzado la variable en dicho mes, mientras que el inferior representa su valor mínimo.

El lado inferior de la caja marca el primer cuartil (Q1), es decir, informa de que el 25% de los valores recogidos durante ese mes están por debajo de dicho punto. El límite superior de la caja corresponde a Q3, el tercer cuartil. Q3 indica que el 75% de los valores de la muestra se sitúan por debajo de él.

Finalmente, la línea negra en trazo grueso que une los distintos meses es la encargada de señalar la media aritmética de los valores. Se ha decidido incluir la media en lugar de la mediana (Q2) porque ofrece una mejor perspectiva de la evolución de la disponibilidad del recurso a lo largo del año.

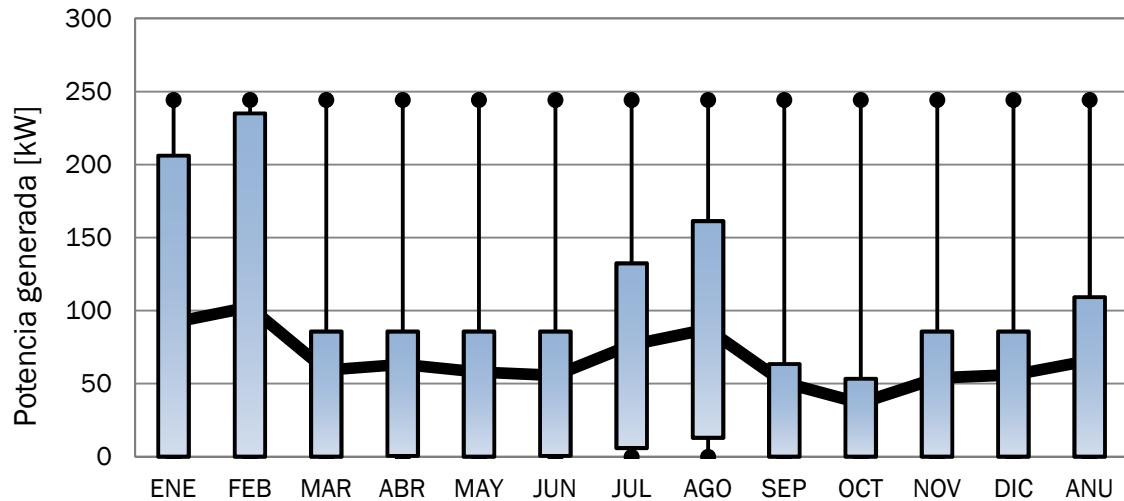


Fig. 13. Recurso eólico anual

Los valores mínimos de cada uno de los meses son cero, correspondiendo a momentos en los que no hay potencia útil extraíble del recurso renovable. Exceptuando Julio y Agosto de la gráfica de recurso eólico (Fig. 13), todos los meses sitúan su Q1 en cero, o muy próximo a cero.

Un primer cuartil situado en cero pone de manifiesto la gran variabilidad de las energías renovables y supone uno de sus principales inconvenientes. Al menos un 25% del tiempo la potencia extraíble del recurso renovable es nula. En el caso fotovoltaico (Fig. 14) es de prever, ya que durante la noche no hay generación alguna, pero en el caso eólico sí es reseñable que más de una cuarta parte del tiempo el aporte neto a la microrred sea cero.

El pico de generación fotovoltaica se sitúa en los meses de primavera y verano, decayendo conforme se acerca el invierno, mientras que en la generación eólica se producen dos: uno en los meses de enero y febrero, y otro en los de julio y agosto.

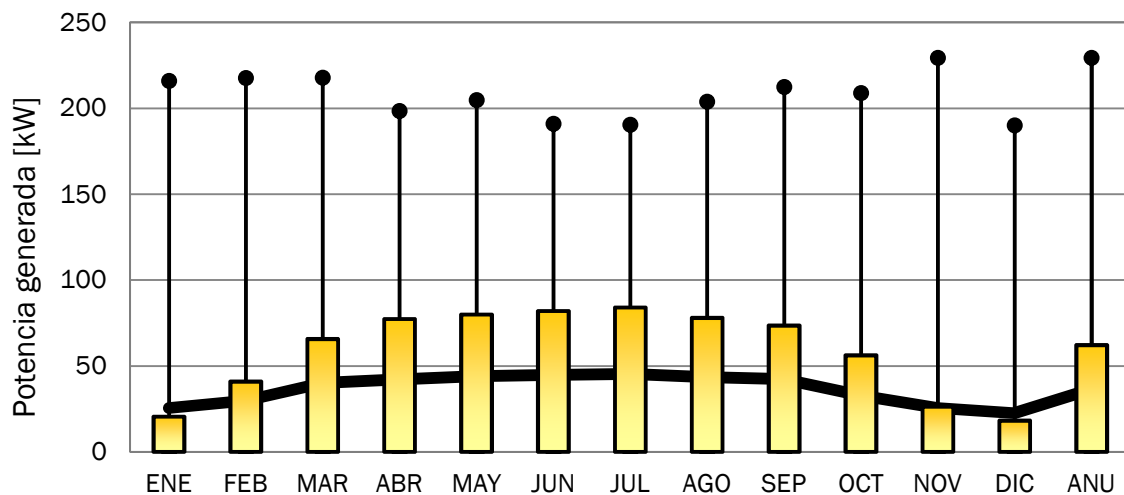


Fig. 14. Recurso fotovoltaico anual



En la Fig. 15 se ha representado el recurso renovable total, esto es, la suma de generación eólica y fotovoltaica. En esta ocasión el promedio se mantiene razonablemente estable. El alto valor que presenta el recurso eólico en los meses de invierno compensa la escasa generación solar durante los mismos. Durante en verano, el incremento de potencia generada que presentan ambos recursos en sus respectivas gráficas se ve reproducido, como no podía ser de otra manera, en el pico de julio y agosto de la Fig. 15.

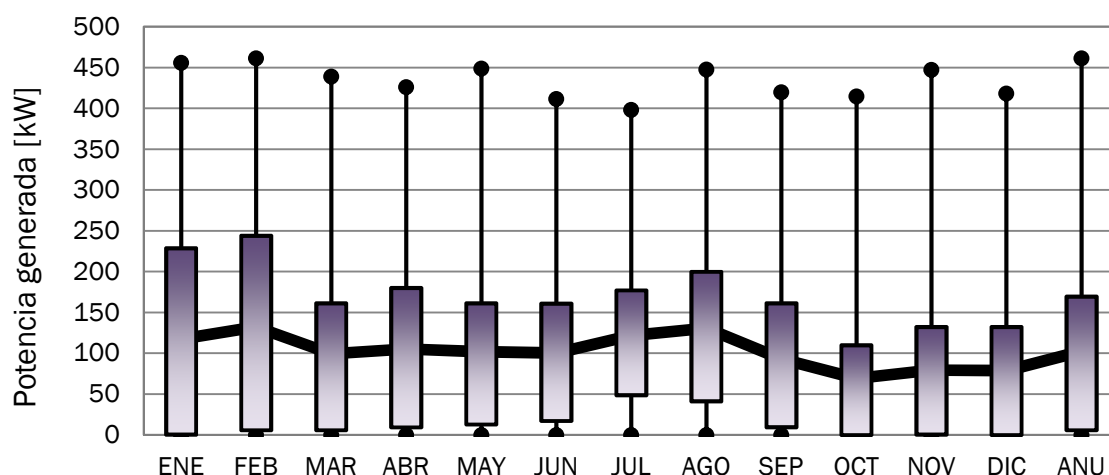


Fig. 15. Recurso renovable (eólico + fotovoltaico)

Para finalizar el análisis del recurso renovable anual, se presentan en la Tabla 2 cinco parámetros estadísticos de especial interés. Tal y como se podía intuir al ver las gráficas, el recurso eólico presenta mayor variabilidad y mayor potencia en general. El recurso fotovoltaico solo puede generar energía de día, de ahí que su mediana se sitúe casi en cero (0.18 kW).

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
$P_{wind}$	65.91	21.96	85.22	0	244.12
$P_{pv}$	36.50	0.18	54.49	0	229.39
$P_{total}$	102.40	66.87	104.55	0	461.71

Tabla 2. Parámetros estadísticos de las series temporales de P renovable (kW)

### 3.2.2 Demanda y balance de potencia en PCC

Una vez estudiada la distribución estadística de la generación renovable del polígono industrial, se pasa a analizar de la distribución de su demanda.

El perfil anual de la demanda se muestra en la Fig. 16, y su fisonomía difiere bastante del de la generación renovable. La línea horizontal que representa el consumo promedio de cada mes se mantiene entorno a los 75 kW durante prácticamente todo el año.

En lo que a valores máximos y mínimos respecta, se observa una ligera variabilidad a lo largo del año. A los meses de febrero, marzo, septiembre y diciembre les corresponde un valor mínimo de cero kW. Estos valores tan atípicos se deben a cortes de luz esporádicos cuyas causas quedan fuera del alcance de este estudio previo. Sucede igual con el máximo del mes de marzo, al no disponer de información para identificar su origen.

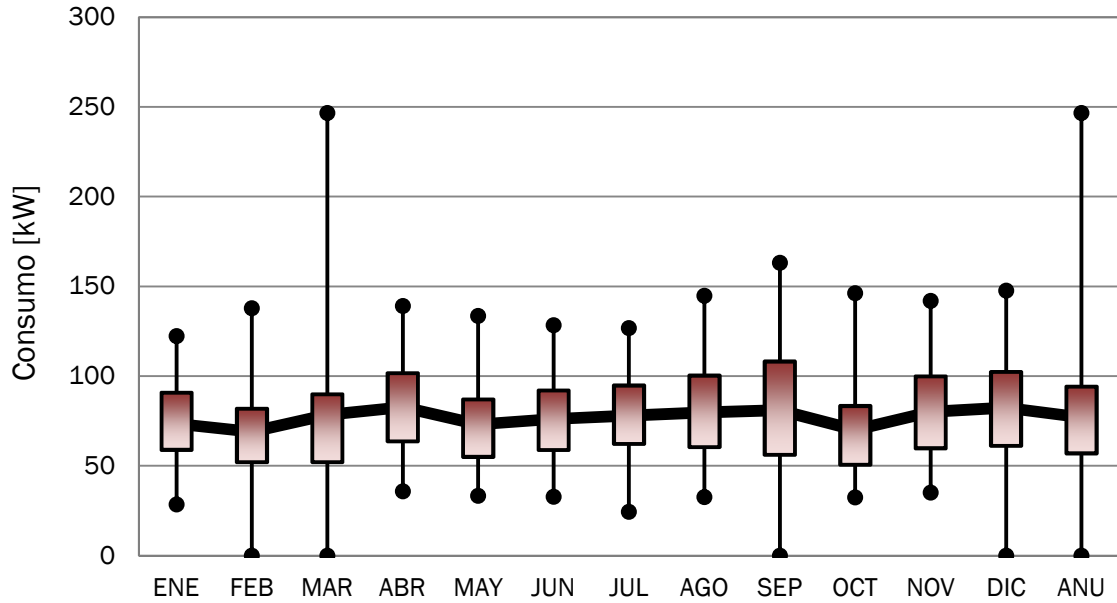


Fig. 16. Demanda de potencia activa anual

La diferencia entre los perfiles de generación y de demanda supone un problema. Ha quedado patente que la gran variabilidad inherente al recurso renovable entra en conflicto con la estabilidad que presenta el consumo a lo largo del año.

La Fig. 17 refleja esta situación al mostrar la evolución anual del flujo de potencia en el punto en el que la microrred se acopla a la red pública de distribución (*Point of Common Coupling*, PCC), tal y como muestra la ecuación (2); donde  $P_{dem}$  representa el consumo y  $P_{gen}$  la potencia generada dentro de la propia microrred. El flujo de potencia intercambiado con la red va a ser concepto fundamental en lo que resta de capítulo, puesto que las estrategias de control implementadas (apdo. 3.3.3), persiguen modificar  $P_{cc}$  en función de una serie de objetivos.

$$P_{cc} = P_{dem} - P_{gen} \quad (2)$$

Si no se tiene en cuenta la generación propia, se infiere de inmediato que  $P_{cc}$  será igual al consumo y no podrán producirse situaciones en los que la microrred ceda potencia activa a la red eléctrica. La distribución estadística de los valores del balance neto juega un papel decisivo a la hora de dimensionar correctamente los sistemas de almacenamiento de la microrred, así como para establecer algunos de sus parámetros de control.

Lo primero que llama la atención de la Fig. 17 es que el promedio anual de  $P_{cc}$  es negativo, es decir, se produce un aporte neto de energía a la red de distribución. La marcada asimetría de la distribución de  $P_{cc}$  deja entrever que la potencia renovable instalada fue pensada para consumos mayores, pues los bigotes que van desde el primer cuartil (Q1, la base de la caja) hasta el mínimo (el punto que cierra el bigote) son entre cinco y seis veces más largos que los bigotes superiores.

Es necesario tener presente la naturaleza de este tipo de gráfico para no dejarse engañar por las diferencias entre sus valores extremos. La longitud de los bigotes inferiores constata únicamente la existencia de fuertes picos de generación a lo largo del año, pero de ninguna manera deben ser tomados como único criterio a la hora de dimensionar las baterías. Diseñar el sistema de almacenaje de energía que se quiere integrar en la microrred para manejar todo el rango de potencias mostrado en la Fig. 17 implica una más que probable infrautilización de recursos durante la mayor parte del tiempo. Las baterías tienen un coste muy elevado y para rentabilizarlas adecuadamente se deben tomar en consideración más aspectos.

Si se presta atención a las cajas y a la línea horizontal que marca el promedio, se advierte que el intervalo de valores comprendido entre -100 kW y 80 kW engloba la mayor parte de las muestras. Esta concentración de valores será tomada en cuenta a la hora de escoger las prestaciones y el tamaño de las baterías.

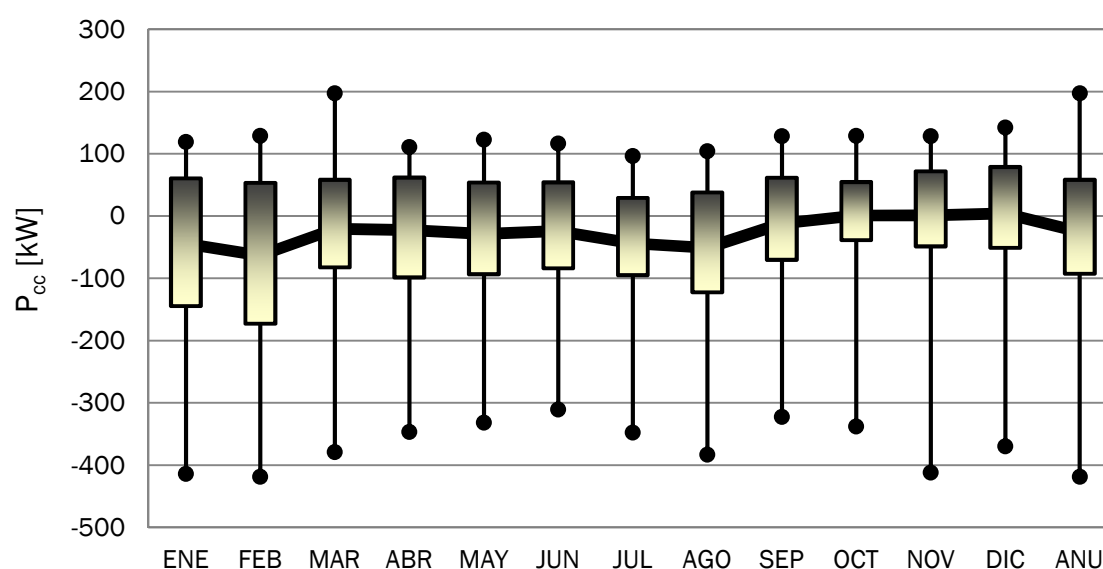


Fig. 17.  $P_{cc}$  anual, con generación eólica y fotovoltaica

### 3.3 Descripción del modelo

El esquema de la microrred a emular se muestra en la Fig. 18, donde aparecen tanto los elementos físicos como los agentes encargados de su control. Los agentes pueden clasificarse en tres grupos, dependiendo del elemento físico al que estén asociados: generación distribuida (**DG**), almacenamiento (**Storage**) y consumo (**Consumption**).

En total se han incluido cinco generadores, uno eólico y cuatro fotovoltaicos; cuatro cargas, y cuatro baterías. El modelo actual asume que todos los elementos se conectan a un mismo bus AC para llevar a cabo el intercambio de energía entre ellos. A su vez, este bus AC se conecta a través de una subestación transformadora con la red eléctrica, permitiendo la interacción con la misma. Las cargas consumen energía eléctrica en AC y se conectan directamente al bus, mientras que los generadores y las baterías necesitan de la intermediación de convertidores para poder conectarse.

Cada figura con sombrero representa a un agente. Hay un total de 14 agentes participantes: uno por cada generador, batería y carga; más el agente supervisor. La nube de la Fig. 18 representa la plataforma en la que los agentes existen e interactúan. Las

líneas que unen a los agentes con la plataforma acaban en flechas por ambos lados para remarcar la naturaleza bidireccional de la comunicación que llevan a cabo.

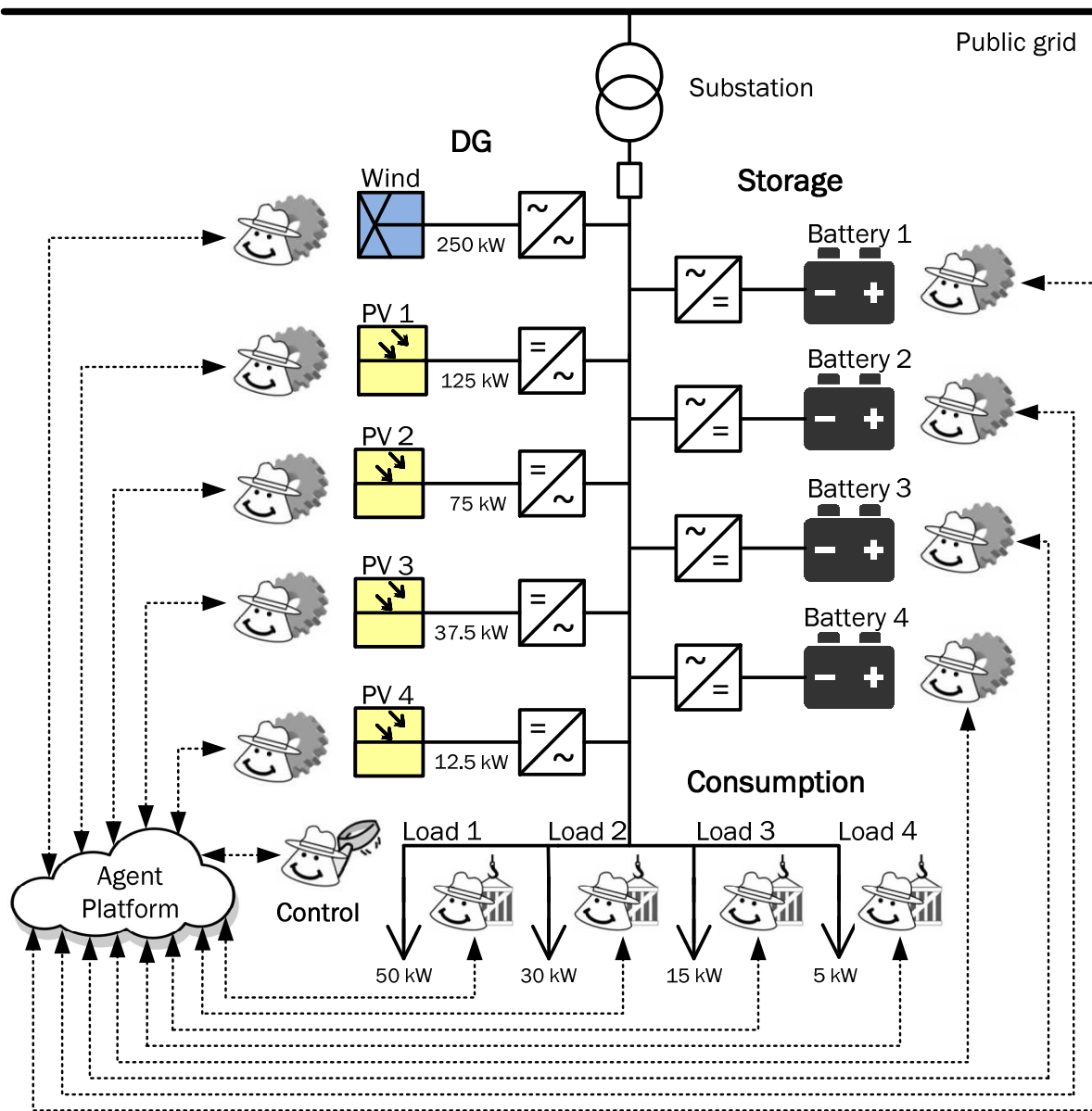


Fig. 18. Esquema de la microrred emulada

Cada uno de los 14 agentes es una instancia de alguna de las cuatro clases **Agent** implementadas para la ocasión (ver apdo. 2.4.3). La primera de estas clases es el agente **DG**, encargado de controlar un elemento de generación distribuida. El modelo cuenta con cinco generadores, uno eólico y cuatro fotovoltaicos, por lo que se crearán un total de cinco instancias de la clase **DG**.

Los agentes encargados del control de los sistemas de almacenamiento, baterías en este caso, son instancias del agente **Storage**. Tal y como puede verse a la derecha de la Fig. 18, se han creado cuatro instancias de la clase **Storage**: una para cada una de las cuatro baterías. El agente **Load** es el responsable del gobierno de las cargas, esto es, de la demanda. La microrred cuenta con cuatro cargas, por lo que también serán precisas

cuatro instancias del agente Load. El agente encargado de llevar a cabo las estrategias de control que serán presentadas en el apartado 3.3.3 se representa por medio de la figura con lupa, situado al lado de la nube, en la parte inferior izquierda.

La Fig. 19 presenta el modelo a un nivel de abstracción superior. El rectángulo central orientado en vertical representa el SMA. Las flechas indican el sentido del flujo de información. Los inputs del emulador (en modo *simulación*) se corresponden con los tres bloques de la izquierda, mientras que los outputs lo hacen en el bloque de la derecha. Los recuadros en color *beige* de la parte inferior representan los objetos que modelan convertidores y baterías respectivamente.

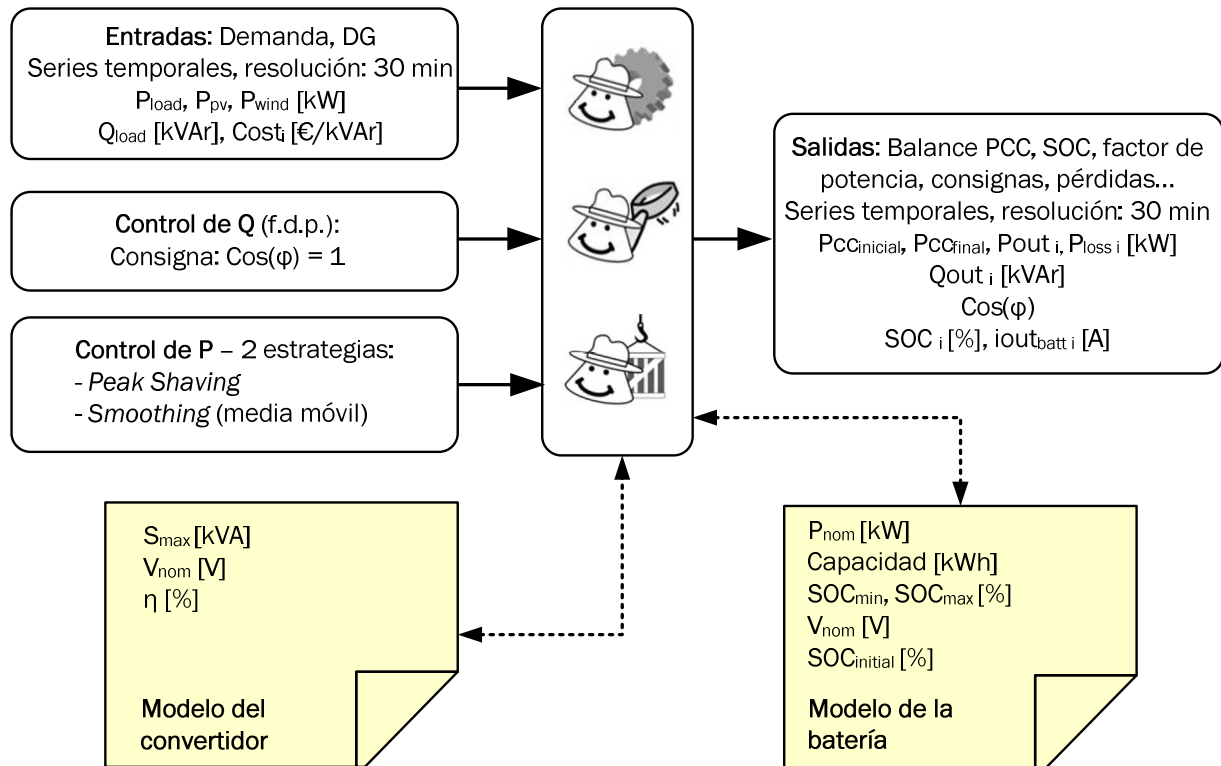


Fig. 19. Diagrama de funcionamiento del modelo

El bloque situado en la esquina superior izquierda contiene los datos de entrada del emulador: series temporales de potencia disponible, coste asociado a la generación de Q y potencia demandada. Los dos recuadros situados debajo de éste informan al SMA acerca de la política de control que se llevará a cabo.

Se ha priorizado la inyección de Q sobre el control de P, pero también podía haberse decidido lo contrario. Se ha establecido de forma arbitraria una consigna de  $\cos(\varphi)$  igual a la unidad (factor de potencia visto desde la red de distribución); lo que implica que el SMA tratará de copar con generación propia toda la demanda de potencia reactiva que pueda. Sólo en caso de no poder suministrarla de forma íntegra, se cubriría importando la Q restante de la red de distribución.

En lo que al control de P respecta, se le debe indicar al agente de control la estrategia de P a ejecutar. Por el momento se han implementado dos estrategias de control de P: *Peak Shaving* (reducción de picos) y *Smoothing* (alisamiento de la curva de demanda con media móvil).

En cada iteración, los agentes leerán sus inputs y se comunicarán posteriormente con otros agentes mediante mensajes ACL para tratar de cumplir con sus objetivos. Los agentes con capacidad de suministrar potencia neta a la microrred (generadores y baterías) cuentan con un convertidor asociado, es decir, con una instancia del objeto que modela el convertidor.

El agente debe conocer los principales parámetros de funcionamiento de su convertidor:  $S_{\max}$  y  $\eta$ , ya que son necesarios para calcular de manera correcta las consignas.  $S_{\max}$  y  $\eta$  definen la potencia que se perderá en el convertidor y el límite superior de potencia aparente que éste podrá manejar. Los agentes asociados a elementos de almacenamiento poseen, además del convertidor asociado, de una instancia del modelo de la batería.

Los datos de salida generados por el SMA son también series temporales treintaminutales. Por citar algunas de las más importantes: flujo de potencia en PCC, antes y después de la actuación de las baterías, f.d.p. visto desde el lado de red y el SOC de las baterías.

### 3.3.1 Datos de entrada y de salida

La validación del emulador se ha llevado a cabo para el modo *simulación*. En este modo de funcionamiento, tanto los inputs como los outputs son archivos que contienen series temporales. El formato de estos archivos ha sido escogido para facilitar su procesamiento con una hoja de cálculo, en este caso Microsoft Excel.

#### 3.3.1.1 Inputs

El primer bloque de entrada de la Fig. 19 corresponde a inputs de carácter local. Un agente asociado a un generador o a una carga (funcionando en modo *simulación*) obtendrá los datos que necesita de un archivo con formato de Libro de Excel (.xlsx). La microrred implementada cuenta con cinco agentes DG y cuatro agentes Load y, por lo tanto, harán falta un total de nueve archivos .xlsx.

En vista de la gran cantidad de archivos que se van a manejar, podría resultar tentador tratar de agruparlos todos en único Libro de Excel, lo que no sería para nada adecuado puesto que se está trabajando con un SMA. En un SMA, un agente puede estar ejecutándose a gran distancia física del resto y que su comunicación sea, por ejemplo, vía Ethernet. Sólo si cada agente cuenta con su propia fuente de datos totalmente independiente podrá preservarse el carácter distribuido del emulador y la modularidad del sistema.

El agente DG instalado en el generador eólico usará directamente la serie temporal de recurso eólico a modo de input (250 kW), mientras que los cuatro elementos de generación fotovoltaica deberán repartirse los valores de la serie temporal de recurso fotovoltaico (también de 250 kW), de forma que cada uno cuente con su propio archivo .xlsx. Con las cuatro cargas ocurre lo mismo: necesitarán compartir la serie temporal de demanda, dividiéndola arbitrariamente para crear cuatro archivos .xlsx independientes.

Se ha decidido fraccionar los valores de las dos series temporales originales siguiendo una proporción 50%, 30%, 15% y 5%. Es decir, si el consumo el 14 de Abril a las 17:30 es de 60 kW y 27 kVAr, la primera carga consumirá 30 kW y 13.5 kVAr, la segunda 18 kW y 8.1 kVAr, la tercera 9 kW y 4.05 kVAr, y la cuarta 3 kW y 1.35 kVAr. El valor de recurso fotovoltaico disponible por cada uno de los cuatro generadores se calcula de idéntica manera.

Los agentes DG interpretan los datos de los archivos .xlsx tal y como se muestra en la Tabla 3, mientras que los agentes Load lo hacen siguiendo la estructura de la Tabla 4.

Columna	Variable	Descripción
1	t	Tiempo de simulación [s]
2	$P_{\text{wind/pv}}$	Potencia extraíble del recurso [kW]
3	Cost	Factor de coste de producción de Q [€/kVAr]

Tabla 3. Archivo de inputs: generador

Columna	Variable	Descripción
1	t	Tiempo de simulación [s]
2	$P_{\text{load}}$	Potencia activa consumida [kW]
3	$Q_{\text{load}}$	Potencia reactiva consumida [kVAr]

Tabla 4. Archivo de inputs: carga

### 3.3.1.2 Outputs

Los agentes DG, Storage y PowerManager (agente de control) generan sus outputs en ficheros .csv (comma-separated values). El formato CSV es muy sencillo y se usa para organizar los datos en forma de tabla. La separación entre columnas viene delimitada por un punto y coma, mientras que las filas se separan por saltos de línea. Si se quiere abrir el fichero con una hoja de cálculo, bastará con informar al programa acerca de estas dos cuestiones.

El fichero de outputs más relevante lo genera el agente de control y se organiza según se indica en la Tabla 5. Este fichero recoge las interacciones de la microrred con la red de distribución.

Columna	Variable	Descripción
1	t	Tiempo de simulación [s]
2	$P_{\text{cc}}$ (inicial)	Flujo inicial de P en PCC [kW]
3	$P_{\text{cc}}$ (final)	Flujo final de P en PCC [kW]
4	$Q_{\text{cc}}$	Potencia reactiva obtenida de red [kVAr]
5	$\text{Cos}(\varphi)$	Factor de potencia visto desde la red
6	SOC	Estado de carga global de las baterías [%]

Tabla 5. Archivo de outputs: control

El control implementado modifica el flujo de potencia inicial en PCC (3) gracias a la actuación del sistema de almacenamiento. Dependiendo de la situación, las baterías cederán o absorberán potencia de la microrred para ajustar el valor de  $P_{\text{cc}}$  al deseado (4).

La variable  $P_{\text{dem}}$  representa la potencia consumida, mientras que  $P_{\text{gen}}$  representa potencia generada dentro de la propa microrred. La potencia cedida (valores positivos) o absorbida (negativos) por el conjunto de las baterías  $P_{\text{bat}}$ .

$$P_{cc}(inicial) = P_{dem} - P_{gen} \quad (3)$$

$$P_{cc}(final) = P_{dem} - P_{gen} - P_{bat} \quad (4)$$

La cuarta columna del fichero guarda los valores de potencia reactiva que la microrred importa de la red (la Q que no ha podido cubrir por sí sola). Se calcula con la ecuación (5), cuyos subíndices son idénticos a los de (3) y (4).

$$Q_{cc} = Q_{dem} - Q_{gen} \quad (5)$$

El factor de potencia o  $\cos(\varphi)$  de la columna cinco se obtiene de forma directa conociendo los valores de P y Q en el punto de conexión a red (columnas tres y cuatro).

Por último, el SOC recogido en la columna seis corresponde a la media aritmética de los SOC de las n baterías que componen la microrred.

$$SOC_{global} = \frac{1}{n} \sum_{i=1}^n SOC_i \quad (6)$$

Los archivos que producen los agentes instalados en un generador (agentes DG) siguen la estructura de la Tabla 6 y almacenan los valores de P y Q que llegan finalmente al bus AC de la microrred, así como la potencia pérdida en el proceso de transformación que se lleva a cabo en el convertidor asociado.

Columna	Variable	Descripción
1	t	Tiempo de simulación [s]
2	P <sub>out</sub>	Potencia activa entregada a la microrred [kW]
3	Q <sub>out</sub>	Potencia reactiva entregada a la microrred [kVAr]
4	P <sub>losses</sub>	Pérdidas en el convertidor [kW]

Tabla 6. Archivo de outputs: generador

La Tabla 7 contiene las salidas de los agentes Storage, es decir, los agentes instalados en las baterías. Los agentes Storage cuentan con dos outputs adicionales respecto a los agentes DG: el valor de SOC y el de la corriente de salida. A diferencia de un generador, el valor de P<sub>out</sub> podrá tomar valores negativos. Valores positivos de P<sub>out</sub> e i<sub>out</sub> suponen cesión de potencia al bus de AC de la microrred y valores negativos, absorción.

Columna	Variable	Descripción
1	t	Tiempo de simulación [s]
2	P <sub>out</sub>	Potencia activa entregada a la microrred [kW]
3	Q <sub>out</sub>	Potencia reactiva entregada a la microrred [kVAr]
4	P <sub>losses</sub>	Pérdidas en el convertidor [kW]
5	SOC	Estado de carga [%]
6	i <sub>out</sub>	Corriente de salida [A]

Tabla 7. Archivo de outputs: batería



### 3.3.2 Casos de estudio

Para validar el modelo se ha llevado a cabo un total de 20 simulaciones, ocho para la estrategia de *Peak Shaving* y doce para la de *Smoothing*. Con objeto de evaluar el impacto que suponen en el modelo los parámetros que dimensionan el SAE, se han considerado dos casos.

El primer caso (caso A) establece una potencia nominal de 100 kW y una capacidad de 1 MWh, mientras que en el segundo (caso B) ambos parámetros han visto duplicado su valor. Como la microrred a emular cuenta con cuatro baterías, los valores de potencia y capacidad totales se han dividido por cuatro. Por ejemplo, para el caso A, cada una de las cuatro baterías tendrá una potencia nominal de 25 kW y una capacidad de 250 kWh.

- Caso A: 100 kW / 1MWh → batería de 25 kW / 250 kWh (x4)
- Caso B: 200 kW / 2MWh → batería de 50 kW / 500 kWh (x4)

Por cada uno de los dos supuestos, se han tomado en consideración cuatro situaciones en relación con la generación renovable distribuida: 1) sin DG, 2) sólo generación eólica, 3) sólo generación fotovoltaica, y 4) eólica + fotovoltaica. Se tienen por lo tanto dos casos para cada una de las dos estrategias, en los que se consideran cuatro situaciones.

Nótese que el tiempo de descarga de las baterías no varía de un caso a otro. Tanto en el caso A como en el caso B, las baterías se descargan por completo (suponiendo potencia nominal) en 10 horas.

### 3.3.3 Estrategias de control de P (potencia activa)

Se han implementado dos estrategias de control de potencia activa en la microrred: *Peak Shaving* y *Smoothing*. El diseño realizado permite añadir nuevas estrategias o funciones de optimización sin tener que hacer modificaciones en el código.

El agente *PowerManager*, máxima autoridad de la microrred, es el responsable de controlar el balance de P y Q. *PowerManager* aparece representado en la Fig. 20 por el recuadro blanco situado dentro del SMA (la nube de color morado).

