



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Modelo uninodal para el análisis de la respuesta de la frecuencia en sistemas eléctricos de potencia

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Juan Lima Rodríguez e Ignacio Nin Iewdiukow

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRICISTA.

TUTORES

Álvaro Giusto Universidad de la República
Nicolás Yedrzejewski Universidad de la República

TRIBUNAL

Juan Bazerque Universidad de la República
Ricardo Franco Universidad de la República
Nicolás Yedrzejewski Universidad de la República

Montevideo
viernes 20 de diciembre de 2019

Modelo uninodal para el análisis de la respuesta de la frecuencia en sistemas eléctricos de potencia, Juan Lima Rodríguez e Ignacio Nin Iewdiukow.

Resumen

La penetración de las energías renovables no convencionales (ERNc) en los sistemas eléctricos de potencia (SEP) trae aparejada una disminución de la inercia en la red, con consecuencias para la regulación de frecuencia.

A fin de evaluar distintas estrategias de respuesta ante este fenómeno, es de interés contar con un modelo simplificado del SEP, que refleje la respuesta de la frecuencia ante desequilibrios de generación y demanda con foco en las variables que inciden en el problema —inercia, generación y demanda totales— a la vez que soslaye aquellas con menor incidencia —distribución geográfica de generación y demanda, transmisión y distribución, modelos específicos de máquinas—.

En el presente trabajo se desarrolla el comportamiento de la frecuencia en un SEP de forma teórica y se postula la existencia de un modelo de un único nodo con generación y demanda agrupadas en un número mínimo de elementos, cuya respuesta de la frecuencia ante desequilibrios entre generación y demanda es equivalente a la del SEP que modela. Su transferencia (lineal) es una función de cuatro parámetros: H y D representan la inercia y amortiguamiento totales en el sistema, y R y T_g el estatismo y la constante de tiempo de regulación primaria equivalentes.

Mediante el análisis de esta transferencia, se desarrolla la teoría necesaria para identificar H , D , T_g y R equivalentes del SEP a partir de la respuesta al escalón —obtenida en forma experimental— de la frecuencia.

Se construye una herramienta que, a partir de la respuesta de la frecuencia de un SEP que se desee modelar, identifica estos parámetros equivalentes y construye un modelo uninodal en formato PSS[®] e cuya respuesta es aproximada a la del SEP. Luego utiliza algoritmos evolutivos para mejorar esta aproximación, hasta obtener un modelo cuya respuesta es equivalente a la del sistema original en las etapas inercial y de regulación primaria de la respuesta de la frecuencia.

El trabajo se concluye con pruebas sobre el modelo del sistema interconectado uruguayo-argentino que validan el modelo producido por la herramienta como equivalente dinámico durante los primeros instantes de la respuesta, y otras que validan la teoría desarrollada para la identificación de parámetros de inercia y regulación primaria equivalentes del SEP.

Se concluye también que el algoritmo desarrollado brinda un método para identificar la inercia presente en el SEP a partir de su respuesta de la frecuencia ante un desequilibrio entre generación y demanda.

Glosario

Amortiguamiento Parámetro de la red relacionado con la variación de la potencia demandada al variar la frecuencia.

Aptitud Sinónimo de *fitness*.

Banda muerta Rango de frecuencia alrededor de la nominal en la que la regulación primaria no actúa.

Caso de uso Selección del usuario del programa de uno de tres tipos de modelo uninodal.

Caso PSS[®]e Datos de una red eléctrica en formato PSS[®]e.

Centro de inercia Análogo al centro de masas para máquinas síncronas, su velocidad representa la velocidad media, pesada por las inercias, de las máquinas síncronas de la red.

Ceteris paribus «Permaneciendo constantes las otras condiciones».

Curva de referencia Para una ejecución de la herramienta, es la curva de salida de la falta original en la red original, que define el *fitness*.

D Constante de amortiguamiento.

Distancia relativa Medida del *fitness* usada por el algoritmo.

Elitismo Parámetro del algoritmo evolutivo que indica si los mejores individuos de cada generación se conservan, y cuántos.

Estatismo Constante que determina la relación entre potencia inyectada y variación de frecuencia en un sistema con regulación primaria.

Exploración Modelo de ejecución del algoritmo evolutivo donde se buscan zonas del espacio de soluciones con buen *fitness*.

Explotación Modelo de ejecución del algoritmo evolutivo donde se buscan intensivamente soluciones en un área acotada.

Fitness Calidad del individuo como solución al problema. Es la cantidad a optimizar en el algoritmo evolutivo.

Función objetivo Función que el algoritmo evolutivo optimiza.

Generación Iteración del algoritmo evolutivo, representando el conjunto de individuos que están activos en un momento dado de la evolución. No confundir con *generación de energía*.

Generación convencional Generación de energía por máquinas síncronas.

Genoma Conjunto de valores que representan la identidad del individuo, parámetros de la función objetivo.

H Constante de inercia.

Individuo Solución candidata en el algoritmo evolutivo, representa un modelo uninodal posible.

Inercia Constante de inercia de una máquina síncrona.

Inercia de la red, inercia equivalente, inercia total Suma de las inercias de las máquinas rotativas conectadas a la red.

Inercia sintética Inercia que se incorpora a la red mediante electrónica de potencia.

Modelo base Modelo que contiene la estructura del modelo uninodal: línea de impedancia nula, barra principal y barra flotante.

Mutación Operador genético que altera aleatoriamente un elemento del genoma del individuo.

Nadir Punto extremo de la desviación de la frecuencia del valor nominal. Se denomina *tiempo de nadir* al tiempo en que esto ocurre.

p Parámetro intermedio de curva para el ajuste no oscilatorio.

Parámetros de corrida Parámetros del modelo uninodal que no cambian a lo largo de una corrida de algoritmo evolutivo. cf. *Parámetros de individuo*.

Parámetros de individuo Genoma del individuo que el modelo uninodal recibe para su ejecución.

Población Conjunto de individuos mantenido por el algoritmo evolutivo. Se renueva en cada generación.

q Parámetro intermedio de curva para ajustes no oscilatorio y oscilatorio.

r Parámetro intermedio de curva para ajustes no oscilatorio.

R Parámetro de *estatismo* de la regulación primaria.

Regulación primaria Regulación autónoma que hacen las máquinas síncronas en los primeros instantes posteriores a un desvío de la frecuencia. Coloquialmente *estatismo*.

Red de referencia u original Para una ejecución de la herramienta, es la red para la cual se obtuvo la falta original (la que se desea modelar).

Resincronización Detección de tiempo de falta en series temporales para su comparación.

ROCOF *Rate of Change of Frequency*, derivada temporal de la frecuencia.

Serie temporal Pareja de sucesiones que representan un muestreo temporal de una variable real.

Solución Sinónimo de *individuo*.

T_g Tiempo de retardo de la regulación primaria.

Tiempo de falta Tiempo en el cual comienza la *respuesta inercial*.

ϕ Parámetro intermedio de curva para ajuste oscilatorio, representa el desfase del nadir.

ω Dependiendo del contexto, indica:

1. Parámetro intermedio de curva para ajuste oscilatorio, representa la frecuencia de oscilación de la frecuencia de la red.
2. Frecuencia o velocidad de rotación de una máquina.

Siglas

AA.EE. Algoritmos Evolutivos

CLI *Command Line Interface*, interfaz de línea de comandos

GUI *Guided User Interface*, interfaz gráfica

ERNC Energías Renovables No Convencionales

PSS[®]e *Power System Simulator for Engineering*

SEP Sistema Eléctrico de Potencia

Tabla de contenido

Resumen	I
Glosario	III
Tabla de contenido	VII
I Preliminares	1
1. Introducción	3
1.1. Contexto: influencia de las ERNC	3
1.1.1. Simulaciones	4
1.2. SEP y frecuencia	4
1.2.1. Rol de la inercia	5
1.3. Objetivos	6
1.4. Metodología	6
1.5. Contribución	7
1.6. Marco y alcance	7
1.7. Organización del documento	8
2. Planteo del problema	9
2.1. El problema y la solución propuesta	9
2.1.1. Requerimientos de una posible solución	9
2.1.2. Solución propuesta	10
2.2. Implementación de la solución y trabajo propuesto	10
2.3. Antecedentes	11
2.4. Trabajos relacionados	11
2.4.1. Impacto de la generación ERNC	11
2.4.2. Modelado simplificado del sistema	12
2.4.3. Estimación de la inercia	12
2.4.4. Equivalentes dinámicos y algoritmos evolutivos	13

II	Marco teórico	15
3.	Respuesta de la frecuencia en un SEP	17
3.1.	Generador síncrono	18
3.1.1.	Constante de inercia	18
3.1.2.	La ecuación de <i>swing</i>	19
3.2.	Modelado de cargas	20
3.2.1.	Modelos de carga estáticos	20
3.2.2.	Cargas con inercia: motor de inducción	23
3.3.	Velocidad del centro de inercia	23
3.3.1.	Centro de inercia	25
3.3.2.	Sistema sin regulación de frecuencia	27
3.4.	Control de frecuencia	28
3.4.1.	Regulación primaria	29
3.5.	Regulación primaria en un SEP	32
3.5.1.	Participación de múltiples generadores en la regulación primaria	33
3.5.2.	Función de transferencia	35
3.5.3.	Características de la respuesta	37
4.	Algoritmos evolutivos	39
4.1.	Introducción	39
4.2.	Influencia de la generación inicial	40
4.3.	Elitismo y diversidad	41
4.4.	Formalización	41
III	Desarrollo de la solución	43
5.	Identificación de parámetros de curva	45
5.1.	Método	45
5.2.	Puntos clave de la curva de entrada	46
5.2.1.	Caso no oscilatorio	46
5.2.2.	Caso oscilatorio	48
5.3.	Transferencia	49
5.4.	Antitransformada (dominio del tiempo)	49
5.4.1.	Polos reales	50
5.4.2.	Polos complejos	50
5.5.	Caracterización de los parámetros	51
5.5.1.	Obtención directa de parámetros	51
5.5.2.	Obtención de D , T_g , y R a través de parámetros intermedios: caso no oscilatorio	52

5.5.3.	Obtención de D , T_g , y R a través de parámetros intermedios: caso oscilatorio	56
5.6.	Generalización a un escalón ΔP	58
5.7.	Consideraciones prácticas	60
5.7.1.	Identificación de t_0 y H en redes con perturbaciones iniciales	60
5.7.2.	Intervalo en la detección de $q - r$ para el caso no oscilatorio	61
6.	Cálculo de <i>fitness</i>	63
6.1.	Evaluación de un individuo	63
6.2.	Curva de salida	64
6.2.1.	Serie temporal	64
6.2.2.	Respuesta de la frecuencia como serie temporal	64
6.3.	<i>Fitness</i> de una curva de salida	65
6.3.1.	Sincronización temporal	65
6.3.2.	Distancia entre curvas como <i>fitness</i> del individuo	67
6.4.	Conclusión	70
7.	Arquitectura de la solución	71
7.1.	Flujo de datos	71
7.2.	Control de PSS [®] e	73
7.3.	Programa controlador del algoritmo evolutivo	74
7.4.	Arquitectura cliente-servidor	75
7.5.	Conceptos de AA. EE. en la solución	76
7.5.1.	Generación inicial	77
8.	Programa controlador de PSS[®]e	81
8.1.	Implementación	81
8.2.	Modelo uninodal	82
8.2.1.	Parámetros de elementos	82
8.2.2.	Parámetros de entrada	84
8.2.3.	Estructura de doble barra	86
8.2.4.	Modelo base	87
8.2.5.	Descripción de los casos de uso	88
8.3.	Ejecución del programa	91
8.3.1.	Construcción del modelo uninodal	92
8.3.2.	Simulación	93
8.3.3.	Respuesta	93
8.4.	Otras utilidades	93
8.4.1.	Obtención de curvas de red de referencia	93
8.4.2.	Obtención de parámetros de la red original	94
8.5.	<i>Docker</i> como motor múltiple para paralelismo	95

Tabla de contenido

9. Programa controlador del motor de AA.EE.	97
9.1. Identificador	97
9.1.1. Parámetros iniciales	98
9.2. Controlador de ECJ	100
9.2.1. Individuos y especie	101
9.2.2. Evaluación	102
9.2.3. Cruzamiento y mutación	102
9.2.4. Ciclo de ejecución	102
9.3. Comunicación con el servidor	102
9.4. Bibliotecas auxiliares	103
9.4.1. Series temporales	103
9.4.2. Cálculo simbólico	105
9.5. Interfaz	106
9.5.1. Parámetros del algoritmo	107
9.5.2. CLI	108
9.5.3. GUI	108
9.6. Reporte	110
IV Resultados y conclusiones	113
10. Evaluaciones y resultados	115
10.1. Parámetros de exploración aleatoria	115
10.1.1. Parámetros a ajustar	115
10.1.2. Condiciones de ejecución	116
10.1.3. Resultados de las corridas	117
10.1.4. Conclusión	118
10.2. Evaluación del modelo uninodal como equivalente dinámico de una red	119
10.2.1. Motivación y objetivo	119
10.2.2. Metodología	119
10.2.3. Evaluación de red de prueba "A"	120
10.2.4. Prueba en red interconectada uruguayo-argentina	124
10.2.5. Discusión	127
10.2.6. Conclusiones	130
10.3. Evaluación del ajuste como semilla para la generación inicial	131
10.3.1. Motivación	131
10.3.2. Método	131
10.3.3. Curvas de entrada y parámetros iniciales	132
10.3.4. Parámetros que varían	133
10.3.5. Resultados: caso no oscilatorio	135
10.3.6. Resultados: caso oscilatorio	138
10.3.7. Discusión	140

10.4. Distribución de potencia en generadores y parámetros del individuo	143
10.4.1. Prueba	143
10.4.2. Resultados y discusión	144
10.5. Dimensión del genoma y convergencia	144
10.5.1. Metodología	145
10.5.2. Resultados y discusión	145
11. Conclusiones	151
11.1. Solución presentada	151
11.2. Modelo de regulación primaria para el SEP	152
11.2.1. Influencia de la generación inicial	153
11.3. Identificación de parámetros de inercia y estatismo	153
11.3.1. Algoritmo de ajuste por mínimos cuadrados	154
11.3.2. Detección de la inercia en base a la respuesta inercial	154
12. Trabajos futuros	157
12.1. Ajuste	157
12.1.1. Detección oscilatoria/no oscilatoria	157
12.1.2. Granularidad	158
12.2. Algoritmo evolutivo	158
12.2.1. Elitismo	158
12.2.2. Criterio de corte	158
12.3. GUI	158
12.4. Reporte	159
12.5. Modelo uninodal	159
Anexos	163
A. Software PSS[®]e	163
A.1. Simulaciones dinámicas en PSS [®] e	163
A.2. Modelos	166
A.2.1. Generadores convencionales	167
A.2.2. Reguladores mediante estatismo	169
A.2.3. Generadores no convencionales	171
B. Guía del usuario	173
B.1. Arquitectura general	173
B.2. Programa servidor	173
B.2.1. Instalación	173
B.2.2. Ejecución	173
B.2.3. Versión <i>Docker</i>	174

Tabla de contenido

B.3. Programa cliente	174
B.3.1. Versión CLI	174
B.3.2. Versión GUI	174
C. Esquema red de pruebas "A"	181
Referencias	183

Parte I
Preliminares

Capítulo 1

Introducción

1.1. Contexto: influencia de las ERNC

Actualmente existe un importante desarrollo en lo denominado *energías renovables no convencionales* (ERNC); en particular la explotación de energías eólica y solar fotovoltaica. Lejos de ser ajeno a este fenómeno, Uruguay ha sido un protagonista de primera línea: con la instalación de 43 parques eólicos entre 2008 y 2018, el país tiene una capacidad instalada de generación eólica de aproximadamente 1500 MW —llegando a superar por sí sola, en ocasiones, la demanda media diaria—¹, a los que se suman aproximadamente 250 MW instalados de generación solar fotovoltaica [1].

Si bien gran parte de la generación en el SEP interconectado uruguayo-argentino continúa siendo de fuentes hidroeléctricas o térmicas («convencionales»), la república hermana tiene como objeto aumentar la capacidad de generación renovable en general, y eólica en particular, en el futuro cercano [2] [3].

Una característica de este tipo de generación es que no corresponde a máquinas síncronas conectadas directamente a la red. En una planta fotovoltaica, los módulos fotovoltaicos generan energía eléctrica en corriente continua y se conectan a la red mediante inversores, mientras que en un parque eólico —en la mayoría de los casos— las máquinas rotativas de los aerogeneradores generan en corriente alterna a una frecuencia distinta de la red, por lo que su conexión a la red está dada también por una etapa de conversión de alterna–continua–alterna. Por tanto, no aportan inercia a la red, como sí lo hace la generación convencional.

La introducción de este tipo de generación, junto a la sustitución de centrales de generación convencionales por parques eólicos o fotovoltaicos, trae aparejada una pérdida de inercia en la red que tiene consecuencias directas sobre la regulación de frecuencia. Algunos ejemplos de for-

¹Por ejemplo, 1200 MW fue la demanda media del 28/2/19 [4].

mas de paliar este efecto son:

- la manutención en servicio de máquinas síncronas que no inyectan potencia activa, sin embargo aportan inercia (y quizá regulación) a la red;
- la incorporación de *inercia sintética* en los generadores basados en convertidores de electrónica de potencia;
- regulación primaria de potencia más rápida que la convencional.

1.1.1. Simulaciones

Interesa predecir el comportamiento del sistema —en particular, la respuesta de la frecuencia ante desbalances entre generación y demanda— para distintos escenarios y combinaciones de tipo de generación (convencional, ERNC, etc.), que tanto reflejen la realidad actual como posibles escenarios futuros, a fin de preparar respuestas (por ejemplo, de uno de los tipos mencionados) al fenómeno de la disminución de inercia en la red.

Tanto UTE como su(s) contraparte(s) argentina(s) han desarrollado un modelo detallado de la red para el sistema interconectado en la plataforma PSS[®]e que incluye la capacidad de modelar el comportamiento dinámico de la frecuencia ante situaciones de desbalance de potencia. Sin embargo, el modelo incluye muchas variables y detalles que, a efectos de estudiar la respuesta de la frecuencia como fenómeno aislado, conviene soslayar.

En el presente trabajo se desarrollará la tesis de que es posible estudiar el comportamiento de la frecuencia, de forma aproximada, en función de un conjunto reducido de parámetros de la red y se construirá una herramienta para este análisis.

1.2. SEP y frecuencia

Se denomina sistema eléctrico de potencia (SEP) al conjunto de componentes eléctricos cuyo objeto es abastecer las demandas energéticas de una región.

Un SEP tiene una *frecuencia nominal* asociada para la cual se diseñan gran parte de sus componentes: máquinas generadoras, hierro de los materiales magnéticos de los convertidores, etc.. Un apartamiento de este valor nominal provoca efectos indeseables en todos los niveles, desde pérdida de eficiencia en la transmisión y el consumo para desviaciones

pequeñas, hasta daño permanente en equipos rotatorios de generación o consumo para desviaciones grandes.

Típicamente un SEP tiene parte de su generación en forma de máquinas síncronas, siendo su velocidad de giro en régimen igual a la frecuencia de la red (a menos de una constante de proporcionalidad); ambas cantidades se alteran mutuamente.

La variación de la frecuencia instantánea (y por tanto, de la velocidad de giro en las máquinas síncronas) responde a cambios en el equilibrio entre generación y demanda. A generación constante, un aumento instantáneo en la demanda crea una deficiencia en la potencia en la red, que ralentizará las máquinas síncronas conectadas provocando una disminución de la frecuencia; mientras que una disminución tendrá el efecto contrario, acelerando las máquinas debido al exceso de potencia y aumentando la frecuencia en el SEP.

La *regulación de frecuencia* consiste en la supervisión constante de la velocidad de giro en las máquinas síncronas (o, indistintamente, de la frecuencia en la red) y el aumento o disminución de la potencia generada con el fin de mantener el equilibrio entre generación y demanda.

1.2.1. Rol de la inercia

El rotor² de una máquina síncrona, como objeto masivo que gira, posee un *momento de inercia* que determina la aceleración angular a la que es sujeto ante un desbalance de carga.

La caída o aumento de la demanda energética es percibida en el eje como un par (o *torque*) que tiende a acelerarlo o ralentizarlo. A mayor momento de inercia, igual par produce una aceleración angular menor, y por tanto un ROCOF³ menor. En otras palabras, el momento de inercia de la máquina está directamente ligado al ROCOF, siendo inversamente proporcionales.

