

Proyecto Fin de Carrera
Ingeniería Industrial



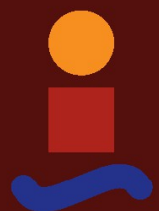
Análisis de Fallos, Simulación y Optimización
basada en Control Predictivo sobre planta real de
Generación Distribuida

Autor: José Podadera Borrego

Tutora: Ascensión Zafra Cabeza

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

Sevilla, 2017



Proyecto Fin de Carrera
Ingeniería Industrial

**Análisis de Fallos, Simulación y Optimización
basada en Control Predictivo sobre planta real de
Generación Distribuida.**

Autor:

José Podadera Borrego

Tutora:

Ascensión Zafra Cabeza

Profesora Titular de Universidad

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Proyecto Fin de Carrera:
Análisis de Fallos, Simulación y Optimización basada en Control Predictivo
sobre planta real de Generación Distribuida.

Autor: José Podadera Borrego

Tutora: Ascensión Zafra Cabeza

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

A mi padre.

Agradecimientos

A los pájaros del Juncarejo, con ellos he compartido mis dudas y me han ayudado a escribir correctamente las ecuaciones, no sólo en este proyecto, sino desde el principio de la carrera.

Nunca hubiera llegado al final de este camino sin la financiación que me han prestado, el mochuelo, el cuco, la perdiz y la grajilla. ¡Nuca dejéis de cantar!

Una planta de generación distribuida es una red eléctrica restringida a un área geográfica localizada, donde se ubican los generadores de energía y también los consumos (distribución). De esta forma se elimina la necesidad del transporte y los problemas que éste conlleva.

En este proyecto se estudia este sistema desde diferentes puntos de vista.

Por un lado se realiza un análisis operativo y de riesgos de fallo de los diferentes elementos que componen la red (microrred). También se cuantifican las pérdidas, y para ello es de utilidad la simulación realizada mediante el software OpenDSS.

Por otro lado se aborda la optimización del funcionamiento de la planta mediante técnicas de control predictivo basado en modelos, con tratamiento de datos de tipo estocástico.

Se construye un algoritmo al efecto, que utiliza la información recopilada en el análisis de riesgos para mejorar el tratamiento de datos, y que aprovecha también la simulación proporcionada por OpenDSS para mejorar el modelo de partida de la microrred, necesario para el control predictivo.

Índice

Agradecimientos	ix
Resumen	xi
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvii
Notación	xix
1 Antecedentes y estado actual de la generación distribuida	1
1.1. <i>La problemática actual. Soluciones</i>	1
1.1.1 Generación distribuída.	1
1.1.2 La reserva rodante virtual.	1
1.2. <i>Microgrid resuelta en este Proyecto</i>	2
1.3 <i>Objetivos del Trabajo</i>	4
1.4 <i>Estructura del Documento</i>	4
2 Análisis de fallos y operatividad	5
2.1 <i>Adaptación del modelo para análisis de fallos</i>	5
2.1.1 Metodologías aplicables al análisis de riesgos del proceso.	5
2.1.2 Adaptación del modelo según jerarquías de decisión.	7
2.2 <i>Análisis Cualitativo de Riesgos</i>	8
2.2.1 Relación de puntos críticos de la planta.	8
2.2.2 Descripción de riesgos e incertidumbres en la planta.	9
2.2.3 Listado de Modos de Fallo y Efectos (FMEA).	10
2.3 <i>Análisis Cuantitativo de Riesgos</i>	12
2.3.1 Método del percentil.	12
2.3.2 Estudio particular del Aerogenerador.	13
2.3.3 Toma de datos del Aerogenerador.	18
2.3.4 Métodos estadísticos para análisis de los datos de viento.	23
2.3.4 Estudio particular del generador fotovoltaico.	27
2.3.5 Cuantificación de Riesgos e Incertidumbres por equipos.	33
2.3.6 Árbol de Fallos.	34
3 Optimización mediante Control Predictivo basado en modelos	39
3.1 <i>Fundamentos Teóricos</i>	39
3.2 <i>Breve Historia. Algoritmos</i>	41
3.2.1 Algoritmos de primera generación.	41
3.2.2 Algoritmos de segunda generación.	41
3.2.3 MPC en la actualidad.	41
3.3 <i>Definición del modelo a resolver</i>	42
3.3.1 Diagrama del proceso.	42
3.3.2 Diagrama del control.	42
3.3.3 Horizonte de predicción y horizonte de control. Modelo de predicción.	44

3.3.4	Obtención de la Ley de Control.	46
3.4	<i>Implementación real</i>	49
3.4.1	Hipótesis utilizadas.	49
3.4.2	Implementación del modelo sin pérdidas en la red.	50
4	Simulación con Herramientas Informáticas	59
4.1	<i>Introducción a OpenDSS</i>	59
4.1.1	Elementos básicos en OpenDSS.	60
4.1.2	Programación en OpenDSS.	60
4.2	<i>Utilización y Control de OpenDSS desde Matlab</i>	70
4.2.1	Comunicación con OpenDSS y compilación desde Matlab.	70
4.2.2	Comandos y programación desde la consola de Matlab.	73
4.2.3	Ejecución de la función de comprobación del circuito.	75
4.2.4	Gráficas e interpretación de resultados.	81
4.2.5	Ejemplo resuelto de red de distribución IEEE13.	82
4.3	<i>Resolución mediante OpenDSS de la microrred INTA de 11 nodos</i>	87
4.3.1	Hipótesis iniciales.	87
4.3.2	Descripción de la microrred.	88
4.3.3	Modelado del proceso de electrólisis y su reacción inversa.	88
4.3.4	Modelado del proceso de carga descarga en las baterías Pb-ácido.	91
4.3.5	Modelado de transformadores e inversores en el BUS de corriente continua	92
4.3.6	Modelado de todas las líneas en la red de 11 nodos.	93
4.3.7	Compensación de energía reactiva.	95
4.3.8	Resultado y conclusiones de la simulación con OpenDSS.	95
4.4	<i>Simulación de la red INTA 11 nodos. Modelo corregido con OpenDSS</i>	103
4.4.1	Implementación del modelo sin pérdidas en la red.	103
4.4.2	El modelo de la red simulado y corregido con OpenDSS.	104
4.5	<i>Jerarquía de Operatividad</i>	107
4.6	<i>Análisis de los resultados</i>	107
5	Conclusiones	109
5.1	<i>Repaso histórico</i>	109
5.2	<i>Perspectivas futuras</i>	110
	Anexo 1: Ampliación sobre el algoritmo de control predictivo basado en modelos	111
	Anexo 2: Códigos usados para la programación en MATLAB® y OpenDSS	127
	Referencias	145

Índice de Tablas

Tabla 2-1.- Listado de modos de fallo y efectos (FMEA)	11
Tabla 2-2.- Pérdidas de producción asociadas a partes de la instalación eólica	14
Tabla 2-3.- Estadística de fallos en componentes de aerogeneradores	14
Tabla 2-4.- Velocidades de viento y frecuencias de aparición (horas sobre 24 h)	23
Tabla 2-5.- Equivalencia VIENTO-PRODUCCIÓN-TIEMPO	24
Tabla 2-6.- Distribución de potencias (producción por periodo programado 24 h)	26
Tabla 2-7.- Valores corregidos de producción eólica por horas (FIABILIDAD 90%)	27
Tabla 2-8.- Pérdidas de producción asociadas a la utilización de inversores	29
Tabla 2-9.- Distribución de potencias (producción -kW- por periodo programado -24h-)	32
Tabla 2-10.- Valores corregidos de producción fotovoltaica por horas (FIABILIDAD 90%)	33
Tabla 2-11.- Relación de riesgos e incertidumbres en la instalación	34
Tabla 2-12.- Resumen de cuantificación de incertidumbres	36
Tabla 2-13.- Resumen de pérdidas por operación	36
Tabla 4-1.- Dimensionamiento de líneas (sección del conductor)	94
Tabla 4-2.- Líneas dimensionadas en la instalación real (INTA)	94
Tabla 4-3.- Listado de pérdidas por líneas y elementos	96
Tabla 4-4.- Listado de voltajes por buses y nodos	98
Tabla 4-5.- Listado de intensidades por líneas y elementos	99
Tabla 4-6.- Pantalla de información en tiempo real de la simulación	107

Índice de Figuras

Figura 1-1.- Curva de demanda diaria	2
Figura 1-2.- Caso práctico a optimizar	3
Figura 2-1.- Diagrama de bloques del análisis de riesgos	6
Figura 2-2.- Jerarquía de operatividad	7
Figura 2-3.- Diagrama de bloques de la simulación del proceso	8
Figura 2-4.- Aerogenerador	13
Figura 2-5.- Cálculo de desviación típica mediante la distribución normal inversa	15
Figura 2-6.- Evolución de la tasa de fallo en los aerogeneradores actuales	16
Figura 2-7.- Fiabilidad y vida útil en una máquina. Distribución normal de Gauss.	17
Figura 2-8.- Velocidad media anual de viento, altura 80 metros	19
Figura 2-9.- Perfil de velocidades de viento en la capa límite superficial	19
Figura 2-10.- Datos de registro de mediciones de viento, altura 80 metros	21
Figura 2-11.- Distribución Weibull de frecuencia de velocidades de viento	22
Figura 2-12.- Curva nominal de producción eólica	24
Figura 2-13.- Curvas de producción eólica para un día aleatorio	25
Figura 2-14.- Curvas de producción eólica esperada, programadas mediante ajuste por mínimas distancias	26
Figura 2-15.- Campo fotovoltaico en El Arenosillo, Moguer, Huelva	28
Figura 2-16.- Curva de potencia fotovoltaica nominal sin corregir	31
Figura 2-17.- Curva de temperatura	31
Figura 2-18.- Curva de potencia fotovoltaica corregida, producción prevista.	32
Figura 2-19.- Árbol de cuantificación de incertidumbres en condiciones nominales	35
Figura 2-20.- Árbol de pérdidas por operación sobre condiciones nominales	37
Figura 3-1.- Estrategia de control predictivo	39
Figura 3-2.- Estructura básica del control predictivo	40
Figura 3-3.- Esquema del sistema de generación distribuida a modelar	42
Figura 3-4.- Diagrama de bloques para el algoritmo de control a modelar	43
Figura 3-5.- Diagrama en bucle cerrado para función de transferencia univariable	44
Figura 3-6.- Ley de control	46
Figura 3-7.- Representación de la función de Liapunov	50
Figura 3-8.- Planta experimental ETSI, propiedad de la Universidad de Sevilla	51
Figura 3-9.- Diagrama de flujos. Variables usadas en el modelo.	52
Figura 3-10.- Programación de entrada/salida de potencias (entrada de datos)	56
Figura 3-11.- Simulación de GPC usando Matlab	57
Figura 4-1.- Interfaz gráfica de OpenDSS	61
Figura 4-2.- Matriz de impedancias	63

Figura 4-3.- Definición del objeto capacitador	65
Figura 4-4.- Comunicación entre OpenDSS y Matlab	70
Figura 4-5.- Estructura de OpenDSS	70
Figura 4-6.- Esquema del IEE13 Node Test Feeder	83
Figura 4-7.- Relación entre las variables del modelo	88
Figura 4-8.- Serie temporal (24 h) de potencia (kW) consumidos por el vehículo híbrido	90
Figura 4-9.- Modelo experimental electrolizador-convertidor DC/DC	91
Figura 4-10.- Esquema unifilar ampliado a equipos AC/DC - DC/DC - DC/AC	92
Figura 4-11.- Esquema unifilar equivalente con BUSES en AC (modelo aproximado)	93
Figura 4-12.- Transitorio de evolución de potencia en descarga de batería	95
Figura 4-13.- Transitorio de evolución de potencia en generador eólico	96
Figura 4-14.- Geometría de buses. Representación de pérdidas en la red mediante grosor de líneas.	97
Figura 4-15.- Transitorio de evolución de potencia en generador fotovoltaico	97
Figura 4-16.- Transitorio de evolución de potencia en electrolizador	98
Figura 4-17.- Transitorio de evolución de potencia en pila de combustible	99
Figura 4-18.- Evolución del nivel de carga SOC en la batería	100
Figura 4-19.- Evolución del nivel de almacenamiento MHL de hidrógeno	101
Figura 4-20.- Evolución de potencias en el BUS de 400 v	101
Figura 4-21.- Evolución de potencias en el BUS de 50 v	102
Figura 4-22.- Evolución de potencias de intercambio con la red exterior y pérdidas	102
Figura 4-23.- Simulación 1 de GPC usando Matlab	103
Figura 4-24.- Simulación 2 de GPC usando Matlab	104
Figura 4-25.- Diferentes simulaciones de algoritmo GPC simultáneamente con OpenDSS	105
Figura 4-26.- Control predictivo sin restricciones. OpenDSS limita entradas y salidas	106
Figura 4-27.- Diagrama de bloques de la doble simulación simultánea	108
Figura 5-1.- Generador de los años 70, cortijo El Madroñal, Estepa (Sevilla)	109
Figura A1-1.- Relación entre variables del modelo	116

Notación

A^*	Conjugado
A'	Traspuesto
c.q.d.	Como queríamos demostrar
e.o.c.	En cualquier otro caso
E	número e
Re	Parte real
Im	Parte imaginaria
sen	Función seno
Tg	Función tangente
arctg	Función arco tangente
sen	Función seno
$\text{sen}^x y$	Función seno de x elevado a y
$\text{cos}^x y$	Función coseno de x elevado a y
Sa	Función sampling
sgn	Función signo
rect	Función rectángulo
Sinc	Función sinc
$\partial y \partial x$	Derivada parcial de y respecto
x°	Notación de grado, x grados.
$\text{Pr}(A)$	Probabilidad del suceso A
CA ó AC	Corriente Alterna (indistintamente cualquiera de los dos)
CC ó DC	Corriente Continua (indistintamente cualquiera de los dos)
:	Tal que
$<$	Menor estricto
$>$	Mayor estricto
\backslash	Backslash
\Leftrightarrow	Si y sólo si

1 ANTECEDENTES Y ESTADO ACTUAL DE LA GENERACIÓN DISTRIBUIDA

“Sólo sé que no sé nada”

- Sócrates -

A modo de introducción para este trabajo, haré un breve repaso al esquema eléctrico tradicional para el abastecimiento de energía, que ha prevalecido desde los tiempos en que el modelo de Nikola Tesla se impuso al de Alba Edison.

Pero el mundo ha cambiado; en aquella época una gran central hidroeléctrica podía abastecer toda la demanda de un país. La Corriente Alterna de Tesla triunfó porque había necesidad de transportar la energía a grandes distancias, y esto empezó a hacerse cada vez a voltaje mayor para minimizar pérdidas.

Pero hoy en día, la demanda eléctrica por habitante en un país desarrollado está tensionando el modelo cada vez más, acercándose peligrosamente a sus límites. La seguridad del suministro, la estabilidad de la red y el rechazo social a energías no limpias (me permito incluir aquí la fisión nuclear), hace necesario plantearse la forma de implantar paulatinamente un mayor aprovechamiento de energías renovables.

1.1. La problemática actual. Soluciones

1.1.1 Generación distribuida.

Frente al modelo tradicional mencionado anteriormente, de generación-transporte-distribución, describo aquí un modelo conocido como *‘de generación distribuida’*, que consiste en obtener la energía final a partir de recursos primarios (sean o no renovables) en el mismo emplazamiento donde se consume. Aquí es donde aparece el concepto de microrred o microgrid, o también de smartgrid, de forma que implícitamente nos referimos a una red de abastecimiento inteligente, que proveerá de energía de fuentes renovables según su disponibilidad, o procedente de un almacenamiento en caso contrario, pero siempre generada y almacenada en el lugar de consumo.

1.1.2 La reserva rodante virtual.

En este apartado haré una reflexión sobre la integración de las Energías Renovables (fundamentalmente Eólica y Fotovoltaica) en la red eléctrica nacional (peninsular). Existen dos barreras que han frenado en seco el enorme crecimiento de la producción renovable (sobre todo eólica), en la primera década de este siglo. No es asunto de los ingenieros (en mi opinión desde una perspectiva técnica) hablar de la política de incentivos o primas, por una determinada energía primaria frente a otra.

Por tanto, al hablar de dos barreras al crecimiento de la planta de renovables, comentaré la segunda de ellas; la que hace que se califiquen como energías no gestionables. A su vez, dividiré el problema en dos puntos de

vista aunque vienen a ser la misma cosa: que estas energías no se pueden almacenar y para ser inyectadas a la red deben “caber”, y que al ser inyectadas no deben comprometer la estabilidad del sistema en cuanto a tensión y frecuencia.

Empezaré explicando el segundo punto: tradicionalmente las grandes turbinas de generación de potencia en la red eléctrica se conocen como “reserva rodante”, ya que se mueven con muchísima inercia y son capaces de absorber picos instantáneos de inestabilidad sin perder las referencias de tensión y frecuencia. Pero está claro que si el porcentaje de generación por turbinas de vapor (paradigma de máquina síncrona) continúa descendiendo frente al aumento de turbinas eólicas (paradigma de máquina asíncrona), la reserva rodante es cada vez menos representativa. Las máquinas asíncronas no contribuyen a esta reserva puesto que en cuanto se produzca un hueco de tensión significativo, se desenganchan de la red.

El otro punto se puede ilustrar con la curva de demanda diaria de la red eléctrica, Figura 1-1. En ella observamos que hay picos diarios de consumo en los que toda la potencia existente en la planta de renovables podría ser absorbida, parando por supuesto turbinas rápidas o de punta (diesel, gas, ciclos combinados, hidráulica). Pero por otro lado hay periodos de demanda valle diarios, en los que el sistema queda abastecido en gran parte por las centrales de base (grandes turbinas de vapor) que no se pueden parar, sino sólo regular parcialmente. El hueco que queda en estos periodos de bajo consumo es donde puede caber o no caber la producción (sobre todo eólica, ya que hablamos del tramo nocturno donde no hay fotovoltaica), y en muchos casos el gestor de la red debe decidir la paralización de plantas renovables, si hay exceso de potencia generada.



Figura 1 -1. Curva de demanda diaria, fuente: REE.

Por tanto, los dos problemas técnicos analizados se solucionarían simplemente almacenando la energía producida y sobrante, para inyectarla posteriormente en un periodo de alta demanda y evitando así el arranque de grupos rápidos que queman combustible fósil. El problema de la estabilidad en tensión y frecuencia, aunque no es tan sencillo y hay otras formas de solucionarlo, se podría abordar también desde el punto de vista del presente trabajo, haciendo que las microgrids gestionen la energía sobrante en la generación distribuida de forma análoga a una “reserva rodante virtual”. Esto último se podría conseguir gracias a la evolución en tamaño y fiabilidad de los grupos electrónicos de potencia, y que podrían simular la onda de la red eléctrica, estabilizando el sincronismo y sustituyendo paulatinamente la dependencia de la reserva rodante inercial.

1.2. Microgrid resuelta en este Proyecto

Haré una descripción somera del caso práctico que se pretende optimizar , y se detallará con más exactitud más adelante, sobre todo en el capítulo 4.

La microrred se representa mediante un esquema unificar simplificado, en la siguiente figura.

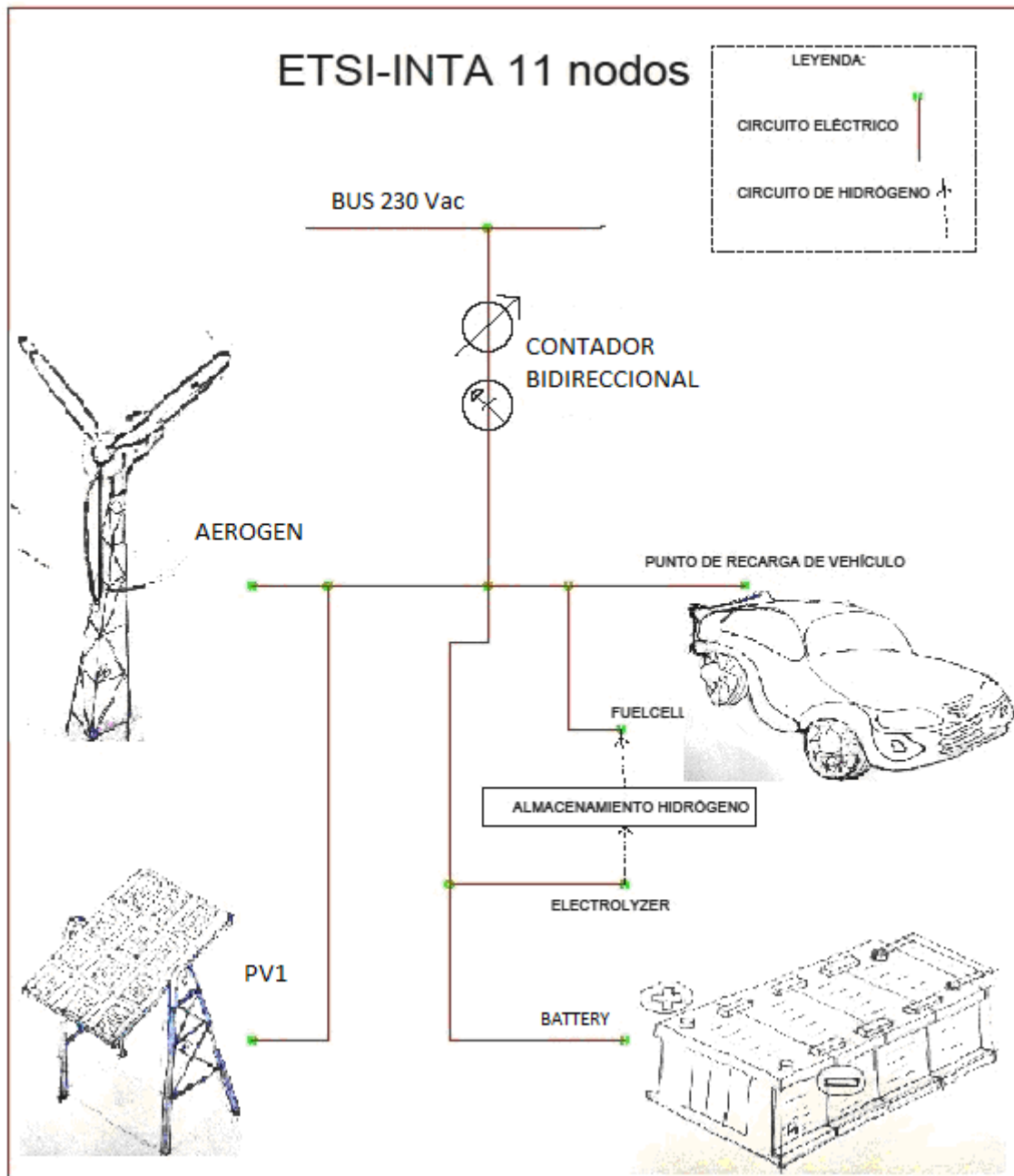


Figura 1-2. Caso práctico a optimizar: MICRORRED.

Para conectar nuestra microrred con la red de distribución, tenemos un transformador principal situado en el bus de referencia 230Vac, como puede verse en el unificar de la figura anterior. A través de este punto, la microgrid intercambia energía con el exterior mediante un contador bidireccional, comprando o vendiendo según disponibilidad o necesidad, respectivamente.

La disponibilidad de energía viene de los generadores de energía renovables AEROGEN y PV1, que son un generador mediante turbina eólica tripala de eje horizontal, y un generador solar fotovoltaico.

La necesidad de energía en la microrred es consecuencia de la fabricación de hidrógeno mediante electrólisis, ELECTROLYZER, que abastece automóviles eléctricos por pila de hidrógeno, FUELCELL.

El hidrógeno producido es almacenado en un depósito a presión mediante hidruros metálicos, teniendo la posibilidad de revertirse el proceso mediante pila de combustible. De esta forma, el almacenamiento puede ser

concebido como reserva para el funcionamiento de los vehículos y también como acopio de energía sobrante que luego puede ser usada en momentos en los que no exista generación procedente de energías renovables: la pila de combustible consume hidrógeno del depósito y lo transforma en energía eléctrica inyectada en la red, evitando así comprar el suministro del mercado eléctrico.

Una característica singular de la red modelada en este proyecto es que la pila de combustible está dentro de un automóvil, y que mientras éste está estacionado la pila puede estar produciendo electricidad a partir del hidrógeno almacenado (en el caso de que le sobre), inyectándola en la microrred para volverla a almacenar. Además, ¡No siempre tiene porque ser el mismo vehículo el que está operando en la red!

Con la misma filosofía de almacenamiento, se incluye en la red una batería de acumuladores Pb-ácido, representada como BATTERY en el esquema anterior.

1.3 Objetivos del Trabajo

Ya he comentado anteriormente la necesidad existente en nuestra sociedad, de cambiar el modelo energético actual, buscando un aprovechamiento más eficiente de las energías renovables, y la generación distribuida puede contribuir a la solución de este problema.

Pero el cambio de modelo hay que hacerlo garantizando la seguridad del suministro; hoy en día no se concibe que haya incertidumbres en la calidad y continuidad del abastecimiento de energía eléctrica. Hay un capítulo en este proyecto dedicado al análisis de fallos en una microrred de generación distribuida.

El otro gran objetivo de este trabajo está relacionado con la optimización mediante algoritmo integrado en el software de control del sistema en cuanto a su funcionamiento global. Se implementará atendiendo a criterios de eficiencia energética o minimización de pérdidas por un lado, y por otro al establecimiento de restricciones que eviten señales de entrada o salida que puedan dañar algún sistema aislado o toda la red en su conjunto.

Esto último se pretende conseguir mediante un algoritmo GPC (*'Generalized Predictive Control'*), usando técnicas de Control Predictivo basado en Modelos, que se explicará detalladamente.

1.4 Estructura del Documento

Además del capítulo I del trabajo, dedicado a esta breve introducción, el resto de capítulos se distribuyen de la siguiente forma:

En el capítulo II se profundiza en la descripción de la planta dedicada al aprovechamiento de energía de origen renovable mediante generación distribuida, desde el punto de vista de su operatividad. Se realiza un análisis de riesgos y fallos, y se derivarán algunas consecuencias que pueden servir para acotar las restricciones necesarias a la hora de implementar un algoritmo de control predictivo basado en modelos como el presentado en el capítulo siguiente.

El capítulo III está dedicado al Control Predictivo Basado en Modelos, con gran contenido teórico que después servirá para su aplicación a una planta real.

Para seguir avanzando en el desarrollo de nuestro caso práctico, pasamos en el siguiente capítulo de la teoría a la simulación mediante herramientas informáticas, en este caso con Matlab® y OpenDSS.

En el capítulo IV por tanto, se resuelven todas las cuestiones planteadas en los precedentes, y se presentan los resultados.

Finalmente, el capítulo V está dedicado a las conclusiones, y se incluyen además dos anexos de ampliación: el Anexo 1 para el desarrollo algebraico del algoritmo GPC y para el caso multivariable con restricciones aplicado en este proyecto; el Anexo 2 incluye los scripts o códigos usados en la resolución del problema mediante herramientas informáticas de simulación.

2 ANÁLISIS DE FALLOS Y OPERATIVIDAD

‘Parezco haber sido sólo un muchacho que jugase en la playa y se divirtiese encontrando a ratos una piedra más suave o una concha más bella, mientras el gran océano de la verdad permanecía sin descubrir delante de mí’

- Isaac Newton -

El análisis de fallos en la planta de generación distribuida presentada en este proyecto, se hace fundamentalmente en términos de minimización de fallos de operación por un lado, y por otro en términos de minimización de pérdidas de producción.

Este capítulo tiene estructura y contenido suficiente como para tener entidad propia dentro de cualquier estudio de operatividad, realizado en una planta de producción mediante energías renovables. En este caso, está concebido para complementar la simulación y optimización basada en control predictivo de una microrred, y se hará de dos formas:

1º) De forma cualitativa en este trabajo, identificando puntos críticos en los que en determinados momentos durante la operación de la planta, es necesario limitar o reducir los valores de potencia o intensidad (con respecto a los valores nominales). En un estudio más avanzado podría darse el salto a cuantificar (no sólo a identificar), en cada punto crítico y en condiciones de operación, si se superan los valores tolerables de riesgo. Habría que implementar un módulo de chequeo de fallos en el algoritmo completo, añadido al módulo de simulación y optimización que sí se implementa en este proyecto.

2º) De forma cuantitativa sí se aporta información sobre incertidumbres asociadas a la producción energética eólica y fotovoltaica, y a la demanda de energía para un coche híbrido, para un periodo típico de un día completo. Estos valores serán usados por el algoritmo de simulación y optimización, y éste actuará en consecuencia.

1.1 Adaptación del modelo para análisis de fallos

2.1.1 Metodologías aplicables al análisis de riesgos del proceso.

Antes de elegir alguna de las metodologías disponibles y ampliamente estudiadas en la bibliografía existente, es necesario tener clara la importancia de una correcta identificación de los posibles riesgos, para su posterior análisis y aplicación de metodología, que puede ser cualitativa o cuantitativa. Las dos opciones que más se pueden ajustar para un análisis cualitativo son las denominadas ‘HAZOP’ (Análisis de Riesgos y Operabilidad) y ‘FMEA’ (Análisis de Modos de Fallo y Efectos). Para pasar a un análisis cuantitativo (pudiendo usar la información generada por HAZOP o por FMEA), sería más conveniente un procedimiento tipo ÁRBOL. Las ideas fundamentales para elaboración de metodologías FMEA y ÁRBOL DE FALLO, han sido extraídas del texto de referencia [3].

La estrategia a seguir se resume de la siguiente forma:

→ Análisis cualitativo, identificando puntos críticos del proceso, tipo o modo de fallo y consecuencias del mismo.

→ Análisis probabilístico de fallo en cada uno de los puntos críticos identificados.

→ Análisis cuantitativo de todo el proceso, integrando toda la información de los dos pasos anteriores.

Arrancaremos de un diagrama de bloques genérico que nos servirá después para ser adaptado al proceso en estudio, figura 2-1.

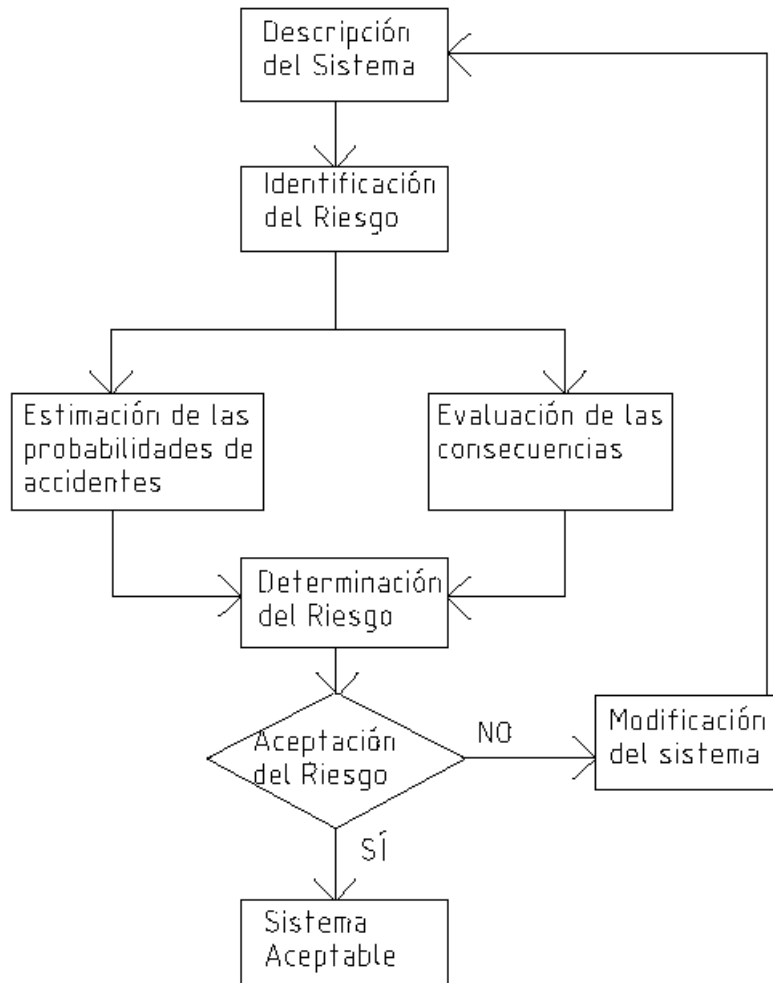


Figura 2-1. Diagrama de Bloques del Análisis de Riesgos.

La integración del diagrama de bloques anterior, con el diagrama de proceso del sistema productivo que se pretende analizar, se hará de forma jerárquica, es decir, las decisiones (ÓRDENES DE OPERACIÓN) que tenga que tomar el algoritmo que controla la planta están supeditadas a la escala de prioridades que se representa en la Figura 2-2. La más importante es la SEGURIDAD FÍSICA, que se refiere a la no ocurrencia de accidentes de cualquier tipo (por ejemplo, incendio por sobrecalentamiento en un conductor eléctrico, cortocircuito, etc).



Figura 2-2. Jerarquía de operatividad:
(ANÁLISIS DE FALLO → ÓRDENES DE OPERACIÓN ← SEÑALES DE OPTIMIZACIÓN)

2.1.2 Adaptación del modelo según jerarquías de decisión.

El análisis de riesgos una vez cuantificadas las incertidumbres, sobre todo en cuanto al dato de potencia disponible, o eficiencia de un equipo en un determinado periodo y en un determinado punto de la red, servirá como criterio de decisión del algoritmo de optimización de la planta, pudiendo modificar sensiblemente los valores esperados en las variables de estado y por tanto en las señales de control en un determinado periodo. Asimismo, y siguiendo la estructura jerárquica descrita anteriormente, se establecerán puntos de control dentro del proceso, que podrían hacer detener un determinado equipo o toda la planta, en función de que se sobrepasen los valores tolerables de riesgo.

En la figura 2-3 (página siguiente) se presenta, de manera igualmente simplificada, el diagrama de bloques para la simulación del proceso mediante puntos de chequeo de tolerancia al riesgo (puntos críticos). Como se ha dicho anteriormente, el bloque de simulación dedicado al análisis de fallo no se implementa en este proyecto junto con la optimización basada en control predictivo, lo cual sería objeto de un trabajo más ambicioso que integre ambos módulos dentro del algoritmo de simulación.

Al final del citado capítulo 4 también se incluye un diagrama de bloques de la simulación del proceso, pero en dicho caso representa una visión simplificada de la lógica del algoritmo de optimización, sin hacer referencia al análisis de fallos.

Conviene puntualizar aquí que el análisis de fallos del sistema de generación distribuida se realiza para una operación en modo “isla”, es decir desconectado de la red externa. Esto no quiere decir que la microrred no vaya a funcionar normalmente conectada a la red general. Lo que se pretende es aislar nuestro análisis de la influencia de incidencias como por ejemplo faltas, huecos de tensión, etc, que pueden obligar al sistema a “desengancharse” por seguridad.

2.2 Análisis Cualitativo de Riesgos

2.2.1 Relación de puntos críticos de la planta.

A continuación se relacionan los puntos críticos de la planta, para proceder posteriormente a la identificación cualitativa de riesgos que pueden provocar fallos en cada uno de los equipos descritos:

- a) Batería de acumuladores ácido-Pb
- b) Electrolizador
- c) Pila de combustible
- d) Transformadores AC-DC
- e) Transformadores DC-DC
- f) Inversor DC-AC
- g) Aerogenerador
- h) Instalación fotovoltaica
- i) Automatismo Digital de gestión de la red (Central de control la microrred)

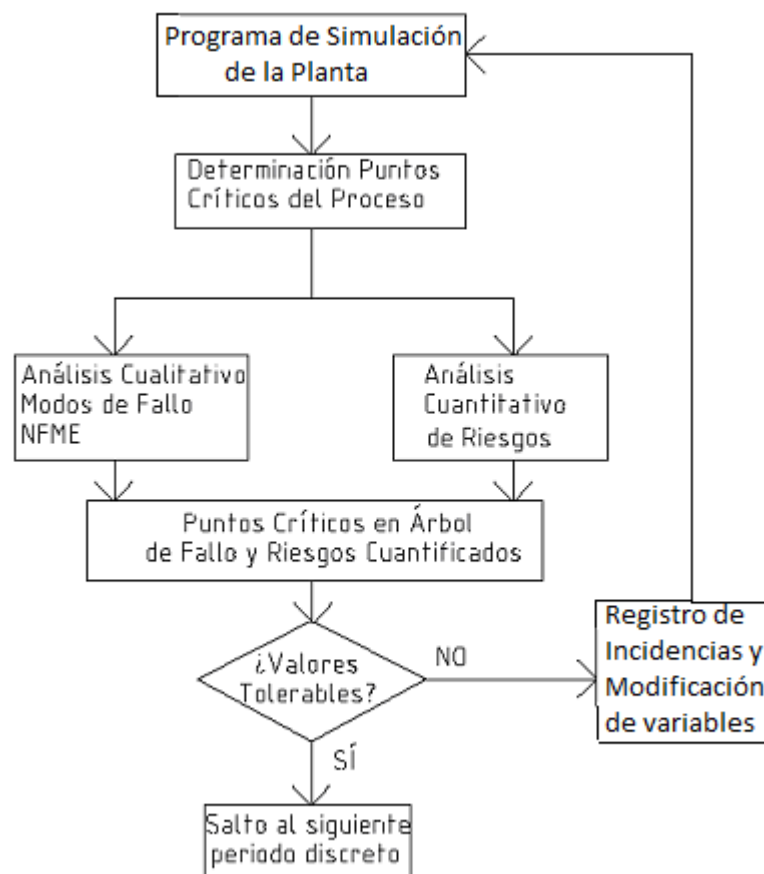


Figura 2 -3. Diagrama de bloques de la simulación del proceso.

2.2.2 Descripción de riesgos e incertidumbres en la planta.

La siguiente descripción cualitativa no exhaustiva, se hace con la información recopilada en las obras de consulta [3], [4], [5] y [6]. Posteriormente se cuantificarán los riesgos con apoyo de los mismos textos de referencia. Las pérdidas por envejecimiento suelen estimarse en un 0.5% anual en todos los casos. A modo de ejemplo, para el aerogenerador que es la máquina de mayor complejidad de todas, la incertidumbre total por fallo de cualquier tipo suele cuantificarse en un 5%, según [4].

BATERÍA DE ACUMULADORES.

Sobrecarga.

Sobrecalentamiento.

Acortamiento de vida útil por mala utilización.

Pérdida por envejecimiento.

Incertidumbres en eficiencia de ciclo carga/descarga.

ELECTROLIZADOR.

Sobrecarga.

Sobrecalentamiento.

Incendio o explosión.

Incertidumbres en el rendimiento.

PILA DE COMBUSTIBLE.

Sobrecarga.

Sobrecalentamiento.

Incendio o explosión.

Incertidumbres en el rendimiento.

TRAFOS DE CORRIENTE.

Sobrecarga.

Cortocircuito.

Corrientes parásitas, saturación.

Incertidumbres en las pérdidas del trafo.

INVERSOR DE CORRIENTE DC-AC.

Sobrecarga.

Cortocircuito.

Desequilibrio de la frecuencia/fases.

Ruido eléctrico. Armónicos.

AEROGENERADOR.

Fallo mecánico en Aerogenerador:

- Rodamientos eje de la turbina. Desequilibrio.
- Rodamiento de veleta / posicionamiento. Desequilibrio.
- Fallo en sistema de orientación en palas paso variable.
- Embalamiento.
- Pérdida de orientación, quedando la máquina a merced del temporal.
- Averías en engranajes. Engrase y lubricación.
- Rotura de palas. Fatiga.
- Fijación y estabilidad de mástil.

Fallo eléctrico:

- Pérdida de excitación.
- Pérdida de sincronismo.
- Sobrecarga generador.
- Cortocircuito.

Errores de medida y en sensores.

Pérdida de disponibilidad por mantenimiento.

Pérdida por envejecimiento.

GENERADOR FOTOVOLTAICO.

Suciedad, inclemencias meteorológicas.

Robos.

Pérdidas por errores del sistema de seguimiento.

Sobrecarga.

Cortocircuito.

Pérdida de disponibilidad por mantenimiento.

Pérdida por envejecimiento.

AUTOMATISMO DIGITAL DE GESTIÓN DE LA RED.

Sobrecarga.

Cortocircuito.

Fallo en aparatos de medida.

Fallo en protecciones/ seccionadores.

Corrientes bidireccionales.

Retardos, transitorios.

2.2.3 Listado de Modos de Fallo y Efectos (FMEA).

A partir del listado anterior de riesgos, vamos a implementar el método FMEA 'Failure Modes Effects and Analysis' tal y como se propone en el texto de referencia [3], y queda resumido mediante la tabla siguiente.

En la FMEA están sólo aquellos riesgos que se describen cualitativamente en cuanto a su modo de fallo. En un análisis posterior se cuantificarán otros riesgos, como por ejemplo incertidumbres por pérdida de disponibilidad, pérdidas por necesidades de mantenimiento, riesgo de robos, catástrofes naturales, etc...

Tabla 2-1.- Listado de Modos de Fallo y Efectos

<u>Item</u>	<u>Descripción</u>	<u>Modo de Fallo</u>	<u>Detección Física</u>	<u>Efectos</u>	<u>Reductor d Riesgo</u>
FME01	Baterías: sobreintensidad	DISPARO	RELÉ	PARADA PROCESO	GPC (u)
FME02	Baterías: sobrecalentam	CONSUMO	SONDA T ^a	PÉRDIDA EFICIENCIA	GPC(Δu)
FME03	Baterías: envejecimiento	DESCARGA	MULTÍMETR	PÉRDIDA EFICIENCIA	GPC (y)
FME04	Electroliz: sobreintensid	DISPARO	RELÉ	PARADA PROCESO	GPC (u)
FME05	Electroliz: sobrecalentam	CONSUMO	SONDA T ^a	PÉRDIDA EFICIENCIA	GPC(Δu)
FME06	Electrol: incendio-explos	IGNICIÓN H2	DET. LLAMA	CATASTRÓFICO	Mantenim
FME07	FuelCell: sobreintensidad	DISPARO	RELÉ	PARADA PROCESO	GPC (u)
FME08	FuelCell: sobrecalentam	CONSUMO	SONDA T ^a	PÉRDIDA EFICIENCIA	GPC(Δu)
FME09	FuelCell: incendio-explos	IGNICIÓN H2	DET. LLAMA	CATASTRÓFICO	Mantenim
FME10	Trafos: sobreintensidad	DISPARO	RELÉ	PARADA PROCESO	GPC (u)
FME11	Trafos: cortocircuito	DISPARO	MAGNETOT	PARADA PROCESO	Mantenim
FME12	Trafos: saturación	CONSUMO	MULTÍMETR	PÉRDIDA EFICIENCIA	GPC(Δu)
FME13	Inversor: sobreintensidad	DISPARO	RELÉ	PARADA PROCESO	GPC (u)
FME14	Inversor: cortocircuito	DISPARO	MAGNETOT	PARADA PROCESO	Mantenim
FME15	Inversor: perturbaciones	DESACOPLE	MULTÍMETR	PARADA PARCIAL	Autómata
FME16	Aerogen: fallo mecánico	ROTURA	MULTÍMETR	PARADA PARCIAL	Mantenim
FME17	Aerogen: fallo eléctrico	DISPARO	RELÉS	PARADA PARCIAL	Mantenim
FME18	Aerogen: fallos medidas	DESACOPLE	MULTÍMETR	PARADA PARCIAL	Autómata
FME19	Fotovoltaica: sobreintens	DISPARO	RELÉ	PARADA PARCIAL	Diseño
FME20	Fotovoltaica: cortocircuito	DISPARO	MAGNETOT	PARADA PARCIAL	Mantenim
FME21	Fotovoltaica: roturas	PÉRDIDA UD	MULTIMETR	PARADA PARCIAL	Mantenim
FME22	Autom. Digital: sobreinten	DISPARO	RELÉ	PARADA PROCESO	GPC (u)
FME23	Autom. Digital: cortocirc	DISPARO	MAGNETOT	PARADA PROCESO	Mantenim
FME24	Autom. Digital: desequilib	DISPARO	CORR. BIDIR	PARADA PROCESO	Autómata

Antes de pasar a cuantificar los riesgos conviene explicar el significado y la **importancia de la última columna del análisis FMEA**, sobre todo en cuanto las consecuencias que se pueden derivar para su conexión con el algoritmo GPC. Son decisiones que afectan a las siguientes variables: restricciones de la variable u , restricciones de la variable Δu y restricciones en los valores máximos y mínimos de la variable de salida y .

Con este detalle podemos ver que es posible interactuar dentro del triángulo de jerarquías presentado en la figura 2-2, de forma que decisiones de optimización pueden al mismo tiempo estar garantizando el funcionamiento seguro, al alejar los parámetros controlados (por ejemplo magnitudes de flujo o de nivel) de los niveles de emergencia.

En cualquier caso la jerarquía prevalece, y en caso de que se sobrepase la intensidad admisible en un conductor, disparará la protección sea cual sea la orden dada en ese momento a la planta por el algoritmo de optimización.

Conviene aquí hacer mención a las recomendaciones que aparecen en el libro de referencia [10], para describir el funcionamiento del sistema en modo seguro en este tipo de plantas de generación distribuida. Este procedimiento es también conocido como de “fallo seguro”, afecta a la lógica del procesador central de gestión, y atiende a algunos criterios fáciles de implantar como pueden ser:

- uso de pocos sensores (sólo los imprescindibles).
- sensores duplicados para cada punto crítico (es útil haber confeccionado antes un árbol de fallos, o bien un listado FMEA).
- relés con lógica negada, que disparan en caso de fallo del sensor (aunque esto no signifique fallo en la planta).

2.3 Análisis Cuantitativo de Riesgos

A la hora de proceder en un análisis de riesgos para una instalación o un proceso cualquiera, es importante definir si nos encontramos en un ambiente de Certidumbre, Incertidumbre o Riesgo. Cuando hay muchos equipos funcionando simultáneamente, y cualquiera de ellos puede fallar y hacerlo por diversas causas, no estamos en ambiente de Certidumbre.

El propósito de un estudio como este es situar el funcionamiento de la planta en un ambiente de Riesgo en el cual hemos podido cuantificar las probabilidades de fallo, sea por un conocimiento exhaustivo del proceso y de cada equipo, o bien por datos estadísticos de los mismos, etc.

En este trabajo no se hace un análisis en profundidad de todos los equipos (se recopilan datos específicos para dos de ellos), pero teniendo en cuenta la bibliografía especializada y consultada, tenemos acotadas las probabilidades de fallos y de pérdidas de potencia nominal.

Para cuantificar los riesgos, se desarrolla a continuación el método del percentil. Por ejemplo, en el caso del aerogenerador, que es una máquina compleja y puede fallar por muchas causas, se aplica el método tal y como se explica en el texto de referencia [4].

Otro aspecto que debo comentar antes de seguir adelante, es que se van a cuantificar los riesgos desde dos aspectos: probabilidad de pérdida de producción y probabilidad de fallo o parada de producción. Ambas están relacionadas porque **la probabilidad de que ocurra un evento o suceso de parada total, es equivalente a que ocurra una pérdida de producción del 100%**. Realmente no se van a calcular probabilidades, sino incertidumbres asociadas a una probabilidad de fallo del 10%, es decir, FIABILIDAD 90%.

2.3.1 Método del percentil.

Teniendo en cuenta lo anterior y usando el método propuesto en dicho Manual de referencia [4], se asume que para errores acotados por debajo del 10% y usando una distribución NORMAL de probabilidades, se puede obtener el dato de la desviación típica a partir de la estadística de fallo disponible, y se toma el dato promedio de producción como el nominal de la máquina. **El método del PERCENTIL 90 nos dice el valor de producción esperada con una FIABILIDAD del 90%**. Los pasos para obtener un valor de producción por

cada periodo estudiado son:

→ definir incertidumbres y cuantificar la desviación típica: σ .

→ obtener (por ejemplo, para el caso del aerogenerador) las estadísticas de viento y los datos de producción para el viento considerado. La información dada por el fabricante nos da el valor de producción media en el periodo, obtenida de datos estadísticos conocidos: μ .

→ obtenemos el percentil 90%: producción por periodo (aerogenerador) cuya probabilidad de ser superada es 90%, usando una distribución normal.

Posteriormente lo resumiré mediante una tabla para todos los equipos, pero a continuación se detalla el cálculo de los porcentajes de pérdidas y de riesgo de fallo para los casos de aeroturbina y de generador fotovoltaico.

2.3.2 Estudio particular del Aerogenerador.

Tenemos un generador eólico para la planta de generación distribuida objeto de este proyecto, modelo AERO5000W ESHIA, (ver foto, figura 2-4), y de las siguientes características:

Aerogenerador de eje horizontal, tripala. Salida 48-240 Vac trifásico, sin regulador de tensión. Potencia de 5 kW. Diámetro de rotor 5 metros y 200 kg de peso. Sistema de orientación por timón-veleta y frenado por cortocircuito.



Figura 2-4. Aerogenerador, web fabricante www.eshia.es.

Como ya se ha dicho anteriormente en este trabajo, se hace un estudio de fallos en los diferentes equipos en funcionamiento desde dos puntos de vista; por un lado las pérdidas implicadas en función de la potencia de utilización (% del valor nominal), y por otro lado la incertidumbre de fallo o parada total del equipo.

Para el caso concreto del generador eólico, procedo a continuación a reflejar datos procedentes de bibliografía especializada, como por ejemplo el manual de Aerogeneradores [5], para el caso de las pérdidas de producción asociadas a determinadas partes de la instalación. La tabla 2-2 que se presenta a continuación se refiere a **condiciones nominales** de funcionamiento de la aeroturbina.

INCERTIDUMBRE POR FALLO O PARADA NO PREVISTA DE LA PLANTA PV:

Para la cuantificación de probabilidades de fallo del equipo, comenzaré tomando datos que puedan relacionar cada uno de los riesgos que en la subsección 2.2.2 se han descrito de forma cualitativa, con un valor que represente la probabilidad de fallo del elemento implicado. Posteriormente se obtendrá la probabilidad de fallo

de la máquina en su conjunto.

Tabla 2-2.- Pérdidas de producción asociadas a partes de la instalación de generación de energía eólica.

PARTE INSTALACIÓN	RENDIMIENTO	PÉRDIDA DE POTENCIA
Transmisión mecánica	98 %	2 %
Generador eléctrico	95 %	5 %
Transformador regulador	98 %	2 %

Una primera toma de datos es obtenida también para este caso del libro [5] de referencia en la bibliografía, y del texto [10] también del mismo autor, según se ve en la siguiente tabla.

Tabla 2-3.- Estadística de fallos en componentes de aerogeneradores.

COMPONENTE	FIABILIDAD	TASA DE FALLO (fallos/año)
Rotura punta de pala	0.5334	1.00e-4
Mecanismo de orientación	0.9013	0.115e-4
Generador	0.99305	0.769e-6
Multiplicador	0.9944	0.63e-6
Freno Mecánico	0.999	2.16e-6
Palas	0.9068	1.116e-5
Pernos	0.9068	1.116e-5
Cubo del rotor	0.9068	1.116e-5
Torre y anclaje torre	0.9997	1.000e-7

Vemos que en todos los casos la fiabilidad es mayor del 90% menos en caso de rotura de la punta de pala, debido a su complejidad (los grandes aerogeneradores disponen de un mecanismo de orientación de punta de pala, que hace que al girarla provoque la entrada en pérdida aerodinámica actuando de freno de seguridad).

En nuestro caso, la aeroturbina no dispone de este elemento, ni tampoco de freno mecánico ni de mecanismo servo electrónico de orientación, por lo que se excluyen del siguiente cálculo. Mediante la siguiente expresión vamos a obtener el equivalente a la desviación típica en cuanto a probabilidad de fallo de toda la máquina, con los datos de la tabla anterior.

El dato de la fiabilidad se refiere a un año, y nos permite calcular la desviación típica mediante la distribución normal inversa, tomando como valor medio el número de equipos que deben estar funcionando durante un año para que falle uno de ellos. Se hace de forma iterativa y mediante la herramienta de la hoja EXCEL, como se

ilustra en la siguiente figura 2-5:

Ejemplo elemento 1: rotura punta de pala.

Tasa de Fallo (DATO): falla 1 de cada 10000
 Fiabilidad (DATO): 0,5334
 Media igual a población 10000
 Desviación típica, a calcular a partir de los datos:

Hagamos un cálculo iterativo mediante la Distribución Normal Inversa, dada la probabilidad acumulada (Fiabilidad), vamos moviendo la desviación típica hasta obtener una media igual a la población+1 unidad

DATOS:

Fiabilidad	Tasa Fallo
0,5334	1,00E-04
0,9013	1,15E-05
0,99305	7,69E-07
0,9944	6,30E-07
0,999	2,16E-06
0,9068	1,12E-05
0,9068	1,12E-05
0,9068	1,12E-05
0,9997	1,00E-07

valor medio población (fijado)	probabilidad acumulada fiabilidad (fijado)	parámetro (a variar)	valor buscado media+1 unidad	DESVIACION TÍPICA (%):
1,00E+04	0,5334	20	1,88	0,2
8,70E+04	0,9013	1	1,29	0,00115
1,30E+06	0,99305	0,5	1,23	3,85E-005
1,59E+06	0,9944	0,5	1,27	0,0000315
4,63E+05	0,999	0,5	1,55	0,000108
8,96E+04	0,9068	1	1,32	0,001116
8,96E+04	0,9068	1	1,32	0,001116
8,96E+04	0,9068	1	1,32	0,001116
1,00E+07	0,9997	0,5	1,72	0,000005

F= FIABILIDAD

valor=DISTRNORMAL.INV(F;μ;σ) →Dada la media μ y la desviación típica σ, obtenemos el valor de población cuya probabilidad de ser superado es del F*100%.

Figura 2-5. Cálculo de la desviación típica mediante la Distribución Normal Inversa, herramienta EXCEL.