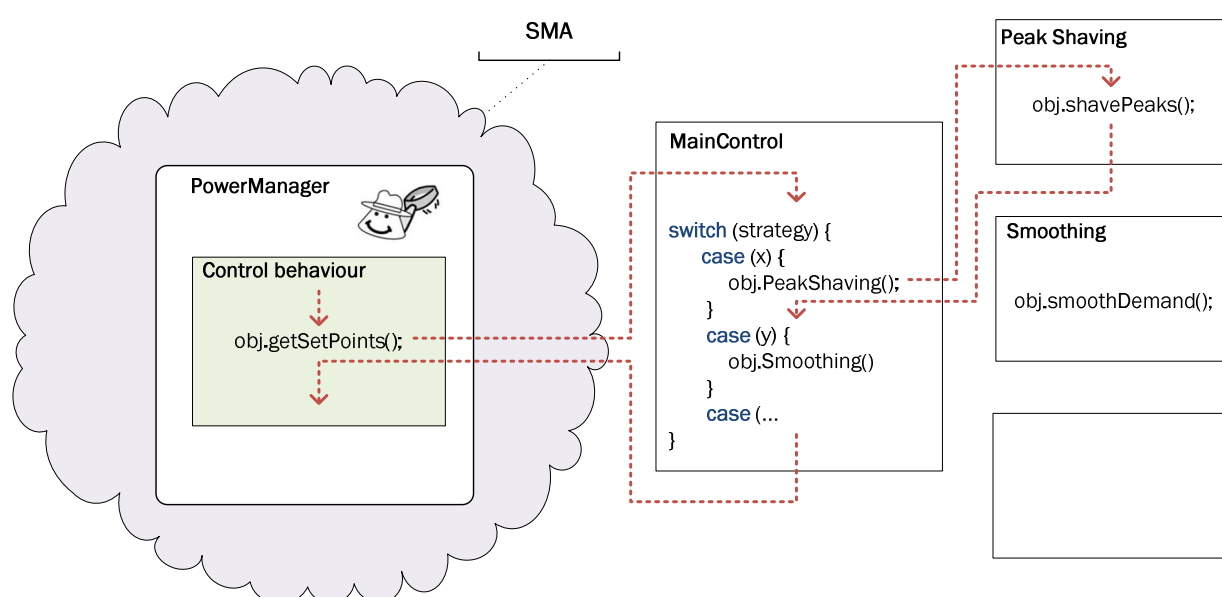


Fig. 20. Ontología de control P y Q

Tal y como se ha explicado en el apartado 2.4.4, un agente lleva a cabo sus funciones a través de comportamientos. **PowerManager** dispone de un comportamiento dedicado en exclusiva al control. Dentro de este comportamiento se realiza una llamada al método `getSetPoints()`, al que se le pasa como parámetros tres vectores con toda la información necesaria de los generadores, baterías y cargas. El método `getSetPoints()` debe retornar un vector con las consignas que **PowerManager** enviará al resto de agentes participantes.

Esta forma de operar ofrece mucha flexibilidad, ya que permite el empleo de cualquier control *ad hoc*, siempre y cuando éste retorne dicho vector. La manera de calcular estas consignas le es completamente irrelevante al agente y sólo es necesario indicarle cuál de las clases que tiene disponibles le corresponde utilizar para ello.

El método `getSetPoints()` pertenece a la clase **MainControl**. A la hora de establecer la actuación de las baterías, **MainControl** solicitará ayuda a la clase que contenga la estrategia de P a ejecutar en ese momento. Esto ofrece un segundo nivel de modularidad, posibilitando un encaje muy directo de cualquier estrategia de control de las baterías que se quiera plantear.

### 3.3.3.1 Peak Shaving

El objetivo de esta estrategia de control de P es reducir los picos de demanda que sobrepasen un valor límite. La microrred del modelo cuenta con una gran potencia de generación instalada, con lo que será habitual encontrar valores negativos de  $P_{cc}$  (la microrred cede energía a la red). Independientemente de su signo, las líneas tienen unos límites de potencia que son capaces de conducir sin resultar dañados, por lo que sobrepasar dichos límites puede ser problemático.

La estrategia *Peak Shaving* se vale del SAE para procurar que esto no ocurra. Para ello se fijan dos límites, uno superior y otro inferior, que establecen la máxima potencia que la microrred puede intercambiar con la red eléctrica. En este caso concreto, tras el estudio llevado a cabo en el apartado 3.2, se ha decidido establecer los valores que aparecen en la Tabla 8.

Parámetro	No DG	Eólica 250 kW	FV 250 kW	Eólica + FV 500 kW
$P_{max}$ [kW]	100	100	100	100
$P_{min}$ [kW]	-	-100	-100	-100

Tabla 8. Parámetros de control: Peak Shaving

Se ha optado por mantener los límites constantes independientemente de la generación instalada para centrar la atención del análisis en el impacto que los parámetros dimensionales del SAE tienen en el modelo. Si se quisiera trasladar el control a una microrred real sí sería necesario ajustar estos parámetros en función de la potencia renovable que se hubiera instalado.

La mecánica de funcionamiento es muy sencilla. Por poner un ejemplo: si se detecta que la microrred está demandando 120 kW de la red, se dará orden a las baterías para que cedan 20 kW; manteniendo así el valor de  $P_{cc}$  entre los límites marcados de  $\pm 100$  kW.

En ocasiones puede ocurrir que las baterías se encuentren tan descargadas que no dispongan de los 20 kW que hacen falta inyectar al bus AC durante todo el periodo de tiempo considerado (30 minutos). Para tratar de evitar esta situación en la medida de lo posible se ha ideado una rutina de carga/descarga automática de las baterías.

La rutina automática de carga sólo entrará en funcionamiento cuando se den a la vez las siguientes condiciones:

- 1) No se sobrepasan los límites de potencia de la Tabla 8.
- 2) La batería se encuentra fuera del rango óptimo de SOC de la Tabla 9.

Parámetro	No DG	Eólica 250 kW	FV 250 kW	Eólica + FV 500 kW
SOC <sub>lower</sub> [%]	90	40	40	30
SOC <sub>upper</sub> [%]	100	50	50	40

Tabla 9. Parámetros de carga/descarga automática: Peak Shaving

El rango óptimo de SOC escogido varía de una situación a otra. En ausencia de DG, por ejemplo, los únicos picos que pueden producirse son los de consumo (positivos). Para recortar dichos picos las baterías ceden potencia a la red. De no existir una rutina automática de recarga, las baterías acabarían descargándose por completo, lo que imposibilitaría seguir aplicando el *Peak Shaving*.

Mantener el SOC a un nivel predeterminado proporciona a la batería un margen de actuación razonable, tanto de carga como de descarga. La ecuación (7) sirve para, una vez activada la rutina automática, calcular las consignas de descarga de las baterías.

$$P_{desc} = \frac{1}{2} * \left[ p_{batt} + p_{batt} * \left( \frac{SOC - SOC_{upper}}{SOC_{max} - SOC_{upper}} \right) \right] \quad (7)$$

La ecuación (8) es su equivalente para la carga.

$$P_{carg} = \frac{1}{2} * \left[ p_{batt} + p_{batt} * \left( \frac{SOC_{lower} - SOC}{SOC_{lower} - SOC_{min}} \right) \right] \quad (8)$$

La variable  $p_{batt}$  es la potencia máxima que puede ceder o absorber la batería. Como la rutina automática se suele activar cuando el SOC se encuentra en valores intermedios,  $p_{batt}$  será igual a la potencia nominal del SAE en cuestión. SOC<sub>lower</sub> y SOC<sub>upper</sub> son las variables que aparecen en la Tabla 9, mientras que SOC<sub>max</sub> y SOC<sub>min</sub> son los valores máximos y mínimos de carga permitidos. SOC es el estado de carga actual de la batería al inicio del periodo.

Con estas ecuaciones se consigue que la recarga (y descarga) automática sea proporcional a la diferencia entre el SOC de la batería y el valor de SOC deseado. Cuanto más alejado se encuentre de este valor, más rápido se cargará. La Fig. 21 muestra la evolución de las consignas en función de  $\Delta SOC$ , considerando una  $p_{batt}$  de 50 kW, un SOC<sub>lower</sub> de 40% y un SOC<sub>min</sub> del 5%.

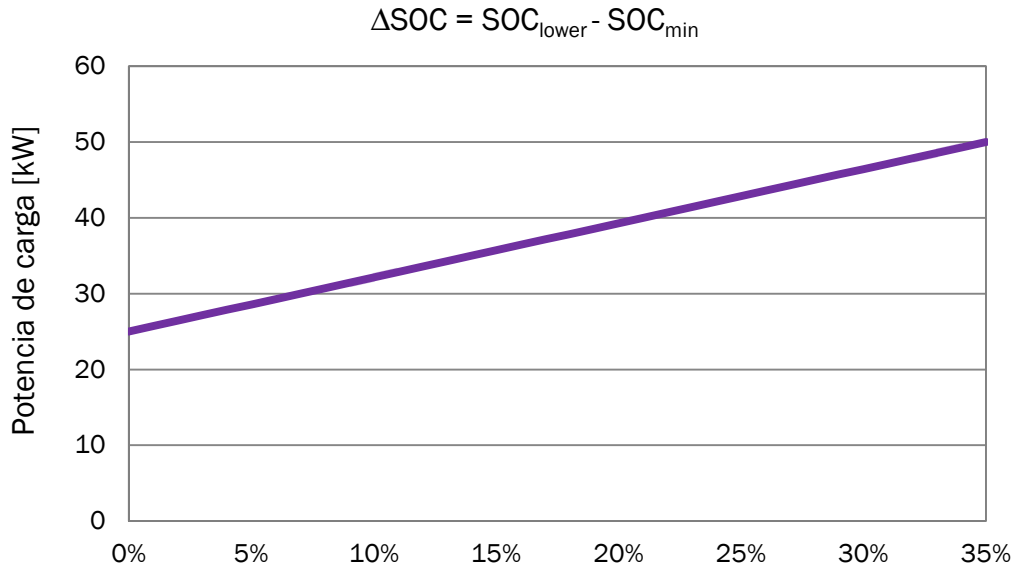


Fig. 21. Rutina de carga automática

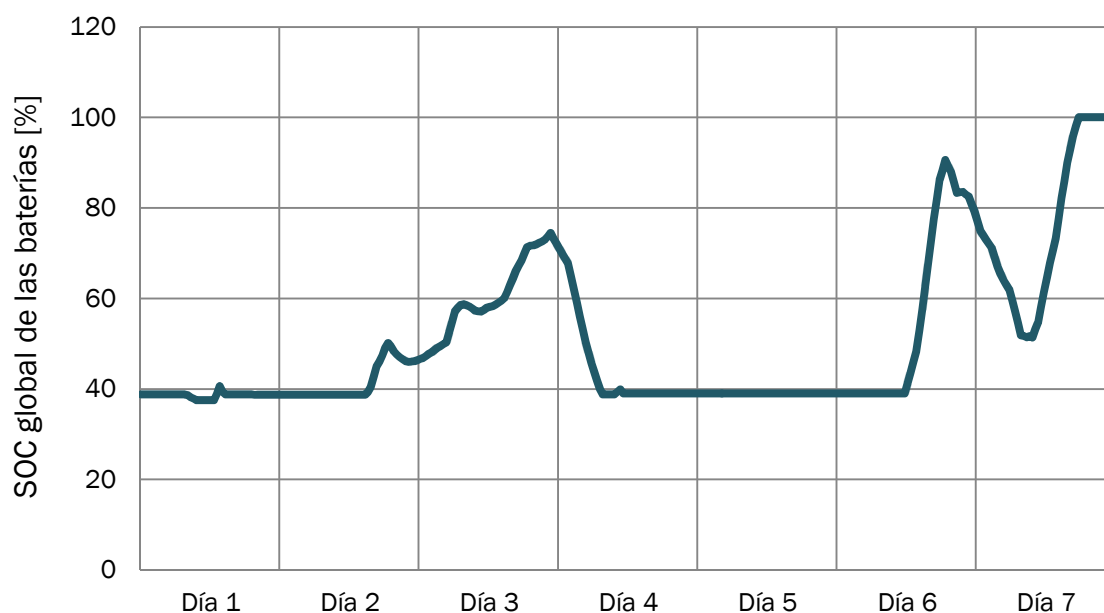
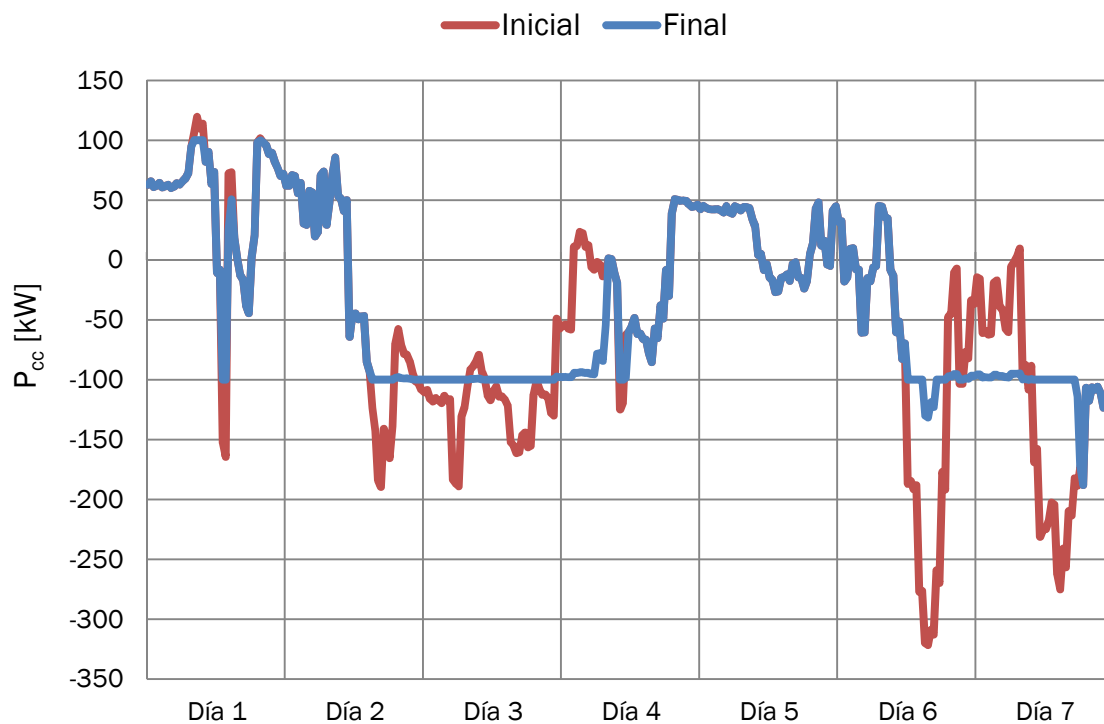
El valor máximo de  $P$  se da con el máximo  $\Delta SOC$ , es decir, cuando la batería se encuentra a su mínimo valor posible de SOC (en este caso el mínimo permitido es del 5%). Se tiene entonces que  $\Delta SOC = 40 - 5 = 35\%$ , a lo que le corresponde una potencia de carga igual a la nominal: 50 kW.

Antes de enviar las órdenes con los valores de carga automática a sus destinatarios, los agentes *Storage*, la estrategia se asegura de que no se estén superando los límites  $P_{max}$  y  $P_{min}$  propios de la estrategia *Peak Shaving*. Si por causa de esta rutina se superara alguno de estos dos valores, el control truncaría la consigna calculada por las ecuaciones (7) y (8) hasta un valor tal que evite que sean sobrepasados. Por lo tanto, se supedita mantener el SOC en el intervalo deseado a la estrategia principal: evitar los picos de demanda.

En la Fig. 22 se pueden observar los resultados conseguidos por la estrategia *Peak Shaving* durante la octava semana del año (febrero-marzo) en un escenario con DG eólica y fotovoltaica. La línea roja representa el valor original de  $P_{cc}$ , es decir, el que tendría lugar en ausencia de baterías; la línea en color azul representa el flujo de potencia en PCC tras la actuación del *Peak Shaving*. Se ha elegido esta semana en concreto porque se producen varias situaciones que ayudarán a entender mejor el funcionamiento de la estrategia.

Lo primero que se debe comprobar es si la estrategia *Peak Shaving* cumple o no con su razón de ser: recortar los picos. Un vistazo rápido basta para constatar que así es. Si se presta atención al día 1, se puede advertir un pico de demanda que ha sido recortado satisfactoriamente. En los días 2 y 3 se produce una serie de picos de generación que también son recortados con éxito. Por el contrario durante el día 6 tiene lugar un aumento repentino de la generación, que no puede ser contrarrestado con el aporte de las baterías.

La potencia necesaria para recortar el pico del día 6 es superior a la máxima que las baterías son capaces de suministrar y se superan los -100 kW establecidos como límite (Tabla 8). A pesar de esta superación, el pico se atenúa considerablemente. En el día 7 se produce otro pico de generación no contrarrestado, pero esta vez la causa tiene su origen en el SOC de las baterías, cuya evolución aparece en la Fig. 23 y será explicada a continuación.



La Fig. 23 presenta la evolución del SOC global del conjunto de baterías que forman la microrred y va a servir para explicar dos situaciones que se dan en la Fig. 22. La primera de ellas tiene que ver con la rutina automática de carga y descarga. Durante los días 2 y 3 se producen varios picos de generación que son recortados a -100 kW. Sin embargo, cuando  $P_{cc}$  (en rojo) vuelve a traspasar el límite inferior, las baterías lo siguen manteniendo a -100 kW (línea azul).

Este modo de actuar se debe a la rutina automática de carga, que tiene órdenes de mantener el SOC entre el 30 y el 40%. En efecto, a lo largo de los días 2 y 3 el SOC de la

batería fluctúa entre valores que están fuera del rango deseado. La estrategia aprovechará los periodos en los que la generación sea más moderada para vaciar sus baterías hasta el 40%, lo que le permitirá estar preparada para absorber los picos venideros, tanto de generación como de demanda. Lo mismo sucede al final del sexto día y principios del séptimo: las baterías mantienen el balance neto a -100 kW al descargarse de forma automática.

La última situación de interés ocurre el último día de la semana, donde se produce un pico de generación que no puede ser ni siquiera atenuado. En este caso las baterías han alcanzado un 100% de SOC y ya no admiten más energía entrante, por lo que la curva de  $P_{cc}$  final coincide con la del inicial. Este análisis pone en relieve la importancia de dimensionar correctamente el sistema, así como de establecer unos valores adecuados de SOC deseado.

### 3.3.3.2 Smoothing (alisamiento de la curva de demanda)

El objetivo de esta estrategia de control de P es suavizar cambios bruscos y picos de corta duración que se producen en la curva de demanda. La consigna de P se calcula mediante una sencilla media móvil con los últimos  $\omega$  valores de  $P_{cc}$  conocidos (9).

$$P_{target} = \frac{1}{\omega} \sum_{i=1}^{\omega} P_i \quad (9)$$

El resultado de esta estrategia se muestra en la Fig. 24, donde vuelve a aparecer el valor de  $P_{cc}$  pre y post estrategia. La línea en rojo se corresponde con la potencia que la microrred intercambiaría con la red de no existir intervención del SAE, es decir, la curva de consumo menos la de generación. Al igual que para el *Peak Shaving*, se ha escogido la octava semana del año con objeto de poder comparar más fácilmente ambas estrategias.

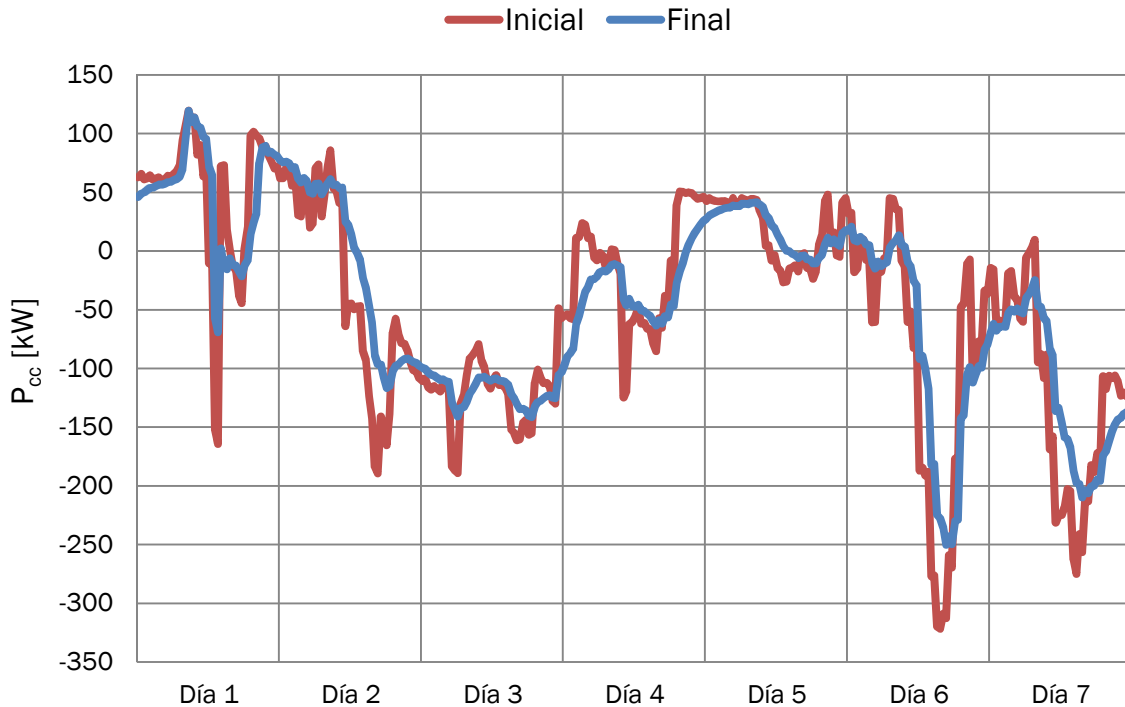


Fig. 24. Smoothing: 8ª semana ( $\omega=4$ , 1MWh)

El número de valores utilizados para calcular la media móvil en el ejemplo de la Fig. 24 ha sido de cuatro ( $\omega = 4$ ). El parámetro  $\omega$  causa gran impacto en los resultados de la estrategia. Una  $\omega$  pequeña provoca un escaso alisamiento de la curva: cuando la estrategia dispone de pocos valores de referencia para el cálculo de la media móvil, sólo se ven modificados los picos de más corta duración; lo que no tiene por qué suponer ningún problema *per se*. De ser ése el objetivo perseguido, no será necesaria una gran inversión en baterías.

Cuanto menor sea  $\omega$ , menor será la necesidad de disponer de un SAE con grandes prestaciones. Por el contrario, valores grandes de  $\omega$  requieren disponer de un suministro de potencia sostenido en el tiempo y, por lo tanto, de un sistema de baterías con mayor capacidad y más caro. En el apartado 3.5 se expondrán los resultados obtenidos para un *Smoothing* con  $\omega = 4$  y  $\omega = 8$ .

La Fig. 25 presenta la evolución del estado de carga del conjunto de baterías correspondiente a la Fig. 24. La estrategia *Smoothing* no necesita de ninguna rutina de carga automática: la propia tendencia de la demanda es la que va cargando y descargando las baterías.

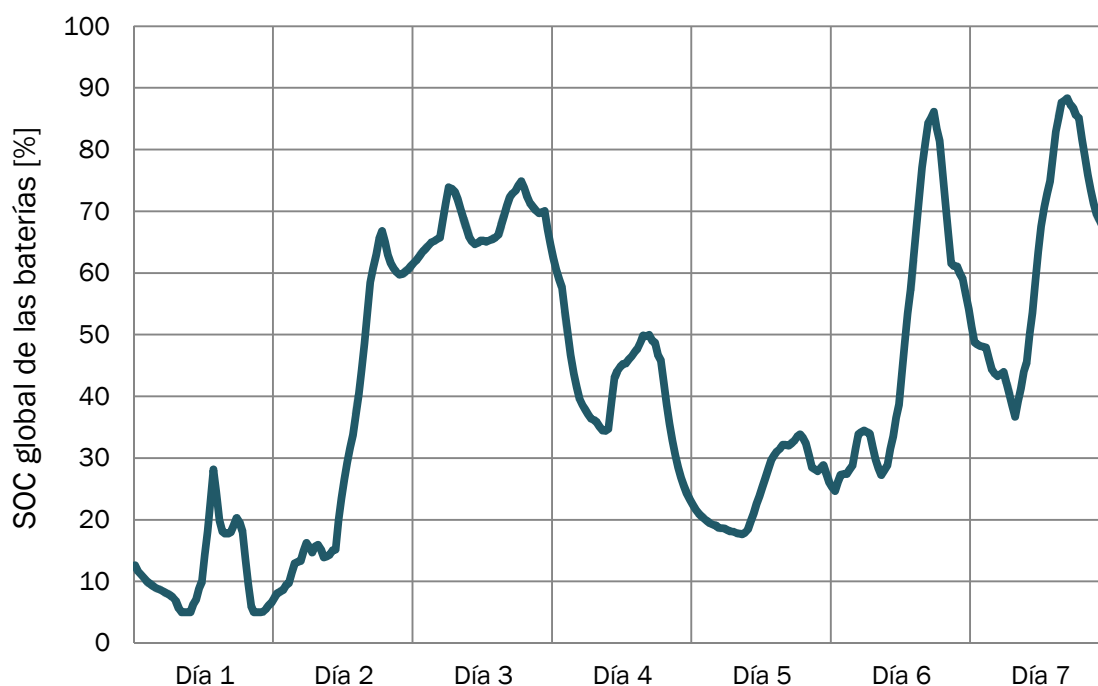


Fig. 25. Evolución del SOC de las baterías: *Smoothing* ( $\omega=4$ , 1MWh)

Se observa que la estrategia *Smoothing* hace un uso más intensivo del SAE que la estrategia *Peak Shaving* (Fig. 23). En este caso no existen periodos de SOC constante, o lo que es lo mismo, periodos en los que la batería permanece inactiva. Alisar la curva de demanda supone un trabajo ininterrumpido ya que la consigna de  $P_{cc}$  se calcula con una media móvil que no para de variar.

Si las variaciones de la curva de demanda guardan cierta simetría, las positivas (picos de demanda) descargarán la batería en la misma medida que la recargan las negativas (picos de generación). El *Smoothing* es por tanto una estrategia fácil de implementar, a la par que efectiva.

Duplicar  $\omega$  y modificar la capacidad del SAE de manera proporcional se consigue un alisado aún mayor de la curva de demanda. El valor de  $P_{cc}$  final de la Fig. 26 (en azul) varía mucho más lentamente que en el caso de la Fig. 24, y obvia la mayor parte de los picos que tienen lugar en la línea del  $P_{cc}$  inicial (en rojo).

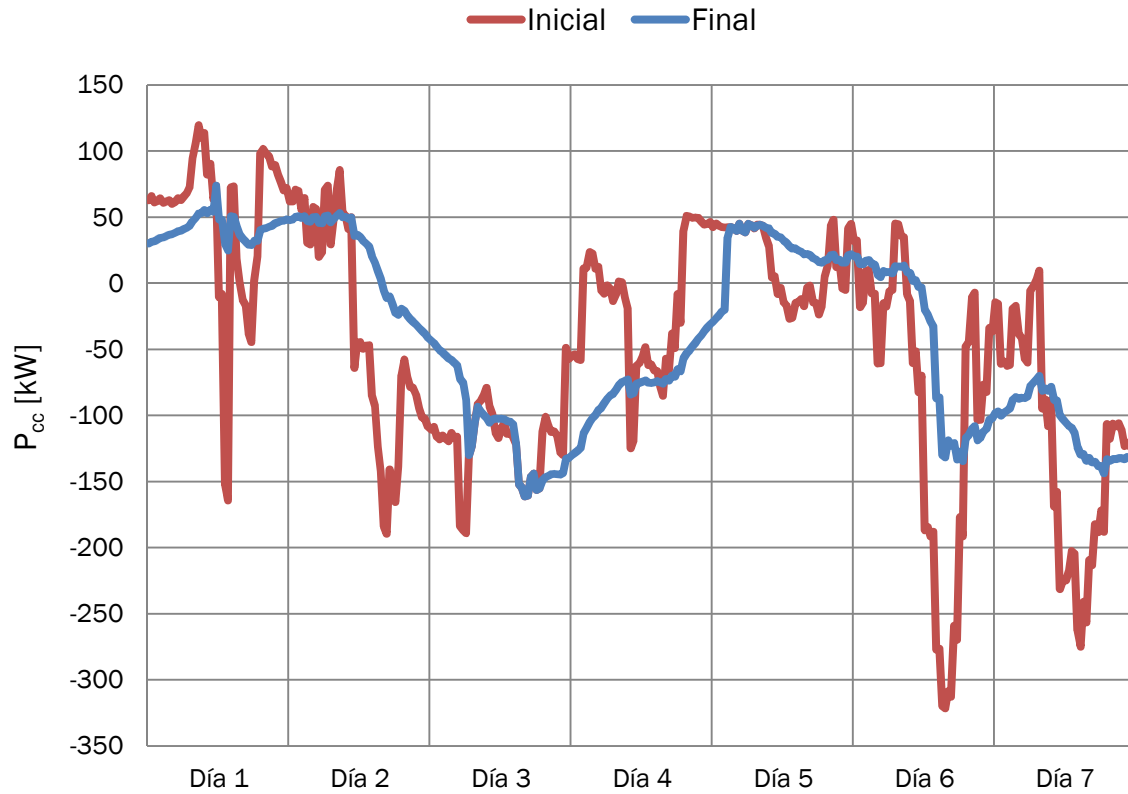


Fig. 26. Smoothing: 8ª semana ( $\omega=8$ , 2MWh)



Fig. 27. Evolución del SOC de las baterías: Smoothing ( $\omega=8$ , 2MWh)



La gráfica de la Fig. 27 constata nuevamente la carga de trabajo de las baterías. Los únicos periodos en los que no se registra actividad se corresponden a instantes de máxima y mínima carga (100 y 5%). El resto del tiempo la batería suplirá o absorberá potencia para cumplir con la consigna de la ecuación (9).

Por ejemplo, durante el pico de generación de los días 2 y 3 las baterías tratan de absorber el excedente cargándose por completo. Cuando esto ocurre la media móvil confluye rápidamente con el  $P_{cc}$  actual y, en el momento en que se invierte la tendencia, las baterías disponen de un 100% de SOC para actuar.

Una media móvil de ocho valores provoca cierta “inercia” o reticencia a los cambios en la curva de demanda, pero sólo si es respaldada por un SAE que disponga de energía suficiente para compensar desviaciones durante grandes lapsos de tiempo.

### 3.3.4 Control de Q (potencia reactiva)

El control de potencia reactiva es independiente de las estrategias de control de P. Se lleva a cabo dentro de la clase `MainControl` (ver Fig. 20) y sigue una máxima muy simple: tratar de copar la máxima demanda de potencia reactiva que sea posible con los recursos de los que dispone la propia microrred, lo que equivale a establecer una consigna de  $\cos(\varphi) = 1$ .

La potencia reactiva es inyectada por los convertidores asociados a la DG, en los cuales Q tiene prioridad sobre la P. Los convertidores de las baterías también pueden inyectar Q, pero en este caso se da preferencia a P, lo que en la práctica implica que todo el peso del control de Q recaiga sobre los elementos de generación.

Cada generador tiene asociado un factor de coste, que no es más que un input, y como tal, puede variar de un periodo a otro (ver apdo. 3.3.1.1). El control tratará de cubrir la demanda de reactiva al mínimo coste. Para ello utilizará primero la Q que oferta el generador con el menor coste asociado: la Q más barata. Si esto no fuera suficiente, seguirá con la segunda más barata, y así sucesivamente. Si se logra copar toda la demanda de reactiva de manera interna no se tendrá que importar Q de la red, por lo que el factor de potencia visto desde ésta será igual a la unidad.

Para poder satisfacer la demanda de potencia reactiva al mínimo coste el agente de control (agente `PowerManager`) necesita saber cuánta Q le puede encargar a cada uno de los agentes DG y a qué precio. Como cualquier agente en un SMA, `PowerManager` no dispone de información global del sistema y se va a ver obligado a comunicarse con otros agentes para recabarla.

`PowerManager` solicitará dicha información enviando un mensaje ACL en el marco del protocolo de comunicación FIPA-ContractNet, que será explicado en el apartado 3.4.2. Los agentes DG responderán a esta solicitud mediante otro mensaje ACL en el que se especificarán la potencia disponible ( $P_{in}$ ), el coste asociado a la inyección de Q, la potencia aparente nominal ( $S_{max}$ ) y el rendimiento ( $\eta$ ) del convertidos asociado.

Ahora que `PowerManager` conoce el estado y disponibilidad de cada agente suministrador de Q ya está en condiciones de realizar el cálculo de consignas para el periodo en curso mediante la secuencia de operaciones que se va a detallar a continuación.

La eficiencia se aplica tanto a P como a Q, o lo que es lo mismo, a S, de forma que:

$$S_{out} = S_{in} * Eff \quad (10)$$

Lo que implica que:

$$P_{losses} = S_{out} * Eff \quad (11)$$

La potencia de los generadores DG de la microrred proviene íntegramente del recurso renovable disponible en ese momento ( $P_{in}$ ), por lo que toda la potencia extraída o disipada en el conjunto generador-convertidor provendrá de ahí, tal y como establece la ecuación (12):

$$P_{in} \geq P_{out} + P_{loss}(P) + P_{loss}(Q) \quad (12)$$

$$P_{loss}(Q) = f(Q) \rightarrow P_{loss}(Q_{max}) \leq P_{in} \quad (13)$$

Los parámetros anteriormente mencionados, junto con la restricción de la ecuación (12) permiten obtener la Q máxima que el agente DG podrá suministrar durante un determinado periodo (Fig. 28).

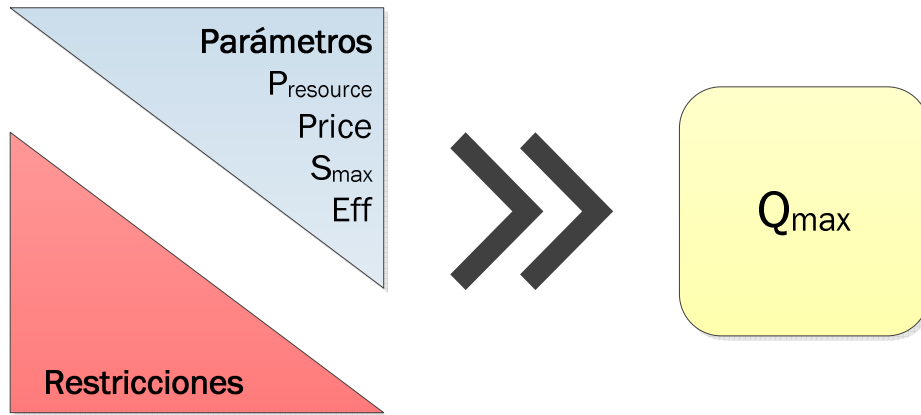


Fig. 28. Oferta de Q por parte de los agentes DG (obtención de  $Q_{max}$ )

En la Fig. 29 se ha tratado de esquematizar la mecánica del control. Una vez se han obtenido los  $n$  valores de Q máximos para cada uno de los  $n$  agentes, PowerManager los ordenará de menor a mayor precio, y comenzará entonces a adjudicar las consignas de Q.

PowerManager realizará  $n$  iteraciones (tantas como agentes que oferten Q) e irá restando la Q adjudicada ( $Q_{max\ i}$ ) de la Q total de consumo ( $Q_{dem}$ ). Si dicha la resta llega a cero, significará que se ha cubierto por completo la demanda de Q y el Cos( $\phi$ ) visto por la red será igual a la unidad ( $Q_{out} = 0$ ). Por el contrario, si después de las  $n$  iteraciones aún no se ha cubierto todo el consumo de Q, se deberá importar lo que reste de la red ( $Q_{out} \neq 0$ ).

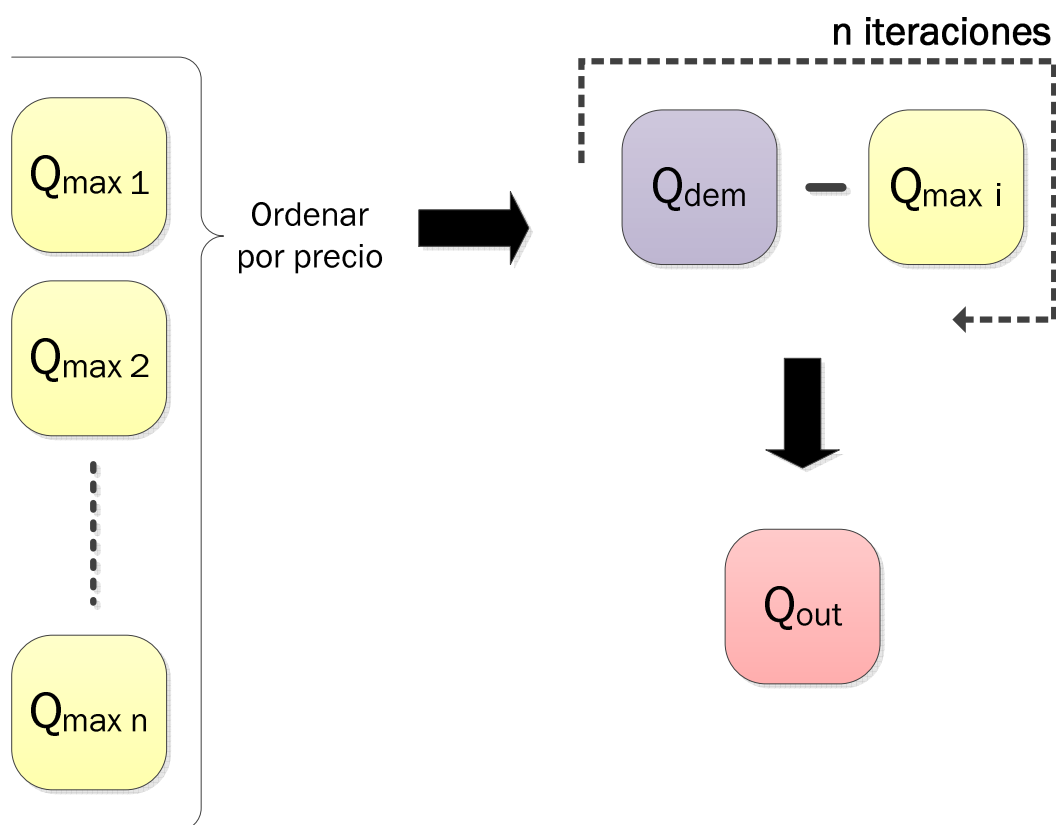


Fig. 29. Mecánica del control de Q

### 3.4 Implementación informática

En este apartado se va a explicar la organización y características internas del SMA que implementa el emulador. No se busca aquí ahondar en los entresijos del código sino proporcionar una panorámica de su arquitectura. La arquitectura de un SMA la definen sus agentes, los objetos que estos pueden manejar, y la manera con la que se producen las interacciones entre todos ellos.