Del hecho de que la inercia tenga este efecto atenuador sobre la rapidez de cambio en la frecuencia se desprende que una mayor inercia otorga un mayor margen al operador del SEP para actuar sobre la potencia inyectada y mantenerla dentro de los parámetros aceptables.

Como será desarrollado en el capítulo 3, en un SEP con múltiples máquinas síncronas conectadas, la rapidez de cambio de la frecuencia está relacionada con la suma aritmética de los momentos de inercia de todas ellas. La incidencia de esta cantidad es uno de los principales objetos de estudio de este trabajo, y nos referiremos en lo sucesivo al momento de inercia del eje de una máquina síncrona como la *inercia* de la máquina y

²De hecho el conjunto rotor–eje–turbina.

³«Rate of Change of Frequency», tasa o rapidez de cambio de la frecuencia.

a la suma de los momentos de inercia presentes en un sistema como la *inercia de ese sistema*.

1.3. Objetivos

El **objetivo general** es contribuir a mejorar las capacidades locales para asegurar un suministro eléctrico seguro y confiable en lo relativo a la respuesta de la frecuencia en un SEP, en un entorno de alta penetración de generación renovable no convencional.

El **objetivo específico** es construir una herramienta que permita simular el despacho de recursos de regulación de frecuencia en el sistema interconectado uruguayo-argentino.

1.4. Metodología

Se plantea modelar el sistema eléctrico interconectado uruguayo-argentino con un modelo de un único nodo, con un número reducido de máquinas conectadas representando cada una el agrupamiento de los diferentes tipos de generación presentes en la red (convencional, no convencional, etc.), y una carga simple que represente la demanda total del sistema. Es decir, un modelo que desprece los efectos tanto de la transmisión y distribución como de la distribución geográfica de la generación y la demanda. A estos modelos reducidos de un SEP se les denomina *equivalente dinámico* del SEP.

Un modelo con tales características sería capaz de representar razonablemente los fenómenos centrales de la respuesta de la frecuencia, y será útil para extraer conclusiones valiosas sobre el dimensionado de recursos, tiempos de actuación, y naturaleza de acciones de control y protección.

El modelo se construirá en PSS[®]e a partir de los usados por UTE para sus estudios.

La construcción de tal modelo es un problema de optimización, donde es necesario buscar una solución dentro de un espacio de múltiples variables (tantas como el modelo uninodal dicte). Para resolverlo se utilizarán *algoritmos evolutivos* (AA.EE.), un modelo de metaheurística de enfoque darwinista que permite, a partir del conocimiento parcial del espacio de soluciones, optimizar una función objetivo ofreciendo buenos resultados en espacios de búsqueda relativamente amplios.

1.5. Contribución

El fruto del proyecto pretende ser un primer paso en el modelado uninodal de la red interconectada. Se extraen conclusiones sobre la viabilidad del uso de AA.EE. como herramienta para hallar el equivalente deseado y se propone un modelo uninodal con una cantidad reducida de variables, mostrando que es posible obtener conclusiones sobre la respuesta de la frecuencia a partir de un modelo uninodal simple.

Como parte del proyecto, se hace un estudio teórico matemático sobre la respuesta de la frecuencia en un SEP con regulación primaria que permite, a partir de la respuesta de un sistema de esas características a un desequilibrio de potencia, estimar los valores de inercia total del sistema y su amortiguamiento, así como las constantes relacionadas a la regulación primaria. Las conclusiones de este estudio son parte central de la ejecución del algoritmo evolutivo para la obtención de resultados, y puede tener aplicación en áreas relacionadas.

Se demuestra también, como parte del proyecto, la posibilidad del uso paralelizado masivo de la versión estudiantil del programa PSS[®]e (allí donde su licencia lo permita) bajo la plataforma Linux mediante el uso de *Docker* y *Wine*, tecnologías de software libre. Éste fue el pilar del paralelismo para la ejecución del algoritmo evolutivo, y muestra una forma de acelerar las simulaciones para proyectos similares.

1.6. Marco y alcance

Siendo el alcance del proyecto acotado por los tiempos y personal máximos para un proyecto de grado de Ingeniero Electricista, éste fue pensado como el primer paso (o prueba de concepto) en la creación de una herramienta para el trabajo en el área de regulación de frecuencia.

Con este marco, se pensó desde el inicio en el uso de software probado, establecido y de uso corriente donde fuere posible, evitando lo que en la industria se conoce como *reinventar la rueda*; así como un énfasis en producir software de calidad, minimizando las implementaciones *ad hoc* u oscuras, pensando en la reutilización y extensión en proyectos futuros. También se ha hecho énfasis en la documentación, y por tanto se provee una guía del usuario y documentación para el desarrollador en el presente documento; así como guías y modelos de prueba para reproducir los resultados obtenidos.

1.7. Organización del documento

En la **Parte I – Preliminares** se encuentran la presente introducción y el capítulo 2 donde se delinearán en forma más clara los objetivos del proyecto, además de presentar formalmente algunas hipótesis de trabajo.

La **Parte II – Marco teórico** cuenta con dos capítulos. En el capítulo 3 se discuten los fundamentos de regulación de frecuencia y el concepto de regulación primaria, y se desarrolla la teoría necesaria para postular la existencia de un equivalente uninodal de las características que se buscan. En el capítulo 4 se expone en forma breve la teoría de algoritmos evolutivos.

En la **Parte III – Desarrollo** se exponen la teoría y productos del proyecto a lo largo de los capítulos 5 al 9. En el capítulo 5 se desarrolla un método para ajustar una curva de respuesta de frecuencia real a una del tipo desarrollado en el capítulo 3, que se usará tanto para proveer al algoritmo evolutivo de soluciones iniciales como para auxiliar en el cálculo del *fitness* de una solución.

En el capítulo 6 se postula una función de *fitness* para su uso en el algoritmo; además se presentan algunos problemas teóricos que aparecen durante su cálculo. En los capítulos 7, 8 y 9 se discute la solución implementada: el primero expone su arquitectura general, mientras que en los siguientes se exponen detalles de los dos programas que componen esta solución.

La **Parte IV – Resultados y conclusiones** contiene los capítulos 10, dedicado a detallar pruebas hechas con el programa para corroborar su validez; 11, donde se exponen las conclusiones del proyecto y 12, donde se proponen ideas de posibles incorporaciones futuras.

En los **Anexos** se ofrecen la guía del usuario, que permite el uso del programa sin necesidad de leer el presente documento en su totalidad, y un capítulo que describe la ejecución de *simulaciones dinámicas* y la biblioteca de *modelos dinámicos* de *rss*[®]e.

Capítulo 2

Planteo del problema

2.1. El problema y la solución propuesta

La penetración de las energías renovables no convencionales (ERNc) en el sistema interconectado uruguayo-argentino —transformándolo de un sistema de generación mayormente convencional y focalizada a uno de generación distribuida y de un mayor componente no convencional— tiene consecuencias para el control de frecuencia, debido a la disminución de la inercia de la red —dado el rol que esta última tiene en la estabilidad de la frecuencia ante desequilibrios de generación y demanda—. Esta disminución debe ser paliada con la introducción de inercia, que puede provenir de diferentes fuentes: sintética, convencional, etc..

A fin de tomar decisiones sobre la política de incorporación de inercia en el sistema, es de interés del planificador y operador del SEP simular la respuesta de la frecuencia ante distintos escenarios y configuraciones de inercia.

A efectos de estudiar y tomar decisiones macroscópicas sobre la inercia, los efectos de la distribución geográfica de la red pueden ignorarse, por lo que sería de interés un modelo simplificado de la red que no los incluya, y permita al operador aproximar la respuesta de la frecuencia en base a parámetros globales de la red (inercia total, generación y demanda totales, etc.).

2.1.1. Requerimientos de una posible solución

La solución deberá ser un modelo de la red que permita probar diferentes configuraciones de inercia

1. ignorando los efectos de la distribución geográfica de la generación y la demanda, y

Capítulo 2. Planteo del problema

2. que respete el comportamiento de la respuesta de la frecuencia para varios tipos de falta, produciendo una respuesta equivalente a la del modelo original o cuyo error sea aceptablemente pequeño durante el período de respuesta inercial.

2.1.2. Solución propuesta

Se propone la construcción de un *modelo uninodal* cuya respuesta de la frecuencia durante los primeros instantes —respuesta inercial— sea equivalente a la de la red, a cuyo nodo se conecta un número pequeño de generadores que representan la generación total y una carga que representa la demanda total, distribuidas en el SEP.

La hipótesis de trabajo es que en un SEP la respuesta de la frecuencia durante los primeros instantes depende —a efectos prácticos— exclusivamente del balance de potencia generada-demandada e inercia *totales* en la red, y es posible ignorar los efectos distribuidos de la generación y demanda sin perder mayor precisión.

Por consiguiente, la tesis de la solución es que la respuesta de la frecuencia a la misma falta en el modelo uninodal propuesto es equivalente a la respuesta en el modelo original.

2.2. Implementación de la solución y trabajo propuesto

Para implementar la solución, se desarrollará una herramienta que tendrá como

entrada el resultado de una simulación dinámica en el modelo completo de un SEP, que incluya la respuesta de la frecuencia ante un escalón en la potencia demandada o generada; y

salida un modelo uninodal que ante una falta de las mismas características, una simulación dinámica en el mismo producirá una respuesta de la frecuencia aproximadamente igual a la provista como entrada durante los primeros instantes de la respuesta inercial.

Hallar las características de las componentes de tal modelo es un problema de optimización no lineal, donde se debe buscar la solución óptima dentro de un espacio multidimensional.

Para resolverlo se proponen usar *algoritmos evolutivos* como motor de optimización, utilizando conocimientos de ingeniería eléctrica de potencia y desarrollos del modelo matemático de la respuesta de la frecuencia, para asegurar una convergencia rápida.

2.3. Antecedentes

[5] es un reporte técnico del *Frequency Response Technical Sub Group* dando cuenta de un análisis de diferentes soluciones para el fortalecimiento de redes que ven disminuida su capacidad inercial, entre ellas la inercia sintética. Para evaluar los futuros requisitos de respuesta de la frecuencia, se construyó un modelo de una sola barra haciendo uso de *Digsilent Power Factory*¹, una herramienta de software para análisis de sistemas. El modelo se muestra en la figura 2.1.

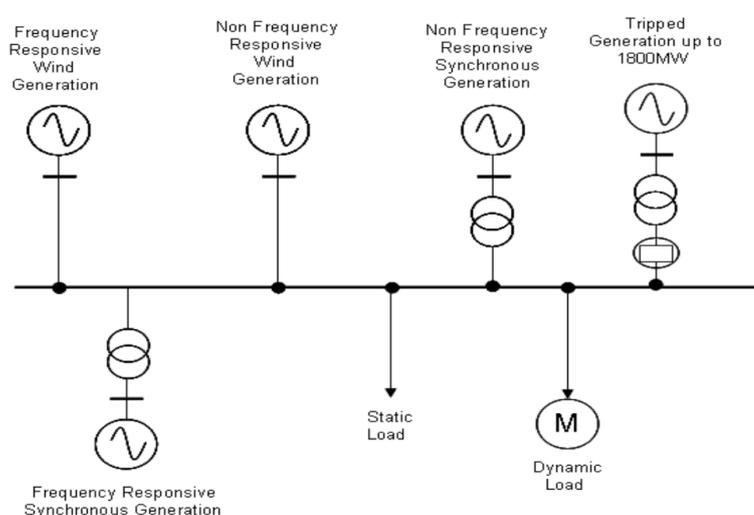


Figura 2.1: Modelo uninodal usado en [5]. Imagen tomada del reporte.

2.4. Trabajos relacionados

2.4.1. Impacto de la generación ERNC

En [6] se analiza el impacto de la generación ERNC en el control y estabilidad de frecuencia en sistemas eléctricos de potencia, en particular en el *Sistema Interconectado del Norte Grande (SING)*. En éste se proyectaba para el año 2017 una gran inserción de este tipo de tecnologías, principalmente de plantas de generación solar fotovoltaica. Se buscó caracterizar la respuesta de la frecuencia frente a desbalances de carga-generación realizando simulaciones dinámicas en un modelo del SING proyectado; concluyéndose que:

- el aumento de generación ERNC deteriora la capacidad de control de frecuencia del sistema,

¹<https://www.digsilent.de/en/powerfactory.html>

Capítulo 2. Planteo del problema

- existe una alta correlación entre la inercia del sistema y el ROCOF y
- existe una estrecha correlación entre el porcentaje de generación ERNC y la inercia del sistema.

2.4.2. Modelado simplificado del sistema

[7] es un estudio sobre la presencia de oscilaciones lentas en la frecuencia del sistema escandinavo, que provocan que la frecuencia de la red tome valores fuera del rango admisible para el sistema provocando un desgaste mecánico en turbinas y generadores. Mediante un estudio realizado por los operadores del sistema de transmisión, entre otras, se sugiere que una posible solución al problema es modificar el ajuste de los reguladores de frecuencia en los generadores hidráulicos.

Se utilizó un modelo de la red escandinava para simulaciones en PSS[®]e y se desarrolló, además, un modelo *Simulink* basado en los datos de ese modelo para poder ejecutar simulaciones sin algunos problemas numéricos encontrados en PSS[®]e.

Se concluyó que las respuestas de frecuencia de los modelos se acercan a la respuesta de la frecuencia real, confirmando la validez de los modelos para el análisis de la frecuencia; y que la inercia en el sistema tiene un fuerte impacto en el período y amplitud de las oscilaciones estudiadas.

En el modelo de Simulink se implementaron bandas muertas en los reguladores de frecuencia y se descubrió que éstas amplifican la amplitud de las oscilaciones, alargando además el tiempo de duración de las mismas.

Además, en el modelo en Simulink desarrollado en este trabajo se representaba al sistema eléctrico de la misma manera que se puede modelar un generador convencional con regulación de frecuencia. Esto sugiere que es posible representar al sistema como si fuera una única máquina, y es posible encontrar un modelo simplificado, como un modelo uninodal, del sistema global.

2.4.3. Estimación de la inercia

En [8] se presenta un procedimiento para estimar la inercia total del sistema eléctrico de Gran Bretaña a partir de transitorios de frecuencia luego de una pérdida instantánea de generación, utilizando medidas de sincrofasores tomadas en diversos puntos de la red.

En primera instancia se detectan eventos adecuados de desbalances de potencia para el análisis y luego se filtran los transitorios medidos para obtener una estimación de la inercia para una región determinada de la

red. La inercia total para todo el sistema se calcula luego como la suma de estas estimaciones.

2.4.4. Equivalentes dinámicos y algoritmos evolutivos

[9] propone realizar el cálculo de equivalentes dinámicos de sistemas eléctricos de potencia mediante la aplicación de algoritmos evolutivos.

El trabajo se realiza con el objetivo de construir un modelo equivalente dinámico del sistema eléctrico argentino, para la realización de estudios dinámicos en el sistema uruguayo, interconectado con el sistema argentino.

Se concluyó que los algoritmos evolutivos pueden tener un buen desempeño en la optimización de soluciones en los sistemas eléctricos de potencia, además de simplificar la búsqueda de equivalentes dinámicos con respecto a la que proponen los métodos tradicionales.

Parte II

Marco teórico

Capítulo 3

Respuesta de la frecuencia en un SEP

En este capítulo se estudia la respuesta de la frecuencia en un SEP a partir de un desbalance entre generación y demanda, partiendo de un modelo teórico de generador y arribando —luego de introducir nociones de control de frecuencia y algunas simplificaciones— a un modelo «equivalente» para los generadores presentes en el SEP a partir de los generadores individuales.

En las secciones 3.1 y 3.2 se presentan los modelos de los generadores y las cargas que componen un SEP y cómo éstos interactúan al estar conectados en la misma red. Estas nociones se extraen de [10].

En 3.3 se define la «frecuencia» de un SEP —que se usará en el resto del trabajo— a partir del concepto de *centro de inercia* y de los modelos anteriormente descritos. Este concepto se extrae de [11].

En 3.4 se presentan los sistemas de regulación de frecuencia, definiendo algunas cantidades asociadas que serán objeto de estudio en capítulos siguientes, y la necesidad de la participación de más de un generador en el control de frecuencia. Se siguen los pasos de [12].

Finalmente en la sección 3.5, se deduce un modelo simplificado que representa la dependencia de la frecuencia con la variación de la potencia generada-demanda de la red desvinculándose de los modelos individuales, y se presentan las características de la respuesta de la frecuencia ante un desbalance de potencia en forma de escalón.

El capítulo no pretende ser un estudio sobre regulación de frecuencia, sino sólo exponer el modelo que luego se usará. Para profundizar en los temas que se ven a continuación se recomienda consultar [10] y [12].

3.1. Generador síncrono

Se le llama generador síncrono al conjunto formado por el acople de una turbina y una máquina síncrona, y es el principal elemento utilizado en un sistema eléctrico de potencia (SEP) como regulador de frecuencia.

Un nombre alternativo en la industria es el de *generador convencional*, para contrastarlo con máquinas generadoras de otras especies como, por ejemplo, generadores desprovistos de una máquina síncrona.

Por su principio de funcionamiento, la velocidad (mecánica) del eje de una máquina síncrona ω es directamente proporcional a la frecuencia eléctrica $f = \frac{p}{2\pi}\omega$ de la red eléctrica a la que está conectada (midiendo f en hercios), siendo p la cantidad de pares de polos de la máquina.

3.1.1. Constante de inercia

En funcionamiento normal, el eje de la máquina esta sometido a un *par mecánico* o *motor* τ_m ejercido por la turbina, tendiente a aumentar la velocidad en el eje; y a un *par electromagnético* o *resistente* τ_e que tiende a disminuirla.

De esta interacción de la máquina con la turbina y la red, aplicando segunda cardinal, surge la ecuación que describe el movimiento del eje:

$$J \frac{d\omega}{dt} = \tau_m - \tau_e \quad (3.1)$$

siendo ω la velocidad angular mecánica del rotor de la máquina y J el momento de inercia del conjunto turbina–eje–rotor.

Introduciendo ω_0 como la velocidad de referencia para el generador, se expresa la ecuación (3.1) en función del desplazamiento respecto de la velocidad de referencia $\Delta\omega = \omega - \omega_0$:

$$\frac{d\omega}{dt} = \frac{d(\omega - \omega_0)}{dt} = \frac{d\Delta\omega}{dt} \triangleq \Delta\dot{\omega} \text{ (}^1\text{)}. \quad (3.2)$$

Al ser proporcionales la velocidad del eje del generador y la frecuencia de la red eléctrica a la cual se conecta, las variaciones de frecuencia Δf y de velocidad $\Delta\omega$ son equivalentes en valores p.u. (Δf [p.u.] = $\Delta\omega$ [p.u.]):

$$\frac{\Delta\omega}{\omega_0} = \frac{\omega - \omega_0}{\omega_0} = \frac{p/2\pi * f - p/2\pi * f_0}{p/2\pi * f_0} = \frac{\Delta f}{f_0}.$$

Tomando como potencia base S_{base} (habitualmente la potencia aparente nominal del generador), como velocidad base $\omega_{base} = \omega_0$ y como par

¹Se estila $\Delta\dot{\omega}$ como la derivada temporal de $\Delta\omega$.

base $\tau_{base} = S_{base} / \omega_{base}$, de la ecuación (3.1) se deduce

$$\frac{J\omega_{base}^2}{S_{base}} \cdot \frac{\Delta\dot{\omega}}{\omega_{base}} = \frac{\tau_m - \tau_e}{\tau_{base}} \stackrel{\text{(def)}}{\Rightarrow} \boxed{2H\Delta\dot{\omega} [\text{p.u.}] = \tau_m [\text{p.u.}] - \tau_e [\text{p.u.}]} \quad (3.3)$$

donde se define la *constante de inercia* H como

$$H = \frac{\frac{1}{2}J\omega_{base}^2}{S_{base}},$$

que representa la energía cinética acumulada en el eje del generador; como se verá, es un parámetro central en el control de sistemas eléctricos.

Dado que $[J] = \text{kg m}^2$, $[\omega_{base}] = \text{rad/s}$ y $[\frac{\omega_{base}}{S_{base}}] = (\text{kg m}^2/\text{s}^2)^{-1}$, resulta en que la inercia se mide en segundos:

$$[H] = \text{s}.$$

3.1.2. La ecuación de *swing*

En la práctica, para un sistema de control la potencia eléctrica es una cantidad más simple de medir que el par electromagnético, por lo que es conveniente expresar la ecuación (3.3) en términos de potencia en lugar de par.