La hoja de cálculo anterior se explicará en la página 17, con la figura 2-7. Llamando σ a la desviación típica:

- σ1= 0.0000385% GENERADOR
- σ2= 0.0000315% MULTIPLICADOR
- σ3= 0.001116% PALAS
- σ4= 0.001116% PERNOS
- σ5= 0.001116% CUBO DEL RÓTOR
- σ6= 0.000005% TORRE Y ANCLAJE DE TORRE

$$\sigma = \text{sqrt}(\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2 + \sigma_5^2 + \sigma_6^2) = \text{sqrt}(3.85e-7^2 + 3.15e-7^2 + 11.16e-6^2 + 11.16e-6^2 + 11.16e-6^2 + 5e-8^2) = 1.93e-5$$

(esta forma de proceder en cuanto a las incertidumbres ha sido extraída del texto de referencia [4]); se convierte el valor de tanto por unidad a tanto por ciento, resultando:

$$\sigma = 0.00193\%$$

Se ha calculado, expresado en porcentaje, el valor de la desviación típica asociada al hecho de que durante el

transcurso de un año, el aerogenerador sufra una parada imprevista por rotura de determinados elementos.

Pero esta información no está realmente completa si no tenemos hacemos un estudio a lo largo de toda la vida útil de la máquina, teniendo en cuenta las necesidad de planificar mantenimientos preventivos periódicos que es lo que asegura que se cumplan las estadísticas de fallo. A los datos reflejados en la tabla 2-3, referidos a parada imprevista, añadimos el dato de parada planificada por mantenimiento, que en el caso típico de aerogeneradores se produce 2 veces al año según el libro [10].

Tomando como referencia un año sin que se produzca fallo no previsto con una fiabilidad mayor del 90%, la tasa de fallo en los aerogeneradores actuales ha mejorado dicho valor de referencia, según la gráfica obtenida de la misma fuente y que se ilustra en la figura 2-6.

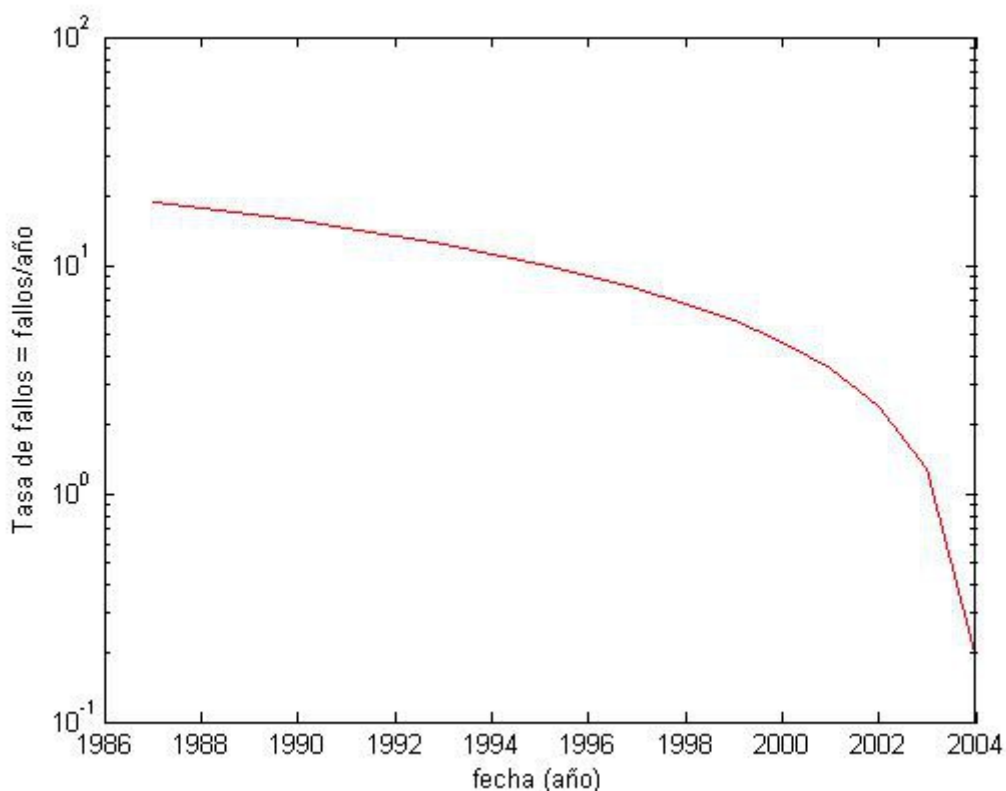


Figura 2-6. Evolución de la tasa de fallo en los aerogeneradores actuales, fuente [10].

No obstante lo anterior, las sucesivas operaciones de mantenimiento preventivo (casi podemos hablar de predictivo), se realizan en cualquier caso mucho antes de transcurrido un año, y las máquinas actuales se diseñan para una vida de 20 años sin fallar, siempre y cuando se cumpla la programación de actuaciones preventivas.

El coste de una avería por fallo imprevisto es muy elevado (sin entrar en cantidades, en nuestro libro de referencia [10] se explica) en comparación con el coste de las paradas programadas.

Vamos a obtener un valor de la incertidumbre asociada a un fallo o parada no programada a partir de los datos anteriores. Usamos la misma herramienta de EXCEL ilustrada en la figura 2-5, y ello sirve también para explicar cómo funciona la hoja de cálculo representada en dicha figura:

Tasa (dato): vida útil 20 años por máquina (1 fallo/año cada 20 máquinas)

Población: 20 máquinas (la inversa de la tasa)

Media: 19 máquinas no fallan (1 falla)

Fiabilidad (dato): 0.90 (valor método percentil, 90% de que hay 20 máquinas que no fallan)

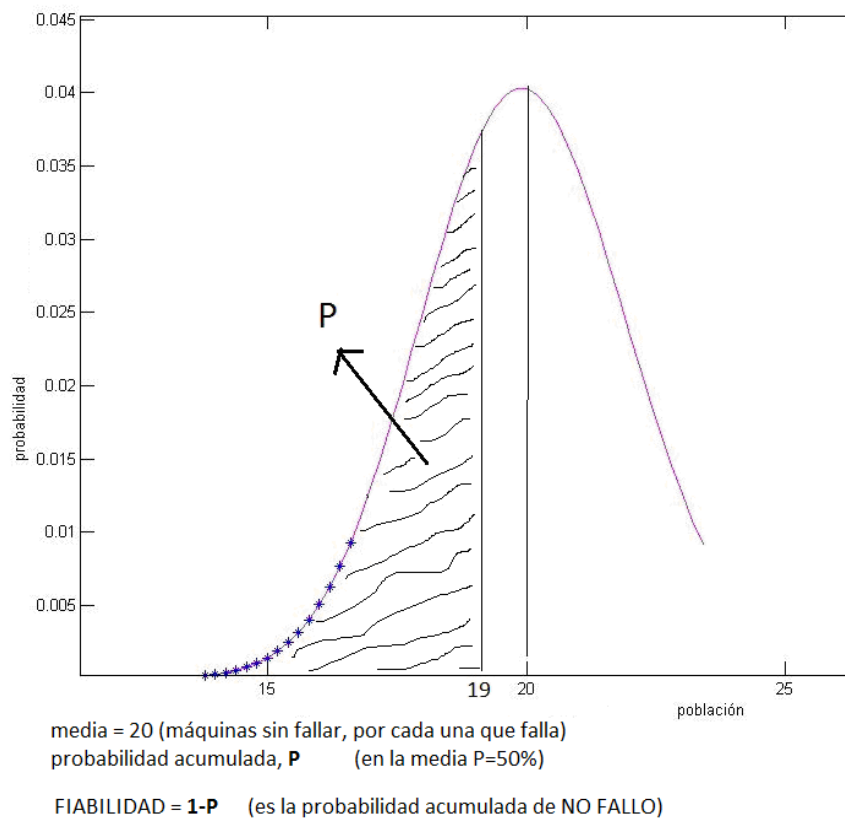


Figura 2-7. Fiabilidad y vida útil en una máquina, distribución normal de Gauss.

Es necesario iterar repetidamente hasta obtener en el lado izquierdo de la ecuación, un valor aproximadamente igual al de la media más 1 unidad.

Es decir, aunque en la figura 2-7 anterior la media representada es 20, en este cálculo estamos usando los valores: **media=19 (máquinas sin fallar por cada una que falla), suceso=20, despejamos σ .**

$$\rightarrow \text{valor} = \text{DISTR.NORMAL.INV}(F; \rho; \sigma)$$

$$20 = \text{DISTR.NORMAL.INV}(0.9; 19; 0.5) = 19.64 \quad \Leftrightarrow \text{NO CUMPLE}$$

$$20 = \text{DISTR.NORMAL.INV}(0.9; 19; 0.6) = 19.77 \quad \Leftrightarrow \text{NO CUMPLE}$$

$$20 = \text{DISTR.NORMAL.INV}(0.9; 19; 0.7) = 19.90 \quad \Leftrightarrow \text{NO CUMPLE}$$

$$20 = \text{DISTR.NORMAL.INV}(0.9; 19; 0.75) = 19.96 \quad \Leftrightarrow \text{NO CUMPLE}$$

$$20 = \text{DISTR.NORMAL.INV}(0.9; 19; 0.77) = 19.99 \quad \Leftrightarrow \text{CUMPLE}$$

$$\rightarrow \sigma(\%) = 0.77 * 100 / 20 = 3.85\%$$

Tomaremos este valor, que engloba la incertidumbre por fallo de la máquina en su conjunto, en comparación con el que se obtenía analizando los respectivos componentes por separado.

→ $\sigma = 3.85\%$ INCERTIDUMBRE ASOCIADA A PARADA IMPREVISTA DEL EQUIPO

Se ha calculado, expresado en porcentaje, el valor de la desviación típica asociado a la probabilidad de fallo o parada de la máquina, por cualquier motivo imprevisto. Puede expresarse también como TASA DE FALLO.

INCERTIDUMBRE POR PÉRDIDA DE PRODUCCIÓN:

Para la cuantificación de incertidumbres en las pérdidas de producción del equipo completo, tomamos todos los datos recopilados hasta ahora en las diferentes consultas bibliográficas referenciadas:

$\sigma_1 = 3\%$ Incertidumbre por errores de medida, y por errores en la toma y procesamiento de datos.

$\sigma_2 = 3.85\%$ Pérdidas de producción asociadas a parada imprevista.

$\sigma_3 = 0.5\%$ Pérdidas de producción por envejecimiento.

$\sigma_4 = 1\%$ Incertidumbre por catástrofes naturales, robos, etc.

$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2} = 5.01\%$

Observemos que la incertidumbre total se ajusta al valor previsto para pérdidas totales de cualquier tipo, según la bibliografía especializada [4] con respecto a estos equipos.

→ $\sigma = 5\%$ INCERTIDUMBRE ASOCIADA A PÉRDIDA DE PRODUCCIÓN DEL EQUIPO

2.3.3 Toma de datos del Aerogenerador.

Para poder aplicar el método del percentil procedemos como ya se ha explicado anteriormente:

1º) Cálculo de la desviación típica: $\sigma = 5\%$ (calculado en el apartado anterior).

2º) Cálculo del valor promedio: ρ . Conocemos el valor de producción nominal de la máquina, pero no es suficiente para el caso de un generador eólico, y por eso se hace un estudio específico más exhaustivo para esta máquina.

El dato de producción media será corregido por periodos de 24 horas, simulando una producción diaria a partir de diversas fuentes de datos de estadísticas de viento conocidas para la zona donde se sitúa la planta.

Se hará un tratamiento estocástico de dichos datos, además de ir recopilando algunos más que son necesarios, como por ejemplo referentes a la información geográfica:

→ la planta se sitúa en las instalaciones que posee el INTA en El Arenosillo, en el término municipal de Moguer, en la Provincia de Huelva.

→ la velocidad media de viento (valor promedio más probable) para esa zona es de 6,5 m/s, según datos del IDAE (Ministerio de Industria, Turismo y Comercio, Figura 3-8).

La información facilitada por el IDAE se refiere a una altura del generador de 80 metros, que es la suficiente como para situarse en la zona de perfil de velocidad constante de la capa límite superficial (Figura 2-9), en zona libre de obstáculos (edificios, etc).

MAPA EÓLICO DE ANDALUCÍA

Velocidad Media Anual a 80 m de altura

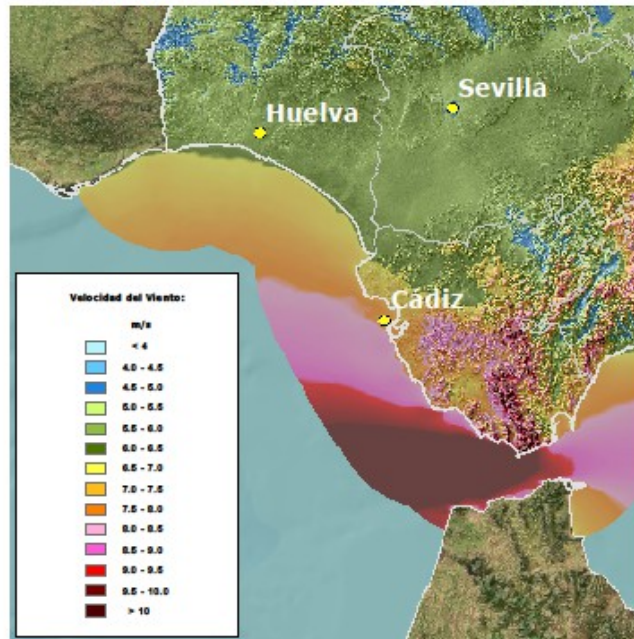


Figura 2-8. Velocidad media anual de viento, altura 80 metros, fuente IDAE.

Según los datos disponibles para la planta real que se modela en este proyecto, la altura del mástil sobre el que se sitúa el aerogenerador es de 20 metros, por lo que corregiré el valor de velocidad media (más probable), con apoyo del libro de referencia [7].

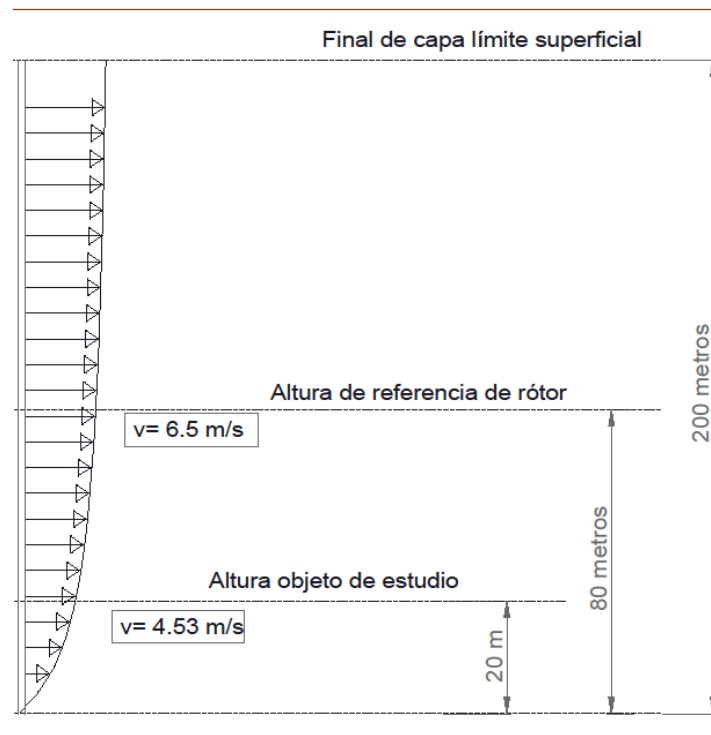


Figura 2-9. Perfil de velocidades de viento en la capa límite superficial.

En el texto citado anteriormente se indica que el perfil vertical de velocidades de viento para el desarrollo de capa límite en la figura 2-9, puede modelarse por la siguiente expresión:

$$v(z) = v_a \left[\frac{z - z_0}{z_a} \right]^\alpha \quad (0-2)$$

siendo:

$v(z)$: velocidad media de viento a la altura objeto de estudio.
 $z = z_R$, altura objeto de estudio (altura del rotor del generador).
 v_a : velocidad a la altura de referencia, conocida.
 z_a : altura de referencia .

z_0 : corrección por obstáculos en el terreno.

α : parámetro dependiente de la rugosidad superficial y de la estabilidad atmosférica media.

Los dos últimos parámetros son desconocidos, y serán calculados mediante las correlaciones propuestas por el autor del texto [7]. Para poder aplicar dichas correlaciones también es necesaria la verificación de las siguientes hipótesis:

La rugosidad del terreno corresponde a un obstáculo densamente distribuido sobre el mismo: HIPÓTESIS 1.

Se debe cumplir que $h_c / z_R \leq 0.75$: HIPÓTESIS 2.

La hipótesis 1 se cumple, ya que la zona donde se sitúa el aerogenerador está rodeada en todas direcciones por un bosque de pinos uniforme, de altura media aproximada 8 metros ($h_c = 8$).

Se puede comprobar fácilmente que también se verifica la hipótesis 2: $h_c / z_R = 8 / 20 < 0.75$.

Las correlaciones para z_0 y α son las siguientes:

$$z_0 = 0.056 h_c^\alpha ; \quad \alpha = \frac{0.37 - 0.088 \ln(v_a)}{1 - 0.088 \ln(0.1 z_a)} ; \quad (2-2)$$

Sustituyendo por los valores conocidos obtenemos:

$$\rightarrow \alpha = 0.2513 \rightarrow z_0 = 0.967 \quad \text{tomamos: } z_0 = 1$$

$$\rightarrow v(z = z_R) = 4.5292 \quad \text{tomamos: } V_m = 4.53 \text{ m/s}$$

La estadística de viento diario para esta zona puede representarse mediante una Distribución Weibull de frecuencia, la cual se obtiene siguiendo la siguiente formulación, tal y como se desarrolla en el texto de referencia [4].

Es importante resaltar que la distribución de Weibull representa muy bien el comportamiento del viento cuando se compara con las curvas de distribución de frecuencia de velocidades, obtenidas mediante toma de datos procedentes de mediciones con anemómetros.

Densidad de probabilidad:

$$f(v) = \frac{k}{c} \left[\frac{v}{c} \right]^{k-1} \exp \left[- \left(\frac{v}{c} \right)^k \right] \quad (3-2)$$

Para aclarar ideas, hago notar aquí el parecido de esta distribución con la normal de Gauss si no fuera por la existencia de asimetría, como se puede ver en la figura 2-11, en la página siguiente.

La pareja de valores (ρ, σ) que caracteriza inequívocamente una distribución normal, se relaciona con los parámetros de Weibull (k, C) , que se interpretan de forma análoga que los de la distribución normal. Es posible calcular esta pareja de valores a partir de la distribución de velocidades construida con las mediciones realizadas.

Suponiendo que conocemos la media ρ y la desviación típica σ del emplazamiento en estudio:

$$k = \left[\frac{\sigma}{\rho} \right]^{-1.086} ; \quad (\text{adimensional}); \quad C = \frac{\rho}{\Gamma[1+1/k]} ; \quad (\text{con unidades en m/s}). \quad (4-2)$$

siendo la función gamma definida como, $\Gamma(n) =$

$$\Gamma(n) = \int_0^{\infty} e^{-x} x^{n-1} dx$$

Que está tabulada, y para $K=2$ toma el valor:

$$\Gamma(1+0.5) = 0.88623; \quad \text{por tanto el factor de escala vale:} \quad C = 4.53 / 0.88623 = 5.11 \text{ m/s}$$



Figura 2-10. Datos de registros de mediciones de viento, altura 80 metros, fuente 'IDAE'.

En los datos obtenidos para el emplazamiento del generador eólico, a partir de mediciones de viento en la web

del IDAE, tenemos la frecuencia de aparición para cada velocidad junto con la orientación de viento. Este último dato es importante para determinados modelos de turbina, pero en nuestro caso la máquina cuenta con auto posicionamiento mediante veleta, por lo que la dirección del viento no afecta a la potencia producida.

A falta de datos más exactos para el emplazamiento, y dado el registro disponible de mediciones de viento anteriormente representado en la figura 2-10, se ha tomado como Factor de asimetría $k=2$ (que se considera óptimo en este tipo de distribuciones).

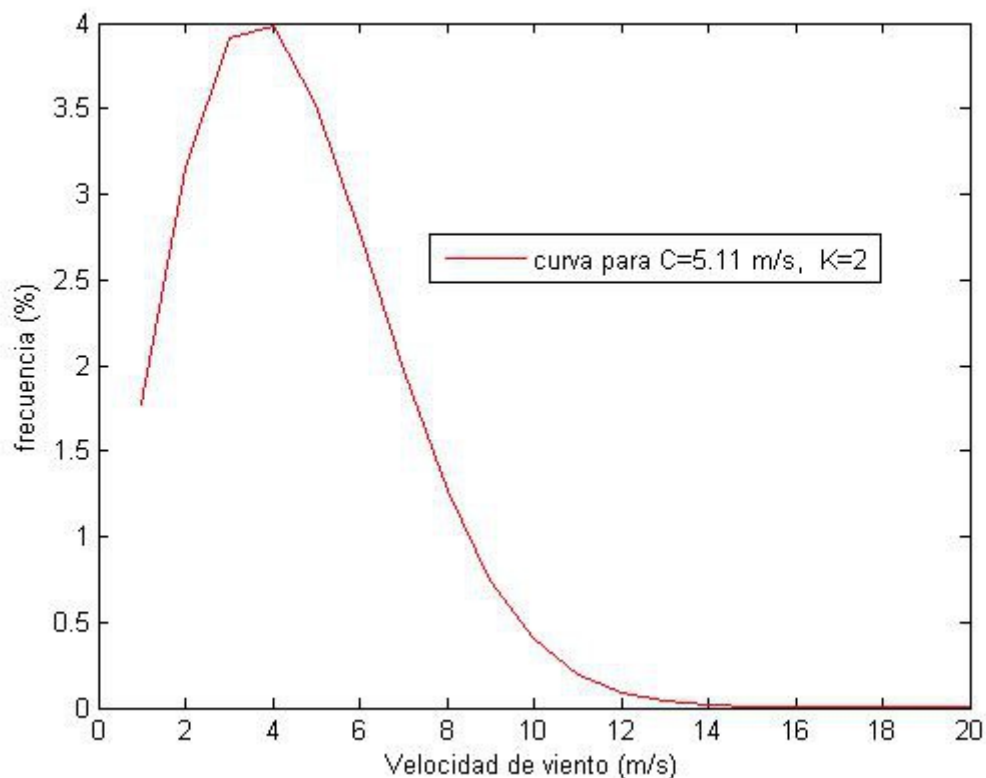


Figura 2-11. Distribución Weibull de frecuencia de velocidades de viento.

Obtenemos la gráfica con los parámetros de la distribución de Weibull, y trasladamos los datos para la obtención de la curva, listados dentro del rango de velocidades de funcionamiento del generador.

Las frecuencias de aparición expresadas en porcentaje, a partir de las cuales se ha obtenido la frecuencia de duración en horas (base 24 h), son las siguientes:

Columns 1 through 10

0.0737 0.1314 0.1628 0.1660 0.1470 0.1158 0.0821 0.0528 0.0310 0.0166

Columns 11 through 20

0.0082 0.0037 0.0015 0.0006 0.0002 0.0001 0.0000 0.0000 0.0000 0.0000

Las frecuencias de aparición expresadas en horas se reflejan en la tabla 3-4, explicándose en el siguiente apartado la forma de proceder para su obtención.

2.3.4 Métodos estadísticos para análisis de los datos de viento.

Para el análisis estadístico de datos de viento, según aparece en el libro de referencia [7], se suele emplear alguno de los métodos relacionados a continuación; aparecen por orden de menor a mayor exactitud en cuanto a la serie temporal (horaria, diaria, etc) de producción energética estimada.

1º) Modelo estático, en el que a partir de una curva de probabilidades de viento y una curva característica de potencia de la máquina, se obtiene una curva de probabilidad de ocurrencia de cada intervalo de potencia.

2º) Modelo semi estático, en el que mediante la comparación de una serie temporal de valores de viento (como dato medido en el pasado) con una curva de probabilidad de ocurrencia de cada intervalo de potencia, se obtiene una serie temporal (horaria, diaria, etc) de valores de potencia eólica.

3º) Modelo cuasi dinámico, en el que a la serie temporal de valores de potencia del modelo anterior, le añadimos datos obtenidos mediante modelos numéricos. Dichos datos simulan el comportamiento ante periodos de desorientación por cambios bruscos de viento, arranques y paradas por mantenimientos programados, etc...

Volvamos a los datos de viento obtenidos en el apartado anterior; para obtener los valores de la tabla (2-4) se redondea el porcentaje de aparición (figura 2-11 anterior) de un determinado valor de velocidad multiplicado por 24 y dividido por 100 (en base a 24 periodos); la suma de todos los valores de duración de cada evento (aparición de una determinada velocidad), vale 24 con este cambio de base.

Lo que hacemos para obtener las tablas 2-4 y 2-5, es la utilización de la curva de probabilidad de ocurrencia, según el *MODELO ESTÁTICO* descrito anteriormente.

Tabla 2-4.- Velocidades de viento y frecuencias de aparición (horas sobre 24 h)

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

Las frecuencias de aparición expresadas en porcentaje, a partir de las cuales se ha obtenido la frecuencia de duración en horas (base 24 h), son las siguientes:

Columns 1 through 10

0.0737 0.1314 0.1628 0.1660 0.1470 0.1158 0.0821 0.0528 0.0310 0.0166

Columns 11 through 20

0.0082 0.0037 0.0015 0.0006 0.0002 0.0001 0.0000 0.0000 0.0000 0.0000

Observe el lector en la tabla 2-4 que el rango de velocidades llega hasta 20 m/s. Este fondo de escala se ha elegido así por dos motivos:

1ª) La frecuencia de aparición por encima de este valor de velocidad es muy improbable como se deduce de la distribución de Weibull.

2ª) A partir de este valor la turbina eólica entra en pérdida aerodinámica, y si sigue aumentando la velocidad de viento actúa el frenado por seguridad, como puede deducirse de la curva de producción (Figura 2-12) y de los datos técnicos disponibles.

Por tanto, aunque se produzca el caso de $v > 20$ m/s, la máquina no entregará potencia.

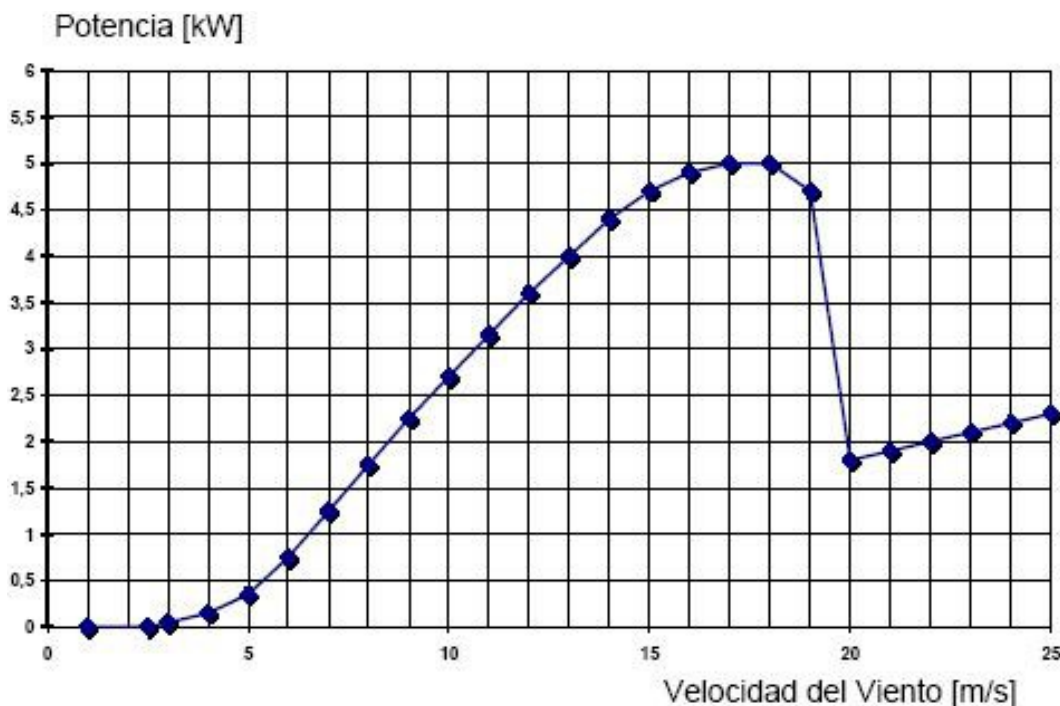


Figura 2-12. Curva de producción eólica, web fabricante www.eshia.es.

Ahora podemos establecer una equivalencia entre velocidad de viento y potencia promedio producida, lo cual se representa en la siguiente tabla, añadiendo una 3ª fila para indicar el número de horas de aparición de cada valor de producción (Tabla 2-5).

Tabla 2-5.- Equivalencia VIENTO-PRODUCCIÓN-TIEMPO

v (m/s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Kw	0	0	1	2	4	7	13	18	23	27	32	36	40	44	47	49	50
Horas	2	3	4	4	3	3	2	1	1	1	0	0	0	0	0	0	0

Finalmente, los datos disponibles de la evolución de viento diario, y los datos de distribución de frecuencias en el rango de velocidades de funcionamiento del aerogenerador, permiten obtener una SERIE TEMPORAL de los valores esperados de producción media en los 24 periodos en que se divide el horizonte temporal de estudio (evolución diaria).

En definitiva, lo que haremos es aplicar el modelo *SEMIESTÁTICO* explicado al principio de este apartado, que no hace más que añadir al resultado del modelo *ESTÁTICO* una serie temporal de valores de viento obtenida como datos del pasado.

Dichos datos deben ser lo más abundantes posible y deben haber pasado por un tratamiento estadístico propio de la obtención de valores de naturaleza estocástica (no determinista), como por ejemplo se ilustra en el artículo consultado para este trabajo, extraído de una publicación especializada [8].

En este proyecto y por sencillez, no se hará un tratamiento tan abundante de datos. Simplemente se cogerán las

gráficas de producción eólica distribuida por horas, disponibles en tiempo real en la web de Red Eléctrica Española, REE, y para varios días anteriores al momento consultado (ver Figura 1-1, apartado 1.1.3). Esta producción no está localizada a ninguna zona geográfica concreta sino que se refiere a la totalidad de la producción española (peninsular).

Por tanto, como tenemos datos de viento para un día cualquiera y los días anteriores, podemos hacer la siguiente aproximación: en una misma gráfica se superponen las curvas de producción eólica global para todos los días analizados, y también se superpone la curva de distribución de frecuencias (por horas) moviéndola hasta ajustarla lo más posible a la suma de las anteriores. Aunque este autor ha hecho dicha superposición de forma manual (figura 2-13), esto sería imposible para cualquier situación real con necesidades de manejar gran número de datos.

Este procedimiento, por tanto, es necesario que se programe mediante una simulación que obtenga la curva de ajuste por mínimas distancias entre puntos (por ejemplo), y de esta forma se obtiene la serie de valores de producción esperada de la figura 2-14. El código usado para dicha programación se incluye en el Anexo 2.

En la figuras 2-13 y 2-14 se representan por sencillez las curvas tomando sólo los datos de viento de dos días consecutivos. Las curvas procedentes de las gráficas facilitadas por REE están adimensionalizadas, para poder compararlas cualitativamente contraponiéndola con nuestra curva diaria de producción esperada, que tiene unidades en kW.

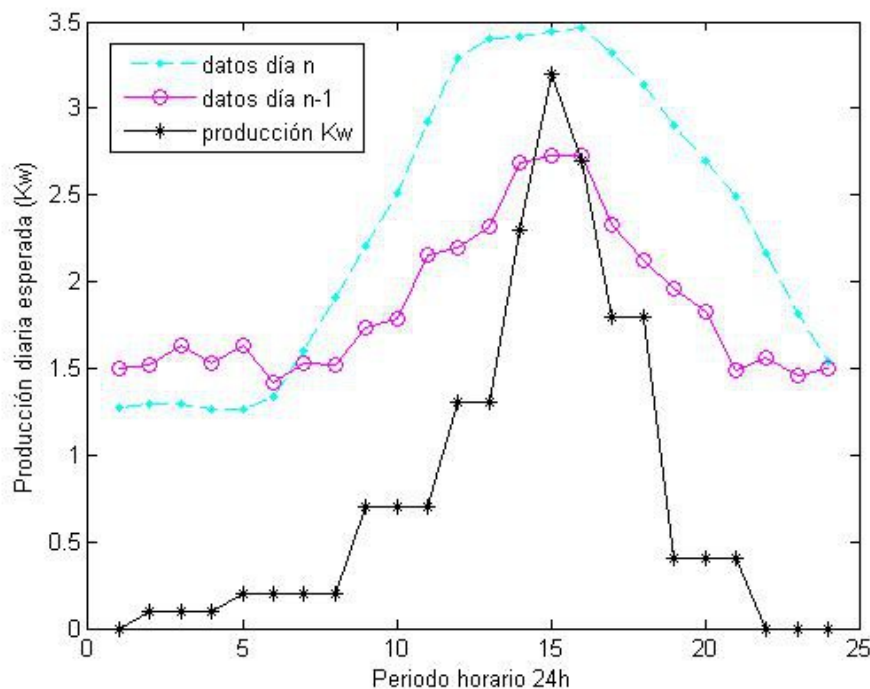


Figura 2-13. Curvas de producción eólica para un día aleatorio, fuente elaboración propia a partir de los datos obtenidos anteriormente (REE, fabricante ESHIA, estadística Weibull).

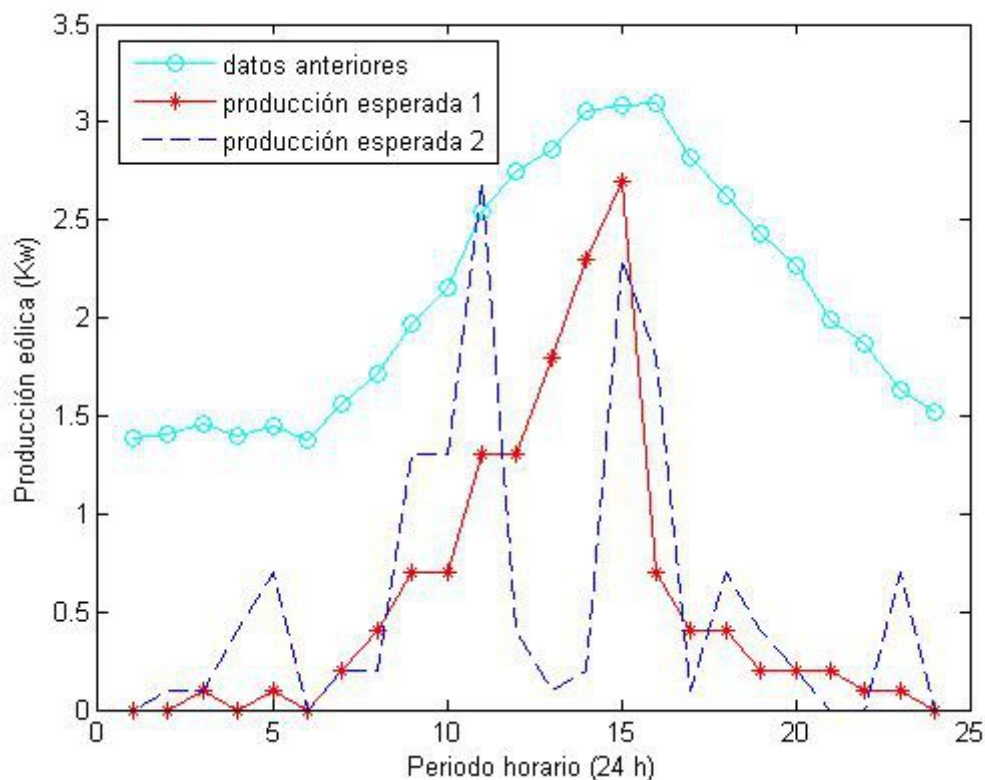


Figura 2-14. Curvas de producción eólica esperada, programadas mediante ajustes por mínimas distancias.

El valor de velocidad de viento nominal del aerogenerador es de 17 m/s, que proporciona la potencia nominal (5 Kw) y que no se alcanza en ningún momento del día (según la estadística); la distribución de potencias (valores promedio) que se alcanzan por periodos es la siguiente (tabla 2-6), según los datos anteriormente representados, en línea de trazos continuos ---*---*---(extraídos de Matlab, ver código en Anexo 2).

Columns 1 through 8	(kW)							
0	0	0.1000	0	0.1000	0	0.2000	0.4000	
Columns 9 through 16	0.7000	0.7000	1.3000	1.3000	1.8000	2.3000	2.7000	0.7000
Columns 17 through 24	0.4000	0.4000	0.2000	0.2000	0.2000	0.1000	0.1000	0

Tabla 2-6.- Distribución de potencias (producción por periodo programado -24 h-)

h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Kw	0	0	0.1	0	0.1	0	0.2	0.4	0.7	0.7	1.3	1.3	1.8	2.3	2.7	0.7	0.4	0.4	0.2	0.2	0.2	0.1	0.1	0

Dichos valores promedio serán corregidos por la desviación típica, que representa las incertidumbres que ya fueron cuantificadas anteriormente. Ahora usaremos la distribución normal (ρ, σ), que está tabulada y que por facilidad se puede obtener para cada pareja de valores mediante la herramienta de Excel:

$P90 = \text{DISTR.NORMAL.INV}((1-0.9); \rho; \sigma) \rightarrow$ Dado el valor promedio ρ y la desviación típica σ , obtenemos el valor de producción cuya probabilidad de ser superado es del 90%.

A modo de ejemplo,

\rightarrow en la hora 12 el valor promedio es 1.3 Kw y la desviación típica en las mismas unidades:

$$5\% * 1.3 = 0.063 \text{ kW,}$$

\rightarrow luego $P90$ (hora 12) = 1.22 kW

Se hace uso de la citada herramienta de Excel para todas las celdas con valor promedio, desde la hora 1 hasta la hora 24, recogiendo los resultados en la siguiente tabla: .

Tabla 2-7.- Valores corregidos de producción por horas (Fiabilidad 90%)

hora	potencia	desviación	potencia corregida
		5%	P90
1	0	0	0
2	0	0	0
3	0,1	0,005	0,09
4	0	0	0
5	0,1	0,005	0,09
6	0	0	0
7	0,2	0,01	0,19
8	0,4	0,02	0,37
9	0,7	0,035	0,66
10	0,7	0,035	0,66
11	1,3	0,065	1,22
12	1,3	0,065	1,22
13	1,8	0,09	1,68
14	2,3	0,115	2,15
15	2,7	0,135	2,53
16	0,7	0,035	0,66
17	0,4	0,02	0,37
18	0,4	0,02	0,37
19	0,2	0,01	0,19
20	0,2	0,01	0,19
21	0,2	0,01	0,19
22	0,1	0,005	0,09
23	0,1	0,005	0,09
24	0	0	0
	Kw	kW	Kw

2.3.4 Estudio particular del generador fotovoltaico.

A semejanza de cómo se ha hecho con el generador eólico, viene a continuación un análisis para determinar la producción esperada en el campo de generación solar fotovoltaica, que forma parte de la instalación real de generación distribuida estudiada en este trabajo.

Los datos conocidos del campo fotovoltaico, situado en el término municipal de Moguer (Huelva), se describen a continuación.



Figura 2-15. Campo fotovoltaico en El Arenosillo, Moguer (Huelva), fuente INTA.

Se acompaña una fotografía reciente en la figura 2-15 anterior. La planta dispone de 144 paneles con potencia pico nominal de 50 Wp y 36 celdas cada uno.

Desde el punto de vista del montaje, están colocados sobre dos estructuras metálicas paralelas con 72 paneles cada una y con una inclinación de 37° orientados hacia el sur.

Desde el punto de vista de la conexión eléctrica, los paneles se dividen en 4 subcampos, de dos series cada uno y 17 módulos paralelizados cada serie. En resumen, como es habitual en estas instalaciones se dispone de una conexión en SERIE-PARALELO con el objetivo de conseguir una determinada tensión de salida. En este caso la salida es de 400 vCC, y la tensión de conexión al BUS de CC de la microrred es de 48 vCC, por lo que se dispone de convertidores CC/CC que serán modelados posteriormente para evaluar sus pérdidas.

En cuanto a la potencia disponible a la salida, tenemos datos del INTA de una caracterización reciente del conjunto de paneles, obteniéndose una aproximación de 5.5 kW de valor nominal.

INCERTIDUMBRE POR FALLO o PARADA NO PREVISTA DE LA PLANTA PV:

Para recopilar información sobre los riesgos de fallo imprevisto en plantas fotovoltaicas se ha consultado el Trabajo Fin de Grado, referenciado en la bibliografía [6].

También se ha consultado el libro de referencia en este proyecto sobre energías Renovables [10], donde aparece que suponiendo el mejor de los materiales utilizado para la construcción de los paneles solares fotovoltaicos, la vida útil de dichos paneles se establece en 20 años de media.

En adelante, para la cuantificación de incertidumbres en el resto de los equipos por FALLO o PARADA NO PREVISTA, tomaremos una vida útil de 20 años, a falta de datos más específicos que podrían obtenerse en un análisis exhaustivo equipo por equipo.

Para ello, al igual que se hizo en el apartado 2.3.2 para el aerogenerador, usamos un criterio propio del MANTENIMIENTO PREDICTIVO como es el análisis estadístico de datos conocidos, para conocer cuándo está previsto que falle un equipo y programar su mantenimiento o sustitución. Esto se hace siempre para una fiabilidad dada, por ejemplo del 90%, que no es más que la probabilidad acumulada de NO FALLO, y que lleva asociada siempre un valor medio y una desviación típica en la DISTRIBUCIÓN NORMAL.

Por último, y para no extenderme más en este comentario, quiero añadir que se suele usar la distribución de Weibull en lugar de la Normal de Gauss para el análisis de fallo en elementos mecánicos de máquinas, por describir mejor el comportamiento en casos como fatiga, por ejemplo. Puede ampliarse más sobre mantenimiento predictivo consultando la obra de referencia [11] para este trabajo.

Por tanto, para la planta de generación fotovoltaica se establece el siguiente valor de la desviación típica en el análisis de fallo o parada imprevista (alternativamente lo podemos expresar también como TASA DE FALLO):

Tasa (dato): vida útil 20 años por equipo (1 fallo/año cada 20 equipos)

Población: 20 equipos (la inversa de la tasa)

Media: 19 equipos no fallan (1 falla)

Fiabilidad (dato): 0.90 (valor método percentil, 90% de que hay 20 equipos que no fallan)

→ $\sigma = 3.85\%$ INCERTIDUMBRE ASOCIADA A PARADA NO PREVISTA DEL EQUIPO

INCERTIDUMBRE POR PÉRDIDA DE PRODUCCIÓN DE LA PLANTA PV:

Al igual que hicimos para el aerogenerador, continuamos recopilando datos procedentes de bibliografía especializada, como por ejemplo el manual de Energías Renovables [10], para el caso de las pérdidas de rendimiento asociadas a determinadas partes de la instalación.

Los inversores son parte indispensable en una instalación de generación de energía eólica, aunque en el resumen por equipos se identifican aparte, puesto que dan servicio a otros elementos de la microrred. Se identifican aquí en cuanto a pérdidas, siguiendo la bibliografía reseñada en el párrafo anterior.

Tabla 2-8.- Pérdidas de producción asociadas a la utilización de inversores en instalaciones de generación de energía solar fotovoltaica.

BUS DE C. CONTINUA	RENDIMIENTO	PÉRDIDA DE POTENCIA
Tensión 24 v	70 %	30%
Tensión 48 v	75 %	25%
Tensión 110 v	85 %	15%

Para la cuantificación de riesgos por pérdida de producción procedemos de igual forma que para el caso del aerogenerador, pero tomando valores más pequeños para las incertidumbres en la toma de datos, ya que el comportamiento de la curva de irradiancia solar no es tan variable desde el punto de vista estadístico.

Se ha consultado para complementar la siguiente información, el libro de referencia de Energías Renovables [10], referenciado en la bibliografía.

$\sigma_1 = 2\%$ Incertidumbre por errores de medida, y por errores en la toma de datos.

$\sigma_2 = 3.85\%$ Pérdidas de producción asociadas a parada imprevista.

$\sigma_3 = 0.5\%$ Pérdidas de producción por envejecimiento.

$\sigma_4 = 1\%$ Incertidumbre por catástrofes naturales, robos, etc.

Cálculo de la desviación típica: $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2} = 4.48\%$

→ $\sigma = 4.48\%$ INCERTIDUMBRE ASOCIADA A PÉRDIDA DE PRODUCCIÓN DEL EQUIPO

Para poder aplicar el método del percentil procedemos como ya se ha hecho con el aerogenerador, pero ahora con las características propias de la instalación de generación fotovoltaica:

1º) Cálculo de la desviación típica: $\sigma = 4.48\%$ (calculado en el apartado anterior).

2º) Cálculo del valor promedio: ρ . Conocemos el valor de producción nominal del campo solar fotovoltaico, $\rho = 5.5$ kW, pero no es suficiente, ya que al igual que para el caso de un generador eólico, es necesario obtener una serie temporal de valores de producción de energía.

Para la obtención de una serie temporal de valores horarios de producción energética, necesitamos combinar o superponer tres curvas (cada una de ellas es una serie de 24 valores), dos de ellas son características del emplazamiento geográfico y la tercera es propia de los paneles instalados y de sus dimensiones y orientación):

→ Curva horaria de IRRADIANCIA (expresada en $W/m^2 - h$).

→ Curva horaria de TEMPERATURA exterior. (expresada en $^{\circ}C - h$).

→ Curva con la CARACTERIZACIÓN del panel fotovoltaico. (representa la producción del panel del panel, en función de la potencia pico y de la temperatura).

Una instalación puede describirse por su potencia pico (Wp) en lugar por los metros cuadrados totales de paneles. De esta forma, se define la unidad Wp como la potencia (en W) suministrada para una radiación de $1000 W/m^2$ a $25^{\circ}C$. Por tanto, esta última curva incluirá una corrección de la potencia pico para el caso en que la temperatura exterior es diferente a $25^{\circ}C$.

En el modelado de una instalación fotovoltaica es habitual añadir a las anteriores, una curva de eficiencia del inversor. Como ya se ha dicho, de momento no se incluye, pero sí se hará en el capítulo dedicado a la simulación mediante el software OpenDSS.

Para el caso concreto que nos ocupa, los datos necesarios se obtienen mediante series horarias de 24 valores, que proceden de la simulación ya realizada con la microgrid experimental de la ETSI, a la que se aludirá en el capítulo dedicado al control predictivo.

→ Datos de irradiancia: tomando como base una radiación de $1000 W/m^2$, tenemos los siguientes multiplicadores (desde la 1 hasta la 24 horas):

mult_1=[0 0 0 0 0 0 0.01 0.1 0.15 0.2 0.8 0.9 1.0 1.0 0.99 0.95 0.8 0.2 0.01 0 0 0 0 0];

→ Datos de temperatura ($^{\circ}C$ en la placa) para un día tipo:

temp=[15 12 12 12 10 12 15 17 21 26 30 35 40 45 50 48 40 35 30 25 20 15 15 15];

→ Datos de corrección de potencia según la temperatura:

x(temp)=[0 25 50 75]; y(multiplicador)=[1.2 1.0 0.8 0.6]; esto quiere decir, aplicando a nuestra planta, por ejemplo para $50^{\circ}C$: multiplicador 0.8 para $50^{\circ}C$.

Como no hay más datos, se hará una interpolación lineal para todos los valores de temperatura comprendidos entre $10^{\circ}C$ y $50^{\circ}C$ (nuestro rango de trabajo), obteniéndose la segunda serie de multiplicadores:

mult_2 =

Columns 1 through 12

1.080 1.104 1.104 1.104 1.120 1.104 1.080 1.064 1.032 0.992 0.960 0.920

Columns 13 through 24

0.880 0.840 0.800 0.816 0.880 0.920 0.960 1.000 1.040 1.080 1.080 1.080

Estos datos están recogidos en diferentes trabajos [9], [12] referenciados en la bibliografía. El emplazamiento geográfico a que se refieren (Sevilla) es análogo al de la planta real objeto de este proyecto (INTA, Moguer,

Huelva). La caracterización de los paneles también es análoga, según los datos facilitados por el propio INTA acerca de su campo solar fotovoltaico.

A partir de dichas series de valores, se obtiene la representación mediante las siguientes gráficas:

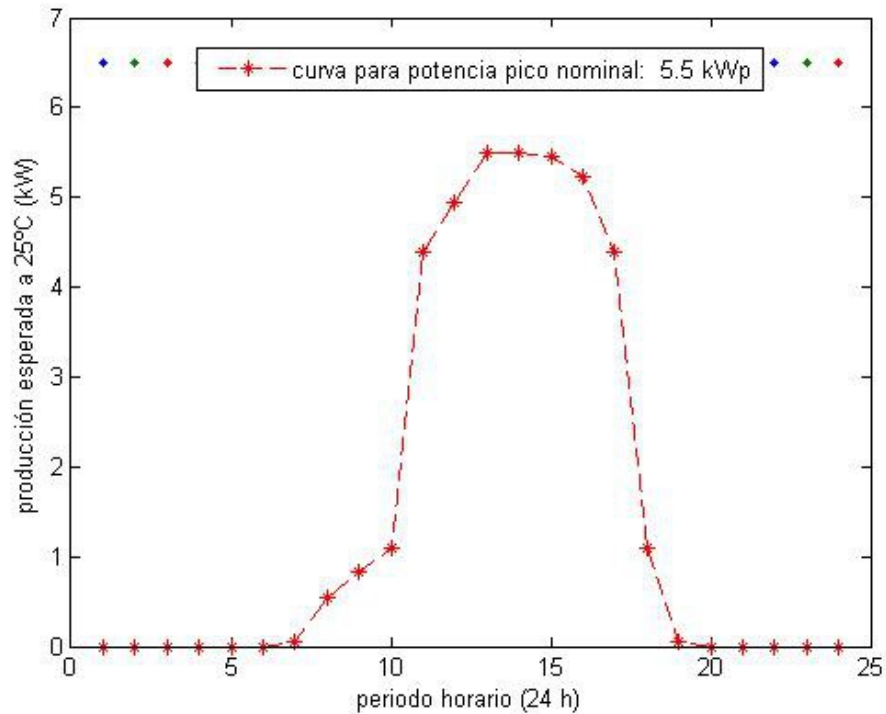


Figura 2-16. Curva de potencia fotovoltaica nominal sin corregir.

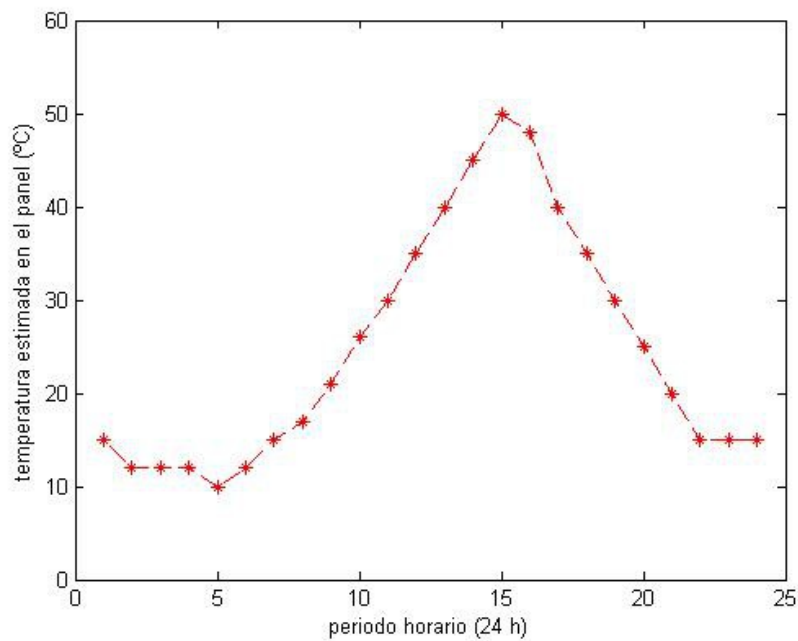


Figura 2-17. Curva de temperatura.

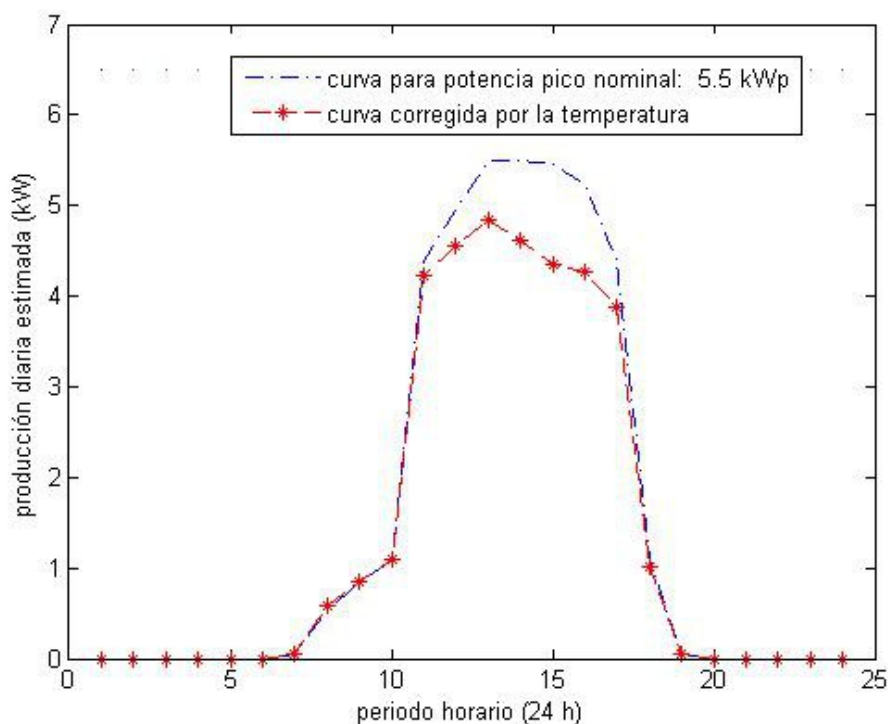


Figura 2-18. Curva de potencia fotovoltaica corregida (producción prevista).