Cada uno de los agentes, así como los dos objetos creados para la ocasión, cuentan con sus respectivos subapartados explicativos; a los que precede otro subapartado que aborda el estándar de comunicación que registrará las interacciones entre agentes.

La Fig. 30 es una captura de pantalla del SMA diseñado para emular la microrred. Los agentes se encuentran distribuidos en función de una hipotética disposición física de los elementos que la configuran. Por ejemplo, el container denominado como Node-3 albergaría a dos agentes, cuyas funciones consistirían en gestionar una batería y un sistema de placas solares respectivamente.

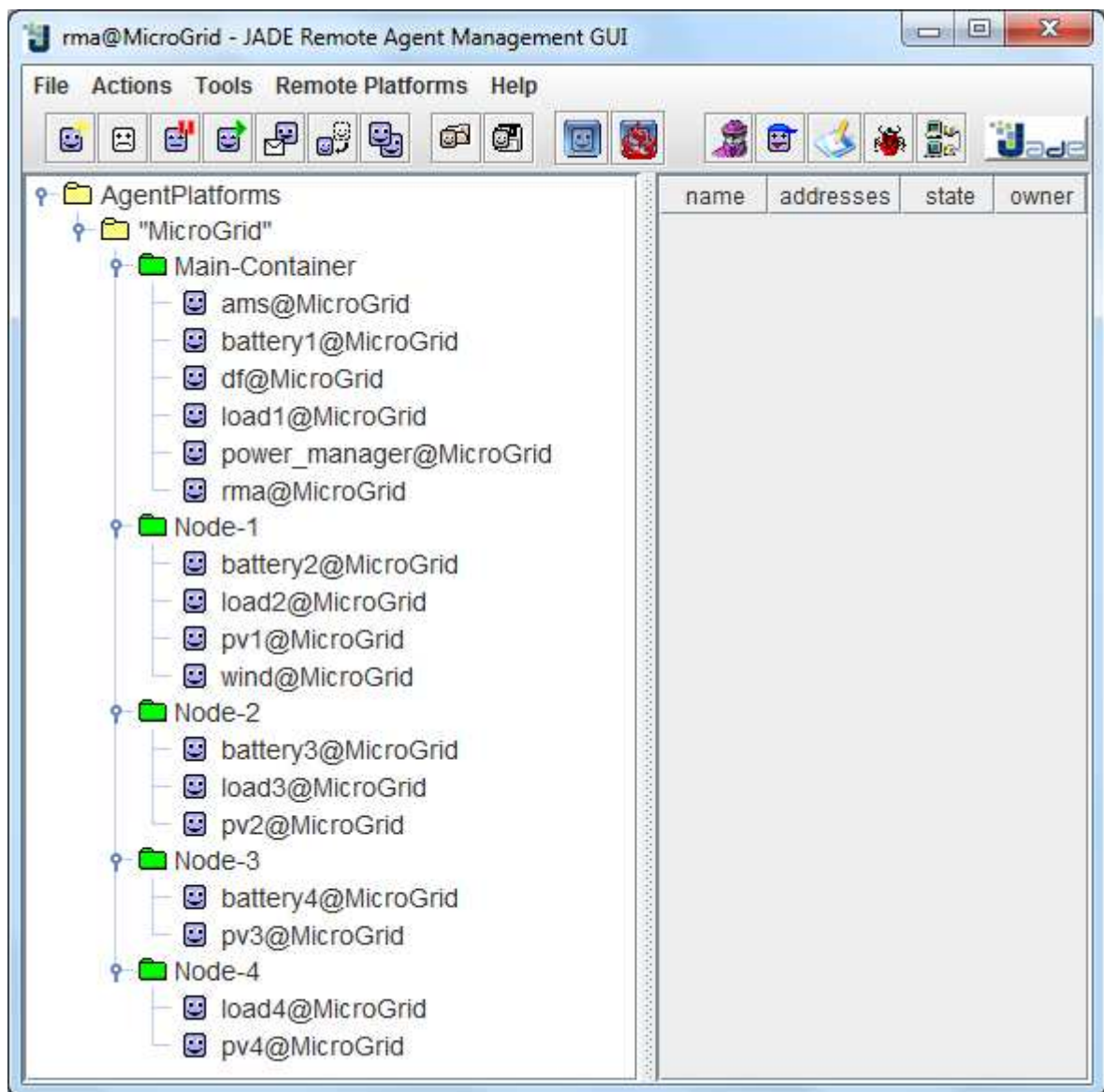


Fig. 30. SMA de la microrred implementada

### 3.4.1 Relación de clases

Desde un punto de vista funcional, los agentes de este SMA se pueden clasificar en los tres grupos de la Fig. 31: agentes de control, de gestión y auxiliares.

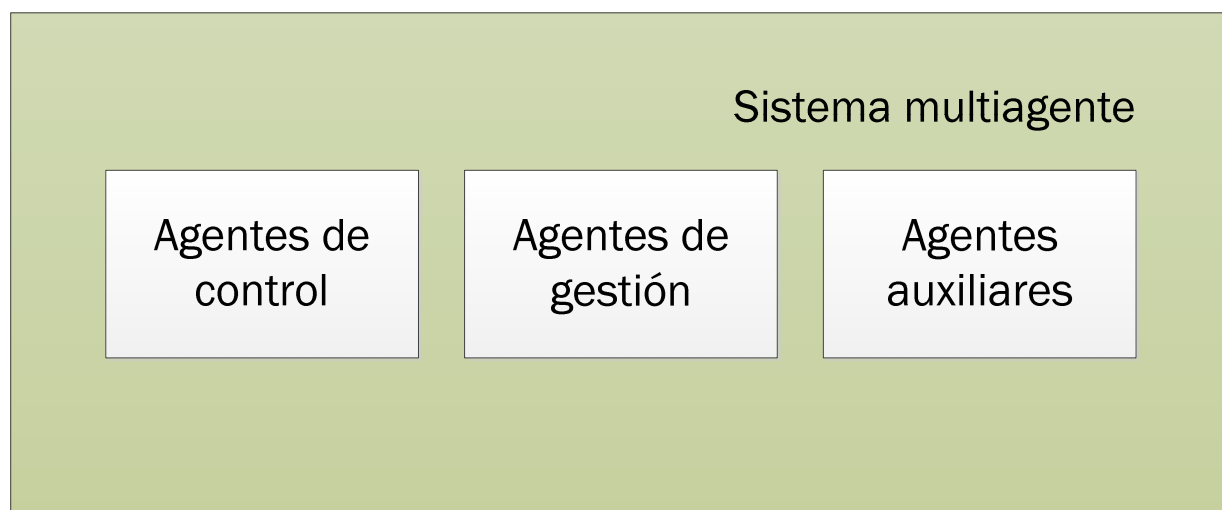


Fig. 31. Tipos de agente según su rol

- La función de un agente de **control**, como su nombre indica, es controlar y organizar la microrred, ya sea de forma total o parcial (controlar únicamente un nudo sería un ejemplo de parcial). Los agentes de control son la máxima autoridad de la microrred.
- Los agentes de **gestión** se encuentran asociados a elementos reales tales como generadores, sistemas de almacenamiento o cargas. Su misión consiste en manejar tales dispositivos: hacer lecturas de los valores de P y Q que esté suministrando, comunicarse con los agentes de control, ejecutar las consignas recibidas, etc.
- Los agentes **auxiliares** proporcionan servicios adicionales al resto de agentes del SMA, la plataforma JADE arranca por defecto tres agentes que entrarían dentro de esta denominación: AMS, DF y RMA (ver apdo. 2.4.2).

En la Tabla 10 aparecen los agentes clasificados según la función. No sólo se han incluido los agentes que están actualmente desarrollados (en negrita), sino también futuras subclases que de ellos pudieran derivarse (en gris).

El actual agente DG, por poner un ejemplo, no diferencia entre un tipo de generación y otra, es decir, procesa los inputs sin tener en cuenta si son de naturaleza eólica, fotovoltaica, etc. Sería interesante, por lo tanto, extender la clase DG para emular mejor las particularidades de cada tipo de generación.

Las clases en gris son sólo algunas de las posibles subclases a desarrollar y se ha decidido incorporarlas a la tabla aunque realmente suponen líneas de continuidad para futuros desarrollos del emulador (apdo. 3.6.2.1).

Grupo	Categoría	Superclase	Clase
Agentes	Gestión	DG	Solar PV
			Wind
			Micro hydro
			Biomass
		Storage	Pumped hydro
			Compressed air
		Load	Priority
			Non-priority
		Grid (PCC)	
	Control	PowerManager	
		Node	
	Auxiliar	SerialPort	
Elementos	Físicos	Battery	Lithium-Ion
			Nickel-Cadmium
		Converter	
		Transmission line	
		Impedance	

Tabla 10. Principales clases Java del SMA

Todos y cada uno de los agentes de la Tabla 10 deben implementar los métodos `setup()` y `takeDown()`, tal y como impone el *framework* JADE. En el método `setup()` los agentes van a registrar sus servicios en el servicio de páginas amarillas del agente DF e iniciarán sus respectivos comportamientos. Al ejecutar `takeDown()`, enviarán una notificación a DF para que borre sus servicios de la lista.

### 3.4.2 Protocolo de interacción FIPA Contract Net

El protocolo de interacción (PI) FIPA Contract Net es una modificación del PI contract net original que añade los actos comunicativos de rechazar y confirmar. En el PI contract net, un agente (el iniciador) desempeña el rol de *manager* que desea que una tarea sea llevada a cabo por uno o más agentes (los participantes) [23].

Es probable que el manager o iniciador desee realizar algún tipo de optimización basada en alguna variable de coste o tiempo de la función que caracteriza la tarea. Por ejemplo, varios agentes pueden ofrecer el mismo servicio a distinto precio, u comprometerse a completar el trabajo en un tiempo determinado.

#### 3.4.2.1 Funcionamiento

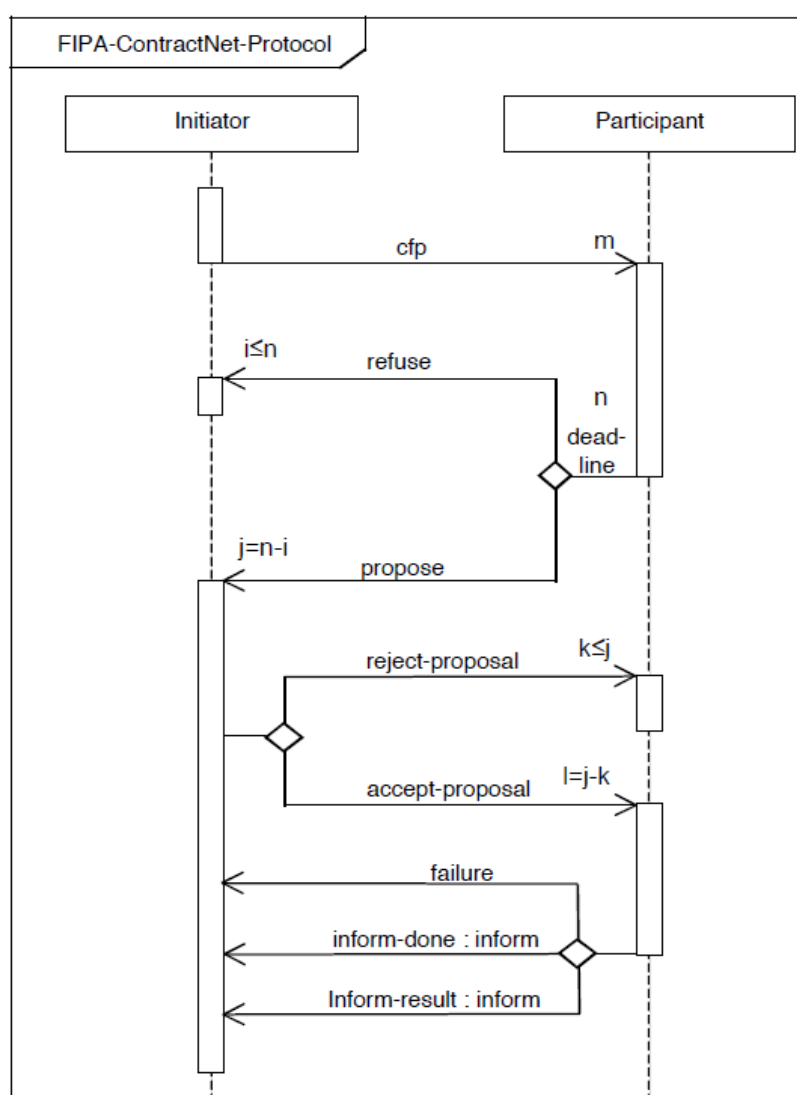
La Fig. 32 contiene la representación UML1 del FIPA Contract Net Protocol, donde se muestra el flujo de mensajes ACL que pueden intercambiar el iniciador y los participantes.

Como su nombre indica, el iniciador decide cuándo comenzar el protocolo y lo hace enviando *m* mensajes ACL de *cfp* (*call for proposal*), que representa el acto comunicativo

de solicitar propuestas. Los agentes participantes reciben las *cfp* y pueden optar por, o bien declinar la invitación, o bien enviar una propuesta. Únicamente las propuestas recibidas dentro de tiempo (antes del *deadline*) serán tomadas en consideración por el iniciador.

Una vez que se han evaluado las diferentes propuestas, el iniciador responderá a los participantes con un mensaje ACL de aceptación o de rechazo. Los mensajes del tipo aceptación (*accept-proposal*) pueden contener instrucciones o especificaciones adicionales al mero hecho de aceptar. El iniciador no está obligado a aceptar la oferta de un participante de manera íntegra. Por ejemplo, si se está llevando a cabo una subasta de energía y un participante envía una propuesta en la que oferta 3 kWh a un precio 0.14€ el kWh, el iniciador podrá contestar con una *accept-proposal* en la que sólo acepte 1.8 kWh.

El último paso del protocolo lo realizan los participantes y consiste en informar al iniciador acerca del éxito o fracaso que el participante ha tenido en el desempeño de la tarea asignada.



**Fig. 32. Diagrama de protocolo: FIPA-ContractNet**

### 3.4.2.2 Clases ContractNetInitiator y ContractNetResponder

Estas clases forman parte del *framework* de JADE y facilitan la implementación del rol del iniciador y del participante. Ambas clases se puede entender como una máquina de estados, puesto que derivan de la clase *FSMBehaviour*. Los estados de la FSM (*Finite State Machine*) de cada clase se pueden deducir a partir de la Fig. 32.

La lógica que sigue la FSM para pasar de un estado a otro utiliza las recepciones y envíos de mensajes ACL a modo de *trigger*, y viene implementada por defecto en la propia clase. Bastará entonces con completar ciertos métodos que modifiquen esta lógica, así como los outputs de cada uno de los estados (si los hay).

*ContractNetInitiator* y *ContractNetResponder* ponen a disposición del programador una serie de métodos con los que especificar cómo actuar en cada uno de los estados de la FSM. Estos métodos están “en blanco” y corresponde al desarrollador sobrescribirlos para dotarlos de funcionalidad. A continuación se va a proceder a enumerarlos, siguiendo el flujo del diagrama de la Fig. 32:

- 1) El iniciador solicita propuestas a otros agentes enviando  $n$  mensajes ACL de *CFP* que especifiquen la acción solicitada y las condiciones de ejecución. El programador dispone para ello del método *prepareCfps()*. Este método exige retornar un vector con las *CFP* a enviar (normalmente un mismo mensaje ACL con múltiples receptores).
- 2) Los participantes podrán enviar entonces una propuesta (*PROPOSE*) en la que se incluyan todas condiciones bajo las que llevarían a cabo su propuesta (precio, tiempo...). Si no realizan ninguna propuesta, los participantes enviarán un mensaje de *REFUSE* o, si ha habido problemas en la comunicación, un *NOT-UNDERSTOOD*. El método *prepareResponse()* es ejecutado cada vez el participante recibe una *CFP*, por lo que deberá implementarse para dar la respuesta adecuada.
- 3) *ContractNetInitiator* identifica los mensajes *PROPOSE* como “respuestas” (*responses*) y cuenta con dos métodos para sobrescribir con los que implementar las reacciones del iniciador: *handlePropose()* y *handleAllResponses()*. El método *handlePropose()* se ejecuta de forma independiente con cada *PROPOSE*, mientras que *handleAllResponses()* lo hace cuando han sido recibidas todas las propuestas o ha finalizado un tiempo de espera predeterminado. Desde cualquiera de estos dos métodos se puede enviar el correspondiente mensaje de *ACCEPT-PROPOSAL* (o *REJECT-PROPOSAL*) en el que se comunique al participante las tareas que debe realizar. El método *handleRefuse()* especificará la reacción del iniciador ante un mensaje de *REFUSE* y el *handleNotUnderstood()* dictaminará el proceder en caso de recibir un *NOT-UNDERSTOOD*.
- 4) Los participantes cuyas propuestas fueron aceptadas y que hayan completado su tarea podrán responder con un mensaje *INFORM*. Si algo ha ido mal, responderán con un mensaje *FAILURE*. El método en el que se debe llevar a cabo esta acción es *prepareResultNotification()*.

Tanto *ContractNetInitiator* como *ContractNetResponder* disponen de métodos que se ejecutan cuando reciben un mensaje ACL que no cumple las especificaciones del protocolo. De nuevo corresponde al programador dotarles de utilidad práctica.



En el modelo implementado, se utiliza el protocolo FIPA Contract Net para organizar el reparto de potencias entre los distintos elementos de la microrred. En este proceso de “subasta”, los agentes sólo tienen en cuenta un tipo determinado de mensaje (a veces más de uno) según el estado de la FSM en el que se encuentren. Por ejemplo, si ya se ha cerrado el periodo de aceptación de ofertas, `ContractNetInitiator` ignorará cualquier mensaje de *PROPOSE* que pudiera recibir.

Para una mayor robustez y seguridad en las comunicaciones, se han establecido dos campos adicionales obligatorios para cualquier mensaje ACL referido al proceso de negociación de consignas, que se muestran en la Tabla 11.

Campo	Descripción
Protocol	"fipa-contract-net"
Conversation-id	"AUCTION"

**Tabla 11.** Campos comunes a todos los mensajes ACL del proceso de subasta

De ahora en adelante, si no se especifica lo contrario, se asume que todo mensaje ACL incluye también estos dos campos.

### 3.4.3 Agentes

En los siguientes cuatro subapartados se detallarán las principales características de cuatro de los cinco agentes implementados para la ocasión, así como información acerca de los mensajes ACL con los que participan en el protocolo FIPA Contract Net.

El quinto agente (agente `SerialPort`) no intercambia ningún mensaje, por lo que no se ha creído necesario incluir explicaciones adicionales a las que se recogen en el apartado 3.4.6, donde se aborda el modo *monitorización*.

#### 3.4.3.1 Agente Power Manager

`PowerManager` extiende la clase `Agent` para implementar el rol de controlador de la microrred. Exhibe dos comportamientos principales: `AMSSubscriberPM` y `MailBoxPQControl`.

`AMSSubscriberPM` es un comportamiento cíclico derivado de la clase `AMSSubscriber`. Con este comportamiento `PowerManager` se puede subscribir al servicio de páginas amarillas del agente `DF`, que le avisa del nacimiento o muerte de cualquier agente `DG`, `Storage` o `Load`, pudiendo así actualizar una lista en la que guarda las AID de los miembros activos de la microrred. Será a los miembros que estén en dicha lista a los que `PowerManager` enviará las *CFP* que dan inicio al protocolo FIPA Contract Net.

El comportamiento `MailBoxPQControl` extiende `ContractNetInitiator` y se ocupa del control propiamente dicho. El nombre *MailBox* hace referencia a su función de procesado de mensajes ACL.

La información interna, tanto de entrada como de salida es almacenada en los vectores de objetos que recoge en la Tabla 12.

Vector	Almacena...
infoGen	...la información de cada uno de los generadores activos de la microrred.
infoStor	...la información de cada uno de las baterías activas de la microrred.
infoLoad	...la información de cada uno de las cargas activas de la microrred.
SetPointsPQ	...las consignas de P y Q para todos los elementos activos de la microrred.
DataOut	...los outputs para cada uno de los periodos.

Tabla 12. Vectores utilizados por MailBoxPQControl

Como cualquier clase que extienda `ContractNetInitiator`, `MailBoxPQControl` sobrescribe los métodos especificados en el apartado 3.4.2.2, de los que se pasa a comentar los más relevantes:

- **handlePropose:** es llamado cada vez que se reciba un mensaje ACL de *PROPOSE* que cumpla con las especificaciones del protocolo. El método procesa las propuestas según hayan sido enviadas por un agente DG, Storage, o Load; y guarda la información extraída en los vectores de la Tabla 12.
- **handleAllResponses:** se ejecuta tras haber recibido las  $n$  propuestas esperadas (o tras agotarse el tiempo de subasta), es decir, tras haberse realizado  $n$  llamadas a `handlePropose`. En este método se hace la llamada al método `getSetPoints`, al que se le pasa como parámetros `infoGen`, `infoStor` e `infoLoad` y el cual retorna `SetPointsPQ`.

Una vez que se dispone del vector de consignas `SetPointsPQ`, se procede a enviar los correspondientes mensajes de *ACCEPT-PROPOSAL* a los agentes que cuyas ofertas han sido aceptadas (Tabla 13).

Campo	Descripción	Unidades
P_SETPOINT	Consigna de potencia activa	kW
Q_SETPOINT	Consigna de potencia reactiva	kVAr

Tabla 13. Mensaje ACL de *ACCEPT-PROPOSAL* enviado por el agente PowerManager

- **handleInform:** es llamado cada vez que se reciba un mensaje ACL de *INFORM* que cumpla con las especificaciones del protocolo. En dichos mensajes los agentes participantes informan de los resultados de sus acciones: los agentes DG comunican sus P y Q de salida ( $P_{out}$  y  $Q_{out}$ ), que pueden no coincidir con las consignas que PowerManager envió. Los agentes Storage notifican, además de  $P_{out}$  y  $Q_{out}$ , su SOC al final del periodo.

Los valores de  $P_{out}$ ,  $Q_{out}$  y SOC se van sumando a tres variables internas para poder calcular posteriormente los balances globales de P y Q, y el SOC medio del conjunto de baterías.

- **handleAllResultNotifications:** se ejecuta tras haber recibido los  $n$  ACL mensajes de *INFORM* esperados. Este método calcula los balances globales de P y Q, así como el  $\cos(\phi)$  visto desde el lado de red, y el SOC global de las baterías.

Para ello se vale de las tres variables internas en las que `handleInform` fue guardando  $P_{out}$ ,  $Q_{out}$  y  $SOC$  de cada uno de los informantes.

Una vez calculados estos valores globales, se almacenan en el vector `DataOut` y se reinician las variables internas para una nueva iteración del protocolo `Contract Net`.

### 3.4.3.2 Agente DG

Este agente se instala en un punto de generación distribuida y gestiona tanto el propio generador como su convertidor asociado. Cuenta con un comportamiento llamado `MailBoxDG` que hereda las características de `ContractNetResponder`, implementando así el rol de participante del protocolo FIPA `Contract Net`. Los métodos sobrescritos por `MailBoxDG` para la ocasión son:

- `handleCfp`: es llamado cuando se reciba un mensaje ACL de *CFP* que cumpla ciertos requisitos del protocolo. El agente DG responderá a dicha *CFP* mediante un mensaje de *PROPOSE* con la información de la Tabla 14.

Campo	Descripción	Unidades
AGENT_SERVICE	Servicio prestado: generación	-
POWER_RESOURCE	Potencia renovable disponible	kW
PRICE	Factor de coste	€/kVAr
VSC_MAX_APPARENT_POWER	Potencia aparente nominal del convertidor asociado	kVA
CONVERTER_EFF	Eficiencia del convertidor asociado	%/1

Tabla 14. Mensaje ACL de *PROPOSE* enviado por el agente DG

- `handleAcceptProposal`: el comportamiento llamará a este método cuando reciba un mensaje de *ACCEPT-PROPOSAL* que satisfaga las condiciones establecidas por `ContractNetResponder`. Dicho mensaje de aceptación contendrá las consignas de la Tabla 13, que serán pasadas como parámetros a la instancia de la clase `Converter`, es decir, al convertidor asociado al elemento generador.

Cuando el convertidor recibe las consignas, les aplica una eficiencia para obtener los valores de  $P$  y  $Q$  a la salida, valores que son realimentados al agente para que éste pueda, si no ha surgido ningún problema, incluirlos en un mensaje de *INFORM* y enviárselos al agente `PowerManager`. Si algo ha salido mal el agente responderá con un mensaje de *FAILURE*.

Campo	Descripción	Unidades
AGENT_SERVICE	Servicio prestado: generación	-
P_OUT	Potencia activa a la salida del convertidor	kW
Q_OUT	Potencia reactiva a la salida del convertidor	kVAr

Tabla 15. Mensaje ACL de *INFORM* enviado por el agente DG

### 3.4.3.3 Agente Storage

Este agente se instala en un punto de almacenamiento de energía y gestiona tanto la batería como el convertidor con el que se conecta al bus AC. De forma análoga al agente DG, Storage cuenta con un comportamiento llamado MailBoxStorage que extiende ContractNetResponder para implementar el rol de participante en el protocolo FIPA Contract Net.

Los métodos sobrescritos son los mismos que los del agente DG (extienden la misma clase), aunque las funcionalidades implementadas cambian:

- **handleCfp**: es llamado cuando se recibe un mensaje ACL de CFP que cumpla los requisitos. El agente Storage responderá a dicha CFP mediante un mensaje de PROPOSE con la información de la Tabla 16.

Campo	Descripción	Unidades
AGENT_SERVICE	Servicio prestado: almacenaje	-
CHARGE_POWER_OFFER	P disponible para carga	kW
DISCHARGE_POWER_OFFER	P disponible para descarga	kW
CHARGING_REACTIVE_OFFER	Q disponible en carga	kVAr
DISCHARGING_REACTIVE_OFFER	Q disponible en descarga	kVAr
PRICE	Factor de coste	€/kVAr

Tabla 16. Mensaje ACL de PROPOSE enviado por el agente Storage

Las baterías ofrecen P y Q distintas para carga y descarga, ya que Storage desconoce a priori si se le va pedir que ceda o que absorba energía. Por ejemplo, una batería cargada al 100% ofrecerá su potencia nominal para descarga, pero no podrá ofrecer potencia de carga (no podrá absorber P); y viceversa.

La secuencia de cálculo de las ofertas es muy simple: Storage conoce la potencia aparente máxima que puede entregar su convertidor ( $S_{con}$ ), así como el estado de la batería al inicio del periodo. Esta información le permite calcular la potencia activa ofertada ( $P_{of}$ ), tanto de carga como de descarga.

La oferta de potencia reactiva ( $Q_{of}$ ) se calcula mediante la ecuación (14), introduciendo cada vez uno de los dos valores de  $P_{of}$ .

$$Q_{of} = \sqrt{(S_{con})^2 - (P_{of})^2} \quad (14)$$

A diferencia de los agentes DG, que priorizan la inyección de Q sobre el suministro de P, los agentes Storage van a dar preferencia a P sobre Q.

- **handleAcceptProposal**: la implementación de este método es idéntica a la del agente DG (apdo. 3.4.3.2), por lo que no se va a volver a explicar aquí.

### 3.4.3.4 Agente Load

Este agente se instala en un punto de consumo y monitoriza la demanda de potencia activa y reactiva. El comportamiento que implementa su rol de participante en el protocolo FIPA Contract Net es *MailBoxLoad*, que también deriva de *ContractNetResponder*.

Las funciones de este agente son mucho más sencillas que las de los agente *DG* y *Storage*, e incluso se podría haber planteado un diseño en el que las cargas no contaran con su propio agente asociado.

Se ha decidido aun así dotarlas del rango de agente con vistas a futuras expansiones del emulador. Además, el hecho de que las cargas puedan intervenir como participantes en el proceso de reparto de energía simplifica sobremanera el funcionamiento del conjunto.

Los métodos sobrescritos para la ocasión son:

- *handleCfp*: es llamado cada vez que se recibe una *CFP* que cumpla los requerimientos del protocolo. Si la carga registra consumo en ese periodo responderá con un mensaje de *PROPOSE*, cuyo contenido aparece en la Tabla 17; de lo contrario responderá con un *REFUSE*.

Campo	Descripción	Unidades
P_DEMAND	Demanda de potencia activa	kW
Q_DEMAND	Demanda de potencia reactiva	kVAr

Tabla 17. Mensaje ACL de *PROPOSE* enviado por el agente Load

(Con la implementación actual, el agente *PowerManager* acepta cualquier propuesta enviada por un agente Load).

- *handleAcceptProposal*: la implementación es idéntica a la del apartado 3.4.3.2, con la salvedad de que las cargas se conectan de forma directa al bus AC. No será por tanto necesaria la intermediación de ningún convertidor, y el mensaje ACL de *INFORM* con el que el agente Load responderá a *PowerManager* contendrá los mismos valores de P y Q que le indicó éste en el mensaje de aceptación de propuesta (que a su vez son idénticos a los que Load le propuso en un principio).

### 3.4.4 Elementos físicos

Por cuestiones de diseño, se ha preferido que los modelos del convertidor y de la batería sean implementados como objetos Java, de los que los agentes pueden crear una instancia (ver apdo. 2.3.2), en lugar de como agentes en sí. Se busca así caracterizarlos como elementos pasivos, como dispositivos carentes de inteligencia que los agentes pueden crear, eliminar o manejar a su antojo.

#### 3.4.4.1 Modelo del convertidor. Clase *Converter*.

Con esta clase se modelan los convertidores que tanto los agentes *DG* como los agentes *Storage* utilizan para conectarse al bus AC de la microrred. Aunque se suponen distintas topologías (Fig. 18), se ha implementado un único modelo de convertidor para todas ellas.

En la Fig. 33 aparece un conversor AC-AC con enlace por corriente continua. Sea o no sea ésta la topología del convertidor en cuestión, el conjunto puede ser considerado una “caja negra”, de la que sólo son relevantes sus entradas y salidas.

Dicha caja negra quedaría delimitada por la línea de trazos y puntos que rodea a los dos VSC. Con esta simplificación se obtiene un modelo de convertidor al que le entran una serie de inputs ( $P_{in}$ ) y, en función de unos parámetros característicos ( $S$  y  $\eta$ ), produce unos outputs ( $P_{out}$ ,  $Q_{out}$  y  $P_{losses}$ ).

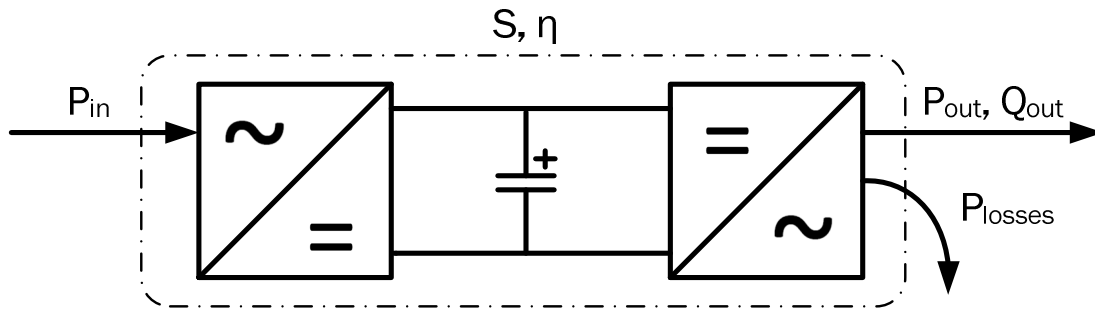


Fig. 33. Modelo del convertidor

Los parámetros característicos del convertidor son su potencia aparente nominal ( $S$ ) y su eficiencia o rendimiento ( $\eta$ ) y se corresponden con las constantes `max_aparent_power` y `EFF`. Las variables internas de la clase `Converter` son tres:

- `p_out`: potencia activa a la salida del convertidor ( $P_{out}$ ), en kW.
- `q_out`: potencia reactiva a la salida del convertidor ( $Q_{out}$ ), en kVar.
- `p_losses`: potencia pérdida en la conversión ( $P_{losses}$ ), en kW.

La clase convertidor cuenta con seis métodos públicos del tipo “get” (getters) accesibles por cualquier agente que haya creado previamente una instancia del objeto. En todos ellos se retorna una de las constantes o variables mencionadas anteriormente.

Por ejemplo el método `getEfficiency()` retorna el valor del rendimiento, el método `getP(SetPointP)` retorna la `p_out` para una consigna de  $P$  dada y el método `getPowerLosses()` devuelve el valor de la variable `p_losses`.

#### 3.4.4.2 Modelo de la batería. Clase `Battery`.

Con esta clase se modelan las baterías. Cada agente `Storage` crea una instancia del objeto `Battery` en su inicialización, dentro del método `setup()`. Una vez hecho esto, el agente `Storage` pasa a tener el control de la batería y puede llevar a cabo múltiples tareas a través de sus métodos. Las constantes que caracterizan la batería son:

- `nom_power`: potencia máxima que el sistema permite ceder o absorber, en kW.
- `nom_voltage`: voltaje nominal de salida, en V.
- `capacity`: cantidad de energía que la batería es capaz de almacenar, en kWh.
- `min_soc`: mínimo estado de carga permitido, en %.
- `max_soc`: máximo estado de carga permitido, en %.
- `initial_soc`: estado de carga inicial, en %.

Para facilitar las operaciones internas se dispone de las variables `energy_capacity` y `energy_stored`, expresadas en kJ y cuyos nombres son suficientemente representativos.

`Battery` dispone de un de ocho métodos accesibles por el agente `Storage`. De entre los que se pueden destacan `getMaxChargePowerOffer`, `getMaxDischargePowerOffer` y `discharge`.

El protocolo FIPA-ContractNet exige a cada agente una propuesta u oferta y el agente `Storage` no es una excepción. `Storage` oferta la potencia activa almacenada en su batería. Esta oferta no depende de la disponibilidad del recurso, como en el caso de las ofertas de los generadores, sino que dependerá únicamente de su SOC o estado de carga interno. Una batería totalmente descargada no podrá ofrecer potencia de descarga, pero sí podrá ofrecer su máxima potencia de carga. Por el contrario, una batería totalmente cargada no tiene capacidad de absorber más energía por lo que la potencia de carga será igual a cero.

Los métodos `getMaxChargePowerOffer` y `getMaxDischargePowerOffer` guardan gran similitud entre sí. Cuando son llamados por `Storage`, primeramente calcularán la energía almacenada. Para el supuesto de carga se emplea la ecuación (15) y para el de descarga la ecuación (16).

La variable  $E_a$  representa la energía disponible (*available*), expresada en kJ. El resto de variables corresponden a las constantes mencionadas anteriormente:  $SOC_{min}$  y  $SOC_{max}$  a `min_soc` y `max_soc` respectivamente,  $SOC_{cu}$  al estado de carga al inicio del periodo (*current*), y  $E_c$  a `energy_capacity`.

$$E_a = \left( \frac{SOC_{max} - SOC_{cu}}{100} \right) * E_c \quad (15)$$

$$E_a = \left( \frac{SOC_{cu} - SOC_{min}}{100} \right) * E_c \quad (16)$$

Una vez obtenida la energía disponible  $E_a$ , y sabiendo que la potencia es energía consumida por unidad de tiempo, se calcula el valor máximo de potencia que sería capaz de entregar la batería durante todo el periodo  $T_s$ .

Como se está trabajando con periodos treintaminutales,  $T_s$  tendrá un valor de 1800 segundos. Por ejemplo, si el valor de `max_power_offer` ( $P_{max}$ ) fuera 450 significaría que hay energía suficiente para entregar 450 kW de forma ininterrumpida durante los próximos 30 minutos.

$$P_{max} = \frac{E_a}{T_s} \quad (17)$$

En la inmensa mayoría de los casos este valor será superior al de potencia máxima permitida (`nom_power`). Para que esto no suceda, la variable `max_power_offer` es evaluada por un sencillo *if statement* antes de ser retornada por el método. Si se supera la potencia nominal, la oferta se trunca al valor de ésta.