Siendo $P = \omega\tau$ donde τ es el par neto, se considera una desviación a partir de un estado inicial

$$P_0 + \Delta P = (\omega_0 + \Delta\omega)(\tau_0 + \Delta\tau). \quad (3.4)$$

Tomando sólo variaciones y despreciando los términos de segundo orden, se obtiene

$$\Delta P = \Delta\omega\tau_0 + \omega_0\Delta\tau. \quad (3.5)$$

En el eje de la máquina, la expresión anterior se traduce en:

$$\Delta P_m - \Delta P_e = \Delta\omega(\tau_{m_0} - \tau_{e_0}) + \omega_0(\Delta\tau_m - \Delta\tau_e); \quad (3.6)$$

y dado que en régimen de equilibrio $\tau_{m_0} = \tau_{e_0}$, y en p.u. $\omega_0 = 1$, resulta

$$\Delta P_m - \Delta P_e = \Delta\tau_m - \Delta\tau_e. \quad [\text{p.u.}] \quad (3.7)$$

Por tanto, considerando pequeños incrementos alrededor del régimen permanente, podemos escribir la ecuación (3.3) como

$$\boxed{\Delta\dot{\omega} = \frac{1}{2H} (\Delta P_m - \Delta P_e)} \quad (3.8)$$

llamada *ecuación de swing*, que establece la relación entre la rapidez de cambio del desplazamiento con la constante de inercia del motor y el desbalance de potencia.

Constante de amortiguamiento

Si bien la ecuación (3.8) expresa una relación lineal entre la derivada del desplazamiento con respecto a la velocidad nominal y el desbalance de potencia (con pendiente $\frac{1}{2H}$), en general existe un efecto de amortiguamiento en los generadores debido, por ejemplo, al efecto del rozamiento dinámico en el rotor y el acoplamiento con la turbina; o a la presencia del bobinado en el rotor de estas máquinas que tiene como objetivo disminuir las oscilaciones propias producidas en eventos de perturbaciones en la red —como son los casos de desbalances de potencias—.

Se define la constante k_D para representar este efecto:

$$2H\Delta\dot{\omega} + k_D\Delta\omega = (\Delta P_m - \Delta P_e). \quad (3.9)$$

Hay también un efecto de amortiguamiento debido a la presencia de las cargas, incluso más notorio. Se verá entonces que la variación de frecuencia en un SEP tiene un comportamiento amortiguado; éste será representado en el modelo uninodal con la constante de amortiguamiento de los generadores k_D de la ecuación (3.9), que englobará ambos efectos.

3.2. Modelado de cargas

Se resumen en esta sección conceptos extraídos de [10].

Existen dos tipos de modelos de carga:

- modelos de carga estáticos, y
- modelos de carga dinámicos.

Los modelos de carga estáticos son ampliamente utilizados debido a que presentan una complejidad menor: utilizan ecuaciones algebraicas a diferencia de los dinámicos, que utilizan ecuaciones diferenciales.

Los modelos estáticos tienen una alta flexibilidad y son capaces de representar distintos tipos de cargas; los dinámicos se utilizan para casos más específicos y anteponen la veracidad física a la flexibilidad.

3.2.1. Modelos de carga estáticos

Representan la potencia activa y reactiva de la carga en cualquier instante mediante ecuaciones algebraicas dependientes de la tensión y la frecuencia para un instante de tiempo.

$$\begin{aligned} P &= P(P_0, Q_0, V_0, f_0, V, f) \\ Q &= Q(P_0, Q_0, V_0, f_0, V, f) \end{aligned} \quad (3.10)$$

en donde:

- P : potencia activa
- Q : potencia reactiva
- V : tensión en barra
- f : frecuencia en barra

El subíndice «0» indica valores iniciales o de estado estable del sistema.

Los modelos estáticos más comunes son el exponencial y el polinomial.

Modelo exponencial

Este modelo representa la dependencia de la potencia demandada a partir de una expresión exponencial respecto de la tensión en su nodo.

$$\begin{cases} P = P_0 \left(\frac{V}{V_0}\right)^a, \\ Q = Q_0 \left(\frac{V}{V_0}\right)^b. \end{cases} \quad (3.11)$$

Algunos casos particulares:

Potencia constante	$a = 0$ $b = 0$	$P = P_0$ $Q = Q_0$
Corriente constante	$a = 1$ $b = 1$	$P = P_0 \left(\frac{V}{V_0}\right)$ $Q = Q_0 \left(\frac{V}{V_0}\right)$
Impedancia constante	$a = 2$ $b = 2$	$P = P_0 \left(\frac{V}{V_0}\right)^2$ $Q = Q_0 \left(\frac{V}{V_0}\right)^2$

El comportamiento real de las cargas eléctricas no se encuentra asociado típicamente a un valor entero en el exponente, ya que su comportamiento no obedece a un modelo tan sencillo. Sin embargo, por simplicidad, se asocia un comportamiento de impedancia constante —por ejemplo— a una lámpara incandescente, y un comportamiento de potencia constante a un motor. De esta misma manera, una carga residencial se asemeja a un comportamiento de impedancia constante, y una carga industrial se acerca más a una de potencia constante.

Capítulo 3. Respuesta de la frecuencia en un SEP

Modelo polinomial

El modelo polinomial (o ZIP) muestra la relación entre la potencia y la tensión mediante una ecuación polinomial, y corresponde a una combinación lineal de los modelos de carga a impedancia (z), corriente (i) y potencia (p) constantes descritos anteriormente. Es ampliamente utilizado dado que permite representar fielmente el comportamiento de las cargas.

Los coeficientes $P_{Z,I,P}$ y $Q_{Z,I,P}$, de la ecuación (3.12) representan la proporción de cada una de las componentes ($P_Z + P_I + P_P = 1$ y $Q_Z + Q_I + Q_P = 1$).

$$\begin{cases} P^{ZIP} = P_0 \left(P_Z \left(\frac{V}{V_0} \right)^2 + P_I \left(\frac{V}{V_0} \right) + P_P \right), \\ Q^{ZIP} = Q_0 \left(Q_Z \left(\frac{V}{V_0} \right)^2 + Q_I \left(\frac{V}{V_0} \right) + Q_P \right). \end{cases} \quad (3.12)$$

En la práctica, cuando una carga representa un conjunto de puntos de demanda —tanto del tipo residencial como industrial—, se suelen utilizar modelos polinomiales para modelar esa carga de forma conjunta. Es práctica común en la industria, a partir de estimaciones empíricas, ponderar las cargas como de 70 % de potencia constante (tipo industrial) y 30 % de impedancia constante (tipo residencial) [13].

Carga dependiente de la frecuencia

La dependencia de la carga con la frecuencia se representa multiplicando el modelo exponencial o polinomial por un factor:

$$\begin{cases} P = P(V) * (1 + K_{pf} \Delta f), \\ Q = Q(V) * (1 + K_{qf} \Delta f). \end{cases} \quad (3.13)$$

De la ecuación (3.13), la relación entre el incremento de demanda total en las cargas P_C y el incremento de frecuencia en un sistema puede expresarse como $\Delta P_C = \Delta P_I + D \Delta \omega$, donde

- ΔP_I es el incremento de potencia independiente de la frecuencia, y
- D es la constante que actúa como mecanismo de amortiguamiento de la variación de frecuencia.

En consecuencia, todo aumento de frecuencia $\Delta \omega$ provocado por la disminución de ΔP_e resulta en un aumento de la demanda que se opone al incremento de frecuencia inicial.

Algunas cargas típicas que presentan este comportamiento son, por ejemplo, los motores de inducción, que también aportan inercia a la red (efecto que hasta este punto no ha sido considerado).

3.2.2. Cargas con inercia: motor de inducción

Las máquinas rotativas conectadas directamente a la red se comportan de forma similar a los generadores, aportando inercia al sistema: el rotor de una máquina de inducción, funcionando como motor, está sometido a un *par resistente* τ_r que en este caso tiende a frenar el eje del mismo y un *par electromagnético* τ_e que tiende a acelerarlo y depende de la potencia P_e que demanda la máquina.

En forma análoga al caso de máquina síncrona, la ecuación (3.14) representa la dinámica de los motores de inducción.

$$\Delta P_e = 2H\Delta\dot{\omega} + k_D\Delta\omega + \Delta P_r \quad (3.14)$$

con P_r la potencia que determina el par τ_r .

De forma general se puede decir que las cargas se comportan como se describe en la ecuación

$$\Delta P_C = \Delta P_l(E) + 2H\Delta\dot{\omega} + k_D\Delta\omega \quad (3.15)$$

siendo:

- P_C la potencia total que demanda la carga;
- P_l la potencia que demanda la carga en función de la tensión E en nodo de la misma;
- H la constante de inercia de las máquinas; y
- k_D la constante de amortiguamiento de las máquinas.

3.3. Velocidad del centro de inercia

Se ha descrito el comportamiento de la frecuencia en los elementos individuales de un sistema. En régimen, la frecuencia en todos los nodos de una red coincide; sin embargo, en eventos de desequilibrios de potencia, cada elemento tiene un comportamiento diferente —a pesar del vínculo que existe al estar conectados a la misma red—, asociado a las leyes físicas que lo gobiernan. Esto deriva en que la frecuencia (o velocidad de giro en el caso de máquinas rotativas) tendrá valores diferentes en distintos puntos de la red. Para esta situación existen diversas formas de definir la «frecuencia del sistema».

Se generalizará ahora lo visto a un conjunto de elementos mediante la introducción del concepto de *centro de inercia*, siguiendo la técnica descrita en [11], y se definirá lo que en el presente trabajo se denominará *frecuencia del sistema* como una magnitud asociada al mismo.

Capítulo 3. Respuesta de la frecuencia en un SEP

El esquema de la figura 3.1 representa en forma generalizada un SEP y los flujos de potencia entre los elementos y la red eléctrica.

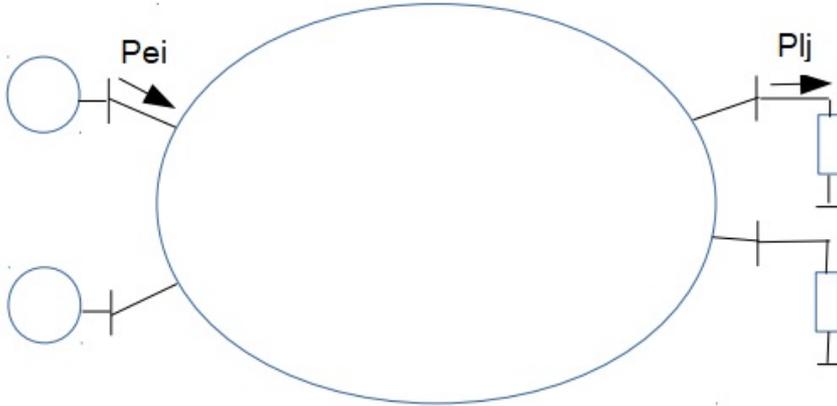


Figura 3.1: Esquema representativo de generación y demanda de un SEP. Imagen tomada de [11].

El generador i , que se asume es una máquina rotativa conectada directamente a la red, recibe una potencia mecánica P_{m_i} proveniente de una máquina de arrastre, y entrega a la red una potencia eléctrica P_{e_i} .

Entonces, para el conjunto de generadores síncronos presentes en el sistema:

- $P_e = \sum_i P_{e_i}$ y
- $P_m = \sum_i P_{m_i}$.

Por otro lado, la carga j está demandando una potencia activa P_{C_j} de la red, la cual se divide en cargas del tipo polinomial o ZIP y máquinas rotativas: $P_C = \sum_j P_{C_j}$.

Definiendo $\Psi = \sum_k \Psi_k$ como las pérdidas eléctricas en la red, el sistema en régimen debe cumplir con la condición del balance de potencia:

$$\sum_i P_{e_i} = \sum_j P_{C_j} + \sum_k \Psi_k \Rightarrow \boxed{P_e = P_C + \Psi.} \quad (3.16)$$

Es decir, la potencia total que el sistema demanda a los generadores síncronos P_e es la potencia total que demandan las cargas P_C más las pérdidas Ψ presentes en el sistema.

Por otro lado, si en el sistema hay presencia de generación no convencional, el balance de potencia es: $P_e = P_C + \Psi - P_{ERN C}$.

3.3. Velocidad del centro de inercia

Modelado del generador i Se modela con la ecuación

$$2H_i\Delta\dot{\omega}_i + k_{D_i}\Delta\omega_i = \Delta P_{m_i} - \Delta P_{e_i} \quad (3.17)$$

siendo:

- ω_i la velocidad en el eje;
- H_i la constante de inercia del generador;
- k_{D_i} la constante de amortiguamiento asociada;
- P_{e_i} la potencia eléctrica entregada a la red;
- P_{m_i} la potencia mecánica entregada por la máquina primaria al generador.

Modelado de la carga j

$$\Delta P_{C_j} = \Delta P_{l_j}(E_j) + 2H_j\Delta\dot{\omega}_j + k_{D_j}\Delta\omega_j \quad (3.18)$$

siendo:

- ω_j la velocidad de giro (si es una máquina rotativa);
- P_{C_j} la potencia total demandada;
- P_{l_j} la potencia independiente de la frecuencia, en función de la tensión E_j ;
- E_j la tensión en nodo de la carga;
- H_j la constante de inercia de la máquina (si es una máquina rotativa);
- k_{D_j} la constante de amortiguamiento asociada.

Lo que se busca es una ecuación que describa en forma global el comportamiento de la frecuencia, logrando independencia de las potencias individuales P_{m_i} y P_{e_i} e incluso de cada ω_i . Se busca analizar el caso cuando se produce un desequilibrio entre P_m y P_e .

3.3.1. Centro de inercia

El *centro de inercia* δ_G se define de forma análoga al centro de masas en mecánica clásica como:

$$\delta_G \triangleq \frac{\sum_i H_i \delta_i + \sum_j H_j \delta_j}{\sum_i H_i + \sum_j H_j} \triangleq \frac{\sum_i H_i \delta_i + \sum_j H_j \delta_j}{H} \quad (3.19)$$

donde:

Capítulo 3. Respuesta de la frecuencia en un SEP

- δ_i es el ángulo del generador i y δ_j el ángulo de la máquina j ; y
- $H = \sum_i H_i + \sum_j H_j$ es la *inercia total* del sistema.

Derivando respecto del tiempo, se obtiene una expresión para la *velocidad del centro de inercia*:

$$\frac{d\delta_G}{dt} \triangleq \omega_G = \frac{\sum_i H_i \omega_i + \sum_j H_j \omega_j}{H}. \quad (3.20)$$

El sistema de ecuaciones (3.21) representa la dinámica de la red, utilizando los modelos de generador y carga descritos en las ecuaciones (3.17) y (3.18).

$$\begin{cases} 2H_i \Delta \dot{\omega}_i + k_{D_i} \Delta \omega_i = \Delta P_{m_i} - \Delta P_{e_i}, \\ 2H_j \Delta \dot{\omega}_j + k_{D_j} \Delta \omega_j = \Delta P_{C_j} - \Delta P_{l_j}. \end{cases} \quad (3.21)$$

Para lograr un sistema lineal, se supone que los coeficientes k_{D_i} son proporcionales a su inercia H_i ; es decir $\frac{k_{D_i}}{H_i} = \frac{k_{D_j}}{H_j} \triangleq \frac{D}{H}$ para todo i y j .

Por lo anterior queda definida la *constante de amortiguamiento equivalente* del sistema como: $D = \sum_i k_{D_i} + \sum_j k_{D_j}$.

Sumando a ambos lados, se puede reducir (3.21) a una única expresión.

En el lado izquierdo se obtiene:

$$\left(\sum_i 2H_i \Delta \dot{\omega}_i + \sum_j 2H_j \Delta \dot{\omega}_j \right) + \left(\sum_i k_{D_i} \Delta \omega_i + \sum_j k_{D_j} \Delta \omega_j \right) = 2H \Delta \dot{\omega}_G + D \Delta \omega_G;$$

mientras que en lado derecho:

$$\sum_i \Delta P_{m_i} - \sum_i \Delta P_{e_i} + \sum_j \Delta P_{C_j} - \sum_j \Delta P_{l_j} = \Delta P_m - \Delta P_e + \Delta P_C - \Delta P_l$$

que resulta en

$$\boxed{2H \Delta \dot{\omega}_G + D \Delta \omega_G = \Delta P_m - \Delta P_e + \Delta P_C - \Delta P_l}. \quad (3.22)$$

Teniendo en cuenta la ecuación (3.16), se obtiene la expresión

$$2H \Delta \dot{\omega}_G + D \Delta \omega_G = \Delta P_m - (\Delta P_l + \Delta \Psi) \Rightarrow \boxed{2H \Delta \dot{\omega}_G + D \Delta \omega_G = \Delta P_m - \Delta P} \quad (3.23)$$

3.3. Velocidad del centro de inercia

donde las constantes H y D son la *inercia total o equivalente* y la *constante de amortiguamiento equivalente* respectivamente, mientras que ΔP representa la variación de potencia total que el sistema le demanda a los generadores síncronos. Si consideramos la generación no convencional, partiendo del balance de la ecuación (3.16), se define como se muestra a continuación:

$$\Delta P = \Delta P_l + \Delta \Psi - \Delta P_{ERNC}.$$

Conclusión

Para obtener la velocidad del centro de inercia en forma exacta es necesario poder medir la velocidad del eje de cada uno de los generadores y motores conectados a la red y, además, conocer la constante de inercia de cada uno de ellos². Esto no es posible en la práctica, pero sí se puede estimar ω_G considerando los elementos de mayor inercia que tengan un aporte sustancial al mismo.

3.3.2. Sistema sin regulación de frecuencia

Se ha definido la frecuencia de un SEP (ecuación (3.20)) y su dependencia con el balance de potencia generación/demanda de una red con la ecuación (3.23). El diagrama de bloques de la figura 3.2 ilustra lo anterior.

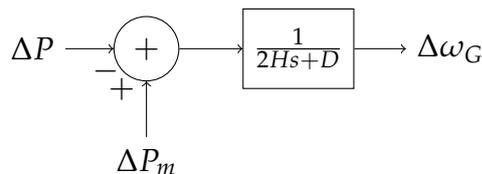


Figura 3.2: Diagrama de bloques del sistema sin regulación

considerando:

- H la inercia equivalente del sistema y
- D la constante de amortiguamiento equivalente del sistema.

²También existen otros elementos que pueden aportar inercia al sistema, por ejemplo:

- de forma electrónica, lo que se conoce como *inercia sintética*;
- generador eólico doblemente alimentado: son máquinas de inducción que se conectan o bien directamente a la red generalmente el bobinado del estator, o bien mediante electrónica de potencia; generalmente en el rotor.

Capítulo 3. Respuesta de la frecuencia en un SEP

Ante un desbalance de potencia ΔP dado en forma de escalón, P_m se mantiene constante; por tanto, la función de transferencia es

$$\frac{\Omega_G(s)}{\Delta P(s)} = -\frac{1}{2Hs + D}$$

siendo $\Omega_G(s)$ y $\Delta P(s)$ las transformadas de Laplace de $\Delta\omega_G$ y ΔP respectivamente.

La frecuencia del sistema tendrá el comportamiento de la figura 3.3, cuyo valor final en régimen es $\Delta\omega = \frac{\Delta P}{D}$.

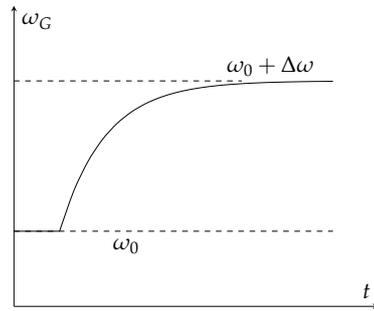


Figura 3.3: Respuesta del sistema sin regulación de frecuencia

El valor final de ω_G puede llegar a ser inadmisibles para un SEP, en particular en escenarios donde se producen escalones de potencia considerables.

La frecuencia del sistema debe permanecer en valores cercanos al nominal, lo cual plantea la necesidad de un control.

3.4. Control de frecuencia

Como se mencionó en 3.1, el par turbina-máquina es el elemento básico en un SEP de regulación de frecuencia. Esto se debe a que controlar la velocidad del eje de una máquina síncrona es equivalente a controlar la frecuencia, dado que son magnitudes proporcionales. En el esquema de la figura 3.4 se observan todos los elementos que participan en la regulación de frecuencia en un sistema turbina-máquina.

El sistema de control toma lectura de la potencia inyectada a la red; así como de la frecuencia de la misma o, en su defecto, de la velocidad del eje de la máquina. A partir de dicha lectura, ordena la acción de un actuador que modifica la potencia que la turbina entrega a la máquina, de forma de mantener la frecuencia dentro de valores admisibles.