La distribución de potencias (valores promedio) que se alcanzan por periodos es la siguiente (tabla 2-9), según los datos anteriormente representados, en línea de trazos continuos ---*-----*---

Columns 1 through 12 (kW)
 0 0 0 0 0 0 0.0594 0.5852 0.8514 1.0912 4.2240 4.5540
 Columns 13 through 24
 4.8400 4.6200 4.3560 4.2636 3.8720 1.0120 0.0528 0 0 0 0 0

Tabla 2-9.- Distribución de potencias (producción -kW- por periodo programado -24 h-)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	0	0	0	0	0.06	0.59	0.85	1.09	4.22	4.55	4.84	4.62	4.36	4.26	3.87	1.01	0.05	0	0	0	0	0

Dichos valores promedio serán corregidos por la desviación típica, que representa las incertidumbres que ya fueron cuantificadas anteriormente. Ahora usaremos la distribución normal (ρ, σ), que está tabulada y que por facilidad se puede obtener para cada pareja de valores mediante la herramienta de Excel:

$P90 = \text{DISTR.NORMAL.INV}((1-0.9); \rho; \sigma) \rightarrow$ Dado el valor promedio ρ y la desviación típica σ , obtenemos el valor de producción cuya probabilidad de ser superado es del 90%.

Se hace uso de la citada herramienta de Excel para todas las celdas con valor promedio, desde la hora 1 hasta la hora 24, recogiendo los resultados en la siguiente tabla.

Tabla 2-10.- Valores corregidos de producción por horas (Fiabilidad 90%)

hora	potencia	desviación	potencia corregida
		4,48%	P90
1	0,000	0,00	0,00
2	0,000	0,00	0,00
3	0,000	0,00	0,00
4	0,000	0,00	0,00
5	0,000	0,00	0,00
6	0,000	0,00	0,00
7	0,059	0,00	0,00
8	0,585	0,03	0,55
9	0,851	0,04	0,8
10	1,091	0,05	1,03
11	4,224	0,19	3,98
12	4,554	0,20	4,29
13	4,840	0,22	4,56
14	4,620	0,21	4,35
15	4,356	0,20	4,11
16	4,264	0,19	4,02
17	3,872	0,17	3,65
18	1,012	0,05	0,95
19	0,053	0,00	0,05
20	0,000	0,00	0,00
21	0,000	0,00	0,00
22	0,000	0,00	0,00
23	0,000	0,00	0,00
24	0,000	0,00	0,00
	kW	kW	kW

2.3.5 Cuantificación de Riesgos e Incertidumbres por equipos.

BATERÍAS ácido-Pb:

- Pérdida de producción: 15% (según el libro [10] de referencia).
- Incertidumbre por fallo imprevisto: 12.8% (tasa = 1/5 fallos/año, cálculo mediante distribución normal).

ELECTROLIZADOR:

- Pérdida de producción: 15% (según [10]).
- Incertidumbre por fallo imprevisto: 3.85% (tasa = 1/20 fallos/año, cálculo mediante distribución normal).

PILA DE COMBUSTIBLE:

- Pérdida de producción: 15% (según [10]).
- Incertidumbre por fallo imprevisto: 3.85% (tasa = 1/20 fallos/año, cálculo mediante distribución normal).

TRANSFORMADORES:

- Pérdida de producción: 2% (según [10]).
- Incertidumbre por fallo imprevisto: 3.85% (tasa = 1/20 fallos/año, cálculo mediante distribución normal).

INVERSORES:

- Pérdida de producción: 25% (según [10]).
- Incertidumbre por fallo imprevisto: 3.85% (tasa = 1/20 fallos/año, cálculo mediante distribución normal).

Veamos a continuación, mediante una tabla, el resumen para todos los equipos o puntos críticos:

Tabla 2-11.- Relación de riesgos e incertidumbres en la instalación

<u>EQUIPO o PUNTO CRÍTICO:</u>	<u>PÉRDIDA DE PRODUCCIÓN</u>	<u>TASA DE FALLO</u> (T = fallos/ año)	<u>INCERTIDUMBRE DE FALLO:</u>
Aerogenerador	5%	1/20	3.85%
Campo Solar Fotovoltaico	4.48%	1/20	3.85%
Baterías ac-Pb	15%	1/5	12.8%
Electrolizador	15%	1/20	3.85%
Pila de Combustible	15%	1/20	3.85%
Transformadores	2%	1/20	3.85%
Inversores	25%	1/20	3.85%

2.3.6 Árbol de Fallos.

La construcción del Árbol de Fallos se podría haber hecho inicialmente de forma cualitativa, pero se incluye en este apartado final ya que para que sea una herramienta efectiva es necesario contar con la mayor cantidad posible de información. Se usan además los datos del análisis cuantitativo anterior, para su integración en la jerarquía de operatividad, tal y como se ha explicado al principio de este capítulo.

(ANÁLISIS DE FALLO → ÓRDENES DE OPERACIÓN ← SEÑALES DE OPTIMIZACIÓN)

En el capítulo siguiente dedicado a la simulación de la planta, veremos otra jerarquía de operatividad que se puede resumir de la siguiente forma:

SEGURIDAD FÍSICA → SEGURIDAD OPERACIONAL → RENTABILIDAD ECONÓMICA

TIEMPO DE RESPUESTA (segundos) → TIEMPO RESPUESTA (horas) → TIEMPO RESPUESTA (días)

El árbol que representa las incertidumbres por fallo o parada del sistema, corresponde al esquema que aparece a continuación en la figura 2-19, mientras que posteriormente, en la figura 3-20 se adaptará el mismo árbol para representar esquemáticamente la secuencia de pérdidas de producción en los diferentes elementos o puntos críticos de la planta.

Los resultados cuantitativos del árbol de riesgo de fallos se integran en la tabla 2-12.

Para obtener un resultado global de la incertidumbre por FALLO en la planta, se recorre el árbol de fuera hacia adentro, y en las puertas lógicas OR se sumarán las incertidumbres en unidades de Tasa de Fallo.

ÁRBOL DE FALLO (INCERTIDUMBRES)

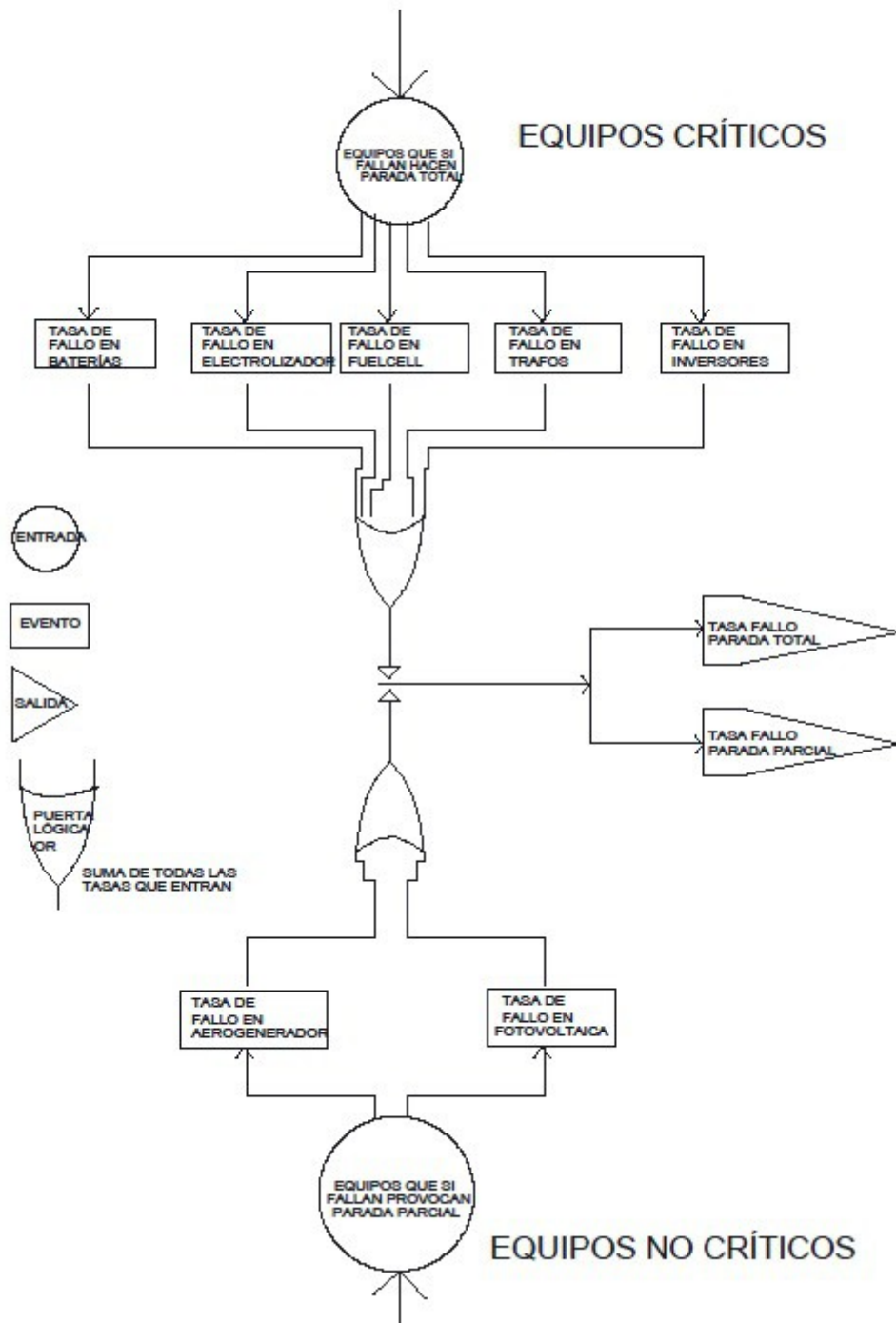


Figura 2-19. Árbol de cuantificación de incertidumbres en condiciones nominales.

Tabla 2-12.- Resumen de Cuantificación de Incertidumbres.

Equipo (pto crítico)	Baterías	Electrolizado r	FuelCel l	Trafo s	Inversores
Tasa Fallos (base 20 años)	4/20	1/20	1/20	1/20	1/20
Tasa Fallo por PARADA TOTAL imprevista	8 fallos/ 20 años $\sigma = 22\%$				
Tasa Fallo Aerogenerador	1/20				
Tasa Fallo Fotovoltaica	1/20				
Tasa Fallo por PARADA PARCIAL imprevista	2 fallos/ 20 años $\sigma = 7.5\%$				
Tasa Fallo por todo tipo de imprevisto (grave o no)	10 fallos/ 20 años = = 1 / 2				
INCERTIDUMBRE (%)	$\sigma = 25\%$				

Los resultados cuantitativos del árbol de pérdidas de producción (figura siguiente 2-20), se resumen en la tabla 2-13, recogiendo datos que han ido apareciendo a lo largo de este capítulo. Las potencias nominales de los equipos han sido facilitadas junto con el resto de características de la microrred existente en las instalaciones del INTA.

Tabla 2-13.- Resumen de Pérdidas por Operación (kW).

Equipo (pto crítico)	Batería s	Electrolizado r	FuelCel l	Trafo s	Inversores
Potencia nominal (kW)	10.5	5	1.2	10.5	6
% pérdida equipo	15%	15%	15%	2%	25%
kW pérdida equipo	1.575	0.75	0.18	0.21	1.5
Pérdida total equipos críticos (kW)	4.215				
Pérdida (kW) total equipos generadores	1.482				
Producción nominal total (kW)	10.5				
kW perdidos por equipos en operación.	5.7				

Si quisiéramos establecer con estos datos un rendimiento del proceso, obtenemos un 45.7%, que es superior al de cualquier máquina térmica quemando combustibles fósiles.

ÁRBOL DE PÉRDIDAS POR OPERACIÓN

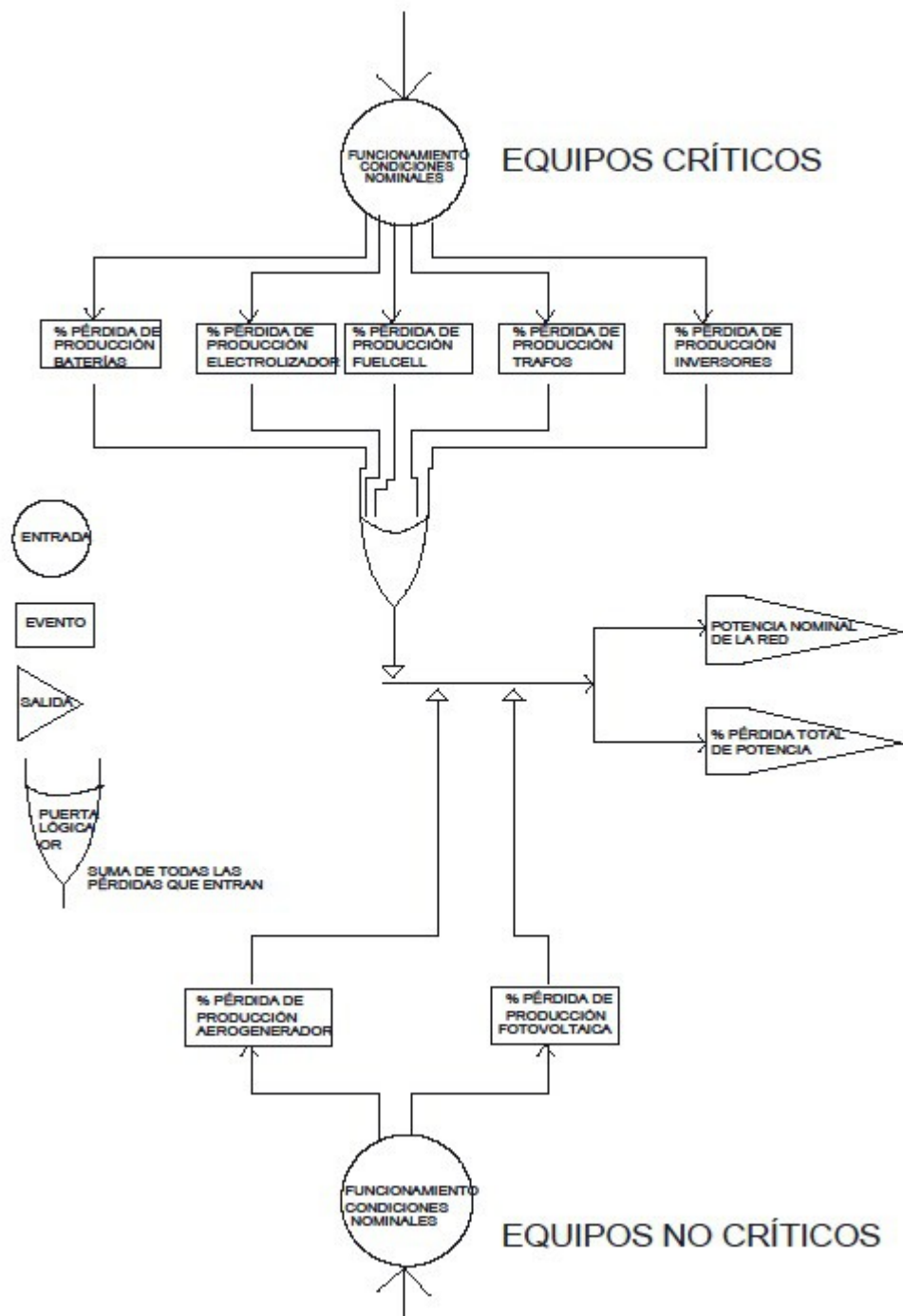


Figura 2-20. Árbol de pérdidas por operación sobre condiciones nominales.

Hemos obtenido las pérdidas por operación suponiendo **todos los equipos funcionando al mismo tiempo y en condiciones nominales**. A la pérdida de producción calculada anteriormente hay que sumar la que se produce en las líneas tanto en los buses de tensión alterna como en los de tensión continua. Estas pérdidas no tienen características probabilísticas sino que sólo dependen de las intensidades que recorran los diferentes nodos: se calcularán mediante el programa de simulación OpenDSS en el capítulo siguiente.

3 OPTIMIZACIÓN MEDIANTE CONTROL PREDICTIVO BASADO EN MODELOS

‘Antes de Kepler todos los hombres estaban ciegos.

Kepler tenía un ojo, Newton tenía los dos.’

- Henri Poincarè-

Este capítulo está dedicado a dar un repaso a las herramientas existentes y disponibles para abordar la gestión óptima mediante automatismo digital de una planta de generación distribuida y su conexión a la red. Se elige una solución usando el Control Predictivo Basado en Modelos y se presenta al final del capítulo. Dada la complejidad teórica del control predictivo, se incluye el Anexo 1 para apoyar la formulación algebraica que acompaña la solución presentada en este trabajo.

1.2 Fundamentos Teóricos

Nos referimos al control predictivo basado en modelos cuando hablamos de una estrategia de control de procesos, muy desarrollada en las últimas décadas a partir de varias familias de algoritmos, que tienen todas las siguientes características en común:

- 1). Uso explícito de un modelo que represente señales de salida en periodos de tiempo futuros (horizonte).
- 2). Cálculo de una secuencia de señales de control que minimice una función objetivo.
- 3). Uso de una estrategia de horizonte deslizante: en cada instante discreto el horizonte temporal se va desplazando hacia el futuro, y se va enviando al proceso la primera señal de control obtenida, desechándose las demás y recalculando una nueva secuencia en el paso siguiente.

Se ilustran las ideas básicas expuestas anteriormente, ayudándome de la figura 3-1, extraída del texto de referencia para toda la teoría del Control Predictivo [1].

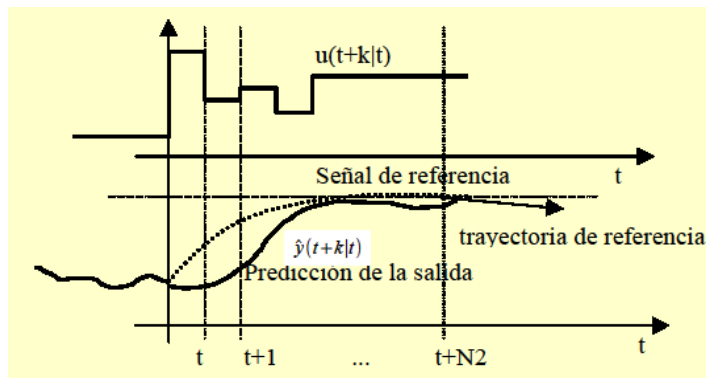


Figura 3-1. Estrategia del control predictivo, fuente Camacho & Bordons [1]

La secuencia de señales de control obtenida en cada paso es $u(t+k|t)$, es decir, señales en los instantes futuros $k=1, \dots, N_2$.

La secuencia de señales de salida predichas es:

$$\hat{y}(t+k|t)$$

Vistas las ideas fundamentales del método, hay que implementarlo para su funcionamiento en una planta (proceso) en forma de algoritmo (software). Además debe (debió en su día) competir con métodos tradicionales fiables y robustos, ya contrastados, como por ejemplo es el PID.

En la figura 3-2 extraída del mismo texto [1], podemos representar el funcionamiento descrito anteriormente, pudiendo además acotar las señales predichas mediante restricciones propias de una planta industrial (intensidades o flujos admisibles, límites de rango de actuadores, etc.), y pudiendo también definir mediante la función objetivo todas aquellas variables que conviene optimizar y que si las incluyésemos como restricciones en forma de inecuaciones pueden llevar a la no factibilidad (no existencia de solución admisible para el sistema de ecuaciones).

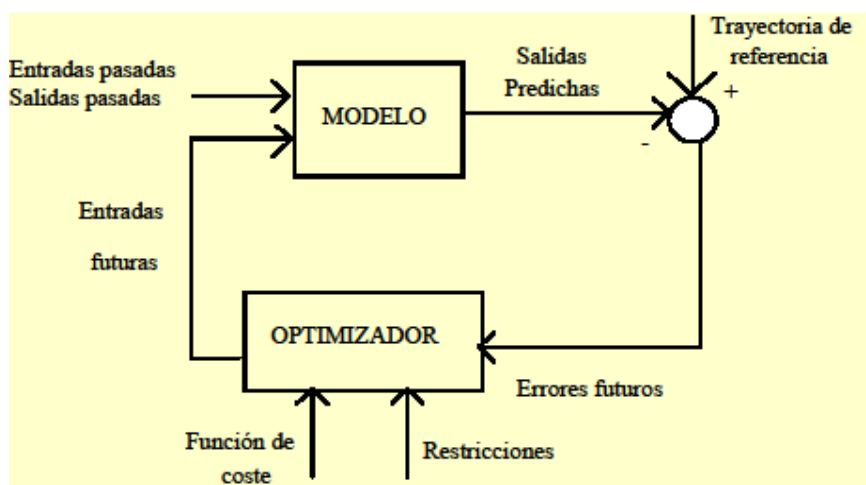


Figura 3-2. Estructura básica del control predictivo, fuente Camacho & Bordons [1]

Además de las restricciones y de los errores predichos, el algoritmo necesita para realizar la optimización (ver figura anterior) una función de coste. Por tanto, la secuencia de funcionamiento del algoritmo es la siguiente:

- 1º) → Predecir el efecto de las entradas futuras en las salidas resultantes.
- 2º) → Calcular las entradas futuras mediante un optimizador.
- 3º) → La optimización consiste en minimizar una función de coste sujeta a restricciones de operación.
- 4º) → La función de coste incluye el valor del esfuerzo de control junto con el error predicho, ya que ambos deben ser minimizados. Opcionalmente se pueden incluir otros aspectos a optimizar en dicha función.
- 5º) → Solucionar el sistema de ecuaciones resultante: la función objetivo suele ser cuadrática y las restricciones lineales. Se usará cualquier método de programación lineal o cuadrática adecuado. Se obtiene la predicción para una secuencia de instantes posteriores conocido el actual, se aplica la señal de control del instante inmediato posterior y se desechan el resto.
- 6º) → En el instante posterior se repite el proceso desde el principio.

A la vista de todo lo expuesto anteriormente, podemos deducir la importancia que tiene elegir un buen modelo de la planta, capaz de reproducir la dinámica del proceso y predecir correctamente señales futuras, así como suficientemente simple como para implementarlo fácilmente.

Para terminar esta introducción, voy a ilustrar el concepto de Control Predictivo basado en Modelos o MPC por sus sigla en inglés (“*Model Predictive Control*”), con el siguiente ejemplo comparativo; éste aparece

recogido en el texto [1] y es algo que nos resulta tan familiar como conducir un automóvil.

Tenemos dos conductores, uno llamado MPC y el otro es PID. El primero conduce el vehículo conociendo en cada instante un horizonte (trayectoria de referencia a seguir), y tomando la decisión de actuar en alguno de los controles como son volante, frenos, acelerador (hasta aquí nada que parezca extraño a quién conduce habitualmente). En cada nuevo instante, vuelve a actualizarse el horizonte y el conductor toma una nueva decisión de control (siempre la inmediata, dejando de momento en espera a las previsibles decisiones posteriores).

En el caso del segundo conductor, sus decisiones de control son tomadas en base a errores pasados. Pero, ¿esto quiere decir que conduce mirando sólo los retrovisores? No es exactamente así. El PID usa una trayectoria deseada futura (sería como conocer exactamente cada curva de la carretera, por ejemplo para quién recorre exactamente el mismo trayecto todos los días durante años). Dicha trayectoria de referencia se utiliza para actualizar el SET POINT. Por tanto no hay tanta diferencia entre los dos métodos, pero sí podemos decir que MPC se anticipa mejor. El día que haya un animal cruzando la carretera, el conductor PID no podrá esquivarlo.

2.2 Breve Historia. Algoritmos

En esta sección incluyo una relación no exhaustiva de los principales algoritmos aparecidos en las últimas décadas, clasificándolos en función del modelado necesario para la obtención del predictor, para una determinada señal de control. Para profundizar en los algoritmos presentados y algunos más que históricamente han ido apareciendo, se puede consultar el capítulo 3 del texto [1].

3.1.1 Algoritmos de primera generación.

En la década de los 70 del siglo pasado comienzan a aparecer artículos presentando los primeros algoritmos. Se trata de los primeros métodos para su aplicación en la industria, de Control Predictivo mediante un modelo dinámico de proceso.

Voy a hacer mención aquí a los dos más importantes, en un caso usando un modelo de respuesta a señal impulso y que fue llamado MAC (*'Model Algorithmic Control'*), y en otro caso del tipo *Step Response Model* donde la entrada es un escalón. Este último fue llamado DMC (*'Dynamic Matrix Control'*).

3.1.2 Algoritmos de segunda generación.

Posteriormente aparecieron los algoritmos EHAC (*'Extended Horizon Adaptive Control'*) y también EPSAC (*'Extended Prediction Self Adaptive Control'*), ambos formulados con un modelo basado en una función de transferencia entrada-salida. En ambos métodos se consigue la predicción de futuras salidas resolviendo la ecuación diofántica (también se resuelve dicha ecuación en el Anexo 1 de este trabajo, una vez modelado el caso real que nos ocupa).

3.1.3 MPC en la actualidad.

En el año 1987 y desarrollado por Clarke, se presenta el GPC (*'Generalized Predictive Control'*) fundamentado en los mismos conceptos que los dos anteriores. Con el paso del tiempo, y hasta hoy, se ha convertido en uno de los métodos más populares en la industria. El GPC será el algoritmo que se implementará en una aplicación real en este Proyecto.

Por último habría que añadir en este breve repaso histórico, aquellos métodos formulados en el contexto de espacio de estados. A pesar de necesitar un fundamento teórico más complejo, y requerir un conocimiento más profundo del proceso a modelar, en muchos casos facilita la generalización a procesos complejos multivariables, no lineales o sometidos a perturbaciones de tipo estocástico.

Un ejemplo de esta última familia de algoritmos puede ser el propuesto por Richalet y que es conocido como PFC (*'Predictive Functional Controller'*).

1.3 Definición del modelo a resolver

3.3.1 Diagrama del proceso.

Empezaré definiendo un esquema del proceso a modelar, donde aparezcan las variables que posteriormente se definirán como variables manipuladas, o bien como variables controladas. Las variables de control o manipuladas suelen representar un flujo (masa, energía, etc), y las variables controladas suelen identificarse con un estado (por ejemplo nivel de un depósito, diferencia de potencial, etc).

En cualquier proceso industrial es habitual someter a restricciones, tanto a las variables de un tipo como a las del otro. Por ejemplo, un tipo de restricción para una variable de control sería el estado de apertura de una válvula, y para una variable de estado podría ser el nivel máximo y mínimo de almacenamiento de un depósito. En la figura 3-3 se identificarán además de dichas variables, el sentido previsto de los flujos.

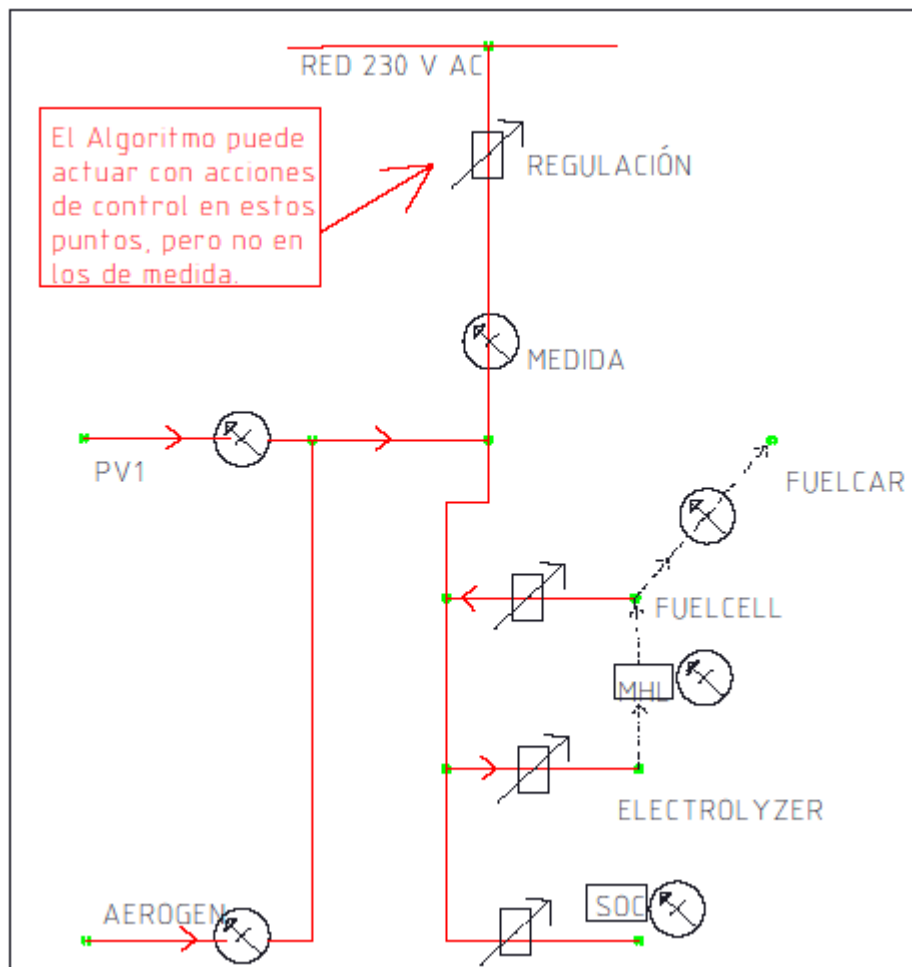


Figura 3-3. Esquema del sistema de generación distribuida a modelar.

Dichas restricciones se modelarán más adelante con el algoritmo, pero en nuestro diagrama inicial ya podemos ir identificando válvulas (interruptores, reguladores), medidores de caudal (amperímetros, watímetros), etc.

3.3.2 Diagrama del control.

El siguiente paso una vez que tenemos definido el proceso y sus variables, es obtener un modelo que nos sirva para el control. Recordando la figura 3-2 en la introducción de este capítulo dedicado al Control Predictivo, se trata de obtener un diagrama de bloques con sus correspondientes bucles cerrados y bucles abiertos, que represente de la manera más fiel posible el algoritmo de control del proceso que se quiere obtener. La fidelidad

con la que el modelo responda a la realidad a la hora de la implementación real, es algo que tendrá que ser comprobado y ajustado (modificaciones del algoritmo, sintonización y validación) para la puesta en marcha.

Se propone a continuación un Diagrama de Bloques, figura 3-4, indicando las variables implicadas según lo expuesto anteriormente.

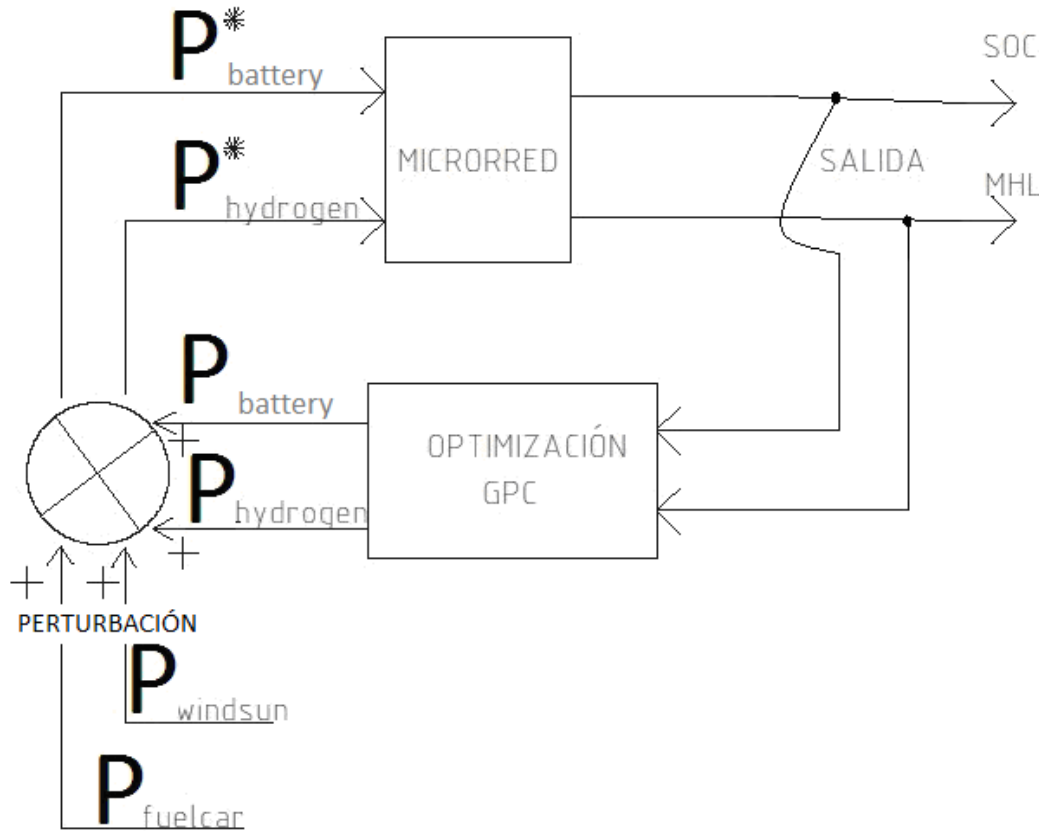


Figura 3-4. Diagrama de bloques para el algoritmo de control a modelar.

Conviene antes de seguir avanzando, repasar las variables introducidas hasta ahora, y que serán utilizadas de aquí en adelante:

VARIABLES DE CONTROL:

P_{BATTERY} , representa la potencia que entra o sale de las baterías (carga o descarga).

P_{HYDROGEN} , representa la potencia que entra o sale del proceso del hidrógeno (producción mediante electrólisis, o bien consumo del mismo en la pila de combustible para producir electricidad).

VARIABLES CONTROLADAS:

SOC, representa el nivel de carga de batería, en unidades (%).

MHL, representa el nivel de almacenamiento en forma de Hidruros Metálicos, en unidades (%). Es una forma segura, a mucha presión, de contener el hidrógeno, pudiendo recuperarlo en forma gaseosa o licuada a presión.

VARIABLES LIMITANTES:

No son ni controladas ni manipuladas (vienen dadas), pero hay que incluirlas en el modelo porque modifican la relación entre la entrada y la salida (ver cambio de variables introducido en la figura 2-4).

P_{WINDSUN} , representa la entrada en la red de potencia procedente del aerogenerador (AEROGEN) o bien del campo fotovoltaico (PV1), o de ambos simultáneamente.

P_{FUELCAR} , representa la salida de potencia de la red, en forma de hidrógeno producido que se lleva el automóvil

cuando se marcha. Se expresa en unidades (kW) gastados en producir ese hidrógeno, y no incluye por supuesto la potencia devuelta a la red por la pila de combustible mientras el vehículo esté estacionado. Tampoco se incluyen en principio las pérdidas derivadas de este proceso, aunque en el capítulo 4 se revisará el algoritmo, esta vez sí teniendo en cuenta dichas pérdidas.

Aunque no se ha dicho hasta ahora, el criterio de signos será magnitud positiva cuando la potencia entra en la red. Por tanto, la variable definida, P_{FUELCAR} tendrá en principio signo negativo, y la variable definida P_{WINDSUN} tendrá signo positivo.

3.3.3 Horizonte de predicción y horizonte de control. Modelo de predicción.

Definimos horizonte de predicción como el número de intervalos discretos de tiempo, a partir del instante actual t , para los cuales el modelo del proceso predice un valor de la salida esperada. En la figura 3-4 al principio de este capítulo, el horizonte de predicción se expresaba como N_2 .

Definimos horizonte de control N , como el número de intervalos desde el instante actual t , desde el cual consideramos que ya no hay variación de la señal de control predicha. Siempre se cumple $N < N_2$ (observar en la figura 3-2 el punto a partir del cual la gráfica de $u(t+k | t)$ permanece constante), y mientras no se diga lo contrario en este trabajo se tomará siempre $N=N_2$.

Para obtener la expresión que proporciona una predicción de la variable de salida en instantes posteriores al actual ($t+k$), necesitamos un modelo que podría ser la respuesta conocida a una entrada en escalón, o a una entrada en rampa, o bien una función de transferencia que relacione la entrada y la salida.

Ya he comentado anteriormente que en este trabajo se modelará el Control Predictivo usando un algoritmo GPC, y es este el caso en que se utiliza función de transferencia.

Para ir profundizando en la materia, voy a suponer en principio un diagrama de bloques para un proceso muy sencillo, con una sola variable de control (entrada) y una sola variable controlada (salida).

El sencillo modelo univariable (definido como *SISO*, *simple IN-simple OUT*), se puede representar de la siguiente forma, figura 3-5.

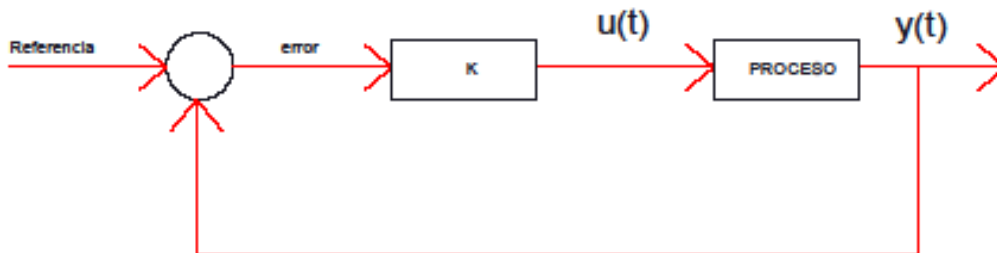


Figura 3-5. Diagrama en bucle cerrado para función de transferencia univariable.

A partir de aquí procedemos a la formulación general del algoritmo, comenzando por los siguientes puntos:

- Obtener un modelo del proceso y un modelo para las perturbaciones.
- Obtener una expresión de la mejor predicción en base a los modelos anteriores.
- Definir una función objetivo.
- Obtención de la ley de control.

El modelo del proceso permite calcular la salida predicha en instantes futuros, y para ello debe incluir un modelo de la perturbaciones que refleje el efecto de las entradas no medibles, el ruido y los errores de modelado.

Una función de transferencia entrada/salida se puede expresar como: $G = B/A$

Desarrollando la formulación:

$$A(z^{-1})y(t) = B(z^{-1})u(t) \quad (1-3)$$

siendo los polinomios de la función de transferencia:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na}$$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \dots + b_{nb} z^{-nb}$$

z^{-1} es la variable discreta que representa cada periodo contando en el tiempo hacia atrás.

I significa vector columna con todos los elementos igual a 1, para el caso univariable es un escalar.

(Tenemos por tanto una función de transferencia en variable discreta)

Como ya se ha comentado anteriormente es necesario incluir un modelo para las perturbaciones:

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})e(t) (I/\Delta) \quad (2-3)$$

siendo los polinomios A y B los mismos que para la ecuac. (1-2),

d es el retardo (número de periodos discretos en los que el sistema tarda en reaccionar al estímulo de la señal entrada).

$e(t)$ es un ruido blanco coloreado por $C(z^{-1})$ que es un polinomio a definir.

(I/Δ) es un integrador en variables discretas, siendo $\Delta = I - z^{-1}$.

Suele usarse $C(z^{-1})$ igual a la unidad con lo que el modelo de perturbación se reduce al ruido blanco multiplicado por un integrador.

La expresión (2-2) corresponde al modelo autoregresivo de media móvil integrado, por sus siglas *CARIMA* (*Controlled Auto-Regressive Integrated Moving Average*)

Veamos la expresión del predictor (expresión de la salida en el instante $t+k$ conocido el valor en el instante actual t), a partir de la ecuación (2-2) se obtiene:

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

Resolviendo la ecuación diofántica, podremos demostrar en el Anexo 1, que para las hipótesis de retardo nulo ($d=0$), y modelo *CARIMA* con ruido blanco no coloreado (polinomio $C(z^{-1}) = I$), la mejor predicción se escribe:

$$\hat{y}(t+k/t) = G \Delta u(t+k-1) + f, \quad \text{para } k=1, \dots, N \quad (3-3)$$

Siendo N el horizonte de control.

Y siendo además:

G una matriz cuadrada de dimensiones $N \times N$, triangular inferior, que depende sólo del valor de los polinomios A y B (del modelo), y es obtenida al resolver la ecuación diofántica.

f es un vector ($N \times 1$) que es suma de varios vectores ($N \times 1$) formados por coeficientes también extraídos de la resolución de A y B en la ecuación diofántica, multiplicados por los valores de señales de control pasadas (conocidas) y también por los de salidas pasadas (conocidas).

En definitiva, la expresión obtenida en (3-2) responde al esquema básico de control anteriormente representado en la figura 3-2, y que volvemos a repasar:

El modelo trabaja con la información de:

- Señales de control futuras: $\Delta u(t+k-1)$, para $k=1, \dots, N$
- Señales de control pasadas: por ejemplo $\Delta u(t-1)$, cuyo valor está contenido en el vector f .
- Valores de salida actual o anteriores: por ejemplo $y(t), y(t-1)$, contenidos en el vector f .

El modelo da como resultado la salida predicha:

$$\text{para } k=1, \dots, N \quad \hat{y}(t+k/t)$$

3.3.4 Obtención de la Ley de Control.

Antes de seguir avanzando en la estrategia del control predictivo (y particularmente en el algoritmo GPC '*Generalized Predictive Control*'), vamos a ilustrar el significado de la ecuación (3-2) con la ayuda de otra ilustración extraída del texto [1] en la figura 3-6.

El diagrama de dicha figura servirá para enlazar con la explicación de cómo obtiene el algoritmo la Ley de Control a partir de la minimización de una función objetivo J que tiene en cuenta fundamentalmente cuánto nos alejamos de la referencia (w , nuevo concepto que aparece a continuación).

La respuesta libre es el vector f de la ecuación (3-2).

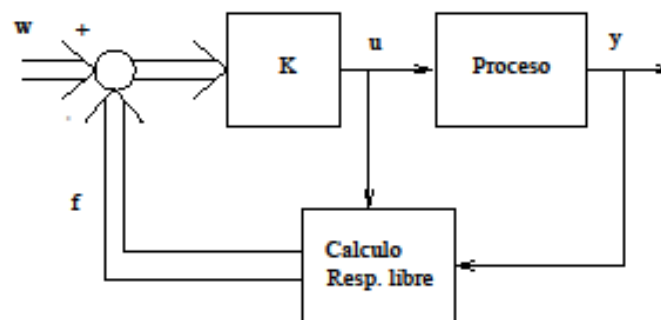


Figura 3-6. Ley de Control, fuente Camacho & Bordons.

Como podemos ver, la salida es una respuesta que se separa en dos partes, una respuesta libre y una respuesta forzada.

La respuesta libre corresponde a la evolución del proceso debido a su estado actual, incluyendo el efecto debido a las acciones pasadas como se ha visto en la formulación anterior (el vector f incluye para su construcción las acciones de control pasadas $\Delta u(t-1)$, el estado actual $y(t)$ y los estados anteriores $y(t-1)$ ya conocidos).

La respuesta forzada es la debida a acciones de control futuras, y se expresa mediante:

$$G \Delta u(t+k-1), \quad \text{para } k=1, \dots, N$$

Para calcular explícitamente la ley de control $u(t+k-1)$ se toma para cada instante t el valor de salida predicha $y(t+k|t)$, y se sustituye en la función de coste J . Se obtiene una función cuadrática en u , que al derivar para minimizar conduce para el caso sin restricciones añadidas, a una expresión analítica (explícita) de $u(t+k-1)$.

En este punto hay que hacer las siguientes consideraciones:

→ u para el caso del algoritmo GPC es la variable incremental $\Delta u(t+k-1)$ que contiene la diferencia entre la señal de control predicha para un instante y la predicha para el anterior, durante los futuros instantes $k=1, \dots, N$.

→ una vez conocido u nos quedamos sólo con el valor predicho en t y deseamos el resto $t+1, t+2, \dots, t+N$; sustituimos $u(t)$ en (3-2) obteniéndose así el valor de salida en $y(t+1)$. Avanzamos t al siguiente periodo discreto y vuelta a empezar el algoritmo.

→ La función objetivo es una función cuadrática que mide por un lado la distancia entre la salida predicha y la trayectoria de referencia (consigna o set-point) hasta el horizonte de predicción, y por otro el esfuerzo de control necesario para obtener dicha salida.

Es la expresión siguiente:

$$J = \delta \sum_{j=1}^N [\hat{y}(t+j|t) - w(t+j)]^2 + \lambda \sum_{j=1}^N [\Delta u(t+j-1)]^2 \quad (4-3)$$

donde δ, λ son coeficientes de ponderación, que en principio y mientras no se diga lo contrario en este trabajo serán considerados iguales a 1.

A continuación se sustituye la expresión de la salida predicha por

$$Gu + f,$$

y prescindiendo de los índices tenemos:

$$J = [Gu + f - w]^T [Gu + f - w] + u^T u$$

función a minimizar igualando su gradiente a cero:

$$\nabla J = 0 \quad \Leftrightarrow$$

(despejando u)

$$u = (G^T G + I)^{-1} G^T (w - f) \quad (5-3)$$

Hemos obtenido, en función del modelo de proceso usado, de las salidas conocidas pasadas, y de las señales de control también pasadas, la expresión explícita de u . Ésta contiene el valor de la primera señal de control que hay que enviar a la planta, desechándose el resto de valores contenidos en u .

Ahora analicemos el problema con restricciones en el caso de que las hubiere, donde ya no se obtiene una expresión analítica y hay que recurrir a métodos numéricos. Aunque se resolverá más adelante, se presenta aquí la formulación:

$$\begin{aligned} \text{Min } J &= [Gu + f - w]^T [Gu + f - w] + u^T u \\ \text{s.a.} & \end{aligned} \quad (6-3)$$

$$R u \leq c$$

Las restricciones recogidas en el sistema lineal de ecuaciones $R \mathbf{u} \leq c$, se agruparán generalmente en alguno de los siguientes tipos:

→ restricciones de amplitud en la señal de control:

$$U^{\text{INF}} \leq u(t) \leq U^{\text{SUP}}$$

→ restricciones de velocidad de cambio en la señal de control:

$$u^{\text{INF}} \leq u(t) - u(t-1) \leq u^{\text{SUP}}$$

→ restricciones de amplitud en la salida:

$$y^{\text{INF}} \leq y(t) \leq y^{\text{SUP}}$$

Cuando estudiemos el caso multivariable veremos cómo transformar las restricciones a la forma matricial,

$$R \mathbf{u} \leq c.$$

En cualquier caso, el problema en forma condensada siempre tendrá la forma que aparece en (6-2), y es importante destacar lo siguiente:

→ el sistema de ecuaciones que forman las restricciones es lineal.

→ la función objetivo es cuadrática.

→ por tanto, el problema de optimización no se puede resolver por programación lineal.

→ Sí que se puede resolver por programación cuadrática.

→ en este trabajo se usará la herramienta ‘quadprog’ incluida en Matlab®.

La definición en Matlab del problema de programación cuadrática es la siguiente:

$$x = \text{quadprog}(H, g, A, c)$$

que resuelve el siguiente problema de programación cuadrática:

$$\text{Min } 0.5 \mathbf{x}^T H \mathbf{x} + \mathbf{g}^T \mathbf{x}$$

$$\text{s.a: } A \mathbf{x} \leq c$$

con lo cual habrá que identificar las correspondientes matrices y vectores H , g , A , c , con los definidos en esta memoria para escribir el mismo problema --ver ecuación (6-2)--. En el desarrollo del algoritmo programado en Matlab que se acompaña en el Anexo 2, podrá verse dicha correspondencia de matrices. A partir de las expresiones (4-2) y (5-2) pueden deducirse:

$$J = 0.5 \mathbf{u}^T H \mathbf{u} + \mathbf{b} \mathbf{u} + f_0$$

$$\text{con } H = 2 (G^T G + \lambda I);$$

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

$$f_0 = (\mathbf{f}-\mathbf{w})^T (\mathbf{f}-\mathbf{w});$$

este último término podemos expresarlo también, a partir de (5-2):

$$f_0 = (\mathbf{f}-\mathbf{w})^T [G^T]^{-1} (G^T G + \lambda I) \mathbf{u};$$

con lo cual podemos agrupar expresando como un vector fila multiplicando la incógnita \mathbf{u} :

$$\mathbf{b} \mathbf{u} + f_0 = (\mathbf{f}-\mathbf{w})^T [2G + [G^T]^{-1} (G^T G + \lambda I)] \mathbf{u} = \mathbf{g}^T \mathbf{u};$$

como a la orden ‘quadprog’ de Matlab hay que darle un vector columna, finalmente:

$$\mathbf{g} = \{ (\mathbf{f}-\mathbf{w})^T [2G + [G^T]^{-1} (G^T G + \lambda I)] \}^T \quad (7-3)$$

1.4 Implementación real

Para la construcción y ajuste del algoritmo de Control Predictivo (GPC) que optimiza el funcionamiento del sistema, asumimos que estamos en un caso multivariable (*MIMO: multi IN – multi OUT*) como ya se ha indicado en el diagrama de bloques representado en la figura 2-5 incluida anteriormente en este capítulo.

La implementación del algoritmo GPC se realiza mediante una simulación real con ayuda de programas informáticos, y se incluirá más adelante, en el capítulo 4 y también en los anexos 1 y 2.

Antes de ello conviene analizar todos los datos posibles de la instalación que se va a modelar y simular; además, en el capítulo 2 se ha realizado un estudio de riesgos y fallos que servirá también para establecer los límites que podemos poner a la operación de la planta, por ejemplo en las restricciones del control predictivo.

3.4.1 Hipótesis utilizadas.

A la hora de implementar el algoritmo de control predictivo (GPC) es fundamental tener un buen modelo y una función de transferencia que represente bien el sistema real. Según he podido ver en el texto de referencia [1], en general una buena forma de conseguir el modelo consiste en estudiar la curva real de reacción del proceso, obteniéndose una matriz de transferencia en continuo (caso multivariable), para posteriormente convertir dicho modelo continuo al equivalente modelo discreto con el que funciona el control Predictivo.

El modelo utilizado para las perturbaciones es CARIMA con la particularidad de que en el caso multivariable el polinomio $C(z^{-1})$ es un matriz polinomial (2x2), para dos entradas y dos salidas.

El modelo está afectado por perturbaciones (medibles y no medibles). La entrada de datos podría incluirse dentro de las perturbaciones medibles del modelo, pero se introduce junto con las no medibles y junto a la incertidumbre en errores de modelado. A decir verdad, en la entrada de datos también manejamos un importante nivel de incertidumbre, como se ha podido ver ampliamente en el capítulo anterior dedicado al análisis de riesgos y operatividad.

El modelo de partida con el que empiezo a trabajar, antes de formular la entrada de perturbaciones medibles y no medibles, responde a una planta experimental existente en la Escuela Superior de Ingenieros de la Universidad de Sevilla. Los coeficientes de la función de transferencia están dimensionados en función de la capacidad de almacenamiento de batería ácido-Pb (SOC) por un lado, y de la capacidad del depósito de Hidruros Metálicos (MHL) por otro.

Dichas capacidades son las siguientes:

→ Batería de acumuladores: 367 A*h conectada a 48 V (equivale a 17,6 kW*h)

→ Depósito MHL: 7 Nm³ presurizado a 5 bar (equivale a dimensionarlo como 35 Nm³*bar)

El modelo puede ser extrapolado a cualquier otra planta con capacidades de almacenamiento diferentes, simplemente con multiplicar los coeficientes por el factor de escala correspondiente.

Las capacidades para la planta (también real), que se pretende modelar, son las siguientes:

→ Batería de acumuladores: 36 kW*h

→ Depósito MHL: 1 Nm³ a 15 bar

Los coeficientes correspondientes al factor de escala son los siguientes:

→ Batería de acumuladores: $K1 = 36 / 17.6 = 2.05$

→ Depósito MHL: $K2 = 15 / 35 = 0.43$

En la manipulación del modelo que se hace en el Anexo 1 se incluirán dichos coeficientes.

El sistema de ecuaciones lineales resultante de las restricciones junto con la función objetivo cuadrática a minimizar, se resuelve mediante programación cuadrática con la herramienta correspondiente incluida en Matlab®.

Se asume en este trabajo que la función objetivo cuadrática es del tipo 'Liapunov', ver figura 3-7, lo que asegura que el problema de minimización es acotado, y además proporciona estabilidad en bucle cerrado para el algoritmo de control, según el libro de referencia [1].

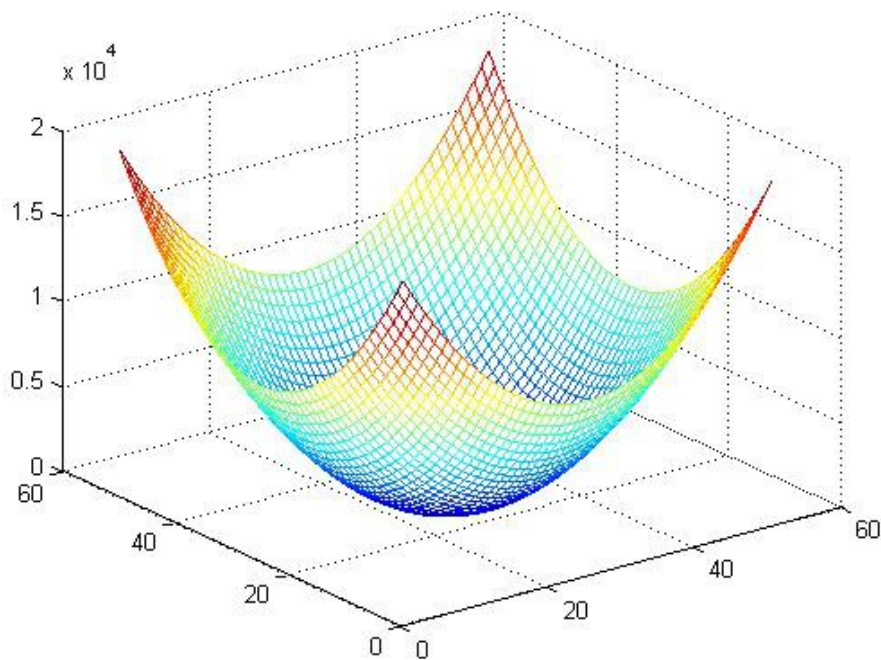


Figura 3-7. Representación de la función de Liapunov.