### 3.4.5 Otras clases

El emulador cuenta con muchas más clases Java además de los agentes y de los modelos de convertidores y baterías. Por razones de espacio no es posible incluir explicaciones de cada una de ellas, pero se proporciona la Tabla 18 a modo de resumen.

Package	Clase	Función
Strategies	MainControl	Control de Q, cálculo de valores agregados de P y Q
	PeakShaving	Control de P: <i>Peak Shaving</i>
	Smoothing	Control de P: Alisamiento de la curva de demanda
Inputs	InputGenerator	Guardar los inputs de la DG (potencia y precio)
	ReaderInputDem	Leer los ficheros .xlsx de demanda
	ReaderInputGenerator	Leer os ficheros .xlsx de generación
Data	Members	Guardar la AID de los miembros activos de la microrred
	DataGenerator	Almacenar la información de la generación
	DataStorage	Almacenar la información del almacenamiento
	DataDemand	Almacenar la información de la demanda
	SetPoints	Almacenar las consignas de P y Q
Outputs	OutputControl	Almacenar los outputs del agente de control
	OutputGenerator	Almacenar los outputs del agente instalado en la DG
	OutputStorage	Almacenar los outputs del agente instalado en una batería
SerialPort	TwoWaySerialComm	Asistir en la configuración y manejo del puerto serie
Assistant	Arguments	Argumentos pasados a los agentes en su arranque
	Constants	Interfaz Java con las constantes comunes a toda el SMA
	myDummy (Agente)	Ayudar en las tareas de depuración de fallos

Tabla 18. Clases secundarias

El *package* “Strategies” lleva a cabo el control P/Q de los apartados 3.3.3 y 3.3.4. En el *package* “Inputs” se encuentran las clases que gestionan los inputs de generación y demanda para el modo *simulación*.

El *package* “Data” contiene las clases utilizadas para almacenar los datos en las operaciones intermedias. Por ejemplo, la clase *DataGenerator* alberga variables para almacenar los parámetros de la Tabla 16. *PowerManager* se valdrá de un vector de objetos *DataGenerator* para guardar la información suministrada por todos y cada uno de los agentes *Storage* participantes.

Las clases del *package* “Outputs” almacenan los datos de salida del periodo, y se organizan en grandes vectores (arrays con funcionalidades extra) para ser posteriormente volcados a ficheros .cvs.



### 3.4.6 Modo monitorización

Si bien el modo monitorización se deja como línea de continuidad para futuros desarrollos del emulador, se han realizado algunos ensayos para verificar su viabilidad.

Un agente asociado a un elemento físico de la microrred (generadores, baterías o cargas) puede funcionar en modo *simulación*, como se ha visto hasta ahora, o en modo *monitorización*. En el modo simulación los datos de entrada provenían de ficheros .xlsx, mientras que los de salida se almacenaban ficheros .csv. En modo *monitorización*, por el contrario, los datos se obtienen comunicándose en tiempo real con el elemento físico.

Para posibilitar esta comunicación se dispondrá de tarjetas DSP (*Digital Signal Processing*), que actuarán como intermediarias entre el ordenador en el que se ejecuta el agente y el convertidor que gobierna el elemento real. Dichas tarjetas cuentan con salidas del tipo “puerto serie” y pueden conectarse a un puerto COM del PC para enviar o recibir paquetes de datos bit a bit, es decir, en serie. En el apartado 3.6.2.1, correspondiente a las líneas futuras del emulador, se ha incluido un esquema de esta configuración (Fig. 53).

La comunicación serie es el protocolo de bajo nivel más común para la comunicación entre dos o más dispositivos. Normalmente un dispositivo es un ordenador, mientras que el otro dispositivo puede ser un módem, una impresora, otro ordenador, o un instrumento científico como un osciloscopio o, como en esta ocasión, una tarjeta DSP. El puerto serie envía y recibe bytes de información utilizando un formato binario o un formato de texto del tipo ASCII.

Antes de enviar o recibir datos, el puerto serie y el instrumento han de tener la misma configuración de comunicación. Para ello es necesario definir los siguientes parámetros:

Parámetro	Descripción
BaudRate	Velocidad de transmisión de datos
DataBits	Número de bits a transmitir en un dato
Parity	Habilita/deshabilita el chequeo de la paridad
StopBits	Numero de bits para indicar el final de un byte
Terminator	Carácter terminación
Flow Control	Control de hardware
Time out	Tiempo límite de comunicación

Tabla 19. Parámetros que definen la comunicación serie

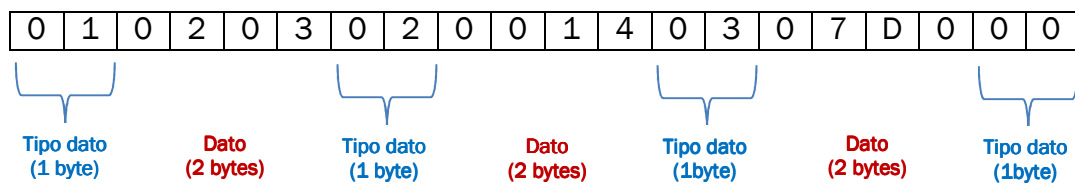
Para validar este concepto se ha creado un nuevo agente: el agente `SerialPort`. Este agente tiene la capacidad de comunicarse con el puerto COM del PC, tanto para configurarlo según la Tabla 19 como para enviar y recibir datos a través de él. Naturalmente, puede haber más de uno dependiendo de cómo esté distribuida la plataforma multiagente.

El agente `SerialPort` se vale de la clase `TwoWaySerialComm` para manejar el puerto serie. `TwoWaySerialComm` está basada en la librería `RXTX`, de libre distribución [24], e implementa una serie de métodos a los que `SerialPort` puede llamar cuando lo desee.

Por ejemplo, el método `setSerialPortParams(int baudRate, int dataBits, int stopBits, int parity)` permite configurar el puerto COM tal y como indica el siguiente fragmento de texto que sale por pantalla:

```
Serial port COM8 has been configured with the following parameters:
Baud Rate: 9600[Bd], Data Bits: 8, Stop Bits: 1, Parity (0 for NON): 0
```

Ahora que COM8 está configurado se va a proceder a enviar repetidas veces el siguiente paquete de datos, formado por un total de 80 bits (10 bytes):



El formato sigue la estructura identificador + dato y se ha pensado con vistas a un emulador más refinado, en el que no sólo se manejen flujos de potencia, sino también tensiones y corrientes. El identificador consta de 1 byte y el dato de 2 bytes, con lo que se pueden representar un total de 65536 valores distintos.

La Tabla 20 adjudica, de forma arbitraria, significado a los identificadores de la columna izquierda. Aparte de estos cinco, el byte reservado a los identificadores cuanta con valores más que suficientes para implementar habilitadores de lectura, de escritura, de fin de transmisión, etc.

Identificador	Descripción
0	Final de dato recibido
1	El dato recibido es tensión
2	El dato recibido es corriente
3	El dato recibido es potencia activa
4	El dato recibido es potencia reactiva

Tabla 20. Identificadores de dato: comunicación serie

Si se diera el caso de no recibir el byte nulo o que la trama estuviera incompleta, ésta sería desestimada y se mantendrían los valores de la lectura anterior.

Interpretando el paquete de datos acorde a la Tabla 20, se ha producido un envío de la siguiente información:

Tipo de dato	Dato
Tensión (01)	515 V (0203)
Corriente (02)	20 A (0014)
Potencia activa (03)	2000 W (07D0)
Fin de dato (00)	-

Tabla 21. Valores enviados a través del puerto serie

Para comprobar que efectivamente se ha producido el envío, se ha utilizado una aplicación que monitoriza los puertos virtuales del ordenador. La Fig. 34 es una captura de pantalla del programa en la que se puede corroborar que se está realizando el envío reiterado del paquete de bits, agrupados en formato hexadecimal, de forma correcta.

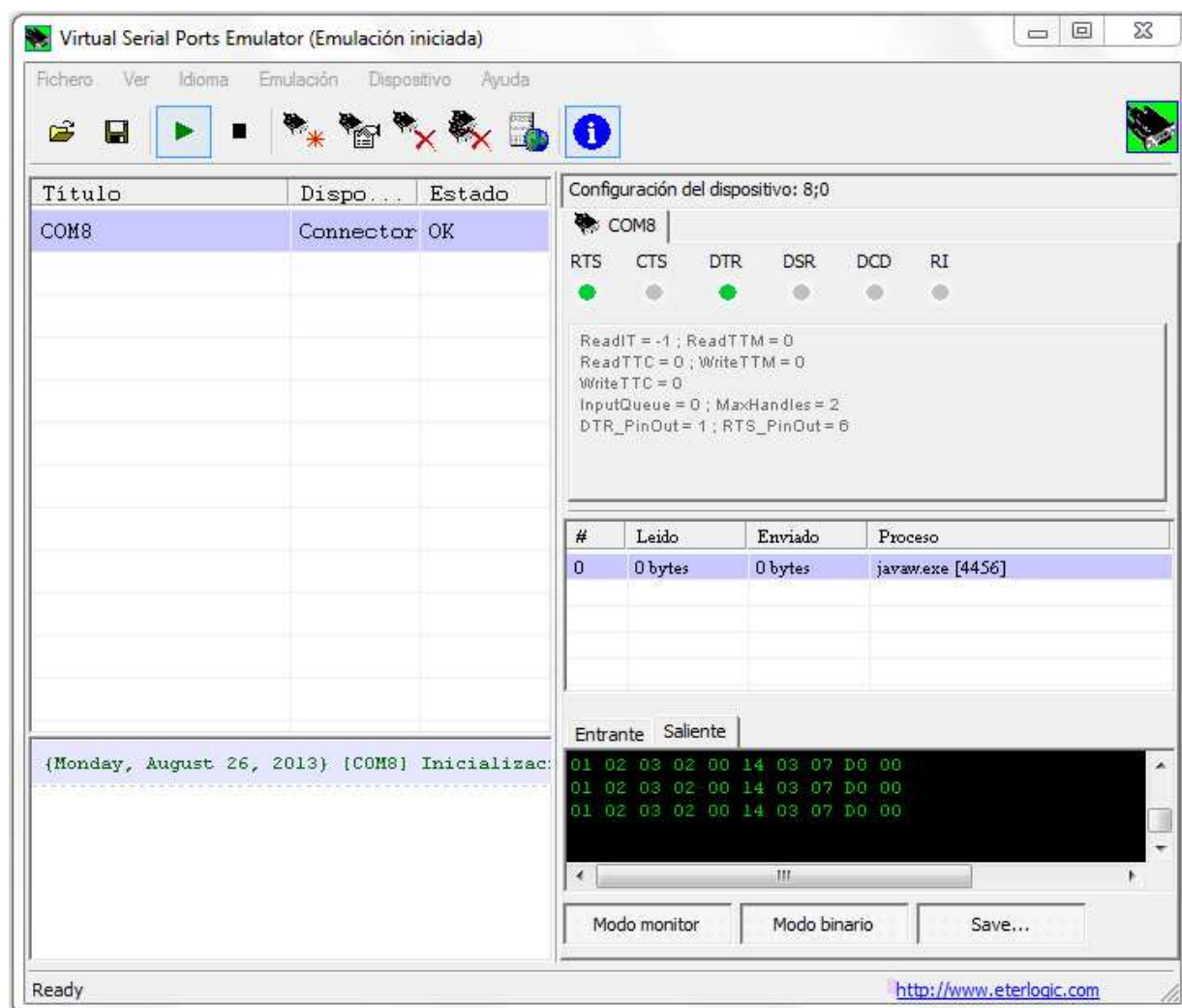


Fig. 34. Comprobación del envío datos a través del puerto COM8

### 3.5 Resultados

En las tablas y gráficas de este apartado se van a recopilar diversos parámetros estadísticos representativos de la actuación de las estrategias de control de P y Q. Los resultados se han agrupado en dos subapartados según se hayan tomado en consideración o no las energías renovables.

Por motivos de eficiencia y ahorro de espacio, se han codificado los diferentes escenarios de simulación acorde a la Tabla 22.

Opción	Codificación	Descripción
Estrategia 1	1	<i>Peak Shaving</i>
Estrategia 2	2	<i>Smoothing</i>
$\omega=4$ / $\omega=8$	$\omega=4$ / $\omega=8$ (subíndices)	Ancho de ventana ( <i>Smoothing</i> )
Caso A	A	100 kW, 1MWh (baterías)
Caso B	B	200kW, 2MWh (baterías)

Tabla 22. Resumen y codificación de las opciones de simulación

#### 3.5.1 Microrred sin DG

El primer escenario que se va a analizar no contempla generación distribuida. Las estrategias de control de potencia activa implementadas funcionan correctamente haya o no haya generación en la propia microrred, aunque por supuesto, los resultados van a diferir de manera notable.

##### 3.5.1.1 Control de P

La Tabla 23 recoge cinco parámetros estadísticos de especial relevancia a la hora de evaluar el efecto producido por las estrategias de P.  $P_{cc}(\text{inicial})$  representa el balance de potencia activa en PCC antes de la acción de las baterías. Como en este caso no hay generación interna, coincide con el valor de la demanda.

El resto de variables  $P_{cc}$  se corresponden con los balances de potencia activa resultantes de aplicar las estrategias de control, con la codificación especificada en la Tabla 22.

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
$P_{cc}(\text{inicial})$	77.0	74.3	26.9	0	246.5
$P_{cc} 1A$	76.9	77.2	23.7	0	222.9
$P_{cc} 1B$	76.9	77.2	23.5	0	232.6
$P_{cc} 2A_{\omega=4}$	77.0	77.8	23.8	0.2	232.6
$P_{cc} 2B_{\omega=4}$	77.0	77.8	23.7	0.2	212.2
$P_{cc} 2B_{\omega=8}$	76.9	79.8	18.9	0	196.8

Tabla 23. Parámetros estadísticos de las estrategias de P: no DG (kW)

La media prácticamente no varía, lo que no es de extrañar puesto que las baterías solo pueden aportar la energía que previamente han absorbido del propio sistema. La mediana

sí que se incrementa ligeramente en todos los casos. En el *Peak Shaving* (1) este incremento se debe a la rutina de descarga automática de las baterías, mientras que para la estrategia de alisado con media móvil (2) se debe a la lentitud con la que la curva de demanda se recupera de los picos de consumo, siendo más acusado cuanto más capacidad tengan las baterías y cuanto más grande sea el ancho de ventana  $\omega$ .

La mayor diferencia entre los casos A y B se aprecia en la desviación estándar, principal indicador de la variabilidad. En el caso B (200 kW, 2 MWh) las estrategias disponen de más margen de actuación respecto al A (100 kW, 1 MWh), lo que se ve reflejado en un valor de  $\sigma$  más pequeño. El caso  $2B_{\omega=8}$  presenta los valores más bajos de  $\sigma$  y de máximo, consecuencia de aumentar las prestaciones del SAE a la par que el ancho de ventana  $\omega$ .

Los histogramas de la Fig. 35 y la Fig. 36 representan respectivamente la función de densidad de probabilidad y la función de distribución acumulada del balance neto en la microrred, donde la línea en negro muestra la distribución original de la demanda.

La estrategia de control 1 (línea roja) produce una altísima concentración de valores en los 100 kW correspondientes al límite  $P_{\max}$  (ver Tabla 8). La estrategia de Smoothing, por el contrario, apenas modifica la distribución de la original.

En lo que al SAE respecta, las líneas de trazo discontinuo representan los casos con mayor capacidad (caso B) y apenas suponen diferencia respecto a las de trazo continuo (caso A), quedando en su mayor parte solapadas por éstas. La línea morada denominada  $2B'$  se corresponde al caso  $2B_{\omega=8}$ .

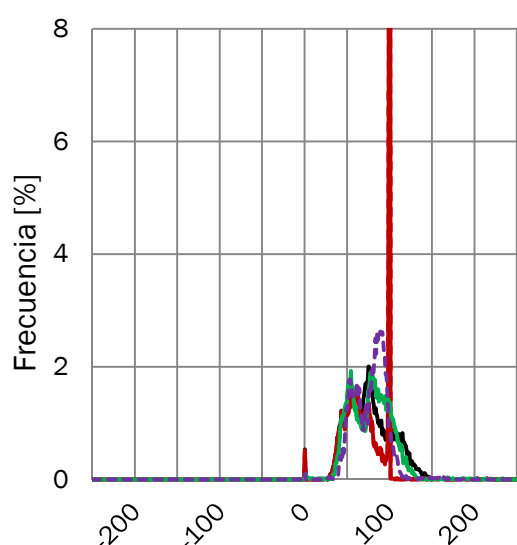


Fig. 35. Distribución de frecuencias de  $P_{cc}$ : no DG (kW)

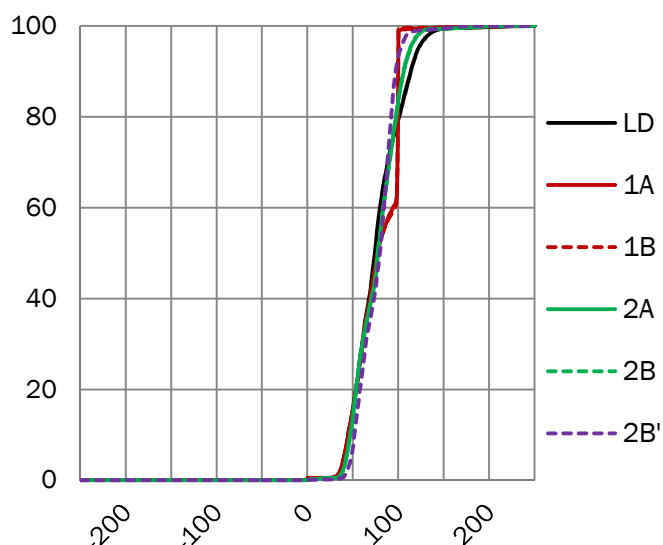


Fig. 36. Distribución acumulada de  $P_{cc}$ : no DG (kW)

La Tabla 24 recoge el número de veces que la potencia en PCC supera los 100 kW establecidos como  $P_{\max}$  por la estrategia *Peak Shaving*. Se observa que el almacenamiento instalado para el caso B ha podido recortar más picos que el instalado en A, aunque habría que estudiar si duplicar el tamaño y prestaciones de las baterías compensa económicamente para el poco margen de mejora que se consigue (un 2%).

Caso	Picos de demanda	Picos de generación	Total	Reducción %
Inicial	3642	0	3642	-
1A	146	0	146	95,99
1B	80	0	80	97.80

Tabla 24. Reducción de picos con Peak Shaving: no DG

### 3.5.1.2 Control de Q

Este caso de estudio no toma en consideración generación distribuida y, puesto que el control de Q se realiza a partir de ella, no tiene sentido hablar aquí de resultados propiamente dichos.

La Tabla 25 recoge los parámetros estadísticos de la demanda de Q y del f.d.p. original que presenta el área industrial estudiada (antes de aplicar ningún tipo de estrategia de control). El f.d.p. ha sido calculado mediante la ecuación (18):

$$f.d.p = \cos(\varphi) = \cos\left[\tan^{-1}\left(\frac{Q}{P}\right)\right] \quad (18)$$

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
$Q_{load}$	62.7	56.4	31.7	0	235.2
$\cos(\varphi)_{load}$	0.79	0.79	0.07	0	1

Tabla 25. Consumo de potencia reactiva (kVAr)

El valor mínimo de cero delata la existencia de periodos en los que hubo algún tipo de apagón y el suministro eléctrico quedó interrumpido.

### 3.5.1.3 Actuación de las baterías

Analizar el comportamiento de las baterías resulta clave para poder dimensionarlas de forma a. Si bien es cierto que cuanto más capacidad y potencia posean mejor funcionarán en general las estrategias, se debe encontrar un equilibrio razonable entre rendimiento y coste. Para ello se han recogido los valores del SOC para cada periodo y se han obtenido posteriormente los resultados que aparecen en la Tabla 26.

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
SOC 1A	85.0	90.7	14.8	5	100
SOC 1B	86.9	90.7	12.8	5	100
SOC 2A $\omega=4$	50.2	50.2	7.8	5	76.1
SOC 2B $\omega=4$	47.5	47.4	3.7	26.2	59.6
SOC 2B $\omega=8$	58.1	60.0	19.2	5	100

Tabla 26. Parámetros estadísticos del SOC (%) de las baterías: no DG

La media de SOC de la estrategia 1 es consecuencia directa de las consignas de SOC deseado de la Tabla 9. En ausencia de DG, la estrategia 1 procura mantener el SOC entre el 90 y el 100% de carga ( $SOC_{lower} = 90\%$ ,  $SOC_{upper} = 100\%$ ), para tener margen suficiente a la hora de recortar los picos de demanda.

El máximo y mínimo SOC alcanzados por el *Peak Shaving* se corresponden con los límites de la baterías; por lo tanto, se corrobora que en algún momento se ha aprovechado la capacidad total de éstas.

En lo que respecta a la estrategia 2, se observa un valor medio entorno al 50%, lo que era previsible si se tiene en cuenta el funcionamiento de la estrategia. De las tres simulaciones realizadas, la que presenta un mayor aprovechamiento del SAE es el caso B, con  $\omega = 8$ . Las baterías del caso A nunca llegan a cargarse del todo y se desaprovecha aproximadamente un cuarto de su capacidad. Este hecho se agrava sustancialmente al duplicar los parámetros dimensionales, manteniendo constante el ancho de ventana de la media móvil. Se puede concluir que, para sacar el máximo partido a las baterías, se deberá ajustar el parámetro  $\omega$  en consonancia con las dimensiones del SAE.

La gráfica de Fig. 37 aporta información adicional a la mostrada en la Tabla 8. La línea que presenta la distribución más uniforme es la morada (caso 2B'), cuyo SOC recorre uniformemente todo el rango de valores posibles.

Las líneas verdes correspondientes a la estrategia 2 con  $\omega = 4$  alcanzan el 100% de frecuencia bastante antes de llegar al estado de máxima carga (hay valores de SOC que no se alcanzan), mientras que las rojas exhiben, tal y como ya se ha comentado, una concentración de valores en el área de mayor SOC.

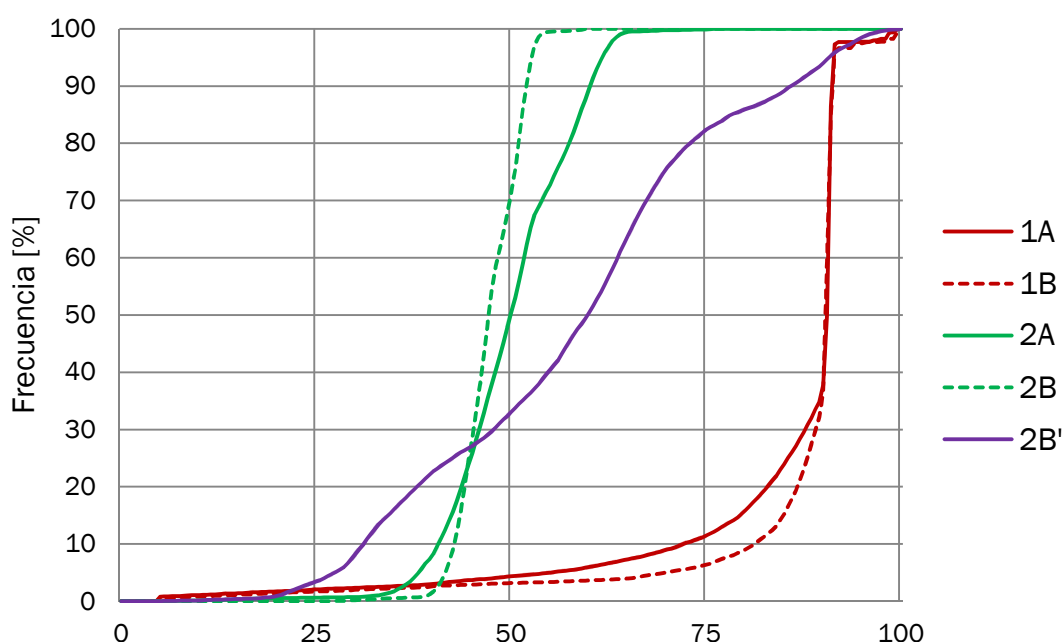


Fig. 37. Distribución acumulada del SOC (%) de las baterías: no DG

### 3.5.2 Microrred con DG

Los resultados que vienen a continuación provienen de simulaciones en las que sí se ha considerado generación distribuida. Concretamente, se ha considerado tres posibilidades:

- 1) Generación eólica (250 kW).
- 2) Generación fotovoltaica (250 kW).
- 3) Generación eólica + fotovoltaica (250 + 250 = 500 kW).

#### 3.5.2.1 Control de P

La utilidad de las estrategias de control de P se ve reforzada con la introducción de DG en la microrred. El recurso renovable es inconstante y, en cierto modo, imprevisible. El intercambio de potencia activa con la red va a variar enormemente, exigiendo respuestas rápidas por parte del control tendentes a minimizar las fluctuaciones.

#### Generación eólica

La Tabla 27 recoge los resultados conseguidos por las estrategias de P en un escenario con generación eólica:

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
P <sub>cc</sub> (inicial)	23.6	47.3	72.7	-189.1	246.5
P <sub>cc</sub> 1A	23.6	47.8	69.9	-187.9	233.4
P <sub>cc</sub> 1B	23.5	47.9	69.1	-155.1	224.1
P <sub>cc</sub> 2A <sub>w=4</sub>	23.5	42.8	65.4	-179.2	215.7
P <sub>cc</sub> 2B <sub>w=4</sub>	23.5	42.9	65.0	-179.2	201.3
P <sub>cc</sub> 2B <sub>w=8</sub>	23.5	44.1	65.0	-189.1	233.4

Tabla 27. Parámetros estadísticos de las estrategias de P: DG eólica (kW)

Lo primero que llama la atención respecto al escenario de la Tabla 23 (no DG) es el efecto que la generación provoca en media y mediana. Si en el anterior escenario el valor promedio de P<sub>cc</sub> se situaba próximo a 77 kW, ahora lo hace en torno a 23.5 kW.

La  $\sigma$  se mueve en dirección contraria y ve duplicado su valor, como resultado de combinar variaciones de la demanda con variaciones de la generación. Las estrategias de control consiguen reducir dichas variaciones en la medida que les es posible, siendo especialmente efectiva la estrategia *Smoothing*.

Los valores máximo y mínimo iniciales se ven ligeramente atenuados tras la ejecución de las estrategias, siendo reseñable el caso 1B. Aunque la estrategia 2 no tenga explicitado ningún objetivo en relación a los picos, el propio alisado de la curva de demanda los termina recortando. La estrategia 1, por el contrario, debe su razón de ser a la limitación de máximos y mínimos, y habrá que evaluar si dichos valores se consideran aceptables.

Observando la Fig. 38 se aprecian altas concentraciones de valores en los  $\pm 100$  kW que establece el *Peak Shaving* como límite. Estas concentraciones quedan reflejadas en la Fig. 39. Si se presta atención a la líneas rojas, tan sólo un pequeño porcentaje de valores se sitúan por debajo de -100 kW siendo, como era de esperar, ligeramente menor para la línea de trazo discontinuo (1B), correspondiente al caso de mayor almacenamiento.



La estrategia número 2 modifica la distribución de  $P_{cc}$  de una forma mucho más sutil: concentra los valores más extremos en puntos más intermedios.

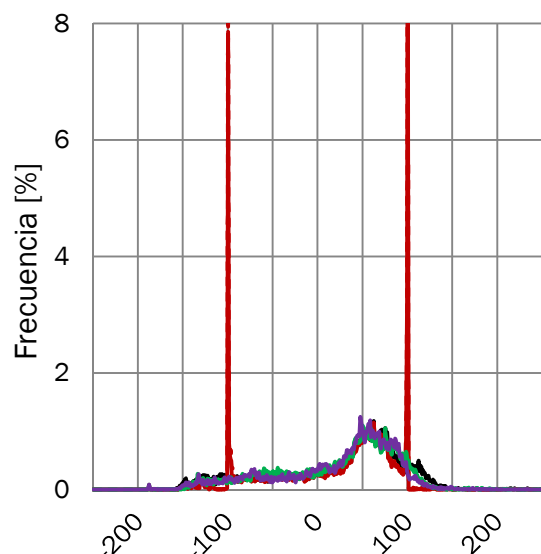


Fig. 38. Distribución de frecuencias de  $P_{cc}$ : DG eólica (kW)

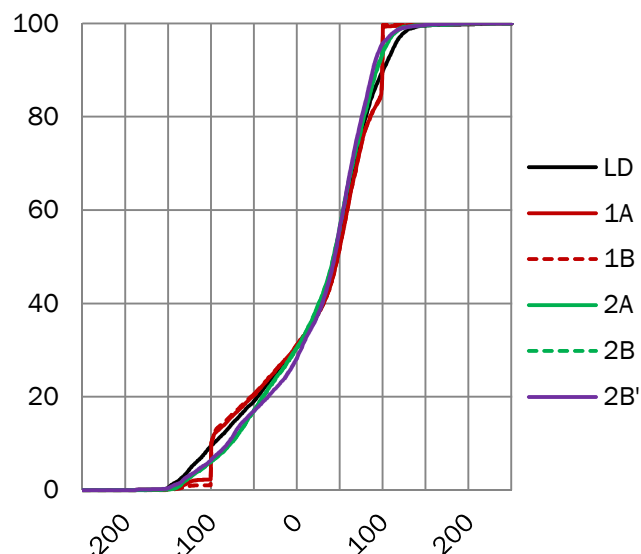


Fig. 39. Distribución acumulada de  $P_{cc}$ : DG eólica (kW)

### Generación fotovoltaica

Las conclusiones extraídas en el caso eólico son extrapolables al caso fotovoltaico, con la salvedad de que el recurso solar presenta un perfil más bajo y menos abrupto que el eólico (ver Fig. 14), tal y como se puede apreciar en los valores de media y mediana, y desviación estándar que aparecen en la Tabla 28.

Se constata que los máximos los sigue marcando la demanda, posiblemente coincidiendo con periodos de nula generación solar. Se observa asimismo que, para la estrategia 1, ningún valor mínimo sobrepasa el límite  $P_{min}$ ; ni siquiera el caso A, con menor almacenamiento. El caso  $2B_{\omega=8}$  vuelve a ser el que mejor minimiza la variabilidad de  $P_{cc}$ .

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
$P_{cc}$ (inicial)	50.0	62.1	50.1	-180.2	246.0
$P_{cc}$ 1A	50.0	62.4	48.7	-100.0	228.1
$P_{cc}$ 1B	50.0	62.4	48.3	-100.0	228.1
$P_{cc}$ 2A $_{\omega=4}$	50.0	57.7	39.4	-109.4	228.1
$P_{cc}$ 2B $_{\omega=4}$	50.0	57.7	39.2	-101.4	207.9
$P_{cc}$ 2B $_{\omega=8}$	49.9	51.8	30.3	-131.2	246.0

Tabla 28. Parámetros estadísticos de las estrategias de P: DG FV (kW)

A continuación se presentan los histogramas correspondientes al caso fotovoltaico. Las diferencias entre la distribución frecuencial del caso fotovoltaico y el caso eólico son muy pequeñas; si acaso destacar el ligero desplazamiento hacia la derecha que se da como consecuencia de una menor potencia generada.

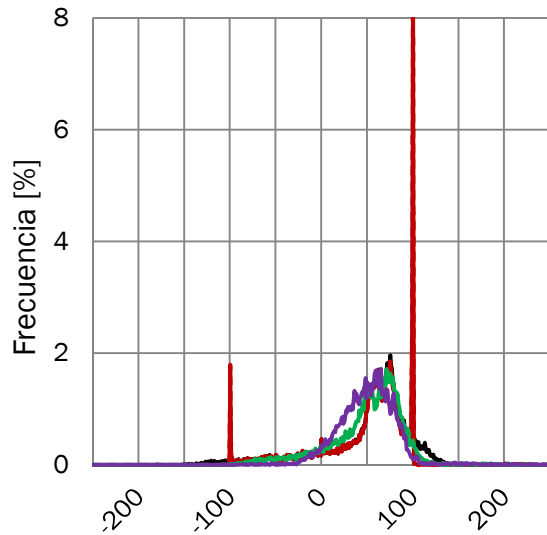


Fig. 40. Distribución de frecuencias de  $P_{cc}$ : DG fotovoltaica (kW)

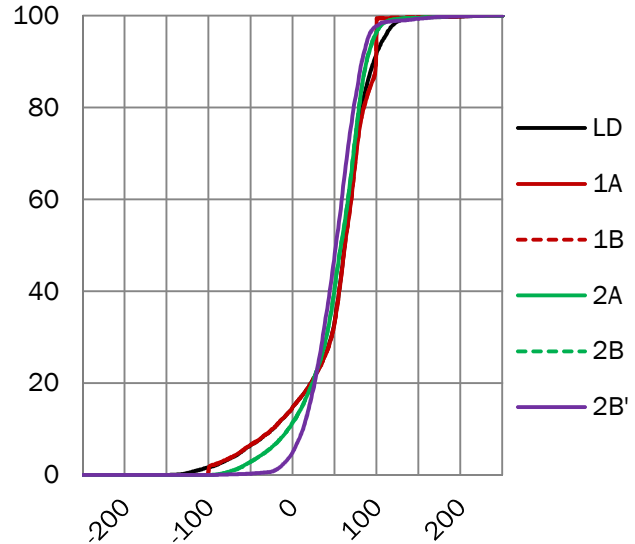


Fig. 41. Distribución acumulada de  $P_{cc}$ : DG fotovoltaica (kW)

### Generación eólica + fotovoltaica

Para finalizar el estudio de P se realizan otras cinco simulaciones, esta vez considerando la integración conjunta de generación eólica y fotovoltaica. Los parámetros estadísticos obtenidos aparecen en la Tabla 29.

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
$P_{cc}$ (inicial)	-8.8	16.8	89.1	-343.7	219.1
$P_{cc}$ 1A	-8.8	8.0	81.2	-337.7	215.9
$P_{cc}$ 1B	-8.8	7.7	79.5	-337.7	215.9
$P_{cc}$ 2A <sub><math>\omega=4</math></sub>	-8.8	9.8	77.7	-338.8	201.0
$P_{cc}$ 2B <sub><math>\omega=4</math></sub>	-8.8	9.4	75.4	-289.7	172.8
$P_{cc}$ 2B <sub><math>\omega=8</math></sub>	-8.8	9.8	74.1	-343.7	199.5

Tabla 29. Parámetros estadísticos de las estrategias de P: DG eólica + FV (kW)

El promedio de  $P_{cc}$  presenta valores negativos, es decir, se ha producido una cesión neta de energía de la microrred a la red; cesión de carácter intermitente si se presta atención al valor de la mediana. El motivo por el que el promedio es negativo se debe sobre todo a momentos puntuales de intensa generación, ya que el grueso de los valores, cuya mitad marca la mediana, se encuentra algo alejado del promedio.

Paradójicamente, el *Peak Shaving* (1) registra “peores” valores mínimos que el *Smoothing* (2). La razón es simple: si el pico que sobrepasa los límites de  $\pm 100$  kW se mantiene en el tiempo, llega un momento en que las baterías se quedan sin energía que ceder o absorber. El *Smoothing*, por el contrario, es menos exigente con la potencia instantánea que debe suministrar el almacenamiento y lo dosifica mejor.

La distribución de frecuencias de P de la Fig. 42 conserva el patrón de los casos anteriores, si bien exhibe una mayor acumulación en la zona negativa. El pico que

presenta la estrategia 1 (líneas rojas) en -100 kW presenta una enorme concentración de valores, mientras que su versión positiva registra un notable descenso.

El correcto funcionamiento de las estrategias se verifica en el histograma de frecuencia relativa acumulada de la Fig. 43. Las líneas rojas contienen muy pocos valores fuera del rango de  $\pm 100$  kW, siendo el caso B el que mejores resultados obtiene. La estrategia 2 vuelve a modificar la distribución original (línea negra) en todo el rango de potencias.

Si se analizan los resultados desde el punto de vista del SAE (caso A vs. caso B) sólo se aprecian diferencias sustanciales en la estrategia *Peak Shaving* (1). En la estrategia *Smoothing* (2) el dimensionamiento de las baterías ejerce menor influencia en el resultado, siendo más determinante el ancho de ventana de la media móvil  $\omega$ .