3.4. Control de frecuencia

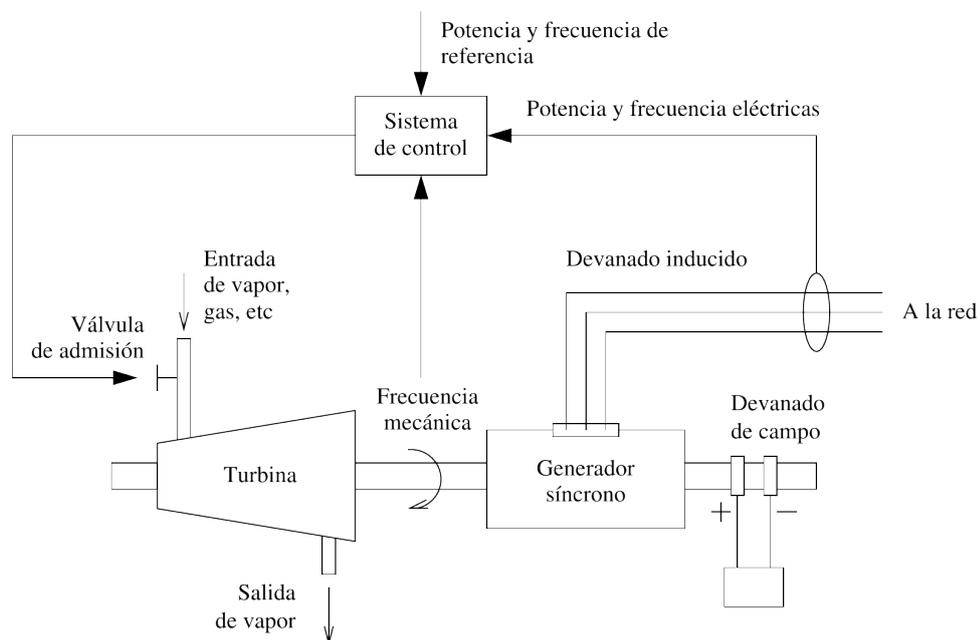


Figura 3.4: Elementos del control de frecuencia en una turbina-generador síncrono. Imagen tomada de[12].

Reservas

Cuando el desbalance producido requiere un aumento de potencia, un parámetro importante es la máxima potencia que puede entregar el conjunto turbina-generador a la red. Estas *reservas de potencia*, dadas por la diferencia entre la potencia máxima y la potencia que está entregando el generador, juegan un papel fundamental en la regulación de frecuencia, siendo el margen que el conjunto de los reguladores posee para actuar de forma autónoma, sin intervención del operador.

Por otro lado, en los casos en los que se requiere una disminución de potencia, es necesario que en ese momento el generador se encuentre inyectando potencia.

3.4.1. Regulación primaria

A continuación se describe cómo se implementa el control de frecuencia en un generador síncrono partiendo del diagrama 3.5, que resulta de la ecuación de *swing* para los generadores (ecuación (3.9)).

Cuando en el control de frecuencia participa un único generador, el sistema de control puede ser implementado mediante un *regulador isócrono* [12], el cual es capaz de llevar la frecuencia al valor de referencia del sistema. Sin embargo, cuando en la regulación participan otros gene-

Capítulo 3. Respuesta de la frecuencia en un SEP

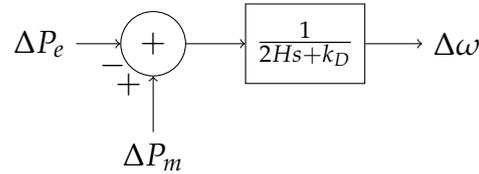


Figura 3.5: Diagrama de bloques de un generador sin regulación de frecuencia

radores, se genera una situación en la que los mismos compiten entre sí para alcanzar su propia velocidad de referencia, generando un comportamiento inestable en el sistema.

Para un SEP cuyas dimensiones no permiten que la regulación de frecuencia se realice en un único punto —dado que los recursos de una máquina suelen no ser suficientes para hacer frente a los desbalances de potencia que se pueden presentar a nivel del sistema—, la solución es usar un *sistema de control mediante estatismo* (figura 3.6).

Este control se conoce también como *regulación primaria*, dado que para restablecer el estado original del sistema —es decir, llevar la frecuencia nuevamente al valor de referencia $\Delta\omega = 0$ — es necesaria la actuación de un segundo control (*regulación secundaria*).

Ante una variación ΔP en la potencia demandada, el SEP debe variar la potencia generada el mismo ΔP para lograr restablecer el equilibrio generación-demanda, estabilizando así la frecuencia.

Para conseguir esto, en cada generador síncrono con regulación primaria —de forma independiente—, el sistema de control modifica la potencia mecánica P_m que la turbina entrega a la máquina en función de la variación de la frecuencia a razón de una constante, R , que se define como *estatismo*:

$$\Delta P_m = -\frac{1}{R}\Delta\omega \text{ [p.u.]}$$

siendo $\Delta\omega$ el valor final de la frecuencia luego de restablecido el balance.

De esta manera, cada generador aporta, en función de su estatismo, a la regulación de frecuencia en el SEP.

R es la pendiente de la curva de la figura 3.6 y definiendo:

- la ω_v frecuencia de vacío ($P_{gen} = 0$ [p.u.]); y
- la ω_{pc} frecuencia a plena carga ($P_{gen} = 1$ [p.u.]).

3.4. Control de frecuencia

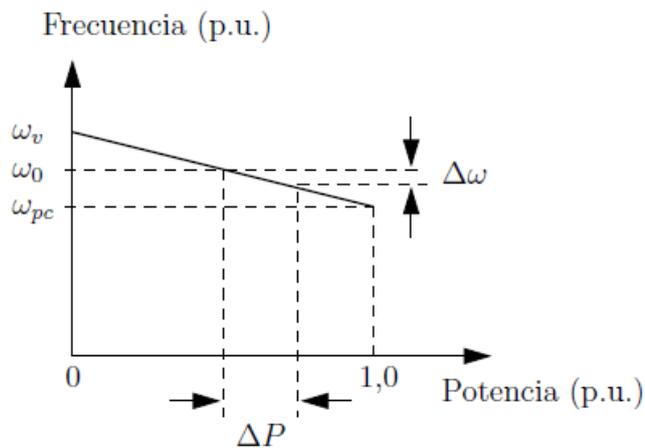


Figura 3.6: Control primario con estatismo en un generador síncrono. Imagen tomada de [12].

se puede calcular a partir de:

$$R = \frac{\omega_v - \omega_{pc}}{\omega_0}$$

Modelo dinámico de regulador primario

La dinámica del regulador primario consiste en la realimentación de la figura 3.7,

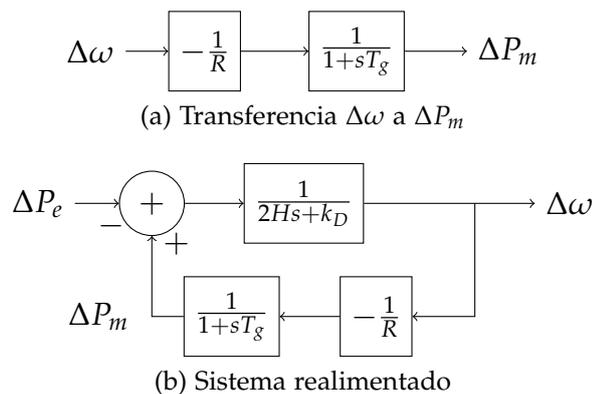


Figura 3.7: Diagrama de bloques de la regulación primaria en un generador

siendo:

- H y k_D las constantes del modelo del generador síncrono;
- P_e y P_m las potencias demandada al generador y entregada por la turbina respectivamente;

Capítulo 3. Respuesta de la frecuencia en un SEP

- R el *estatismo*; y
- T_g el tiempo de retardo de la realimentación.

Debe destacarse que el diagrama de bloques presentado no tiene en cuenta los fenómenos físicos que se dan desde el momento de intervención del actuador hasta que efectivamente se produce el incremento/disminución de la potencia mecánica de la turbina: estos fenómenos introducen efectos de orden superior que no forman parte de este análisis.

Considerando la ecuación (3.9), como $\Delta P_m = -\frac{1}{R}\Delta\omega$ y con la hipótesis de $k_D \ll \frac{1}{R}$, en régimen ($\Delta\dot{\omega} = 0$) la frecuencia resulta:

$$k_D\Delta\omega = -\frac{1}{R}\Delta\omega - \Delta P_e \Rightarrow \Delta\omega = -\frac{\Delta P_e}{k_D + \frac{1}{R}} \simeq -R\Delta P_e. \quad (3.24)$$

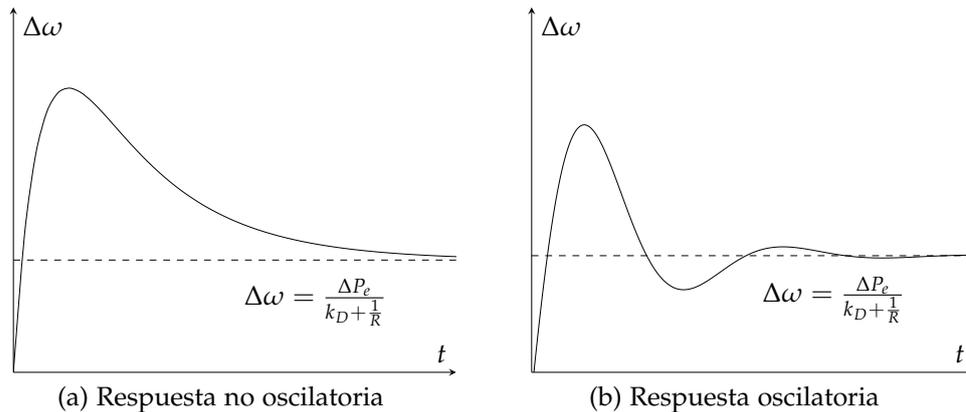


Figura 3.8: Respuesta de la velocidad de una máquina con regulación primaria de frecuencia

Por tanto, este sistema de control restablece la frecuencia a un valor que, si bien difiere del de referencia, es aceptable para el sistema y logra eliminar el desbalance de potencia.

3.5. Regulación primaria en un SEP

La regulación mediante estatismo permite entonces la participación de varias máquinas al mismo tiempo en la regulación de frecuencia en un SEP.

Para dos máquinas con regulación primaria, en el caso que se presenta en la figura 3.9, se dio un evento de desequilibrio entre generación y demanda ($P_{dem} > P_{gen}$), lo que generó una caída en la frecuencia.

3.5. Regulación primaria en un SEP

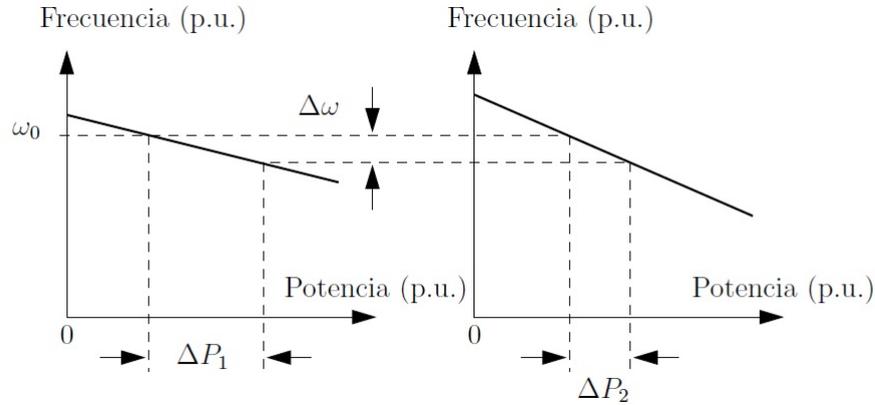


Figura 3.9: Regulación conjunta de frecuencia mediante estatismo. Imagen tomada de [12].

Para compensar ese desequilibrio, cada generador aumenta la potencia entregada ΔP_i en función de su estatismo R_i hasta llegar nuevamente al equilibrio. Se tiene, por un lado el aporte de ambos generadores:

$$\Delta P_{tot} = \Delta P_1 + \Delta P_2$$

y la frecuencia final resulta en

$$\Delta\omega = R_1\Delta P_1 = R_2\Delta P_2.$$

3.5.1. Participación de múltiples generadores en la regulación primaria

En la sección 3.3 se vio que para un sistema con n máquinas (y m cargas), podemos encontrar un modelo con parámetros equivalentes H_{eq} y D_{eq} que representan fielmente la relación entre la variación de potencia y frecuencia del sistema sin regulación (figura 3.2).

Se conoce también cómo se modela la regulación primaria para un generador en forma individual (ver figura 3.7).

Surge entonces el diagrama de la figura 3.10, considerando la participación de $\tilde{n} \leq n$ generadores síncronos en la regulación primaria de frecuencia:

Para los generadores que cuenten con regulador se cumplirá:

$$P_{m_i}(s) = -\frac{1}{R_i} \frac{1}{1 + T_{gi}s} \Omega(s)$$

siendo $P_{m_i}(s) = \mathcal{L}\{\Delta P_{m_i}\}$ y $\Omega(s) = \mathcal{L}\{\Delta\omega\}$ las transformadas de Laplace de ΔP_{m_i} y $\Delta\omega$ respectivamente. Por otro lado $P_m = \sum_i P_{m_i}$, de lo que

Capítulo 3. Respuesta de la frecuencia en un SEP

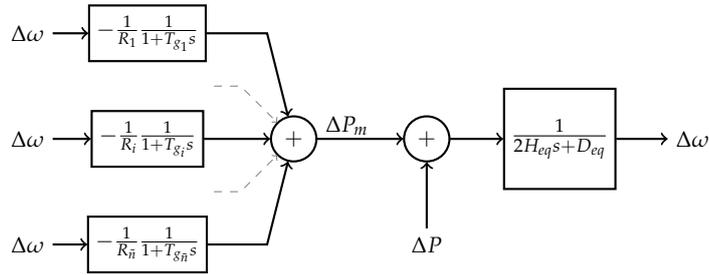


Figura 3.10: Regulación primaria con múltiples generadores

resulta que el sistema con entrada $\Delta\omega$ y salida ΔP_m tiene como función de transferencia:

$$P_m(s) = - \sum_i \frac{1}{R_i} \frac{1}{1 + T_{gi}s} \Omega(s). \quad (3.25)$$

Interesa encontrar, tal como se hizo para H y D , parámetros equivalentes de la red R_{eq} y T_{geq} que modelen de forma única los múltiples R y T_g (posiblemente de forma aproximada, despreciando efectos de orden superior); de forma de simplificar el sistema a un modelo de una sola máquina equivalente.

Esta búsqueda se justifica de forma empírica al observar comportamientos ante faltas de SEP reales: en la figura 3.11 se muestra la respuesta de la frecuencia ante una falta en un sistema eléctrico real [7], donde se observa una curva oscilatoria similar a la de la figura 3.8.

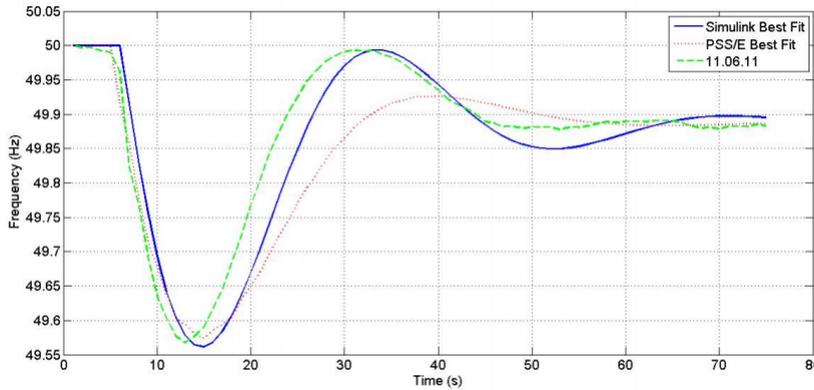


Figura 3.11: Evento en la red escandinava, 11/06/2011 (verde). Imagen tomada de [7].

La equivalencia buscada es

$$\frac{P_m(s)}{\Omega(s)} = - \sum_i \frac{1}{R_i} \frac{1}{1 + T_{gi}s} \simeq - \frac{1}{R_{eq}} \frac{1}{1 + T_{geq}s}. \quad (3.26)$$

3.5. Regulación primaria en un SEP

Estimación de R_{eq} y T_{geq}

Se analiza la transferencia $\frac{P_m(s)}{\Omega(s)}$ de (3.26) para distintos rangos de frecuencias. En este punto se habla de *frecuencia de la transferencia*; no debe confundirse con la *frecuencia de la red*, que es el objeto de estudio en esta tesis.

Para frecuencias de régimen, es decir frecuencias bajas de la respuesta cuando $t \rightarrow \infty$ y $s \rightarrow 0$, cada generador cumplirá $\Delta P_{m_i} \rightarrow -\frac{\Delta\omega}{R_i}$ (sección 3.4.1); por tanto, la transferencia para frecuencias bajas es $\frac{1}{R_i}$.

$$\text{Como } \Delta P_m = \sum_i \Delta P_{m_i},$$

$$\begin{aligned} \Delta P_m &= \sum_i -\frac{1}{R_i} \Delta\omega = -\frac{1}{R_{eq}} \Delta\omega \\ \Rightarrow \frac{1}{R_{eq}} &= \sum_i \frac{1}{R_i}. \end{aligned} \quad (3.27)$$

Se deduce de la ecuación (3.27) que R_{eq} es equivalente al paralelo de los coeficientes R_i :

$$\boxed{R_{eq} \simeq \parallel_i R_i.}$$

Por otro lado, tomando como hipótesis que los T_{gi} y T_{geq} son del mismo orden de magnitud, consideramos ahora las frecuencias altas; es decir, la respuesta inmediata a la falta con t pequeño:

$$\left. \begin{aligned} -\sum_i \frac{1}{R_i} \frac{1}{1 + T_{gi}s} &\simeq -\frac{1}{R_{eq}} \frac{1}{1 + T_{geq}s} \\ \text{Para } t \text{ pequeños, } sT_g \gg 1 \text{ y:} \\ -\sum_i \frac{1}{R_i} \frac{1}{T_{gi}s} &\simeq -\frac{1}{R_{eq}} \frac{1}{T_{geq}s} \end{aligned} \right\} \frac{1}{T_{geq}} \simeq R_{eq} \sum_i \frac{1}{R_i} \frac{1}{T_{gi}}. \quad (3.28)$$

Se concluye que se puede estimar T_{geq} como el paralelo de los coeficientes T_{gi} ponderados por los R_i :

$$\boxed{T_{geq} \simeq \frac{1}{R_{eq}} \parallel_i R_i T_{gi}.$$

3.5.2. Función de transferencia

Se ha llegado así a un modelo aproximado unificado del comportamiento de la frecuencia bajo un control primario de un SEP, que engloba los elementos individuales que lo componen.

Capítulo 3. Respuesta de la frecuencia en un SEP

Para expresarlo es necesario entonces conocer todos los elementos del sistema que aportan inercia H y amortiguamiento D , la velocidad de todos los ejes en movimiento de máquinas que se conectan directamente a la misma —para encontrar la velocidad del centro de inercia—, y las constantes R y T_g que modelan los sistemas de control de regulación primaria de frecuencia.

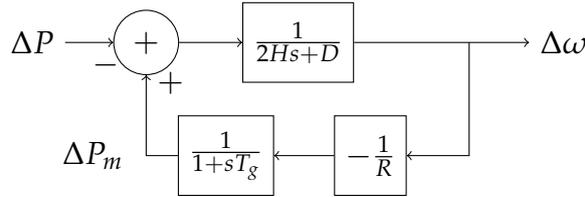


Figura 3.12: Diagrama del sistema con regulación primaria

En la figura 3.12 se muestra el diagrama de bloques final, que representa un SEP que cuenta con regulación primaria distribuida de frecuencia. Todas las magnitudes están en [p.u.], siendo:

- H la inercia total o equivalente del sistema,
- D la constante de amortiguamiento del sistema,
- R el estatismo permanente equivalente de la regulación del sistema,
- T_g el tiempo de retardo equivalente de la regulación del sistema,
- $\Delta\omega$ la variación de la velocidad del centro de inercia, o bien la velocidad del centro de inercia, en pequeña señal, respecto a ω_0 , y
- ΔP la variación de la potencia demandada por las cargas modeladas como cargas polinomiales.

Despejando del diagrama de la figura 3.12 se obtiene, si $\Omega(s) = \mathcal{L}\{\Delta\omega\}$ es la transformada de Laplace de $\Delta\omega$, y $P(s) = \mathcal{L}\{\Delta P\}$ la de la variación de potencia, la ecuación

$$\Omega(s) = \frac{1}{2Hs + D} \left(-P(s) - \frac{1}{R} \frac{1}{T_g s + 1} \Omega(s) \right)$$

desde donde es posible obtener la función de transferencia

$$F(s) = \frac{\Omega(s)}{P(s)} = -\frac{T_g s + 1}{(2Hs + D) + (T_g s + 1) + R^{-1}}. \quad (3.29)$$

3.5.3. Características de la respuesta

La respuesta de la frecuencia ante un desbalance de potencia en forma de escalón, a partir de la ecuación (3.29), es el que se muestra en la figura 3.13.