El horizonte de predicción, que como ya se avanzaba en el capítulo 1 coincide con el horizonte de control, se toma $N=2$ ($k=1,2$) por sencillez para la formulación que sigue a continuación.

A la hora de programar e implementar el algoritmo con ayuda de Matlab, es conveniente aumentar el valor del horizonte de predicción para conseguir mayor robustez y estabilidad. En ese sentido, los códigos de programación con el algoritmo GPC que se anexan al final de este trabajo están contruidos para un horizonte de control $N=3$.

Finalmente, al implementar el algoritmo para su resolución mediante métodos numéricos, se le introduce el dato de potencias, $P_{WINDSUN}$, $P_{FUELCAR}$, en forma de vector de valores conocidos para un periodo de 24 horas.

En páginas pasadas dentro de este mismo trabajo, se justificaban las series de valores elegidos para dichos vectores (capítulo 2), y su generalización para cualquier situación real en la que pudiéramos implementar el algoritmo, mediante estimaciones a partir de un análisis de tipo estocástico (no determinista).

3.4.2 Implementación del modelo sin pérdidas en la red.

Para la construcción y ajuste del algoritmo de Control Predictivo (GPC) que optimiza el funcionamiento del sistema, asumimos que estamos en un caso multivariable (*MIMO: multi IN – multi OUT*) como ya se ha indicado en la descripción anterior del modelo GPC.

La implementación del algoritmo de control predictivo se realiza mediante una simulación real con ayuda de programas informáticos, cuyos códigos de programación se desarrollarán en el anexo 2.

En el capítulo 2 se realizó un estudio de riesgos y fallos que ahora sirve también para establecer los límites que podemos poner a la operación de la planta, por ejemplo en las restricciones del control predictivo.

Como ya se comentaba en el apartado anterior, el modelo de la planta puede obtenerse experimentalmente, o modificando uno que ya ha sido experimentado y validado, y este es el caso del que presento a continuación.

El modelo del sistema tiene que expresar las variables de salida en función de las de entrada, y además tiene que tener en cuenta las perturbaciones como ya se explicó en la subsección 3.3.2. Usaremos el mismo modelo CARIMA en el caso multivariable, y el desarrollo teórico es análogo al caso univariable.

Se parte de un modelo explícito de una planta real, existente en los laboratorios de la Escuela Superior de Ingenieros de la Universidad de Sevilla. Dicha planta ha sido experimentada y validada, y sus resultados publicados en diversas revistas especializadas algunas de las cuales se citan en la bibliografía de referencia de este proyecto, por ejemplo [2]. Se muestra en la figura 3-8.

En el Anexo 1 se formula con detalle el proceso que nos lleva mediante los siguientes pasos a la obtención de un algoritmo de Control Predictivo (GPC).

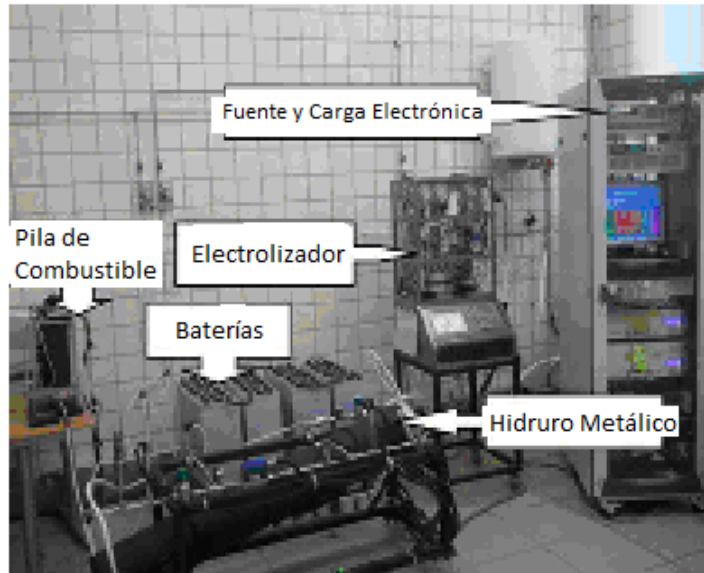


Figura 3-8. Planta experimental ETSI, propiedad de la Universidad de Sevilla.

MODELO DE PARTIDA:

Llamaré Modelo ETSI al siguiente, que relaciona entrada y salida, obtenido como ya se ha mencionado anteriormente de la publicación [2]:

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + \begin{bmatrix} 8.136 & 5.958 \\ -15.289 & 0 \end{bmatrix} \begin{bmatrix} P_{H_2}(k) \\ P_{GRID}(k) \end{bmatrix} + \begin{bmatrix} 5.958 \\ 0 \end{bmatrix} P_{NET} \quad (8-3)$$

Es indiferente usar para las magnitudes de las variables, unidades de potencia o unidades de intensidad, pues están relacionadas por una constante: TENSIÓN 400V en el BUS de corriente continua DC. Para pasar a la siguiente expresión del MODELO, se han hecho algunos cambios de variables incluyendo el que permite escribir las ecuaciones en variables de intensidad (Amperios).

Al final de todo el desarrollo se volverá a expresar la formulación en términos de potencias de entrada, con lo cual trabajaremos en las mismas unidades que en el citado modelo de referencia existente en la ETSI.

Hay que señalar que en la figura 3-4 diagrama de bloques, se usa $P_{BATTERY}$ junto con $P_{HYDROGEN}$ como variables de entrada. En el desarrollo que sigue se sustituye la primera por $P_{CIRCUIT}$, como se puede ver en la figura 3-9. En cualquier caso, y como se verá de forma práctica al implementar el algoritmo de simulación, los valores de $P_{CIRCUIT}$ y $P_{BATTERY}$ están siempre acoplados, por lo que se usarán indistintamente como variable manipulada. Las relaciones entre las variables se explican en el Anexo 1. Ya en dicho Anexo se manipulan variables en forma de Intensidades. En cuanto a las variables de salida Δy_1 , Δy_2 , se refieren a las variables de estado SOC y MHL respectivamente.

Modelo ETSI-INTA (MODIFICADO POR LA ENTRADA DE DATOS):

$$\begin{bmatrix} \Delta y_1(k+1) \\ \Delta y_2(k+1) \end{bmatrix} = \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{hydrogen}(k) \\ I_{circuit}(k) \end{bmatrix} + \begin{bmatrix} 0 & 4.885 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{fuelcar} \\ I_{windsun} \end{bmatrix} \quad (9-3)$$

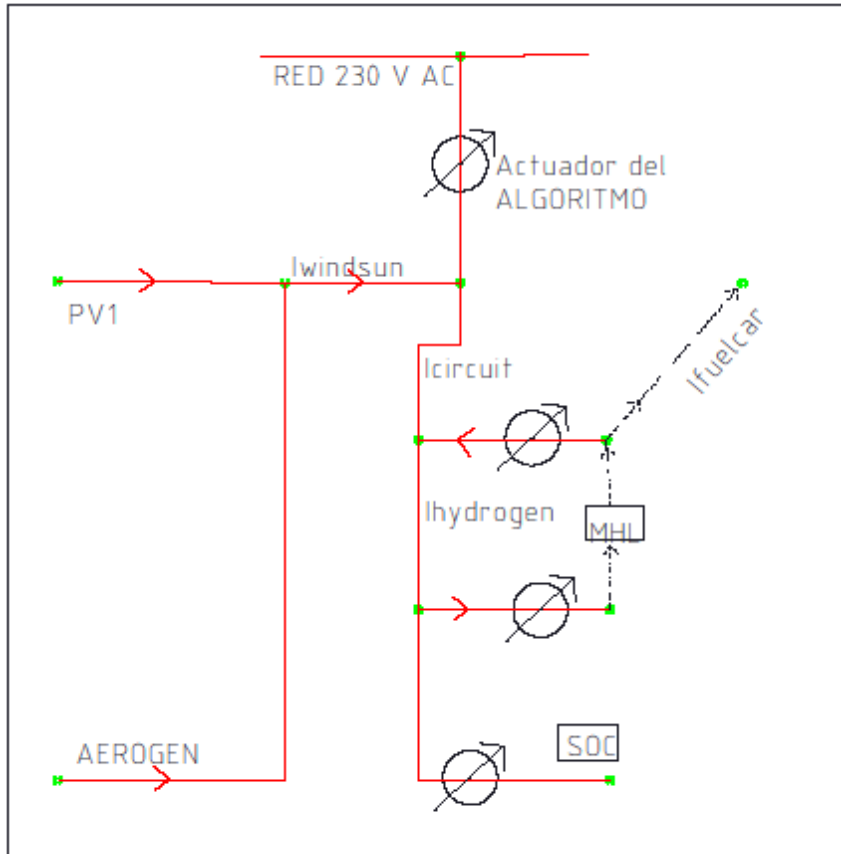


Figura 3-9. Diagrama de Flujos. Variables usadas en el modelo.

MODELO INCLUIDAS PERTURBACIONES (CARIMA):

$$\Delta y(t+k+1) = \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} u(t+k) + \frac{C(z^{-1})}{\Delta} e(t+k)$$

donde

$$u(t+k) = \begin{bmatrix} I_{hydrogen} \\ I_{circuit} \end{bmatrix}$$

y donde el polinomio de perturbación incluye la incertidumbre en entrada de datos

$$C(z^{-1}) = \begin{bmatrix} 1 + 4.885 I_{windsun} z^0 & -4.885 I_{windsun} z^{-1} \\ -1 - 2.630 I_{fuelcar} z^0 & -2.630 I_{fuelcar} z^{-1} \end{bmatrix} \quad (10-3)$$

EXPRESIÓN DEL MEJOR PREDICTOR:

Al igual que para el caso univariable se obtiene una solución para la salida (no es la única pero sí la mejor predicción), resolviendo la ecuación diofántica esta vez para matrices polinomiales, con horizonte de predicción es $N=2$. Este predictor ya tiene unidades en kW (potencias) para las variables de entrada.

$$\begin{bmatrix} \hat{y}_1(t+1) \\ \hat{y}_2(t+1) \\ \hat{y}_1(t+2) \\ \hat{y}_2(t+2) \end{bmatrix} = \begin{bmatrix} \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & 0 & 0 \\ \frac{6.575}{\Omega_2} & 0 & 0 & 0 \\ \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} \\ \frac{6.575}{\Omega_2} & 0 & \frac{6.575}{\Omega_2} & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \\ \Delta u_1(t+1) \\ \Delta u_2(t+1) \end{bmatrix} +$$

$$+ \begin{bmatrix} \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & 0 & 0 \\ \frac{6.575}{\Omega_2} & 0 & 0 & 0 \\ \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & 0 & 0 \\ \frac{6.575}{\Omega_2} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \\ \Delta u_1(t-2) \\ \Delta u_2(t-2) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \quad (11-3)$$

Expresión de la forma: $\hat{y} = Gu + f$, donde se han incluido los parámetros: (12-3)

$$\Omega_1 = (1 + 12.213 P_{\text{windmax}})$$

$$\Omega_2 = -(1 + 6.575 P_{\text{fuelcar}})$$

Se ha resuelto de forma recursiva para $k=1,2$; la expresión obtenida responde a la forma genérica ya comentada en la subsección 3.3.3, ecuación (3-3), de forma que f sólo dependerá de salidas y entradas pasadas.

DEFINICIÓN DE LAS RESTRICCIONES:

→ restricciones de amplitud en la señal de control:

$$U^{\text{INF}} \leq u(t) \leq U^{\text{SUP}}$$

→ restricciones de velocidad de cambio en la señal de control:

$$u^{\text{INF}} \leq u(t) - u(t-1) \leq u^{\text{SUP}}$$

→ restricciones de amplitud en la salida:

$$y^{\text{INF}} \leq y(t) \leq y^{\text{SUP}}$$

Para la planta que se modela en este proyecto, se toman los siguientes valores para las restricciones escritos en notación matricial (como ya se ha dicho, la señal \mathbf{u} y su variación $\Delta\mathbf{u}$, ya tienen unidades en kW):

$$U^{\text{INF}} = \begin{bmatrix} -3 \\ -7.25 \end{bmatrix}; (A) \quad U^{\text{SUP}} = \begin{bmatrix} 3 \\ 7.25 \end{bmatrix}; (A)$$

$$u^{\text{INF}} = \begin{bmatrix} -1.5 \\ -5 \end{bmatrix}; (A) \quad u^{\text{SUP}} = \begin{bmatrix} 1.5 \\ 5 \end{bmatrix}; (A)$$

$$y^{\text{INF}} = \begin{bmatrix} 40 \\ 10 \end{bmatrix}; (\%) \quad y^{\text{SUP}} = \begin{bmatrix} 90 \\ 90 \end{bmatrix}; (\%)$$

DEFINICIÓN DE TRAYECTORIAS DE REFERENCIA:

Son las consignas para las variables de salida. En este trabajo se definirán siempre como una constante, pero podría introducirse una expresión en función de t , que obligaría mediante su inclusión en la función objetivo a que la salida $y(t)$ siguiera en cada instante a la citada referencia $w(t)$.

$$w(t) = \begin{bmatrix} 50 \\ 50 \end{bmatrix}; (\%)$$

FUNCIÓN OBJETIVO Y FORMULACIÓN DEL PROBLEMA:

Matricialmente, el problema a resolver ya se expresó en la subsección 3.3.4, ecuación (6-3), en la forma:

$$\begin{aligned} \text{Min } J &= [G\mathbf{u} + f - w]^T [G\mathbf{u} + f - w] + \mathbf{u}^T \mathbf{u} \\ &\text{s.a.} \\ &R\mathbf{u} \leq c \end{aligned} \quad (13-3)$$

Ya conocemos la matriz G y la referencia w ; el vector f se actualiza en cada paso; el vector \mathbf{u} es nuestra incógnita.

Para expresar las restricciones, usamos la forma descrita en el texto de Camacho & Bordons [1] que permite construir la matriz R y el vector c del sistema lineal de ecuaciones:

$$\text{para } k=1,2.$$

$$R = \begin{bmatrix} I_{4 \times 4} \\ -I_{4 \times 4} \\ T_{4 \times 4} \\ -T_{4 \times 4} \\ G \\ -G \end{bmatrix}; \quad c = \begin{bmatrix} u^{SUP} \\ -u^{INF} \\ U^{SUP} - u(t-1) \\ -U^{INF} + u(t-1) \\ y^{SUP} - f \\ -y^{INF} + f \end{bmatrix};$$

Como novedad en la formulación desarrollada hasta ahora, aparece la matriz T, que es una matriz triangular inferior por bloques cuyos elementos no nulos son matrices identidad 2x2 .

PROBLEMA DE OPTIMIZACIÓN CUADRÁTICA A RESOLVER:

Mediante la implementación en el programa Matlab del algoritmo descrito obtenemos en cada momento una solución de $u(t+1)$, dados los valores conocidos anteriores de $y(t)$, $u(t)$, y también dada una trayectoria de referencia para la función objetivo $J[w(t)]$, y dadas unas restricciones.

El periodo de muestreo inicialmente para observar la evolución del sistema de generación distribuida, se establece en 24 horas. La entrada de datos procedente de los generadores de energía renovable por un lado, y del consumo de hidrógeno para automóviles híbridos por otro, se actualiza en cada paso discreto, establecido en 1 hora.

La simulación por tanto, arranca con una serie de 24 valores conocidos para el aporte de energía de origen renovable, y otra serie de 24 datos del consumo de hidrógeno a lo largo del periodo estudiado.

En el capítulo dedicado al análisis de operatividad, se hizo un estudio completo de las características estocásticas de la energía procedente de los citados generadores renovables, con lo que se justificó la serie de 24 valores de entrada previstos.

También se justificará en el capítulo 4, la serie de 24 valores como dato de consumo de hidrógeno, por parte de un automóvil que está repostando energía.

A continuación se presentan los resultados mediante diferentes gráficas y se discuten; en el Anexo 2 se desarrollan con detalle los diferentes códigos que permiten la implementación del algoritmo.

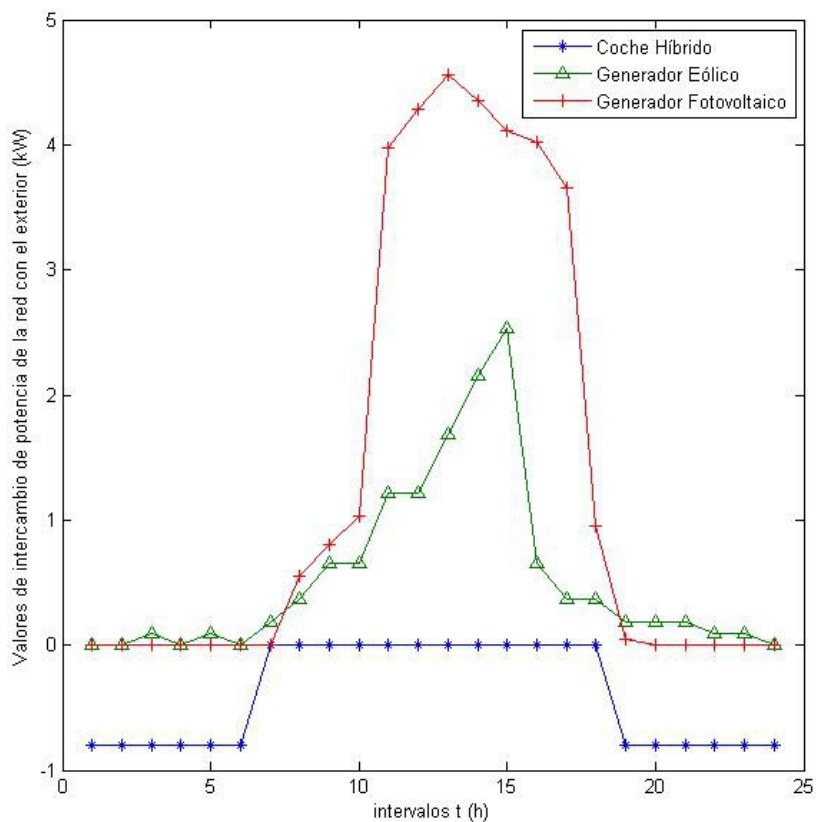


Figura 3-10. Programación de entrada/ salida de potencias (entrada de datos).

Hemos insertado previamente a la simulación, las gráficas con las potencias previstas para la introducción de datos de entrada de energía procedente de renovables ($P_{WINDSUN}$), y de salida de energía en forma de hidrógeno para la recarga del vehículo híbrido ($P_{FUELCAR}$).

A continuación se ilustra una gráfica con el resultado del control predictivo sobre nuestro sistema de dos entradas y dos salidas.

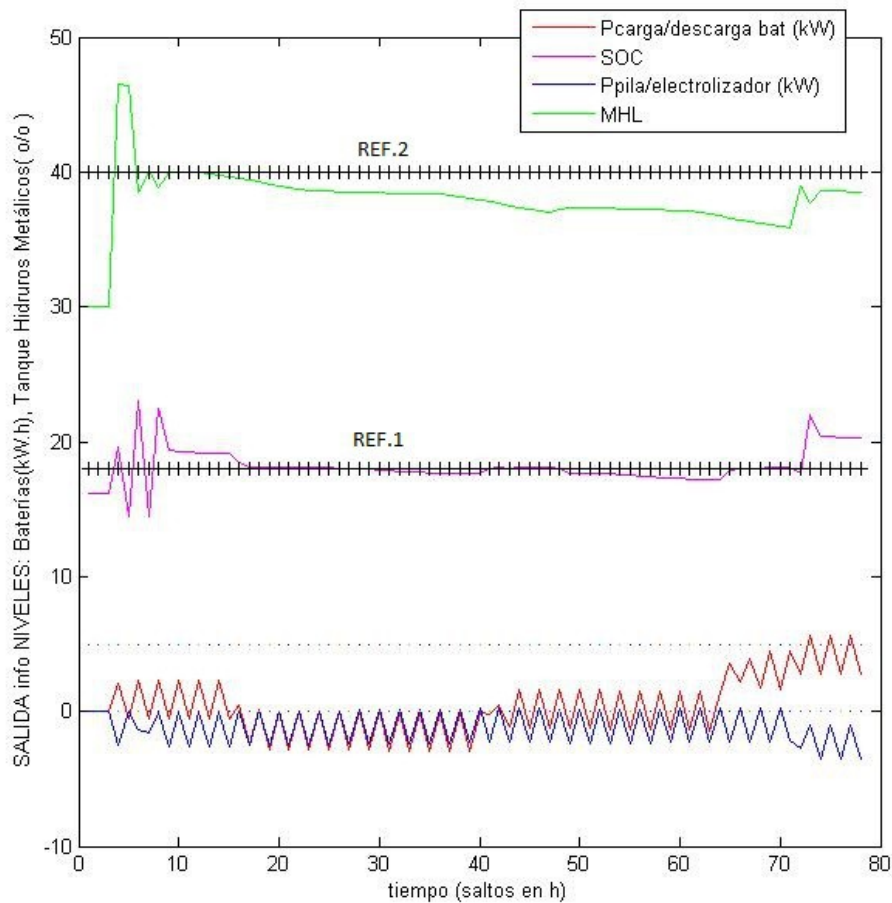


Figura 3-11. Simulación de GPC usando Matlab.

notas: REF.2 al 40% (0.4 Nm³), nivel de referencia del depósito MHL.
REF.1 al 50% (18 kW.h) nivel de referencia de almacenamiento SOC.

Es importante señalar que los resultados expuestos anteriormente se han obtenido para un sistema de distribución sin pérdidas en las líneas.

Una vez que también hemos modelado la red con ayuda del software OpenDSS, en la subsección siguiente se vuelve a simular el funcionamiento del algoritmo GPC en el control de la red, pero teniendo en cuenta las pérdidas tanto en conductores como en los diferentes equipos (baterías, electrolizador, pila de combustible, transformadores, rectificadores, reguladores e inversores).

La principal diferencia entre ambos modelos es que mediante OpenDSS se simulan las pérdidas reales en que se incurre por la operación de la microrred.

4 SIMULACIÓN CON HERRAMIENTAS INFORMÁTICAS

‘¡Dios no juega a los dados con el mundo!’

- Albert Einstein-

Todo lo desarrollado anteriormente sería un intento más o menos afortunado de tratado teórico, si no fuera por el hecho de que disponemos de herramientas informáticas potentes, que nos permiten realizar una simulación del funcionamiento real de la planta, de forma que el usuario puede modificar parámetros en ‘tiempo real virtual’, y observar además mediante gráficas si la evolución de la planta es la deseada. En este capítulo se explica la implementación con los programas Matlab® y OpenDSS (software libre), incluyendo un Anexo 2 al final, como ampliación de los códigos utilizados.

La introducción y familiarización del autor de este proyecto con el software OpenDSS ha sido más fácil en gran medida por la consulta del Proyecto Fin de Carrera referenciado en la bibliografía [6], del cual se extraen prácticamente la totalidad de los contenidos de las dos próximas secciones 4.1 y 4.2.

Por mi parte animo a la consulta de dicho trabajo, a cualquier iniciado en OpenDSS que esté interesado en ampliar conceptos, ya que los utilizados y explicados en este proyecto son muy básicos.

4.1 Introducción a OpenDSS

OpenDSS es un software de simulación de sistemas de distribución de energía eléctrica desarrollado por EPRI (Electric Power Research Institute). Se trata de un programa del tipo software libre utilizado en la simulación de propiedades eléctricas en el dominio de la frecuencia con las funcionalidades propias de los simuladores comerciales.

Es un software aún en desarrollo que va implementando progresivamente nuevas herramientas que tienen en cuenta futuras necesidades relacionadas con los esfuerzos de modernización de las redes actuales.

Una de las características innovadoras de OpenDSS respecto a otras herramientas típicas de análisis de sistemas de distribución de energía eléctrica, es su modo de solución cuasiestática, es decir, las simulaciones en tiempo secuencial. El programa puede llevar a cabo simulaciones de ciclos de trabajo con la periodicidad deseada: anuales, diarios, etc.

Cada elemento del sistema eléctrico bajo estudio puede tener una única curva de carga o funcionamiento; esta curva es una característica importante porque los contadores de energía modernos pueden proporcionar datos de un intervalo de demanda para cada cliente. La facilidad con la que OpenDSS realiza esta tarea, con sus opciones *Monitor* y *EnergyMeter*, permite capturar los resultados de series temporales en largas simulaciones.

Estas opciones pueden ser muy útiles para el análisis con recursos en la utilización de energías renovables, almacenamiento, vehículos eléctricos, etc., cuya solución es difícil de obtener sin modelar el comportamiento del sistema como función del tiempo.

Otra de las ventajas de utilizar OpenDSS, es que el código fuente es libre, de forma que los investigadores de redes inteligentes que necesiten capacidades avanzadas de simulación que aún no hayan sido implementadas, pueden modificar dicho código para desarrollarlas.

4.1.1 Elementos básicos en OpenDSS.

Como se ha dicho anteriormente, el programa OpenDSS se utiliza para modelar circuitos y sistemas eléctricos; para ello utiliza una serie de elementos básicos que pueden ser configurados con una gran flexibilidad. Algunos de los elementos más utilizados son:

a) Elementos de suministro de potencia.

- Líneas. Todo tipo de líneas y cables para distribución.
- Transformadores. Multifase y multibobinados.
- Condensadores. En serie y en derivación.

b) Elementos de conversión de potencia.

- Generadores.
- Cargas disipadoras de energía.
- PVSysyem. Sistemas de energía fotovoltaica incluyendo panel e inversor.
- Almacenaje.

c) Medidores.

- Energy Meter. Permite medir potencias y pérdidas.
- Monitor. Permite medir cantidades en un determinado punto del circuito.
- Sensores.

4.1.2 Programación en OpenDSS.

El software OpenDSS está diseñado para funcionar de forma autónoma mediante el archivo ejecutable *OpenDSS.exe*, esto es, con su propia interfaz de usuario del programa, según se representa en la figura 4-1.

Este programa recibe instrucciones en forma de texto permitiendo mayor flexibilidad al usuario. Mediante estas instrucciones se pueden generar los códigos de diseño de circuitos, simulación y otras funciones ya mencionadas. Además ofrece multitud de herramientas de control y análisis de cada elemento del circuito.

Puede accederse al programa tanto a través de una aplicación independiente (archivo ejecutable *OpenDSS.exe*), así como desde la consola de Matlab a través del módulo COM server.

La aplicación independiente cuenta con una interfaz de usuario muy básica, aunque funcional, que permite interactuar con el programa.

El módulo que llamamos COM server permite conectar OpenDSS con otros programas tales como Matlab, Visual Basic, Python o Excel, proporcionando de esa manera una gran capacidad de análisis de la información.

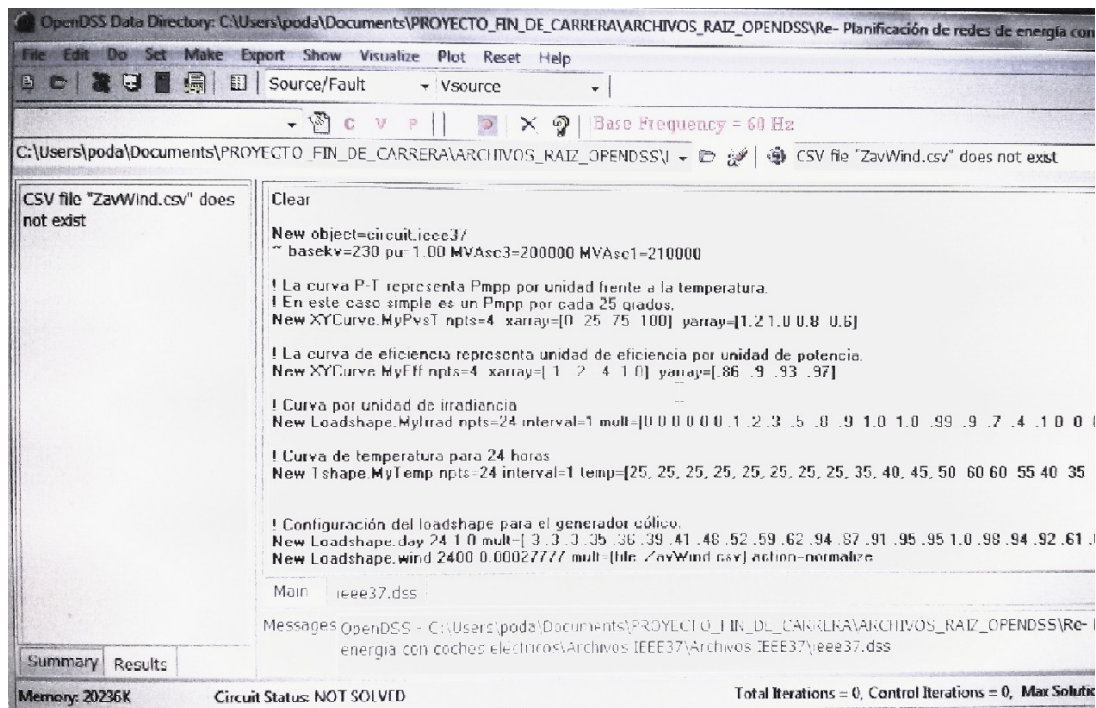


Figura 4-1. Interfaz gráfica de OpenDSS.

PASO PREVIOS EN LA PROGRAMACIÓN:

a) Primero, usar la frecuencia correcta. Antes de empezar es aconsejable configurar la frecuencia base por defecto a la que se vaya a utilizar en el circuito normalmente. En caso contrario, si el circuito tiene especificada una frecuencia diferente a la frecuencia base, no funcionará.

Set DefaultBaseFrequency = 60

b) Configurar un SourceBus. Según las necesidades del modelo se configura el *sourcebus* (bus de referencia, con este nombre para que el programa lo reconozca) a través de una línea o un transformador.

c) OpenDSS imita redes reales. Por lo tanto se necesita colocar monitores y medidores.

CONFIGURACIÓN BÁSICA DE ELEMENTOS Y DISPOSITIVOS:

a) **Comando "Clear"**. Este comando debe encabezar todos los scripts de OpenDSS para evitar fallos y errores al compilar.

Su función es eliminar de memoria todas las configuraciones, simulaciones y archivos temporales que pudiesen existir.

b) **Configurar la ruta de trabajo**. Esto es especialmente útil cuando se trabaja con varios scripts que no necesariamente se encuentran en el mismo fichero. Su sintaxis es:

set datapatch = C:\OpenDSS\CircuitosEjemplo

c) Crear el nuevo circuito. Se crea el circuito como un nuevo objeto de OpenDSS con el comando *New*. Posteriormente se configura el circuito y la fuente de tensión principal.

En el siguiente ejemplo se trata de una fuente alterna de 3 fases con 115 kV de tensión trabajando a una frecuencia de 60 Hz, y con un ángulo de fase nulo.

```
New object= circuit .DSSLibtestckt
```

```
~ basekv=115 pu=1.00 Angle=0.0 Frequency=60.0 Phases=3 Mvasc3=20000 Mvasc1=21000
```

```
~ x1r1=4.0 x0r0=3.0
```

Hay otros parámetros que no son fundamentales pero que se utilizan para un modelado más exacto de la fuente de tensión, y son los siguientes:

Mvasc3 y *Mvasc1* hacen referencia a tensiones en cortocircuito para 3 y 1 fase respectivamente; mientras que *x1r1* y *x0r0* se utilizan para configurar relaciones de impedancias internas.

d) Curvas LoadShape. El objeto *LoadShape* se utiliza para simulaciones temporales (diarias, anuales, etc). Se define mediante una serie de multiplicadores que típicamente varían de 0.0 a 1.0; estos multiplicadores se aplican a los valores de la carga en kW para representar la variación de la carga respecto al tiempo. Un objeto *loadshape* es, al fin y al cabo, un vector cuyas componentes son los multiplicadores. Veamos un ejemplo.

```
New loadshape.day 8 3.0
```

```
~ mult=(.3 .36 .48 .62 .87 .95 .94 .60)
```

Siguiendo la estructura de cualquier objeto en OpenDSS, se crea con el comando *New*, se define el tipo de objeto (*LoadShape*) y el nombre (*day*). El valor 8 representa el número de puntos para definir la curva. El valor 3.0 define el tiempo de intervalo en horas. Por último se añaden los multiplicadores. En el caso de usarse, el comando *normalize* escala los multiplicadores para que el valor pico sea igual a 1.

e) Líneas. El elemento *Línea* de OpenDSS se utiliza para modelar la mayoría de cables o líneas multifase. En el software se trata como un elemento de potencia que se caracteriza por su impedancia. Las impedancias de una línea pueden ser especificadas mediante valores de componentes simétricos, una matriz o haciendo referencia a un objeto *linecode* existente.

En el caso de que se especifique la propiedad *Geometría*, las matrices de impedancias o el objeto *linecode* serán ignorados. Con la geometría introducida, el software calcula las matrices de impedancia cada vez que cambia de frecuencia. Las longitudes de línea pueden ser introducidas en cualquier unidad, pero el cambio de unidad hay que declararlo o se entenderán por defecto.

Los objetos *linecode* son librerías que contienen características de impedancia para las líneas y cables. En OpenDSS se puede describir una línea por su *linecode* y su longitud. También es posible especificar la geometría y que OpenDSS calcule las impedancias de la línea. Se puede prescindir del objeto *linecode*, ya que hay forma de introducir las impedancias manualmente, pero es aconsejable hacerlo cuando hay un número considerable de líneas con las mismas características o ya se tienen almacenadas en la librería, que suele ser lo habitual.

La impedancia de una línea se describe con su matriz de impedancia y su matriz nodal capacitiva. Estas matrices se pueden especificar directamente o pueden ser generadas mediante los datos de los componentes simétricos.

$$Z = R + jX = \begin{bmatrix} Z_{11}+Z_g & Z_{12}+Z_g & Z_{12}+Z_g \\ Z_{21}+Z_g & Z_{22}+Z_g & Z_{23}+Z_g \\ Z_{31}+Z_g & Z_{32}+Z_g & Z_{33}+Z_g \end{bmatrix};$$

Figura 4-2. Matriz de impedancias.

Para pasar a programación OpenDSS el ejemplo de la figura 4-2 usamos un objeto *linecode*, siguiendo el mismo procedimiento que para crear otro tipo de objetos: se crea la línea con el comando *New* y se le asigna un nombre (*336matrix*) y el número de fases.

Se va a caracterizar la línea mediante matrices triangulares inferiores, tal que: $Z = R + jX$, donde R es la resistencia y X es la reactancia medida en unidades (Ω / unidad de longitud).

También se crea la matriz de capacitancia en unidades (nanoFaradios/ unidad de longitud) y se configura la corriente máxima (*Normamps*) y la corriente máxima de emergencia (*Emergamps*).

```
New linecode.336matrix nphases=3
~ rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455) !  $\Omega$  /1000ft
~ xmatrix=(0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 0.2025449) !  $\Omega$  /1000ft
~ cmatrix=(2.74 | 0.70 2.96 | 0.34 0.71 2.74) ! nanofaradios /1000 ft
~ Normamps=400 Emergamps=600
```

En el ejemplo anterior se observa que cada fila de cada matriz está compuesta por 6 números. Estos seis números corresponden al triángulo superior derecho de la matriz.

Las filas quedan separadas en la programación por el símbolo ‘ | ‘. Cada fila y columna representan las diferentes fases del conductor, por lo que un elemento (1,2) indicará que se encuentra entre las fases 1 y 2. La matriz es simétrica, se puede ver por ejemplo que el elemento $rmatrix(2,1)=0.02992424$ es el mismo que el $rmatrix(1,2)$.

Una vez definido y configurado el *linecode*, se crean las líneas con el comando *New*. También se designan los buses para cada una de ellas, y se describen mediante *linecode* y longitud, como se puede ver en el siguiente ejemplo:

```
New line. line1 bus1=subbus bus2=loadbus1 linecode=336matrix length=10
New line. line2 bus1=loadbus1 bus2=loadbus2 linecode=336matrix length=10
New line. line3 bus1=loadbus2 bus2=loadbus3 linecode=336matrix length=20
```

f) Cargas. En nuestro análisis una carga se modela como un elemento de conversión de energía.

Como el triángulo de impedancias es rectángulo, queda definido por dos lados: normalmente la carga se define por la potencia activa y por su factor de potencia (PF), o bien por la potencia activa (kW) y la potencia reactiva (kVr). Una vez definida puede ser modificada por una serie de multiplicadores (*loadshapes*).

La configuración predeterminada hace que la carga se comporte como una fuente de inyección de corriente. Por lo tanto, su matriz primitiva contiene sólo la impedancia que podría existir desde el neutro de una carga

conectada en estrella a tierra.

Se puede cambiar a un modo de admitancia con el comando *Set LoadModel*, haciendo que la carga se convierta en una admitancia que es incluida en la matriz *Y* del sistema. Éste sería el modelo utilizado para los estudios de fallos en los que no se alcanza la convergencia a causa de las bajas tensiones.

Las cargas se suponen equilibradas para el número de fases especificadas. Si se desean cargas desequilibradas, será necesario introducir las cargas monofásicas por separado.

Se define una carga mediante el código del siguiente ejemplo. Para asegurarse que el funcionamiento sea el deseado, hay que introducir esas propiedades en el orden indicado.

```
New load.load1 bus1=loadbus1 phases=3 kv=12.47 kw=1000 pf=0.88 model=1 class=1 duty=day
```

```
New load.load2 bus1=loadbus2 phases=3 kv=12.47 kw=500 pf=0.88 model=1 class=1 duty=day
```

~ conn=delta %Idem a la carga anterior, conn indica la configuración delta o estrella.

El comando *model* escoge una opción de entre 8 modelos diferentes; se ha seleccionado el 1 que es aquel donde P y Q son constantes.

El comando *class* es el número entero de segregación de la carga

El comando *duty* es el nombre del ciclo de trabajo, que se ha hecho coincidir con el *loadshape* previamente creado.

g) Transformador. Un transformador se implementa como un elemento de suministro de energía con dos o más terminales.

Los transformadores tienen una o más fases y el número de conductores por terminal, es siempre el número de fases más uno.

Dicho elemento consta de dos o más bobinados que pueden estar conectados de diferentes maneras, siendo la conexión estrella-triángulo la que se establece por defecto.

Para bobinados conectados en estrella (*wye*) el conductor adicional es el punto neutro; para bobinados conectados en triángulo, el terminal adicional está abierto internamente.

Para incorporar al circuito una subestación transformadora se utiliza un código similar al comentado en el siguiente ejemplo:

La línea *New transformer.subxfrm* crea nuevo objeto tipo transformador con nombre *subxfrm*.

El parámetro *windings* hace referencia al número de bobinados.

El parámetro *phases* hace referencia al número de fases.

La línea *buses=(SourceBus subbus)* indica los dos buses donde está la subestación, siendo equivalente definirlos *bus1=SourceBus bus2=subbus*.

El parámetro *conns* sirve para configurar la conexión en delta o en estrella.

El comando *kvs=(115 12.47)* se utiliza para indicar el ratio de transformación.

El comando *kvas=(20000 20000)* indica los KVA de referencia.

El parámetro *XHL=7* es el porcentaje de reactancia de alto a bajo, del bobinado 1 al 2.

```
New transformer.subxfrm phases=3 windings=2
```

```
~ buses=(SourceBus subbus)
```

```
~ conns='delta wye' kvs=(115 12.47)
```

```

~ kvas=(20000 20000) XHL=7
New transformer.reg1 phases=3 windings=2
~ buses=(loadbus3 regbus) conns='wye wye'
~ kvs=(12.47 12.47) kvas=(8000 8000) XHL=1

```

Mediante OpenDSS es posible controlar el voltaje de un bus remoto para emular varios dispositivos de una red inteligente. Esto se consigue al incorporar un control que hace la función de regulador de tensión (*RegControl*) de un transformador, conectado a un determinado bobinado del mismo.

En el siguiente código se propone un ejemplo de regulador de tensión.

```

New regcontrol . subxfmCtrl transformer =subxfm windings=2 vreg=125
~ band=3 ptratio =60 delay=10
New regcontrol . reg1Ctrl transformer =reg1 windings=2 vreg=122
~ band=3 ptratio =60 delay=15

```

En la configuración se indica el número de bobinados con la sentencia *windings*.

El parámetro *Vreg* es la configuración del regulador de tensión; multiplicando este valor por el *ptratio* se obtiene el voltaje a través del bobinado bajo control.

El parámetro *Band* hace referencia al ancho de banda en voltios para el bus controlado.

Pratio es la relación entre la tensión del bobinado y del regulador.

El parámetro *Delay* es el retraso en segundos, que se usa para determinar qué controlador actúa primero.

h) Capacitores. El modelo de condensador se implementa, básicamente, como un elemento de suministro de potencia de dos terminales, y sus conexiones se ilustran en la figura 4-3.

Si no se especifica una conexión para el segundo bus, se toma por defecto dicha conexión al nodo 0 (referencia de tierra) del mismo bus al que está conectado el primer terminal. Es decir, el valor predeterminado es una batería de condensadores de derivación a tierra en estrella (*wye*).

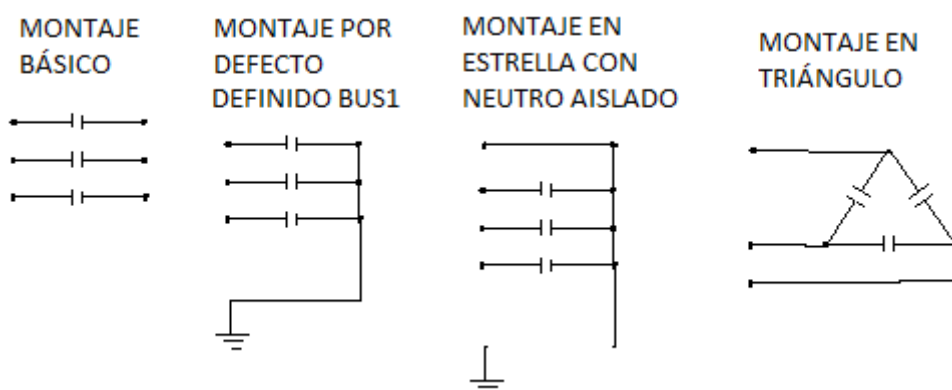


Figura 4-3. Definición del objeto *capacitor*.

El segundo terminal se elimina si se especifica que la conexión es de tipo "*delta*".

Para configurar un condensador en serie se tiene que especificar una conexión para el segundo bus.

Para un condensador en estrella sin conexión a tierra se ajustan todos los segundos terminales a un nodo vacío en el terminal del primer bus.

Cualquier otra conexión es posible especificando correctamente la conexión de los terminales.

Se puede usar un objeto *CapControl* para controlar un objeto *capacitor*.

Esto tiene su aplicación cuando se quiere implementar un banco de filtros de varios pasos sintonizado. Cuando un objeto *Capcontrol* conmuta un condensador, lo hace para aumentar o disminuir el paso activo.

En el próximo ejemplo se incorpora un capacitador con un sistema de control *CapControl*.

```
New capacitor .Cap1 bus1=loadbus2 phases=3 kvar=600 kv=12.47
```

```
New capcontrol.Cap1Ctrl element=line . line3 terminal=1 capacitor =Cap1
```

```
~ type=current ctratio =1 ONsetting=60 OFFsetting=55 delay=2
```

Como no se ha especificado una conexión para el segundo bus, se toma por defecto la referencia a tierra.

Con *Capcontrol* se configura el control del condensador. Básicamente se puede monitorear la tensión y la corriente de un elemento y enviar mensajes de conmutación a un condensador.

En el ejemplo se indica el elemento que va a ser monitoreado (*line3*), y se indican además terminal=1, condensador bajo control (*Cap1*) y tipo de control (*type=current*).

El tipo de control puede ser de corriente, tensión o tiempo. En el ejemplo se utiliza un tipo de control en corriente.

El parámetro *ONsetting* indica el valor para cambiar el condensador a *ON* mientras que *OFFsetting* hace lo mismo con el estado *OFF*. Las unidades de ambos parámetros dependen del tipo de control (en este caso amperios).

Delay es el retraso en el tiempo (en segundos) desde el momento en que se arma el control hasta que se envía la orden de conmutación.

i) Generador. El objeto *Generator* en OpenDSS es un elemento de conversión de energía similar a una carga.

Al igual que aquella, se define por kW nominales y PF, o bien por kW y kvar, pudiendo ser modificados por una serie de multiplicadores o *LoadShape*.

En un estudio de flujo de potencia, el generador se modela como una carga negativa, es decir, una carga que genera potencia en lugar de consumirla.

Para el modo *Armónicos*, el generador se convierte en una fuente de tensión.

Para el modo *Dinámico*, el generador se comporta como una fuente de tensión detrás de una impedancia, dependiendo esta impedancia del modelo elegido.

Si el valor de "envío" (propiedad *DispValue*) es 0, el generador siempre sigue la curva de funcionamiento asociada, que es simplemente un objeto *Loadshape*. Decimos en este caso que el generador está siempre encendido, siempre siguiendo la curva especificada.

Si *DispValue* > 0, entonces el generador se enciende sólo cuando el multiplicador de la carga supera el valor del parámetro *DispValue*.

Los generadores tienen sus propios medidores de energía que graban los siguientes parámetros:

1. Total kwh.
2. Total kvarh.
3. Máximo kW.

4. Máximo kVA.

5. Horas de operación.

6. \$ (Precio * energía generada).

Los diferentes modelos de generador para las simulaciones de flujo de potencia son:

1. P constante, Q (dependiente del *LoadShape*).

2. Z Constante (Para una aproximación de una solución simple).

3. P constante, jV_j como un flujo de corriente estándar con magnitudes de tensión y ángulos como variables.

4. P constante, Q fijada. P sigue el *LoadShape* mientras que Q es siempre la misma.

7. Corriente limitada y modelo de P y Q constantes. Este es el comportamiento de algunos inversores, elemento básico en una microrred y que veremos más adelante.

En el siguiente ejemplo se configura un generador eólico de 8 MW como un objeto generador.

```
New generator.gen1 bus1=regbus kV=12.47 kW=8000 pf=1 conn=delta duty=wind Model=1
```

j) Monitor. Un monitor es un objeto en OpenDSS que se conecta al terminal de otro elemento del circuito.

Su misión es tomar datos de tensión, intensidad o potencia con respecto al tiempo en todas las fases, funcionando como un monitor de potencia real. Se puede configurar de varias formas para obtener diferentes parámetros.

Para la configuración del monitor, se selecciona el elemento y terminal bajo estudio. El parámetro *mode* es un código para describir las magnitudes a archivar. El monitor puede guardar dos tipos de cantidades: A) Tensión y corriente; B) Alimentación.

Veamos algunos de los modos:

→ 0. Modo estándar: V, I en cada fase

→ 1. Potencia en cada fase (kW y kVARs)

→ 2. Tomas del transformador. Conexión a un bobinado

→ 3. Variables de estado conectado a un PCElement (Power Conversion Element)

En el ejemplo siguiente se incluyen 3 monitores: el primero para analizar las potencias generadas en el generador eólico; el segundo para monitorizar tensión y corrientes en la segunda carga; y el tercero para monitorizar la secuencia de tensiones y corrientes en la línea 3.

```
New Monitor.gen1 element=generator.gen1 terminal=1 mode=1
```

```
New Monitor.loadbus2 element=load.load2 terminal=1 mode=0
```

```
New Monitor.line3 element=line.line3 terminal=1 mode=48
```

k) Energy Meter. Es un medidor inteligente que está conectado al terminal de un elemento del circuito.

Simula el comportamiento de un contador de energía real, pero con más capacidad, ya que puede acceder a los valores de otros puntos del circuito, además de donde esté instalado.

Mide potencias y energías, y también pérdidas y valores de sobrecarga dentro de una región definida del circuito.

Tiene 2 tipos de registro: A) Valores de energía. B) Potencia máxima.

En el siguiente ejemplo se incorpora un *Energy meter* para obtener datos de una línea y poder hacer gráficas.

```
New Energymeter.em1 element=line.line1
```

I) Modo de control. *ControlMode* se utiliza con el modo *dutycycle* o *dynamic*, y puede tomar valores [OFF | STATIC | EVENT | TIME].

El valor por defecto es *STATIC*. En el modo *STATIC* el tiempo no avanza. Las acciones de control se ejecutan en orden desde la de menor tiempo de actuación hasta la finalización de todas las acciones. Se utiliza este modo para soluciones de flujo de potencia que requieran varios cambios del regulador por solución.

Modo *EVENT*. Es un tipo de solución orientada a eventos. Sólo se ejecutan las acciones de control cercanas al evento y el tiempo se hace avanzar automáticamente hasta el instante del mismo.

Modo *TIME*. Es un tipo de solución orientada a control del tiempo. Las acciones de control se ejecutan cuando el tiempo de espera de la acción se alcanza o se supera. Se utiliza el modo *TIME* para modelar un control y un modo de solución externo al software OpenDSS, tales como *Daily* o *Dutycycle* que avanzan en el tiempo, o cuando se ajusta el tiempo (horas y segundos) explícitamente desde un programa externo como es el caso de Matlab.

En el siguiente ejemplo, como paso previo a seleccionar el modo de control, se ejecutan dos comandos:

```
Set voltagebases =(115 12.47 .48)
```

```
Calc voltagebases
```

El primero sirve para definir las bases de tensiones del circuito, de forma que los informes sean expresados en la unidad correspondiente.

El segundo, *Calc voltagebases*, genera la lista de buses y calcula las tensiones base de éstos.

En el siguiente ejemplo se muestra cómo se seleccionan los modos de control de la simulación para hacer que las funciones de captura y registro funcionen de manera sincronizada con el resto de la simulación.

```
Set controlmode=time
```

```
Set controlmode=duty number=86400 hour=0 stepsize=1 sec=0
```

La última configuración mostrada es para una simulación basada en tiempo. Durante 24 horas (86400 segundos), con intervalos de 1 segundo, empezando en la hora 0 y segundo 0.

II) Ejecución de la simulación. Para llevar a cabo la simulación se ejecuta el comando que resuelve el circuito.

```
Solve
```

Este comando ejecuta la solución en el modo, *Set controlmode=comando*, definido en el paso anterior.

Se puede ejecutar una o cientos de soluciones. La "solución" pasa a ser un objeto de DSS asociado al circuito activo.

m) Ejecución del código. Para ejecutar el código desde OpenDSS, éste deberá estar seleccionado, luego se

debe hacer clic derecho sobre él y selecciona la opción *Do*.

Una operación análoga es pulsar *Ctrl + D* cuando el código esté seleccionado.

Para ejecutar la opción *Do (Ctrl + D)*, se puede seleccionar una o más o líneas de código, siendo solo ejecutado la parte del código seleccionado.

Por ejemplo, una vez ejecutado y resuelto (*solve*) el circuito, se pueden generar gráficas para la interpretación de resultados. En OpenDSS como aplicación independiente se hace ejecutando (*do*) el código con el comando *plot*, como vemos a continuación. También se podría haber incluido la opción de gráficos en el código completo antes de resolver, *solve*, con lo cual esta información ya queda incorporada al objeto “solución” asociada al circuito activo.

n) Gráficas. Hay muchas opciones para el comando *Plot*, de forma que tiene su propia rama en el árbol de ayuda de la aplicación OpenDSS.

Algunas opciones que aquí se utilizan son:

Selección del tipo de gráfica. "type":

Circuit | Monitor | Daisy | AutoAdd | General (bus data) | Loadshape | Tshape | Princeshape | Profile .

Un *plot* de un circuito requiere que se definan las coordenadas de los buses. Por defecto, el grosor de las líneas del circuito se dibuja proporcional a la potencia.

También se pueden realizar gráficas a partir de los monitores, dibujando uno o más canales del elemento que está siendo monitorizado.

Selección de la cantidad mostrada. "quantity":

Voltage | Current | Power | Losses | Capacity.

Opción que sirve para indicar qué magnitud será representada en la gráfica *plot*.

– Selección del objeto de OpenDSS utilizado, "object" =

=[Nombre de zona a graficar| Nombre de monitor| Nombre de Bus/Circuito| Nombre Loadshape].

Sirve para especificar qué objeto será el usado en la función *plot*.

– Canales que se mostrarán. "Channels": Sirve para indicar los canales graficados en el *plot*.

Hay más opciones adicionales que se pueden consultar en la ayuda. Veamos un ejemplo.

Plot type=monitor, object =loadbus2, Channels =(1,3,5)

Visualize element=line . line1 what=powers

Plot profile

El primer *plot* sirve para mostrar los valores registrados por el monitor instalado en la segunda carga en diferentes canales.

En la segunda carga se especifica que se van a visualizar las potencias.

El comando *Plot profile* muestra el perfil de la tensión en el alimentador.

OpenDSS genera un archivo de texto en el cual se encuentran las potencias que fluyen a través de todos los elementos del sistema en forma de componentes simétricas.

Para generar un archivo con la información acerca de la potencia de sobrecarga de cada elemento y las pérdidas totales del sistema se utiliza el siguiente comando.

Show Powers

show losses

4.2 Utilización y Control de OpenDSS desde Matlab

En esta sección pretendo explicar las claves de la programación y los pasos necesarios para hacer funcionar OpenDSS desde la consola de Matlab, sin necesidad de ejecutar el primero directamente. Para recopilar las siguientes líneas, haciendo una pequeña adaptación de códigos y figuras, se ha consultado el referido anteriormente [12], Trabajo de Fin de Grado que a su vez tomaba sus contenidos del documento de Sandia Corporation [14].

4.2.1 Comunicación con OpenDSS y compilación desde Matlab.

Se puede hacer uso de OpenDSS sin necesidad de utilizar su interfaz gráfica (OpenDSS GUI), y para ello recurriremos al programa Matlab (pero también podría usarse Visual Basic, Excel o Python) a través del COM Server, el cual ha sido desarrollado como una aplicación DLL.

El software Matlab tiene integrado un servidor ActiveX que se utiliza para comunicarse con el COM Server de OpenDSS. De esta manera dicho servidor será la interfaz entre ambos programas, como se puede ver en la figura 4-4.