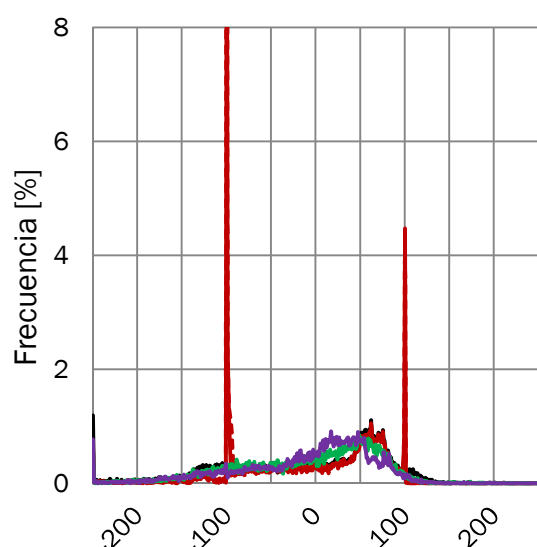


Fig. 42. Distribución de frecuencias de  $P_{cc}$ : DG eólica + fotovoltaica (kW)

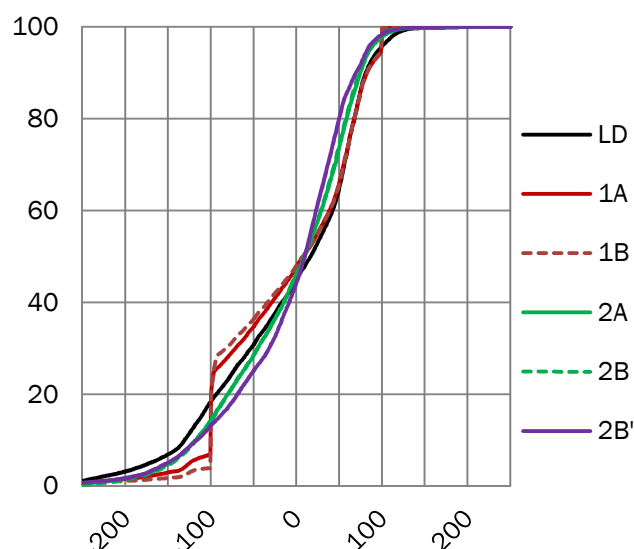


Fig. 43. Distribución acumulada de  $P_{cc}$ : DG eólica + fotovoltaica (kW)

### Resumen del recorte de picos

La Tabla 30 presenta el cómputo anual de los picos registrados en la curva de  $P_{cc}$  cuando se ejecuta la estrategia *Peak Shaving*, para cada una de las situaciones de DG.

Un periodo en el que, de media, la microrred ha absorbido más de 100 kW de la red se considera pico de demanda, y un periodo en el que la microrred ha cedido más de 100 kW a la red se considera pico de generación (-100 kW). Conforme la penetración de la DG aumenta, los picos de generación suceden más frecuentemente, en detrimento de los de demanda.

En los casos de DG eólica/DG fotovoltaica la efectividad en la supresión de picos ronda el 90%, es decir, las baterías han sido capaces de equilibrar el balance con la red según lo estipulado para nueve de cada diez picos. Cuando se suman ambas DG, el valor baja hasta un 68% de reducción, para un SAE de 100kW/1MWh (caso A); y hasta un 82% para el caso de 200kW/2MWh (caso B).

Caso	Picos de demanda	Picos de generación	Total	Reducción %
<b>DG eólica</b>				
Inicial	1796	1660	3456	-
1A	125	403	528	84.72
1B	51	182	233	93.26
<b>DG fotovoltaica</b>				
Inicial	1476	293	1769	-
1A	125	0	125	92.93
1B	105	0	105	94.06
<b>DG eólica + fotovoltaica</b>				
Inicial	752	3228	3980	-
1A	57	1213	1270	68.09
1B	30	696	726	81.76

Tabla 30. Reducción de picos con Peak Shaving: microrred con DG (kW)

### 3.5.2.2 Control de Q

Con la inclusión de la DG entra en juego el control de Q. La Tabla 32 recopila los parámetros estadísticos de los valores de f.d.p. vistos desde el lado de red para los casos de un único tipo de DG, mientras que la Tabla 32 los recopila para el caso agregado.

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
$\text{Cos}(\varphi)_{\text{load}}$	0.79	0.79	0.07	0	1
<b>DG eólica</b>					
$\text{Cos}(\varphi)$ 1A	0.92	1	0.11	0	1
$\text{Cos}(\varphi)$ 1B	0.92	1	0.12	0	1
$\text{Cos}(\varphi)$ 2A $\omega=4$	0.92	1	0.13	0.01	1
$\text{Cos}(\varphi)$ 2B $\omega=4$	0.92	1	0.13	0.01	1
$\text{Cos}(\varphi)$ 2B $\omega=8$	0.91	1	0.16	0	1
<b>DG fotovoltaica</b>					
$\text{Cos}(\varphi)$ 1A	0.90	0.90	0.11	0	1
$\text{Cos}(\varphi)$ 1B	0.90	0.90	0.11	0	1
$\text{Cos}(\varphi)$ 2A $\omega=4$	0.87	0.89	0.16	0	1
$\text{Cos}(\varphi)$ 2B $\omega=4$	0.87	0.89	0.16	0	1
$\text{Cos}(\varphi)$ 2B $\omega=8$	0.83	0.88	0.2	0	1

Tabla 31. Parámetros estadísticos del control de Q (f.d.p.): Eólica / FV

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
<b>DG eólica + fotovoltaica</b>					
Cos( $\phi$ ) 1A	0.96	1	0.1	0	1
Cos( $\phi$ ) 1B	0.96	1	0.09	0	1
Cos( $\phi$ ) 2A <sub><math>\omega=4</math></sub>	0.94	1	0.14	0	1
Cos( $\phi$ ) 2B <sub><math>\omega=4</math></sub>	0.94	1	0.14	0	1
Cos( $\phi$ ) 2B <sub><math>\omega=8</math></sub>	0.92	1	0.19	0	1

**Tabla 32. Parámetros estadísticos del control de Q (f.d.p): Eólica + FV**

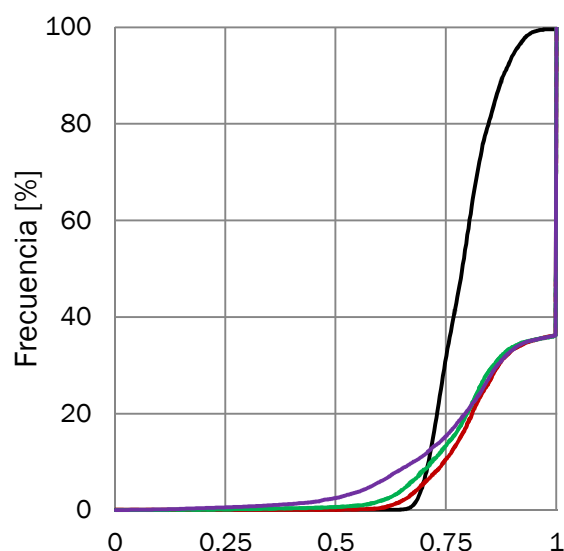
El f.d.p se ha calculado mediante la ecuación (18). Esto quiere decir que para un mismo flujo de potencia reactiva en PCC, el Cos( $\phi$ ) variará según lo haga el flujo de potencia activa. Debido a este hecho se registran variaciones entre la estrategia 1 y 2, pese a que ambas implementan el mismo control de Q.

De entre las tres situaciones de DG simuladas, la que mejores resultados proporciona es la que incluye recurso eólico y fotovoltaico, lo que no es de extrañar teniendo en cuenta que a más recurso disponible, mayor capacidad para aportar Q al sistema.

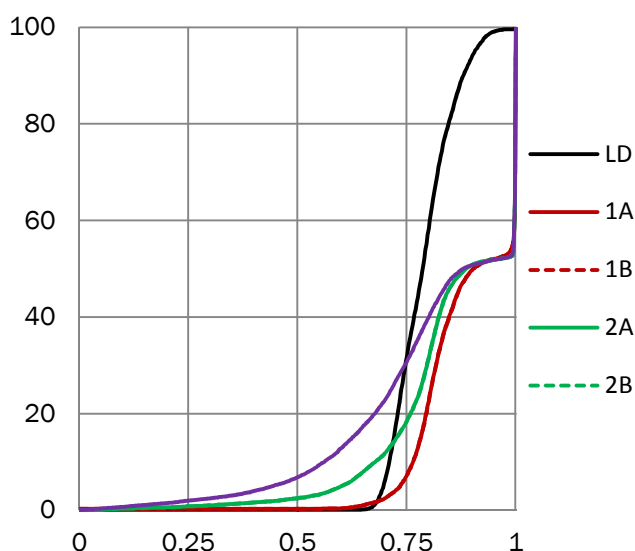
Independientemente de la estrategia de P escogida, el control de Q mejora notable el f.d.p original en todas las situaciones estudiadas. El Cos( $\phi$ ) promedio para las dos situaciones con penetración eólica es muy superior al 0.79 que presenta originalmente la demanda.

En ambas situaciones la mediana se sitúa en 1, lo que significa que en al menos el 50% de las ocasiones el valor del f.d.p registrado ha sido igual a la unidad.

Las siguientes tres figuras muestran la distribución acumulada del factor de potencia visto desde el lado de red. El f.d.p inicial (línea negra) registra casi un 40% de valores por debajo del 0.75, dato que mejora sustancialmente tras la adición del control de Q.



**Fig. 44. Distribución acumulada del f.d.p: DG eólica**



**Fig. 45. Distribución acumulada del f.d.p: DG fotovoltaica**

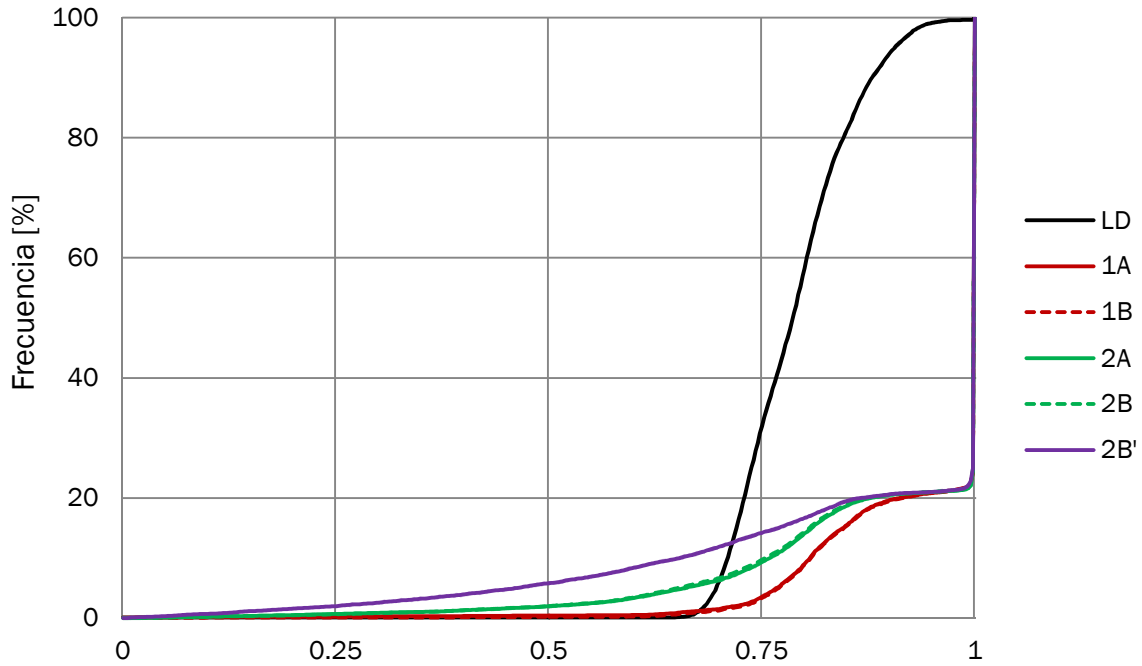


Fig. 46. Distribución acumulada del f.d.p. DG eólica + fotovoltaica

En la Fig. 44 se observa que en más del 60% de las ocasiones se ha registrado un  $\cos(\varphi)$  igual a la unidad. La Fig. 45 (recurso solar) también mejora notablemente el f.d.p. inicial, aunque en menor medida debido a las razones anteriormente mencionadas.

Tal y como se preveía, los mejores resultados se obtienen en la microrred con más potencia renovable instalada (Fig. 46), donde en el 80% de los casos se registra un f.d.p. igual a uno.

### 3.5.2.3 Actuación de las baterías

Se pasa ahora a evaluar el impacto que la generación distribuida y las estrategias de control tienen sobre el estado de carga de las baterías. Para ello se ha recopilado en la Tabla 33 los principales parámetros estadísticos obtenidos a partir de los datos de SOC, agrupados según la DG considerada en cada situación.

La microrred con mayor penetración de energías renovables hace un uso más intensivo de las baterías. Con la excepción de los casos  $2B_{\omega=4}$  de eólica y de fotovoltaica, el SOC recorre todo el rango de valores que puede tomar (del 5 a 100%).

Se puede observar que la desviación estándar ( $\sigma$ ) del caso que contempla DG combinada es notablemente superior al resto de situaciones, es decir, permanece menos tiempo estático o varía de forma más abrupta.

En lo que respecta a las estrategias de P, el caso  $2B_{\omega=8}$  vuelve a ser el de mayor exigencia, destacando una  $\sigma$  de 36.7 kW para el caso eólico.

Variable	Media	Mediana	DS ( $\sigma$ )	Mínimo	Máximo
<b>DG eólica</b>					
SOC 1A	45.9	43.7	13.7	5	100
SOC 1B	45.7	44.0	11.2	5	100
SOC 2A $\omega=4$	38.5	34.2	21.8	5	100
SOC 2B $\omega=4$	45.9	42.8	10.2	18.0	77.7
SOC 2B $\omega=8$	44.7	39.6	36.7	5	100
<b>DG fotovoltaica</b>					
SOC 1A	43.2	42.2	6.0	5	62.8
SOC 1B	43.5	43.5	5.5	5	55.7
SOC 2A $\omega=4$	48.7	47.1	13.8	5	100
SOC 2B $\omega=4$	51.8	50.6	6.2	26.9	74.9
SOC 2B $\omega=8$	44.0	42.1	19.3	5	100
<b>DG eólica + fotovoltaica</b>					
SOC 1A	45.70	38.81	19.68	5	100
SOC 1B	44.90	38.78	18.22	5	100
SOC 2A $\omega=4$	47.6	47.9	28.9	5	100
SOC 2B $\omega=4$	51.0	49.2	30.1	5	100
SOC 2B $\omega=8$	44.1	40.9	25.3	5	100

**Tabla 33. Parámetros estadísticos del SOC (%) de las baterías: microrred con DG**

Las tres figuras de la siguiente página ilustran la distribución de valores de SOC, agrupados nuevamente según la DG considerada.

La distribución de la estrategia *Peak Shaving* (línea roja) exhibe una elevada concentración de datos en torno al intervalo marcado por las rutinas automáticas de carga de la Tabla 9. En los casos de un único recurso renovable se sitúa alrededor del 45% de SOC, mientras que para el caso de mayor penetración de DG el valor baja un 10% aproximadamente.

En lo que respecta al *Smoothing* (líneas verdes y morada) el perfil de la distribución obedece a una combinación de dos factores: el ancho de ventana de la media móvil,  $\omega$ ; y las propias dimensiones del SAE.

El caso 2B sigue infrautilizando las baterías, puesto que dispone de demasiada capacidad y potencia para las consignas que obtiene cada vez que calcula la media móvil de los últimos cuatro valores de  $P_{cc}$  ( $\omega=4$ ).

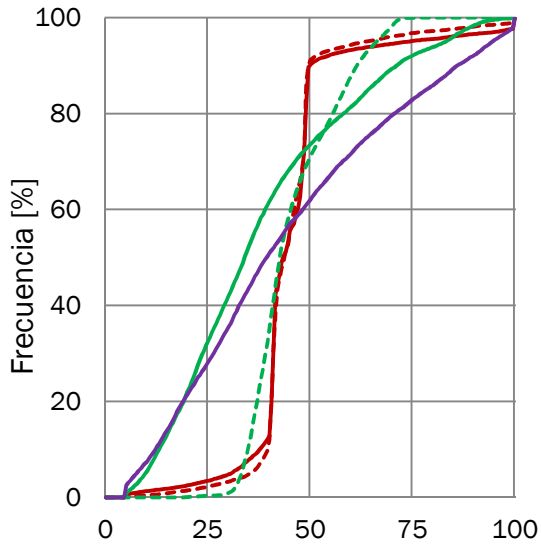


Fig. 47. Distribución acumulada del SOC (%): DG eólica

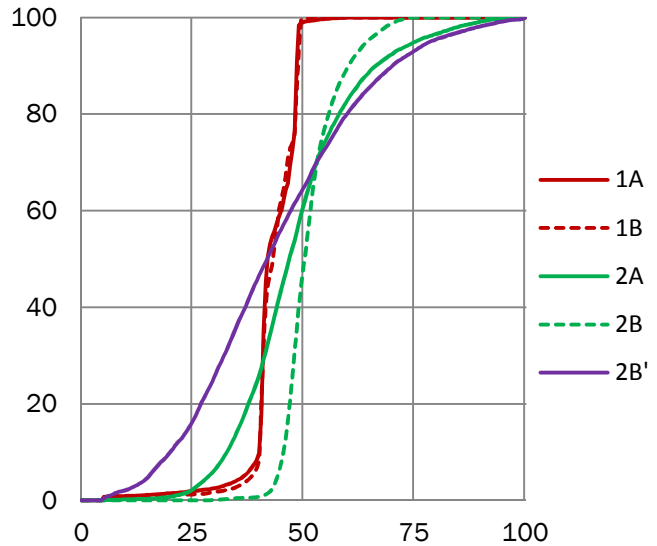


Fig. 48. Distribución acumulada del SOC (%): DG fotovoltaica

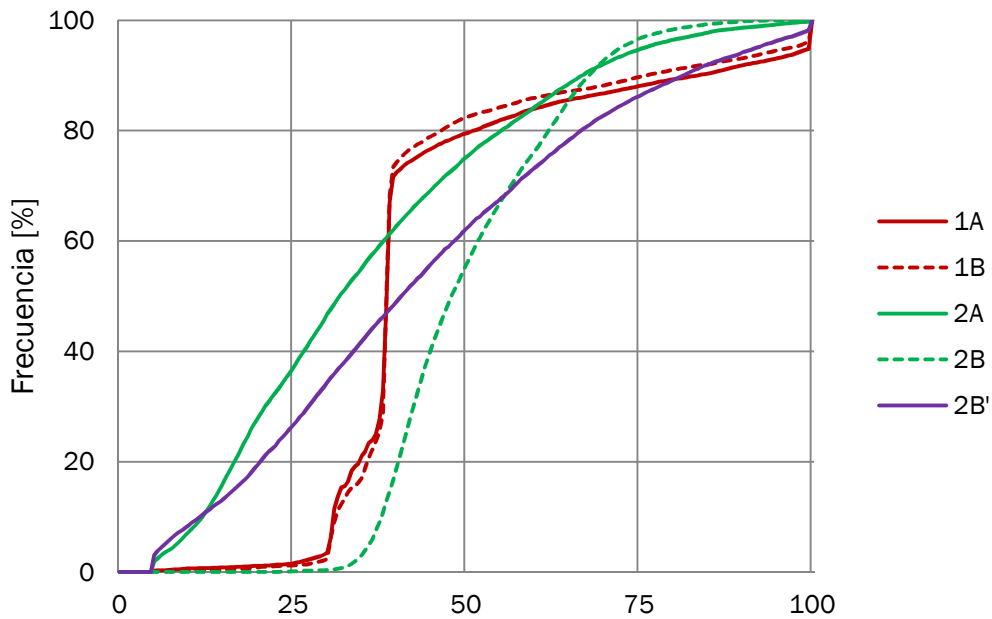


Fig. 49. Distribución acumulada del SOC (%): DG eólica + fotovoltaica

Llegados a este punto, puede resultar interesante aglutinar en una misma gráfica las diferentes combinaciones de DG y de almacenamiento, para cada una de las estrategias.

La Fig. 50 aglutina todas estas combinaciones para la estrategia 1 (*Peak Shaving*). En ella se distinguen muy claramente los diferentes intervalos de SOC deseado de las rutinas automáticas recogidas en la Tabla 9: 90-100% en ausencia de DG, 40-50% en caso de que la microrred cuente con un único recurso renovable y 30-40% cuando se consideren ambos recursos.

Las diferencias entre el caso A y el B son muy pequeñas.



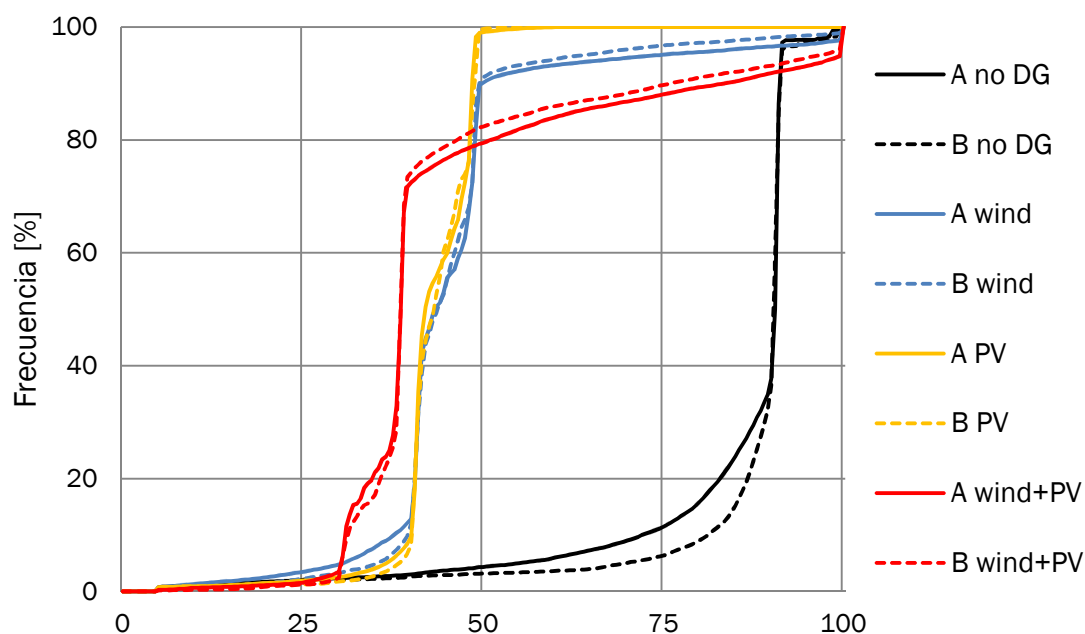


Fig. 50. Distribución acumulada del SOC (%): Peak Shaving

La Fig. 51 hace lo propio para la estrategia de alisado por media móvil o Smoothing. La penetración de la DG vuelve a ser clave: de las tres líneas negras correspondientes a una microrred sin DG, en dos de ellas ni siquiera se recorren todos los valores posibles de SOC, mientras que para el caso de más DG (líneas rojas) la distribución se equilibra hasta asemejarse a una diagonal que barre todo el rango de la variable.

Se concluye que a más generación, mejor aprovechamiento hace la estrategia *Smoothing* del SAE y, para un mismo nivel de generación, el factor  $\omega$  juega un papel decisivo.

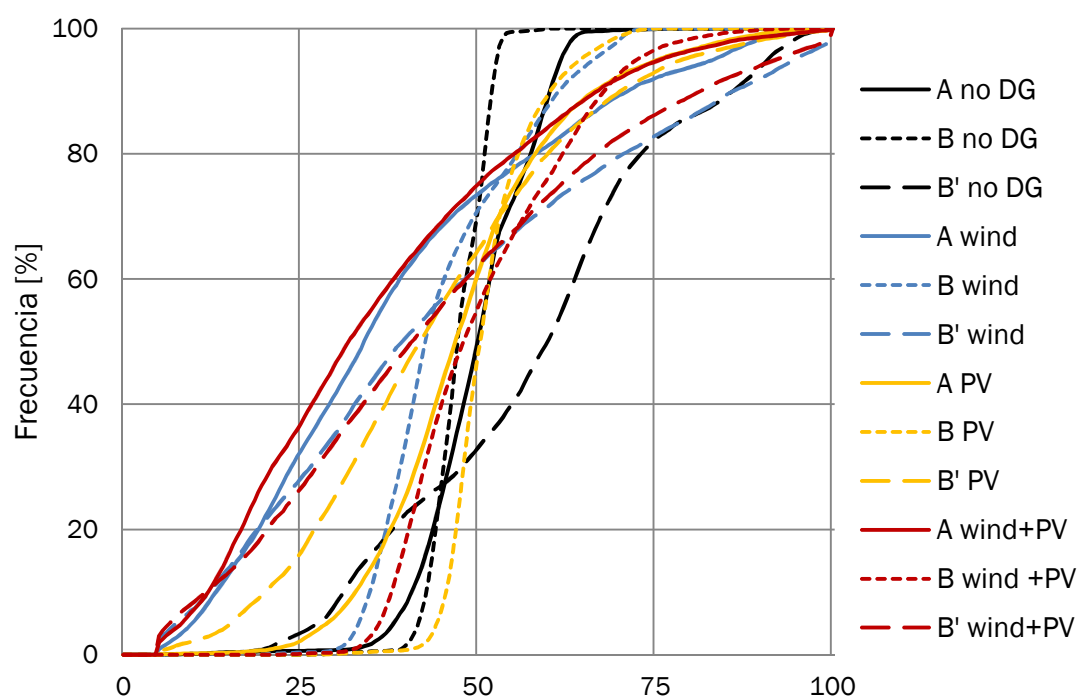


Fig. 51. Distribución acumulada del SOC (%): Smoothing

## 3.6 Conclusiones y líneas futuras

En este capítulo se ha expuesto el trabajo realizado para el proyecto OptimaGrid, consistente en la realización de un emulador de microrredes mediante un sistema multiagente. El SMA ha implementado un control de potencia activa y reactiva, cuya efectividad ha sido evaluada con un total de 20 simulaciones en las que se han empleado series temporales de datos reales.

El apdo. 3.1 ha repasado brevemente el trabajo previo desde el que se partía, y ha definido las especificaciones y funcionalidades con las que debía de contar el emulador. En el apartado 3.2 se ha realizado un análisis estadístico de las series temporales utilizadas. El apartado 3.3 contiene la explicación detallada del modelo y de su funcionamiento. En el apdo. 3.4 se pueden encontrar aclaraciones adicionales acerca de la arquitectura interna del SMA, así como una recopilación de las clases Java implementadas para la ocasión. Los resultados de las simulaciones se recogen en el apartado 3.5, acompañado del correspondiente análisis estadístico.

### 3.6.1 Conclusiones

A continuación se expondrán las conclusiones obtenidas, tanto las relativas al emulador en sí como a las referentes al modelo de simulación.

#### 3.6.1.1 Emulador

En vista de los resultados obtenidos, se concluye que:

- La arquitectura de los SMA resulta válida a la hora de emular el control distribuido de una microrred.
- La estructura de clases escogida se ajusta perfectamente a cualquier microrred cuya configuración combine DG, almacenamiento y cargas.
- La interacción entre agentes mediante el protocolo FIPA Contract Net asegura la interoperabilidad del SMA, al obligar al desarrollador a implementar el proceso de negociación de una forma muy concreta.
- La decisión de asignar o no un agente a un elemento de la microrred es delicada, y debe tomarse tras estudiar sus pros y contras. En las ocasiones en que la utilización de un agente resulte excesiva, se podrá optar por modelar el elemento mediante un objeto Java “normal”; como se ha hecho con las baterías y convertidores de la microrred.
- El modo *simulación* permite utilizar datos reales para reproducir gran variedad de situaciones y produce información útil que ayuda en la tarea de dimensionar de forma óptima el sistema de almacenamiento eléctrico.
- El modo *monitorización* es viable mediante comunicación serie.

#### 3.6.1.2 Modelo simulado

A partir de las simulaciones llevadas a cabo, se puede concluir que:

- La estrategia de control 1 (*Peak Shaving*) resulta eficaz a la hora de recortar los picos de demanda pero, al combinarla con generación distribuida, es necesario establecer los valores adecuados de  $P_{\max}$  y  $P_{\min}$ .

- La estrategia de control 2 (*Smoothing*) resulta eficaz a la hora de alisar la curva de demanda, siendo el parámetro  $\omega$  (ancho de ventana de la media móvil) determinante a la hora de lograr un mayor o menor grado de alisamiento.
- La elección de la potencia y capacidad del SAE dependerá de la estrategia que se tenga previsto emplear para gestionarlo. Si se dimensiona para la estrategia 1, habrá que prestar especial atención a la relación entre el coste y el porcentaje de reducción de picos. La estrategia 2 admite un amplio abanico de posibilidades en este aspecto. A la luz de los resultados se concluye que, primero se debe seleccionar el grado de alisamiento mediante el parámetro  $\omega$  y luego seleccionar las dimensiones del SAE en consecuencia, es decir, a más  $\omega$ , baterías más grandes y potentes.
- La generación distribuida reduce el consumo de potencia activa de red, lo que obliga a implementar también un control de reactiva si se quiere evitar factores de potencia próximos a cero.
- En una microrred con alta penetración de energías renovables, los picos de generación resultan más críticos que los de consumo.
- Ninguna de las estrategias asegura por sí sola una protección total de la microrred frente a picos potencialmente peligrosos. Aunque con la estrategia 1 es teóricamente factible, sería necesario sobredimensionar el sistema de almacenamiento hasta el punto de hacerlo económicamente inviable. En lo que respecta a la estrategia 2, la atenuación de picos es un efecto colateral, no un objetivo en sí. Se propone por lo tanto combinar baterías con elementos de protección.
- El recurso renovable eólico ofrece una gran potencia, pero de manera muy intermitente. El recurso fotovoltaico provee de menos potencia, aunque de manera más estable. Por lo tanto, es preferible integrar de manera conjunta eólica y solar antes que sólo eólica. Asimismo, se recomienda limitar la potencia eólica en origen.

### 3.6.2 Líneas futuras

Se proponen ahora algunas ideas para continuar desarrollando tanto el emulador como el modelo escogido para las simulaciones.

#### 3.6.2.1 Emulador

Al emulador implementado le queda todavía un largo camino por recorrer hasta su versión definitiva, especialmente en lo que respecta al modo *monitorización*. Algunas de las posibles mejoras serían:

- El emulador actual sólo puede gestionar un nodo. Con el modelo actual esto no supone ningún problema porque todos los elementos que componen la microrred se suponen conectados a un único nodo (ver Fig. 18). Para poder emular microrredes de varios nodos se plantea la posibilidad de un nuevo agente: el agente *Node*, que se encargaría de recoger la información de los elementos conectados a un determinado punto de la microrred. El control se llevaría a cabo mediante la interacción de todos los nodos participantes y *PowerManager*.
- Implementación de los agentes propuestos en la Tabla 10 (en gris), mediante la extensión de las que están actualmente implementados. Por ejemplo, los agentes *SolarPV*, *Wind* y *MicroHydro* serían “hijos” de la clase *DG*. El agente *Load* podría

extenderse para distinguir entre cargas prioritarias y no prioritarias. Cargas prioritarias serían las que no admiten desconexión por parte del control, mientras que las no prioritarias sí podrían ser desconectadas por Node o PowerManager en caso de necesidad.

- El punto de conexión a red (PCC) podría dejar de considerarse un ente “pasivo” y contar con un agente propio (agente Grid).
- Incluir tensiones y corrientes, además de flujos de P y Q, para dotar de un mayor nivel de realismo al emulador.
- En una microrred multinodal: desarrollar las clases *TransmissionLine* e *Impedance* para realizar las conexiones entre nodos.
- Realización de GUIs para cada agente, en las que poder especificar potencia nominal del elemento asociado, modo de funcionamiento, nombre del archivo desde el que se leerán los datos en modo *simulación*, etc.
- Desarrollo del modo *monitorización*. Evaluar distintas posibilidades: ¿es preferible tener un agente dedicado exclusivamente a gestionar el puerto serie o que cada agente tenga implementada esa función dentro de su código?
- Estudiar el empleo del protocolo de comunicación serie Modbus para dotar de mayor robustez e interoperabilidad al SMA funcionando en modo *monitorización*.



**Fig. 52. Servidores adquiridos por CIRCE para el emulador**

Para finalizar, se presenta en la Fig. 53 el esquema de una hipotética futura implementación del emulador de microrredes.

En la parte superior se sitúan las tres fases de la red, sin neutro, con 400 V de valor eficaz entre cada una de ellas. La microrred se conecta a la red por medio de un transformador y de un convertidor AC-AC con enlace por corriente continua compuesto por dos VSC.

En la parte de inferior se ha representado el esquema de conexión entre las DSP y los ordenadores en los que se ejecutarían los agentes.

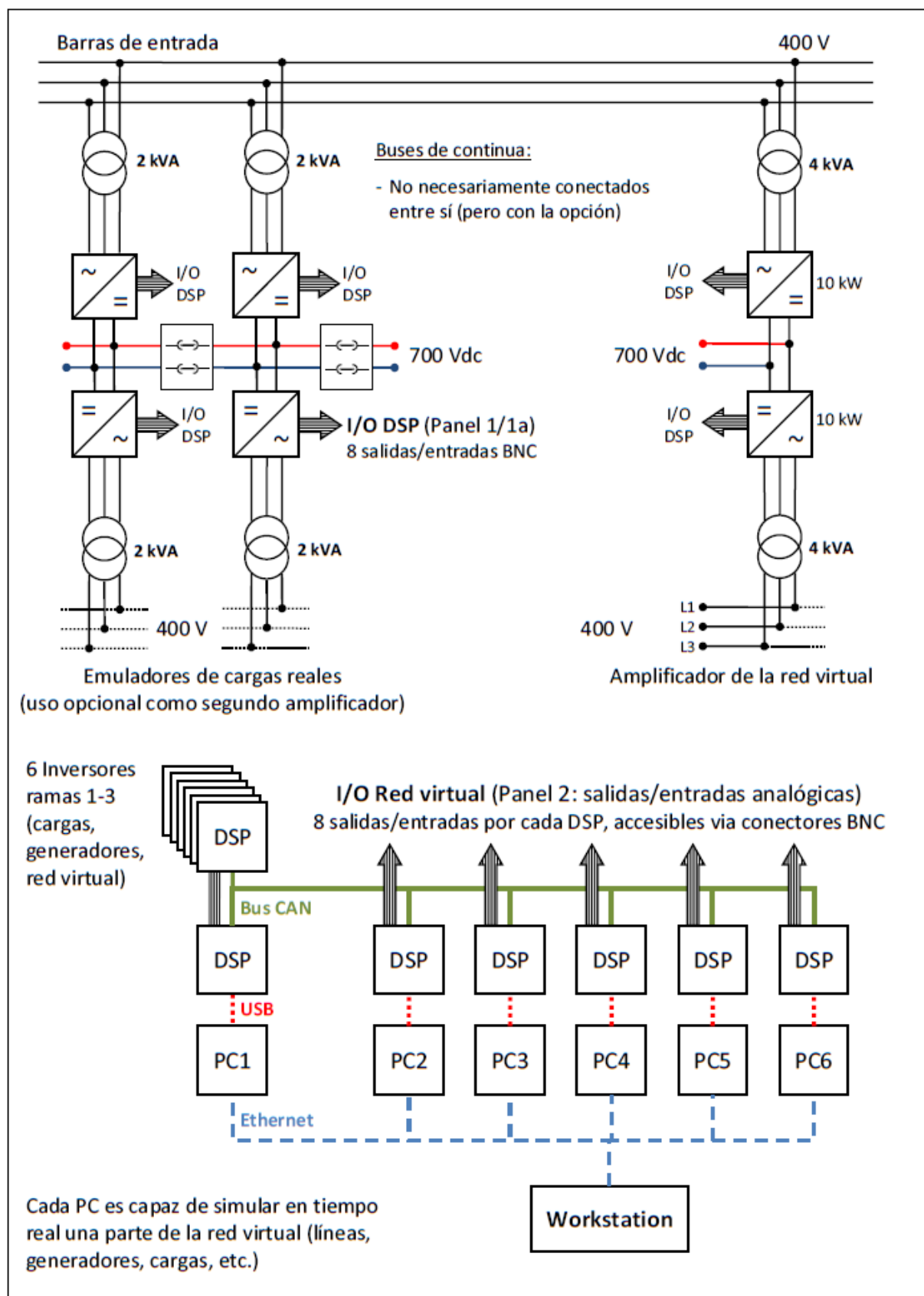


Fig. 53. Emulador de la microrred: Power Hardware in the Loop (PHIL)

### 3.6.2.2 Modelo simulado

En lo que respecta al modelo en cuestión, se proponen como líneas de continuidad los siguientes puntos:

- La estrategia *Peak Shaving* podría refinarse para dosificar mejor la energía almacenada. En el modelo actual se destina siempre el 100% de los recursos a la supresión de picos, independientemente de su magnitud y/o duración. Esto puede llegar a suponer un problema en según qué circunstancias. Por ejemplo, si se produce un pico de poca magnitud pero de duración suficiente para descargar totalmente las baterías, la estrategia será incapaz de reaccionar antes nuevos repuntes de demanda, que quizás sean mucho más dañinos para el sistema que los eliminados previamente. Para evitar esta situación se podría incluir como parámetro la derivada de la curva de demanda en punto, o variar los límites  $P_{\min}$  y  $P_{\max}$  según un modelo más o menos elaborado de previsión.
- La estrategia *Smoothing* podría tener en cuenta la evolución de la media móvil para ajustar el parámetro  $\omega$  en tiempo real, dentro de un intervalo marcado por las dimensiones del sistema de almacenamiento eléctrico instalado.
- Creación de nuevas estrategias de P: control por programa horario. La mecánica es muy sencilla: en las horas del día en las que la energía es más barata (habitualmente de noche) se efectúa la recarga de las baterías y en las de mayor consumo y precio se descargan. El horario se podría ajustar en función de si se cuenta con placas fotovoltaicas, del día de la semana, etc.
- Tanto el modelo de la batería como el del convertidor pueden ser mejorados. La corriente de salida de la batería  $i_{\text{out}}$  (y por consiguiente  $P_{\text{out}}$ ) podría variar en función del SOC, de forma que cuando éste se encontrara en torno a valores extremos la corriente decayera.