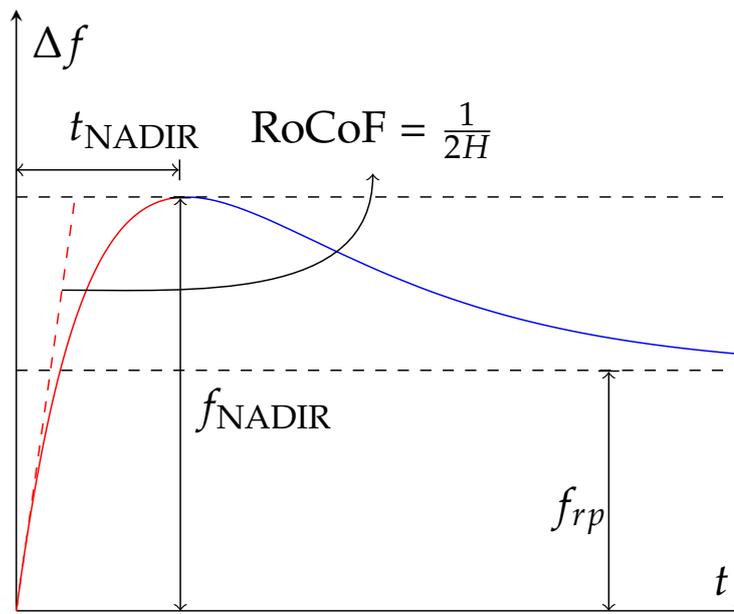


Figura 3.13: Características de la respuesta de la frecuencia

Etapa 1: respuesta inercial (roja) Respuesta natural de las máquinas síncronas, similar a lo visto en la figura 3.3. La velocidad con la que disminuye/aumenta la frecuencia inicialmente, denominada habitualmente *ROCOF inicial*, depende de la inercia H del sistema (ver ecuación (5.8)); mientras que el valor extremo que se alcanza, denominado *frecuencia de nadir* f_{NADIR} ³, dependerá principalmente de la constante de amortiguamiento D . Esta etapa culmina antes de alcanzar la frecuencia de nadir.

Etapa 2: respuesta de control primario (azul) En esta etapa comienzan a verse los efectos del control primario. El valor final de la frecuencia f_{rp} depende principalmente de la constante R , según lo visto en la figura 3.8; mientras que T_g es determinante en el tiempo en que se demora en llegar al régimen permanente.

³Estrictamente *nadir* refiere al valor mínimo de la curva, correspondiente a la respuesta con entrada de escalón de potencia positivo (aumento de demanda o disminución de generación). De todas formas, denominamos también indistintamente *nadir* al punto en que la curva alcanza su valor máximo para los casos de escalones de potencia negativos.

Capítulo 3. Respuesta de la frecuencia en un SEP

Etap 3: respuesta de control secundario El objetivo del control secundario es restablecer la frecuencia a valores nominales. Escapa al interés del presente proyecto, y empieza a actuar una vez que la regulación primaria estabiliza la frecuencia.

Capítulo 4

Algoritmos evolutivos

Los *algoritmos evolutivos* (AA.EE.) o *algoritmos genéticos* son un modelo (*metaheurística*) de optimización útil en problemas con un amplio espacio de soluciones, con un enfoque darwinista inspirado en la evolución biológica, y un fuerte componente aleatorio.

Se ofrece una somera introducción a los conceptos básicos de los AA.EE., pero debe advertirse que no se trata de una guía completa. Para profundizar en el tema, se recomiendan los recursos [14], [15] u otro material introductorio.

4.1. Introducción

Los AA.EE. son útiles para problemas donde se desea optimizar una *función objetivo* $f(x_1, x_2, \dots, x_n)$ —es decir, cuyos extremos en alguna región del espacio se desea encontrar— y donde se carece de una forma explícita de invertirla, pero sí es posible evaluarla en cualquier punto de la región. El problema planteado es entonces encontrar la n -upla x_1, x_2, \dots, x_n tal que optimiza f en el espacio.

Se hace un paralelismo con la evolución biológica, y cada n -upla se considera un *individuo*, conformando las variables x_1, x_2, \dots, x_n su *genoma*. La función objetivo es su *función de aptitud* (o *fitness*), que cuantifica cuán apto es el individuo como solución al problema. Optimizar f es, entonces, encontrar el individuo más apto.

Se postula que para un individuo cuya aptitud es «buena», algunas partes de su genoma x_i serán las causantes de esa aptitud, y que sería posible mejorarla mejorando las otras partes del genoma. Aquí se toma inspiración en el enfoque darwinista, con foco en la cruce y la mutación: las especies se adaptan progresivamente al ambiente a partir de la supervivencia del más apto, mediante la evolución del individuo a través de la evolución de su genoma.

Como mecanismo para mejorar elementos individuales, la *cruza* de dos individuos padres para producir un individuo hijo toma aleatoriamente partes disjuntas del genoma de ambos, mientras que la *mutación* de un individuo consiste en la alteración aleatoria de uno (o más) de sus genes. La *selección* emula la supervivencia del más apto: de entre un conjunto de individuos posibles, se toman aquellos cuya función de aptitud los destaque del resto.

El algoritmo es iterativo: se parte de un conjunto inicial de soluciones o individuos (*generación inicial*), desde donde se hace una selección en base a su aptitud, con un componente aleatorio —no necesariamente tomando exclusivamente los individuos con mejor aptitud, aunque es una práctica posible—. Los individuos así seleccionados son cruzados dos a dos: a partir de dos individuos padres se obtiene un individuo hijo que toma elementos de ambos. Los individuos hijos son agrupados en una nueva generación, y el proceso es repetido desde la selección.

Al cabo de un número predeterminado de generaciones, el proceso se detiene, y se toma una de las soluciones encontradas —en general de la generación final; aunque podría ser el individuo con mejor aptitud histórica— como la *solución* producida por la ejecución. Al haber en cada paso un factor aleatorio, es esperable que corridas sucesivas no produzcan la misma solución, aún partiendo desde los mismos parámetros.

A las manipulaciones genéticas basadas en el darwinismo se las llama comúnmente *operadores genéticos*: cruzamiento, selección, mutación; operan sobre el conjunto de individuos produciendo nuevos.

La función de aptitud o *fitness* en estas condiciones no necesariamente es la función objetivo del problema, pero sí debe estar relacionada directamente, y debe facilitar la optimización de forma que optimizándola se obtenga el valor deseado de la función objetivo.

Si bien «aptitud» es la traducción habitual de *fitness*, en el resto del trabajo se permitirá el anglicismo, debido a ser un término ampliamente extendido en el campo de AA.EE. y que tiene la ventaja de referir exclusivamente a la aptitud darwinista, evitando confundirse con otros significados de «aptitud».

4.2. Influencia de la generación inicial

Una característica de los AA.EE. es que su velocidad de convergencia a una solución óptima está directamente influenciada por la presencia de buen *material genético* en su generación inicial: de partir de un conjunto inicial de soluciones aleatorias, el algoritmo demoraría algunas generaciones en encontrar zonas del espacio de soluciones con buena aptitud,

que podrían ahorrarse si la generación inicial contase con individuos que perteneciesen a tales zonas.

Es por eso que todo conocimiento posible de la solución debe incorporarse a la evolución guiada, en particular en la generación inicial. Si se conocen características de los individuos más aptos de forma aproximada, éstas deben incluirse.

4.3. Elitismo y diversidad

El *elitismo* es la práctica de conservar los individuos más aptos de generación a generación, o —indistintamente— incluir en la selección los individuos más aptos para su cruzamiento. La *diversidad* consiste en conservar individuos no tan aptos, bajo el supuesto de que parte de su genoma, que no es reflejado en la aptitud, sea beneficioso para individuos descendientes.

Empíricamente se ve que una evolución es más exitosa cuando se logra balancear ambas fuerzas. El elitismo está relacionado con la *explotación* de un área de alta aptitud, tomando individuos próximos a un extremo con la confianza de que sus descendientes se puedan aproximar aún más. Por el contrario, la diversidad está relacionada con la *exploración* del espacio de soluciones, permitiendo encontrar otras zonas fértiles para explotar, y evita que el algoritmo caiga en un extremo local y pueda —potencialmente— avanzar hacia un extremo absoluto.

4.4. Formalización

Sea S el dominio de las soluciones al problema P . Los elementos de S representan *individuos*, cada uno de ellos describiendo una solución al problema.

La *función de aptitud* es $f : S \rightarrow \mathbb{R}^+ \cup \{0\}$; y es capaz de asignar a cada individuo un valor numérico que representa la calidad de la solución. El conjunto de salida no es necesariamente los reales no negativos; sino que puede ser cualquier conjunto con elementos ordenados y acotado (por ejemplo \mathbb{N}), donde el valor nulo representa un mínimo teórico (que puede ser no alcanzable en el problema).

El problema P se reduce entonces a encontrar el valor

$$s_{\text{ópt}} = \arg \min_{s \in S} f,$$

aunque se asume que ese valor no es computable de forma explícita.

Capítulo 4. Algoritmos evolutivos

La generación inicial se obtiene en base a una función generatriz $g : \mathbb{A} \rightarrow \mathbb{S}$, donde \mathbb{A} representa una fuente infinita de entropía, representando la aleatoriedad en el proceso. La aplicación repetida de este proceso, o *génesis*, produce un conjunto \mathcal{G}_0 de tamaño N que constituye la generación inicial:

$$G : \mathbb{A} \rightarrow \{\mathcal{G}_0 \in 2^{\mathbb{S}} / |\mathcal{G}_0| = N\}^1.$$

El proceso de *selección* se hace en base a la aptitud de los individuos, aunque también considera aleatoriedad, produciendo un conjunto menor a partir de los cuales se obtendrá la nueva generación:

$$S : \{\mathcal{G}_i \in 2^{\mathbb{S}} / |\mathcal{G}_i| = N\} \rightarrow \{\mathcal{S}_i \in 2^{\mathbb{S}} / |\mathcal{S}_i| < N\}.$$

El *cruzamiento* se ejecuta a partir de los individuos seleccionados, produciendo un número N de *hijos*:

$$C : \mathbb{A} \times \{\mathcal{S}_i \in 2^{\mathbb{S}} / |\mathcal{S}_i| < N\}^2 \rightarrow \{\mathcal{G}_{i+1} \in 2^{\mathbb{S}} / |\mathcal{G}_{i+1}| = N\}.$$

La *mutación* es un proceso opcional en el que una solución es alterada ligeramente a fin de producir nuevas soluciones que pudieran no obtenerse por cruzamiento. La mutación es producida por una función $m : \mathbb{S} \rightarrow \mathbb{S}$, siendo su proceso

$$M : \mathbb{A} \times \{\mathcal{G}_i \in 2^{\mathbb{S}} / |\mathcal{G}_i| = N\} \rightarrow \{\mathcal{G}'_i \in 2^{\mathbb{S}} / |\mathcal{G}'_i| = N\},$$

no cambiando el número de individuos, sino sustituyendo el mutado.

La *solución al problema* P para una corrida es la mejor solución de la última generación o de todo el proceso

$$s_{\text{opt}}^* = \arg \min_{s \in \mathcal{G}_i} f.$$

A los operadores g , S , c y m se los llama *operadores genéticos* y a f *función de aptitud* o de *fitness* y son, en general, dependientes del problema P y de la representación del individuo s en el espacio \mathbb{S} .

¹Denotamos $\mathcal{G}_i \in 2^{\mathbb{S}}$ en lugar de $\mathcal{G}_i \subseteq \mathbb{S}$ para indicar que \mathcal{G}_i puede tener elementos repetidos.

Parte III

Desarrollo de la solución

Capítulo 5

Identificación de parámetros de curva

Es de interés obtener, a partir de una curva de respuesta de frecuencia, aproximaciones para los parámetros H , D , T_g y R —que determinan su característica de regulación primaria—, el valor previo A y el tiempo de falta t_0 . Para esto es necesario invertir, en el dominio del tiempo, las ecuaciones asociadas a la respuesta del sistema básico con regulación primaria (figura 3.12) con la transferencia de la ecuación (3.29).

En el presente capítulo se desarrolla una teoría cuyos resultados son tales aproximaciones. Debido a que para los casos oscilatorio y no oscilatorio se obtienen ecuaciones diferentes, se trabajan ambos casos por separado.

Estos resultados serán usados en dos instancias del presente trabajo: *a)* el cómputo de la generación inicial de individuos (sección 7.5.1) y *b)* el cálculo de *fitness* de un individuo particular (sección 6.3.1).

5.1. Método

En primera instancia se definen puntos clave en la curva de entrada que puedan ser determinados de forma programable. A continuación, se expresan las ecuaciones para ambos casos de la curva y se define un conjunto de parámetros intermedios (H , p , q y r para el caso no oscilatorio; H , p , ω y ϕ para el oscilatorio), que tienen la ventaja de ser fácilmente representables en función de los puntos clave. Se desarrollan entonces las relaciones entre los puntos clave y estos parámetros.

Finalmente, se expresan los parámetros H , D , T_g y R en función de los parámetros intermedios según el caso.

En este capítulo y en otros donde se utilizan estos resultados, se hace referencia al parámetro ω de la curva de respuesta oscilatoria que, como se verá, refiere a la *frecuencia de oscilación de la frecuencia de la red*, y no a la

frecuencia de la red en sí, objeto de estudio de esta tesis. Si bien se usan en contextos separados, debe tenerse en cuenta para no confundirlos.

5.2. Puntos clave de la curva de entrada

La curva de respuesta de la frecuencia para un sistema con regulación primaria tiene un comportamiento que es discutido en la sección 3.5.3.

A fin de desarrollar un algoritmo para la identificación programable de parámetros, es necesario poder identificar puntos clave en la curva, donde sea posible medir cantidades expresables en términos de los parámetros de salida. En esta sección se definen tales puntos para ambos tipos de curva.

5.2.1. Caso no oscilatorio

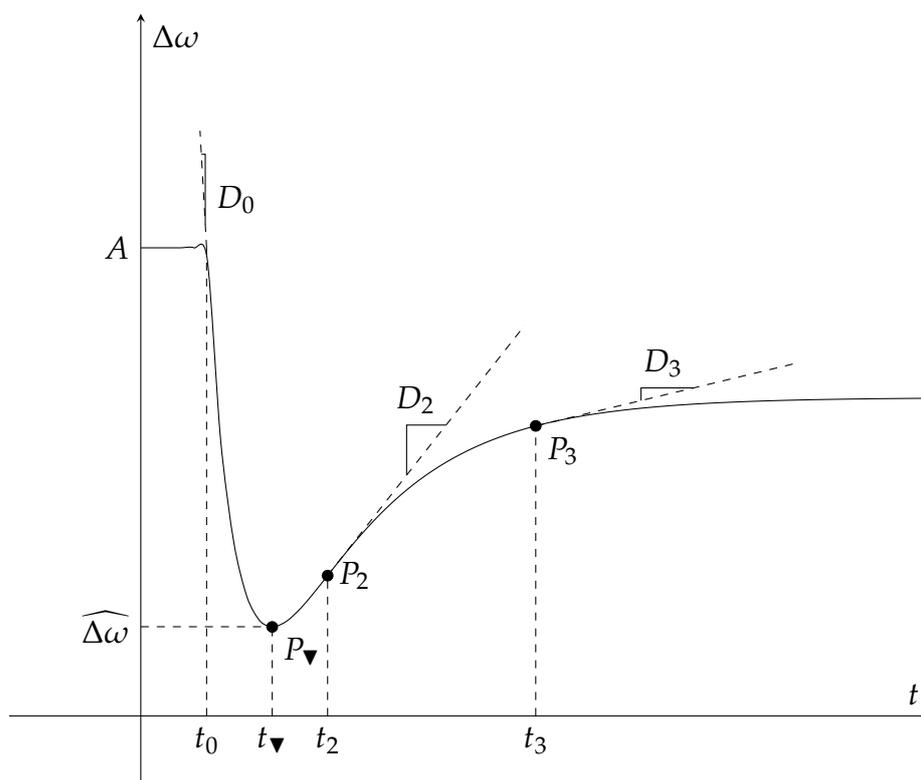


Figura 5.1: Puntos clave para el caso no oscilatorio

En la figura 5.1 se muestran los puntos clave necesarios para identificar los parámetros en el caso no oscilatorio.

5.2. Puntos clave de la curva de entrada

Tiempo de falta t_0 : El tiempo en el cual se comienza a observar la *respuesta inercial* (ver sección 3.5.3).

Este tiempo no es necesariamente aquél en el cual la falta ocurrió: en un SEP real algunos fenómenos físicos pueden introducir un retardo entre la ocurrencia de la falta y la respuesta inercial; sin embargo, para un tratamiento teórico se considerará que la falta ocurre en este momento, dado que es de interés ignorar los comportamientos previos a la respuesta inercial.

Para el desarrollo del algoritmo, el hecho de que el tiempo de falta sea no nulo simplemente introduce un retardo $t_0 \neq 0$ en la escala temporal; por lo que *excepto para tratar la detección de t_0 , se considerará en el desarrollo que $t_0 = 0$ para simplificar los cálculos, sin pérdida de generalidad*. A la hora de programar el algoritmo, debe sustituirse t por $t - t_0$ para considerar el retardo.

Frecuencia previa a la falta A : El valor previo de la frecuencia no necesariamente es la frecuencia nominal del SEP, por lo que el valor de $\Delta\omega$ al inicio puede no ser nulo.

El hecho de que sea no nulo introduce un término constante en las ecuaciones; por lo tanto, *excepto para tratar la detección de A , se considerará que $A = 0$* . Para la programación debe agregarse este término constante.

Derivada en el tiempo de falta D_0 : La derivada inicial está directamente relacionada con la inercia H (ecuación (3.8)), y es un parámetro relativamente simple de calcular numéricamente a partir de la curva de entrada.

Tiempo de subida al 90% $t_{90\%}$: Conociendo el valor base y el de nadir, el tiempo de subida al 90% permite aproximar la constante de tiempo de subida despreciando parcialmente efectos de orden superior que se dan en el tiempo de nadir.

Tiempo de nadir t_{∇} : Extremo de la respuesta; máximo o mínimo dependiendo del signo de la falta, que a su vez se determina como el signo de $\widehat{\Delta\omega} - A$.

Punto de inflexión P_2 y su tiempo asociado t_2 : Punto de inflexión inmediato al tiempo de nadir t_{∇} . Se puede determinar numéricamente buscando el cambio de signo de la derivada segunda.

Derivada en el punto de inflexión D_2 : Valor de la derivada primera en $t = t_2$.

Punto P_3 , con derivada D_3 en tiempo t_3 : Punto en el cual la derivada primera cumple $D_3 = e^{-1}D_2$; se detalla en la sección 5.7.2 por qué es de interés.

5.2.2. Caso oscilatorio

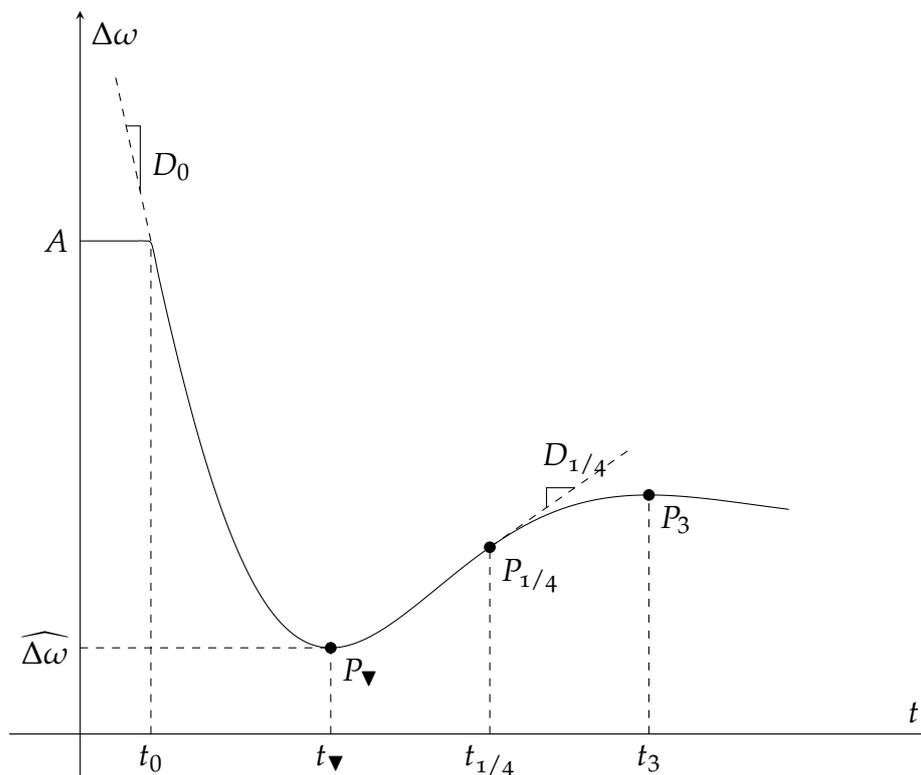


Figura 5.2: Puntos clave para el caso oscilatorio

En la figura 5.2 se muestran los puntos clave para el caso oscilatorio.