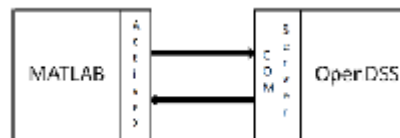


Figura 4-4. Comunicación entre OpenDSS y Matlab.

Esta característica proporciona una gran capacidad de análisis de la información, permitiendo incluso implementar algoritmos que utilicen los resultados de las simulaciones de OpenDSS. La figura 4-5 muestra cómo interactúan los diferentes módulos dentro de la estructura de OpenDSS.

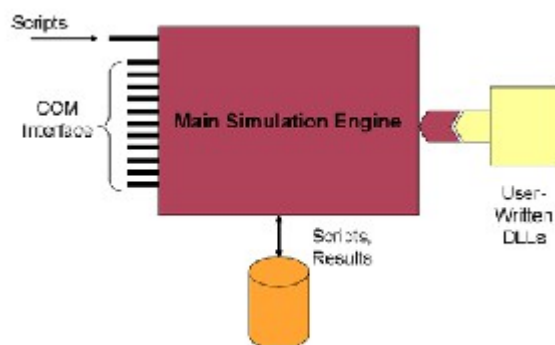


Figura 4-5. Estructura de OpenDSS.

A través de Matlab hay mucha documentación para cada función a la que se puede acceder mediante la siguiente línea.

help getBusInfo;

La inicialización del COM Server es realizada a través de una función creada en el espacio de trabajo de Matlab.

El primer paso para iniciar la interfaz COM es hacer una llamada a la función *DSSStartup*. Hay varias formas de hacerlo y se muestra un ejemplo de la más habitual.

```
[DSSCircObj, DSSText, gridpyPath] = DSSStartup;
```

DSSStartup inicia OpenDSS en segundo plano y devuelve un puntero con tres salidas:

- ***DSSCircObj***, contiene el circuito activo (*DSSCircObj.ActiveCircuit*) y la interfaz de texto para OpenDSS (*DSSCircObj.Text*), y estará vacío mientras que el circuito no sea compilado.
- ***DSSText***, es la interfaz de texto contenida dentro de *DSSCircObj*. Ha sido definido de esta manera para facilitar el uso dentro de la ventana de comandos de Matlab. *DSSCircObj.Text.Command*, *Text.command* y *DSSText.Command* devolverán el mismo resultado en forma de interfaz de texto.
- ***gridpyPath***, en el ejemplo es una cadena que contiene la ruta de localización en el ordenador para la herramienta. *DSSStartup* devolverá un error si MATLAB no es capaz de crear la conexión con OpenDSS. El motivo más común de error es que la versión de OpenDSS no se haya instalado correctamente en el ordenador.

Es importante conocer que el programa OpenDSS que se conecta con MATLAB por el servidor COM es diferente al ejecutable con interfaz gráfica de OpenDSS (GUI). Es decir, ninguna información, circuito, solución o parámetro configurado en la interfaz gráfica de OpenDSS será mostrado por la versión del servidor COM de OpenDSS y viceversa.

Para abrir un circuito creado en OpenDSS desde Matlab es necesario haber compilado mediante la script (*.dss*) del circuito. Para compilar se usa la interfaz de texto con el código *compile*, indicándole la ruta en el caso de que el circuito en OpenDSS y el archivo (*.m*) de Matlab se encuentren en directorios diferentes. Se muestra a continuación.

```
DSSText.Command = 'Compile ¿Directorio? archivo.dss';
```

Es recomendable utilizar siempre la ruta del circuito en formato *.DSS* con el comando *compile* para evitar errores.

Ahora que la interfaz COM ha sido iniciada y el circuito compilado, se puede comenzar a utilizar la consola de Matlab para la comunicación entre los dos programas. A continuación se muestran los comandos más habituales para interactuar, extraídos de la documentación de OpenDSS.

DSSCircObj.get

El comando devuelve algunos otros punteros de objetos de la interfaz del servidor COM en OpenDSS. Uno de estos sub-punteros es la interfaz de *ActiveCircuit*, que hace mención al circuito compilado en OpenDSS y contiene todos los parámetros y todas las soluciones de flujo de potencia. Es útil redefinir este puntero para utilizarlo con comandos más cortos que cumplirán exactamente la misma función sin tener que llamar a la función completa *DSSCircObj*, como se ve en el siguiente ejemplo.

```
DSSCircuit = DSSCircObj.ActiveCircuit ;
```

Análogamente, como ya se ha hecho con *ActiveCircuit*, se pueden volver a utilizar los comandos *get* y *methods* en elementos u objetos para obtener más posibilidades o incluso consultar información sobre campos específicos de dichos elementos.

```
DSSCircuit.Lines .methods
```

```

DSSCircuit.Lines . get
DSSCircuit.Lines .LineCode
DSSCircuit. Capacitors . get
DSSCircuit. Capacitors .Name

```

Es conveniente tener en cuenta que la mayoría de campos en estas interfaces tienen información sobre un elemento individual, generalmente el primer elemento por defecto. Para obtener información de un elemento en particular (una línea o un capacitor concreto), tendrá que ser *activado* previamente en la interfaz.

Es importante asimilar el concepto de cómo utilizar las iteraciones necesarias para obtener todos los datos de un circuito. Los métodos *.first* y *.next* que son devueltos al hacer la llamada *DSSCircuit.Lines.get* son usados para cambiar el índice del objeto. Se usa el método *.first* para asegurarse de que se resetea el índice y se empieza por el primer objeto en la lista de un determinado tipo de elemento (línea, capacitor, generador, etc.) y entonces se usa el método *.next* para ir iterando paso a paso en dicha lista. Veamos un ejemplo.

```

% Se selecciona el primer elemento de la lista de transformadores
DSSCircuit.Transformers .first ;
% Se obtiene el número total de transformadores
numXfmr = DSSCircuit.Transformers.count;
% Se asigna previamente
xfmrNames = cell(numXfmr, 1);
% Se itera
for ii = 1:numXfmr
%Se obtiene el nombre del elemento correspondiente al índice ii
xfmrNames{ii} = DSSCircuit.Transformers .Name;
% Se continúa al siguiente transformador
DSSCircuit.Transformers .Next;
end

```

Si se observa el uso de la variable *numXfmr*, se podría pensar que es más útil no utilizarla y hacer sólo la llamada *DSSCircuit.Transformers.count* en las dos ubicaciones en las que se ha utilizado *numXfmr*, sin embargo, la función *DSSCircuit.Transformers.count* tendría que pasar dos veces por el servidor COM y ralentizaría considerablemente el programa, por lo tanto es más eficiente computacionalmente llamar a esa función una sola vez y usar una variable del espacio de trabajo de Matlab para recorrerla dos veces.

Hay varios métodos y campos alternativos que devuelven o contienen información global, lo que ayuda a evitar procesos de iteración y hacer más eficiente el código. El siguiente método hace exactamente lo mismo que el bucle anterior pero con una sola instrucción.

```

xfmrNames = DSSCircuit.Transformers.AllNames;

```

Otra forma de obtener información de un determinado elemento del circuito es mediante la interfaz de *elemento activo*, que contiene más datos incluso que los de la interfaz vista anteriormente (de elemento tipo), por ejemplo facilita la solución de los flujos de energía/potencia.

Llamando a las funciones:

```
DSSCircuit.ActiveCktElement.get
```

```
DSSCircuit.ActiveCktElement.methods,
```

se puede comprobar que en este modo no existen los métodos *.First* o *.Next*. Esto es porque la interfaz de *elemento activo* requiere que el elemento sea seleccionado manualmente mediante la función *DSSCircuit.SetActiveElement*. Se muestra un ejemplo a continuación:

```
%Se consiguen los nombres de línea y se monta la estructura
lineNames = DSSCircuit.Lines .AllNames;
Lines = struct ( 'name',lineNames);
% Se itera y se recuperan los datos de los buses
for ii =1:length(Lines)
% Se nombra elemento activo a las diferentes l líneas
DSSCircuit.SetActiveElement([ ' line . ' Lines( ii ) .name]);
% Se obtiene el nombre de los buses , una matriz de 2 columnas
lineBusNames = DSSCircuit.ActiveElement.BusNames;
Lines( ii ) .bus1 = lineBusNames{1};
Lines( ii ) .bus2 = lineBusNames{2};
end
```

4.2.2 Comandos y programación desde la consola de Matlab.

Aparte de la interfaz de circuito, otra herramienta para interactuar con el servidor COM es la interfaz de texto. Esta interfaz puede ser utilizada para pasar comandos en su propio lenguaje de programación a OpenDSS y que sean ejecutadas directamente por dicho programa.

En el siguiente ejemplo, la interfaz de texto ha sido usada para resolver el circuito después de configurar un determinado modo de control, el tiempo y el intervalo de tiempo (*h*). Esta cadena de comandos es compilada en OpenDSS, por lo que que la interfaz de texto puede ser utilizada para cualquier tarea que se pudiese hacer en OpenDSS con sus propios scripts.

```
DSSText.command = 'Set controlmode=static' ;
DSSText.command = 'Set mode=snapshot number=1 hour=0 h=1 sec=0';
DSSText.command = 'solve';
```

Uno de los usos más comunes de la interfaz de texto es añadir y editar elementos del circuito.

```
% Comando para asegurarse de que no existen generadores
DSSCircuit.Generators . get ;
% Se añade un generador llamado PV
DSSText.command = 'new generator.PV bus1= n292757 phases=3 kv=34.5
```

```

kw=500 pf=1 enabled=true';
% Si se vuelve a ejecutar el comando, ahora sí saldrá el generador
DSSCircuit.Generators . get ;
% Se selecciona como elemento activo y se comprueba la información de sus buses
DSSCircuit.SetActiveElement(' generator .pv' );
DSSCircuit.ActiveElement.BusNames
% Ahora se cambia a otro bus y se observa el cambio
DSSText.command = 'edit generator .PV bus1=n1325391 kv=13.2';
DSSCircuit.ActiveElement.BusNames

```

Usando los comandos de OpenDSS "*new*" y "*edit*", diferentes elementos pueden ser añadidos, movidos y cambiados desde Matlab como se muestra en el ejemplo anterior.

Cuando se resuelve el circuito en un bucle de Matlab, OpenDSS por defecto reinicia la configuración en cada paso si no hacemos algo para evitarlo y no perder la continuidad.

Una forma de hacerlo es asignar un valor para el intervalo. Tras ajustar $h = 1$ (h es el paso de tiempo en segundos), se consigue que el software realice las sucesivas soluciones sin reiniciar su contador temporal interno, algo muy útil para simulaciones temporales.

Veamos ahora cómo obtener información de los diferentes elementos de un circuito desde el servidor COM de Matlab con la función *get* ya introducida anteriormente.

```

% Llamada sin especificar nombres. Devuelve todos los buses .
Buses = getBusInfo(DSSCircObj);
% Llamada especificando nombres (poner entre llaves )
Buses = getBusInfo(DSSCircObj,{'N1311915'});
% Llamada con un vector de nombres
Buses = getBusInfo(DSSCircObj,{'N1311915', 'N312536'});
% Llamada especificando los nombres vía servidor COM
Buses = getBusInfo(DSSCircObj, DSSCircObj.ActiveCircuit.AllBusNames);

```

Las funciones *get* son herramientas muy útiles para automatizar algunos de los aspectos más tediosos de la interacción con el servidor COM. Cuando se hace una llamada a esta función, se debe pasar el puntero que se quiere consultar y, opcionalmente, un vector con el nombre de los elementos a consultar. Si no se incluye el nombre de los elementos, serán devueltos por defecto todos los dispositivos habilitados; en cambio, si se incluye el nombre del elemento, se obtendrá información de él aunque este inhabilitado en el circuito.

Las funciones *get* han sido diseñadas para devolver todos los parámetros posibles de cada objeto. Estas funciones no devuelven punteros a otros objetos, sino que obtienen estructuras que contienen datos estáticos correspondientes a la solución más reciente del circuito.

Si el circuito se modifica o si hay una nueva solución del circuito, habrá que hacer una nueva llamada a las funciones *get* para actualizar las estructuras con los datos más recientes.

Para hacer un uso más eficiente de las estructuras de datos que se obtienen con la función *get*, el valor de cada campo en la estructura se obtiene mediante los siguientes pasos:

1.- poner el nombre del campo una vez que haya transcurrido al menos un periodo de la simulación.

2.- buscar los nombres de los campos de una estructura dada mediante la llamada *fieldnames()*.

El resultado de una función *get* es una estructura en forma de matriz, como se ilustra en el siguiente ejemplo.

```
% Encontrar todos los nombres en la estructura Buses
fields = fieldnames (Buses);
% Devuelve el nombre del primer bus
Buses(1) .name
% Devuelve el número de fases del segundo bus
Buses(2) .numPhases
% Devuelve cuántos buses hay en la estructura
length(Buses)
```

Aquí se observa como en *Buses* están almacenados los parámetros de todos los buses, y cada bus individual es una estructura dentro de *Buses* y sus valores pueden ser obtenidos mediante el adecuado índice.

Cuando se intenta acceder a datos de múltiples elementos en una estructura hay que asegurarse se incluya en la llamada los paréntesis (o llaves en el caso de vectores) para obtener una matriz de resultados, como por ejemplo con las siguientes líneas.

```
Cargas = getLoadInfo(DSSCircObj);
% Llamar sin paréntesis devuelve los kW de cada carga de forma separada
Cargas.kW
% Frente a llamar con paréntesis , que devuelve todos los kW en una matriz
[Cargas.kW]
% Lo mismo es válido para vectores
{Loads.name}
```

Este uso de Matlab es muy útil para filtrar resultados en base a ciertos criterios. Por ejemplo, se pueden filtrar las estructuras de datos de sólo las cargas de tres fases o las cargas por debajo de un determinado voltaje.

```
% Filtro para cargas con tres fases únicamente
Cargas_tres_fases = Loads([Loads.numPhases]==3);
% Filtro para cargas de bajo voltaje del sistema secundario
Cargas_secundarias = Cargas([Cargas.kV]<=0.24);
```

4.2.3 Ejecución de la función de comprobación del circuito.

Una herramienta particularmente útil al comienzo de cualquier análisis de un circuito es la función *circuitCheck*. Esta función examinará el circuito de OpenDSS por si existen errores tipográficos o inconsistencias potenciales que puedan conducir a una solución que pueda resultar extraña para OpenDSS.

Es útil para solucionar errores desde Matlab y para detectar anomalías en el código que están creando errores aparentes aunque permiten que el código se compile. Se recomienda utilizar esta función antes de comenzar a trabajar con cualquier circuito.

La función *circuitCheck* comprueba numerosos errores del modelo del circuito. Un ejemplo puede ser el configurar una carga con un tamaño demasiado elevado, este hecho permitiría compilar el código perfectamente, pero sería un error de modelado que ocasionaría la sobrecarga de un transformador. En este ejemplo, la comprobación dará el nombre del transformador sobrecargado para que el código puede ser corregido adecuadamente.

Otro ejemplo puede ser tener una línea con un número de fases que no corresponde a las fases del elemento al que está conectado. Esto generaría un error en la solución de OpenDSS y no sería compilado el código, pudiendo ser comprobado previamente con la función *getLineInfo* que avisaría del error. Sin embargo, la ventaja de *circuitCheck* es que tiene programadas todo este tipo de comprobaciones de forma automática y devolverá el error, identificando la causa y el nombre del elemento u objeto que debe ser corregido.

Para ejecutar manualmente *circuitCheck*, primero se debe compilar y resolver el circuito. Entonces, incluir el puntero COM para los objetos y, opcionalmente, la opción de advertencias de la función *circuitCheck*. Esta opción de advertencia viene por defecto activada, pero se puede desactivar como se indica a continuación.

```
warnSt = circuitCheck (DSSCircObj, 'Warnings', 'off');
```

```
warnSt = circuitCheck (DSSCircObj);
```

La función *circuitCheck* es llamada automáticamente en cualquiera de las funciones *get* si se detecta un error. Si hay problemas importantes en el circuito, OpenDSS puede no ser capaz de devolver una solución del flujo de potencia válido para el circuito. Entonces, *circuitCheck* comprobará varios errores potenciales que tiene programados, aunque no todos los casos posibles pueden ser detectado y analizados.

Si las advertencias están activadas, cualquier problema con el circuito se mostrará en la ventana de comandos.

Independientemente de si las advertencias se encuentran activas o no, al ejecutar la función *circuitCheck*, ésta siempre tendrá como salida una estructura con los errores, aunque no se muestre directamente en pantalla.

Cada elemento de la estructura corresponde a un solo aviso y contendrá cierta información que describe la advertencia, así como una lista de los elementos que provocan ese error. El usuario tendrá que inspeccionar la salida para determinar cuáles de las advertencias son en realidad errores en el circuito.

Las advertencias que pueden mostrarse se enumeran a continuación:

- warnSt.IsolatedElem

- Objetivo: clasifica a los elementos que no están conectados a la red como *aislados*.

- warnSt.str: "There are n isolated elements, m of which are enabled and without voltage."

- Explicación: Esto no tiene por qué ser un error per se, puede haber elementos aislados o desconectados de la red intencionadamente. En ese caso de que sea intencionado, es aconsejable deshabilitarlos. Hay tres categorías de elementos *aislados*:

1. "Isolated& Enabled without Voltage". Son elementos aislados que no están deshabilitados pero tampoco están conectados a la red.

2. "Isolated& Disabled". Son elementos que OpenDSS devuelve como aislados a pesar de que estén deshabilitados.

3. "Isolated& Enabled with Voltage". Elementos que tienen tensión, generalmente forman parte de una red "isla", es decir, que no está conectada a la red principal.

- warnSt.offenders: Esta tabla incluye una columna para cada categoría mencionada, además de otra columna que contiene todos los elementos aislados.

- warnSt.IsolatedNodes

- Objetivo: Para comprobar nodos aislados de la red.

- warnSt.str: "There are n isolated nodes in the circuit. Investigate the node by using DSSText.command='show busflow BUSNAME kVA elem'; where BUSNAME is the name of the isolated node without the decimal phasing."

- Explicación: Esto detecta cualquier nodo que puede estar aislado de la red. Un ejemplo de esto puede ser una carga de tres fases conectada a una línea de una sola fase: dos de los nodos (creados por la carga) estarían aislados con un voltaje desconocido.

- warnSt.offenders: cada fila de la tabla incluye el nombre del nodo aislado.

- warnSt.Loops

- Objetivo: Comprobar si existe algún bucle en el circuito.

- warnSt.str: "There are n loops in the circuit. The loops can also be viewed using DSSText.command='show loops';"

- Explicación: Generalmente los alimentadores son diseñados de forma radial. Esto devuelve el resultado de un algoritmo de OpenDSS para detectar bucles.

- warnSt.offenders: Cada fila de la tabla incluye la lista de elementos que forman parte de un bucle.

- warnSt.InvalidLineBusName

- Objetivo: Comprueba que la nomenclatura de un bus para líneas específicas coincide con las fases de esa línea.

- warnSt.str: "One or more line has a bus name that does not match the number of phases of the line. (e.g. A 2-phase line should have both bus 1 and bus 2 with names similar to 'BUSNAME.2.3' with 2 phases indicated in the decimal notation."

- Explicación: La herramienta usa esta nomenclatura para ayudar a determinar las fases que tiene una determinada línea. El número de fases en esa línea debe coincidir con el número de fases del bus al que está conectada.

- warnSt.offenders: Cada fila de la tabla incluye el nombre de la línea (*LineName*), el número de fases (*NumPhases*) y el nombre de cada bus. A partir de esto, debe ser obvio qué parte de la definición de las líneas está causando los errores.

- warnSt.NoBusCoords

- Objetivo: Comprueba la presencia de las coordenadas del bus.

- warnSt.str: "There are no bus coordinates with this compiled circuit. Toolbox functionality will be severely limited."

- Explicación: La herramienta usa las coordenadas de los buses para realizar las gráficas del circuito así como la integración solar para algunas simulaciones.

- warnSt.offenders: n/a.

- warnSt.MissingBusCoords
 - Objetivo: Comprueba para buses de medio voltaje (por debajo de 600V) que han perdido coordenadas.
 - warnSt.str: "There are n buses above 600V that are missing coordinates."
 - Explicación: En general, cualquier objeto que está conectado al bus sin coordenadas, no será mostrado en las gráficas.
 - warnSt.offenders: Lista de buses sin coordenadas.

- warnSt.LineLength
 - Objetivo: Comprueba si hay líneas con largas longitudes sin sentido en el circuito.
 - Límites: 5 km.
 - warnSt.str: "n of the lines exceed 5 km"
 - Explicación: Introducir accidentalmente largas longitudes puede generar resultados erróneos e irregularidades en las simulaciones de flujo de potencia.
 - warnSt.offenders: Nombre de la línea y longitud.

- warnSt.LineOverLoading
 - Objetivo: Comprueba si hay violaciones térmicas en las líneas.
 - Límites: 100% .
 - warnSt.str: "n Lines are load more than 100%. Visualize using `plotCircuitLines(DSSCircObj, 'Coloring', 'lineLoading')`"
 - Explicación: Advierte que una línea está sobrecargada que puede ser consecuencia de una mala configuración de los parámetros en el circuito o en las líneas.
 - warnSt.offenders: Nombre de la línea y su porcentaje de carga.

- warnSt.BusDistance
 - Objetivo: Comprueba si se han introducido incorrectamente líneas que están excesivamente lejos de las líneas.
 - Límites: 25 km.
 - warnSt.str: "n of the bus distances exceeds 25 km from the substation. "
 - Explicación: Introducir parámetros del circuito de forma errónea accidentalmente, así como la longitud de la línea, puede llevar a que un bus esté excesivamente lejos de una subestación.
 - warnSt.offenders: Nombre de los buses y distancia.

- warnSt.CapacitorRatingMismatch
 - Objetivo: Comprueba si se han introducido accidentalmente datos de kV poco consistentes.
 - Límites: 5%.

-
- warnSt.str: "n of the capacitor kV ratings differs from its bus kV rating by more than 5%. "
 - Explicación: Introducir incorrectamente datos de tensiones pueden causar irregularidades en la solución sin dar un error en la simulación. Lo más probable es que este problema se deba a que, en una fase, no se introdujo correctamente la tensión de línea a neutro; o para dos o tres fases, no se introdujo el valor línea a línea.
 - warnSt.offenders: Nombre de cada elemento y su tensión línea a línea, así como el nombre de todos los buses y su tensión línea a línea.
-
- warnSt.LoadRatingMismatch
 - Objetivo: *idem* al anterior.
 - Límites: 5%.
 - *idem* al anterior.
 - Explicación: *idem* al anterior.
 - warnSt.offenders: *idem* al anterior.
-
- warnSt.GeneratorRatingMismatch
 - Objetivo: *idem* al anterior.
 - Límites: 5%.
 - *idem* al anterior.
 - Explicación: *idem* al anterior.
 - warnSt.offenders: *idem* al anterior.
-
- warnSt.PVRatingMismatch
 - Objetivo: *idem* al anterior.
 - Límites: 5%.
 - *idem* al anterior.
 - Explicación: *idem* al anterior.
 - warnSt.offenders: *idem* al anterior.
-
- warnSt.TransformerRatingMismatch
 - Objetivo: *idem* al anterior.
 - Límites: 5%.
 - *idem* al anterior.
 - Explicación: *idem* al anterior.
 - warnSt.offenders: *idem* al anterior.

- warnSt.TransformerOverloaded

- Objetivo: Comprueba si hay violaciones térmicas en las líneas.
- Límites: 5% .
- warnSt.str: "n of the transformer kVA ratings differs from its bus1 power by more than 5%. Check that the loads on the transformer are entered correctly."
- Explicación: Advierte que un transformador está sobrecargado, pudiendo ser consecuencia de una mala configuración de los parámetros en el circuito.
- warnSt.offenders: Nombre del transformador y su porcentaje de carga.

- warnSt.TransformerNoLoad

- Objetivo: Comprueba si hay transformadores con ninguna carga aguas abajo.
- Límites: flujos de potencia en el transformador inferiores al 1% del ratio del transformador y ninguna carga aguas abajo.
- warnSt.str: "n of the transformer have no load on them. Check that the loads on that transformer are entered correctly."
- Explicación: Detecta cualquier problema durante la asignación de cargas cuando cargas no están asignadas a un transformador en servicio.
- warnSt.offenders: Nombre del transformador y sus ratios de kVA.

- warnSt.TransformerLowLoad

- Objetivo: Comprueba si hay transformadores con bajo flujo de potencia en comparación a su ratio de funcionamiento.
- Límites: flujos de potencia en el transformador inferiores al 1% del ratio del transformador y ninguna carga aguas abajo.
- warnSt.str: "n of the transformer have less than 1% power flow of their kVA rating. Check that the loads on that transformer are entered correctly."
- Explicación: Detecta cualquier problema durante la asignación de cargas cuando cargas no están asignadas a un transformador en servicio.
- warnSt.offenders: Nombre del transformador, sus ratios de kVA, suma de kW para todas las cargas instaladas aguas abajo y la lista de cargas aguas abajo.

- warnSt.BusVoltage

- Objetivo: Comprueba si hay violación de tensiones (sobrevoltaje o déficit de voltaje).
- Límites: 1_0:05 pu.
- warnSt.str: "n of the enabled bus voltages are outside of the range 1_0:05 pu. Visualize using *plotVoltageProfile(DSSCircObj)*"
- Explicación: Advierte sobre violaciones de voltaje que pueden ser causa de una incorrecta configuración de parámetros y pueden provocar grandes cambios de tensión.
- warnSt.offenders: Nombre del bus y tensión (pu y kV).

- warnSt.LineRatingMismatch

- Objetivo: Comprueba si se han introducido accidentalmente datos incorrectos en los códigos de línea.
- Límites: 150%.
- warnSt.str: "n of the line ratings are 150% the size of the immediately upstream line. Visualize using `plotCircuitLines(DSSCircObj, 'Thickness', 'lineRating')`"
- Explicación: Los ratios de las líneas que se incrementan aguas abajo *pueden* ser indicativo de incorrectos parámetros de línea (o puede ser por el diseño).
- warnSt.offenders: Nombre de la línea aguas arriba (la más pequeña) y el nombre de la línea aguas abajo (más grande), seguida por ratio de cada línea y su código de línea.

4.2.4 Gráficas e interpretación de resultados.

La generación de gráficas es sencilla y además hay varias formas alternativas de conseguirlo. Las gráficas son útiles y representativas de la solución del flujo de potencia en la simulación más reciente. En caso de querer conocer la solución gráfica en un determinado instante de tiempo, habrá que configurar la simulación para que ese sea el último instante resuelto por OpenDSS. Veamos en un ejemplo dicha configuración.

```
DSSText.command = 'Set mode=duty number=10 hour=13 h=1 sec=1800';
DSSText.command = 'Set controlmode = static ' ;
DSSText.command = 'solve';
figure ; plotCircuitLinesOptions (DSSCircObj);
```

Llamando a `plotCircuitLines` sin asignar ninguna propiedad, será llamada por defecto la interfaz gráfica de OpenDSS (GUI) para lanzar la función `plotCircuitLinesOptions`.

Las gráficas obtenidas serán generadas con OpenDSS, pero añadiendo el comando de Matlab `figure` antes de hacer la llamada a `plotCircuitLines`, se abrirá en el entorno gráfico de Matlab.

La función `plotCircuitLinesOptions` no acepta más parámetros, pero sí es posible llamar a la función `plotCircuitLines` con un determinado número de parámetros:

```
figure;
plotCircuitLines(DSSCircObj, 'Coloring', 'PerPhase', 'Thickness', 3, 'MappingBackground', 'hybrid');
```

Las funciones usadas para la creación de gráficas, utilizan el nombre de los parámetros de Matlab. Los argumentos que se le dan, después de la llamada a `DSSCircObj`, van emparejados. Esto es, aunque no importa el orden de las especificaciones dadas a la función, cada opción requiere dos entradas: el nombre de la configuración que se está modificando y la especificación de este cambio.

Por ejemplo, en la línea de código anterior, el parámetro `'Coloring'` es configurado a `'PerPhase'` y la propiedad `'Thickness'` es configurada a 3.

Para concluir el apartado de generación de gráficas, quiero subrayar que el uso de Matlab abre un abanico de opciones alternativas al uso del propio OpenDSS como aplicación independiente. Es posible, por ejemplo, extraer los valores de objetos o variables en cada salto de iteración, guardarlos en un vector y posteriormente representarlos con las herramientas propias de Matlab.

Para concluir la sección dedicada al Control de OpenDSS desde Matlab, se incluye a continuación un ejemplo completo pero sencillo de cómo se puede compilar y ejecutar un circuito mediante esta opción.

```

clc ; clear all ; close all ;
%% ARRANQUE DE OPENDSS. Activación del COM Server y comunicación con ActiveX de Matlab.
% Se inicia el COM Server de OpenDSS. Matlab utiliza su servidor ActiveX
% integrado para comunicarse con el COM Server de OpenDSS, de esta manera
% dicho servidor será la interfaz entre ambos programas.
DSSObj = actxserver( 'OpenDSSEngine.DSS');
% Se inicia la DSS. (Sólo necesario la primera vez que se inicia Matlab).
DSSObj.Start(0) % Devolverá un 1 si OpenDSS está disponible y funcionando.
DSSText = DSSObj.Text; % Se define la interfaz de texto . Con este comando se introducirán las instrucciones a
OpenDSS %desde Matlab
DSSCircuit = DSSObj.ActiveCircuit; % Se define la interfaz del circuito .
DSSSolution = DSSCircuit.Solution ; % Se define la interfaz de la solución.
%% Compilación del script del circuito leee37 bus desde el directorio donde está guardado
DSSText.Command = 'Compile "C:\example\leee37\ieee37.dss"';
%% Algunas instrucciones para OpenDSS
DSSText.Command = 'Show Losses' % Genera un archivo con las pérdidas de potencia
DSSText.Command = 'Show Powers' % Genera un archivo con la potencia de sobrecarga de cada elemento del
circuito .
DSSText.Command = 'Show Voltages LN' % Comando que genera un archivo que contiene las tensiones simples de
cada %bus en el sistema
% Exportar voltajes a un archivo de excel
DSSText.Command = 'Export Voltages';
Filename = DSSText.Result;
%% Solución de circuito
DSSSolution.Solve() ;
if DSSSolution.Converged,
disp( 'El circuito ha sido resuelto satisfactoriamente ' )
end

```

4.2.5 Ejemplo resuelto de red de distribución IEEE13.

A continuación se explica a modo de ejemplo y se listan los códigos utilizados, para la resolución con Matlab y OpenDSS de la red tipo denominada *IEEE13 Node Test Feeder*, que está incluida entre los ejemplos facilitados por el organismo IEEE (*The Institute of Electrical and Electronic Engineers, Inc*).

Este circuito está resuelto en el Trabajo Fin de Grado referenciado [15], y de él se han extraído tanto los códigos como el procedimiento para generarlos.

En este circuito, representado en la figura 4-6, podemos diferenciar:

- Bus 650
- Diferentes nodos 645, 652, 680, etc.
- Un regulador entre el nodo 632 y el bus 650 (a efectos del OpenDSS se tratará como un transformador especial)

- Un transformador entre los nodos 633 y 634
- Un interruptor entre los nodos 671 y 692
- Diferentes líneas entre los distintos nodos

Aunque en este caso y habitualmente en los esquemas del IEEE no suelen estar representados, hay un transformador con la fuente, que viene a separar nuestro circuito de la red eléctrica.

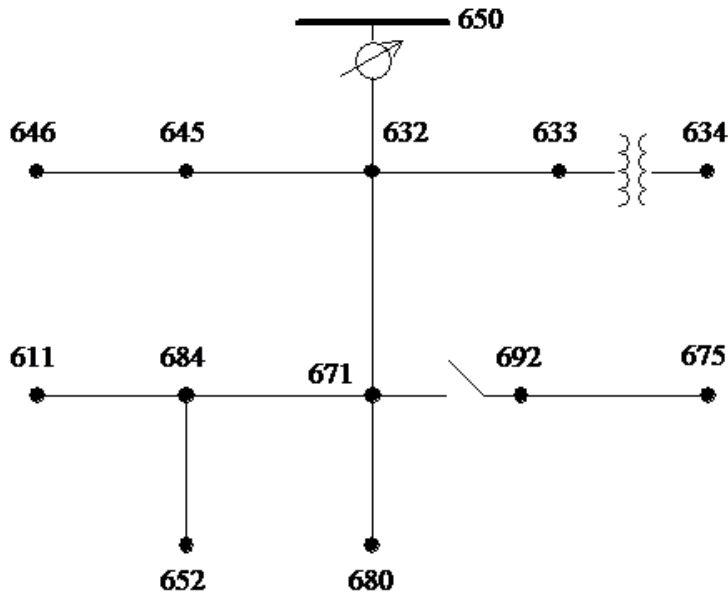


Figura 4-6. Esquema del IEEE13 Node Test Feeder.

A continuación se propone, en aras de mantener un orden lógico, la siguiente estructura de programación. Se usarán aquellos objetos en OpenDSS que ya han sido definidos en la primera sección de este capítulo, dedicada a la programación de este software como aplicación independiente.

1). Datos básicos del circuito. Introduciremos:

- el nombre del archivo '.dss'
- el voltaje base que nos llega a nuestro circuito desde la red eléctrica
- la relación entre las cantidades reales y las cantidades básicas (*pu* o *per unit*)
- $MVAsc3$, que se define como , para las 3 fases
- $MVAsc1$, que se define como , donde . Debe de ser de la misma magnitud que $MVAsc3$.

Esta notación es una forma de definir las impedancias de la fuente. También podríamos usar directamente las corrientes de cortocircuito mediante $ISC1$ e $ISC3$.

2). Transformador principal. Lo definiremos como:

- Nombre del transformador
- Número de fases del transformador
- Número de bobinas
- Relación entre las reactancias del devanado 1 al 2 (XHL), 2 al 3 (XLT) o 1 al 3 (XHT)
- A continuación se definen las características de cada devanado, p.ej. $wdg = 1$ si se trata del devanado 1.
 - Se introduce en qué nodo o bus está conectado cada devanado '*bus*'.
 - Configuración de conexión (estrella o triángulo) mediante '*conn*'.

- Valores nominales para cada devanado de kV y kVA.
- Porcentaje de la resistencia del devanado en base a los kVA del devanado (%r).

3). Otros transformadores del circuito. Se definen exactamente igual que el transformador principal. La única diferencia radica en la conexión de cada devanado a cada bus o nodo, lo que hará localizar cada transformador en una zona del circuito diferente.

4). Tipos de líneas que nos encontraremos en el circuito. Esto dependerá del tipo de conductor que puede haber entre nodos o buses y sus características, como sus propias impedancias entre otros.

- Nombre de la línea, número de fases y frecuencia
- Matriz r , que corresponde a la matriz de resistencias (por fases) del conductor.
- Matriz x , que corresponde a la matriz de reactancias (por fases) del conductor. Tanto la matriz de reactancias como la de resistencias estarán en ohmios/unidad longitud.
- Matriz c , matriz de capacitancia (capacidad para almacenar carga y energía potencial eléctrica) en nanofaradios/unidad longitud.

Podemos introducir aquí '*Normamps*' o '*Emergamp*' que nos indicarían las corrientes máxima y de emergencia de cada tipo de línea.

5). Líneas existentes en el circuito.

Normalmente las líneas corresponden con la unión de los nodos. Se definen por:

- Nombre de la línea
- Número de fases
- Buses entre la que se encuentra (Bus1= Bus2=). Además debemos de indicar el tipo de conexión con cada una de las fases. Por defecto y con una conexión estándar, será '1.2.3'.
- Código de línea, es decir, el tipo de línea al que corresponde. El tipo de línea debe corresponder con algunas de las clases de líneas definidas en el apartado anterior
- Longitud

6). Regulador. Este control se usa para emular una herramienta reguladora del voltaje o control LTC (Load Tap Changing Control) que controla y ajusta el arrollamiento de los devanados del transformador. Es necesario caracterizarlo mediante los siguientes pasos:

- Definir un transformador como ya hemos visto anteriormente
- Definimos el nombre del regulador y a qué transformador se asocia, junto con las siguientes características:
 - a). Devanados: indica el número de devanados que controla el regulador. Por defecto 1 o 2.
 - b). V_{reg} : define los ajustes del voltaje del regulador para los devanados que se controlarán. El valor por defecto es 120 V.
 - c). Band: ancho de banda para el bus controlado. Por defecto 3 V.
 - d). P_{ratio} : relación de la PT o relación entre la tensión del bobinado controlado, y la tensión V_{reg} del regulador. Valor por defecto 60.
 - e). C_{prim} : relación, en amperios, entre bobinado primario y secundario para la conversión en la línea. Por defecto el amperaje en el secundario es de 0,2 amperios.
 - f). R y X: ajustes para compensar el cambio de voltaje en la línea

7). Cargas puntuales en los diferentes nodos. Se definirán como:

- Nombre de la carga
- Bus o nodo donde se encuentre
- Fases donde se encuentre la carga

- Configuración de la carga

- Modelo de la carga. Es una variable que define cómo variará la carga en función del voltaje. Valdrá 1 si la potencia activa (P) y la reactiva (Q) son constantes; 2 si la impedancia (Z) es constante; 3 si P es constante y Q cuadrática; 4 si P y Q son exponenciales; 5 si la corriente es constante, etc.

- kV, kVAR y kW

8). Definición de Generadores, para simular aerogeneradores, placas solares o cualquier elemento que aporte potencia al circuito. La estructura de programación es muy parecida a la de las cargas, estableciendo el nombre, el bus, la fase, la configuración, kV, kW o kVAR. Los dos principales parámetros son el modelo del generador (1 si inyecta kW=cte, 2 con admitancia=cte, 3 con kW y kV =c cte, etc) y el tipo de generador 'duty'.

Para asignar el tipo de generador 'duty' debemos usar el objeto llamado 'Loadshape' que se define a continuación.

9). Loadshape. Es la forma que tienen las cargas y se definen mediante un vector donde se incluyen valores o datos, uno para cada hora del día, por ejemplo.

Para las cargas, es necesario escribir al final de cada línea correspondiente a cada carga, su definición mediante 'Yearly= nombre_loadshape_año' o 'Daily=nombre_loadshape_día'. Todo lo anterior también se puede definir para los generadores. Introducimos el vector *loadshape* mediante un archivo .csv de Excel, que debe estar incluido en el mismo directorio que el archivo .dss utilizado para resolver y ejecutar el circuito con la aplicación OpenDSS.

10). Medidores de Energía, para controlar y poder ver de forma acumulada, los parámetros que nos interesen en determinadas zonas de nuestra red. Este tipo de medidores tienen gran complejidad a la hora de establecer las zonas de medida y puntos de partida.

11). Monitores, que representarán cómo evolucionan los parámetros a estudiar de los distintos elementos en función del tiempo. Debemos elegir el modo del monitor, siendo 1 para que nos muestre el voltaje y la intensidad o 2 para la potencia, por ejemplo.

12). Introducción del voltaje base que tomará el programa y le indicamos los resultados que queremos obtener.

En el código siguiente vamos a decirle que represente el circuito y las potencias en todos los elementos. Para ello le habremos introducido, al igual que en punto 9, un archivo .csv de Excel con las coordenadas de todos los buses / nodos del circuito.

Por el mismo orden establecido anteriormente, se lista ahora el código para resolver el circuito usando la interfaz gráfica de OpenDSS.

1.-

clear

new circuit.IEEE13 basekv=115 pu=1 MVAsc1=21000 MVAsc3=20000

2.-

new transformer.Sub phases=3 windings=2 Xhl=4

~wdg=1 bus=sourcebus conn=delta kv=115 kva=5000 %r=0.05 XHT=4

~wdg=2 bus=650 conn=wye kv=4.16 kva=5000 %r=0.05 XLT=4

3.-

new transformer.XFM1 phases=3 windings=2 XHL=2

~wdg=1 bus=633 conn=wye kv=4.16 kva=500 %r=0.55 XHT=1

~wdg=2 bus=634 conn=wye kv=0.48 kva=500 %r=0.55 XLT=1

4.-

new linecode.336matrix nphases=3 BaseFreq=60

~ rmatrix=(0.0868455 | 0.0298305 0.0887966 | 0.0288883 0.0298305 0.0868455)

~ xmatrix=(0.2025449 | 0.0847210 0.1961452 | 0.0719161 0.0847210 0.2025449)

~ cmatrix=(2.74 | 0.70 2.96| 0.34 0.71 2.74)

~ Normamps = 400 Emergamps=600

!nota: las unidades están dadas en ohmios por 100 ft, y en nanofaradios por 100ft

!(datos procedentes del IEEE)

5.-

new line.650632 Phases=3 Bus1=SOB.1.2.3 Bus2=632.1.2.3 linecode=336matx length=2000 units=ft

new line.632670 Phases=3 Bus1=632.1.2.3 Bus2=670.1.2.3 linecode=336matx length=667 units=ft

new line.670671 Phases=3 Bus1=670.1.2.3 Bus2=671.1.2.3 linecode=336matx length=1333 units=ft

new line.671680 Phases=3 Bus1=671.1.2.3 Bus2=680.1.2.3 linecode=336matx length=1000 units=ft

new line.632633 Phases=3 Bus1=632.1.2.3 Bus2=633.1.2.3 linecode=336matx length=500 units=ft

new line.632645 Phases=2 Bus1=632.3.2 Bus2=645.3.2 linecode=336matx length=500 units=ft

new line.645646 Phases=2 Bus1=645.3.2 Bus2=646.3.2 linecode=336matx length=300 units=ft

new line.692675 Phases=3 Bus1=692.1.2.3 Bus2=675.1.2.3 linecode=336matx length=500 units=ft

new line.671684 Phases=2 Bus1=671.1.3 Bus2=684.1.3 linecode=336matx length=300 units=ft

new line.684611 Phases=1 Bus1=684.3 Bus2=611.3 linecode=336matx length=300 units=ft

new line.684652 Phases=1 Bus1=684.1 Bus2=652.1 linecode=336matx length=800 units=ft

!nota: por sencillez se supone que sólo existe un tipo de línea,

¡que fue definida anteriormente como 336matx (por su matriz de impedancias)

¡en los buses se define el orden de conexión de las fases, p.ej. 1.2.3

6.-

new transformer.reg1a phases=1 windings=2 buses=(650.1.2 650r.1.2) conns='delta delta'

~kvs="4.16 4.16" kvas="2000 2000" XHL=1

new regcontrol.creg1a transformer=reg1a winding=2 vreg=122 band=2

~ptratio=40 cprim=350 R=-0.201 X=3.348

new transformer.reg1c like=reg1a buses=(650.3.2 650r.3.2)

new regcontrol.creg1c like=reg1a transformer=reg1c R=2.799 X=1.848

!nota: se definen dos reguladores porque no se actúa en todas las fases de la línea.

7.-

New Load.S634a Bus1=634.1.2 Phases=1 Conn=Wye Model=1 kV=0.48 kW=160.0 kVAR= 110.0

New Load.S634b Bus1=634.2.3 Phases=1 Conn=Wye Model=1 kV=0.48 kW=120.0 kVAR=90.0

New Load.S634c Bus1=634.3.1 Phases=1 Conn=Wye Model=1 kV=0.48 kW=120.0 kVAR=90.0

New Load.S645b Bus1=645.2.3 Phases=1 Conn=Wye Model=1 kV=4.16 kW=170.0 kVAR= 125.0

New Load.S646b Bus1=646.2.3 Phases=1 Conn=Delta Model=2 kV=4.16 kW= 230.0 kVAR= 132.0

New Load.S652a Bus1=652.1.2 Phases=1 Conn=Wye Model=2 kV=4.16 kW= 128.0 kVAR= 86.0

New Load.S671 Bus1=671 Phases=1 Conn=Delta Model=1 kV=4.16 kW= 385.0 kVAR= 220.0

New Load.S675a Bus1=675.1.2 Phases=1 Conn=Wye Model=1 kV=4.16 kW=485.0 kVAR= 190.0

New Load.S675b Bus1=675.2.3 Phases=1 Conn=Wye Model=1 kV=4.16 kW=68.0 kVAR= 60.0

New Load.S675c Bus1=675.3.1 Phases=1 Conn=Wye Model=1 kV=4.16 kW=290.0 kVAR= 212.0

New Load.S692c Bus1=692.3.1 Phases=1 Conn=Delta Model=5 kV=4.16 kW=170.0 kVAR=151.0

New Load.S611c Bus1=611.3.1 Phases=1 Conn=Wye Model=5 kV=4.16 kW=170.0 kVAR= 80.0

pasos 8.- y 9.-

```

; loadshape debe estar en realidad al principio del código; en todo caso antes de definir el generador
new loadshape.wind 24 1.0
~mult=(0 0 0.1 0 0.1 0 0.2 0.4 0.7 0.7 1.3 1.3 1.8 2.3 2.7 0.7 0.4 0.4 0.2 0.2 0.2 0.1 0.1 0)
~action=normalize
!nota: con el comando normalize escalamos los multiplicadores para que el máximo sea 1
New generator.genwind Bus1=633 kV=4.16 kW=8000 pf=1 conn=wye duty=wind Model=1
;nota: generador eólico de 8 MW
10.-
new energymeter.em1 element=line.650632 1 action=Save
11.-
new monitor.mw1 element=generator.genwind 1 mode=1 ppolar=no
12.-
Set voltagebases ="115 4.16 0.48"
Calcvoltagebases
solve
solve mode=daily trapezoidal=no
show mon mw1
;nota: trozo de código alternativo para representar el circuito
solve mode=direct
set maxiterations=100
solve
Plot Type=Circuit Quantity=Losses Dots=Y Labels=Y
show powers kva elements

```

4.3 Resolución mediante OpenDSS de la microrred INTA de 11 nodos

4.3.1 Hipótesis iniciales.

El modelado de una microrred completa con los diferentes elementos que la integran, para su funcionamiento en una situación real, aparece ampliamente descrito en el artículo de referencia en la bibliografía [9], y que ha servido como consulta para este proyecto. También ha servido en gran medida para resolver las hipótesis necesarias para el funcionamiento en una instalación real, que como ya se ha dicho es el objeto de este trabajo.

Visto el esquema general de la red INTA de 11 nodos, hay que señalar que es un sistema de múltiples frecuencias (coexisten elementos en AC y en DC), y el software OpenDSS no las puede resolver todas al mismo tiempo. Por tanto, como trabajamos con dos frecuencias, habría que ir resolviendo simultáneamente dos problemas con OpenDSS. Una forma de hacerlo puede ser, por ejemplo, resolver en cada salto de tiempo el circuito DC, y usar la solución como condición inicial o dato para la resolución simultánea del circuito AC.

Otra consideración a tener en cuenta es que el software de OpenDSS no ha desarrollado la simulación de convertidores DC/DC. Esta limitación se puede resolver usando una secuencia de equipos DC/AC – AC/AC – AC/DC, para lo cual habría que resolver simultáneamente dos frecuencias, como se ha explicado en el párrafo anterior.

Otra opción posible es modelar junto con cada equipo que funciona en DC, el inversor correspondiente, de forma que la red completa se simula en AC. Esta alternativa tiene el problema de que se desprecian las distancias de líneas en corriente continua que unan el inversor y el equipo que se modela, pero puede valer si somos capaces de acotar dicho error. Esta es la situación en la que a juicio del autor, nos encontramos en este proyecto, y por tanto la alternativa elegida para modelar.

Finalmente, a la hora de programar la simulación en OpenDSS, todos los elementos del circuito se definen en corriente alterna monofásica a 50 Hz, quedando los diferentes elementos que funcionen en DC integrados en el modelo concreto de cada elemento, habiendo tenido en cuenta ya las previsible pérdidas en dicho modelo.

4.3.2 Descripción de la microrred.

Para describir la instalación de la manera más simplificada posible voy a usar el mismo esquema que ya sirvió para explicar la optimización por control predictivo, figura 4-7.

Hasta ahora no se ha detallado en este trabajo el funcionamiento del circuito de hidrógeno, por lo que se hace a continuación. Este circuito consta básicamente de tres elementos que aparecen representados en la figura: electrolizador (nodo 675), pila de combustible (nodo 634) y depósito de hidruros metálicos (MHL, por sus siglas en inglés).

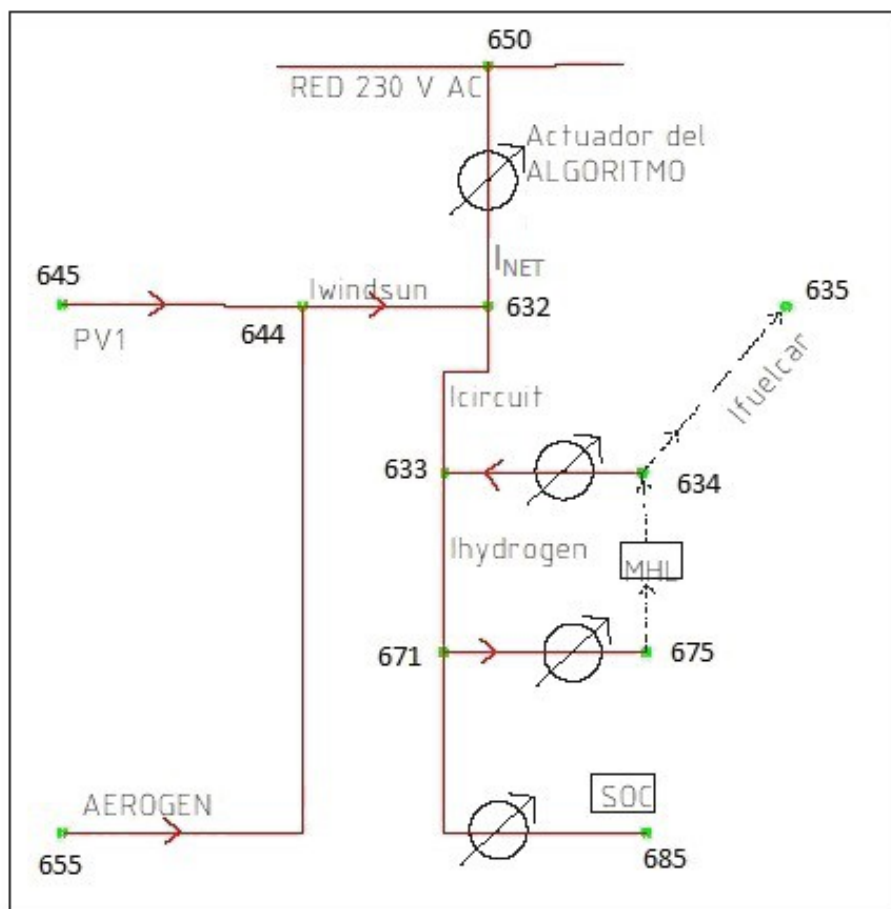


Figura 4-7. Relación entre variables del modelo.

El electrolizador utiliza energía sobrante en el BUS de la microrred, para fabricar hidrógeno según el proceso descrito en la subsección siguiente. Esta energía en forma de hidrógeno (más concretamente en forma de hidruros metálicos a alta presión) se almacena para su utilización cuando más convenga, bien para devolverla al BUS de la microrred o bien para suministrarla a un automóvil híbrido que está en repostaje.

El proceso inverso a la electrólisis permite obtener la energía eléctrica mediante la pila de combustible, que comúnmente aparece en manuales denominada '*fuel cell*'.

4.3.3 Modelado del proceso de electrólisis y su reacción inversa.

A continuación voy a realizar una breve descripción de la reacción química de la electrólisis para producir H_2 , y de la reacción inversa para producir electricidad.

Electrolizador: $2\text{H}_2\text{O (líquida)} + e^- \rightarrow \text{H}_2 + \frac{1}{2} \text{O}_2$

Pila de combustible: $\text{H}_2 + \frac{1}{2} \text{O}_2 \rightarrow 2\text{H}_2\text{O (vapor)} + e^-$

El electrolizador utiliza el principio de electrólisis para separar el hidrógeno y el agua, aplicando una corriente eléctrica muy alta que es forzada a atravesar un electrolito. Para ello se disponen dos electrodos (ánodo y cátodo) y entre ellos se aplica la correspondiente diferencia de potencial. Esto se hace por celdas, sumadas en serie o en paralelo dependiendo de las tensiones de trabajo.

La pila de combustible es un reactor; basta poner en contacto los dos reactivos en las condiciones termodinámicas adecuadas, y se producirá la reacción. Dichas condiciones requieren un aporte de calor externo, y el depósito de hidruros metálicos también necesita dotarse de resistencias ya que el hidrógeno almacenado en forma de hidruros necesita calor para liberarse. Estos aportes energéticos suponen una pérdida de eficiencia global del proceso.

Las pérdidas citadas en el párrafo anterior están modeladas en el software OpenDSS para los equipos electrolizador y pila, que se sumarán a las de tipo electromagnético ocurridas en los convertidores DC-DC.

Veamos la relación entre la intensidad y el flujo de hidrógeno generado / consumido respectivamente, que viene dada por:

$$\text{Electrolizador: } \hat{m}_{elect} = \frac{I_{elect} n_z}{F}; \quad \text{Pila: } \hat{m}_{fc} = \frac{I_{fc}}{2 A_{fc} F} \quad (1-4)$$

siendo $F = 9.6496e4$ la constante de Faraday, que es la cantidad de carga eléctrica presente en un mol de sustancia. El físico experimental Michael Faraday consiguió establecer una ley que relaciona la cantidad (en gramos) de sustancia producida a partir de una reacción de electrólisis en un tiempo (medido en segundos). El resto de parámetros de la reacción vienen simplificados en las ecuaciones (1-4) mediante los valores de n_z y A_{fc} .

Dado el valor (experimental) dado por Faraday para su constante, obtenemos el flujo de hidrógeno producido en unidades de litros/s.

Los valores A_{fc} , n_z son la superficie efectiva de intercambio en la pila de combustible y el número de celdas en el electrolizador, respectivamente.

Con los datos disponibles para nuestra planta (suponemos de las mismas características que las de los equipos presentados en la publicación de referencia [2]) tenemos:

$$A_{fc} = 1.4826 \text{ m}^2$$

$$n_z = 30$$

Estas constantes nos servirán para modelar en la simulación del circuito mediante OpenDSS y Matlab, la variable de estado MHL (sensible a los flujos de entrada/salida de hidrógeno) conocidas las variables de Intensidad consumida/generada en el ciclo de hidrógeno representado en la figura 4-7.