## 4 Optimización clásica

En el presente capítulo se va a abordar el funcionamiento de una microrred monofásica de corriente alterna de baja tensión (230 V) cuyo algoritmo de control realiza una optimización mediante programación lineal que minimiza el coste económico del balance energético.

La microrred se compone de un generador eólico, un grupo electrógeno diésel y dos baterías de almacenamiento; dichos elementos están conectados a través de una línea de distribución a varias cargas sobre las que no se tiene control alguno: una fija conocida y otras de naturaleza aleatoria. Para su regulación y monitorización se dispone de dos elementos adicionales: agentes Optimizadores y Reguladores Inteligentes.

El trabajo llevado a cabo en este capítulo parte de un modelo Simulink realizado por varios investigadores del departamento de IER de CIRCE. Este modelo incorpora el código de los Reguladores Inteligentes, y fue desarrollado para el proyecto Smart City. Se propone ahora utilizar dicho modelo para que un agente en el entorno JADE, el agente Optimizador, ejecute en él la consabida optimización del despacho económico de potencia.

El capítulo se estructura de la siguiente manera: en el apartado 4.1 se caracterizarán los elementos que forman el modelo de la microrred, el 4.2 explicará el algoritmo de control utilizado, mientras que el apartado 4.3 versará sobre su implementación informática. Los diferentes escenarios simulados, así como sus respectivos resultados, aparecen recogidos en el apartado Fig. 61. Finalmente, el apartado 4.5 contiene un resumen del capítulo y sugerencias para desarrollos futuros.

### 4.1 Caracterización del modelo

Se pretende diseñar un sistema de control que asegure el funcionamiento óptimo de la microrred a distintos niveles y atendiendo a múltiples objetivos al mismo tiempo: regulación de potencia activa y reactiva, y minimización del coste del balance energético. La microrred dispone para ello de dos nuevos elementos: el agente Optimizador y los Reguladores Inteligentes, que son adicionales a los existentes en una red de distribución de baja tensión convencional y cuya estructura jerárquica se muestra en la Fig. 54.

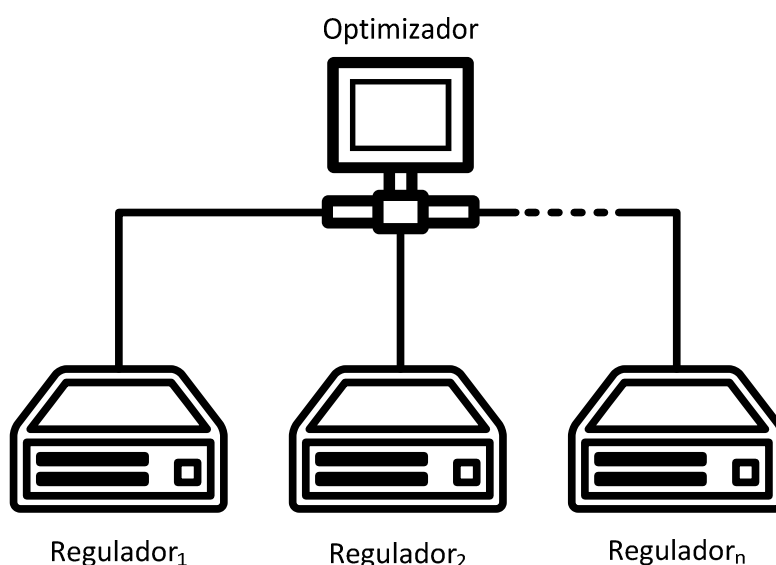


Fig. 54. Jerarquía del control

### 4.1.1 Reguladores Inteligentes

En el nivel más bajo de la estructura organizacional del control se encuentran los Reguladores, encargados del control distribuido de la potencia activa y reactiva que cada elemento generador intercambia con la microrred. La técnica elegida ha sido la *p-q droop*, válida tanto para redes conectadas como para redes aisladas.

El parámetro que define el control de potencia activa es la recta que relaciona potencia y frecuencia, también denominada recta estática potencia-frecuencia, que se corresponde con la gráfica izquierda de la Fig. 55. Esta curva establece la variación de frecuencia del convertidor, en fuente de tensión (Voltage Source Converter, VSC), para una variación de potencia activa dada.

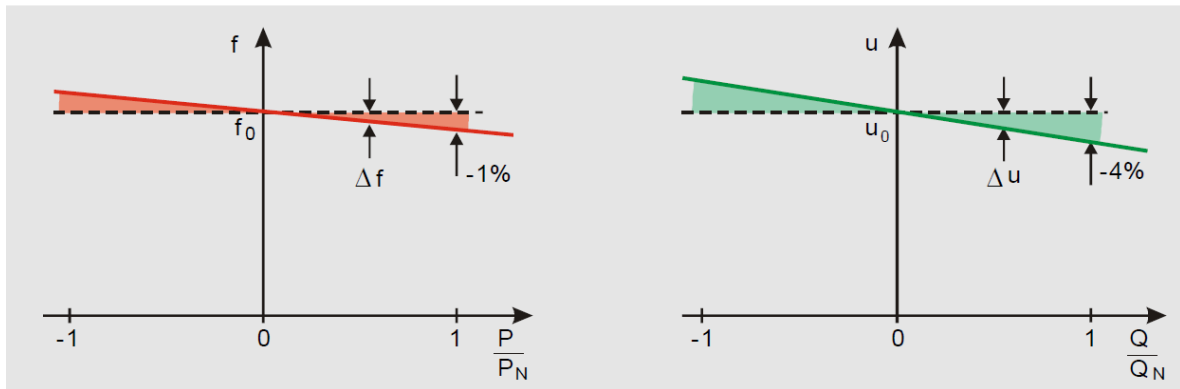


Fig. 55. Droops de frecuencia y voltaje [25]

El Regulador va a controlar la potencia de salida del VSC mediante pequeñas variaciones de frecuencia. Estas pequeñas variaciones de frecuencia, de carácter transitorio, modifican el desfase entre la tensión entregada y la tensión de red. Según exista un desfase u otro, el flujo de potencia activa de la red al generador cambiará, pudiéndose ajustar al deseado.

El esquema de conexión de la Fig. 56 ilustra la topología de la conexión. El Regulador controla  $V\angle\delta_v$ , tanto en magnitud como en fase, por lo que es capaz de modificar el desfase entre  $V\angle\delta_v$  y  $E\angle\delta_E$  para regular el flujo de potencia activa entre el generador o batería y el bus AC de la microrred.

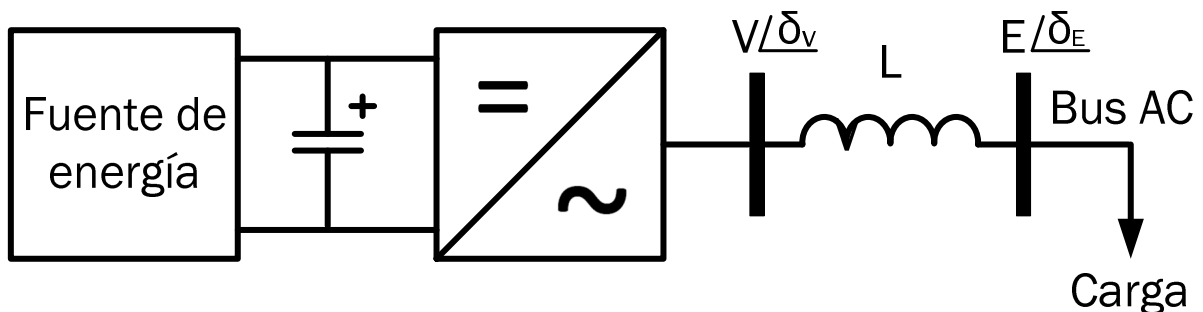


Fig. 56. Esquema de conexión a la microrred



La potencia activa cedida por el Regulador se puede obtener mediante la ecuación (19).

$$P = \frac{VE}{\omega L} \sin(\delta_v - \delta_e) \quad (19)$$

La potencia reactiva exhibe un comportamiento íntimamente ligado a la diferencia de potencial existente entre la salida del convertidor y la microrred. Para su control, el Regulador modificará ligeramente la amplitud del voltaje en bornes de los VSC, pero manteniéndola siempre dentro del margen de seguridad establecido.

La ecuación (20) rige la inyección de Q en dicha configuración:

$$Q = \frac{V^2}{\omega L} - \frac{VE}{\omega L} \cos(\delta_v - \delta_e) \quad (20)$$

La Tabla 34 presenta las cuatro formas en las que puede fluir la potencia, en función de cuál sea la relación entre las fases y magnitudes de V y E.

Relaciones de V y $\delta$	Flujo de potencia	
	VSC $\longleftrightarrow$ AC BUS	
$\delta_v > \delta_e$	P	$\longrightarrow$
$\delta_v < \delta_e$	P	$\longleftarrow$
$V > E$	Q	$\longrightarrow$
$V < E$	Q	$\longleftarrow$

Tabla 34. Principios de funcionamiento: VSC conectado a red

#### 4.1.2 Agente Optimizador

Un nivel jerárquico por encima de los Reguladores se encuentra el agente Optimizador, máxima autoridad dentro de la microrred. El Optimizador ejecuta de forma periódica un algoritmo que tiene por objeto lograr que el coste del balance energético en la microrred sea óptimo. También se encarga de hacer llegar las consignas optimizadas de potencia activa y reactiva a los Reguladores.

Con el algoritmo utilizado, el Optimizador realiza dicha optimización de la gestión de la demanda para los siguientes  $n$  periodos (cuatro en este caso). Para ello se vale de un bloque predictor de demanda, que toma como referencia los  $n$  periodos inmediatamente anteriores. El algoritmo no está sujeto a restricciones de tiempo, ya que se ejecuta con periodicidades lo suficientemente grandes.

##### 4.1.2.1 Bloque de previsión de demanda

La demanda futura de energía por parte del consumidor es un dato que a priori escapa al control de la empresa suministradora. Una previsión de la demanda es de vital importancia para gestionar adecuadamente los sistemas de almacenamiento de la microrred. En el caso de estudio, se ha elegido hacer la previsión a cuatro periodos vista.

Teniendo en cuenta que un día tiene 24 horas, quedará dividido en periodos de 6 horas con una demanda diferente en cada uno de ellos.

Para el cálculo de la estimación se ha optado por un enfoque “naïf” frente a otros mecanismos como las medias móviles ponderadas. El enfoque naïf considera que el valor de la demanda durante un determinado periodo del día objeto de la previsión será la misma que la demanda que hay durante ese mismo periodo, en el día actual. Por ejemplo, si hoy durante la noche se ha medido que el consumo ha tenido lugar a un ratio de 4 kW de media, se espera que mañana por la noche el consumo sea el mismo.

Se podría haber considerado también la estacionalidad en la demanda que suponen los distintos días de la semana (el consumo de un domingo será distinto al de un día laborable), o incluso haber incluido otros factores correctores. Por citar alguno, si se sabe que va a haber una bajada de temperaturas al día siguiente se puede esperar un mayor consumo debido a los sistemas de calefacción. Por motivos de simplicidad y claridad, se ha decidido mantener el enfoque original.

### 4.1.3 Otros elementos de la microrred

La caracterización del modelo a estudiar se completa con la descripción del resto de componentes que la forman: generadores, sistemas de almacenamiento, cargas, líneas de transmisión y la propia red eléctrica a la que se conecta la microrred.

#### 4.1.3.1 Generador eólico (recurso renovable)

El aerogenerador tiene su rango de funcionamiento limitado a la generación y no se le permite que pase a trabajar como carga. Su potencia aparente nominal es de 4 kVA. Este valor se corresponde a la limitación física del convertidor VSC que tiene asociado.

Además de esta restricción, la potencia que es capaz de entregar en cada periodo se ve condicionada por la disponibilidad del recurso natural. Este valor de potencia disponible no es controlable en forma alguna y tampoco se realimenta al Optimizador.

#### 4.1.3.2 Grupo electrógeno diésel (recurso no renovable)

El generador basado en energía no renovable se modela y comporta de la misma forma que el aerogenerador, con la salvedad de que, en este caso, no existe la limitación de recurso.

#### 4.1.3.3 Sistemas de almacenamiento

La microrred cuenta con dos baterías, de idénticas características. A diferencia de los elementos generadores, que sólo pueden ceder energía al sistema, el flujo de potencia de las baterías puede ser bidireccional. Según las consignas que reciban, se cargarán o descargarán atendiendo a las necesidades energéticas de la microrred durante el periodo en curso.

En este escenario de simulación se ha considerado un rendimiento durante la carga y descarga del 90%, una capacidad de 1 kWh, que el convertidor asociado soporta una potencia aparente máxima de 500 VA, y que ambas baterías comienzan con un estado de carga del 50% (SOC = 500 Wh).

Todos estos parámetros se recogen en la Tabla 35.

Parámetro	Valor	Unidades
Capacidad	1	kWh
$S_{\max}$ del convertidor	500	VA
Eficiencia	90	%
$SOC_{\text{inicial}}$	50	%

Tabla 35. Parámetros de las baterías

#### 4.1.3.4 Cargas

El modelo cuenta con tres cargas consumidoras de potencia. En este escenario de simulación se ha preferido que todas sean de carácter no controlado. La conexión y desconexión de una carga no controlada es totalmente independiente de cualquier decisión del sistema.

- Carga óhmica no controlada de  $20 \Omega$ , permanentemente conectada a la red eléctrica.
- Carga óhmica no controlada que se conecta de forma directa y brusca al sistema. Para un mayor realismo y aleatoriedad, se hace variar su valor en cada periodo entre 15, 20, 25  $\Omega$ , o la desconexión.
- Carga inductiva no controlada de  $L = 84 \text{ mH}$ .

#### 4.1.3.5 Línea de transmisión

Los generadores y las baterías se conectan a las cargas y la red a través de un cable de 400 m, modelado por una impedancia óhmico-inductiva de valores  $R = 0.7 \Omega/\text{km}$  y  $L = 0.8 \text{ H/km}$ .

#### 4.1.3.6 Red

Las redes a las que se pueden conectar los Reguladores son redes de corriente alterna convencionales, con la topología típica de estos sistemas. Los VSC se conectan como una carga más de la red de baja tensión. Una red se puede caracterizar según el valor de su potencia de cortocircuito:

- Red fuerte: poseen una potencia de cortocircuito alta, y por consiguiente, no presentan problemas de estabilidad.
- Red débil: debido a su baja potencia de cortocircuito, son susceptibles de derivas en alguna de sus variables (amplitud, frecuencia o fase), y de problemas de estabilidad en transitorios bruscos, provocados por conexiones repentinas en las cargas. Todos los escenarios de simulación han sido realizados para una red débil, puesto que si el sistema funciona bajo este supuesto no tendrá problema alguno funcionando con una red fuerte.
- Red aislada, o en isla: son un caso extremo de red débil. El principal problema de este tipo de red es que siempre deben mantener, en tiempo real, el balance energético entre generación y demanda, puesto que no hay posibilidad de intercambio con otra red. Se verificará también el comportamiento de la microrred en esta situación.

## 4.2 Algoritmo de optimización para la gestión de la energía

El algoritmo propuesto minimiza la función del coste de la energía en la microrred. En dicha función de coste se suma el coste de los generadores y el coste que supone el sistema de almacenamiento.

Para los generadores se tiene en cuenta un precio de la energía (activa y reactiva) por periodo y generador, mientras que para las baterías se ha optado por un coste de funcionamiento, con lo que intentará minimizar los ciclos de carga y descarga. En otras implementaciones se podría considerar que las baterías fueran agentes independientes en el mercado, y vendieran o compraran energía buscando maximizar su beneficio (comprar energía barata para venderla más cara).

Respecto a las restricciones, se ha considerado el balance de potencias en la red en cada periodo, el balance energético en cada batería y en cada periodo, y las restricciones técnicas de generadores y baterías.

Los datos de entrada del algoritmo de optimización se muestran en la Tabla 36:

Variable	Descripción	Unidades
T	Periodos del problema: /per-1, per-2, per-3, per-4/	-
G	Generadores considerados: /gen-1, gen-2, red/	-
B	Baterías consideradas /bat-1, bat-2/	-
$dt_i$	Duración de cada periodo	h
$cvpg_{j,i}$	Coste de la energía activa de cada generador, por periodo	€/kWh
$cvqg_{j,i}$	Coste de la energía reactiva de cada generador, por periodo	€/kVArh
$Sgmax_j$	Estimación de potencia aparente máxima disponible de cada generador, por periodo	VA
$rend_k$	Rendimiento de las baterías	°/1
$Sbmax_k$	Potencia aparente máxima de cada batería	VA
$cvpb_{k,i}$	Coste variable de la energía activa de cada batería, por periodo	€/kWh
$cvqb_{k,i}$	Coste variable de la energía reactiva de cada batería, por periodo	€/kVArh
$demq_{m,i}$	Previsión del promedio de demanda de potencia reactiva, por periodo	VAr
$demp_{m,i}$	Previsión del promedio de demanda de potencia activa, por periodo	W

Tabla 36. Datos de entrada del algoritmo de optimización

Las variables a optimizar se recogen en la Tabla 37:

Variable	Descripción	Unidades
$pg_{j,i}$	Consigna de potencia activa para cada generador, por periodo	W
$qg_{j,i}$	Consigna de potencia reactiva para cada generador, por periodo	VAR
$pbc_{k,i}$	Potencia activa de carga para cada batería, por periodo	W
$pbd_{k,i}$	Potencia activa de descarga para cada batería, por periodo	W
$qbc_{k,i}$	Potencia reactiva de carga para el convertidor de cada batería, por periodo	VAR
$qbd_{k,i}$	Potencia reactiva de descarga para el convertidor de cada batería, por periodo	VAR
$SOC_{k,i}$	Carga de cada batería, al final de cada periodo	Wh

**Tabla 37. Variables a optimizar**

Para obtener las consignas de P y Q de las baterías, el Optimizador substraer la potencia de carga de la respectiva potencia de descarga. Se mantiene así el convenio de signos utilizado en los generadores, en las que un valor positivo indica ceder potencia a la microrred. El SOC final es utilizado como SOC inicial para la iteración posterior.

La ecuación a minimizar es la del coste total de la microrred (21).

- Minimizar:

$$\begin{aligned}
 Coste = & \sum_{j \in G} \sum_{i \in P} (pg_{j,i} \cdot cvpg_{j,i} + qg_{j,i} \cdot cvqg_{j,i}) \cdot dt_i \\
 & + \sum_{k \in B} \sum_{i \in P} (cvpb \cdot (pbc_{k,i} + pbd_{k,i}) + cvqb \cdot (qbc_{k,i} + qbd_{k,i})) \cdot dt_i
 \end{aligned} \quad (21)$$

El problema está sujeto a las restricciones que vienen a continuación:

- Balance de potencias en la red, en cada periodo i:

$$\sum_{j \in G} pg_{j,i} + \sum_{k \in B} (-pbc_{k,i} + pbd_{k,i} \cdot rend_k) \geq \sum_{m \in L} demp_{m,i} \quad (22)$$

$$\sum_{j \in G} qg_{j,i} + \sum_{k \in B} (-qbc_{k,i} + qbd_{k,i}) \geq \sum_{m \in L} demq_{m,i} \quad (23)$$

- Balance energético en cada batería k y en cada periodo i:

$$SOC_{k,i} = SOC_{k,i-1} + (pbc_{k,i} \cdot rend_k - pbd_{k,i}) \cdot dt_i \quad (24)$$

- Restricciones técnicas en los generadores, para cada generador  $j$  y periodo  $i$ :

$$\sqrt{(pg_{j,i})^2 + (qg_{j,i})^2} \leq sgmax_{j,i} \quad (25)$$

- Restricciones técnicas en las baterías, para cada batería  $k$  y periodo  $i$ :

$$\sqrt{(pbc_{k,i})^2 + (qbc_{k,i})^2} \leq sbmax_k \quad (26)$$

$$\sqrt{(pbd_{k,i})^2 + (qbd_{k,i})^2} \leq sbmax_k \quad (27)$$

$$SOC_{k,i} \leq SOCmax_k \quad (28)$$

$$pbd_{k,i} \leq \frac{SOC_{k,i-1}}{dt_i} \quad (29)$$

### 4.3 Implementación informática

En este apartado se pretenden explicar las cuestiones de carácter práctico acerca de la simulación de la microrred y del funcionamiento del algoritmo de optimización. Se divide en dos partes, en la primera se describirá el software utilizado, mientras que en la segunda abordará el esquema conceptual de la implementación informática.

#### 4.3.1 Software utilizado

La simulación de la microrred no se llevó a cabo utilizando un único software. Para implementar todas las funcionalidades requeridas han hecho falta cuatro programas distintos: JADE, Simulink, MACSimJX y AMPL. El *framework* JADE ya ha sido abordado en la apartado 2.4, por lo que sólo se explicarán los tres programas restantes.

##### 4.3.1.1 MATLAB-Simulink

Este programa permite modelar sistemas de cierta complejidad a un alto nivel de abstracción gracias a su entorno de programación gráfico. Pese a que existen alternativas de libre distribución, Simulink presenta varias ventajas:

- 1) Elevado grado de implantación, tanto en el ámbito académico como en el empresarial.
- 2) Disponibilidad de múltiples librerías de bloques y modelos listos para ser utilizados. En concreto, la librería *SimPowerSystem* contiene una amplia variedad de elementos eléctricos, que van desde sencillos componentes pasivos hasta elaborados generadores síncronos, pasando por toda suerte de dispositivos electrónicos de potencia.
- 3) La versatilidad que supone poder mezclar código con programación gráfica. En la simulación de la microrred, se ha explotado dicha ventaja para modelar los Reguladores, el bloque de previsión de demanda, la emulación de conexión

aleatoria de cargas y el elemento auxiliar que dicta la duración de los periodos de la simulación.

- 4) La existencia de una herramienta que permitiera el intercambio de datos entre Simulink y la plataforma multiagente en la que iba a actuar el Optimizador: el *middleware* MACSimJX.

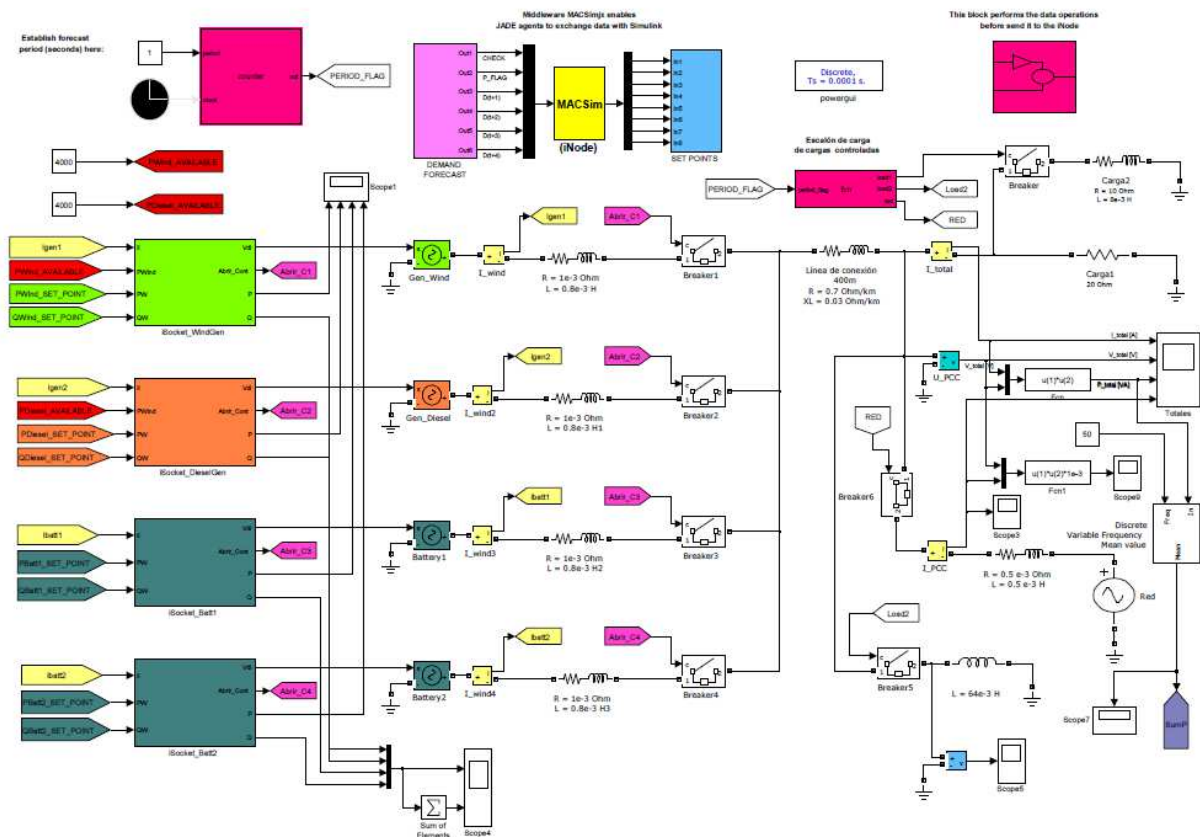


Fig. 57. Panorámica del modelo Simulink utilizado

#### 4.3.1.2 MACSimJX

Aunque Simulink es muy efectivo a la hora de llevar a cabo simulaciones, no ofrece las herramientas necesarias para implementar en él una plataforma multiagente. Sin embargo, las denominadas *S-functions* de las que dispone, permiten encapsular programas escritos en otros lenguajes de programación y hacer que funcionen dentro del propio entorno Simulink [26].

MACSimJX se vale de una *S-function* para crear una canal de comunicación de datos entre Simulink y un programa externo que posea la capacidad de procesamiento en paralelo que caracteriza a un sistema multiagente, en este caso JADE. MACSimJX presenta una arquitectura cliente-servidor.

Como puede observarse en la Fig. 58, la parte del cliente es embebida en Simulink a través de la *S-function* y la parte del código del servidor es incorporado en un programa separado. La comunicación entre cliente y servidor se lleva a cabo a través de las llamadas *named pipes* de Windows [27].

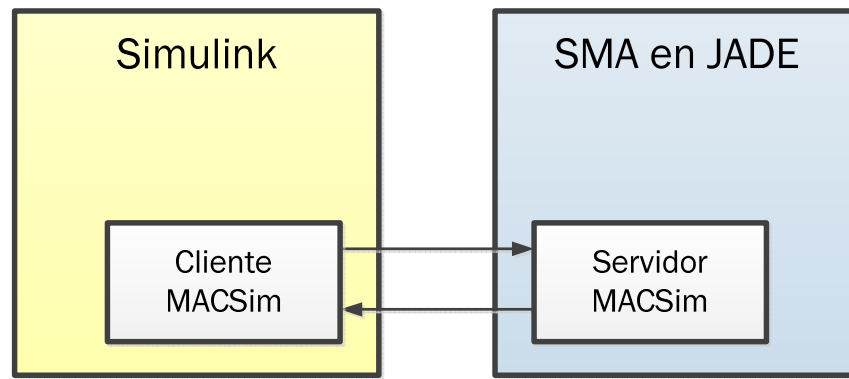


Fig. 58. Estructura de MACSim

MACSimJX permite recibir datos a través de la interfaz (*S-function*) “MACSim” y pasarlos a los agentes que se le especifique para que estos puedan procesarlos. Una vez que dichos agentes han finalizado todas las operaciones con los datos, los resultados son devueltos a Simulink a través de los mismos canales. Los agentes suelen ser diseñados para desempeñar operaciones concretas con los datos de entrada, como por ejemplo realizar alguna optimización.

Para llevar a cabo esta tarea, parece lógico dividir a los agentes en dos grupos: el *Agent Environment* (AE) y la *Agent Task Force* (ATF). El AE es responsable de la conexión entre Simulink y JADE y el usuario no ha de preocuparse de su funcionamiento interno. En la ATF se encuentran los agentes responsables de la manipulación de los datos, en el caso de estudio del presente trabajo la ATF está formada exclusivamente por el agente Optimizador.

Los desarrolladores de MACSimJX recomiendan usar una plantilla con funciones y protocolos ya escritos para programar los agentes que formarán parte de la ATF. Estos protocolos aseguran la sincronización durante el proceso de intercambio de datos. Aparte de esto, su implementación es idéntica al de un agente cualquiera en JADE, con los comportamientos y objetivos que normalmente tendrían.

La Fig. 59 muestra de forma intuitiva el funcionamiento de MACSimJX.

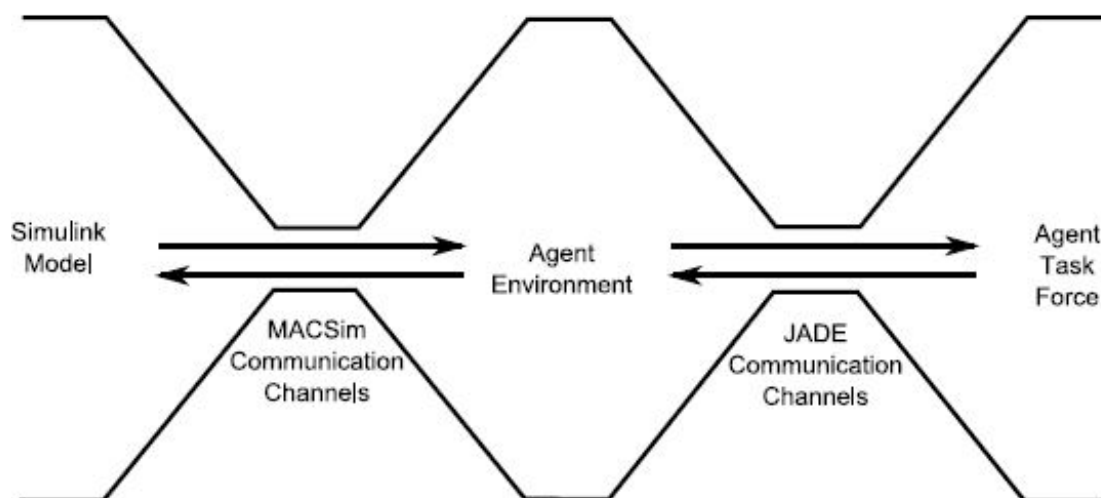


Fig. 59. Canales de comunicación establecidos por MACSimJX [27]



#### 4.3.1.3 A Modeling Language for Mathematical Programming (AMPL)

AMPL es un lenguaje de modelado algebraico para describir y resolver problemas de elevada complejidad matemática que cuenten con un gran número de variables y restricciones [28]. Fue desarrollado por Robert Fourer, David Gay y Brian Kernighan en los Laboratorios Bell.

AMPL funciona con docenas de *solvers*, tanto de código abierto como comerciales: CBC, CPLEX, FortMP, Gurobi, MINOS, IPOPT, SNOPT y KNITRO por citar algunos. Los problemas de optimización son pasados a los *solvers* como archivos *.nl*. El uso de AMPL está ampliamente extendido entre agencias gubernamentales, instituciones académicas y empresas.

La principal ventaja de AMPL radica en la similitud de su sintaxis con la de la notación matemática, lo que permite una definición concisa pero a la vez legible del problema.

#### 4.3.2 Esquema conceptual del modelo de simulación

Toda la microrred, a excepción del Optimizador, ha sido modelada en MATLAB-Simulink. Los Reguladores y el bloque de previsión de demanda han sido programados dentro de sus respectivas *embedded MATLAB function*. Todos los elementos circuitales (impedancias, fuentes de tensión, shunts de corriente, etc.) provienen de la librería *SimPowerSystem*.

El Optimizador se ejecuta en el entorno JADE, e intercambia información en tiempo real con Simulink a través del *middleware* MACSimJX. La Fig. 60 muestra una captura de pantalla de la GUI RMA del SMA implementado para la ocasión.

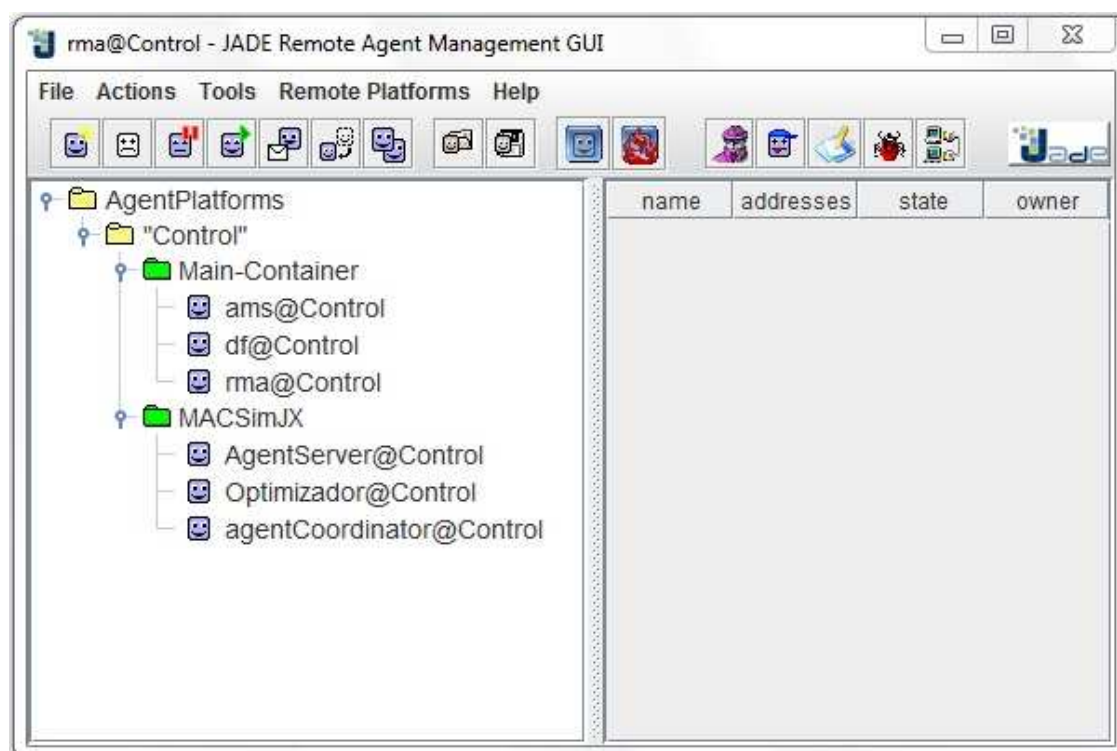


Fig. 60. SMA en el que se ejecuta el agente Optimizador

Para llevar a cabo la optimización del balance energético, es decir, para calcular las consignas de P y Q que enviará a los Reguladores, el Optimizador solicitará la asistencia del programa AMPL.

La Fig. 61 describe de forma gráfica e intuitiva la relación entre los cuatro programas usados para simular la microrred. Las flechas negras representan flujos de datos, de las que sólo son accesibles por el usuario las dibujadas horizontalmente.

Tanto la ecuación a optimizar como las restricciones que definen el problema (ver apdo. 4.2) están escritas siguiendo la notación que requiere AMPL, y se guardan en un archivo de formato *.mod*. Los datos de entrada al algoritmo de optimización se especifican en otro archivo distinto (*.dat*). Un tercer archivo (*.run*) es el encargado de indicar al *solver* el modelo a ejecutar, los datos a leer y en qué archivos escribir los resultados con las consignas PQ optimizadas y con el SOC final de las baterías, para que estos puedan ser accesibles por el Optimizador.