El tiempo de falta t_0 , la derivada en la falta D_0 , el valor inicial A y el tiempo de nadir t_∇ se definen igual que en el caso no oscilatorio.

Extremo posterior al nadir P_3 en tiempo t_3 : Es el extremo local posterior al nadir; máximo o mínimo dependiendo del signo de la falta. Es un valor fácil de determinar numéricamente que permite determinar la frecuencia de oscilación.

Tiempo de cuarto período $t_{1/4}$ y primera derivada $D_{1/4}$: Tiempo medio entre t_∇ y t_3 , marcando cuarto período en la oscilación. La derivada de este punto está relacionada con la razón de atenuación en la oscilación.

5.3. Transferencia

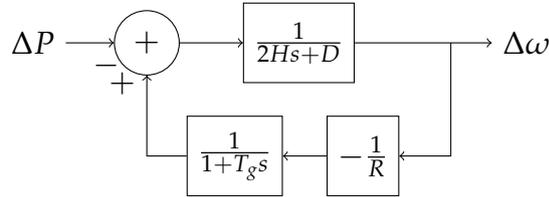


Figura 5.3: Diagrama de bloques de la regulación primaria

El diagrama de bloques para la variación de la frecuencia con un sistema de regulación primaria (figura 5.3) se estudia en la sección 3.5.2, resultando en una función de transferencia (ecuación (3.29)):

$$F(s) = -\frac{T_g s + 1}{(2Hs + D)(T_g s + 1) + R^{-1}} \quad (5.1)$$

Notar que es adimensional, siendo apropiada para cantidades ΔP y $\Delta \omega$ en (p.u.). Normalizando,

$$F(s) = -\frac{1}{2H} \frac{s + T_g^{-1}}{(s + \frac{D}{2H})(s + T_g^{-1}) + (2HT_g R)^{-1}} \quad (5.2)$$

con polos en¹

$$-\frac{1}{2} \left(\frac{D}{2H} + \frac{1}{T_g} \right) \pm \sqrt{\frac{1}{4} \left(\frac{D}{2H} - \frac{1}{T_g} \right)^2 - \frac{1}{2HT_g R}} \triangleq -q \pm \sqrt{p^2 - c} \quad (5.3)$$

definiendo p , q y c acordemente y tomando para p el signo mostrado en la ecuación.

5.4. Antitransformada (dominio del tiempo)

Se obtiene una expresión explícita de la respuesta escalón para ambos tipos de curva y se definen los parámetros intermedios que serán útiles para la ejecución del algoritmo de ajuste.

¹Soluciones para $(s + a)(s + b) + c = 0$:

$$s^2 + (a + b)s + ab + c = 0$$

$$s = \frac{-a - b \pm \sqrt{a^2 + 2ab + b^2 - 4ab - 4c}}{2} = \boxed{-\frac{a + b}{2} \pm \sqrt{\left(\frac{a - b}{2}\right)^2 - c}}$$

Capítulo 5. Identificación de parámetros de curva

5.4.1. Polos reales

Si los polos de $F(s)$ son reales ($c \leq p^2$), se obtiene una respuesta no oscilatoria. La antitransformada para $\frac{s+z}{(s+A)(s+B)}$ es

$$\frac{1}{B-A} \left((B-z)e^{-Bt} - (A-z)e^{-At} \right).$$

Para los parámetros con los que se trabaja, tomando $A = q - r$ y $B = q + r$ los opuestos de los polos y $z = T_g^{-1}$ el del cero (y sin considerar aún el factor $-\frac{1}{2H}$ de la transferencia $F(s)$), se obtiene

$$\begin{aligned} B - A &= q + \sqrt{p^2 - c} - q + \sqrt{p^2 - c} = 2\sqrt{p^2 - c} \triangleq 2r \\ A - z &= \frac{1}{2} \left(\frac{D}{2H} + \frac{1}{T_g} \right) - \frac{1}{T_g} - \sqrt{p^2 - c} = \frac{1}{2} \left(\frac{D}{2H} - \frac{1}{T_g} \right) - \sqrt{p^2 - c} = \\ &= p - r \\ B - z &= p + r \end{aligned}$$

definiendo acordemente r . La antitransformada de la transferencia, en función de los parámetros intermedios definidos, es:

$$f(t) = -\frac{1}{2H} \frac{1}{2r} \left((p+r)e^{-(q+r)t} - (p-r)e^{-(q-r)t} \right); \quad (5.4)$$

mientras que su integral definida entre 0 y t , que corresponde a la función en el tiempo de la respuesta a una entrada escalón unitario es:

$$\int_0^t f(t) dt = -\frac{1}{2H} \frac{1}{2r} \left(\frac{p+r}{q+r} (1 - e^{-(q+r)t}) - \frac{p-r}{q-r} (1 - e^{-(q-r)t}) \right). \quad (5.5)$$

5.4.2. Polos complejos

En caso de polos complejos ($c > p^2$), se obtiene una respuesta oscilatoria con amplitud exponencial decreciente. Se define la frecuencia de oscilación ω de forma que los polos se sitúen en

$$-q \pm j\omega \triangleq -q \pm j\sqrt{c - p^2}.$$

La antitransformada para $\frac{s+z}{(s+A)^2 + \omega^2}$ es

$$\sqrt{1 + \left(\frac{z-A}{\omega} \right)^2} e^{-At} \text{sen}(\omega t + \phi) \quad \text{donde } \phi = \arctan \frac{\omega}{z-A}.$$

5.5. Caracterización de los parámetros

Para los parámetros con que se trabaja, tomando $A = q$ el opuesto de la parte real de los polos, ω el valor absoluto de su parte imaginaria y $z = T_g^{-1}$ el opuesto del cero, se obtiene

$$z - A = \frac{1}{T_g} - \frac{1}{2} \left(\frac{1}{T_g} + \frac{D}{2H} \right) = \frac{1}{2} \left(\frac{1}{T_g} - \frac{D}{2H} \right) = -p$$

por lo que

$$\phi = \arctan \frac{\omega}{-p} \quad \text{y} \quad p = -\omega \cot \phi$$

y por tanto:

$$f(t) = -\frac{1}{2H} \sqrt{1 + \cot^2 \phi} e^{-qt} \operatorname{sen}(\omega t + \phi) = -\frac{1}{2H} \frac{\operatorname{sen}(\omega t + \phi)}{\operatorname{sen} \phi} e^{-qt}, \quad (5.6)$$

donde se ha usado la identidad $1 + \cot^2 \phi = \operatorname{sen}^{-2} \phi$; mientras que su integral definida entre 0 y t , correspondiente a la función en el tiempo de la respuesta a una entrada escalón unitario, es:

$$\int_0^t f(t) dt = -\frac{1}{2H} \frac{1}{q^2 + \omega^2} \left(\frac{q \operatorname{sen} \phi + \omega \cos \phi}{\operatorname{sen} \phi} - \frac{q \operatorname{sen}(\omega t + \phi) + \omega \cos(\omega t + \phi)}{\operatorname{sen} \phi} e^{-qt} \right). \quad (5.7)$$

5.5. Caracterización de los parámetros

Para esta sección se considerará a la curva de entrada como la respuesta a un *escalón unitario positivo*; en la sección 5.6 se detallan las transformaciones necesarias para generalizar la entrada a un escalón ΔP .

Téngase en cuenta que la curva de entrada, conocida numéricamente, es $f(t) \star Y(t) = \int_0^t f(t) dt$, siendo $f(t)$ sólo conocida a través de derivación numérica: los parámetros que la definen son las incógnitas del problema.

5.5.1. Obtención directa de parámetros

H : derivada en $t = 0^+$ de la respuesta escalón

Para un escalón *unitario* en la entrada, haciendo uso de las propiedades de la transformada de Laplace y del teorema del valor inicial,

$$\begin{aligned} \left. \frac{d}{dt} (f(t) \star Y(t)) \right|_{t=0^+} &= \lim_{s \rightarrow \infty} s \mathcal{L} \left\{ \frac{d}{dt} (f(t) \star Y(t)) \right\} = \lim_{s \rightarrow \infty} \frac{1}{s} s^2 F(s) = \\ &= \lim_{s \rightarrow \infty} -\frac{1}{2H} \frac{s(s + \cancel{T_g^{-1}})}{(s + \cancel{\frac{D}{2H}})(s + \cancel{T_g^{-1}}) + \cancel{(2HT_g R)^{-1}}} = -\frac{1}{2H} \end{aligned} \quad (5.8)$$

Capítulo 5. Identificación de parámetros de curva

confirmando para ambos casos, oscilatorio y no, a $2H$ como el inverso del opuesto de la derivada de la frecuencia (por unidad) en el momento de la falta:

$$H = -\frac{1}{2} \frac{1}{\left. \frac{d}{dt} (f(t) \star Y(t)) \right|_{t=0^+}}. \quad (5.9)$$

R : Valor final de la respuesta escalón

Nuevamente tomando un escalón *unitario* y haciendo uso del teorema del valor final,

$$\begin{aligned} \lim_{t \rightarrow \infty} f(t) \star Y(t) &= \lim_{s \rightarrow 0} \frac{1}{s} F(s) = \\ &= \lim_{s \rightarrow 0} -\frac{1}{2H} \frac{s + T_g^{-1}}{(s + T_g^{-1})(s + \frac{D}{2H}) + (2HT_g R)^{-1}} = \\ &= -\frac{1}{2HT_g} \frac{1}{\frac{D}{2HT_g} + \frac{1}{2HT_g R}} = -\frac{1}{D + R^{-1}}. \end{aligned} \quad (5.10)$$

Tomando ahora la aproximación $R^{-1} \gg D$,

$$\lim_{t \rightarrow \infty} f(t) \star Y(t) \approx -R \Rightarrow R \approx -\lim_{t \rightarrow \infty} f(t) \star Y(t). \quad (5.11)$$

Si bien es una expresión útil para R (cuando la aproximación es válida), no se usará en el desarrollo, dado que se obtendrán expresiones más exactas para R en función de los parámetros intermedios.

5.5.2. Obtención de D , T_g , y R a través de parámetros intermedios: caso no oscilatorio

A diferencia de H y R , D y T_g no tienen expresiones simples para obtener aproximaciones a través de la curva. Para el caso de R , además, si bien se obtuvo una expresión a partir del teorema del valor final y la aproximación $R^{-1} \gg D$ —razonable en la práctica—; para muchos casos la convergencia hacia un valor límite es lenta, lo que obligaría a usar series temporales muy largas para estimar R .

En cambio, sí se pueden obtener expresiones relativamente simples para los parámetros p , q y r anteriormente definidos para el caso real; desde los cuales pueden, a su vez, obtenerse los objetivos D , T_g y R . En las siguientes secciones se desarrollan métodos para la estimación de estos parámetros intermedios, pasando además por otra serie de parámetros intermedios $q + r$, $p - r$ y r .

5.5. Caracterización de los parámetros

Hipótesis: polos rápido y lento

La respuesta para el caso real es la suma de dos exponenciales con constantes de tiempo $q + r$ y $q - r$.

Considerando que se desea *aproximar* la salida, puede admitirse que el polo más rápido $q + r$ llegará a un valor cercano a su valor final en los primeros instantes de la respuesta (antes del nadir), mientras que el polo más lento gobernará la salida en los instantes posteriores al nadir, cuando la exponencial asociada al primero ya tiene un valor cercano a ser constante.

$q + r$: tiempo de subida del 90% (polo rápido)

Para una exponencial creciente con constante de tiempo τ , $1 - e^{-t/\tau}$, el tiempo de subida de 0% a 90% del valor final, $t_{90\%}$, cumple:

$$1 - e^{-t_{90\%}/\tau} = \frac{9}{10} \Rightarrow e^{-t_{90\%}/\tau} = \frac{1}{10} \Rightarrow \tau = \frac{t_{90\%}}{\log 10}.$$

Aceptando la hipótesis anterior, y aproximando $t_{90\%}$ como el tiempo de subida al 90% del valor de nadir —es decir, aproximando el valor final de la exponencial de polo rápido con el nadir—, se obtiene una aproximación para $q + r$, el inverso de la constante de tiempo de la exponencial con polo rápido:

$$q + r = \frac{1}{\tau_{\text{rápido}}} \approx \frac{\log 10}{t_{90\%}}.$$

$q - r$: razón de derivadas (polo lento)

La derivada de una exponencial creciente de constante de tiempo τ es

$$\frac{d}{dt}(1 - e^{-t/\tau}) = \tau^{-1}e^{-t/\tau}.$$

Si se conoce el valor de la derivada en dos puntos $t = t_1$ y $t = t_1 + \Delta t$

$$D_a = \tau^{-1}e^{-t_1/\tau}$$

$$D_b = \tau^{-1}e^{-(t_1+\Delta t)/\tau}$$

se puede expresar su razón ρ como

$$\rho = \frac{D_a}{D_b} = \frac{\tau^{-1}e^{-t_1/\tau}}{\tau^{-1}e^{-(t_1+\Delta t)/\tau}} = e^{\Delta t/\tau}. \quad (5.12)$$

Capítulo 5. Identificación de parámetros de curva

Por tanto

$$\frac{\Delta t}{\tau} = \log \rho \Rightarrow \boxed{\tau = \frac{\Delta t}{\log \rho}} \quad (5.13)$$

es una expresión para la constante de tiempo en función de la razón de derivadas y la diferencia de tiempo.

Desde el punto de inflexión en adelante, se puede considerar que domina la derivada del polo lento, por lo que tomando el punto de inflexión t_2 y un punto más a la derecha $t_3 > t_2$ como referencias, a partir de los valores de la derivada en esos puntos es posible obtener una aproximación para el inverso de la constante del polo lento $q - r$:

$$q - r \approx \frac{1}{\tau_{\text{lento}}} = \frac{\log \frac{\frac{d}{dt}(f(t) \star Y(t))|_{t=t_2}}{\frac{d}{dt}(f(t) \star Y(t))|_{t=t_3}}}{t_3 - t_2}. \quad (5.14)$$

A partir de esta aproximación y la obtenida para $q + r$ en la sección anterior, se obtienen expresiones aproximadas para q y r .

La distancia óptima entre t_2 y t_3 para minimizar el error en este cálculo es tal que $f(t_3) = e^{-1}f(t_2)$, por lo que la mejor elección para el punto t_3 es aquel cuya derivada toma ese valor. Esto se demuestra en la sección 5.7.2.

p : tiempo de nadir (derivada de la respuesta escalón)

Se busca aproximar el tiempo de nadir t_{\blacktriangledown} (extremo de la función en el tiempo),

$$\begin{aligned} \frac{d}{dt}(f(t) \star Y(t)) \Big|_{t=t_{\blacktriangledown}} &= \mathcal{L}^{-1} \left\{ \frac{1}{s} F(s) \right\} \Big|_{t=t_{\blacktriangledown}} = f(t_{\blacktriangledown}) = 0 \\ -\frac{1}{2H} \frac{1}{2r} \left((p+r)e^{-(q+r)t_{\blacktriangledown}} - (p-r)e^{-(q-r)t_{\blacktriangledown}} \right) &= 0 \\ \frac{p+r}{p-r} = \frac{e^{-(q-r)t_{\blacktriangledown}}}{e^{-(q+r)t_{\blacktriangledown}}} &= e^{2rt_{\blacktriangledown}}. \end{aligned} \quad (5.15)$$

Introduciendo ahora $\frac{p+r}{p-r} = \frac{p-r+2r}{p-r} = 1 + \frac{2r}{p-r}$,

$$1 + \frac{2r}{p-r} = e^{2rt_{\blacktriangledown}} \Rightarrow p-r = \frac{2r}{e^{2rt_{\blacktriangledown}} - 1} \Rightarrow \boxed{p = \frac{2r}{e^{2rt_{\blacktriangledown}} - 1} + r}$$

obteniéndose una para p a partir del tiempo de nadir y r ; con lo que, con una medida experimental de t_{\blacktriangledown} y la aproximación para r obtenida en la parte anterior, se logra una expresión aproximada para p .

5.5. Caracterización de los parámetros

Como corolario, se obtiene la expresión para el nadir en función de los parámetros

$$t_{\nabla} = \frac{1}{2r} \log \frac{p+r}{p-r}. \quad (5.16)$$

En la misma puede verse que, para que exista nadir en tiempo positivo, el argumento del logaritmo debe ser mayor a la unidad. Como $r = \sqrt{p^2 - c} < p$, para esto es necesario que $p > 0$, lo que implica $\frac{D}{2H} > \frac{1}{T_g}$; es decir $T_g > \frac{2H}{D}$ es condición necesaria para la existencia de nadir.

Tiempo de punto de inflexión

Si bien no es utilizada para la aproximación de parámetros, interesa obtener una ecuación para el punto de inflexión t_2 en función de los parámetros intermedios.

Para ello se iguala la segunda derivada de la respuesta escalón a 0:

$$\begin{aligned} \frac{d^2}{dt^2} (f(t) \star Y(t)) \Big|_{t=t_2} &= \frac{d}{dt} f(t_2) = 0 \\ -\frac{1}{2H} \frac{1}{2r} \left(-(p+r)(q+r)e^{-(q+r)t_2} + (p-r)(q-r)e^{-(q-r)t_2} \right) &= 0 \\ \frac{p+r}{p-r} \frac{q+r}{q-r} &= \frac{e^{-(q-r)t_2}}{e^{-(q+r)t_2}} = e^{2rt_2} \end{aligned}$$

desde donde se llega a

$$t_2 = \frac{1}{2r} \log \frac{p+r}{p-r} \frac{q+r}{q-r}. \quad (5.17)$$

Capítulo 5. Identificación de parámetros de curva

Despeje de parámetros finales

Los parámetros finales se obtienen de los intermedios mediante

$$\begin{aligned}\frac{D}{2H} &= q + p \\ \Rightarrow \boxed{D = 2H(q + p)}\end{aligned}\quad (5.18)$$

$$\begin{aligned}\frac{1}{T_g} &= q - p \\ \Rightarrow \boxed{T_g = \frac{1}{q - p}}\end{aligned}\quad (5.19)$$

$$\begin{aligned}r^2 = p^2 - c \Rightarrow c = p^2 - r^2 \Rightarrow \frac{1}{2HT_g R} = p^2 - r^2 \\ \Rightarrow \boxed{R = \frac{1}{2HT_g} \frac{1}{p^2 - r^2}}\end{aligned}\quad (5.20)$$

5.5.3. Obtención de D , T_g , y R a través de parámetros intermedios: caso oscilatorio

Para el caso oscilatorio también es posible obtener expresiones aproximadas para los parámetros intermedios q , ω y ϕ a partir de la respuesta escalón, obteniéndose p fácilmente con la expresión para ϕ en función de p y ω .

ω : distancia entre valle y pico

Siendo la frecuencia de pulsación de oscilación, ω se puede obtener para el caso oscilatorio con una medida de un pico y un valle, eligiéndose por ejemplo t_{\blacktriangledown} como valle; y t_3 , el máximo local siguiente al punto de inflexión t_2 , como pico. Siendo esta distancia un semiperíodo, se obtiene la expresión

$$\boxed{\omega = \frac{\pi}{t_3 - t_{\blacktriangledown}}}\quad (5.21)$$

5.5. Caracterización de los parámetros

ϕ : tiempo de nadir (derivada de la respuesta escalón)

Se busca una expresión para el tiempo de nadir t_{∇} en función de los parámetros ϕ y ω :

$$\begin{aligned} \left. \frac{d}{dt} (f(t) \star Y(t)) \right|_{t=t_{\nabla}} &= \mathcal{L}^{-1} \left\{ \frac{1}{s} sF(s) \right\} \Big|_{t=t_{\nabla}} = f(t_{\nabla}) = 0 \\ -\frac{1}{2H \cancel{\text{sen } \phi}} e^{-qt_{\nabla}} \text{sen}(\omega t_{\nabla} + \phi) &= 0 \end{aligned} \quad (5.22)$$

La función se anula con el seno. En particular se busca el cero para el cual el argumento es π , donde la derivada pasa de negativa a positiva (para un escalón positivo en la entrada):

$$\omega t_{\nabla} + \phi = \pi \Rightarrow \boxed{\phi = \pi - \omega t_{\nabla}}. \quad (5.23)$$

Usando la expresión para ω obtenida en la sección anterior, se obtiene una expresión para ϕ . Notar que ϕ es el desfase del nadir con respecto a la falta, por lo que toma valores $0 < \phi < \pi$.