En nuestro algoritmo tenemos que programar la conversión de unidades, para obtener los flujos de salida/entrada de hidrógeno en Nm^3/h , ya que el almacenamiento lo tenemos aforado en Nm^3 , y los saltos o periodos de tiempo entre cada simulación tienen una duración de una hora. También manejaremos los flujos en unidades de $\text{Nm}^3/\text{minuto}$, ya que cada simulación en OpenDSS a su vez tiene programados los saltos en minutos.

A la vista de la figura 4-7, hay que comentar que aunque el nodo 635 se modela como una carga en el circuito AC, en realidad es una carga que se abastece con H_2 procedente del depósito.

Con los datos facilitados por personal técnico del INTA en sus instalaciones de El Arenosillo, Moguer (Huelva), se ha podido modelar el consumo de hidrógeno de los dos coches híbridos que forman parte de la microrred. Lamentablemente, tengo que mencionar que en las fechas en que escribo este trabajo, un grave incendio forestal en el Parque Natural ha quemado parte de esta planta incluyendo los dos vehículos.

- Hasta ese momento, la energía que consumían equivale a 12 kW h, recargando durante todo 1 día, cada 21 días.
- → equivale aproximadamente a cargar 1 h al día durante 21 días y un total de 12 kW h
- → o bien a cargar 12 h al día y un total de 1.2 kW h cada 2 días
- → es decir, 0.6 kW h al día, repartidos durante un tramo de 12 horas, que por lógica será el tramo horario fuera de la jornada laboral, por ejemplo **de 19:00 h hasta 07:00 h**.
- La potencia consumida (que llega al coche) en cada hora es de $P' = 0.6 / 12 = 0.05$ kW

Para calcular la potencia eléctrica consumida de la red, hay que incrementar las pérdidas en la producción de hidrógeno por el electrolizador, que como ya hemos visto en el capítulo 3 ascienden a 0.75 kW en condiciones nominales (ver tabla 3-13).

Por tanto, $P_{FUELCAR} = 0.8$ kW

El valor calculado anteriormente no es del todo realista pues supone que toda la potencia del electrolizador se está utilizando para generar hidrógeno que se va a emplear en abastecer al vehículo híbrido, y todas las pérdidas se imputan al coche. En cualquier caso lo podemos tomar como una cota superior del consumo.

En el modelo aproximado de la red que se va a resolver a continuación, se incluye un nodo 635 que representa una carga: como se comentó anteriormente en esta misma página, este punto es el consumo de hidrógeno para recarga del automóvil híbrido, y aunque no es una carga eléctrica se modela como tal, derivándola desde el electrolizador (en realidad deriva desde el depósito de hidruros metálicos, ver figura 1-2 al principio, y posteriores esquemas derivados de éste, a lo largo de este proyecto).

La distribución del consumo se resume en la siguiente gráfica, figura 4-8.

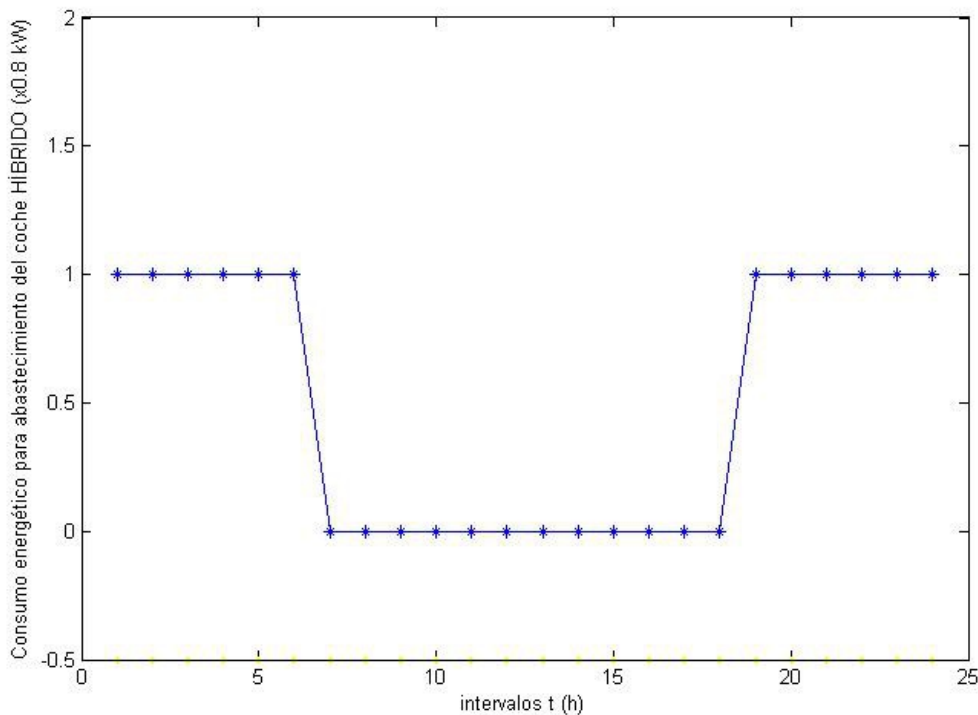


Figura 4-8. Serie temporal (24 h) de potencia (kW) consumidos por vehículo híbrido.

La pila de combustible se modela en OpenDSS como un generador de potencia pico 1.2 kW, con una curva de

funcionamiento tipo *loadshape* donde ya se tienen en cuenta las pérdidas en el convertidor.

Modelamos la potencia consumida por el electrolizador, para una carga de valor medio de funcionamiento de 0.62 veces la potencia pico (5 kW) definida en las características técnicas del electrolizador. Esto ya tiene en cuenta (según figura 4-9) las pérdidas en el conjunto electrolizador-convertidor DC/DC.

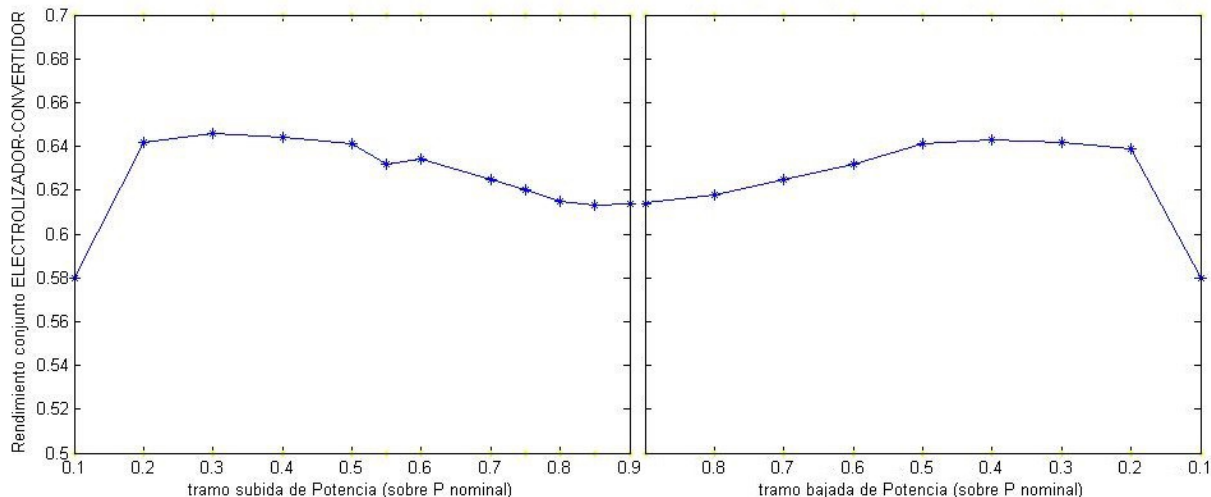


Figura 4-9. Modelo experimental electrolizador-convertidor DC/DC, fuente [12].

4.3.4 Modelado del proceso de carga descarga en las baterías Pb-ácido.

Es conveniente también hacer aquí una breve reseña del principio y funcionamiento de las baterías de acumuladores, mediante la reacción química de OXIDACIÓN-REDUCCIÓN que permite el intercambio de energía en forma de electrones, concretamente en las del tipo PLOMO-ÁCIDO que son las baterías usadas en esta red de generación distribuida.

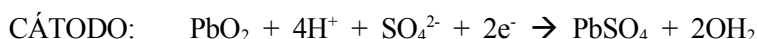
La batería de acumuladores es un conjunto de celdas estancas entre ellas, ya que están llenas de electrolito. Cada celda mantiene una tensión aproximada entre bornes de 2 v, por lo que se conectarán en serie tantas como sea necesario para obtener el voltaje nominal del conjunto de la batería. Para explicar el proceso de almacenamiento de energía eléctrica tomamos una sola celda.

Tenemos un conjunto de placas de plomo y placas de óxido de plomo, apiladas de forma que no se tocan entre ellas, y sumergidas en el electrolito que es una solución ácido sulfúrico. Las placas de plomo están conectadas al ANODO y las placas de óxido de plomo están conectadas al CÁTODO. El potencial reductor del conjunto mantiene una diferencia de potencial entre bornes aproximada de 2 v, como ya se ha dicho.

Las reacciones REDOX que posibilitan la circulación de electrones al cerrar el circuito son las siguientes:

PROCESO DE DESCARGA :

(en la carga se producirán justo las reacciones inversas).



Como se puede ver, en la reacción de descarga se consume ácido sulfúrico (entre otras cosas) del electrolito y se produce sulfato de plomo que se deposita en las placas. Es bastante común oír hablar de que una batería que “SE HA SULFATADO” es inservible. Esto ocurre porque ha sufrido un proceso de descarga demasiado severo, en la placas se ha acumulado tanto sulfato de plomo que llegan incluso a deformarse, y el proceso se convierte en irreversible.

En el estado actual de la tecnología, las baterías de Plomo son las más utilizadas frente a otras alternativas

como Litio, Niquel-Cadmio, etc, por adaptarse mejor a condiciones severas de funcionamiento (ciclos carga-descarga), pudiendo estimar su vida útil en condiciones nominales en 5 años. Tienen su desventaja en el mayor peso, por ejemplo en el uso para vehículos híbridos, pero no afecta en una aplicación estática como es el almacenamiento de energía objeto de este proyecto.

El programa de simulación OpenDSS posee varias alternativas a la hora de implementar los ciclos de carga/descarga en el elemento baterías, obteniéndose un funcionamiento en el que ya se tienen en cuenta las pérdidas originadas (pérdidas de calor, ciclos de histéresis).

En la simulación que se presenta a continuación, he elegido el modo *FOLLOW*, que utiliza una curva *loadshape* impuesta, y que el programa trata de seguir modulando la intensidad de carga o descarga según el caso.

4.3.5 Modelado de transformadores e inversores en el BUS de corriente continua

Como ya se ha introducido en las hipótesis iniciales, y puede verse en el siguiente esquema unificar ampliado, en el modelado de la red hay que tener en cuenta la existencia de convertidores DC-DC desde los paneles fotovoltaicos, y de trafos AC-DC desde el aerogenerador. Por otro lado, desde el BUS CC hasta la red de distribución 230 CA, están los inversores DC-AC, figura 4-10.

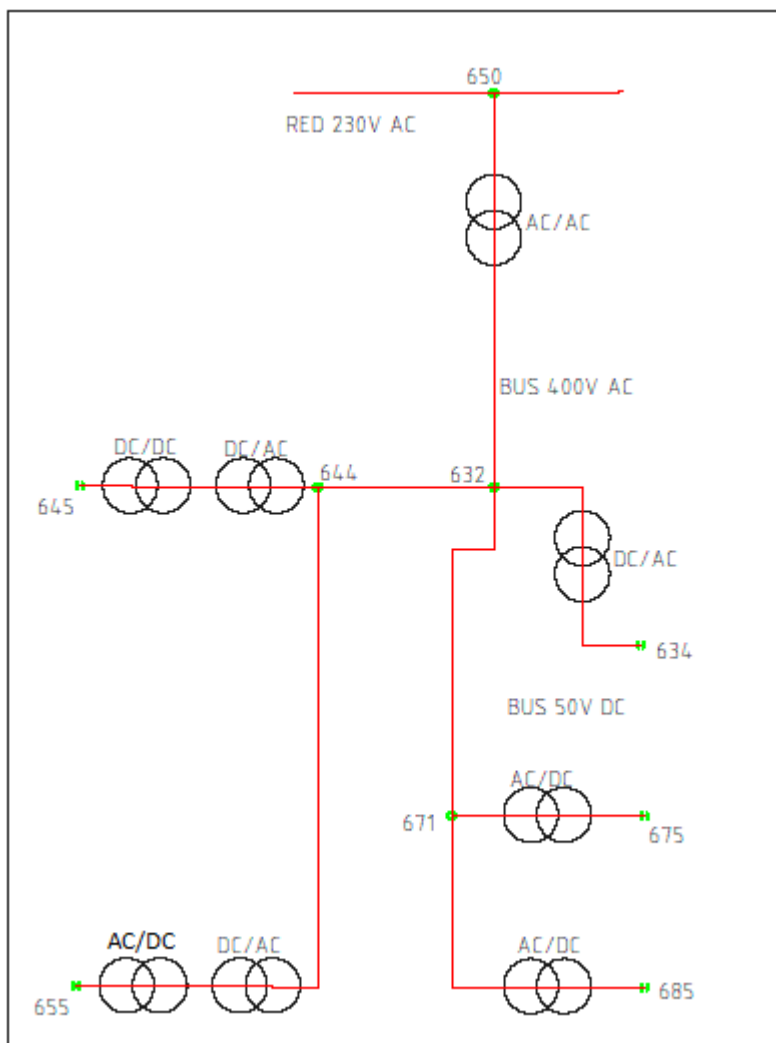


Figura 4-10. Esquema unifilar ampliado a equipos AC/DC – DC/DC – DC/AC

La simulación en OpenDSS se realizará integrando dichos elementos dentro del correspondiente equipo de

generación, por lo que se despreciará en este caso la pérdida, no más de 1 metro para cada equipo, correspondiente al tramo de cable en DC.

En resumen, y como ya se comentó al principio de esta sección, los dos modelos alternativos en función de las pérdidas que es lo que se pretende evaluar, son los siguientes:

→ Modelar los inversores junto con cada equipo. En esta opción despreciamos la distancia existente en el tramo de CC entre cada inversor y el propio equipo. Si esta distancia es pequeña, el error está acotado.

→ Modelar primero la parte de DC con distancias reales, y usar la solución obtenida como condición inicial para el resto de la red en AC. Esta solución es más exacta, y sería la ideal para un estudio avanzado de las pérdidas por operación en una microrred.

4.3.6 Modelado de todas las líneas en la red de 11 nodos.

Vamos a recopilar los datos de las líneas una por una, teniendo en cuenta las hipótesis planteadas anteriormente, de forma que la red trabaja (según el modelo aproximado de partida, figura 4-10) en corriente alterna, a una tensión de 230 V en monofásico, y a una frecuencia de 50 Hz.

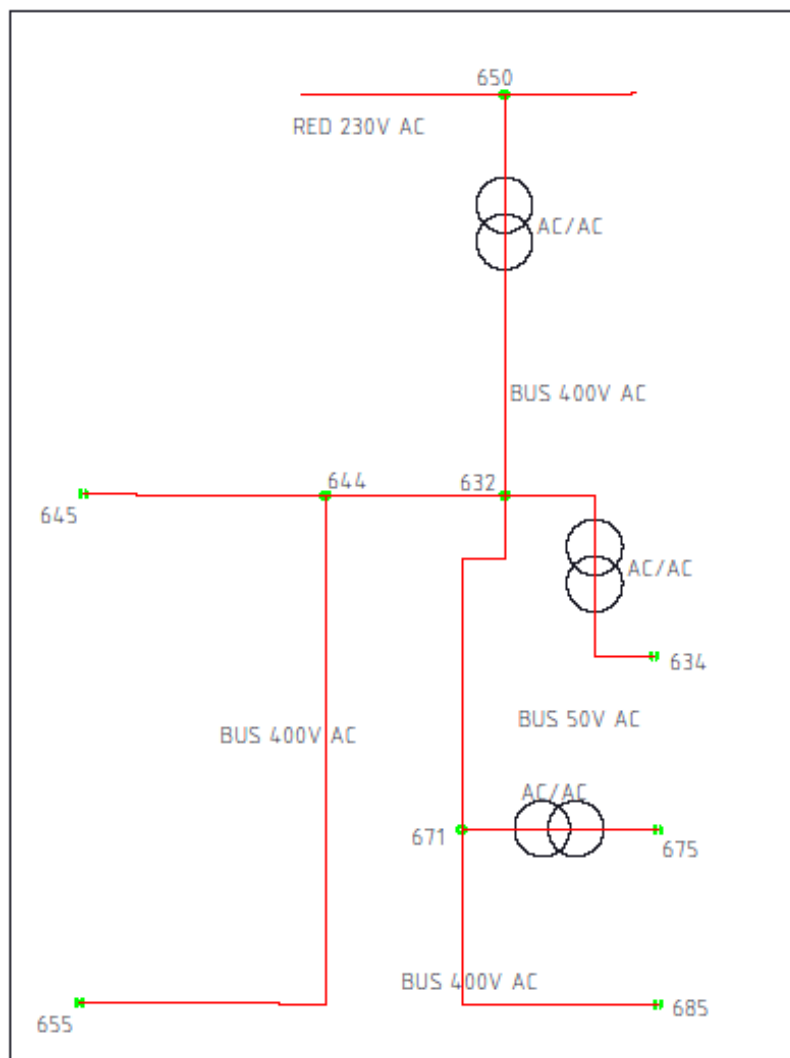


Figura 4-11. Esquema unifilar equivalente con BUSES en AC (modelo aproximado)

El dimensionamiento de cada línea, representada en la figura 4-11 anterior, se hace en función de la potencia nominal de las cargas conectadas a ella y siguiendo criterios del Reglamento Electrotécnico de Baja Tensión vigente [13]. Se resume en la tabla 4-1 a continuación.

Tabla 4-1.- Dimensionamiento de líneas (sección del conductor).

Bus	Sección (mm²)
650-632	10
632-644	4
644-645	2.5
644-655	2.5
632-633	4
633-634	2.5
634-635	2.5
633-671	4
671-685	4
671-675	35

La tabla anterior ha sido calculada siguiendo criterios dados por el citado REBT [13]. Con los datos facilitados por el INTA, las líneas existentes están del lado de la seguridad (mayor sección), lo que hace que las pérdidas debido a resistencia eléctrica sean menores. Se resume en la tabla 4-2 a continuación, tanto la sección de los conductores existentes como la impedancia resultante teniendo en cuenta la distancia en metros de cada tramo.

Tabla 4-2.- Líneas dimensionadas en la instalación real (INTA).

Bus	Sección (mm²)	Longitud (m)	Resistencia (ohmios / m)
650-632	16	50	1.07e-3
632-644	10	1	1.71e-3
644-645	6	25	2.85e-3
644-655	10	50	1.71e-3
632-633	16	1	1.07e-3
633-634	2.5	20	6.84e-3
634-635	2.5	1	6.84e-3
633-671	16	1	1.07e-3
671-685	16	16	1.07e-3
671-675	70	20	0.24e-3

Es importante mencionar que al código de OpenDSS le introduciremos los valores en km y en ohmios/km. Esto es debido a que se trata de un software desarrollado para simular redes de distribución con largas distancias entre nodos, y funciona mejor trabajando con unidades en km, en millas, y hasta incluso en unidades de 1000 pies según he podido ver en los tutoriales del programa.

4.3.7 Compensación de energía reactiva.

Mediante el software OpenDSS se ha podido simular el circuito de forma óptima, desde el punto de vista de la compensación de energía reactiva.

Esto se hace de dos formas: bien colocando objetos capacitadores en determinados nodos del circuito, o bien incluyendo valores de capacitancia en las líneas definidas mediante *linecode*.

En la resolución presentada a continuación se opta por colocar una batería de condensadores en conexión de estrella en el nodos 634 del bus de 48 Vcc.

En el algoritmo que se construye más adelante, al final de este capítulo con la ayuda y comunicación con el programa Matlab, y donde se integra la simulación del circuito con el Control Predictivo presentado en el Capítulo 3, se podría manipular la batería de condensadores de la siguiente forma:

- en función de las pérdidas por energía reactiva se conectará o desconectará dicha batería.
- el algoritmo tendrá en cuenta la estabilidad de los nodos en cuanto a caídas de tensión para conectar o desconectar la batería de condensadores.

Dicho lo cual, en la solución final presentada en este proyecto se ha optado por mantener un valor fijo en la batería de condensadores sea cual sea la configuración existente en el circuito. Con ello también se asume y se ha podido comprobar en las simulaciones, que las pérdidas por energía reactiva permanecen muy por debajo de las pérdidas por resistencia que sí se cuantifican y se tiene en cuenta.

4.3.8 Resultado y conclusiones de la simulación con OpenDSS.

En esta subsección se presentan los siguientes resultados y conclusiones:

Por una parte se ha simulado el circuito mediante un código OpenDSS (código A2-3 del Anexo2), que se ejecuta de forma independiente mediante la interfaz gráfica de este programa, generándose las siguientes figuras representativas. Se ha elegido una simulación dinámica de tipo *duty* (es decir, definiendo cargas y generadores mediante una curva que evoluciona temporalmente), y el salto de tiempo se configura en una hora.

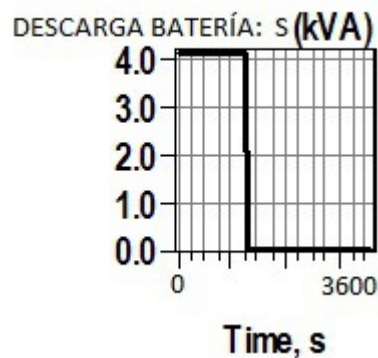


Figura 4-12. Transitorio de evolución de potencia en descarga de batería.

Tabla 4-3.- Listado de pérdidas por líneas y elementos.

LOSSES REPORT

Power Delivery Element Loss Report

Element	kW Losses	% of Power	kvar Losses
"Transformer.SUB"	0.00001,	0.0	0.000341867
"Transformer.TRAFO1"	0.00308,	0.04	0.0616484
"Transformer.TRAFO2"	0.00214,	0.04	0.0428104
"Capacitor.CAP1"	0.00000,	0.0	-0.00665647
"Line.L0"	0.00001,	0.0	0.000347418
"Line.L1"	0.00000,	0.0	-1.61015E-007
"Line.L2"	0.00000,	0.0	-8.04803E-007
"Line.L3"	0.00000,	0.0	0.000171037
"Line.L4"	0.00109,	0.02	0.0723344
"Line.L5"	0.00004,	0.0	0.00268781
"Line.L6"	0.00000,	0.0	6.8806E-005
"Line.L7"	0.00000,	0.0	8.58953E-005
"Line.L8"	0.00002,	0.0	0.000783117
"Line.L9"	0.00002,	0.0	0.000697829

TOTAL LOAD POWER = 6.0 kW

Percent Losses for Circuit = 0.11 %

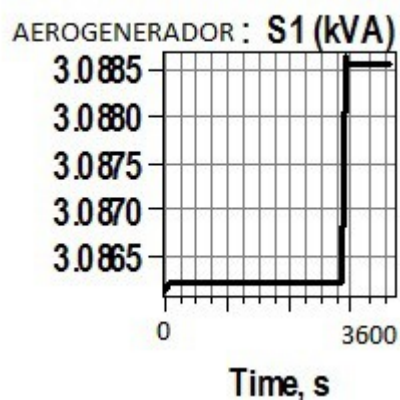


Figura 4-13. Transitorio de evolución de potencia en generador eólico.

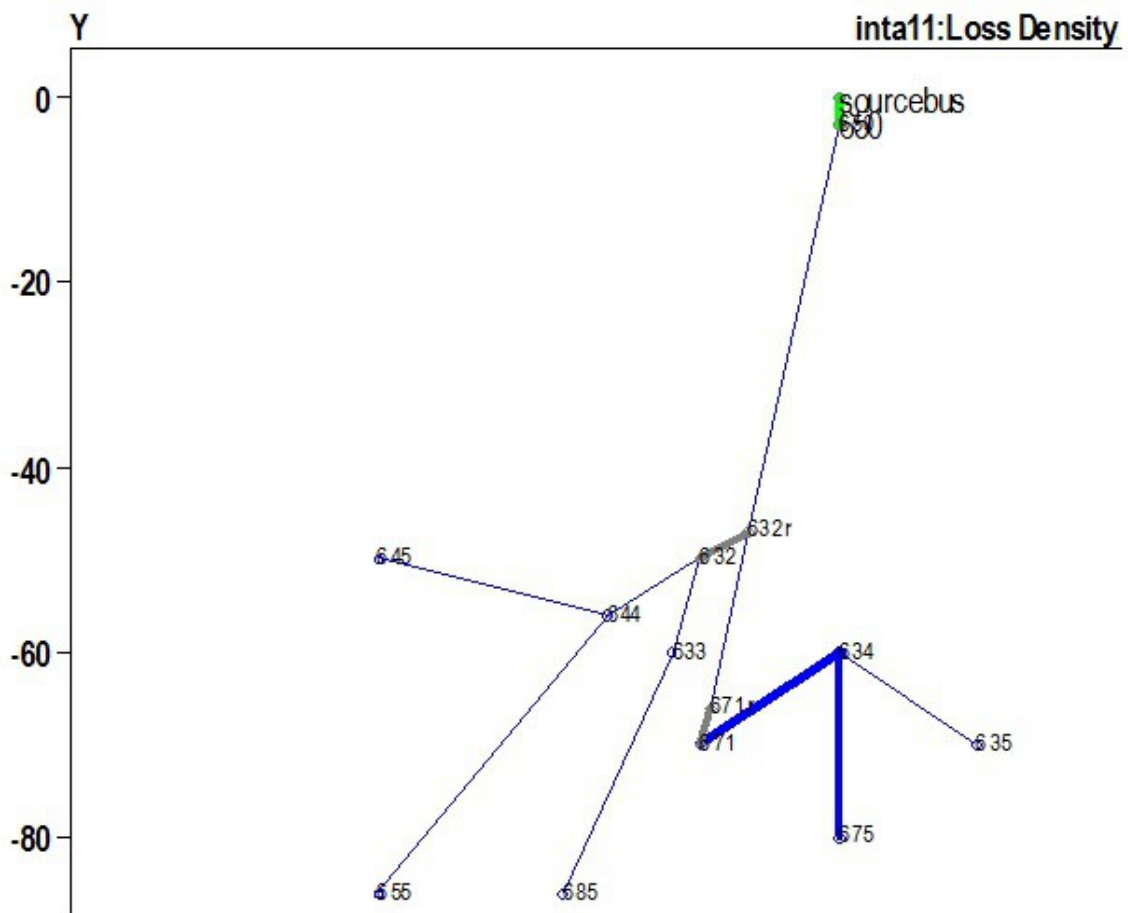


Figura 4-14. Geometría de buses. Representación de pérdidas en la red mediante grosor de líneas.

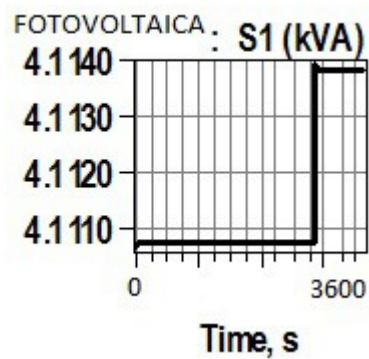


Figura 4-15. Transitorio de evolución de potencia en generador fotovoltaico.

Tabla 4-4.- Listado de voltajes por buses y nodos.

SYMMETRICAL COMPONENT VOLTAGES BY BUS (for 3-phase buses)

Bus	Mag:	V1 (kV)	p.u.	V2 (kV)	%V2/V1	V0 (kV)	%V0/V1
sourcebus		2.402	1	0	0	0	0
650		0.2309	1	0	0	0	0
632r		0.2309	0.9999	0	0	0	0
632		0.3927	1	0	0	0	0
671r		0.2309	0.9999	0	0	0	0
671		0.0502	0.9995	0	0	0	0
634		0.05019	0.9992	0	0	0	0
633		0.3927	1	0	0	0	0
685		0.3927	1	0	0	0	0
675		0.05019	0.9992	0	0	0	0
635		0.05019	0.9992	0	0	0	0
644		0.3927	1	0	0	0	0
655		0.3927	1	0	0	0	0
645		0.3927	1	0	0	0	0

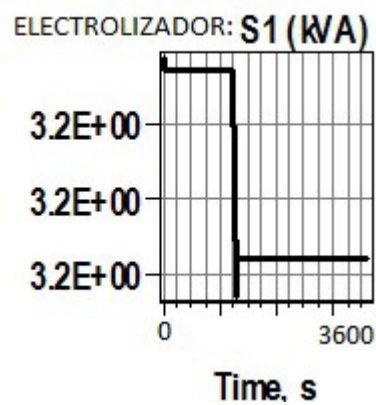


Figura 4-16. Transitorio de evolución de potencia en electrolizador.

Tabla 4-5.- Listado de intensidades por líneas y elementos.

SYMMETRICAL COMPONENT CURRENTS BY CIRCUIT ELEMENT (first 3 phases)

Element	Term	I1	I2	%I2/I1	I0	%I0/I1	%Normal	%Emergency
"VSOURCE.SOURCE"	1	0.1677	0.1677	100.00	0.1677	100.00	0.00	0.00
"TRANSFORMER.SUB"	1	0.5031	0	0.00	0	0.00	0.38	0.28
"TRANSFORMER.TRAFO1" ..	1	31.032	0	0.00	0	0.00	22.57	16.55
"TRANSFORMER.TRAFO2" ..	1	25.858	0	0.00	0	0.00	18.81	13.79
"CAPACITOR.CAP1"	1	0.13262	0	0.00	0	0.00	0.00	0.00
"LINE.L0"	1	5.2321	0	0.00	0	0.00	1.31	0.87
"LINE.L1"	1	0.084933	0	0.00	0	0.00	0.02	0.01
"LINE.L2"	1	0.084933	0	0.00	0	0.00	0.02	0.01
"LINE.L3"	1	25.858	0	0.00	0	0.00	6.46	4.31
"LINE.L4"	1	118.89	0	0.00	0	0.00	29.72	19.81
"LINE.L5"	1	102.49	0	0.00	0	0.00	25.62	17.08
"LINE.L6"	1	16.399	0	0.00	0	0.00	4.10	2.73
"LINE.L7"	1	18.339	0	0.00	0	0.00	4.58	3.06
"LINE.L8"	1	7.8642	0	0.00	0	0.00	1.97	1.31
"LINE.L9"	1	10.475	0	0.00	0	0.00	2.62	1.75
"LOAD.SL1"	1	102.49	0	0.00	0	0.00	0.00	0.00
"LOAD.SL2"	1	16.399	0	0.00	0	0.00	0.00	0.00
"STORAGE.BATTERY"	1	0.084933	0	0.00	0	0.00	0.00	0.00
"PVSYSTEM.PV"	1	10.475	0	0.00	0	0.00	0.00	0.00
"GENERATOR.GENWIND" ...	1	7.8642	0	0.00	0	0.00	0.00	0.00
"GENERATOR.FUELCELL" ..	1	0	0	0.00	0	0.00	0.00	0.00

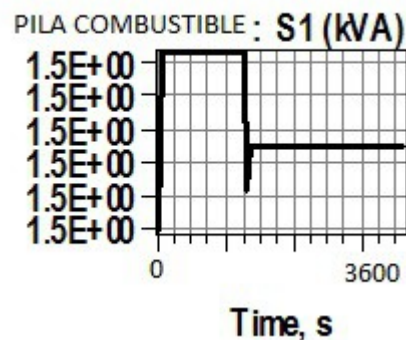


Figura 4-17. Transitorio de evolución de potencia en pila de combustible.

A continuación también se incluyen algunas gráficas que han sido obtenidas con Matlab, utilizando la

comunicación con OpenDSS a través del COM Server como ya se ha explicado al principio de este capítulo.

Se pueden importar cuantos datos se quieran una vez resuelto el circuito, hacer el tratamiento de dichos datos mediante lenguaje Matlab, y usar el entorno gráfico de este último programa que a juicio de este autor es más ilustrativo en simulaciones.

A partir de este punto en este trabajo, ya toda la simulación del circuito representada gráficamente se hace usando Matlab.

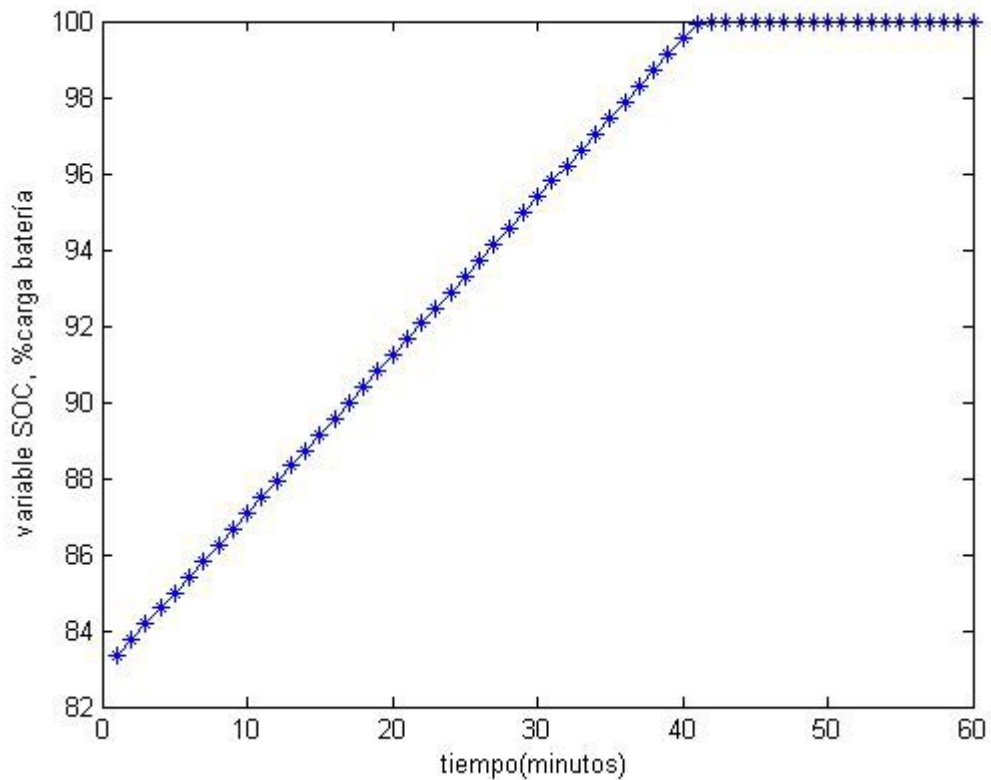


Figura 4-18. Evolución del nivel de carga SOC en la batería.

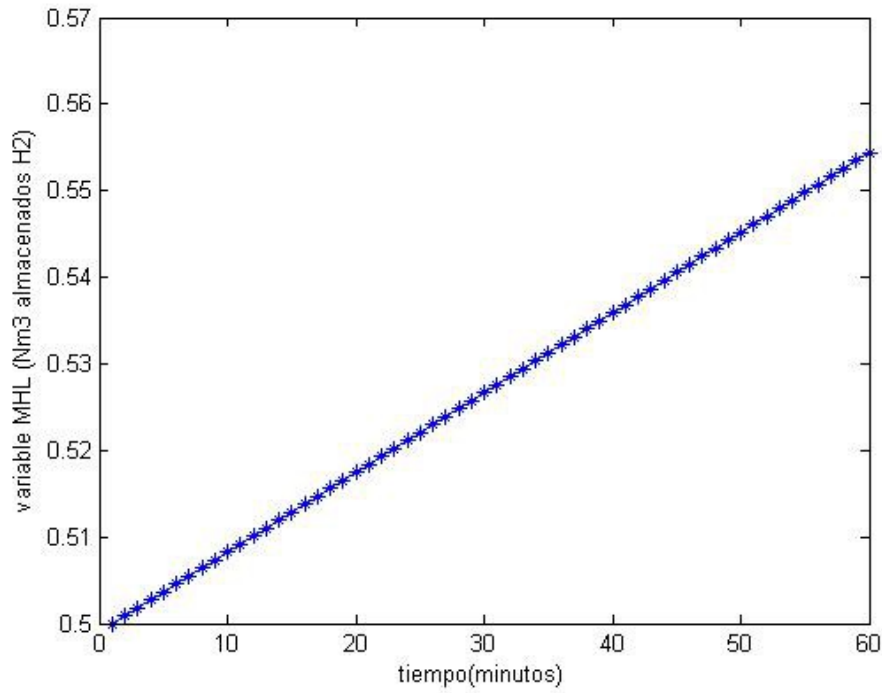


Figura 4-19. Evolución del nivel de almacenamiento MHL de hidrógeno.

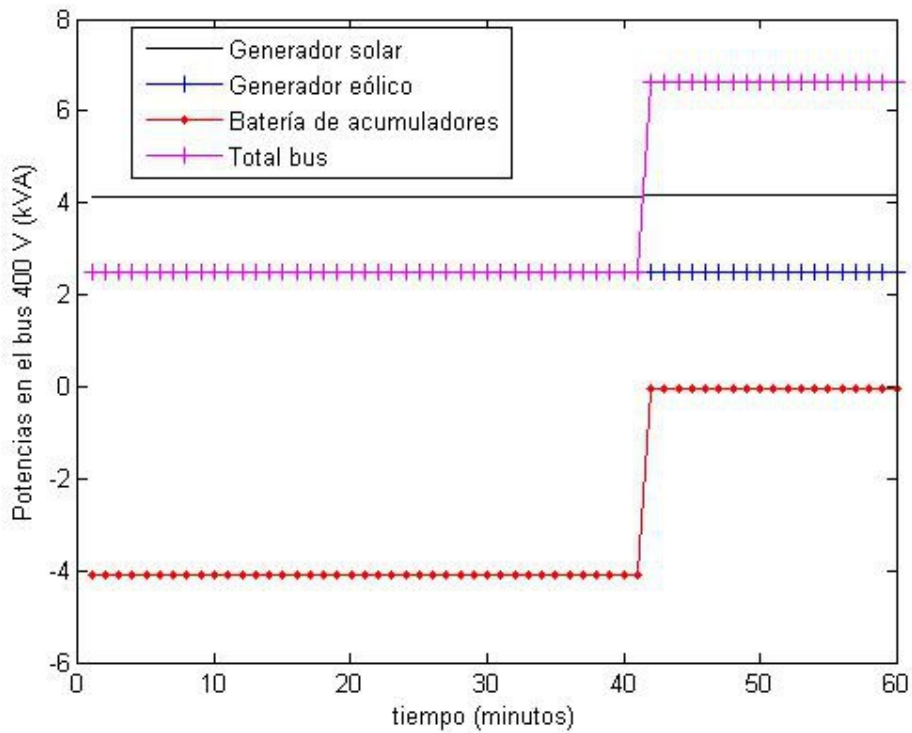


Figura 4-20. Evolución de potencias en el BUS de 400v.

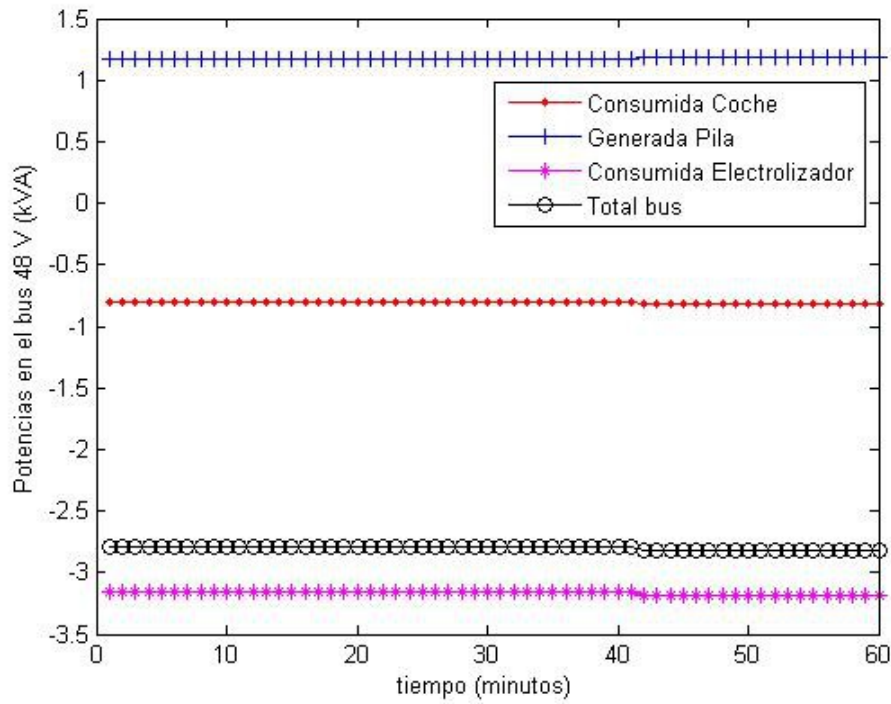


Figura 4-21. Evolución de potencias en el BUS de 50v.

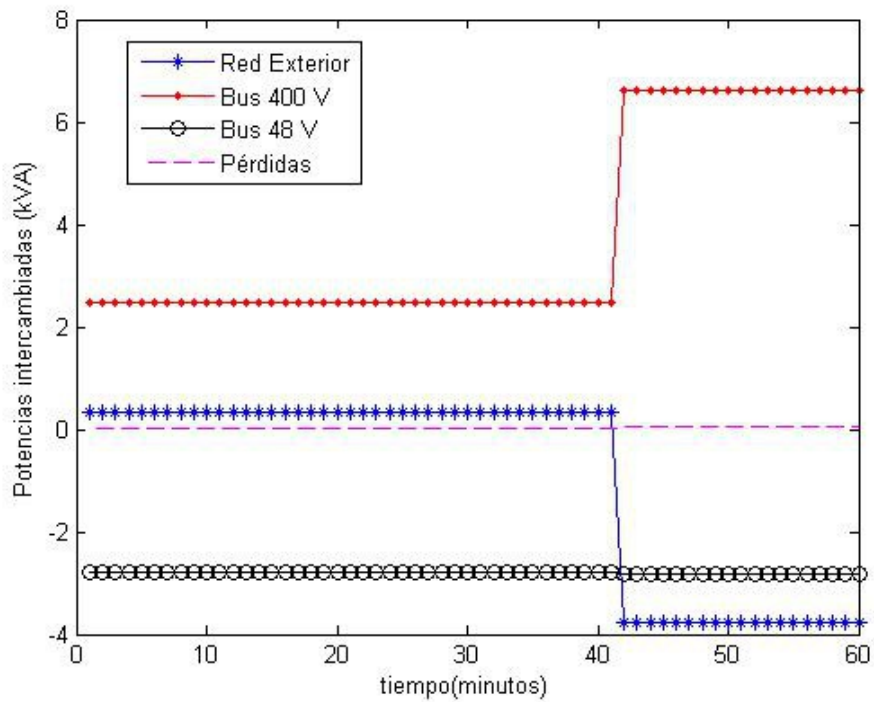


Figura 4-22. Evolución de potencias de intercambio con red exterior y pérdidas.

Los códigos utilizados en OpenDSS y Matlab para la obtención de los resultados anteriores se incluyen en el Anexo A2, al final de este trabajo.

4.4 Simulación de la red INTA 11 nodos. Modelo corregido con OpenDSS

4.4.1 Implementación del modelo sin pérdidas en la red.

Es lo que se ha hecho en el capítulo anterior, y para refrescar los resultados se insertan dos gráficas representativas de los resultados que se obtuvieron.

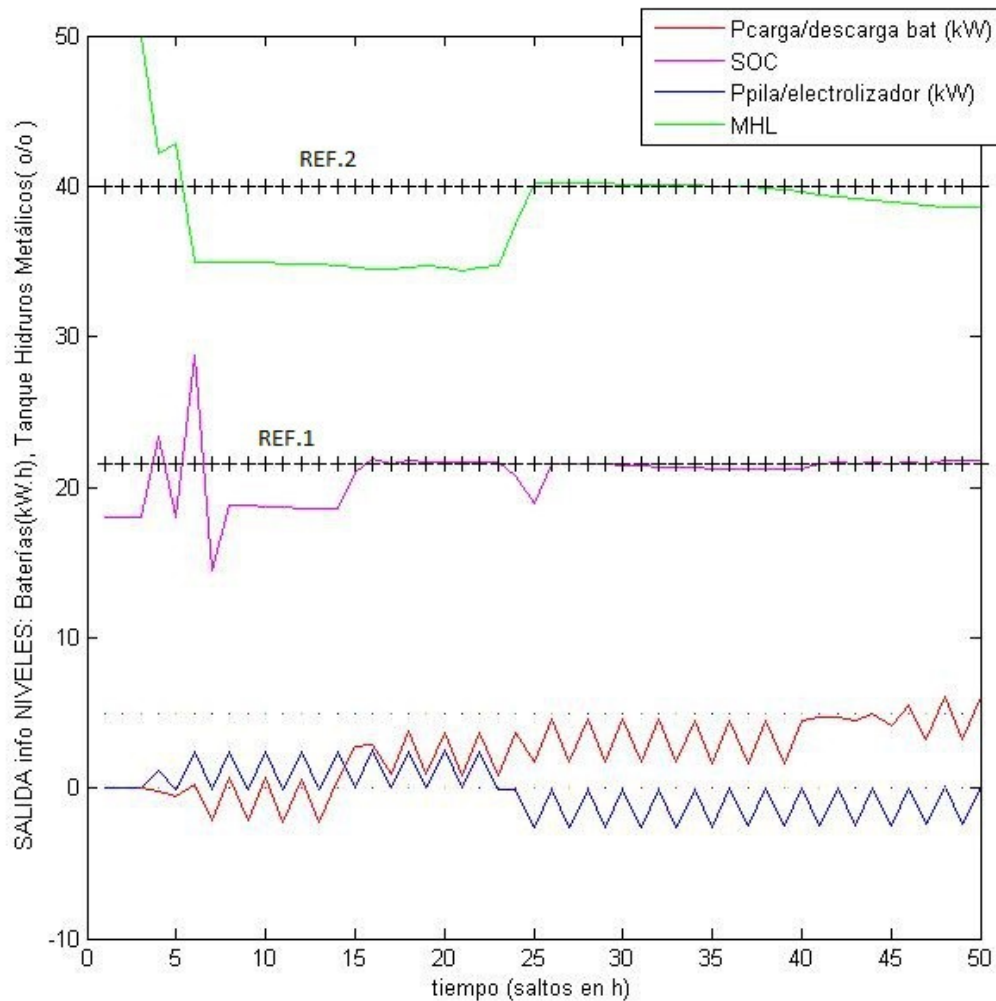


Figura 4-23. Simulación de GPC usando Matlab.

notas: REF.2 al 40% (0.4 Nm³), nivel de referencia del depósito MHL.
REF.1 al 60% (21.6 kW.h) nivel de referencia de almacenamiento SOC.

En ellas podemos ver la evolución de las dos señales de entrada y las dos señales de salida del algoritmo GPC, en una serie temporal de periodos discretos que simulan periodos horarios, y sometiendo a la planta a diferentes situaciones en cuanto a consumos y generación de potencia desde el exterior. El máximo horizonte temporal en estudio, o periodo de muestreo en esta simulación, es de 2 días.

Tanto la evolución gráfica anterior, como la de la siguiente figura, representan situaciones idénticas excepto en

las referencias impuestas para los estados de salida, como se puede ver en las leyendas al pie de gráfica. También se obliga a dichos estados a arrancar desde valores diferentes, como se aprecia en las líneas representadas.

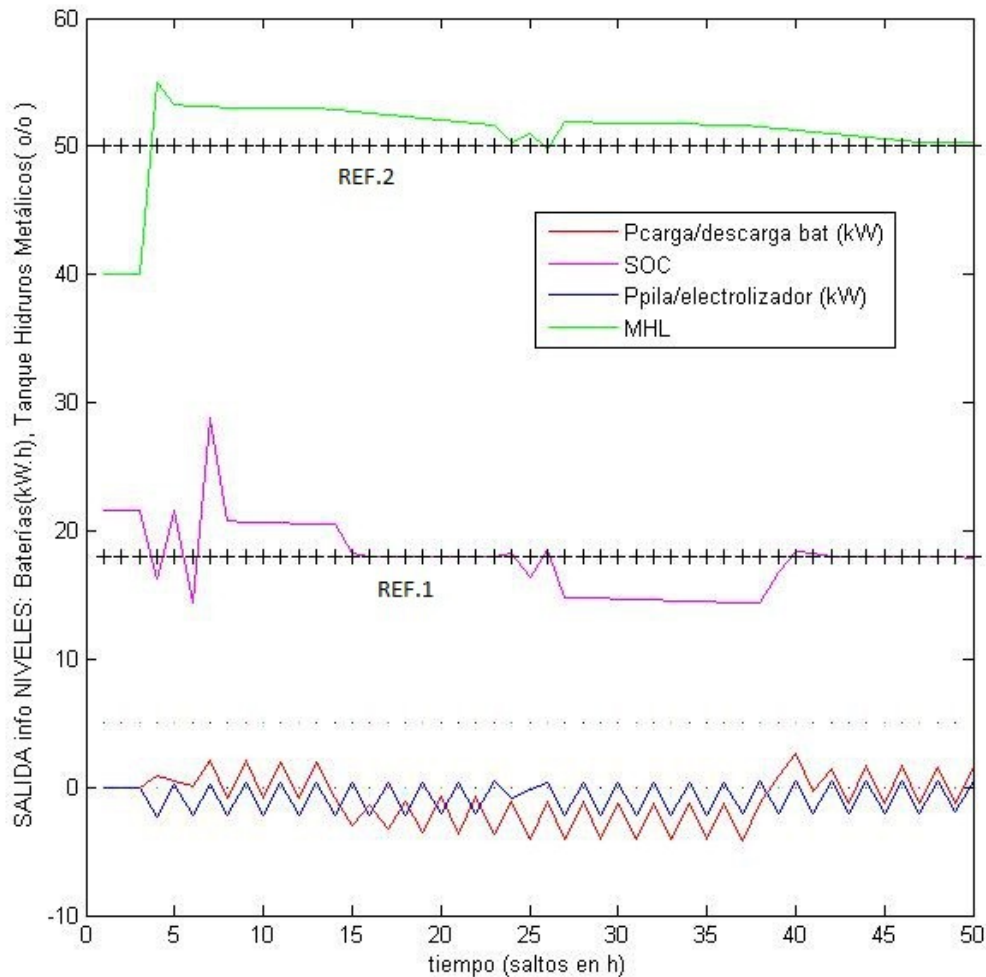


Figura 4-24. Simulación 2 de GPC usando Matlab.

notas: REF.2 al 50% (0.5 Nm³), nivel de referencia del depósito MHL.
REF.1 al 50% (18 kW.h) nivel de referencia de almacenamiento SOC.

4.4.2 El modelo de la red simulado y corregido con OpenDSS.

Se representan en las gráficas de esta subsección, dos variantes del algoritmo que se ha resuelto, con la siguiente leyenda de interpretación:

- En línea continua se representa la evolución del algoritmo usando el modelo de función de transferencia, construido en base a los datos disponibles de la planta.
- en asteriscos se superponen las señales reales medidas en la planta por el software de simulación OpenDSS.

1ª variante: simulación del algoritmo de control predictivo en comunicación simultánea con el programa OpenDSS, para una estructura GPC con restricciones lineales y función objetivo cuadrática.

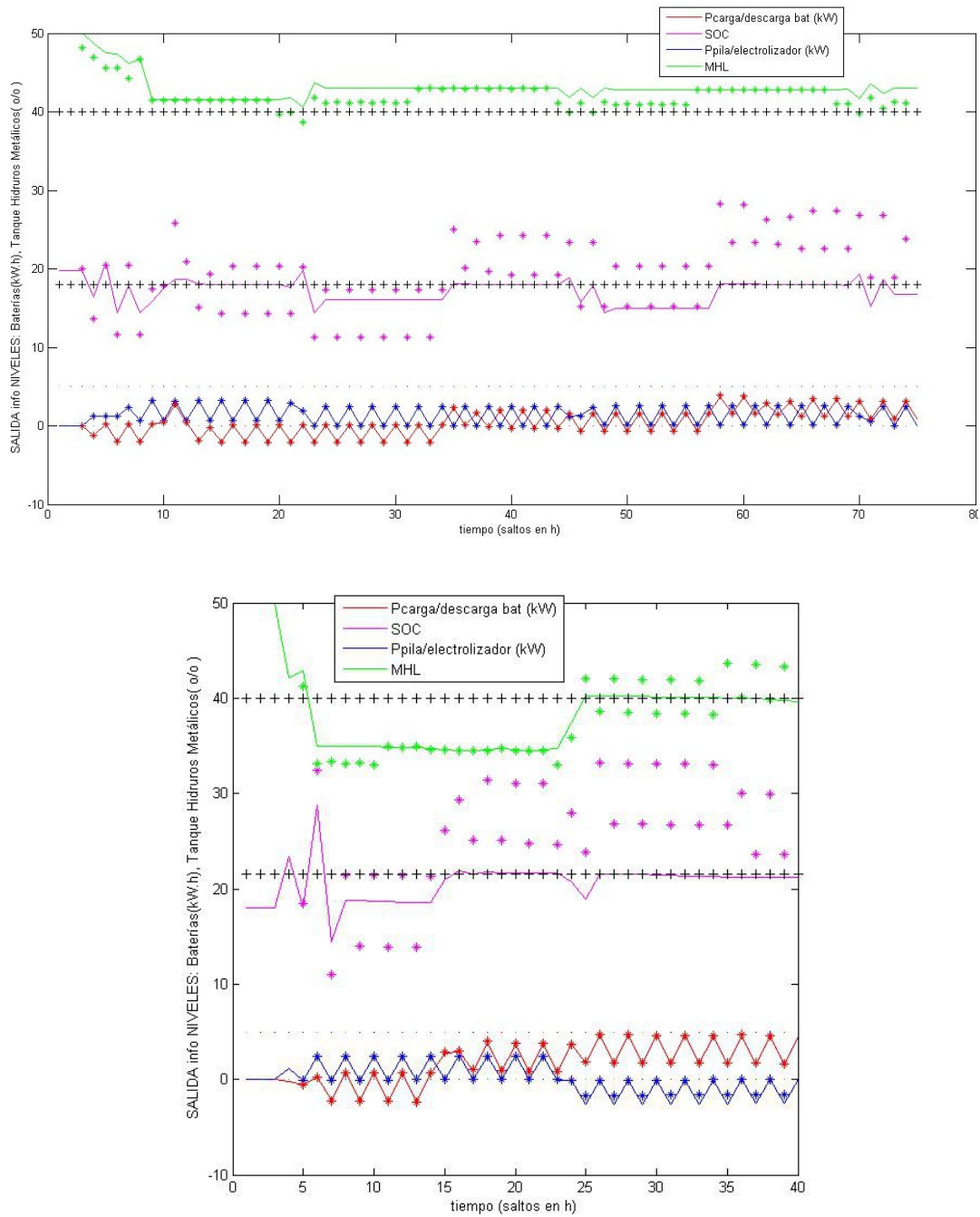


Figura 4-25. Diferentes simulaciones de algoritmo GPC simultaneamente con OpenDSS.