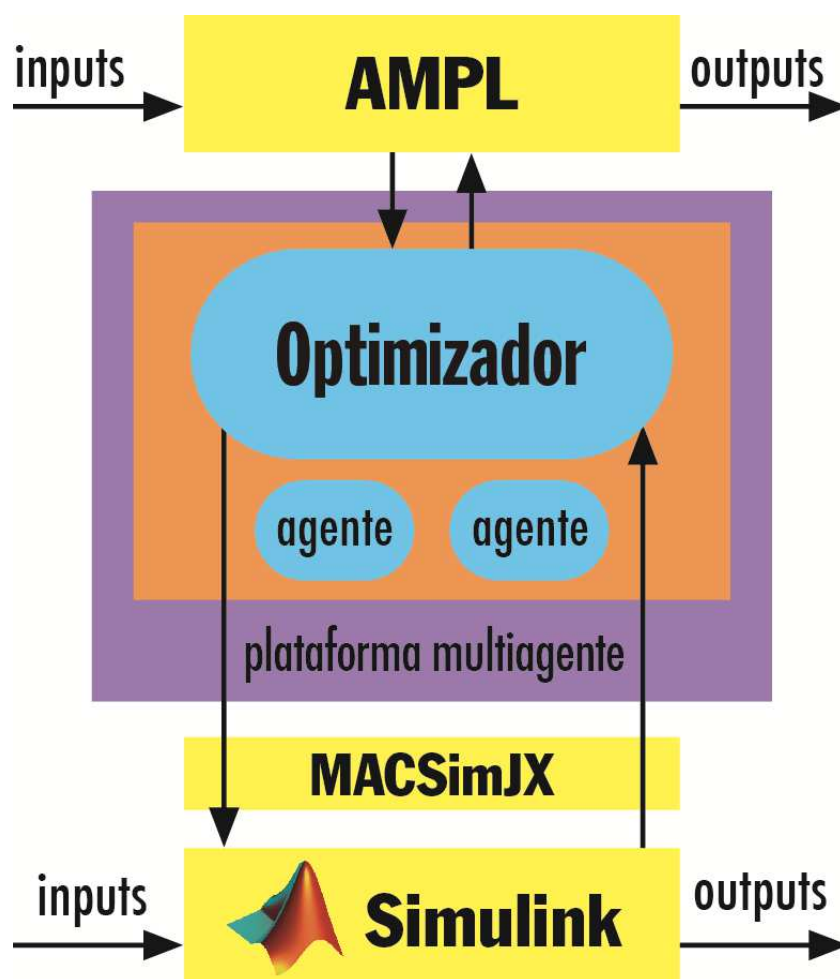


Fig. 61. Esquema conceptual del modelo de simulación

Las interacciones del Optimizador con el programa de optimización AMPL se producen momentos puntuales bien definidos, correspondientes a cambios de ciclo (cuatro periodos). Los *inputs* de Simulink de la Fig. 61 son los parámetros que caracterizan los elementos físicos que componen la microrred: los valores de las impedancias de conexión entre los distintos VSC, de la línea de distribución y de las cargas; valores de la tensión y frecuencia de la red, etc. Estos *inputs* son introducidos de forma manual en los menús de cada uno de los elementos gráficos.

Los *outputs* de Simulink son tensiones, corrientes, potencias y, en general, toda variable del circuito eléctrico susceptible de ser leída y estudiada con posterioridad.

Los *inputs* y *outputs* del algoritmo de optimización en AMPL han sido detallados en el apartado 4.2. Se les ha dado carácter de *inputs* y *outputs* por el hecho de ser accesibles al usuario mediante un editor de texto.

## 4.4 Casos de estudio y resultados

En este apartado se van a presentar los resultados obtenidos para los distintos escenarios de simulación considerados. El apartado 4.4.1 se va centrar en la optimización propiamente dicha, que establecerá las consignas de los Reguladores, mientras que los siguientes tres apartados centrarán su análisis en la efectividad de los Reguladores a la hora de seguirlas.

### 4.4.1 Resultados de una optimización para cuatro periodos mediante AMPL

En este apartado se comprobará el funcionamiento del algoritmo de optimización en la situación que resulta de introducir la serie de datos detallada en la Tabla 38. Los valores de consumo esperado demp y demq son proporcionados por el bloque de previsión de demanda.

Variable	per-1	per-2	per-3	per-4
dt [horas]	6	6	6	6
cvp(gen-1) [€/kWh]	0.07	0.07	0.07	0.07
cvp(gen-2) [€/kWh]	0.30	0.30	0.30	0.30
cvp(red) [€/kWh]	0.26	<b>0.36</b>	<b>0.36</b>	0.26
cvq(gen-1) [€/kVArh]	0.07	0.07	0.07	0.07
cvq(gen-1) [€/kVArh]	0.30	0.30	0.30	0.30
cvq(red) [€/kVArh]	0.26	<b>0.36</b>	<b>0.36</b>	0.26
Smax(gen-1) [VA]	4000	4000	<b>600</b>	<b>0</b>
Smax(gen-2) [VA]	4000	4000	4000	4000
Smax(red) [VA]	10000	10000	10000	10000
demq [VAr]	101.55	224.03	468.76	217.87
demp [W]	2606.38	5129.44	8129.44	5032.41

Tabla 38. Datos de entrada para el algoritmo de optimización

A la hora de implementar el algoritmo se consideró adecuado introducir de forma explícita el coste de la energía reactiva suministrada. Se le ha dado el mismo precio que a la activa y es de suponer que el consumidor tratará de evitar al máximo su demanda, mejorando la instalación para conseguir un factor de potencia próximo a la unidad.

Si se observa la tabla con atención, acontecen dos situaciones interesantes desde el punto de vista de la optimización. La primera, marcada en rojo, es la variación del recurso renovable. Se estima que para los periodos 3 y 4 el generador eólico sólo será capaz de extraer del viento 600 y 0 VA respectivamente. La segunda es un aumento de la tarifa de

la energía de red. De esta manera, el precio de la misma se sitúa por encima del de coste del generador 2.

Se ejecuta la optimización en AMPL con el siguiente resultado:

Consignas	per-1	per-2	per-3	per-4
<b>Potencia activa [W]</b>				
P(gen-1)	2976.75	4000	600	0
P(gen-2)	0	1129.44	4000	0
P(red)	0	0	3229.44	5032.41
P(bat-1)	-185.185	0	166.667	0
P(bat-2)	-185.185	0	166.667	0
P(Total)	2606.38	5129.44	8129.44	5032.41
<b>Potencia reactiva [VAr]</b>				
Q(gen-1)	0	0	0	0
Q(gen-2)	0	0	0	0
Q(red)	0	0	0	0
Q(bat-1)	108.31	219.485	153.918	217.87
Q(bat-2)	-6.76	4.545	312.842	0
Q(Total)	101.55	224.03	468.76	217.87

**Tabla 39. Consignas optimizadas de P y Q**

Durante el primer periodo, la demanda de potencia activa será suministrada en su totalidad por el aerogenerador. Resulta obvio que esto sea así debido a que su coste de generación es notablemente inferior al del resto, 0,07 €/kWh.

Se ha aprovechado la disponibilidad del recurso renovable para recargar las baterías a un ratio de 185 W cada una. Esta energía almacenada será usada durante el pico de demanda del periodo 3. Las consignas de las baterías coinciden porque ambas han sido modeladas de igual manera y parten de niveles de SOC idénticos.

Puede resultar extraño el hecho de que las baterías no se hayan cargado a más potencia, puesto que el aerogenerador dispone durante este periodo de potencia de sobra; pero cabe recordar aquí que hay unas pérdidas asociadas a variaciones netas de SOC del 10% que limitan su rentabilidad. Además, el algoritmo realiza la optimización para cuatro periodos y ha considerado que no es más ventajoso en términos económicos recargarlas más.

La potencia reactiva es proporcionada de forma prioritaria por ellas, puesto que el SOC en este caso no varía de un periodo a otro. La función de las baterías es, por lo tanto, equivalente a la de un banco de condensadores.

En el segundo periodo se produce un incremento en el precio de la energía de red, siendo ahora el más caro de los tres. Este hecho, junto al incremento que se prevé en la demanda, obliga al aerogenerador a entregar la máxima potencia disponible y al grupo electrógeno a arrancar. Las baterías no ceden ni absorben energía neta al sistema, pero continúan proporcionando la potencia reactiva.

El tercer periodo es crítico debido a que se espera en él el pico de demanda del día. Coincide también con un notable descenso del recurso eólico, por lo que se le va a requerir al generador diésel que entregue la máxima potencia que es capaz de suministrar. Para minimizar la energía comprada a la red (que mantiene el precio del periodo 2), las baterías se descargarán a un ratio de 166,67 W.

En el último periodo, el recurso renovable previsto cae a cero y el precio de la energía de red se vuelve a colocar por debajo del precio del recurso no renovable. La opción óptima será, obviamente, desconectar el grupo electrógeno y que la microrred compre toda la energía al exterior.

Hay que tener en cuenta el hecho de que no hay ningún Regulador controlando el flujo de potencia que viene de la red, modelada como un generador ideal de tensión en Simulink. Los valores de P y Q presentes en la tabla de resultados que corresponden a ésta no son consignas. Su función se reduce a la de calcular el coste total del suministro. El input Smax(red) es el valor máximo de S que se quiere solicitar a la red. El modelo Simulink lo limita por sí mismo este valor ya que la red está modelada por un generador ideal de tensión, capaz de entregar toda la corriente que se le requiera (y por lo tanto toda la potencia, al mantenerse V constante). Si se prefiere no acotar la potencia comprada a la red, bastará con asignar un valor suficientemente grande al input Smax(red).

Por último, conviene hacer hincapié en que, debido al carácter aleatorio de las cargas y del recurso natural, surgirán diferencias entre las estimaciones y los valores reales de demanda y generación eólica, que deberán ser compensadas por la red.

#### 4.4.2 Verificación del seguimiento de las consignas

En este apartado se verificará el seguimiento de las consignas mostradas en la Tabla 39. Para ello se va a representar la evolución temporal de la potencia entregada por cada elemento durante el cambio que se produce al pasar del periodo 1 al periodo 2. Se ha elegido esta situación debido a que tanto los dos generadores como las baterías registran actividad en ella.

La Tabla 40 recoge las consignas de potencia activa para los periodos 1 y 2:

Consigna	per-1	per-2
<b>Potencia activa [W]</b>		
P(gen-1)	2976.75	4000
P(gen-2)	0	1129.44
P(red)	0	0
P(bat-1)	-185.185	0
P(bat-2)	-185.185	0

Tabla 40. Consignas de P para los periodos 1 y 2

La Tabla 41 recoge las consignas de potencia activa para los periodos 1 y 2:

Consigna	per-1	per-2
<b>Potencia reactiva [VAr]</b>		
Q(gen-1)	0	0
Q(gen-2)	0	0
Q(red)	0	0
Q(bat-1)	108.31	219.485
Q(bat-2)	-6.76	4.545

Tabla 41. Consignas de Q para los periodos 1 y 2

En la Fig. 62 se comprueba que los Reguladores siguen sin problemas las consignas que les dicta el Optimizador. Ante el aumento de la demanda, el aerogenerador pasa a dar su máxima potencia, y el grupo electrógeno arranca hasta entregar los 1130W que se le exigen. Por su parte las baterías van dejando de cargarse y, tras un tiempo de transición razonable, logran un intercambio energético neto con la microrred igual a cero.

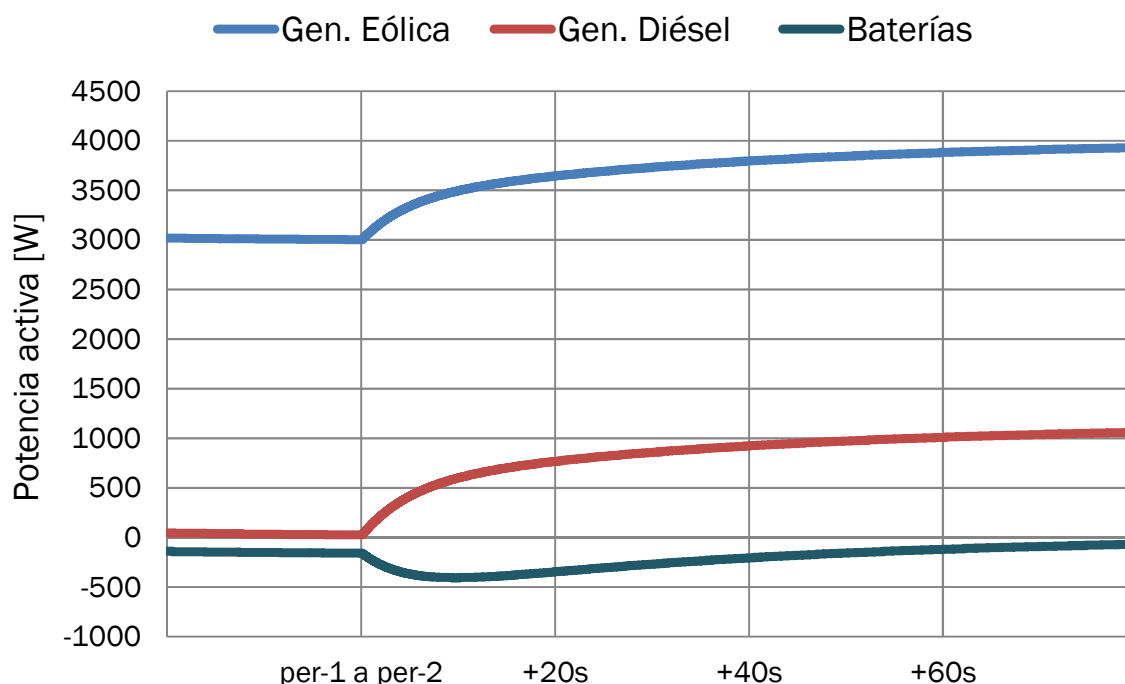


Fig. 62. Seguimiento de consignas de P: transición del periodo 1 al periodo 2

El seguimiento de consignas para la potencia reactiva es similar al de activa. Tras el transitorio producido al cambiar las referencias, los cuatro elementos se ajustan en poco más de 30 segundos a su nuevo estado estable.

El aerogenerador y el grupo diésel vuelven a no suministrar potencia reactiva, mientras que las baterías entregan las nuevas consignas con un grado de exactitud más que aceptable.

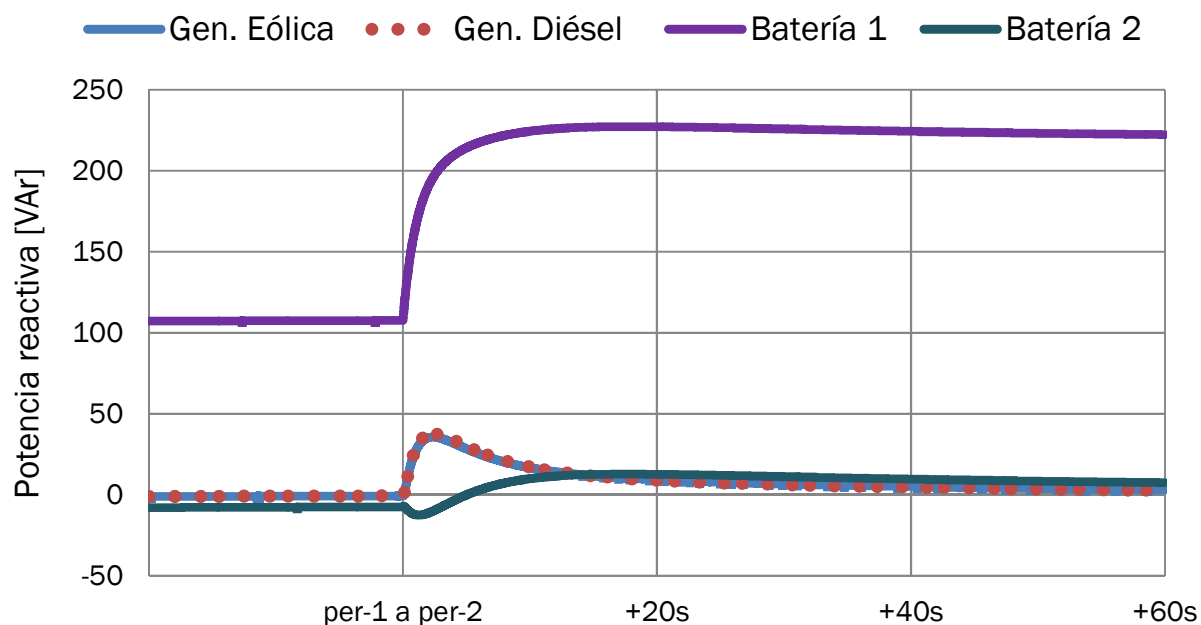


Fig. 63. Seguimiento de consignas de Q: transición del periodo 1 al periodo 2

La pequeña diferencia que se aprecia es la normal en controles *droop*, puesto que los Reguladores se asegurarán siempre de no llevar a cabo acciones desmesuradas, que pudieran comprometer los valores de tensión fijados en nodos de la microrred.

#### 4.4.3 Respuesta a un cambio brusco de carga

En este caso se observará la reacción de los Reguladores ante una perturbación producida por una conexión repentina de una carga resistiva no controlada, durante el periodo 3. Las consignas de potencia activa permanecen invariables, independientemente del cambio en el valor de la impedancia que modela el consumo.

La Tabla 42 contiene las consignas de P cada elemento para el periodo 3. Los Reguladores tratarán de seguirlas en el nuevo régimen de carga.

Consigna	per-3
Potencia activa [W]	
P(gen-1)	600
P(gen-2)	4000
P(red)	3229.44
P(bat-1)	166.667
P(bat-2)	166.667

Tabla 42. Consignas de P para el periodo 3

La Tabla 42 contiene las consignas de P cada elemento para el periodo 3. Los Reguladores tratarán de seguirlas en el nuevo régimen de carga.

Consigna	per-3
<b>Potencia reactiva [VAr]</b>	
Q(gen-1)	0
Q(gen-2)	0
Q(red)	0
Q(bat-1)	153.918
Q(bat-2)	312.842

Tabla 43. Consignas de Q para el periodo 3

El consumo inicial del periodo 3 lo realiza una carga óhmica pura de  $20\ \Omega$ . De forma repentina, se conecta una segunda carga, de idéntico valor en paralelo a la primera, dando lugar a la carga de  $10\ \Omega$ :

- Carga no controlada inicial:  $20\ \Omega$  (2.6 kW)
- Carga no controlada inicial:  $10\ \Omega$  (5.2 kW)

En la Fig. 64 se observa que el sistema se encuentra en el régimen permanente, con los cuatro elementos entregando la potencia que les solicita el Optimizador. En el punto temporal denominado como “conexión repentina” se conecta la segunda carga.

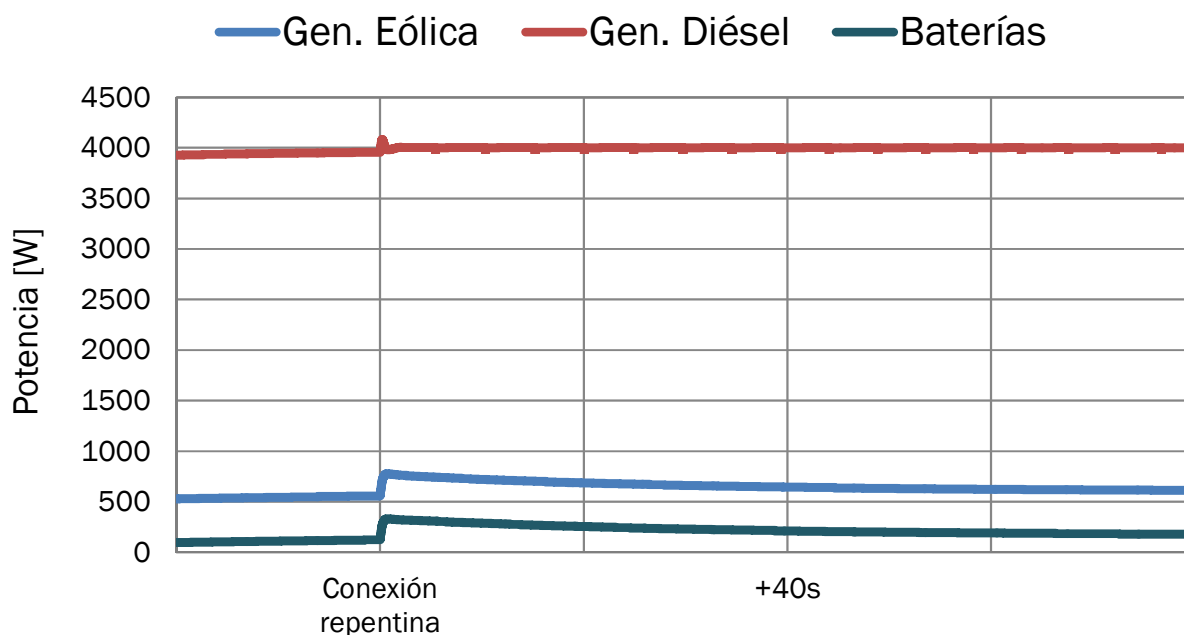


Fig. 64. Respuesta a cambio brusco en la carga durante el periodo 3: P



Durante este periodo, el grupo electrógeno diésel entrega 4kW, su máxima potencia. Es por ello que no puede superar este valor pese al cambio producido. Las dos baterías y el aerogenerador sí que verán variar su potencia entregada. Al producirse este escalón, los Reguladores actúan variando la frecuencia de los VSC hasta dejar el desfase de tensiones en el punto deseado. Al cabo de un tiempo, las potencias vuelven a ser las requeridas.

Es importante prestar atención a dos hechos. El primero es que esta reacción sólo es posible si la microrred se encuentra conectada a red, ya que será la red la que asuma el incremento de consumo de las cargas. Los cuatro generadores no ven modificada, tras el periodo de transición que se muestra en la gráfica, la potencia que entregan.

Tal y como está implementado el bloque de previsión, el Optimizador no modifica las consignas de forma instantánea, sino que considera razonable atender a la previsión y esperar que sea un cambio de carácter temporal.

El segundo hecho a destacar es que la estimación del recurso renovable ha sido algo conservadora. Se observa que el aerogenerador dispone de algo más de potencia que la prevista y puede entregar el pico producido por la conexión repentina de la carga. En el caso que viene a continuación se estudiará la situación contraria, menor disponibilidad del recurso natural de la que se prevé.

En lo que respecta a la potencia reactiva, los Reguladores reaccionan rápidamente ante el aumento de corriente que se ha producido y consiguen mantener las consignas del Optimizador sin dificultad, tal y como se muestra en la Fig. 65.

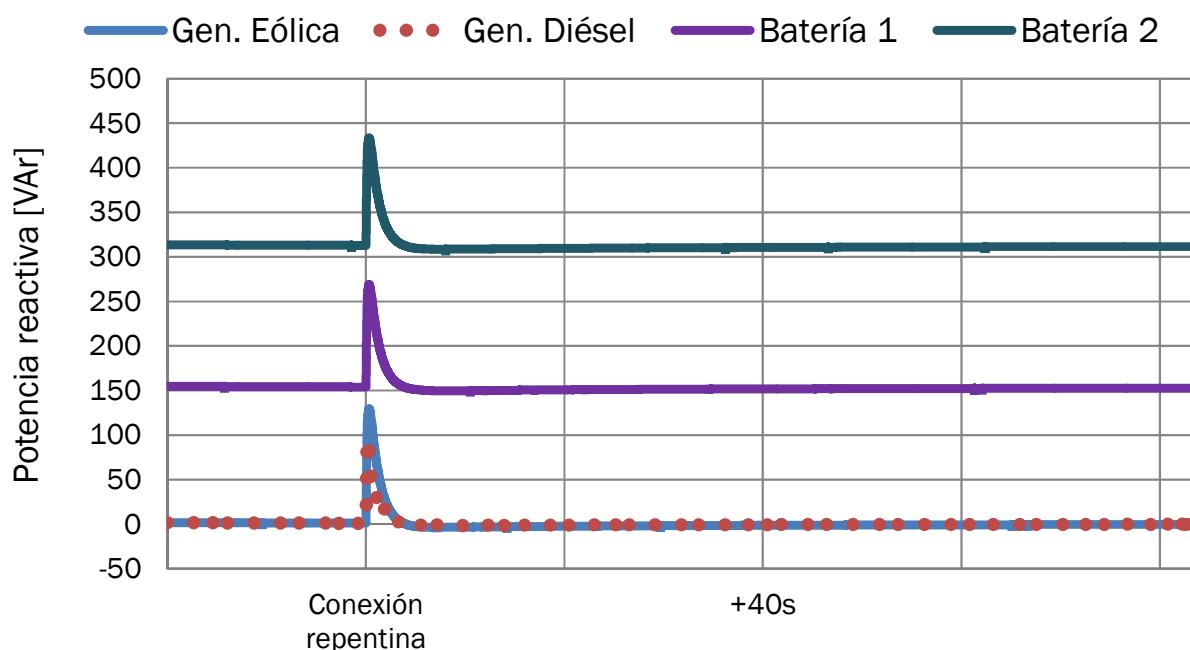


Fig. 65. Respuesta a cambio brusco de la carga óhmica durante el periodo 3: Q

Un caso más interesante para analizar la reacción del control de Q sería conectar una carga inductiva, en lugar de la anterior puramente óhmica. Para ello, se sustituye la resistencia de  $20\Omega$  por una bobina de  $84\text{mH}$  y se vuelve a simular el modelo. La respuesta es la siguiente:

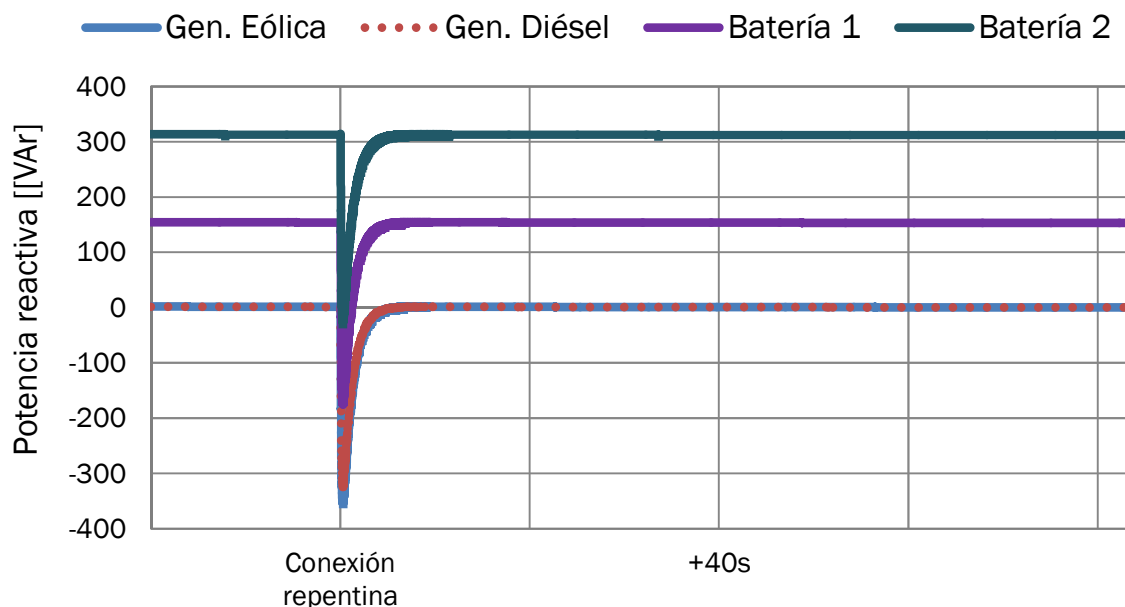


Fig. 66. Respuesta a cambio brusco de la carga inductiva durante el periodo 3: Q

Los Reguladores tampoco tienen ningún problema para volver al estado de consignas iniciales. Al igual que en el caso anterior, va a ser la red la que supla la diferencia entre demanda de potencia reactiva y el sumatorio de potencias entregadas por los generadores.

#### 4.4.4 Funcionamiento en isla

Interesa conocer también el comportamiento de la microrred trabajando de manera aislada, o “en isla”. Para ello, basta con desconectar el generador que modela la red en Simulink. Hasta el momento, la red había sido la responsable de compensar la diferencia entre la demanda y el sumatorio de consignas, pero ahora ya no cabrá esta posibilidad.

El escenario simulado no forma parte de ningún periodo de la optimización del apartado A, puesto que ésta no se realizó bajo el supuesto de desconexión de red. No obstante, si se desea utilizar el algoritmo para obtener el balance óptimo de la microrred funcionando en isla, bastará con indicarle que la potencia disponible  $S_{\text{max}}(\text{red})$  es 0 VA.

Las consignas escogidas para la ocasión se presentan en la Tabla 44 y se han sido escogido arbitrariamente para poder visualizar lo mejor posible la respuesta del sistema.

El Optimizador va a requerir al Regulador del aerogenerador su potencia nominal (4 kW), mientras que al grupo electrógeno no le va a encomendar generación alguna. A las baterías les va a dictar unas consignas de carga de 500 y 200 W respectivamente.

Consigna	[W]
P(gen-1)	4000
P(gen-2)	0
P(bat-1)	-500
P(bat-2)	-200

Tabla 44. Consignas de potencia activa

Para analizar la respuesta de la microrred ante un decremento de disponibilidad del recurso renovable, la potencia extraíble del viento caerá de forma temporal hasta los 2000 W, para volver después a su valor de partida (la consigna no varía).

A los 100 segundos de simulación se conectará, en paralelo, una segunda carga que sumará su consumo al de la que había inicialmente. Los valores inicial y final de dicha carga coinciden con los del caso anterior (20 y 10  $\Omega$ ).

La Fig. 67 muestra la forma de reaccionar de la microrred ante los eventos anteriormente mencionados cuando se encuentra funcionando aislada de la red.

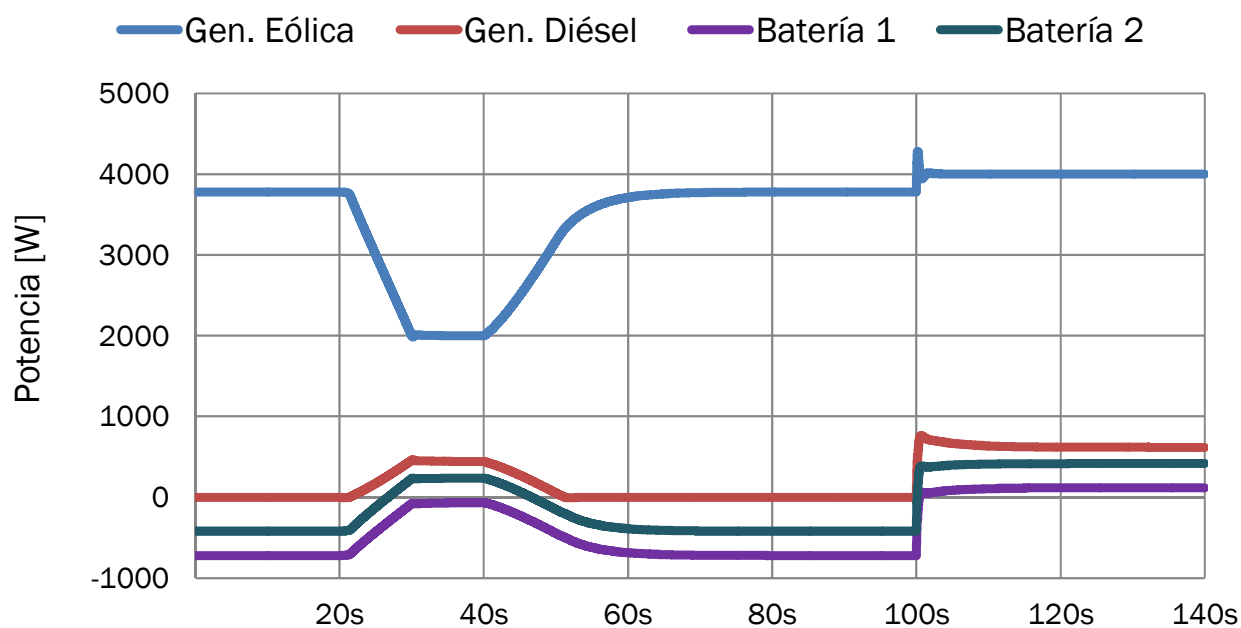


Fig. 67. Funcionamiento en isla: P

Se observa que el reparto de potencias difiere sustancialmente de los casos de simulación anteriores, en los que sí existía conexión a red. En efecto, el aerogenerador (línea azul) comienza entregando 3500 W, pese a que la consigna del Optimizador son 4000 W y dispone de viento suficiente. La razón de esta discrepancia radica en que en este tipo de redes, prima mantener el equilibrio energético entre generación y demanda por encima de cualquier otro requerimiento. Se verifica que esto es así fácilmente: de estos 3500 W, 700 W los absorben las baterías y 2600 W la carga. Los 200 W sobrantes se disipan en las líneas que unen los generadores con las cargas, modeladas como impedancias RL.

Transcurridos 20 segundos el viento baja de intensidad, obligando al grupo electrógeno a arrancar y a las baterías a devolver energía a la red para mantener el equilibrio. Cuando se vuelve a disponer de potencia proveniente del recurso natural, el generador diésel y las baterías recuperan sus puntos de funcionamiento anteriores a la perturbación.

Finalmente, se analiza la respuesta de los cuatro elementos ante el brusco aumento de demanda, que tiene lugar a los 100 segundos como consecuencia de conectar una segunda carga en paralelo. El aerogenerador comienza a entregar su potencia nominal y el resto de elementos incrementan de forma análoga su potencia para mantener el balance en la microrred, manteniendo así el equilibrio entre generación y demanda.

## 4.5 Conclusiones y líneas futuras

En este capítulo se ha propuesto un sistema de control y optimización para una microrred de corriente alterna. Se ha optado por un sistema de control semidistribuido, con dos elementos reguladores, de funciones y responsabilidades diferenciadas.

La potencia activa y reactiva es controlada en tiempo real por los Reguladores mediante la técnica P-Q *droop*. Un nivel jerárquico por encima de ellos se sitúa el Optimizador, cuya función es optimizar el coste del balance energético de la microrred. Se ha presentado el algoritmo de control, la idiosincrasia del software utilizado para la simulación, y se ha verificado el buen funcionamiento del sistema en diversos escenarios y situaciones.

### 4.5.1 Conclusiones

Las conclusiones extraídas son:

- El control implementado funciona correctamente, tanto al nivel de la optimización como al del control PQ. La arquitectura semidistribuida permite dotar al sistema de flexibilidad manteniendo la posibilidad de tomar decisiones a un nivel global.
- La comunicación en tiempo real entre un SMA en JADE y un modelo Simulink es posible mediante el *middleware* MACSimJX, aunque su utilización presenta un inconveniente: ralentiza la simulación. Estas comunicaciones se llevan a cabo con una periodicidad definida e invariable, que debe coincidir con el paso de simulación del modelo Simulink. Este hecho impide a Simulink aumentar el paso de tiempo en los momentos de menor carga de cálculo, lo que implica que toda la simulación se lleve a cabo con el mínimo paso de simulación. La diferencia llega a ser de varios órdenes de magnitud.
- El acierto del agente supervisor a la hora de optimizar depende de lo acertadas que sean sus previsiones de demanda y de generación.
- AMPL resulta ideal para el modelado de problemas de esta naturaleza. La curva de aprendizaje es muy favorable debido a una sintaxis sencilla e intuitiva.
- Para aprovechar al máximo las posibilidades de integrar la tecnología de agentes y el entorno Simulink, los modelos que implementen cada uno de ellos deberán coincidir en la escala de tiempo. Por ejemplo, si Simulink modela con un paso de tiempo adecuado para el régimen transitorio (del orden de  $10^{-4}$  segundos), el SMA deberá trabajar a una escala de tiempo similar. Esto no se ha respetado, al haber mezclado optimización a un orden de magnitud de horas con un control P/F en el entorno del milisegundo.

### 4.5.2 Líneas futuras

Las posibles líneas de continuidad son:

- Previo a un cambio radical en el modelo, sería conveniente tratar de lograr que los canales de comunicación abiertos entre el agente Optimizador y Simulink funcionaran únicamente en los momentos puntuales requeridos. En la implementación actual este flujo de datos nunca cesa. Por un lado los inputs que recibe la función MACSim (estimaciones de demanda), son enviados en cada paso de simulación; aunque tan sólo son leídas y guardadas de forma puntual una vez que han pasado los  $n$  periodos. Asimismo, los outputs de la función MACSim (consignas para los Reguladores) permanecen inalterados durante un periodo completo, mientras que son enviados a través de MACSim de forma continua. Debería ser posible acceder al código y modificarlo para “cerrar” dichos canales cuando no se necesiten.
- El modelo de previsión de la demanda y de la generación no controlable puede mejorarse con el desarrollo de algún modelo basado en redes neuronales o similar, que permitiría mejorar el control y obtener mejores precios en el mercado eléctrico.
- El agente Optimizador podría comunicarse con otros agentes Optimizadores mediante los canales habituales de JADE para, mediante un proceso de subasta, intercambiar energía con otras microrredes para minimizar su coste.
- Para acelerar la simulación se plantea la utilización de un clúster de ordenadores para dividir la carga de cálculo necesaria. Si esto no fuera posible se podría reducir el nivel de detalle de los objetos modelados (utilizar modelos propios en lugar de los de la librería *SimPowerSystem*) o reducir al mínimo los elementos de la microrred que Simulink tiene que simular.



## 5 Conclusiones personales

El desarrollo de este PFC me ha dado la oportunidad de trabajar en el campo de las energías renovables, por el que siempre he tenido un gran interés. He podido colaborar en investigaciones llevadas a cabo por la fundación CIRCE, de aplicación real en la industria eléctrica, lo que me ha hecho ser consciente de la gran dificultad que los proyectos de esta índole entrañan.

Gracias a este PFC he aprendido y manejado conceptos y herramientas nuevos, tanto en el ámbito del software como en el de los sistemas de potencia; y he tenido la ocasión de trabajar codo a codo con ingenieros e investigadores de gran experiencia.

Las tareas de programación han supuesto una excelente forma de entrenar mi paciencia, especialmente durante las fases de depuración de fallos. Asimismo, la redacción de la memoria me ha permitido mejorar los conocimientos de Microsoft Excel y Word.

Este PFC no es un punto y final, sino más bien un punto y aparte, puesto que se continuará trabajando en él a lo largo del año académico 2013/2014; siguiendo las líneas de continuidad que aquí se han detallado.