Como corolario, el tiempo de nadir en función de los parámetros es

$$t_{\nabla} = \frac{\pi - \phi}{\omega}. \quad (5.24)$$

q : derivada en punto medio entre nadir y primer pico

Siendo t_3 —como definido anteriormente— el tiempo del primer pico posterior al nadir, para la derivada en el punto medio entre éste y el nadir se tiene que el seno es máximo en valor absoluto, por ser el punto donde el argumento es $\frac{3}{2}\pi$ (cuarto período después del nadir).

Recordando que $\phi = \pi - \omega t_{\nabla}$ y $\omega = \frac{\pi}{t_3 - t_{\nabla}}$,

$$\omega \frac{t_3 + t_{\nabla}}{2} + \phi = \omega \frac{t_3 + t_{\nabla}}{2} + \pi - \omega t_{\nabla} = \omega \frac{t_3 - t_{\nabla}}{2} + \pi = \frac{\pi}{\cancel{t_3 - t_{\nabla}}} \frac{\cancel{t_3 - t_{\nabla}}}{2} + \pi = \frac{3}{2}\pi. \quad (5.25)$$

Por tanto, el valor del seno es -1 , y para la derivada se obtiene

$$\left. \frac{d}{dt} (f(t) \star Y(t)) \right|_{t=\frac{t_3+t_{\nabla}}{2}} = -\frac{1}{2H} \frac{\text{sen } \frac{3}{2}\pi}{\text{sen } \phi} e^{-q \frac{t_3+t_{\nabla}}{2}} = \frac{1}{2H} \frac{1}{\text{sen } \phi} e^{-q \frac{t_3+t_{\nabla}}{2}}, \quad (5.26)$$

por lo que conocido el valor de la derivada $D_{1/4}$ en ese punto y los otros parámetros, puede despejarse q :

$$\boxed{q = -\frac{2}{t_3 + t_{\nabla}} \log(2HD_{1/4} \text{sen } \phi)}. \quad (5.27)$$

Capítulo 5. Identificación de parámetros de curva

Despeje de parámetros finales

Los parámetros finales se obtienen de los intermedios mediante

$$\begin{aligned} \frac{D}{2H} &= q + p = q - \omega \cot \phi \\ \Rightarrow \boxed{D = 2H(q - \omega \cot \phi)} \end{aligned} \quad (5.28)$$

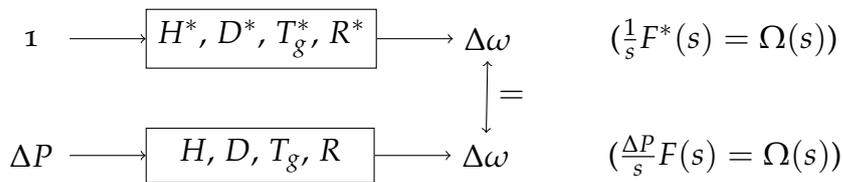
$$\begin{aligned} \frac{1}{T_g} &= q - p = q + \omega \cot \phi \\ \Rightarrow \boxed{T_g = \frac{1}{q + \omega \cot \phi}} \end{aligned} \quad (5.29)$$

$$\begin{aligned} \omega^2 = c - p^2 \Rightarrow c = p^2 + \omega^2 \Rightarrow \frac{1}{2HT_g R} &= \omega^2(1 + \cot^2 \phi) \\ \Rightarrow \boxed{R = \frac{1}{2HT_g} \frac{\text{sen}^2 \phi}{\omega^2}} \end{aligned} \quad (5.30)$$

5.6. Generalización a un escalón ΔP

Los cálculos en las secciones anteriores están hechos para una entrada unitaria. En la práctica, puede tenerse una curva que es la respuesta a un escalón dado ΔP no necesariamente unitario. El sistema estudiado es lineal, por lo que al aplicar una entrada $\Delta P \neq 1$ la salida estará amplificada por ΔP .

Si se usan las fórmulas de las secciones anteriores, se obtienen parámetros H^* , D^* , T_g^* y R^* que conforman un sistema que amplifica la salida por ΔP con respecto al real.



Se buscan entonces parámetros H , D , T_g y R tales que

$$1 \longrightarrow \boxed{H, D, T_g, R} \longrightarrow \frac{\Delta\omega}{\Delta P}$$

5.6. Generalización a un escalón ΔP

por lo que $F^*(s) = \Delta P \times F(s)$:

$$\begin{aligned} & -\frac{1}{2H^*} \frac{s + T_g^{*-1}}{\left(s + \frac{D^*}{2H^*}\right)(s + T_g^{*-1}) + (2H^*T_g^{*-1}R^*)^{-1}} = \\ & = -\Delta P \frac{1}{2H} \frac{s + T_g^{-1}}{\left(s + \frac{D}{2H}\right)(s + T_g^{-1}) + (2HT_g^{-1}R)^{-1}}. \end{aligned}$$

Igualando el factor constante

$$\frac{1}{2H^*} = \frac{\Delta P}{2H} \Rightarrow \boxed{H^* = \frac{H}{\Delta P}}$$

e igualando el numerador,

$$T_g^{*-1} = T_g^{-1} \Rightarrow \boxed{T_g^* = T_g}.$$

Sustituyendo ahora H^* y T_g^* en el denominador del lado izquierdo,

$$\begin{aligned} & \left(s + \frac{\Delta P D^*}{2H}\right)(s + T_g^{-1}) + \left(2\frac{H}{\Delta P}T_gR^*\right)^{-1} = \\ & = \left(s + \frac{D}{2H}\right)(s + T_g^{-1}) + (2HT_gR)^{-1} \end{aligned}$$

de donde se concluye

$$\boxed{D^* = \frac{D}{\Delta P}} \quad \text{y} \quad \boxed{R^* = \Delta P \times R}.$$

Conclusión: ponderamientos

Si un escalón ΔP produce una curva de salida, desde esa curva las ecuaciones anteriores producen parámetros H^* , D^* , T_g^* y R^* tales que H^* y D^* están ponderados por $\frac{1}{\Delta P}$, R^* está ponderado por ΔP , y T_g^* está sin ponderar.

Para obtener los parámetros reales de la curva,

$$\begin{cases} H = \Delta P \times H^* \\ D = \Delta P \times D^* \\ T_g = T_g^* \\ R = \frac{1}{\Delta P} \times R^* \end{cases} \quad (5.31)$$

Puede corroborarse que tanto los parámetros intermedios p , q y r para el caso no oscilatorio como ϕ , q y ω para el oscilatorio no se encuentran ponderados por ΔP .

5.7. Consideraciones prácticas

5.7.1. Identificación de t_0 y H en redes con perturbaciones iniciales

La detección del *tiempo de falta* t_0 —definido como el tiempo a partir del cual se observa la respuesta inercial— y de la inercia total H , derivada de la respuesta en tiempo t_0 , puede no ser directa a partir de la curva, debido a que en los instantes iniciales de la falta la «frecuencia» no está bien definida, sino que depende de qué magnitud se esté considerando para medirla (sección 3.3); por lo que aparecen irregularidades en la curva en los momentos iniciales. Estas irregularidades deben ignorarse; sin embargo, debe postularse un método para obtener una aproximación de H y t_0 .

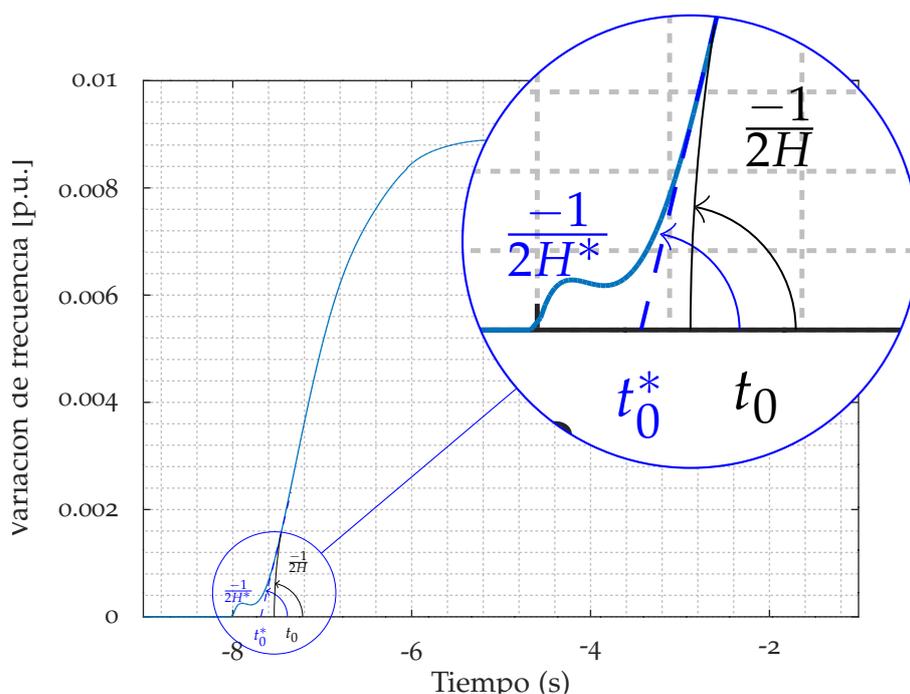


Figura 5.4: Red con perturbaciones iniciales. En verde marino, la curva original; en azul punteado, la estimación a partir del punto de máxima derivada; en gris t_0 y H reales.

El método utilizado se muestra en la figura 5.4. A partir de la curva original, no pueden inferirse H o t_0 ; sin embargo, se obtiene una buena aproximación para ambos buscando el punto de máxima derivada en la respuesta inercial, y proyectando ese valor hasta el valor previo A . En el corte, se toma el punto t_0^* , mientras que para H^* se toma el valor correspondiente al de la derivada máxima (en azul punteado), confiando

que ésta estará cerca del valor de derivada «real» (en gris en la figura). Finalmente, el ajuste de mínimos cuadrados halla H y t_0 fácilmente a partir de H^* y t_0^* .

5.7.2. Intervalo en la detección de $q - r$ para el caso no oscilatorio

Si bien en las secciones anteriores se delinea un método para obtener una aproximación para el polo lento a partir de dos medidas de la derivada, queda sin responder la pregunta: si se deben elegir dos puntos t_a y t_b asociados a dos medidas de derivada D_a y D_b para aproximar la constante de tiempo τ , dada una elección t_a , ¿con qué elección $t_b > t_a$ se obtienen los mejores resultados?

Es deseable minimizar la incertidumbre $\delta\tau$ introducida por las medidas, sabiendo que depende de la medida de D_b (asumiendo dada y fija la incertidumbre introducida por la medida D_a).

Se busca entonces expresar $\delta\tau$ en función de las variables del problema. Como se dedujo en la ecuación (5.13),

$$\tau = \frac{t_b - t_a}{\log \rho}, \text{ y}$$

$$\delta\tau = \left| \frac{\partial\tau}{\partial\rho} \right| \delta\rho.$$

Ahora $\delta\rho = \left| \frac{\partial\rho}{\partial D_b} \right| \delta D_b$ al considerar D_a dado, por tanto

$$\delta\tau = \left| \frac{\partial\tau}{\partial\rho} \right| \left| \frac{\partial\rho}{\partial D_b} \right| \delta D_b.$$

Como $\left| \frac{\partial\tau}{\partial\rho} \right| = \frac{t_b - t_a}{\rho \log^2 \rho}$ y $\left| \frac{\partial\rho}{\partial D_b} \right| = \frac{\rho}{D_b}$,

$$\delta\tau = \frac{t_b - t_a}{\log^2 \rho} \frac{\delta D_b}{D_b}.$$

La derivada D_b es calculada a partir de dos medidas consecutivas de la salida f_b y f_{b+1} :

$$D_b = \frac{f_{b+1} - f_b}{t_{b+1} - t_b} = \frac{(1 - e^{-t_{b+1}/\tau}) - (1 - e^{-t_b/\tau})}{t_{b+1} - t_b} = \frac{e^{-t_b/\tau} - e^{-(t_b+dt)/\tau}}{dt} =$$

$$= \frac{e^{-t_b/\tau}(1 - e^{-dt/\tau})}{dt} \triangleq e^{-t_b/\tau} K$$

Capítulo 5. Identificación de parámetros de curva

donde $dt = t_{b+1} - t_b$ es una cantidad fija dada por el muestreo; y definiendo K constante acordemente. Para ρ se obtiene

$$\rho = \frac{D_a}{D_b} = \frac{\tau^{-1}e^{-t_a/\tau}}{\tau^{-1}e^{-t_b/\tau}} = e^{(t_b-t_a)/\tau} \Rightarrow \log^2 \rho = \frac{(t_b - t_a)^2}{\tau^2}$$

con lo que se obtiene una expresión para $\delta\tau$ en función de t_b :

$$\delta\tau(t_b) = \frac{\tau(t_b - t_a)}{(t_b - t_a)^2} e^{t_b/\tau} K^{-1} \delta D_b = \frac{\tau}{t_b - t_a} e^{t_b/\tau} K^{-1} \delta D_b.$$

Se puede considerar que δD_b no depende de t_b^2 . Ahora bien, tanto $\lim_{t_b \rightarrow t_a} \delta\tau = \infty$ como $\lim_{t_b \rightarrow \infty} \delta\tau = \infty$, por lo que tomar valores demasiado próximos o lejanos a t_a maximiza la incertidumbre. Por tanto, se busca el valor intermedio óptimo minimizando:

$$\frac{d}{dt_b} \delta\tau = \frac{\tau K^{-1}}{(t_b - t_a)^2} \left(\frac{1}{\tau} e^{t_b/\tau} (t_b - t_a) - e^{t_b/\tau} \right) = 0 \Rightarrow \boxed{t_b = t_a + \tau.}$$

Como no se conoce τ de antemano, puede verse que la razón de derivadas es

$$\frac{D_a}{D_{a+\tau}} = \frac{\cancel{\tau^{-1}} e^{-t_a/\tau}}{\cancel{\tau^{-1}} e^{-(t_a+\tau)/\tau}} = \frac{e^{-t_a/\tau}}{e^{-t_a/\tau} e^{-\tau/\tau}} = e$$

y, dado D_a , buscarse el valor t_b para el cual $D_b \approx \frac{D_a}{2.718}$, siendo éste el valor que minimiza la incertidumbre para τ .

²El error δD_b depende del error de dos medidas consecutivas de la salida. Usando $D_b = \frac{f_{b+1} - f_b}{t_{b+1} - t_b} = \frac{f_{b+1} - f_b}{dt}$ como se definió,

$$\delta D_b = \sqrt{\left| \frac{\partial D_b}{\partial f_b} \right|^2 \delta f_b^2 + \left| \frac{\partial D_b}{\partial f_{b+1}} \right|^2 \delta f_{b+1}^2} = \sqrt{\frac{1}{dt^2} \delta f_b^2 + \frac{1}{dt^2} \delta f_{b+1}^2}$$

Asumiendo la misma incertidumbre para ambas medidas, $\delta f_b = \delta f_{b+1} = \delta f$ y

$$\delta D_b = \frac{\sqrt{2}}{dt} \delta f \quad \text{constante.}$$

Capítulo 6

Cálculo de *fitness*

Es necesario definir la función de *fitness* de un individuo del algoritmo evolutivo.

Informalmente, un individuo es más *apto* cuando el modelo uninodal que representa se comporta de forma más parecida a la red que modela. Es necesario traducir estos conceptos a una función que asocie a cada genoma un número real positivo, como descrita en la sección 4.4. Para ello, son necesarios algunos conceptos que serán desarrollados en este capítulo:

- la *curva de salida* asociada a un individuo;
- representada por una *serie temporal*; y
- una definición para la *distancia entre curvas*.

Luego de definidos, se puede establecer formalmente una función de *fitness* o aptitud para el individuo.

6.1. Evaluación de un individuo

Un *individuo* del algoritmo evolutivo estará representado por un *genoma*, que será una lista de los parámetros necesarios para instruir al programa PSS[®] e en cómo ejecutar una simulación dinámica sobre ese modelo. A fin de comparar el comportamiento del individuo con el de la red de referencia, es necesario ejecutar una simulación en ambas redes —la de referencia y la uninodal representada por el individuo— y comparar la respuesta de la frecuencia ante una falta en ambas, cuantificándola de manera que se pueda comparar frente a otras simulaciones.

Con el fin de ahorrar en simulaciones en la red original, y aprovechando la linealidad del modelo uninodal de la red, se ejecutarán todas las simulaciones dinámicas con una misma falta. Esto hace posible tomar

la respuesta de la frecuencia en una única falta en la red original como referencia y compararla contra una simulación de esa misma falta en el modelo uninodal.

Tales simulaciones dinámicas producirán una *curva de salida*, una representación del valor de la frecuencia (en los hechos, una medida de la velocidad del centro de inercias, como se describe en la sección 3.3.1) en forma de valores muestreados cada un período predeterminado. Esta curva debe analizarse de forma programada para identificar sus características y compararlas contra la *curva de referencia* —la curva de salida para la misma falta en la red original—. De esta comparación debe definirse una medida de cercanía, y tal medida será en definitiva el *fitness* de este individuo.

6.2. Curva de salida

6.2.1. Serie temporal

A efectos de describir la curva de salida, se define una *serie temporal* como una pareja ordenada $F = (t_i, f_i)$ de sucesiones reales, llamadas sucesiones *temporal* y *funcional* respectivamente, cuya sucesión temporal t_i está ordenada: $i < j \implies t_i < t_j$.

Se usa la notación $t[i]$ (o $f[i]$) para denotar el i -ésimo elemento de la sucesión t (o f), y no se descarta abusar de la notación refiriéndose a la serie temporal $F = (t, f)$ mediante el nombre de su sucesión funcional f , de no haber lugar a confusión.

6.2.2. Respuesta de la frecuencia como serie temporal

El muestreo de cualquier función real en el tiempo puede verse y tratarse como una serie temporal, y la respuesta de la frecuencia tal como es devuelta por el PSS[®]e es un ejemplo de ello (figura 6.1).

A la serie temporal que representa la velocidad del centro de inercia para la simulación dinámica de determinada falta en un individuo, se le denomina *curva de salida del individuo frente a esa falta*.

Al ser una serie temporal, se puede ya tratar matemáticamente el comportamiento de un individuo del algoritmo evolutivo frente a la falta de referencia estudiando su serie temporal. Se reduce entonces el problema de hallar el *fitness* de un individuo al de hallar una medida de la cercanía entre dos series temporales: las curvas de respuesta de referencia de la red original y la del individuo.

6.3. *Fitness* de una curva de salida

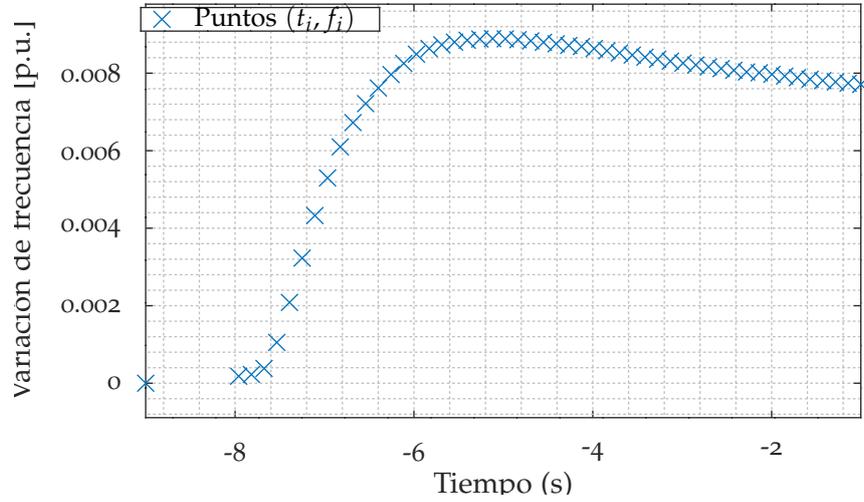


Figura 6.1: La respuesta de la frecuencia como serie temporal

6.3. *Fitness* de una curva de salida

Dadas dos series temporales, $F_{\text{ref}} = (t_{\text{ref}}, f_{\text{ref}})$ de referencia y $F_1 = (t_1, f_1)$ asociada a un individuo, para determinar el *fitness* de F_1 interesa encontrar una forma de medir su aproximación a la curva objetivo F_{ref} .