Línea continua: GPC siguiendo modelo aproximado del circuito.

Asteriscos: evolución siguiendo señales medidas por OpenDSS.

Vemos que la segunda gráfica sólo difiere de la primera (dentro de la figura 4-25), en que hemos subido la consigna o referencia para la salida SOC. El modelo de función de transferencia utilizado (líneas continuas) hace que cualquier cambio en una variable de entrada o salida, afecte inmediatamente a la otra variable.

2ª variante: simulación del control predictivo mediante un algoritmo construido sin restricciones, dejando que sea el software OpenDSS el que se encargue en cada paso de limitar los valores tanto de variables de entrada (potencias), como de variables de salida (estados).

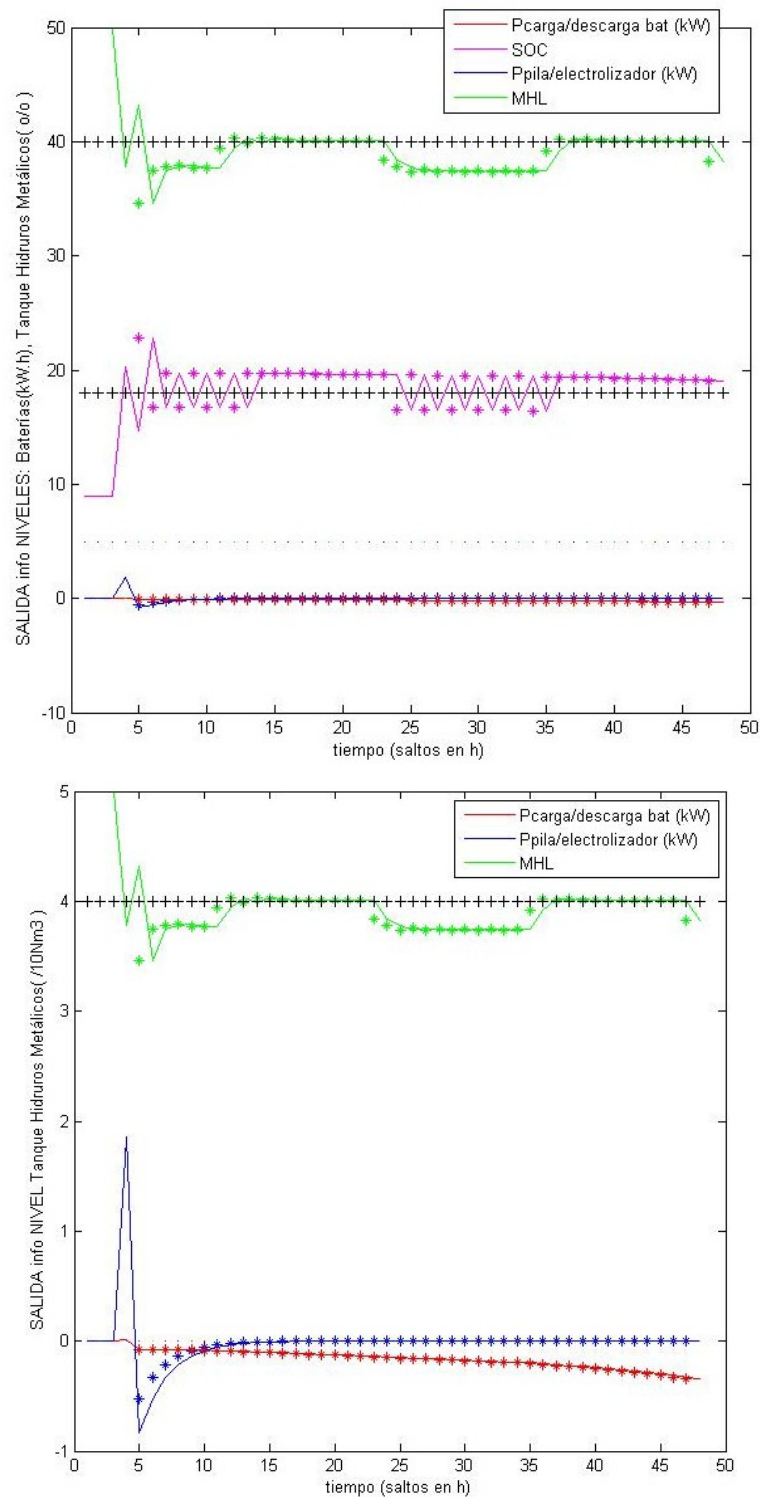


Figura 4-26. Control Predictivo sin restricciones. OpenDSS limita entradas y salidas.

En la segunda pantalla gráfica de Matlab mostrada en la figura anterior, se ha ampliado la escala para que se aprecien correctamente las fluctuaciones de las señales de control $P_{HYDROGEN}$ y $P_{CIRCUIT}$. Al final de la simulación, en la consola del programa también se muestra el último estado medido con las variables del

circuito, y también el balance final de los monitores de medida de energía (CONTADORES), de la siguiente forma:

Tabla 4-6.- Pantalla de información en tiempo real de la simulación.

```

*****
Electrolizador ON
Batería cargando
-----
Acumulado de energía procedente de renovables: 158.169 kW.h
Acumulado de energía exportada a la red exterior: 147.003 kW.h
Acumulado de energía cedida al vehículo híbrido: 15.2 kW.h
*****
LOS DATOS ACTUALES DE LA PLANTA SON LOS SIGUIENTES:
NIVEL SOC      Consumo H2      NIVEL MHL      ENERGÍA renovables
71.049444(o/o) -0.800000(kW)  38.599016(o/o)  0.240000(kW)
*****

```

En el Anexo2 se incluye el código completo (incluida llamada a función que conecta con OpenDSS en cada salto del predictivo), usado para la obtención de los resultado anteriores.

4.5 Jerarquía de Operatividad

En esta sección, como ya se avanzaba en el capítulo anterior dedicado al análisis de operatividad de la planta, simplemente voy a describir la secuencia de actuación de los diferentes niveles (capas) de la simulación completa, es decir:

→ saltos en minutos de la simulación → resolución mediante OpenDSS → señales de alarma / desconexión contenidas en este software.

PRIORIDAD 1ª: seguridad física (no accidente) de la planta. Tiempo de respuesta: MINUTOS.

→ saltos en horas mediante algoritmo GPC → optimización y limitación (restricciones) de las señales de control.

PRIORIDAD 2ª: seguridad operacional de la planta (anticipación a condiciones de riesgo). Tiempo de respuesta: HORAS.

→ periodo diario de muestreo de la simulación completa → análisis de parámetros relativos a la gestión óptima económica (aprovechamiento máximo) de los recursos energéticos disponibles.

PRIORIDAD 3ª: seguridad en cuanto a la rentabilidad económica de la inversión. Tiempo de respuesta: DÍAS.

4.6 Análisis de los resultados

Una vez obtenidos los resultados anteriores, se expone finalmente una visión global del algoritmo:

CARACTERÍSTICAS GENERALES:

- periodo de muestreo 1 día, dividido a su vez en 24 subperiodos de muestreo:

- en cada subperiodo de muestreo de 1 hora se realiza una simulación simultánea mediante OpenDSS, teniendo como entrada las señales de control obtenidas por GPC y obteniendo como resultado las variables de estado, al mismo tiempo que el control predictivo también calcula el valor de la salida.
- las señales de control usadas al principio de cada salto son las que se obtienen como solución del algoritmo GPC, y éstas junto con las señales de salida (variables de estado) resueltas por OpenDSS, sirven al algoritmo GPC para obtener la siguiente predicción.

Al comienzo de cada salto se realizan dos cálculos simultáneos, uno mediante la resolución del predictor GPC y otro mediante la solución dada por OpenDSS para las mismas señales de entrada. Las entradas pasadas que sirven para la resolución del predictivo del salto siguiente son las que calcule el algoritmo GPC. Las salidas pasadas que se realimentan serán las obtenidas por OpenDSS, y no las que calcula el modelo del GPC. A la vista de las gráficas presentadas anteriormente, dicho modelo difiere del comportamiento real al que la simulación OpenDSS puede aproximarse mejor.

En resumen, la simulación conjunta de OpenDSS y GPC utiliza los mismos valores de entradas para predecir las salidas, y en el salto siguiente se realimenta la ecuación de predicción del GPC con el valor de salidas dado por OpenDSS por ser más realista (salidas pasadas junto con entradas pasadas). La señal de control obtenida se envía a la planta (a las dos plantas, por decirlo así), y vuelta a empezar.

Se ilustra el esquema descrito anteriormente con una gráfica.

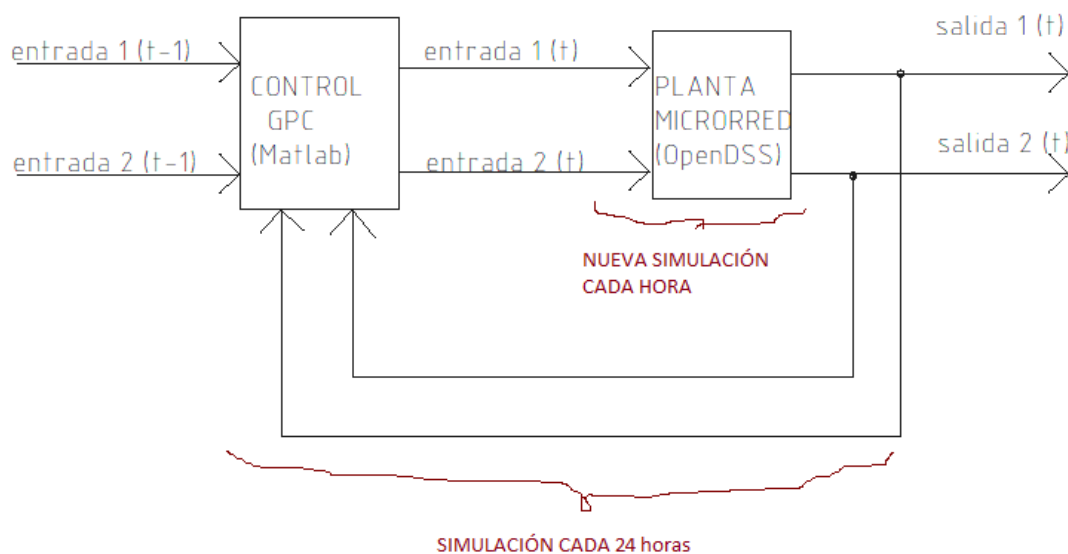


Figura 4-27. Diagrama de bloques de la doble simulación simultánea.

CONSECUENCIAS Y VENTAJAS DE SUPERPONER EL ENFOQUE PREDICTIVO USANDO UN MODELO PREDISEÑADO, Y CONTRASTAR EN LA SIMULACIÓN DICHO MODELO CON UN COMPORTAMIENTO REAL MEDIANTE OPENDSS:

→ Es una buena forma de facilitar el diseño previo de un controlador para cualquier proceso, con garantías de que su comportamiento real no diferirá mucho del previsto.

5 CONCLUSIONES

Si uno creía que el universo tuvo un principio, la pregunta obvia era, ¿qué sucedió con anterioridad a ese comienzo? ¿qué estaba haciendo Dios antes de hacer el Mundo? ¿estaba Él acaso preparando el Infierno para la gente que se hiciera tales preguntas?

- Stephen Hawking -

A modo de conclusión, repasaré los aspectos más importantes que se han tratado en este trabajo, como son el estudio de microrredes de generación distribuida, el control predictivo basado en modelos como solución para la optimización de dichas instalaciones, y los procedimientos o métodos de análisis de fallos que posibilitan el funcionamiento en MODO SEGURO.

5.1 Repaso histórico

En la década de los 70 del siglo XX había en nuestro país un gran número de viviendas rurales sin acceso a la electricidad. Y cuando en estos hogares se usaban todavía medios tradicionales para calentarse y alumbrarse, en muchos de ellos apareció la generación distribuida mediante motores de combustión interna alternativa y dinamos de CC, cargándose baterías para la instalación de alumbrado a 24/ 48 v, ver ejemplo de la figura 5-1.



Figura 5-1. Generador años 70, CORTIJO EL MADROÑAL, Estepa (Sevilla).

Por tanto, en aquellos años ya podía abastecerse cualquier instalación rural, como mínimo con una instalación eléctrica de alumbrado. Además de esto, otros electrodomésticos como frigorífico y cocina ya podían funcionar con la popular bombona de butano. Para la calefacción se venía usando lo que hoy en día llamaríamos biomasa, es decir leña, orujo, aceites vegetales reciclados, restos de poda, etc.

Precisamente en 1973, con la crisis del petróleo se estimuló el estudio de fuentes alternativas (¡fue la gran oportunidad!). En el contexto descrito anteriormente, podrían haberse electrificado las zonas rurales mediante energía limpia, pero el atraso tecnológico de España no lo permitió, como sí ocurrió por ejemplo en Holanda donde la tecnología de los molinos eólicos ya estaba muy avanzada. Esta situación se describe de forma introductoria en los libros [7], [10] de la bibliografía.

La tendencia general fue en sentido contrario al descrito, y el posterior avance de nuestro país se fundamentó entre otros aspectos tecnológicos, en hacer llegar la red eléctrica (generación-transporte-distribución) hasta el último núcleo rural. Ese camino ya está andado y ahora será mucho más difícil recorrerlo a la inversa.

Hoy en día sería diferente si una familia se planteara las dos alternativas de inversión. El coste actual de electrificar una vivienda rural enganchando al tendido eléctrico más cercano oscila entre 6000 y 18000 € / Km. Por tanto, a partir de 1 Km, sin ninguna duda ya es rentable gastar ese dinero en un sistema de generación distribuida mediante energías renovables, que además se amortiza más rápido porque no hay que pagar peajes ni consumos a ninguna compañía. En el libro consultado [10] se detallan incluso los cuadros de amortización para este tipo de inversiones.

5.2 Perspectivas futuras

Como continuación de los resultados obtenidos en este proyecto se podría plantear el uso de las estrategias de control predictivo junto con un análisis modal de fallo, para concebir un sistema de “RIESGO CERO”.

Supongamos que podemos aislar una avería según su modo de fallo (típicamente rotura de rodamiento), y que podemos cuantificar con una fiabilidad del 95% (muy cerca del riesgo cero) el número de horas necesarias para que vuelva a fallar. En ese caso, podremos poner este valor entre las restricciones de un algoritmo de control predictivo que optimice la programación de la máquina (horas productivas frente a horas no productivas, anuales por ejemplo).

Esto último nos ayudaría a implantar sistemas modernos de gestión del mantenimiento en un proceso productivo: FILOSOFÍA LEAN, MANTENIMIENTO PREDICTIVO. Podremos mandar la orden de sustitución de la pieza (en nuestro ejemplo, rodamiento del eje del rotor), justo en el momento en que sabemos que va a romper, ni mucho antes ni después.

Un estudio de este tipo puede ser costoso para implantar en un determinado proceso industrial, pero actualmente en la Ingeniería de Procesos es sabido que las otras dos alternativas salen mucho más caras:

→ 1ª alternativa, MANTENIMIENTO CORRECTIVO, esperar a que la pieza rompa para sustituirla: típica situación de la antigua gestión del mantenimiento, que conlleva sobre todo pérdidas por paradas imprevistas en el proceso productivo, tener un equipo de electromecánicos siempre dispuestos para eventualidades y necesidad de gran estocaje en piezas de repuesto para reducir tiempos no productivos.

→ 2ª alternativa, MANTENIMIENTO PREVENTIVO, sustituir la pieza mucho antes de que rompa: sistema que ya ha sido ampliamente usado por la gran industria durante las últimas décadas. Mediante una adecuada gestión documental se puede garantizar el número de horas de funcionamiento sin incidencias; es lo que nos dice un fabricante de coches cuando compramos uno de sus modelos, “*cambie Vd la correa de la distribución a los 100.000 Km, y por supuesto hágalo en nuestra red de talleres*”. Lo que nunca nos van a decir es que casi con toda seguridad la correa de la distribución no romperá antes de los 500.000 Km, cuando el valor contable del equipo es cero.

Esta última alternativa, aunque no es tan fácil de valorar como la primera pues para ello hay que saber con exactitud cuándo se produce el fallo, sigue siendo costosa puesto que estamos desaprovechando recursos productivos (piezas o elementos y mano de obra consumidos en la sustitución prematura).

Creo que por tanto, queda debidamente manifiesta en este trabajo la importancia del enfoque predictivo en la ingeniería actual.

ANEXO 1: AMPLIACIÓN SOBRE EL ALGORITMO DE CONTROL PREDICTIVO BASADO EN MODELOS

RESOLUCIÓN DE LA ECUACIÓN DIOFÁNTICA, CASO UNIVARIABLE, RUIDO BLANCO:

Partiendo de un modelo para las perturbaciones, el llamado autoregresivo de media móvil integrado, por sus siglas *CARIMA* (*Controlled Auto-Regressive Integrated Moving Average*)

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})e(t) (I/\Delta) \quad (1-A1)$$

siendo los polinomios A y B :

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

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

d es el retardo (número de periodos discretos en los que el sistema tarda en reaccionar al estímulo de la señal entrada).

$e(t)$ es un ruido blanco coloreado por $C(z^{-1})$ que es un polinomio a definir.

(I/Δ) es un integrador en variables discretas, siendo $\Delta = 1 - z^{-1}$.

Vamos a resolver aquí el caso $C(z^{-1})$ igual a la unidad con lo que el modelo de perturbación se reduce al ruido blanco multiplicado por un integrador.

Veamos la expresión del predictor (expresión de la salida en el instante $t+k$ conocido el valor en el instante actual t), a partir de la ecuación (1-A1) se obtiene:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})z^{-d}}{A(z^{-1})} u(t+k-1) + \frac{C(z^{-1})e(t+k-1)}{\Delta A(z^{-1})} \quad (2-A1)$$

A continuación hay que resolver la ecuación diofántica. Es una ecuación que admite por definición más de una solución---se puede ampliar información consultando sobre la obra del matemático Diofanto de Alejandría (siglo III d.C.)---pero el procedimiento seguido aquí es el que se explica en el capítulo 4 del libro de referencia [1].

$$I = E_k(z^{-1})\Delta A + z^k F_k(z^{-1}) \quad (3-A1)$$

resolviéndose en este caso para un horizonte de predicción $N=3$. La obtención de los polinomios E_k, F_k se hará recursivamente para $k=1, 2, 3$.

Antes de resolver la ecuación diofántica, veamos cómo obtener una forma de la mejor predicción agrupando (2-A1) y (3-A1). En esta operación ya estamos dando valores, $d=0, C(z^{-1})=I$.

$$E_k(z^{-1})\Delta A(z^{-1})y(t+k) = E_k\Delta[B(z^{-1})u(t+j-1) + (1/\Delta)e(t+k)]$$

$$I = E_k(z^{-1})\Delta A + z^{-k}F_k(z^{-1}) \quad \rightarrow \text{introducimos ésta en el primer miembro de la anterior:}$$

$$\Leftrightarrow (I - z^{-k}F_k(z^{-1})) y(t+k) = E_k(z^{-1})B(z^{-1})\Delta u(t+k-1) + E_k(z^{-1})e(t+k) \Leftrightarrow$$

$$\Leftrightarrow y(t+k) = F_k(z^{-1})y(t) + E_k(z^{-1})B(z^{-1})\Delta u(t+k-1) + E_k(z^{-1})e(t+k) \Leftrightarrow$$

Posteriormente al resolver recursivamente (3-A1) observaremos que el grado del polinomio E_k es siempre exactamente el valor $k-1$.

Si nos fijamos en el término $E_k(z^{-1})e(t+k)$, al ser el grado de E_k igual a $k-1$, los términos $e(t+k)$ están en el futuro (instante k). Por tanto para hallar LA MEJOR PREDICCIÓN, prescindimos de este término:

$$\Leftrightarrow y(t+k)(I - F_k) = E_k(z^{-1})B(z^{-1})\Delta u(t+j-1)$$

si definimos la matriz polinomial G_k a partir de E_k y el polinomio B : $G_k(z^{-1}) = E_k(z^{-1})B(z^{-1})$

$$\Leftrightarrow \hat{y}(t+k|t) = G_k(z^{-1})\Delta u(t+k-1) + F_k(z^{-1})y(t) \quad (4-A1)$$

Hemos obtenido la mejor predicción y hemos visto cual es la utilidad de resolver la ecuación diofántica.

Para conocer la salida debemos finalmente resolver la ecuación diofántica y obtener las matrices G_k, F_k a partir del modelo conocido (polinomios A, B).

Con el propósito de hacer menos árida la exposición tomaré valores numéricos concretos de dichos polinomios, a modo de ejemplo, para un caso de una entrada y una salida que puede representar cualquier proceso típico en la industria.

$$A(z^{-1}) = 1 - 0.8z^{-1};$$

$$B(z^{-1}) = 0.4 + 0.6z^{-1};$$

Se resuelve de forma recursiva para $k=1,2,3$.

Empezamos en $k=1$ tomando $E_1(z^{-1}) = I \rightarrow$

\rightarrow Despejamos $F_1(z^{-1})$ en la ecuación (3-A1) particularizada para $k=1$.

$$\rightarrow I = E_1(z^{-1})\Delta A + z^{-1}F_1(z^{-1}) \Leftrightarrow$$

$$\Leftrightarrow F_1 = z(I - \Delta A) = z\{1 - (1 - 0.8z^{-1})(1 - 0.8z^{-1})\} = z(1 - 1.8z^{-1} + 0.8z^{-2}) \Leftrightarrow$$

(5-A1)

$$\Leftrightarrow F_1(z^{-1}) = 1.8 - 0.8z^{-1}; \Leftrightarrow E_1(z^{-1}) = I;$$

Lo que vamos a hacer a continuación sirve para cualquier caso general. Se avanza el índice $k=2,3,\dots,N$, y se toma el polinomio E_k como:

$$E_k = E_{k-1} + f_{k-1,0} z^{-(k-1)} ;$$

siendo $f_{k-1,0}$ el coeficiente de grado 0 del polinomio F_{k-1} .

Para el resto de pasos se procede de la misma forma que en (5-A1) hasta obtener los polinomios (matrices) E_k y F_k .

El grado de F_k siempre resulta ser igual a del polinomio $A(z^{-1})$, mientras que el grado de E_k siempre es $(k-1)$.

Para $k=2$:

$$E_2 = E_1 + f_{1,0} z^{-1} = 1 + 1.8 z^{-1} ;$$

$$\rightarrow I = E_2(z^{-1})\Delta A + z^2 F_2(z^{-1}) \Leftrightarrow$$

$$\Leftrightarrow F_2 = z^2 (I - E_2 \Delta A) = z^2 \{1 - (1 + 1.8z^{-1})(1 - z^{-1})(1 - 0.8z^{-1})\} = z^2 \{1 - (1 + 1.8z^{-1})(1 - 1.8z^{-1} + 0.8z^{-2})\}$$

\Leftrightarrow

$$\Leftrightarrow F_2(z^{-1}) = z^2 (2.44 z^{-2} - 1.44 z^{-3}) ; \Leftrightarrow F_2(z^{-1}) = 2.44 - 1.44 z^{-1} ;$$

Para $k=3$:

$$E_3 = E_2 + f_{2,0} z^{-2} = 1 + 1.8 z^{-1} + 2.44 z^{-2} ;$$

$$\rightarrow I = E_3(z^{-1})\Delta A + z^3 F_3(z^{-1}) \Leftrightarrow$$

$$\Leftrightarrow F_3 = z^3 (I - E_3 \Delta A) = z^3 \{1 - (1 + 1.8z^{-1} + 2.44 z^{-2})(1 - z^{-1})(1 - 0.8z^{-1})\}$$

$$\Leftrightarrow F_3 = z^3 \{1 - (1 + 1.8z^{-1} + 2.44 z^{-2})(1 - 1.8z^{-1} + 0.8z^{-2})\} \Leftrightarrow$$

$$\Leftrightarrow F_3(z^{-1}) = z^3 (2.95 z^{-3} - 1.95 z^{-4}) ; \Leftrightarrow F_3(z^{-1}) = 2.95 - 1.95 z^{-1} ;$$

Resumiendo, los polinomios obtenidos en forma de matrices:

$$E(z^{-1}) = \begin{bmatrix} 1 \\ 1 + 1.8z^{-1} \\ 1 + 1.8z^{-1} + 2.44z^{-2} \end{bmatrix} ; \quad F(z^{-1}) = \begin{bmatrix} 1.8 - 0.8z^{-1} \\ 2.44 - 1.44z^{-1} \\ 2.95 - 1.95z^{-1} \end{bmatrix} ; \quad (6-A1)$$

Como se ha visto en (4-A1), G se obtiene a partir de E y B , también de forma recursiva (para $k=1, 2, 3$):

$$\rightarrow G_1(z^{-1}) = E_1(z^{-1}) B(z^{-1}) = 1 (0.4 + 0.6 z^{-1}) = 0.4 + 0.6 z^{-1}$$

$$\rightarrow G_2(z^{-1}) = E_2(z^{-1}) B(z^{-1}) = (1 + 1.8 z^{-1}) (0.4 + 0.6 z^{-1}) = 0.4 + 1.32 z^{-1} + 1.08 z^{-2}$$

$$\begin{aligned} \rightarrow G_3(z^{-1}) &= E_3(z^{-1}) B(z^{-1}) = (1 + 1.8 z^{-1} + 2.44 z^{-2}) (0.4 + 0.6 z^{-1}) \Leftrightarrow \\ &\Leftrightarrow G_3(z^{-1}) = 0.4 + 1.32 z^{-1} + 2.06 z^{-2} + 1.46 z^{-3} \end{aligned}$$

$$G(z^{-1}) = \begin{bmatrix} 0.4 + 0.6z^{-1} \\ 0.4 + 1.32z^{-1} + 1.08z^{-2} \\ 0.4 + 1.32z^{-1} + 2.06z^{-2} + 1.46z^{-3} \end{bmatrix}; \quad (7-A1)$$

Ahora podemos usar la ecuación (4-A1) y los polinomios F_k , G_k de forma recursiva, para obtener de forma matricial la expresión del predictor:

$$k=1;$$

$$\begin{aligned} \hat{y}(t+1/t) &= G_1(z^{-1}) \Delta u(t+1-1) + F_1(z^{-1}) y(t) = \\ &= (0.4 + 0.6 z^{-1}) \Delta u(t) + (1.8 - 0.8 z^{-1}) y(t) = \\ &= 0.4 \Delta u(t) + 0.6 \Delta u(t-1) + 1.8 y(t) - 0.8 y(t-1) \end{aligned}$$

$$k=2;$$

$$\begin{aligned} \hat{y}(t+2/t) &= G_2(z^{-1}) \Delta u(t+2-1) + F_2(z^{-1}) y(t) = \\ &= (0.4 + 1.32 z^{-1} + 1.08 z^{-2}) \Delta u(t+1) + (2.44 - 1.44 z^{-1}) y(t) = \\ &= 0.4 \Delta u(t+1) + 1.32 \Delta u(t) + 1.08 \Delta u(t-1) + 2.44 y(t) - 1.44 y(t-1) \end{aligned}$$

$$k=3;$$

$$\begin{aligned} \hat{y}(t+3/t) &= G_3(z^{-1}) \Delta u(t+3-1) + F_3(z^{-1}) y(t) = \\ &= (0.4 + 1.32 z^{-1} + 2.06 z^{-2} + 1.46 z^{-3}) \Delta u(t+2) + (2.95 - 1.95 z^{-1}) y(t) = \\ &= 0.4 \Delta u(t+2) + 1.32 \Delta u(t+1) + 2.06 \Delta u(t) + 1.46 \Delta u(t-1) + 2.95 y(t) - 1.95 y(t-1) \end{aligned}$$

Expresado de forma compacta:

$$\begin{bmatrix} \hat{y}(t+1/t) \\ \hat{y}(t+2/t) \\ \hat{y}(t+3/t) \end{bmatrix} = \begin{bmatrix} 0.4 & 0 & 0 \\ 1.32 & 0.4 & 0 \\ 2.06 & 1.32 & 0.4 \end{bmatrix} \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \Delta u(t+2) \end{bmatrix} + \begin{bmatrix} 0.6 \\ 1.08 \\ 1.46 \end{bmatrix} \Delta u(t-1) + \begin{bmatrix} 1.8 \\ 2.44 \\ 2.95 \end{bmatrix} y(t) - \begin{bmatrix} 0.8 \\ 1.44 \\ 1.95 \end{bmatrix} y(t-1) \quad (8-A1)$$

Finalmente, en la ecuación anterior hemos obtenido la expresión matricial para la predicción de salida:

$$\hat{y} = Gu + f$$

c.q.d.

MODELO TRANSFORMADO ETSI-INTA. CAMBIOS DE VARIABLES:

Se procede a continuación igual que hemos hecho en el caso univariable con un modelo real, a modo de ejemplo. Hay que decir que el anterior es un modelo que no tiene nada que ver con el proceso estudiado en este proyecto. El que viene a continuación sí es el modelo de la planta real de generación distribuida a la que he llamado ETSI-INTA.

Siguiendo en todo momento el guión del capítulo 2, voy a desarrollar la predicción de salida del modelo que simula la red de generación distribuida.

He de recordar que el modelo de referencia de la bibliografía [2] responde a una instalación experimental existente en Escuela Técnica Superior de Ingenieros de la Universidad de Sevilla:

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + \begin{bmatrix} 8.136 & 5.958 \\ -15.289 & 0 \end{bmatrix} \begin{bmatrix} P_{H2}(k) \\ P_{GRID}(k) \end{bmatrix} + \begin{bmatrix} 5.958 \\ 0 \end{bmatrix} P_{NET} \quad (9-A1)$$

El mismo modelo expresado en intensidades, teniendo en cuenta que circulan todas por un BUS de 0.4 kV y que las potencias están expresadas en kW--multiplicamos los coeficientes de (9-A1) por 0.4--, se formula a continuación:

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + \begin{bmatrix} 3.254 & 2.383 \\ -6.116 & 0 \end{bmatrix} \begin{bmatrix} I_{H2}(k) \\ I_{GRID}(k) \end{bmatrix} + \begin{bmatrix} 2.383 \\ 0 \end{bmatrix} I_{NET} \quad (10-A1)$$

el cambio de coeficiente realizado es:

$$\begin{aligned} \text{base: } 400 \text{ V} \Rightarrow 1 \text{ kV equivale } \frac{1000}{400} \text{ A} \Rightarrow \text{coeficiente: } K \left(\frac{\%}{\text{kW}} \right) \Rightarrow \\ \Rightarrow K \left(\frac{\%}{\text{kW}} \right) \times \left(\frac{1 \text{ kW}}{\frac{1000}{400} \text{ A}} \right) = 0.4 K \left(\frac{\%}{\text{A}} \right) = K^* \end{aligned}$$

Y volvemos a multiplicar los coeficientes por los factores K1, K2, es decir, coeficientes de factor de escala,

para adaptar el modelo de partida al modelo final, según se explicó en la subsección 4.4.1.

Los coeficientes correspondientes al factor de escala son los siguientes:

→ Batería de acumuladores: $K1 = 36 / 17.6 = 2.05$

→ Depósito MHL: $K2 = 15 / 35 = 0.43$

El coeficiente K1 afecta a la primera fila del sistema de ecuaciones (10-A1), es decir a la variable de salida SOC, y el coeficiente de factor de escala K2 afecta a la segunda fila, variable MHL.

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + \begin{bmatrix} 6.671 & 4.885 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{H2}(k) \\ I_{GRID}(k) \end{bmatrix} + \begin{bmatrix} 4.885 \\ 0 \end{bmatrix} I_{NET} \quad (11-A1)$$

Vamos a hacer algunas transformaciones en la ecuación (11-A1) apoyándonos en la figura 1-A1. Se numeran los nodos para aplicar en ellos la ley de Kirchhoff.

I_{NET} representa la diferencia entre la energía producida en la red y la energía demandada; aplicando Kirchhoff en el nodo 632:

$$\rightarrow I_{NET} + I_{WINDSUN} + I_{CIRCUIT} = 0; \quad (12-A1)$$

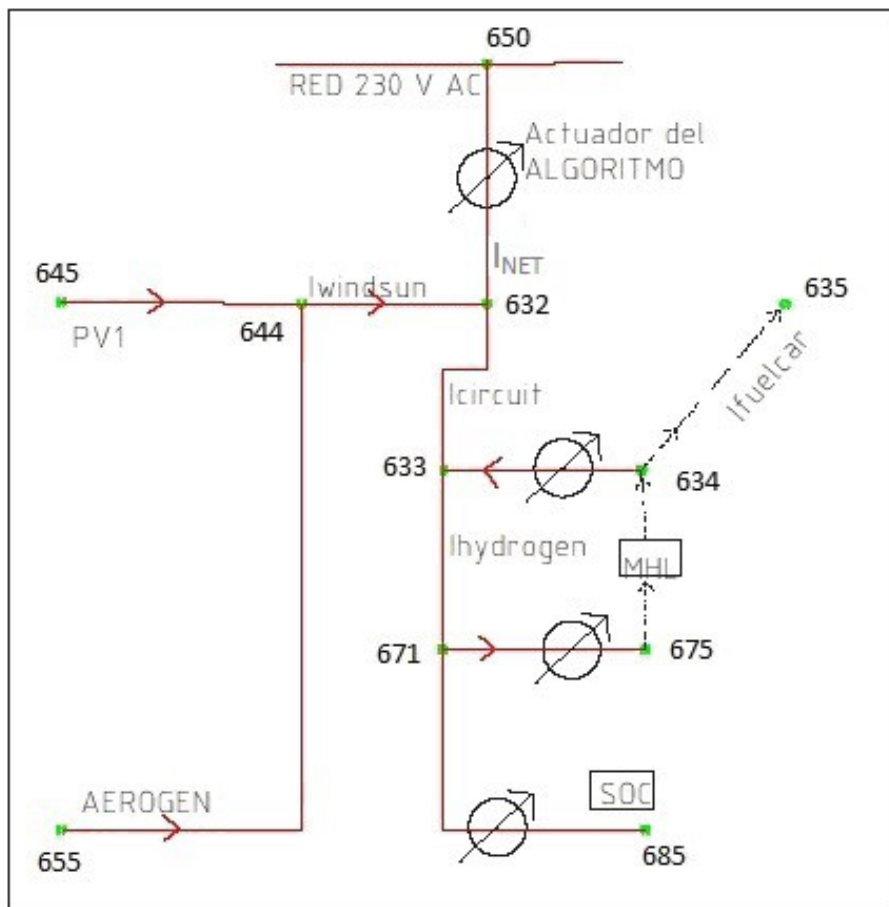


Figura A1-1. Relación entre variables del modelo.

A partir de la ecuación (11-A1) y del diagrama de la figura A1-1, podemos asociar las variables de ambos, teniendo en cuenta las siguiente hipótesis:

- $I_{WINDSUN}$ está simulada mediante fuente electrónica programable en el modelo (11-A1).
- si no hay demanda externa de H_2 ($I_{FUELCAR} = 0$), y si $I_{WINDSUN} = 0$, ambos modelos coinciden.
- en la hipótesis anterior I_{GRID} del primer modelo coincide con $I_{CIRCUIT}$ del segundo.
- además, por (12-A1), I_{NET} del primer modelo también coincide con $I_{CIRCUIT}$ del segundo.

Por tanto, para pasar del primer modelo al segundo lo que hacemos es desacoplar I_{GRID} de $I_{WINDSUN}$, para sumarle posteriormente esta última intensidad en el nodo 632—ecuación (12-A1)—con la particularidad de que se sumará en forma de perturbación aunque sea un dato conocido.

$$\begin{aligned} \begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} &= \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + \begin{bmatrix} 6.671 & 4.885 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{hydrogen}(k) \\ I_{circuit}(k) \end{bmatrix} + \begin{bmatrix} 4.885 \\ 0 \end{bmatrix} I_{circuit} \\ &\Leftrightarrow \\ \begin{bmatrix} \Delta y_1(k+1) \\ \Delta y_2(k+1) \end{bmatrix} &= \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{hydrogen}(k) \\ I_{circuit}(k) \end{bmatrix} \end{aligned} \quad (13-A1)$$

Después de todas las transformaciones preparatorias realizadas al modelo inicial, vamos buscando transformarlo a un modelo de sistema de generación distribuida que tiene en cuenta la entrada y salida de potencia de la red, a través de datos conocidos, $I_{FUELCAR}$, $I_{WINDSUN}$:

Por tanto, a la ecuación (13-A1) le sumamos los datos teniendo en cuenta:

- Volvemos a aplicar Kirchoff en el nodo 632 teniendo en cuenta que $I_{WINDSUN}$ no es nulo (12-A1).
- El parámetro $I_{FUELCAR}$ es un consumo de H_2 que se puede modelar.
- A efectos del modelo entrada/salida, $I_{FUELCAR}$ se comporta igual que $I_{HYDROGEN}$ (mismo coeficiente, es decir, un valor positivo de $I_{FUELCAR}$ supone una disminución en el nivel almacenado MHL).

$$\begin{aligned} \begin{bmatrix} \Delta y_1(k+1) \\ \Delta y_2(k+1) \end{bmatrix} &= \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{hydrogen}(k) \\ I_{circuit}(k) \end{bmatrix} + \begin{bmatrix} 4.885 \\ 0 \end{bmatrix} I_{windsun} + \begin{bmatrix} 0 \\ -2.630 \end{bmatrix} I_{fuelcar} \\ &\Leftrightarrow \\ \begin{bmatrix} \Delta y_1(k+1) \\ \Delta y_2(k+1) \end{bmatrix} &= \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{hydrogen}(k) \\ I_{circuit}(k) \end{bmatrix} + \begin{bmatrix} 0 & 4.885 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} I_{fuelcar} \\ I_{windsun} \end{bmatrix} \end{aligned} \quad (14-A1)$$

Resumiendo, sumamos un término que afecta a la primera fila del sistema de ecuaciones (13-A1) mediante $I_{WINDSUN}$ multiplicado por un coeficiente, y otro término que afecta a la segunda fila mediante $I_{FUELCAR}$ multiplicado por otro coeficiente. Los citados coeficientes vienen arrastrándose desde la ecuación (11-A1).

ENTRADA DE DATOS EN FORMA DE PERTURBACIONES:

Como ya se justificó en las hipótesis iniciales al desarrollo del algoritmo de Control Predictivo, la entrada de datos podría incluirse dentro de las perturbaciones medibles del modelo, pero se introduce junto con las no medibles y junto a la incertidumbre en errores de modelado.

$$\Delta y(t+k+1) = \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} u(t+k) + \frac{C(z^{-1})}{\Delta} e(t+k)$$

donde

$$u(t+k) = \begin{bmatrix} I_{hydrogen} \\ I_{circuit} \end{bmatrix}$$

y donde el polinomio de perturbación incluye la incertidumbre en entrada de datos

$$C(z^{-1}) = \begin{bmatrix} 1 + 4.885 I_{windsun} z^0 & -4.885 I_{windsun} z^{-1} \\ -1 - 2.630 I_{fuelcar} z^0 & -2.630 I_{fuelcar} z^{-1} \end{bmatrix} \quad (15-A1)$$

Por tanto, se ha elegido un polinomio (matriz) $C(z^{-1})$ para las perturbaciones, y posteriormente veremos por qué se han elegido tales expresiones en los polinomios de la matriz.

OBTENCIÓN DE LA PREDICCIÓN DE SALIDA:

Se trata de obtener la mejor expresión del predictor resolviendo la ecuación diofántica, y formular el problema completo, incluyendo matrices de restricciones y función objetivo.

Al principio de este capítulo vimos que la expresión del predictor es la siguiente:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})z^{-d}}{A(z^{-1})} u(t+k-1) + \frac{C(z^{-1})e(t+k-1)}{\Delta A(z^{-1})} \quad (16-A1)$$

Aunque en este caso estemos tratando el caso multivariable, y aunque A , B y C sean matrices polinomiales en lugar de polinomios, la formulación es la misma.

Según se vió en (15-A1) las matrices A y B se escribirán, para $k=1$:

$$B = \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix}; \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};$$

Consideraremos aquí también el retardo nulo, $d=0$.

Sin embargo ahora estamos en el caso de ruido coloreado, es decir $C(z^{-1})$ no vale la unidad; para obtener la expresión correspondiente a ruido blanco y resolver la ecuación diofántica como ya hemos hecho al principio del anexo, multiplicamos toda la ecuación (16-A1) por la inversa de $C(z^{-1})$, que expresamos C^{-1} .

$$A^*(z^{-1})y(t+k) = B^*(z^{-1})u(t+j-1) + (1/\Delta)e(t+k) \quad (17-A1)$$

Posteriormente se invertirá el cambio de variables, para así tener en cuenta el efecto del polinomio de perturbación C . Veamos la notación empleada para multiplicar por C^{-1} o por C :

$$C^{-1}(z^{-1}) = \begin{bmatrix} \frac{1}{1 + 4.885 I_{windsun} - 4.885 I_{windsun} z^{-1}} \\ \frac{1}{-1 - 2.630 I_{fuelcar} - 2.630 I_{fuelcar} z^{-1}} \end{bmatrix}$$

Ahora puedo explicar el motivo de elegir de esta forma los polinomios para $C(z^{-1})$; como aparecen dividiendo en esta última expresión no pueden valer cero, y por otro lado cuando los valores de $I_{WINDSUN}$, $I_{FUELCAR}$ (siempre positivos o cero) son nulos, la perturbación apenas afecta a la ecuación de predicción.

$$A^* = C^{-1} \Delta A; \quad B^* = C^{-1} \Delta B$$

nota:

voy a explicar en las siguientes líneas cómo actúa el operador por filas Δ que multiplica fila a fila el vector C por la correspondiente fila de la matriz, es decir, la componente 1 de $C(z^{-1})$ afecta sólo a la variable 1 de salida en la ecuación, y la componente 2 de C afecta sólo a la variable 2 de salida.

Lo anterior se puede expresar en notación Matlab, por ejemplo C^1 y B :

$$C^{-1} \Delta B = \begin{bmatrix} C^{-1}(1) * B(1, :) \\ C^{-1}(2) * B(2, :) \end{bmatrix}$$

Multiplicar C-1 por B mediante operador por filas:

Para obtener la expresión de la mejor predicción procedemos de igual forma que para el caso univariable, agrupando la ecuación (17-A1) con la ecuación diofántica.. Para empezar se muestra la forma de la ecuación diofántica, adaptada al cambio de variables explicado anteriormente:

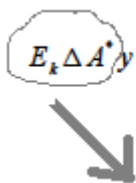
$$I = E_k^* \Delta A + z^{-k} F_k^* \tag{18-A1}$$

donde:

$$F_k^* = C^{-1} \Delta F_k \quad ; \quad G_k^* = C^{-1} \Delta G_k$$

A continuación operamos sobre la ecuación (17-A1) multiplicando en ambos miembros por $E_k \Delta$.

$$E_k \Delta A^* y(t+k) = E_k \Delta [B^* u(t+k-1) + e \frac{(t+k)}{\Delta}] \tag{19-A1}$$



La expresión marcada se puede obtener de (18-A1):

$$\begin{aligned} \rightarrow I &= E_k^* \Delta A + z^{-k} F_k^* \Leftrightarrow I = C^{-1} E_k \Delta A + z^{-k} F_k^* \Leftrightarrow I = E_k \Delta C^{-1} A + z^{-k} F_k^* \Leftrightarrow \\ \Leftrightarrow I &= E_k \Delta A^* + z^{-k} F_k^* \rightarrow \end{aligned}$$

$$\rightarrow E_k \Delta A^* = (I - z^{-k} F_k^*) ; \quad \text{sustituimos en (19-A1)}$$

$$(I - z^{-1} F_k^*) y(t+k) = E_k B^* \Delta u(t+k-1) + E_k e(t+k)$$

TIENE
GRADO k-1

El término $e(t+k)$ tiene grado k , mayor que la matriz de coeficientes que lo multiplica (grado $k-1$), luego está en el futuro y lo sacamos de la mejor predicción.

Por otro lado:

$$\begin{aligned} \rightarrow E_k B^* &= E_k C^{-1} B = C^{-1} E_k B = C^{-1} G_k \\ [\text{al igual que se hizo en (4-A1), se agrupa } G_k(z^{-1}) &= E_k(z^{-1}) B(z^{-1})] \\ \rightarrow F_k^* &= C^{-1} F_k \end{aligned}$$

$$\Leftrightarrow y(t+k) - C^{-1} F_k y(t) = C^{-1} G_k \Delta u(t+k-1) \Leftrightarrow$$

(multiplico toda la ecuación por C, con el operador por filas Δ)

$$\Leftrightarrow C \Delta \hat{y}(t+k|t) = G_k \Delta u(t+k-1) + F_k y(t) \tag{20-A1}$$

Para desarrollar el mejor predictor, resolveré la ecuación diofántica con los polinomios (matrices) G_k y F_k , de forma recursiva para $k=1,2$. Mediante la expresión (20-A1) nos hemos ahorrado tener que arrastrar el polinomio de perturbación en este proceso.

La ecuación diofántica se escribe para las matrices A , E_k , F_k en el caso multivariable,

$$I = E_k(z^{-1}) \Delta A + z^{-k} F_k(z^{-1})$$

resolviéndose en este caso sólo para un horizonte de predicción $N=2$, para poder exponer el desarrollo con claridad.

Como ya se dijo en el capítulo 2, a la hora de programar el algoritmo en Matlab conviene aumentar el horizonte de predicción.

$k=1$;

tomamos para arrancar las iteraciones el valor inicial de E_1 :

$$E_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};$$

y resolvemos la ecuación: $I = E_1 \Delta A + z^{-1} F_1 \Leftrightarrow$

$$\Leftrightarrow F_1 = z^{-1} (I - E_1 \Delta A)$$

\Leftrightarrow

$$\text{calculamos } \Delta A: \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \rightarrow \quad \Delta A = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \end{bmatrix};$$

$$F_1 = z^{-1} [I - E_1 \Delta A] = z^{-1} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \end{bmatrix} \right) =$$

$$= z^{-1} \begin{bmatrix} \cancel{1} - (\cancel{1} - z^{-1}) & 0 \\ 0 & \cancel{1} - (\cancel{1} - z^{-1}) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Damos un paso al siguiente intervalo; el valor de E_k ahora y en adelante se obtiene recursivamente a partir del paso anterior:

$$E_k = E_{k-1} + f_{(k-1),0} z^{-(k-1)} \quad (21-A1)$$

siendo $f_{(k-1),0}$ la matriz de coeficientes de grado 0, del polinomio (matriz) del paso anterior F_{k-1}

$k=2;$

la expresión (20-A1) se escribe: $E_2 = E_1 + f_{1,0} z^{-1} \Leftrightarrow$

\Leftrightarrow

$$\begin{aligned}
 E_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} z^{-1} = \begin{bmatrix} 1+z^{-1} & 0 \\ 0 & 1+z^{-1} \end{bmatrix} \\
 F_2 &= z^2 (I - E_2 \Delta A) = z^2 \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1+z^{-1} & 0 \\ 0 & 1+z^{-1} \end{bmatrix} \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \end{bmatrix} \right) = \\
 &= z^2 \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} (1+z^{-1})(1-\Delta) & 0 \\ 0 & (1+z^{-1})(1-\Delta) \end{bmatrix} \right) = \\
 &= z^2 \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1-z^{-2} & 0 \\ 0 & 1-z^{-2} \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
 \end{aligned}$$

Observe el lector que el grado de F_k siempre es igual al grado de A , en este caso 0.

$$E_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1+z^{-1} & 0 \\ 0 & 1+z^{-1} \end{bmatrix}; \quad (22-A1)$$

$$F_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix};$$

La matriz G_k se obtiene recursivamente, para $k=1,2$, como $G_k(z^{-1}) = E_k B$:

$$\begin{aligned}
 G_1 &= E_1 B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} = \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix}; \\
 G_2 &= E_2 B = \begin{bmatrix} 1+z^{-1} & 0 \\ 0 & 1+z^{-1} \end{bmatrix} \begin{bmatrix} 6.671 & 9.77 \\ -2.63 & 0 \end{bmatrix} = \begin{bmatrix} 6.671+6.671z^{-1} & 9.77+9.77z^{-1} \\ -2.63-2.63z^{-1} & 0 \end{bmatrix} \quad (23-A1)
 \end{aligned}$$

Una vez resueltas las matrices G_k, F_k , desarrollamos el predictor (20-A1) para $k=1,2$.

Como ha pasado con toda la formulación hasta ahora, hay que hacerlo de manera recursiva.

$$\hat{y}(t+k|t) = G_k \Delta u(t+k-1) + F_k y(t)$$

⇔

Comencemos el desarrollo primero sin polinomio $C(z^{-1})$, para posteriormente incluirlo en la expresión del predictor:

$$k=1;$$

$$\begin{bmatrix} \hat{y}_1(t+1) \\ \hat{y}_2(t+1) \end{bmatrix} = \begin{bmatrix} 6.671 & 9.770 \\ -2.630 & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$$

$$\begin{aligned} \hat{y}_1(t+1/t) &= 6.671 \Delta u_1(t) + 9.770 \Delta u_2(t) + y_1(t) \\ \hat{y}_2(t+1/t) &= -2.630 \Delta u_1(t) + y_2(t) \end{aligned}$$

$$k=2;$$

$$\begin{bmatrix} \hat{y}_1(t+2/t) \\ \hat{y}_2(t+2/t) \end{bmatrix} = \begin{bmatrix} 6.671+6.671z^{-1} & 9.770+9.770z^{-1} \\ -2.630-2.630z^{-1} & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t+1) \\ \Delta u_2(t+1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$$

$$\begin{aligned} \hat{y}_1(t+2/t) &= 6.671 \Delta u_1(t+1) + 6.671 \Delta u_1(t) + 9.770 \Delta u_2(t+1) + 9.770 \Delta u_2(t) + y_1(t) \\ \hat{y}_2(t+2/t) &= -2.630 \Delta u_1(t+1) - 2.630 \Delta u_1(t) + y_2(t) \end{aligned} \quad (24-A1)$$

Agrupando términos en las ecuaciones desarrolladas para $k=1, 2$, podemos expresar la predicción de salidas futuras de forma matricial más compacta.

$$\begin{bmatrix} \hat{y}_1(t+1) \\ \hat{y}_2(t+1) \\ \hat{y}_1(t+2) \\ \hat{y}_2(t+2) \end{bmatrix} = \begin{bmatrix} 6.671 & 9.770 & 0 & 0 \\ -2.630 & 0 & 0 & 0 \\ 6.671 & 9.770 & 6.671 & 9.770 \\ -2.630 & 0 & -2.630 & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \\ \Delta u_1(t+1) \\ \Delta u_2(t+1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \quad (25-A1)$$

Sobre la expresión del predictor obtenida anteriormente de forma matricial, añadimos el polinomio de perturbación de datos $C(z^{-1})$, multiplicando por filas como se ha explicado anteriormente:

$$C \Delta \hat{y}(t+k/t) = G_k \Delta u(t+k-1) + F_k y(t) \quad \Leftrightarrow$$

$$\begin{aligned} & \begin{bmatrix} \hat{y}_1(t+1) (1+4.885 I_{windsun} z^0 - 4.885 I_{windsun} z^{-1}) \\ \hat{y}_2(t+1) (-1-2.630 I_{fuelcar} z^0 - 2.630 I_{fuelcar} z^{-1}) \\ \hat{y}_1(t+2) (1+4.885 I_{windsun} z^0 - 4.885 I_{windsun} z^{-1}) \\ \hat{y}_2(t+2) (-1-2.630 I_{fuelcar} z^0 - 2.630 I_{fuelcar} z^{-1}) \end{bmatrix} = \\ & = \begin{bmatrix} 6.671 & 9.770 & 0 & 0 \\ -2.630 & 0 & 0 & 0 \\ 6.671 & 9.770 & 6.671 & 9.770 \\ -2.630 & 0 & -2.630 & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \\ \Delta u_1(t+1) \\ \Delta u_2(t+1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \end{aligned}$$

Pasamos al segundo miembro de la expresión los términos que estén en el pasado:

$$\begin{aligned}
 & \begin{bmatrix} \hat{y}_1(t+1) (1+4.885 I_{windrun}) \\ \hat{y}_2(t+1) (-1-2.630 I_{fuelcar}) \\ \hat{y}_1(t+2) (1+4.885 I_{windrun}) \\ \hat{y}_2(t+2) (-1-2.630 I_{fuelcar}) \end{bmatrix} = \\
 & \begin{bmatrix} 6.671 & 9.770 & 0 & 0 \\ -2.630 & 0 & 0 & 0 \\ 6.671 & 9.770 & 6.671 & 9.770 \\ -2.630 & 0 & -2.630 & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \\ \Delta u_1(t+1) \\ \Delta u_2(t+1) \end{bmatrix} + \\
 & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} \Omega_1 & 0 \\ 0 & \Omega_2 \\ \Omega_1 & 0 \\ 0 & \Omega_2 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}
 \end{aligned} \tag{26-A1}$$

donde hemos utilizado los parámetros:

$$\Omega_1 = (1+4.885 I_{windrun})$$

$$\Omega_2 = -(1+2.630 I_{fuelcar})$$

Analicemos un poco la expresión:

El primero de los dos términos de (26-A1) contiene la incógnita del control predictivo, es decir, la próxima señal de control que hay que mandar a la planta.