## 6 Anexos

### 6.1 Feedback modo *Debug* (Iteration 1 of 17520, *Peak Shaving*)

A continuación se presenta el feedback que proporciona el emulador cuando se activan los tres modos de depuración de fallos: *BatteryDebug*, *ReactiveDebug* y *ACLdebug*.

El carácter asíncrono inherente a un SMA se pone de manifiesto en el orden en el que este *feedback* aparece en pantalla. El texto en **negrita** se sigue mostrando estén o no estén activados los modos *Debug*.

```
ago 30, 2013 7:36:30 PM jade.core.Runtime beginContainer
INFO: -----
      This is JADE 4.2.0 - revision 6574 of 2012/06/20 15:38:00
      downloaded in Open Source, under LGPL restrictions,
      at http://jade.tilab.com/
-----
Retrieving CommandDispatcher for platform MicroGrid
ago 30, 2013 7:36:31 PM jade.imtp.leap.LEAPIMTPManager initialize
INFO: Listening for intra-platform commands on address:
- jicp://155.210.29.118:1099

ago 30, 2013 7:36:33 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
ago 30, 2013 7:36:33 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
ago 30, 2013 7:36:33 PM jade.core.BaseService init
INFO: Service jade.core.resource.ResourceManagement initialized
ago 30, 2013 7:36:34 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
ago 30, 2013 7:36:34 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
ago 30, 2013 7:36:34 PM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
ago 30, 2013 7:36:34 PM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://155.210.29.118:7778/acc
ago 30, 2013 7:36:35 PM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@155.210.29.118 is ready.
-----
<load2> agent reading data from: [load2.xlsx]
<wind> agent reading data from: [wind1.xlsx]
<load1> agent reading data from: [load1.xlsx]
<pv1> agent reading data from: [pv1.xlsx]
<pv2> agent reading data from: [pv2.xlsx]
<load3> agent reading data from: [load3.xlsx]
<pv4> agent reading data from: [pv4.xlsx]
<load4> agent reading data from: [load4.xlsx]
<pv3> agent reading data from: [pv3.xlsx]
<battery2@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@ab1bf>: READY
<battery1@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@138fec2>: READY
<battery4@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@dd9715>: READY
<battery3@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@165532d>: READY
<power_manager> agent registered successfully its service CONTROL, type "POWER_CONTROL".

<power_manager> waiting in "blockingReceive" mode.
1 messages received.

<power_manager> waiting in "blockingReceive" mode.
2 messages received.

<power_manager> waiting in "blockingReceive" mode.
3 messages received.

<power_manager> waiting in "blockingReceive" mode.
4 messages received.

<power_manager> waiting in "blockingReceive" mode.
<battery_manager@MicroGrid> agent is ready!
```

```

<battery2> agent registered successfully its service STORAGE, type "BATTERY".

<battery2> agent added to its active behaviour queue "MailBoxStorage".
<battery4> agent registered successfully its service STORAGE, type "BATTERY".

<battery4> agent added to its active behaviour queue "MailBoxStorage".
<battery2@MicroGrid> agent is ready!

<battery4@MicroGrid> agent is ready!

<battery1> agent registered successfully its service STORAGE, type "BATTERY".

<battery1> agent added to its active behaviour queue "MailBoxStorage".
<battery1@MicroGrid> agent is ready!

<battery3> agent registered successfully its service STORAGE, type "BATTERY".

<battery3> agent added to its active behaviour queue "MailBoxStorage".
<battery3@MicroGrid> agent is ready!

<pv2> agent # of samples read from pv2.xlsx: 17520
<pv4> agent # of samples read from pv4.xlsx: 17520
<pv1> agent # of samples read from pv1.xlsx: 17520
<wind> agent # of samples read from wind1.xlsx: 17520
<pv3> agent # of samples read from pv3.xlsx: 17520
<load1> agent # of samples read from load1.xlsx: 17520
<load4> agent # of samples read from load4.xlsx: 17520
<load3> agent # of samples read from load3.xlsx: 17520
<load2> agent # of samples read from load2.xlsx: 17520
<wind@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@1494d31>: READY
5 messages received.

<power_manager> waiting in "blockingReceive" mode.
<load3@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@17b66b1>: READY
<pv1@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@3cae4a>: READY
<load1@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@9a7be7>: READY
<pv4@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@577474>: READY
<load2@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@102cbee>: READY
<pv3@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@93100d>: READY
<pv2@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@1d6aa8e>: READY
<load4@MicroGrid> sending {INFORM} to <jade.util.leap.ArrayList$1@1a4b0c4>: READY
6 messages received.

<power_manager> waiting in "blockingReceive" mode.
7 messages received.

<power_manager> waiting in "blockingReceive" mode.
8 messages received.

<power_manager> waiting in "blockingReceive" mode.
9 messages received.

```

---

```

<wind> agent registered successfully its service GENERATION, type "RENEWABLE".

<power_manager> agent added to its active behaviour queue "AMSSubscriberRPM".
<load3> agent registered successfully its service CONSUMPTION, type "PRIORITY".
<pv1> agent registered successfully its service GENERATION, type "RENEWABLE".

<wind> agent added to its active behaviour queue "MailBoxGen".

<pv1> agent added to its active behaviour queue "MailBoxGen".
<load1> agent registered successfully its service CONSUMPTION, type "PRIORITY".
<power_manager> agent added to its active behaviour TickerBehaviour with 250 milliseconds iteration.

<pv4> agent registered successfully its service GENERATION, type "RENEWABLE".
<wind@MicroGrid> agent is ready!

<pv4@MicroGrid> agent is ready!

<pv4> agent added to its active behaviour queue "MailBoxGen".
<load3@MicroGrid> agent is ready!

```

```

<load1@MicroGrid> agent is ready!

<pv1@MicroGrid> agent is ready!

<pv2> agent registered successfully its service GENERATION, type "RENEWABLE".
<pv2@MicroGrid> agent is ready!

<pv2> agent added to its active behaviour queue "MailBoxGen".
<pv3> agent registered successfully its service GENERATION, type "RENEWABLE".
<pv3@MicroGrid> agent is ready!

<pv3> agent added to its active behaviour queue "MailBoxGen".
<load2> agent registered successfully its service CONSUMPTION, type "PRIORITY".
<load2@MicroGrid> agent is ready!

<load4@MicroGrid> agent is ready!

<load4> agent registered successfully its service CONSUMPTION, type "PRIORITY".
<power_manager> agent subscribed to GENERATION agent <wind>.
<power_manager> agent subscribed to GENERATION agent <pv1>.
<power_manager> agent subscribed to GENERATION agent <pv2>.
<power_manager> agent subscribed to GENERATION agent <pv4>.
<power_manager> agent subscribed to GENERATION agent <pv3>.
<power_manager> agent subscribed to STORAGE agent <battery2>.
<power_manager> agent subscribed to STORAGE agent <battery4>.
<power_manager> agent subscribed to STORAGE agent <battery3>.
<power_manager> agent subscribed to STORAGE agent <battery1>.
<power_manager> agent subscribed to CONSUMPTION agent <load2>.
<power_manager> agent subscribed to CONSUMPTION agent <load3>.
<power_manager> agent subscribed to CONSUMPTION agent <load4>.
<power_manager> agent subscribed to CONSUMPTION agent <load1>.

<power_manager> agent added to its active behaviour queue "MailBoxPQControl".

Initiating FIPA_CONTRACT_NET protocol...

<power_manager@MicroGrid> sending {CFP} to <wind@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <pv1@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <pv2@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <pv4@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <pv3@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <battery2@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <battery4@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <battery3@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <battery1@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <load2@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <load3@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <load4@MicroGrid>: Calling for proposals...
<power_manager@MicroGrid> sending {CFP} to <load1@MicroGrid>: Calling for proposals...

{CFP} sent by <power_manager@MicroGrid> successfully received by <wind@MicroGrid>: Calling for
proposals...
<wind@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: GENERATION | Power resource [kW]:
48.247 | Price [€/kWh]: 0.003 | VSC max S: [kVA]: 200.0 | VSC efficiency: 0.95

{CFP} sent by <power_manager@MicroGrid> successfully received by <pv4@MicroGrid>: Calling for
proposals...
<pv4@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: GENERATION | Power resource [kW]:
0.0 | Price [€/kWh]: 0.12 | VSC max S: [kVA]: 100.0 | VSC efficiency: 0.95

{CFP} sent by <power_manager@MicroGrid> successfully received by <battery2@MicroGrid>: Calling for
proposals...
CHARGE:
energy_available = [(max_soc - current_soc)/100] * energy_capacity
450000.0 = [(100.0 - 50.0)/100] * 900000.0
DISCHARGE:
energy_available = [(current_soc - min_soc)/100] * energy_capacity
405000.0 = [(50.0 - 5.0)/100] * 900000.0
MAX_CHARGE_POWER_OFFER, agent <battery2> is 23.75 [kW]
MAX_DISCHARGE_POWER_OFFER, agent <battery2> is 23.75 [kW]
MAX_CHARGE_REACTIVE_OFFER, agent <battery2> is 45.13244398434457 [kVar]

```

```

MAX_DISCHARGE_REACTIVE_OFFER, agent <battery2> is 45.13244398434457 [kVar]
<battery2@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: STORAGE | Charge power offer
[kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVar]: 45.13244398434457 |
Discharging reactive offer [kVar]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]: 50.0

{CFP} sent by <power_manager@MicroGrid> successfully received by <pv2@MicroGrid>: Calling for
proposals...
<pv2@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: GENERATION | Power resource [kW]:
0.0 | Price [€/kWh]: 0.045 | VSC max S: [kVA]: 60.0 | VSC efficiency: 0.95

{CFP} sent by <power_manager@MicroGrid> successfully received by <pv1@MicroGrid>: Calling for
proposals...
<pv1@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: GENERATION | Power resource [kW]:
0.0 | Price [€/kWh]: 0.035 | VSC max S: [kVA]: 100.0 | VSC efficiency: 0.95

{CFP} sent by <power_manager@MicroGrid> successfully received by <load1@MicroGrid>: Calling for
proposals...
<load1@MicroGrid> sending {PROPOSE} ACL message to <power_manager@MicroGrid>: Type: CONSUMPTION | P
[kW]: 29.227598165 | Q [kVar]: 21.07277874

{CFP} sent by <power_manager@MicroGrid> successfully received by <load3@MicroGrid>: Calling for
proposals...
<load3@MicroGrid> sending {PROPOSE} ACL message to <power_manager@MicroGrid>: Type: CONSUMPTION | P
[kW]: 8.7682794495 | Q [kVar]: 6.321833622

{CFP} sent by <power_manager@MicroGrid> successfully received by <load2@MicroGrid>: Calling for
proposals...
<load2@MicroGrid> sending {PROPOSE} ACL message to <power_manager@MicroGrid>: Type: CONSUMPTION | P
[kW]: 17.536558899 | Q [kVar]: 12.643667244

{CFP} sent by <power_manager@MicroGrid> successfully received by <battery1@MicroGrid>: Calling for
proposals...
CHARGE:
energy_available = [(max_soc - current_soc)/100] * energy_capacity
450000.0 = [(100.0 - 50.0)/100] * 900000.0
DISCHARGE:
energy_available = [(current_soc - min_soc)/100] * energy_capacity
405000.0 = [(50.0 - 5.0)/100] * 900000.0
MAX_CHARGE_POWER_OFFER, agent <battery1> is 23.75 [kW]
MAX_DISCHARGE_POWER_OFFER, agent <battery1> is 23.75 [kW]
MAX_CHARGE_REACTIVE_OFFER, agent <battery1> is 45.13244398434457 [kVar]
MAX_DISCHARGE_REACTIVE_OFFER, agent <battery1> is 45.13244398434457 [kVar]
<battery1@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: STORAGE | Charge power offer
[kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVar]: 45.13244398434457 |
Discharging reactive offer [kVar]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]: 50.0

{CFP} sent by <power_manager@MicroGrid> successfully received by <battery4@MicroGrid>: Calling for
proposals...
CHARGE:
energy_available = [(max_soc - current_soc)/100] * energy_capacity
450000.0 = [(100.0 - 50.0)/100] * 900000.0
DISCHARGE:
energy_available = [(current_soc - min_soc)/100] * energy_capacity
405000.0 = [(50.0 - 5.0)/100] * 900000.0
MAX_CHARGE_POWER_OFFER, agent <battery4> is 23.75 [kW]
MAX_DISCHARGE_POWER_OFFER, agent <battery4> is 23.75 [kW]
MAX_CHARGE_REACTIVE_OFFER, agent <battery4> is 45.13244398434457 [kVar]
MAX_DISCHARGE_REACTIVE_OFFER, agent <battery4> is 45.13244398434457 [kVar]
<battery4@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: STORAGE | Charge power offer
[kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVar]: 45.13244398434457 |
Discharging reactive offer [kVar]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]: 50.0

{CFP} sent by <power_manager@MicroGrid> successfully received by <pv3@MicroGrid>: Calling for
proposals...
<pv3@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: GENERATION | Power resource [kW]:
0.0 | Price [€/kWh]: 0.08 | VSC max S: [kVA]: 30.0 | VSC efficiency: 0.95

{CFP} sent by <power_manager@MicroGrid> successfully received by <battery3@MicroGrid>: Calling for
proposals...
CHARGE:
energy_available = [(max_soc - current_soc)/100] * energy_capacity
450000.0 = [(100.0 - 50.0)/100] * 900000.0
DISCHARGE:
energy_available = [(current_soc - min_soc)/100] * energy_capacity

```

```
405000.0 = [(50.0 - 5.0)/100] * 900000.0
MAX_CHARGE_POWER_OFFER, agent <battery3> is 23.75 [kW]
MAX_DISCHARGE_POWER_OFFER, agent <battery3> is 23.75 [kW]
MAX_CHARGE_REACTIVE_OFFER, agent <battery3> is 45.13244398434457 [kVAr]
MAX_DISCHARGE_REACTIVE_OFFER, agent <battery3> is 45.13244398434457 [kVAr]
<battery3@MicroGrid> sending {PROPOSE} to <power_manager@MicroGrid>: Type: STORAGE | Charge power offer
[kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVAr]: 45.13244398434457 |
Discharging reactive offer [kVAr]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]: 50.0

{CFP} sent by <power_manager@MicroGrid> successfully received by <load4@MicroGrid>: Calling for
proposals...
<load4@MicroGrid> sending {PROPOSE} ACL message to <power_manager@MicroGrid>: Type: CONSUMPTION | P
[kW]: 2.9227598165 | Q [kVAr]: 2.107277874
```

---

```
{PROPOSE} sent by <pv4@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
Power resource [kW]: 0.0 | Price [€/kWh]: 0.12 | VSC max S: [kVA]: 100.0 | VSC efficiency: 0.95

{PROPOSE} sent by <load3@MicroGrid> successfully received by <power_manager@MicroGrid>: Type:
CONSUMPTION | P [kW]: 8.7682794495 | Q [kVAr]: 6.321833622

{PROPOSE} sent by <battery1@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| Charge power offer [kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVAr]:
45.13244398434457 | Discharging reactive offer [kVAr]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]:
50.0

{PROPOSE} sent by <load2@MicroGrid> successfully received by <power_manager@MicroGrid>: Type:
CONSUMPTION | P [kW]: 17.536558899 | Q [kVAr]: 12.643667244

{PROPOSE} sent by <battery4@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| Charge power offer [kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVAr]:
45.13244398434457 | Discharging reactive offer [kVAr]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]:
50.0

{PROPOSE} sent by <load1@MicroGrid> successfully received by <power_manager@MicroGrid>: Type:
CONSUMPTION | P [kW]: 29.227598165 | Q [kVAr]: 21.07277874

{PROPOSE} sent by <pv1@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
Power resource [kW]: 0.0 | Price [€/kWh]: 0.035 | VSC max S: [kVA]: 100.0 | VSC efficiency: 0.95

{PROPOSE} sent by <pv2@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
Power resource [kW]: 0.0 | Price [€/kWh]: 0.045 | VSC max S: [kVA]: 60.0 | VSC efficiency: 0.95

{PROPOSE} sent by <pv3@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
Power resource [kW]: 0.0 | Price [€/kWh]: 0.08 | VSC max S: [kVA]: 30.0 | VSC efficiency: 0.95

{PROPOSE} sent by <battery3@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| Charge power offer [kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVAr]:
45.13244398434457 | Discharging reactive offer [kVAr]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]:
50.0

{PROPOSE} sent by <load4@MicroGrid> successfully received by <power_manager@MicroGrid>: Type:
CONSUMPTION | P [kW]: 2.9227598165 | Q [kVAr]: 2.107277874

{PROPOSE} sent by <battery2@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| Charge power offer [kW]: 23.75 | Discharge power offer [kW]: 23.75 | Charging reactive offer [kVAr]:
45.13244398434457 | Discharging reactive offer [kVAr]: 45.13244398434457 | Price [€/kWh]: 0.2 | SOC [%]:
50.0

{PROPOSE} sent by <wind@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION
| Power resource [kW]: 48.247 | Price [€/kWh]: 0.003 | VSC max S: [kVA]: 200.0 | VSC efficiency: 0.95

13 proposals have been received by agent <power_manager@MicroGrid>.
Control strategy selected to calculate set points: PEAK SHAVING.

Aggregate real power demand = + 8.7682794495 from <load3> + 17.536558899 from <load2> + 29.227598165
from <load1> + 2.9227598165 from <load4> = 58.45519633[W]

Aggregate reactive power demand = + 6.321833622 from <load3> + 12.643667244 from <load2> + 21.07277874
from <load1> + 2.107277874 from <load4> = 42.14555748[VAR]

Calculating Q set points for the generators...
```

```

<pv4> added to priorityQ vector.
<pv1> added to priorityQ vector.
<pv2> added to priorityQ vector.
<pv3> added to priorityQ vector.
<wind> added to priorityQ vector.

```

```

Sorted priorityQ vector:
<wind>, with a price of 0.003
<pv1>, with a price of 0.035
<pv2>, with a price of 0.045
<pv3>, with a price of 0.08
<pv4>, with a price of 0.12

```

```

Q set point for <wind>: 44.36374471578948 [kVar]
aggregate_q_demand = aggregate_q_demand - desired_q = 42.14555748 - 42.14555748 = 0.0 [kVar]
Q set point for <pv1>: 0.0 [kVar]
aggregate_q_demand = aggregate_q_demand - desired_q = 0.0 - 0.0 = 0.0 [kVar]
Q set point for <pv2>: 0.0 [kVar]
aggregate_q_demand = aggregate_q_demand - desired_q = 0.0 - 0.0 = 0.0 [kVar]
Q set point for <pv3>: 0.0 [kVar]
aggregate_q_demand = aggregate_q_demand - desired_q = 0.0 - 0.0 = 0.0 [kVar]
Q set point for <pv4>: 0.0 [kVar]
aggregate_q_demand = aggregate_q_demand - desired_q = 0.0 - 0.0 = 0.0 [kVar]

```

Calculating P set points for the batteries...

Debugging priorityBatt vector BEFORE SORTING...

```

priorityBatt(0) holds: <battery1> data: Charge power offer[W]: 23.75 | Discharge power offer[W]: 23.75 |
Charging reactive offer[VAR]: 45.13244398434457 | Discharging reactive offer[VAR]: 45.13244398434457 |
SOC[%]: 50.0 | Price[€/kWh]: 0.2
priorityBatt(1) holds: <battery4> data: Charge power offer[W]: 23.75 | Discharge power offer[W]: 23.75 |
Charging reactive offer[VAR]: 45.13244398434457 | Discharging reactive offer[VAR]: 45.13244398434457 |
SOC[%]: 50.0 | Price[€/kWh]: 0.2
priorityBatt(2) holds: <battery3> data: Charge power offer[W]: 23.75 | Discharge power offer[W]: 23.75 |
Charging reactive offer[VAR]: 45.13244398434457 | Discharging reactive offer[VAR]: 45.13244398434457 |
SOC[%]: 50.0 | Price[€/kWh]: 0.2
priorityBatt(3) holds: <battery2> data: Charge power offer[W]: 23.75 | Discharge power offer[W]: 23.75 |
Charging reactive offer[VAR]: 45.13244398434457 | Discharging reactive offer[VAR]: 45.13244398434457 |
SOC[%]: 50.0 | Price[€/kWh]: 0.2

```

MicroGrid power balance is: 14.727824204000008 [kW], consumption within power limits (-100.0 < 14.727824204000008 < 100.0). No further action to be taken.

STORAGE set points [P, Q] after adding P are:

```

<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <battery1@MicroGrid>: P set point [kW]: 0.0 | Q
set point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <battery4@MicroGrid>: P set point [kW]: 0.0 | Q
set point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <battery3@MicroGrid>: P set point [kW]: 0.0 | Q
set point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <battery2@MicroGrid>: P set point [kW]: 0.0 | Q
set point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <wind@MicroGrid>: P set point [kW]:
46.02881276421052 | Q set point [kVar]: 44.36374471578948
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <pv1@MicroGrid>: P set point [kW]: 0.0 | Q set
point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <pv2@MicroGrid>: P set point [kW]: 0.0 | Q set
point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <pv3@MicroGrid>: P set point [kW]: 0.0 | Q set
point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <pv4@MicroGrid>: P set point [kW]: 0.0 | Q set
point [kVar]: 0.0
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <load3@MicroGrid>: P set point [kW]: 8.7682794495
| Q set point [kVar]: 6.321833622
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <load2@MicroGrid>: P set point [kW]: 17.536558899
| Q set point [kVar]: 12.643667244
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <load1@MicroGrid>: P set point [kW]: 29.227598165
| Q set point [kVar]: 21.07277874
<power_manager@MicroGrid> sending {ACCEPT-PROPOSAL} to <load4@MicroGrid>: P set point [kW]: 2.9227598165
| Q set point [kVar]: 2.107277874

```

{ACCEPT-PROPOSAL} sent by <power\_manager@MicroGrid> successfully received by <battery1@MicroGrid>: P set point [kW]: 0.0 | Q set point [kVar]: 0.0

```

POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(51.0^2 - 0.0^2) - 0.0 = 51.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<battery1@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: STORAGE | P out [kW]: 0.0 | Q
out [kVar]: 0.0 | SOC [%]: 50.0

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <battery4@MicroGrid>: P set
point [kW]: 0.0 | Q set point [kVar]: 0.0
POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(51.0^2 - 0.0^2) - 0.0 = 51.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<battery4@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: STORAGE | P out [kW]: 0.0 | Q
out [kVar]: 0.0 | SOC [%]: 50.0

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <pv2@MicroGrid>: P set
point [kW]: 0.0 | Q set point [kVar]: 0.0
POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(60.0^2 - 0.0^2) - 0.0 = 60.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<pv2@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: GENERATION | P out [kW]: 0.0 | Q
out [kVar]: 0.0

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <battery2@MicroGrid>: P set
point [kW]: 0.0 | Q set point [kVar]: 0.0
POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(51.0^2 - 0.0^2) - 0.0 = 51.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<battery2@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: STORAGE | P out [kW]: 0.0 | Q
out [kVar]: 0.0 | SOC [%]: 50.0

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <load3@MicroGrid>: P set
point [kW]: 8.7682794495 | Q set point [kVar]: 6.321833622
<load3@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: CONSUMPTION | Real power consumed
[kW]: 8.7682794495 | Reactive power consumed [kVar]: 6.321833622

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <load2@MicroGrid>: P set
point [kW]: 17.536558899 | Q set point [kVar]: 12.643667244
<load2@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: CONSUMPTION | Real power consumed
[kW]: 17.536558899 | Reactive power consumed [kVar]: 12.643667244

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <load1@MicroGrid>: P set
point [kW]: 29.227598165 | Q set point [kVar]: 21.07277874
<load1@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: CONSUMPTION | Real power consumed
[kW]: 29.227598165 | Reactive power consumed [kVar]: 21.07277874

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <wind@MicroGrid>: P set
point [kW]: 46.02881276421052 | Q set point [kVar]: 44.36374471578948
POWER_LOSSES AFTER getQ(): 2.218187235789479
MARGIN_FOR_P = SQRT(200.0^2 - 42.14555748^2) - 2.218187235789479 = 193.29076904166604 [kW]
p_out = SetPointP * EFF = 46.02881276421052 * 0.95 = 43.72737212599999 [kW]
POWER_LOSSES AFTER getP(): 4.519627874000008
RETURNING POWER_LOSSES BY getPowerLosses(): 4.519627874000008
<wind@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: GENERATION | P out [kW]:
43.72737212599999 | Q out [kVar]: 42.14555748

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <pv1@MicroGrid>: P set
point [kW]: 0.0 | Q set point [kVar]: 0.0
POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(100.0^2 - 0.0^2) - 0.0 = 100.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<pv1@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: GENERATION | P out [kW]: 0.0 | Q
out [kVar]: 0.0

```

```

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <pv3@MicroGrid>: P set
point [kW]: 0.0 | Q set point [kVar]: 0.0
POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(30.0^2 - 0.0^2) - 0.0 = 30.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<pv3@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: GENERATION | P out [kW]: 0.0 | Q
out [kVar]: 0.0

{INFORM} sent by <load3@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: CONSUMPTION
| Real power consumed [kW]: 8.7682794495 | Reactive power consumed [kVar]: 6.321833622

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <battery3@MicroGrid>: P set
point [kW]: 0.0 | Q set point [kVar]: 0.0
POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(51.0^2 - 0.0^2) - 0.0 = 51.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<battery3@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: STORAGE | P out [kW]: 0.0 | Q
out [kVar]: 0.0 | SOC [%]: 50.0

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <load4@MicroGrid>: P set
point [kW]: 2.9227598165 | Q set point [kVar]: 2.107277874
<load4@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: CONSUMPTION | Real power consumed
[kW]: 2.9227598165 | Reactive power consumed [kVar]: 2.107277874

{ACCEPT-PROPOSAL} sent by <power_manager@MicroGrid> successfully received by <pv4@MicroGrid>: P set
point [kW]: 0.0 | Q set point [kVar]: 0.0
POWER_LOSSES AFTER getQ(): 0.0
MARGIN_FOR_P = SQRT(100.0^2 - 0.0^2) - 0.0 = 100.0 [kW]
p_out = SetPointP * EFF = 0.0 * 0.95 = 0.0 [kW]
POWER_LOSSES AFTER getP(): 0.0
RETURNING POWER_LOSSES BY getPowerLosses(): 0.0
<pv4@MicroGrid> sending {INFORM} to <power_manager@MicroGrid>: Type: GENERATION | P out [kW]: 0.0 | Q
out [kVar]: 0.0

{INFORM} sent by <load1@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: CONSUMPTION
| Real power consumed [kW]: 29.227598165 | Reactive power consumed [kVar]: 21.07277874

{INFORM} sent by <load2@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: CONSUMPTION
| Real power consumed [kW]: 17.536558899 | Reactive power consumed [kVar]: 12.643667244

{INFORM} sent by <load4@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: CONSUMPTION
| Real power consumed [kW]: 2.9227598165 | Reactive power consumed [kVar]: 2.107277874

<pv4> just added: | 0,0000 | 0,0000 | 0,0000 | to its DataOutVector.
<pv1> just added: | 0,0000 | 0,0000 | 0,0000 | to its DataOutVector.
<battery1> just added: | 0,00000000000000 | 0,00000000000000 | 0,00000000000000 | 50,00000000000000 |
0,00000000000000 | to its DataOutVector.
<wind> just added: | 43,7274 | 42,1456 | 4,5196 | to its DataOutVector.
<battery4> just added: | 0,00000000000000 | 0,00000000000000 | 0,00000000000000 | 50,00000000000000 |
0,00000000000000 | to its DataOutVector.

{INFORM} sent by <pv4@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
P out [kW]: 0.0 | Q out [kVar]: 0.0

{INFORM} sent by <pv1@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
P out [kW]: 0.0 | Q out [kVar]: 0.0

{INFORM} sent by <battery1@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| P out [kW]: 0.0 | Q out [kVar]: 0.0 | SOC [%]: 50.0
1. <battery1> power_batteries = 0.0 | p_out = 0.0
2. <battery1> power_batteries = 0.0

{INFORM} sent by <wind@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
P out [kW]: 43.72737212599999 | Q out [kVar]: 42.14555748

{INFORM} sent by <battery4@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| P out [kW]: 0.0 | Q out [kVar]: 0.0 | SOC [%]: 50.0
1. <battery4> power_batteries = 0.0 | p_out = 0.0
2. <battery4> power_batteries = 0.0

```



```
<battery2> just added: | 0,00000000000000 | 0,00000000000000 | 0,00000000000000 | 50,00000000000000 |
0,00000000000000 | to its DataOutVector.
<pv2> just added: | 0,0000 | 0,0000 | 0,0000 | to its DataOutVector.
<pv3> just added: | 0,0000 | 0,0000 | 0,0000 | to its DataOutVector.
<battery3> just added: | 0,00000000000000 | 0,00000000000000 | 0,00000000000000 | 50,00000000000000 |
0,00000000000000 | to its DataOutVector.

{INFORM} sent by <battery2@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| P out [kW]: 0.0 | Q out [kVAr]: 0.0 | SOC [%]: 50.0
1. <battery2> power_batteries = 0.0 | p_out = 0.0
2. <battery2> power_batteries = 0.0

{INFORM} sent by <pv2@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
P out [kW]: 0.0 | Q out [kVAr]: 0.0

{INFORM} sent by <pv3@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: GENERATION |
P out [kW]: 0.0 | Q out [kVAr]: 0.0

{INFORM} sent by <battery3@MicroGrid> successfully received by <power_manager@MicroGrid>: Type: STORAGE
| P out [kW]: 0.0 | Q out [kVAr]: 0.0 | SOC [%]: 50.0
1. <battery3> power_batteries = 0.0 | p_out = 0.0
2. <battery3> power_batteries = 0.0

<power_manager> just added: | 14,72782420400000800 | 14,72782420400000800 | 0,0000000000000000 |
1,0000000000000000 | 50,0000000000000000 | to its DataOutVector.

Final power net balance: 14.727824204000008 [kW]
Reactive power net balance: 0.0 [kVAr]
Global state of charge (SOC) of the storage system: 50.0 %
Power factor, as seen from the grid side: 1,0000000000000000

END OF FIPA-CONTRACT-NET PROTOCOL. ITERATION #1 of 17520.
```

---



## 7 Bibliografía

- [1] G. Piernavieja Izquierdo, «Microrredes con elevada penetración de Energías Renovables y Almacenamiento Energético,» de *Integración en red: soluciones para altas penetraciones eólicas en sistemas insulares*, Las Palmas de Gran Canaria, 2010.
- [2] European Comission, «Vision and Strategy for Europe's Electricity Networks of the Future,» *RTD info*, 2006.
- [3] P. Crossley, A. Beviz, J. M. Angulo y S. Arnaltes, «Smart energy systems: Transitioning renewables onto the grid,» *Renewable Energy Focus*, vol. 11, nº 5, pp. 54-56, 58-59, 2010.
- [4] C. A. Bes, *Control distribuido de micro-redes*, PFC Ingeniería Industrial, Universidad de Zaragoza, 2012.
- [5] J. A. Domínguez, *Servicios complementarios para la correcta gestión de una microrred*, Sangüesa, 2011.
- [6] S. A. Ramos, «Modelos y Optimización I: Apunte de Heurísticas,» noviembre 2007. [En línea]. Available: <http://materias.fi.uba.ar/7114/guiaTP.php>. [Último acceso: 16 8 2013].
- [7] A. Ramos, P. Sánchez, J. M. Ferrer, J. Barquín y P. Linares, «Modelos matemáticos de optimización,» septiembre 2010. [En línea]. Available: [http://www.gams.com/docs/contributed/modelado\\_en\\_gams.pdf](http://www.gams.com/docs/contributed/modelado_en_gams.pdf). [Último acceso: 19 8 2013].
- [8] J. J. Gómez Sanz y R. Fuentes, *Introducción a la tecnología de agentes*, Universidad Complutense de Madrid, 2003.
- [9] F. Bellifemine, G. Caire, G. Rimassa y A. Poggi, «JADE: A White Paper,» Septiembre 2003. [En línea]. Available: <http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf>. [Último acceso: 10 04 2013].
- [10] Univ. Politécnica de Valencia. Dept. Sistemas Informáticos y Computación, «SMA, curso de doctorado,» 2002. [En línea]. Available: <http://www.upv.es/sma/web/sma.htm>. [Último acceso: 5 8 2013].
- [11] Foundation for Intelligent Physical Agents, FIPA, «The Foundation for Intelligent Physical Agents,» 2012. [En línea]. Available: <http://www.fipa.org/specs/fipa00037/index.html>. [Último acceso: 23 Julio 2013].
- [12] G. Caire, «Jade Programming for Beginners,» 30 Junio 2009. [En línea]. Available: <http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf>. [Último acceso: 10 04 2013].
- [13] N. Hatziargyriou, H. Asano, R. Iravani y C. Marnay, «Microgrids,» *IEEE power & energy magazine*, nº Julio/Agosto, pp. 78-94, 2007.
- [14] O. Mohammed, M. Nayeem y A. Kaviani, «A laboratory based microgrid and distributed

- generation infrastructure for studying connectivity issues to operational power systems,» de *Power and Energy Society General Meeting, IEEE*, Minneapolis, MN, 2010.
- [15] H. Nwe Aung, A. M. Khambadkone, D. Srinivasan y T. Logenthiran, «Agent-based Intelligent Control for Real-time Operation of a Microgrid,» de *PEDES-POWER INDIA 2010 : Joint International Conference on Power Electronics, Drives and Energy Systems & 2010 Power India*, New Delhi, India, 2010.
- [16] C. Colson, M. Nehrir y R. Gunderson, «Multi-agent Microgrid Power Management,» de *18th IFAC World Congress*, Milán, Italia, 2011.
- [17] A. Ruiz-Álvarez, A. Colet-Subirach, F. Álvarez-Cuevas Figuerola, O. Gomis-Bellmunt y A. Suidrà-Andreu, «Operation of a Utility Connected Microgrid Using an IEC 61850-Based Multi-Level Management System,» *The IEEE Transactions on Smart Grid (TSG)*, vol. 3, nº 2, pp. 858 - 865, 2012.
- [18] M. Kouluri y R. Pandey, «Intelligent agent based micro grid control,» de *2nd International Conference on Intelligent Agent and Multi-Agent Systems (IAMA)* , Chennai, 2011.
- [19] M. Rajendram y R. Pandey, «Multi Agent Control for two area power system network,» de *International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, , Kumaracoil, 2012.
- [20] H. Bludszweit, «Simulation of mini-grids with high penetration of renewable energies and storage systems to manage energy,» Zaragoza, 2012.
- [21] C.-H. Yoo, W.-J. Choi, I.-Y. Chung, D.-J. Won, S.-S. Hong y B.-J. Jang, «Hardware-In-the-Loop Simulation of DC Microgrid with Multi-Agent System for Emergency Demand Response,» de *Power and Energy Society General Meeting*, San Diego, CA, 2012.
- [22] A. L. Dimeas y N. D. Hatziaargyriou, «A MAS architecture for Microgrids control,» de *Intelligent Systems Application to Power Systems, Proceedings of the 13th International Conference on*, Arlington, VA, 2005.
- [23] Foundation for Intelligent Physical Agents, FIPA, «The Foundation for Intelligent Physical Agents,» [En línea]. Available: <http://www.fipa.org/specs/fipa00029/SC00029H.html>. [Último acceso: 8 8 2013].
- [24] K. Jarvi, «RXTX,» 7 7 2011. [En línea]. Available: [http://rxtx.qbang.org/wiki/index.php/Main\\_Page](http://rxtx.qbang.org/wiki/index.php/Main_Page). [Último acceso: 2013 8 12].
- [25] A. Engler, «ISET's Microgrid Research: Past, Present, Future,» de *Symposium on Microgrids*, Nagoya, Japón, 2007.
- [26] C. R. Robinson, P. Mendham y T. Clarke, «Systems Design using Agent Control,» University of York, 2009. [En línea]. Available: <http://www.agentcontrol.co.uk/>. [Último acceso: 10 04 2013].
- [27] C. R. Robinson, P. Mendham y T. Clarke, «MACSimJX: A Tool for Enabling Agent Modelling with Simulink Using JADE,» *Journal of Physical Agents*, vol. 4, nº 3, Septiembre 2010.
- [28] R. Fourer, D. M. Gay y B. W. Kernighan, AMPL A Modeling Language for Mathematical

Programming, Segunda ed., Duxbury/Thomson, 2003.

- [29] D. Grimshaw, «JADE Administration Tutorial,» 26 Marzo 2010. [En línea]. Available: <http://jade.tilab.com/doc/tutorials/JADEAdmin/index.html>. [Último acceso: 10 04 2013].
- [30] F. Bellifemine, G. Caire, T. Trucco y G. Rimassa, «JADE Programmer's Guide,» 8 Abril 2010. [En línea]. Available: <http://jade.tilab.com/doc/programmersguide.pdf>. [Último acceso: 10 04 2013].
- [31] IEEE computer Society, «The Foundation for Intelligent Physical Agents,» 2013. [En línea]. Available: <http://www.fipa.org/index.html>. [Último acceso: 10 04 2013].
- [32] J. Lewis y W. Loftus, Java software solutions: foundations of program design, Boston: Addison-Wesley, 2012.