El segundo y tercer término contienen valores ya conocidos, pues están en el pasado.

Por otro lado, vemos que la variable de salida está multiplicada por una expresión que sólo depende de datos conocidos, en forma de perturbaciones y que podemos resumir en los parámetros Ω_1 , Ω_2 para simplificar la expresión.

Analizando la expresión en forma matricial, a continuación habría que pasar los citados parámetros al segundo miembro dividiendo por filas.

El predictor anterior no incluye término en función de entradas pasadas (una matriz G' que multiplica al vector $\Delta u(t-1)$ que está en el pasado); ello es debido a que la matrices $A(z^{-1})$, $B(z^{-1})$ de la función de transferencia tienen todos sus coeficientes con grado igual a cero.

Es fácil demostrar que si modificamos ligeramente la matriz B para tener un sistema de 1er orden:

$$B(z^{-1}) = \begin{bmatrix} 6.671+6.671 z^{-1} & 9.770+9.770 z^{-1} \\ -2.630-2.630 z^{-1} & 0 \end{bmatrix};$$

donde las unidades de la matriz de transferencia son: $\Delta \text{nivel}(\%) / I(\text{A})$

El predictor resultante es muy similar a 27-A1, con la particularidad de que añade precisamente un término que tiene en cuenta las entradas pasadas, como se verá en 28-A1.

Finalmente, como última manipulación antes de obtener la expresión definitiva del predictor que habría que

programar en el algoritmo informático, deshacemos el cambio de unidades para volver a expresar las entradas en términos de potencia en lugar de intensidad (BUS de corriente continua a 0.4 kV). Hay que dividir todos los coeficientes por 0.4; es el proceso inverso al descrito en 10-A1.

$$B(z^{-1}) = \begin{bmatrix} 16.676 + 16.676 z^{-1} & 24.425 + 24.425 z^{-1} \\ -6.575 - 6.575 z^{-1} & 0 \end{bmatrix};$$

donde las unidades de la matriz de transferencia son en este caso: $\Delta \text{nivel}(\%) / P \text{ (kW)}$

Los parámetros Ω_1 , Ω_2 son adimensionales, pero los coeficientes que contienen no lo son, y hay que cambiarlos de unidades; también deben dividirse por 0.4 y veamos por qué.

$\Rightarrow C(z^{-1})$ es adimensional;

$\Rightarrow C(z^{-1}) = 1 + K * I z^0 - z^{-1}$;

$\Rightarrow C(z^{-1}) \sim K * I$;

para que $C(z^{-1})$ sea adimensional,

el coeficiente K tiene que tener unidades inversas a I:

$\Rightarrow K \left(\frac{1}{\text{amperios}} \right)$;

\Rightarrow el cambio de coeficiente sería igual que si fueran unidades $\left(\frac{\%}{A} \right)$,

como ya se vió en 10-A1.

$$\begin{aligned} \Omega_1 &= (1 + 12.213 P_{\text{windman}}) \\ \Omega_2 &= -(1 + 6.575 P_{\text{fuelcar}}) \end{aligned} \quad (28-A1)$$

Para deducir cómo queda el PREDICTOR, una vez modificada (mejorada) la matriz $B(z^{-1})$, no se va a repetir todo el desarrollo. Habría que empezar desde 21-A1 resolviendo una nueva ecuación diofántica con la nueva matriz B y los nuevos coeficientes, expresados ahora en unidades de potencia, de forma recursiva para $k=1, 2$.

Como se ha dicho anteriormente, para aumentar la robustez del algoritmo es conveniente aumentar el horizonte de predicción, y se hará al menos hasta $N=3$ a la hora de programar en Matlab en este trabajo.

Se programan en el algoritmo las matrices resultantes para el predictor $y(t+k/t) = G u + f$, que se han obtenido siguiendo exactamente el mismo desarrollo anterior, desde hasta (26-A1) y para $k=1,2,3$.

Dicho desarrollo no se incluye por no aportar nada nuevo al contenido del Anexo 1, pero sí que se listan los códigos utilizados en el Anexo 2.

Resumiendo, el predictor para $k=1,2$ (con variables de entrada en unidades de potencia, kW):

$$\hat{y} = G u + f \quad \Leftrightarrow$$

⇔

$$\begin{bmatrix} \hat{y}_1(t+1) \\ \hat{y}_2(t+1) \\ \hat{y}_1(t+2) \\ \hat{y}_2(t+2) \end{bmatrix} = \begin{bmatrix} \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & 0 & 0 \\ \frac{6.575}{\Omega_2} & 0 & 0 & 0 \\ \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} \\ \frac{6.575}{\Omega_2} & 0 & \frac{6.575}{\Omega_2} & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \\ \Delta u_1(t+1) \\ \Delta u_2(t+1) \end{bmatrix} + \\
 + \begin{bmatrix} \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & 0 & 0 \\ \frac{6.575}{\Omega_2} & 0 & 0 & 0 \\ \frac{16.676}{\Omega_1} & \frac{24.425}{\Omega_1} & 0 & 0 \\ \frac{6.575}{\Omega_2} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \\ \Delta u_1(t-2) \\ \Delta u_2(t-2) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \quad (29-A1)$$

ANEXO 2: CÓDIGOS USADOS PARA LA PROGRAMACIÓN EN MATLAB® Y OPENDSS

Código A2-1: simulación del control predictivo usando únicamente Matlab.

```
close all;clc;clear
%algoritmo gpc para 2 entradas 2 salidas
%variables de entrada en unidades de potencia (kW)
w=[60;40];%referencia de salida
formato='%f';
ref1=w(1,1);
fid=fopen('REF_SOC.txt','wt');
    fprintf(fid,formato,ref1);
    fclose(fid);
ref2=w(2,1);
    fid=fopen('REF_MHL.txt','wt');
    fprintf(fid,formato,ref2);
    fclose(fid);
%pesos de ponderación del predictor
alpha=1;
lambda=1;
%valores de arranque
SOC=50;
    fid=fopen('SOC.txt','wt');
    fprintf(fid,formato,SOC);
    fclose(fid);
MHL=50;
    fid=fopen('MHL.txt','wt');
    fprintf(fid,formato,MHL);
    fclose(fid);
PH2=0;
    fid=fopen('PH2.txt','wt');
    fprintf(fid,formato,PH2);
    fclose(fid);
Pred=0;
    fid=fopen('Pred.txt','wt');
    fprintf(fid,formato,Pred);
    fclose(fid);
%MULTIPLICADORES PARA ENTRADA DE DATOS (series temporales horarias,
24+24 h)
t=1;
multiR=zeros(1,48);%multiplicadores para viento+sol
multiR(1,25:48)=[0 0 0.09 0 0.09 0 0.19 0.92 1.46 1.69 5.2 5.51 6.24 6.5
6.64 4.68 4.02 1.32 0.24 0.19 0.19 0.09 0.09 0];
multiH=zeros(1,48);%multiplicadores para periodos de recarga de H2
multiH(1,25:31)=ones(1,7);
multiH(1,42:48)=ones(1,7);
Prenov=multiR(t);%previsión aporte Energ. por periodos (dependiendo del
valor de t en cada salto)
Phibrid=multiH(t)*0.8;%previsión consumo H2 por periodos(dependiendo del
valor de t en cada salto)
%DATA [entrada1 entrada2 salida1 salida2]
DATA(t,:)= [0 0 SOC MHL];
t=2;%ojo!sin retardo y horizonte temporal k=3
DATA(t,:)= [0 0 SOC MHL];
```



```

G=[16.676/OM1    24.425/OM1    0    0    0    0
   6.575/OM2    0    0    0    0    0
   16.676/OM1    24.425/OM1    16.676/OM1    24.425/OM1    0    0
   6.575/OM2    0    6.575/OM2    0    0    0
   16.676/OM1    24.425/OM1    16.676/OM1    24.425/OM1    16.676/OM1
24.425/OM1
   6.575/OM2    0    6.575/OM2    0    6.575/OM2    0
];
%
Gp=[16.676/OM1    24.425/OM1
   6.575/OM2    0
   16.676/OM1    24.425/OM1
   6.575/OM2    0
   16.676/OM1    24.425/OM1
   6.575/OM2    0    ];
%
Fp=[ 1    0
     0    1
     1    0
     0    1
     1    0
     0    1];
%
f=Gp*u_t_1+Fp*y_t;
T6=[eye(2) zeros(2) zeros(2);eye(2) eye(2) zeros(2);eye(2) eye(2)
eye(2)];
R =[eye(6)
   -eye(6)
     T6
   -T6
     G
   -G];
c1=[cotasup_u;cotasup_u;cotasup_u];
c2=[-cotainf_u;-cotainf_u;-cotainf_u];
c3=[cotasup_P-u_t_1;cotasup_P-u_t_1;cotasup_P-u_t_1];
c4=[-cotainf_P+u_t_1;-cotainf_P+u_t_1;-cotainf_P+u_t_1];
y_f1=[cotasup_y;cotasup_y;cotasup_y]-f;
y_f2=-[cotainf_y;cotainf_y;cotainf_y]+f;
c=[c1;c2;c3;c4;y_f1;y_f2];
%actualización de la función objetivo (J=1/2*u'Hu+bu+f0)
w3=[w;w;w];
f0=[f-w3]'*[f-w3];
b=2*[f-w3]'*G;
H=2*alpha*eye(6)*G'*G+lambd*eye(6);
%resolución del problema de optimización cuadrática
%transformamos la expresión F.O : (J= 1/2*u'Hu+ g*u)
g=[[f-w3]'*(2*G+inv(G')*(G'*G+lambd*eye(6)))]';
[u,fval,exitflag]=quadprog(H,g,R,c);
if exitflag<0
    fprintf('\nOjo! Se están obteniendo SOLUCIONES NO
ADMISIBLES\n(posible violación de restricciones)\n');
end
%captura de nuevos datos para el predictor
format short
PH2=DATA(t,1)+u(1);
    fid=fopen('PH2.txt','wt');
    fprintf(fid,formato,PH2);
    fclose(fid);
Pred=DATA(t,2)+u(2);
    fid=fopen('Pred.txt','wt');
    fprintf(fid,formato,Pred);
    fclose(fid);
%predictor

```

```

y=G*u+f;
SOC=y(1);
    fid=fopen('SOC.txt','wt');
    fprintf(fid,formato,SOC);
    fclose(fid);
MHL=y(2);
    fid=fopen('MHL.txt','wt');
    fprintf(fid,formato,MHL);
    fclose(fid);
%representación del comportamiento en dos días consecutivos
tau=[1:t];
plot(tau-24,DATA(:,2),'-r',tau-24,0.1*DATA(:,4),'-c',tau-24,DATA(:,1),'-
b',tau-24,0.36*DATA(:,3),'-m');
xlabel('últimas 24 h) ***** (próximas 24 h)');
ylabel('SALIDA info NIVELES: Baterías(kW.h), Tanque Hidruros
Metálicos(x10Nm3)');
legend('Pcarga/descarga bat (kW)', 'MHL', 'Ppila/electrolizador
(kW)', 'SOC');
hold on
plot(tau,0,tau,5);
pause(0.05);
if t==48
    NEWDATA=DATA(t-24:t,:);
    DATA=zeros(24,6);
    DATA=NEWDATA;
    t=25;
    hold off
end
Prenov=multiR(t);%previsión aporte Energ. por periodos (dependiendo del
valor de t en cada salto)
Phibrid=multiH(t)*0.8;%previsión consumo H2 por periodos(dependiendo del
valor de t en cada salto)
end

```

Código A2-2: ajuste estocástico de curvas de producción eólica esperada.

```

clear;clc;close all
%datos U=[u1 u2 d];
%u1,u2 son mediciones
%d son los valores esperados más probables ordenados de menor a mayor
[U]= [1.2777    1.4997    0
    1.2918    1.5210    0
    1.2918    1.6311    0
    1.2636    1.5259    0
    1.2636    1.6311    0
    1.3340    1.4206    0.1000
    1.6016    1.5259    0.1000
    1.9114    1.5204    0.1000
    2.2071    1.7307    0.1000
    2.5168    1.7868    0.2000
    2.9252    2.1582    0.2000
    3.2913    2.1969    0.2000
    3.4039    2.3193    0.2000
    3.4180    2.6852    0.4000
    3.4462    2.7239    0.4000
    3.4602    2.7229    0.4000
    3.3194    2.3246    0.7000
    3.1364    2.1197    0.7000
    2.8970    1.9634    0.7000
    2.6999    1.8307    1.3000
    2.4887    1.4874    1.3000
    2.1648    1.5646    1.8000

```

```

    1.8128    1.4539    2.3000
    1.5453    1.4997    2.7000];
u1=U(:,1);
u2=U(:,2);
d=U(:,3); %vector ordenado
n=length(d);
u(:,1)=(u1(:,1)+u2(:,1))/2;
%definir el n° de iteraciones para ajustar el vector de valores de
%producción esperados
espe=d;
for i=1:n %recorre posiciones de u
    resto=ones(1,n)*1e3;
    for j=1:n %recorre posiciones de valores esperados disponibles
        if espe(j,1)<0
            else
                resto(j)=abs(u(i,1)-espe(j,1)-min(u));
                %ojo! con el fondo de escala min(u)
            end
        end
    end
    m=min(resto);
    for j=1:n
        if resto(j)==m
            marca=j;
            marcados(i)=marca;
        end
    end
    espe(marca,1)=-1;
end
for i=1:n
    v(i,1)=d(marcados(i),1);
end
%fin de primera estimación de producción esperada
%definir el n° de iteraciones para ajustar el vector de valores de
%producción esperados (2ª estimación)
it=3e3;
for j=1:it
    w=rando(d);%llamada a función, ordena aleatoriamente vector
    V(:,j)=w;
    R(:,j)=abs(v(:,1)-w(:,1));
    %minimicemos el error y la variación del error
    e(1,j)=max(R(:,j));
    f=zeros(n,1);
    for i=1:n-1
        f(i+1,1)=R(i+1,j)-R(i,j);
    end
    E(1,j)=max(f);
    %introducir un término que penaliza que el primer y último valor
de
    %producción esperada no sean cero (mínimos de ambas curvas
coinciden)
    if V(1,j)==0 & V(n,j)==0
        P(1,j)=0;
    else
        P(1,j)=100;
    end
end
EV(1,:)=e(1,:)+E(1,:)+P(1,:);
m=min(EV);
for i=1:it
    if EV(1,i)==m
        marca=i;
    end
end
end

```

```

%fin de iteraciones de la 2ª estimación
w=V(:,marca);
t=[1:n];
plot(t,u,'c-o',t,v,'r-*',t,w,'b--');
xlabel('Periodo horario (24 h)');
ylabel('Producción eólica (Kw)');
legend('datos anteriores','producción esperada 1','producción esperada 2');

```

Código A2-3: resolución de la red *inta11* mediante la interfaz de OpenDSS.

```

clear

new circuit.inta11 basekv=4.16 pu=1 MVAsc1=21000 MVAsc3=21000

New XYCurve.MyPvsT npts=4 xarray=[0 25 75 100] yarray=[1.2 1.0 0.8 0.6]

New XYCurve.MyEff npts=4 xarray=[.1 .2 .4 1.0] yarray=[.86 .9 .93 .97]

New Loadshape.MyIrrad npts=1 interval=1 mult=[0.8]

New Tshape.MyTemp 1 1.0 temp=[25]

new loadshape.wind 1 1.0 mult=[0.5]

!new loadshape.fc 1 1.0 mult=[0.8] !pila on
new loadshape.fc 1 1.0 mult=[0] !pila off

!new loadshape.store 1 1.0 mult=[0.9] !descarga
new loadshape.store 1 1.0 mult=[-0.9] !carga

new loadshape.ez 1 1.0 mult=[0.62] !electrolizador on
!new loadshape.ez 1 1.0 mult=[0] !electrolizador off

new loadshape.hibrid 1 1.0 mult=[1]

new transformer.Sub phases=1 windings=2 Xhl=2
~ wdg=1 bus=sourcebus conn=LN kv=4.16 kva=500 %r=0.05
~ wdg=2 bus=650 conn=LN kv=0.4 kva=500 %r=0.05

new transformer.trafo1 phases=1 windings=2 Xhl=2
~ wdg=1 bus=632r conn=LN kv=0.4 kva=50 %r=0.05
~ wdg=2 bus=632 conn=LN kv=0.68 kva=50 %r=0.05

new transformer.trafo2 phases=1 windings=2 Xhl=2
~ wdg=1 bus=671r conn=LN kv=0.4 kva=50 %r=0.05

```



```
~ wdg=2 bus=671 conn=LN kv=0.087 kva=50 %r=0.05
```

```
New linecode.LCD1 nphases=1 BaseFreq=50
```

```
~ rmatrix = [0.00107]
```

```
New linecode.LCD2 nphases=1 BaseFreq=50
```

```
~ rmatrix = [0.00024]
```

```
New linecode.LCD3 nphases=1 BaseFreq=50
```

```
~ rmatrix = [0.00684]
```

```
New linecode.LCD4 nphases=1 BaseFreq=50
```

```
~ rmatrix = [0.00171]
```

```
New linecode.LCD5 nphases=1 BaseFreq=50
```

```
~ rmatrix = [0.00285]
```

```
new capacitor.cap1 bus1=634 phases=1 conn=delta kvar=0.01 kv=0.087
```

```
new line.L0 Phases=1 Bus1=650 Bus2=632r linecode=LCD1 length=50e-3
```

```
new line.L1 Phases=1 Bus1=632 Bus2=633 linecode=LCD1 length=1e-3
```

```
new line.L2 Phases=1 Bus1=633 Bus2=685 linecode=LCD1 length=5e-3
```

```
new line.L3 Phases=1 Bus1=632r Bus2=671r linecode=LCD1 length=1e-3
```

```
new line.L4 Phases=1 Bus1=671 Bus2=634 linecode=LCD2 length=20e-3
```

```
new line.L5 Phases=1 Bus1=634 Bus2=675 linecode=LCD2 length=1e-3
```

```
new line.L6 Phases=1 Bus1=634 Bus2=635 linecode=LCD3 length=1e-3
```

```
new line.L7 Phases=1 Bus1=632 Bus2=644 linecode=LCD4 length=1e-3
```

```
new line.L8 Phases=1 Bus1=644 Bus2=655 linecode=LCD4 length=50e-3
```

```
new line.L9 Phases=1 Bus1=644 Bus2=645 linecode=LCD5 length=25e-3
```

```
New Load.SL1 Bus1=675 Phases=1 Conn=LN Model=1
```

```
~ kV=0.087 kw=15 kvar=0 duty=eZ basefreq=50
```

```
New Load.SL2 Bus1=635 Phases=1 Conn=LN Model=1
```

```
~ kV=0.087 kW=2.4 pf=1 duty=hibrid basefreq=50
```

```
new Storage.Battery
```

```
Storage.Battery.bus1=685 phases=1 kw=10.5 kvar=0 kV=0.68 kWrated=10 kWhrated=36
```

```
~ kWhstored=15 dispmode=follow conn=LN duty=store model=1 basefreq=50
```

```
New PVSystem.PV phases=1 bus1=645 kV=0.68 kva=16.5 Pmpp=16.5 irrad=0.8 temperature=25
```

```
~ pf=1 basefreq=50 effcurve=Myeff P-TCurve=MyPvsT conn=LN Duty=MyIrrad TDuty=MyTemp
```

```
New generator.genwind phases=1 bus1=655 kV=0.68 kW=15 kvar=0 conn=LN
```

```
~ duty=wind basefreq=50 Model=1
```

```
New generator.fuelcell phases=1 bus1=634 kV=0.087 kw=3.6 kvar=0
```

```
~ conn=LN duty=fc basefreq=50
```



```

T=temper(hora);
W=windy(hora);
csvwrite('circuit.csv',-P_circuit/4);%adimensionalizado con la potencia
máxima real en batería
csvwrite('SOC.csv',nivelSOC);
csvwrite('MHL.csv',nivelMHL);
csvwrite('SOC.csv',nivelSOC);
csvwrite('Pfuelcar.csv',F);
csvwrite('irradi.csv',I);
csvwrite('temperatura.csv',T);
csvwrite('hourwind.csv',W);
%llamada a función
%RESULTADO=simula24(P_hydrogen,P_circuit,nivelSOC,nivelMHL,hora);
DSSText.Command='new loadshape.MyIrrad 1 1 mult=(file=irradi.csv)';
DSSText.Command='new tshape.MyTemp 1 1 temp=(file=temperatura.csv)';
DSSText.Command='new loadshape.wind 1 1 mult=(file=hourwind.csv)';
if P_hydrogen>0
    P_fuelcell=-P_hydrogen*0.8/1.2; %multiplicador adimensionalizado con
el nominal de la pila y su rendimiento
    csvwrite('fuelcell.csv',P_fuelcell);
    fprintf('\nPila de combustible ON\n');
    DSSText.Command='new loadshape.fcell 1 1 mult=(file=fuelcell.csv)';
    DSSText.Command='new loadshape.electroliz 1 1 mult=[0]';
else
    P_electroliz=P_hydrogen*0.62/5;%multiplicador adimensionalizado con
el nominal de electrolizador y su rendimiento
    csvwrite('electroliz.csv',P_electroliz);
    fprintf('\nElectrolizador ON\n');
    DSSText.Command='new loadshape.electroliz 1 1
mult=(file=electroliz.csv)';
    DSSText.Command='new loadshape.fcell 1 1 mult=[0]';
end
DSSText.Command='new loadshape.fuelcar 1 1 mult=(file=Pfuelcar.csv)';
%aunque no se puede relacionar cuantitativamente, SÍ se hace
%cuantitativamente, entre P_circuit y la acción de carga o descarga de
%batería
DSSText.Command='new loadshape.bateria 1 1 mult=(file=circuit.csv)';
if P_circuit>0
    fprintf('\nBatería cargando\n');
else
    fprintf('\nBatería descargando\n');
end
%definición de objetos y elementos del circuito
DSSText.Command='new transformer.sub phases=1 windings=2 xhl=2';
DSSText.Command='~ wdg=1 bus=sourcebus conn=LN kv=4.16 kva=500
%r=0.05';
DSSText.Command='~ wdg=2 bus=650 conn=LN kv=0.4 kva=500
%r=0.05';
DSSText.Command='new transformer.trafo1 phases=1 windings=2 xhl=2';
DSSText.Command='~ wdg=1 bus=632r conn=LN kv=0.4 kva=50 %r=0.05';
DSSText.Command='~ wdg=2 bus=632 conn=LN kv=0.68 kva=50 %r=0.05';
DSSText.Command='new transformer.trafo2 phases=1 windings=2 xhl=2';
DSSText.Command='~ wdg=1 bus=671r conn=LN kv=0.4 kva=50 %r=0.05';
DSSText.Command='~ wdg=2 bus=671 conn=LN kv=0.087 kva=50 %r=0.05';
DSSText.Command='new linecode.LCD1 nphases=1 basefreq=50
rmatrix=[1.07]';
DSSText.Command='new linecode.LCD2 nphases=1 basefreq=50
rmatrix=[0.24]';
DSSText.Command='new linecode.LCD3 nphases=1 basefreq=50
rmatrix=[6.84]';
DSSText.Command='new linecode.LCD4 nphases=1 basefreq=50
rmatrix=[1.71]';
DSSText.Command='~ new linecode.LCD5 nphases=1 basefreq=50

```

```

rmatrix=[2.85]';
DSSText.Command='new capacitor.cap1 bus1=634 phases=1 conn=delta
kvar=0.01 kv=0.087';
DSSText.Command='new line.L0 phases=1 bus1=650 bus2=632r linecode=LCD1
length=50e-3';
DSSText.Command='new line.L1 phases=1 bus1=632 bus2=633 linecode=LCD1
length=1e-3';
DSSText.Command='new line.L2 phases=1 bus1=633 bus2=685 linecode=LCD1
length=5e-3';
DSSText.Command='new line.L3 phases=1 bus1=632r bus2=671r linecode=LCD1
length=1e-3';
DSSText.Command='new line.L4 phases=1 bus1=671 bus2=634 linecode=LCD2
length=20e-3';
DSSText.Command='new line.L5 phases=1 bus1=634 bus2=675 linecode=LCD2
length=1e-3';
DSSText.Command='new line.L6 phases=1 bus1=634 bus2=635 linecode=LCD3
length=1e-3';
DSSText.Command='new line.L7 phases=1 bus1=632 bus2=644 linecode=LCD4
length=1e-3';
DSSText.Command='new line.L8 phases=1 bus1=644 bus2=655 linecode=LCD4
length=50e-3';
DSSText.Command='new line.L9 phases=1 bus1=644 bus2=645 linecode=LCD5
length=25e-3';
DSSText.Command='new load.SL1 bus1=675 phases=1 conn=LN model=1 kv=0.087
kw=15 pf=1 duty=electroliz basefreq=50';
DSSText.Command='new load.SL2 bus1=635 phases=1 conn=LN model=1 kv=0.087
kw=2.4 pf=1 duty=fuelcar basefreq=50';
DSSText.Command='new storage.battery';
DSSText.Command='storage.battery.bus1=685 phases=1 conn=LN model=1
kv=0.68 kw=10.5';
DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=20 pf=1 duty=bateria
basefreq=50 dispmode=follow';
DSSText.Command='new PVsystem.PV phases=1 bus1=645 kv=0.68 kva=16.5
Pmpp=16.5 irrads=0.8 temperature=25';
DSSText.Command='~ pf=1 basefreq=50 effcurve=Myeff P-TCurve=MyPVst
conn=LN duty=MyIrrad tduty=MyTemp';
DSSText.Command='new generator.genwind bus1=655 phases=1 conn=LN model=1
kv=0.68 kw=12 pf=1 duty=wind basefreq=50';
DSSText.Command='new generator.fuelcell bus1=634 phases=1 conn=LN
model=1 kv=0.087 kw=3.6 pf=1 duty=fcell basefreq=50';
DSSText.Command='new monitor.nivel_bateria storage.battery 1 mode=3';
DSSText.Command='new monitor.Pbateria storage.battery 1 mode=1';
DSSText.Command='new monitor.Ielectro load.SL1 1 mode=0';
DSSText.Command='new monitor.Ifuelcell generator.fuelcell 1 mode=0';
DSSText.Command='new monitor.Psund PVsystem.PV 1 mode=1';
DSSText.Command='new monitor.Pwind generator.genwind 1 mode=1';
DSSText.Command='set voltagebases=(4.16 0.68 0.4 0.087)';
DSSText.Command='calc voltagebases';
DSSText.Command='solve';
DSSText.Command='solve mode=duty stepsize=60 number=60 hour=0 sec=0';
DSSText.Command='export monitor object=nivel_bateria';
DSSText.Command='export monitor object=Pbateria';
DSSText.Command='export monitor object=Ielectro';
DSSText.Command='export monitor object=Ifuelcell';
%obtención de datos de monitores
RESULTADO1=csvread('intall_mon_Ielectro.csv',1,1);
RESULTADO2=csvread('intall_mon>Ifuelcell.csv',1,1);
RESULTADO3=csvread('intall_mon_nivel_bateria.csv',1,1);
RESULTADO4=csvread('intall_mon_Pbateria.csv',1,1);
%preparación de datos
[L,N]=size(RESULTADO1);
cap=1; %capacidad depósito hidruros metálicos Nm3
nivel_0=0.5; %nivel inicial del depósito Nm3

```

```

I_electroliz=RESULTADO1(:,6);
I_fuelcell=RESULTADO2(:,6);
F=9.6496e4; %constante de Faraday
Nez=30; %número de celdas del electrolizador
Afc=1.48; %área de intercambio en la pila de combustible
%flujos (conversión de unidades litros/s a unidades Nm3/min)
m_ez=I_electroliz*Nez*60/(1000*F);
m_fc=I_fuelcell*60/(2000*Afc*F);
%actualización de la variable de estado MHL
nivel=zeros(L,1);
nivel(1)=nivel_0;
for i=2:L
    nivel(i)=nivel(i-1)+m_ez(i)-m_fc(i);
end
%representación de datos
SOC1=RESULTADO3(1,2)/0.36;
SOC2=RESULTADO3(L,2)/0.36;
MHL1=nivel(1)*100;
MHL2=nivel(L)*100;
Pcircuit=RESULTADO4(L,2);
if P_circuit<0
    Pcircuit=-Pcircuit;
end
I_hydrogen=RESULTADO2(L,6)-RESULTADO1(L,6);
fprintf('\n-----\n');
fprintf('\nValores de las variables de entrada principio y final del
periodo (1 hora):\n');
fprintf('\nP_circuit=%f(kW)  I_hydrogen=
%f(A)\n',P_circuit,P_hydrogen/0.05);
fprintf('\nP_circuit=%f(kW)  I_hydrogen=%f(A)\n',Pcircuit,I_hydrogen);
fprintf('\n-----\n');
fprintf('\nValores de las variables de estado(salida) al principio y
final del periodo:\n');
fprintf('\nSOC=%f(o/o)  MHL=%f(o/o) \n',SOC1,MHL1);
fprintf('\nSOC=%f(o/o)  MHL=%f(o/o) \n',SOC2,MHL2);
fprintf('\n-----\n');

```

Código A2-5: programa de comunicación entre control predictivo y simulación OpenDSS.

(*) nota: se incluye a continuación el programa ejecutable, que contiene una llamada a función ("simula24"). Al final del script ejecutable, seguidamente se lista la función.

```

close all;clc;clear
%reseteo de contadores de energía:
Gen=0;Export=0;Hibrid=0;tiempo=0;
%algoritmo gpc para 2 entradas 2 salidas
%variables de entrada en unidades de potencia (kW)
w=[50;40];%referencia de salida
%pesos de ponderación del predictor
alpha=0.8;
lambda=0.8;
%valores de arranque
formato='%f';
SOC=25;
    fid=fopen('SOC.txt','wt');
    fprintf(fid,formato,SOC);
    fclose(fid);
MHL=50;
    fid=fopen('MHL.txt','wt');
    fprintf(fid,formato,MHL);
    fclose(fid);
PH2=0;

```

```

        fid=fopen('PH2.txt','wt');
        fprintf(fid,formato,PH2);
        fclose(fid);
Pred=0;
        fid=fopen('Pred.txt','wt');
        fprintf(fid,formato,Pred);
        fclose(fid);
%MULTIPLICADORES PARA ENTRADA DE DATOS (series temporales horarias,
24+24 h)
t=1;
T=2;
multiR=zeros(1,48);%multiplicadores para viento+sol
multiR(1,25:48)=[0 0 0.09 0 0.09 0 0.19 0.92 1.46 1.69 5.2 5.51 6.24 6.5
6.64 4.68 4.02 1.32 0.24 0.19 0.19 0.09 0.09 0];
multiH=zeros(1,48);%multiplicadores para periodos de recarga de H2
multiH(1,25:30)=ones(1,6);
multiH(1,43:48)=ones(1,6);
Prenov=multiR(T);%previsión aporte Energ. por periodos (dependiendo del
valor de t en cada salto)
Phibrid=multiH(T)*0.8;%previsión consumo H2 por periodos(dependiendo del
valor de t en cada salto)
%DATA [entrada1 entrada2 salida1 salida2]
DATA(t,:)= [0 0 SOC MHL];
t=2;%ojo!sin retardo y horizonte temporal k=3
DATA(t,:)= [0 0 SOC MHL];
while t>0
    t=t+1;
    T=T+1;
    %actualización de datos
    fid=fopen('SOC.txt','r');
    SOC=fscanf(fid,formato);
    fclose(fid);
    fid=fopen('MHL.txt','r');
    MHL=fscanf(fid,formato);
    fclose(fid);
    fid=fopen('PH2.txt','r');
    PH2=fscanf(fid,formato);
    fclose(fid);
    fid=fopen('Pred.txt','r');
    Pred=fscanf(fid,formato);
    fclose(fid);
if PH2==0 | PH2>0
    Ppila=PH2;
    Pelect=0;
else
    Ppila=0;
    Pelect=PH2;
end
if Pred==0 | Pred>0
    Pcarga=Pred;
    Pdescarga=0;
else
    Pcarga=0;
    Pdescarga=Pred;
end
fprintf('\n*****\n');
fprintf('LOS DATOS ACTUALES DE LA PLANTA SON LOS SIGUIENTES:\n\n');
fprintf('\tNIVEL SOC\tConsumo H2\t\t\tNIVEL MHL\tENERGÍA renovables\n');
fprintf('%f(o/o)\t%f(kW)\t\t%f(o/o)\t%f(kW)\n',SOC,-Phibrid,MHL,Prenov);
fprintf('*****\n\n');
DATA(t,:)=[PH2 Pred SOC MHL];
%algoritmo gpc: linealizar y resolver Min J, s.a. R*u<=c
u_t_1=[DATA(t,1)-DATA(t-1,1);DATA(t,2)-DATA(t-1,2)];

```

```

y_t=[DATA(t,3);DATA(t,4)];%datos de entrada y salida pasados
%parámetros usados en la formulación (representan las perturbaciones
%externas)
OM1=1+12.213*Prenov;
OM2=-1-6.575*Phibrid;
%las matrices del algoritmo están resueltas para horizonte k=2
G=[16.676/OM1    24.425/OM1    0    0    0    0
   6.575/OM2    0    0    0    0    0
   16.676/OM1    24.425/OM1    16.676/OM1    24.425/OM1    0    0
   6.575/OM2    0    6.575/OM2    0    0    0
   16.676/OM1    24.425/OM1    16.676/OM1    24.425/OM1    16.676/OM1
24.425/OM1
   6.575/OM2    0    6.575/OM2    0    6.575/OM2    0
];
%
Gp=[16.676/OM1    24.425/OM1
   6.575/OM2    0
   16.676/OM1    24.425/OM1
   6.575/OM2    0
   16.676/OM1    24.425/OM1
   6.575/OM2    0    ];
%
Fp=[ 1    0
     0    1
     1    0
     0    1
     1    0
     0    1];
%
f=Gp*u_t_1+Fp*y_t;
w3=[w;w;w];
b=[2*[f-w3]'*G]';
H=2*[alpha*eye(6)*G'*G+lambda*eye(6)];
%resolución del problema sin restricciones
u=-inv(H)*b;
%captura de nuevos datos para el predictor
format short
PH2=DATA(t,1)+u(1);
    fid=fopen('PH2.txt','wt');
    fprintf(fid,formato,PH2);
    fclose(fid);
Pred=DATA(t,2)+u(2);
    fid=fopen('Pred.txt','wt');
    fprintf(fid,formato,Pred);
    fclose(fid);
%predictor
y=G*u+f;
SOC=y(1);
    fid=fopen('SOC.txt','wt');
    fprintf(fid,formato,SOC);
    fclose(fid);
MHL=y(2);
    fid=fopen('MHL.txt','wt');
    fprintf(fid,formato,MHL);
    fclose(fid);
%representación del comportamiento en dos días consecutivos
tau=[1:t];
plot(tau,DATA(:,2),'-r',tau,DATA(:,1),'-b',tau,0.1*DATA(:,4),'-
g',tau,0.1*w(2),'-+k');
xlabel('tiempo (saltos en h)');
ylabel('SALIDA info NIVEL Tanque Hidruros Metálicos( /10Nm3 )');
legend('Pcarga/descarga bat (kW)','Ppila/electrolizador (kW)','MHL');
hold on

```

```

plot(tau,0,tau,5);
pause(0.05);
if T<5
    start=0;
else
    start=1;%iniciamos el ciclo para entrada de datos
end
if T==5 | T==49
    T=25;
end
if t==48
    return
end
Prenov=multiR(T);%previsión aporte Energ. por periodos (dependiendo del
valor de t en cada salto)
Phybrid=multiH(T)*0.8;%previsión consumo H2 por periodos(dependiendo del
valor de t en cada salto)
if start==1
    hora=T-24;
    RESULTADO=simula24(DATA(t,1),DATA(t,2),SOC,MHL,hora);
    P_circuit=RESULTADO(1);
    Phydrogen=0.05*RESULTADO(2);
    nivelSOC=RESULTADO(4);
    nivelMHL=RESULTADO(6);
    %actualización de contadores de energía
    Gen=Gen+RESULTADO(7);
    Export=Export+RESULTADO(8);
    Hibrid=Hibrid+RESULTADO(9);
    fprintf('-----\n');
    fprintf('\nAcumulado de energía procedente de renovables:   %g
kW.h\n',Gen);
    fprintf('Acumulado de energía exportada a la red exterior: %g
kW.h\n',Export);
    fprintf('Acumulado de energía cedida al vehículo híbrido:  %g
kW.h\n',Hibrid);
    plot(t,0.1*nivelMHL,'*g',t,Phydrogen,'*b',t,P_circuit,'*r')
    SOC=nivelSOC;%corrección por OpenDSS de la señal de salida
    fid=fopen('SOC.txt','wt');
    fprintf(fid,formato,SOC);
    fclose(fid);
    MHL=nivelMHL;%corrección por OpenDSS de la señal de salida
    fid=fopen('MHL.txt','wt');
    fprintf(fid,formato,MHL);
    fclose(fid);
    PH2=Phydrogen;%corrección por OpenDSS de la señal de entrada
    fid=fopen('PH2.txt','wt');
    fprintf(fid,formato,PH2);
    fclose(fid);
    Pred=P_circuit;%corrección por OpenDSS de la señal de entrada
    fid=fopen('Pred.txt','wt');
    fprintf(fid,formato,Pred);
    fclose(fid);
end
end
**LLAMADA A FUNCIÓN:
function RESULTADO=simula24(P_hydrogen,P_circuit,nivelSOC,nivelMHL,hora)
P_fuelcar=[1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1];
irrad=[0 0 0 0 0 0 0 0.55 0.8 1.03 3.98 4.29 4.56 4.35 4.11 4.02 3.65
0.95 0.05 0 0 0 0 0];
temper=[9 9 9 9 9 9 9 10 15 15 25 25 25 25 25 25 25 25 15 9 9 9 9 9];
windy=[0 0 0.09 0 0.09 0 0.19 0.37 0.66 0.66 1.22 1.22 1.68 2.15 2.53
0.66 0.37 0.37 0.19 0.19 0.19 0.09 0.09 0];

```



```

H=P_fuelcar(hora);
I=irrad(hora)/5.5;%adimensionalizado con la potencia nominal
T=temper(hora);
W=windy(hora)/5;%adimensionalizado con la potencia nominal
DSSObj=actxserver('OpenDSSEngine.DSS');
DSSObj.Start(0);
DSSText=DSSObj.Text;
DSSCircuit=DSSObj.ActiveCircuit;
DSSSolution=DSSCircuit.Solution;
DSSText.Command='clear';
DSSText.Command='new circuit.intall basekv=4.16 pu=1 MVAsc1=21000
MVAsc3=21000';
DSSText.Command='new XYCurve.MyPVst npts=4 xarray=[0 25 75 100]
yarray=[1.2 1 0.8 0.6]';
DSSText.Command='new XYcurve.MyEff npts=4 xarray=[0.1 0.2 0.4 1]
yarray=[0.86 0.9 0.93 0.97]';
csvwrite('circuit.csv',-P_circuit/4)%adimensionalizado con la potencia
máxima real en batería
csvwrite('Pfuelcar.csv',H)
csvwrite('irradi.csv',I)
csvwrite('temperatura.csv',T)
csvwrite('hourwind.csv',W)
DSSText.Command='new loadshape.MyIrrad 1 1 mult=(file=irradi.csv)
action=normalize';
DSSText.Command='new tshape.MyTemp 1 1 temp=(file=temperatura.csv)';
DSSText.Command='new loadshape.wind 1 1 mult=(file=hourwind.csv)';
if P_hydrogen>0
    P_fuelcell=P_hydrogen*0.8/1.2;%multiplicador adimensionalizado con
el nominal de la pila y su rendimiento
    csvwrite('fuelcell.csv',P_fuelcell);
    fprintf('\nPila de combustible ON\n');
    DSSText.Command='new loadshape.fcell 1 1 mult=(file=fuelcell.csv)';
    DSSText.Command='new loadshape.electroliz 1 1 mult=[0]';
else
    P_electroliz=-P_hydrogen*0.62/5;%multiplicador adimensionalizado con
el nominal de electrolizador y su rendimiento
    csvwrite('electroliz.csv',P_electroliz);
    fprintf('\nElectrolizador ON\n');
    DSSText.Command='new loadshape.electroliz 1 1
mult=(file=electroliz.csv)';
    DSSText.Command='new loadshape.fcell 1 1 mult=[0]';
end
DSSText.Command='new loadshape.fuelcar 1 1 mult=(file=Pfuelcar.csv)';
%aunque no se puede relacionar cuantitativamente, Sí se hace
%cuantitativamente, entre P_circuit y la acción de carga o descarga de
%batería
DSSText.Command='new loadshape.bateria 1 1 mult=(file=circuit.csv)';
if P_circuit>0
    fprintf('\nBatería cargando\n');
else
    fprintf('\nBatería descargando\n');
end
%definición de objetos y elementos del circuito
DSSText.Command='new transformer.sub phases=1 windings=2 xhl=2';
DSSText.Command='~ wdg=1 bus=sourcebus conn=LN kv=4.16 kva=500
%r=0.05';
DSSText.Command='~ wdg=2 bus=650 conn=LN kv=0.4 kva=500
%r=0.05';
DSSText.Command='new transformer.trafo1 phases=1 windings=2 xhl=2';
DSSText.Command='~ wdg=1 bus=632r conn=LN kv=0.4 kva=50 %r=0.05';
DSSText.Command='~ wdg=2 bus=632 conn=LN kv=0.68 kva=50 %r=0.05';
DSSText.Command='new transformer.trafo2 phases=1 windings=2 xhl=2';
DSSText.Command='~ wdg=1 bus=671r conn=LN kv=0.4 kva=50 %r=0.05';

```

```

DSSText.Command='~ wdg=2 bus=671 conn=LN kv=0.087 kva=50 %r=0.05';
DSSText.Command='new linecode.LCD1 nphases=1 basefreq=50
rmatrix=[1.07]';
DSSText.Command='new linecode.LCD2 nphases=1 basefreq=50
rmatrix=[0.24]';
DSSText.Command='new linecode.LCD3 nphases=1 basefreq=50
rmatrix=[6.84]';
DSSText.Command='new linecode.LCD4 nphases=1 basefreq=50
rmatrix=[1.71]';
DSSText.Command='new linecode.LCD5 nphases=1 basefreq=50
rmatrix=[2.85]';
DSSText.Command='new capacitor.cap1 bus1=634 phases=1 conn=delta
kvar=0.01 kv=0.087';
DSSText.Command='new line.L0 phases=1 bus1=650 bus2=632r linecode=LCD1
length=50e-3';
DSSText.Command='new line.L1 phases=1 bus1=632 bus2=633 linecode=LCD1
length=1e-3';
DSSText.Command='new line.L2 phases=1 bus1=633 bus2=685 linecode=LCD1
length=5e-3';
DSSText.Command='new line.L3 phases=1 bus1=632r bus2=671r linecode=LCD1
length=1e-3';
DSSText.Command='new line.L4 phases=1 bus1=671 bus2=634 linecode=LCD2
length=20e-3';
DSSText.Command='new line.L5 phases=1 bus1=634 bus2=675 linecode=LCD2
length=1e-3';
DSSText.Command='new line.L6 phases=1 bus1=634 bus2=635 linecode=LCD3
length=1e-3';
DSSText.Command='new line.L7 phases=1 bus1=632 bus2=644 linecode=LCD4
length=1e-3';
DSSText.Command='new line.L8 phases=1 bus1=644 bus2=655 linecode=LCD4
length=50e-3';
DSSText.Command='new line.L9 phases=1 bus1=644 bus2=645 linecode=LCD5
length=25e-3';
DSSText.Command='new load.SL1 bus1=675 phases=1 conn=LN model=1 kv=0.087
kw=15 pf=1 duty=electroliz basefreq=50';
DSSText.Command='new load.SL2 bus1=635 phases=1 conn=LN model=1 kv=0.087
kw=2.4 pf=1 duty=fuelcar basefreq=50';
DSSText.Command='new storage.battery';
DSSText.Command='storage.battery.bus1=685 phases=1 conn=LN model=1
kv=0.68 kw=10.5';
%actualización del nivel inicial SOC batería
if nivelSOC<30
    DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=9 pf=1
duty=bateria basefreq=50 dispmode=follow';
end
if nivelSOC>29 & nivelSOC<40
    DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=13 pf=1
duty=bateria basefreq=50 dispmode=follow';
end
if nivelSOC>39 & nivelSOC<50
    DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=17 pf=1
duty=bateria basefreq=50 dispmode=follow';
end
if nivelSOC>49 & nivelSOC<60
    DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=20 pf=1
duty=bateria basefreq=50 dispmode=follow';
end
if nivelSOC>59 & nivelSOC<70
    DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=23 pf=1
duty=bateria basefreq=50 dispmode=follow';
end
if nivelSOC>69 & nivelSOC<80
    DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=27 pf=1

```

```

duty=bateria basefreq=50 dispmode=follow';
end
if nivelSOC>79
    DSSText.Command='~ kWrated=10 kWhrated=36 kWhstore=32 pf=1
duty=bateria basefreq=50 dispmode=follow';
end
DSSText.Command='new PVsystem.PV phases=1 bus1=645 kv=0.68 kva=16.5
Pmpp=16.5 irrads=0.8 temperature=25';
DSSText.Command='~ pf=1 basefreq=50 effcurve=Myeff P-TCurve=MyPVst
conn=LN duty=MyIrrad tduty=MyTemp';
DSSText.Command='new generator.genwind bus1=655 phases=1 conn=LN model=1
kv=0.68 kw=12 pf=1 duty=wind basefreq=50';
DSSText.Command='new generator.fuelcell bus1=634 phases=1 conn=LN
model=1 kv=0.087 kw=3.6 pf=1 duty=fcell basefreq=50';
DSSText.Command='new monitor.nivel_bateria storage.battery 1 mode=3';
DSSText.Command='new monitor.Pbateria storage.battery 1 mode=1';
DSSText.Command='new monitor.Ielectro load.SL1 1 mode=0';
DSSText.Command='new monitor.Pelectro load.SL1 1 mode=1';
DSSText.Command='new monitor.Pfuelcar load.SL2 1 mode=1';
DSSText.Command='new monitor.Ifuelcell generator.fuelcell 1 mode=0';
DSSText.Command='new monitor.Pfuelcell generator.fuelcell 1 mode=1';
DSSText.Command='new monitor.Pnet line.L0 1 mode=1';
DSSText.Command='new monitor.Psun PVsystem.PV 1 mode=1';
DSSText.Command='new monitor.Pwind generator.genwind 1 mode=1';
DSSText.Command='set voltagebases=(4.16 0.68 0.4 0.087)';
DSSText.Command='calc voltagebases';
DSSText.Command='solve';
DSSText.Command='solve mode=duty stepsize=60 number=60 hour=0 sec=0';
DSSText.Command='export monitor object=nivel_bateria';
DSSText.Command='export monitor object=Pbateria';
DSSText.Command='export monitor object=Ielectro';
DSSText.Command='export monitor object=Ifuelcell';
DSSText.Command='export monitor object=Psun';
DSSText.Command='export monitor object=Pwind';
DSSText.Command='export monitor object=Pfuelcar';
DSSText.Command='export monitor object=Pfuelcell';
DSSText.Command='export monitor object=Pelectro';
DSSText.Command='export monitor object=Pnet';
%obtención de datos de monitores
RESULTADO1=csvread('intall_mon_Ielectro.csv',1,1);
RESULTADO2=csvread('intall_mon>Ifuelcell.csv',1,1);
RESULTADO3=csvread('intall_mon_nivel_bateria.csv',1,1);
RESULTADO4=csvread('intall_mon_Pbateria.csv',1,1);
CargaDatos1 = csvread('intall_Mon_psun.csv',1,1);
CargaDatos2 = csvread('intall_Mon_pwind.csv',1,1);
CargaDatos8 = csvread('intall_Mon_pnet.csv',1,1);
%preparación de datos
[L,N]=size(RESULTADO1);
cap=1; %capacidad depósito hidruros metálicos Nm3
nivel_0=nivelMHL*cap/100; %nivel inicial del depósito Nm3
I_electroliz=RESULTADO1(:,6);
I_fuelcell=RESULTADO2(:,6);
F=9.6496e4; %constante de Faraday
Nez=30; %número de celdas del electrolizador
Afc=1.48; %área de intercambio en la pila de combustible
%flujos (conversión de unidades litros/s a unidades Nm3/min)
m_ez=I_electroliz*Nez*60/(1000*F);
m_fc=I_fuelcell*60/(2000*Afc*F);
%consumo de hidrógeno (en Nm3/min)
%equivalente a 0.8 kW (incluyendo pérdidas eléctricas)
if H==1
    m_car=ones(L,1)*(0.8*1000/48)*Nez*60/(1000*F);
else

```

```

        m_car=zeros(L,1);
    end
    %actualización de la variable de estado MHL
    nivel=zeros(L,1);
    nivel(1)=nivel_0;
    for i=2:L
        nivel(i)=nivel(i-1)+m_ez(i)-m_fc(i)-m_car(i);
    end
    SOC1=RESULTADO3(1,2)/0.36;
    SOC2=RESULTADO3(L,2)/0.36;
    MHL1=nivel(1)*100;
    MHL2=nivel(L)*100;
    Pcircuit=RESULTADO4(L,2);
    if P_circuit<0
        Pcircuit=-Pcircuit;
    end
    I_hydrogen=RESULTADO2(L,6)-RESULTADO1(L,6);
    %contadores de energía en el periodo (1 hora)
    Gensun=(1/60)*sum(CargaDatos1(:,2));
    Genwind=(1/60)*sum(CargaDatos2(:,2));
    Gen=Gensun+Genwind;%energía generada por renovables
    % sentido del intercambio de energía con la red exterior
    sentido=CargaDatos8(:,3);
    for i=1:L
        if sentido(i)>90 | sentido(i)<-90
            intercambio(i)=-CargaDatos8(i,2);
        else
            intercambio(i)=CargaDatos8(i,2);
        end
    end
    Export=(1/60)*sum(-intercambio);%energía exportada a la red
    Hibrid=H*0.8;%energía cedida al coche híbrido
    %fin de función
    RESULTADO=[Pcircuit,I_hydrogen,SOC1,SOC2,MHL1,MHL2,Gen,Export,Hibrid];

```

REFERENCIAS

- [1] Eduardo Fernández Camacho & Carlos Bordons, «*Model Predictive Control*», chapters 1-7, Editorial Springer, Second Edition, 2004.
- [2] Carlos Bordons, Félix García-Torres y Luis Valverde, «*Gestión Óptima de la Energía en Microrredes con Generación Renovable*», Revista Iberoamericana de Automática e Informática Industrial, publicación 12 (2015) p. 117-132
- [3] Inmaculada Fdez. de la Calle, Alfonso Camacho López, Carlos Javier Gasco Lallave, Ana M^a Macías Juárez, M^a Ángeles Martín Hernández, Gabriela Reyes Delgado, Julio Rivas Escudero, «*Seguridad funcional en instalaciones de proceso: sistemas instrumentados de seguridad y análisis SIL*», capítulo 6, Díaz de Santos, 2012.
- [4] Salvador Cucó Pardillos, «*Manual de Energía Eólica. Desarrollo de Proyectos e Instalaciones*», cap. 3, Editorial Universitat Politècnica de València, 2017.
- [5] Antonio Creus Solé, «*Aerogeneradores*», capítulo 4, Editorial Técnica Cano Pina, 2008.
- [6] Pablo González Martínez, «*Evaluación de Pérdidas de Energía en la Operación de Plantas Fotovoltaicas*», Trabajo Fin de Grado, Grado en Ingeniería de Tecnologías Industriales, Tutor Dr. D. Isidoro Lillo Bravo, Escuela Técnica Superior de Ingeniería de Sevilla, 2016.
- [7] José M^a Fernández Salgado, «*Guía Completa de la Energía Eólica*», capítulo 4, Ediciones Madrid, 2010.
- [8] Roberto Carapellucci, Lorena Giordano, «*A methodology for the synthetic generation of hourly wind speed time series based on some known aggregate input data*», Applied Energy, 101 (2013) p. 541-550
- [9] L. Valverde, F. Rosa, A. J. Del Real, A. Arce & C. Bordons, «*Modeling, simulation and experimental set-up of a renewable hydrogen-based domestic*», International Journal of Hydrogen Energy, publication 38 (2013) p. 11672 - 11684
- [10] Antonio Creus Solé, «*Energías Renovables*», capítulos 3 y 7, Editorial Técnica Cano Pina, 2^a edic, 2009.
- [11] Antonio José Besa González, Javier Carballeira Morado, «*Diagnóstico y corrección de fallos de componentes mecánicos*», Editorial Universitat Politècnica de València, 2015.
- [12] Juan Pablo Claro Báez, «*Modelado y Simulación de Smart Grid con OpenDSS y Matlab*», Trabajo Fin de Grado, Grado en Ingeniería Aeroespacial, Tutora Ascensión Zafra Cabeza, Escuela Técnica Superior de Ingeniería de Sevilla, 2016.
- [13] «*Reglamento Electrotécnico de Baja Tensión*», Real Decreto 842/2002, de 2 de Agosto, publicado en B.O.E número 224 de 18 de Septiembre de 2002.
- [14] Matthew Reno and Kyle Coogan, «*Grid Integrated Distributed PV (GridPV)*», Tech. Report. Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), aug 2013.
- [15] Rafael Espinosa Velazco, «*Estudio de Técnicas de Toma de Decisiones con Incertidumbre. Aplicación a Sistemas Reales*», Trabajo Fin de Grado, Grado en Ingeniería de Tecnologías Industriales, Tutora Ascensión Zafra Cabeza, Escuela Técnica Superior de Ingeniería de Sevilla, 2017.