6.3.1. Sincronización temporal

Antes de poder definir y medir una distancia entre las curvas, debe considerarse que, a priori, las series F_{ref} y F_1 tienen cada una sus propios tiempos de inicio y fin y frecuencia de muestreo, lo que hace imposible una comparación numérica directa.

Adicionalmente, aun si estos tiempos fueran iguales inicialmente o arreglados para coincidir, debe considerarse que los *tiempos de falta* en cada una pueden también diferir: no sólo dependen de la ejecución de cada simulación dinámica particular, sino que otros efectos en la red pueden introducir retardos que repercuten en los mismos (ver sección 5.7.1). Para que una comparación numérica tenga sentido, las faltas deben ocurrir al mismo tiempo dentro del sistema de tiempo de los objetos que se están comparando; es decir, la frecuencia en un tiempo T luego del momento de la falta en $F_{\text{ref}} \rightarrow t_{\text{ref}}^{(0)}$ debe compararse con la frecuencia en un tiempo T luego del momento de la falta en $F_1 \rightarrow t_1^{(0)}$, independientemente del valor específico de ambos tiempos de falta.

Se concluye entonces que los pasos que deben ejecutarse para poder

comparar las curvas son, en orden:

- resincronización de los momentos de falta en cada serie,
- recorte de ambas series a los tiempos más ajustados posibles; es decir, limitarlas a los intervalos temporales antes y después de la falta menores entre ambas, y
- remuestreo de una de las series al vector temporal de la otra, para que los valores puedan compararse directamente.

Resincronización del tiempo de falta

Para ello se identifican, mediante el algoritmo de ajuste desarrollado en el capítulo 5 para la respuesta escalón en sistemas con estatismo, los tiempos de falta en ambas curvas¹, $t_{\text{ref}}^{(0)}$ y $t_1^{(0)}$; y se traslada en el tiempo una de las dos. Si bien es indistinto, se hace con t_1 :

$$t_2[i] = t_1[i] + t_{\text{ref}}^{(0)} - t_1^{(0)},$$

produciendo una serie $F_2 = (t_2, f_2)$ cuyo tiempo de falta $t_2^{(0)}$ coincide con el de F_{ref} .

Recorte

Se identifican los tiempos extremos

$$(t^{(\text{mín})}, t^{(\text{máx})}) = (\text{máx}\{\text{mín } t_{\text{ref}}, \text{mín } t_2\}, \text{mín}\{\text{máx } t_{\text{ref}}, \text{máx } t_2\})$$

y se recortan ambas series:

$$\begin{aligned} (t'_{\text{ref}}, f'_{\text{ref}}) &= (\{(t_{\text{ref}}[i], f_{\text{ref}}[i]) \quad / t^{(\text{mín})} \leq t_{\text{ref}}[i] \leq t^{(\text{máx})}\}) \\ (t_3, f_3) &= (\{(t_2[i], f_2[i]) \quad / t^{(\text{mín})} \leq t_2[i] \leq t^{(\text{máx})}\}) \end{aligned}$$

produciendo dos series F'_{ref} y F_3 que cubren intervalos de tiempo equivalentes (con la posible excepción de los puntos extremos) y cuyos tiempos de falta coinciden.

¹Debe notarse que el tiempo de falta de una serie temporal no necesariamente pertenece al dominio temporal de la misma; es decir, no tiene por qué haber una muestra tomada en el tiempo de falta. El mismo es determinado por el algoritmo de ajuste independientemente del tiempo de las muestras, a partir de la curva subyacente.

Remuestreo

La serie con más muestras en el intervalo se remuestrea a los tiempos dados por la otra serie temporal².

Por ejemplo, si F_3 se remuestrea, si j es tal que

$$t_3[j] \leq t'_{\text{ref}}[i] < t_3[j+1],$$

se interpola linealmente el valor para ese momento

$$f_4[i] = f_3[j] + \frac{t'_{\text{ref}}[i] - t_3[j]}{t_3[j+1] - t_3[j]} (f_3[j+1] - f_3[j])$$

produciendo así una serie $F_4 = (t_4, f_4) = (t_{\text{ref}}, f_4)$.

Al cabo de estas transformaciones, F'_{ref} y F_4 cumplen que

- la falta ocurre en el mismo tiempo,
- comparten límites temporales, y
- comparten tiempos de muestreo.

Es posible entonces comparar F'_{ref} y F_4 valor a valor.

6.3.2. Distancia entre curvas como *fitness* del individuo

Recordar de 4.4 que la función de *fitness* debe producir un número real no negativo a partir de un individuo.

Dado que el objetivo es producir un individuo cuya *curva de salida* para determinada falta sea lo más parecida posible (idealmente idéntica) a la producida por la red que el modelo uninodal modela, tiene sentido definir la función de *fitness* como una medida del parecido de estas curvas; de forma más abstracta, su *distancia* en el espacio de funciones.

Resta definir una función de distancia entre curvas, llamadas F_1 y F_2 , que se pueda usar como *fitness*.

Distancia relativa entre campos escalares

Una medida natural es usar una norma: integrar su diferencia, o alguna de las potencias de la misma; por ejemplo, usando la norma 2, $\|f_1(t) - f_2(t)\|_2 = \sqrt{\frac{1}{\int dt} \int (f_1(t) - f_2(t))^2 dt}$ o en su versión discreta $\sqrt{\frac{1}{\sum \Delta t_i} \sum (f_1 - f_2)^2 \Delta t_i}$.

²Esta elección se da para aprovechar la mayor cantidad de información disponible a la hora de interpolar.

Una desventaja de este método es que la distancia queda así expresada en las unidades de F_1 y F_2 . Esto hace que a la hora de definir un valor de corte necesariamente deba definirse en base a estas unidades en términos absolutos, cuando puede ser deseable definir una distancia *relativa* como punto de quiebre.

Una distancia relacionada a las normas es la *distancia relativa entre dos campos escalares* Δ_D [16], definida para dos campos escalares no negativos \vec{f}_1 y \vec{f}_2 definidos en un dominio V como

$$\begin{cases} \frac{\int_V |\vec{f}_1 - \vec{f}_2| dV}{\int_V |\vec{f}_1 + \vec{f}_2| dV} & \text{si } \vec{f}_1 \text{ ó } \vec{f}_2 \neq \vec{0}, \\ 0 & \text{si } \vec{f}_1 \equiv \vec{f}_2 \equiv \vec{0}; \end{cases}$$

llevándola al dominio discreto y unidimensional de las series temporales,

$$\begin{cases} \frac{\sum_i |f_1[i] - f_2[i]|}{\sum_i |f_1[i] + f_2[i]|} & \text{si } F_1 \text{ ó } F_2 \neq \vec{0}, \\ 0 & \text{si } F_1 \equiv F_2 \equiv \vec{0}. \end{cases}$$

En el trabajo citado se demuestra que esta distancia define una métrica en el espacio de los campos escalares y que adquiere valores entre 0 y 1, valiendo 0 para dos campos idénticos y 1 cuando uno y sólo uno de los campos es nulo. Es por tanto utilizable como medida de aproximación entre curvas, y por tanto, como *fitness*, considerando que debe minimizarse como función objetivo.

Esta distancia está definida correctamente para campos no negativos. Si bien la frecuencia de un SEP es no negativa, las series con las que se trabaja representan Δf —su desviación en p.u. con respecto a una frecuencia de referencia f_0 — que no es no negativa.

El trabajo referido no recomienda soluciones para esto; algunas transformaciones posibles son:

- Transformación exponencial: la diferencia entre $\{e^{f_1[i]}\}$ y $\{e^{f_2[i]}\}$. Se experimenta cierta pérdida de precisión: la distancia jamás será próxima a 1, puesto que para ello sería necesario que una de las funciones fuese [idénticamente] negativa infinita.
- Traslación del plano de dominio al plano positivo. Siendo la frecuencia inherentemente no negativa, las series trasladadas a la frecuencia absoluta sumándoles la frecuencia de referencia $f_0 = 1$ (trabajando en p.u.) $\{1 + f_1[i]\}$ y $\{1 + f_2[i]\}$ cumplirán la condición de no negatividad:

$$\frac{\sum_i |f_1[i] + 1 - f_2[i] - 1|}{\sum_i |f_1[i] + 1 + f_2[i] + 1|} = \frac{\sum_i |f_1[i] - f_2[i]|}{\sum_i |f_1[i] + f_2[i] + 2|}.$$

Sin embargo, la cota inferior de la frecuencia será en general mucho mayor, por lo que sumar 2 al denominador resta mucha precisión innecesariamente.

Se opta por tomar la cota inferior absoluta de F_1 y F_2 —en el caso en que sea ésta negativa, caso contrario no se ajusta—:

$$M = \text{mín}\{\text{mín } f_1[i], \text{mín } f_2[i], 0\},$$

y comparar $\{f_1[i] - M\}$ con $\{f_2[i] - M\}$. Así, la distancia será 1 cuando una y sólo una de las series sea idénticamente M , logrando precisión máxima:

$$\Delta_D = \begin{cases} \frac{\sum_i |f_1[i] - f_2[i]|}{\sum_i |f_1[i] + f_2[i] - 2M|} & \text{si } F_1 \text{ ó } F_2 \neq M, \\ 0 & \text{si } F_1 \equiv F_2 \equiv M. \end{cases}$$

Cálculo de la cota inferior

Se agrega una consideración práctica para el algoritmo de cálculo de la cota inferior.

Para calcular la distancia se pueden recorrer a un mismo tiempo ambas series y acumular su suma y diferencia, obteniéndose su cociente al final. Sin embargo, no es necesaria una recorrida extra para calcular M . Se puede ajustar durante la recorrida de ambas series con el siguiente algoritmo:

1. Se comienza presumiendo que $M_0 = 0$.
2. Para cada índice i comenzando desde 1,
 - Se suma $f_1[i] - f_2[i]$ al acumulador del numerador y $f_1[i] + f_2[i] - 2M_{i-1}$ al del denominador.
 - Si $f_1 < M_{i-1}$, hágase $M_i = f_1$; de lo contrario, $M_i = M_{i-1}$. A continuación, si $f_2 < M_i$, hágase $M_i = f_2$.
 - Si $M_i < M_{i-1}$, el acumulador del denominador dista del valor que debería haberse computado en $2(M_i - M_{i-1}) \times i$. Réstesele este valor, notando que esta operación puede hacerse aun cuando haya igualdad.
3. Hállese el cociente entre ambos acumuladores y tómesese su valor absoluto, resultando en una distancia relativa entre campos escalares válida.

Consecuencias del uso de cota inferior de dos curvas

Si bien comparar trasladando según la cota inferior de ambas curvas es una solución útil desde el punto de vista práctico, debe considerarse que esto introduce una posible asimetría entre comparaciones de curvas: al comparar F_1 y F_2 se puede estar usando una diferente traslación que al comparar F_2 y F_3 , y esto redundaría en perder las propiedades de la distancia.

Para la presente aplicación, sin embargo, debe considerarse que las curvas siempre son comparadas con una curva de referencia F_{ref} y no habrá comparaciones entre curvas que no incluyan la curva de referencia. Esto da cierta libertad para tomar esta licencia, dado que las comparaciones no serán demasiado asimétricas, excepto en el caso de que una curva tenga una cota inferior mucho menor a F_{ref} : en ese caso M será muy negativa y redundará en un *fitness* alto, por lo que la curva será descartada.

Dos curvas cercanas a F_{ref} por encima y por debajo pueden compararse de forma ligeramente asimétrica; sin embargo, la diferencia no será lo suficientemente notoria como para que no se justifique ganar esta precisión extra.

6.4. Conclusión

El *fitness* de un individuo representando a un modelo uninodal queda definido, a partir de una falta dada, por el siguiente algoritmo:

1. Se define la serie temporal correspondiente a la curva de salida de la red original ante esa falta como la *curva de referencia*.
2. Se obtiene la serie temporal correspondiente a la curva de salida del modelo uninodal representado por el individuo ante la misma falta.
3. El *fitness* es la *distancia entre ambas series temporales* tal como definida en las secciones anteriores.

Capítulo 7

Arquitectura de la solución

La solución propuesta (o *herramienta*) consta de dos programas que interactúan entre sí: el programa que *controla la ejecución del algoritmo evolutivo* —que además hace las veces de interfaz con el usuario (*front-end*)—, y el programa que *ejecuta las pruebas en el PSS^{®e}*, que es invocado por el primero (*back-end*).

En este capítulo se presenta el flujo de datos, mostrando los principales componentes y puntos de interés, y se da cuenta de las decisiones generales de arquitectura.

7.1. Flujo de datos

En la figura 7.1 se muestra un esquema general del flujo de datos en la herramienta, destacándose los puntos principales.

① Red original: se cuenta con un modelo de la red del cual se desea obtener el equivalente uninodal.

② Se introduce un evento ΔP en la red, simulado con PSS^{®e}, lo cual produce una curva de respuesta (③). Esta curva no sólo se usará para generar la población inicial, sino también para calcular el *fitness* de los individuos durante la ejecución del algoritmo.

④ Con las técnicas desarrolladas en el capítulo 5 se realiza el ajuste de la curva de referencia y se obtienen los parámetros $H^{(0)}$, $D^{(0)}$, $T_g^{(0)}$ y $R^{(0)}$ equivalentes de la red original (⑤). Este paso es ejecutado por el programa, que acepta la curva (y el escalón ΔP original) como única entrada.

⑥ Para la generación inicial, se generan individuos al azar añadiendo un ruido gaussiano a los parámetros de la curva de referencia. La justificación de este modelo de generar la población inicial se funda en el desarrollo del capítulo 3 y se justifica en la sección 10.3.

Capítulo 7. Arquitectura de la solución

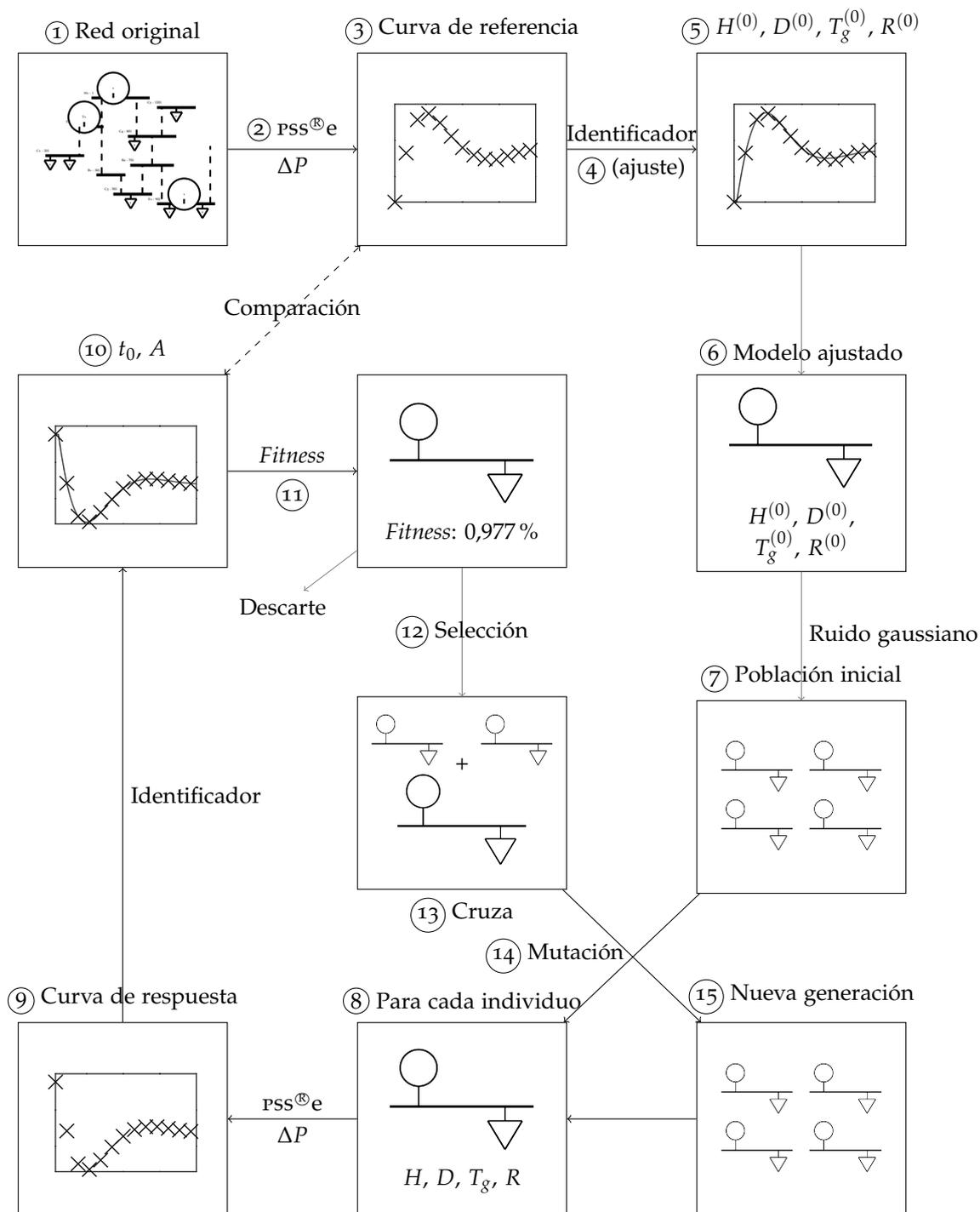


Figura 7.1: Esquema general del flujo de datos

⑦ El ciclo del algoritmo evolutivo se repite para cada generación. Se parte de un conjunto de individuos (población); cada individuo (⑧) es *evaluado*: se ejecuta en el PSS[®]e una simulación dinámica del modelo uninodal que representa, lo cual produce una curva de salida para el individuo (⑨) que se procede a identificar para extraer el tiempo de falta (⑩) y, de forma sincronizada, comparar con la curva de referencia (⑪). La distancia entre estas curvas representa el *fitness* de este individuo.

⑫ El individuo se somete a una *selección* en base a su *fitness*, y puede descartarse o seleccionarse para la cruce. Finalmente, los individuos seleccionados se cruzan (⑬) y mutan (⑭), produciendo los individuos de la siguiente generación (⑮).

El algoritmo se interrumpe luego de un número prefijado de generaciones, y el individuo con *mejor fitness histórico* es seleccionado como el *modelo uninodal equivalente dinámico* de la red original.

Se garantiza que de entre los individuos evaluados, es éste el que más se acerca (en el sentido definido en el capítulo 6) al comportamiento dinámico de la red original *para la falta* ΔP .

Los pasos ①, ② y ③ son ejecutados por el operador de forma externa; no forman parte del programa. Los pasos ④, ⑤, ⑥ y ⑦ son ejecutados una vez al inicio del mismo.

El ciclo del algoritmo (pasos ⑧ en adelante) es administrado por la biblioteca ECJ que «guía» a la generación a un nivel macro, y además ejecuta los pasos inherentes a este proceso (⑧ y ⑮); mientras que algunos pasos nombrados son ejecutados o por el programa controlador de PSS[®]e (⑨, capítulo 8) o por una clase en el programa controlador de ECJ (⑩ y ⑪, ⑫, ⑬ y ⑭, capítulo 9).

7.2. Control de PSS[®]e

La evaluación del individuo requiere la ejecución del modelo uninodal que representa en el PSS[®]e (paso ⑨ en la figura 7.1), que el algoritmo evolutivo debe solicitar en forma programada. PSS[®]e provee tres interfaces (API) posibles para ello: una línea de comandos con lenguaje propio, una API para el lenguaje FORTRAN, y *psspy*, para el lenguaje *Python*.

El uso de la API de *Python* fue una elección natural, dado que es un lenguaje de acceso universal y bien documentado, dando una mayor flexibilidad para trabajos futuros. Esto también otorga cierta flexibilidad para ejecutar cálculos *in situ*, permitiendo usar las bibliotecas propias de *Pyt-*

hon: como se verá, en particular, para la comunicación con el programa controlador del algoritmo evolutivo y el intercambio de datos.

A priori no sería necesaria una separación en dos programas —pudiéndose correr el algoritmo evolutivo en el mismo programa de control: [9] toma este camino de forma exitosa—; sin embargo, existen algunas ventajas en separarlos.

- Si se fuere por una solución de un solo programa, se estaría atado a la elección de *Python 2.7* como entorno de programación, bajo la plataforma *Windows*. Si bien existen motores de AA.EE. para este entorno, es ineficiente para la computación, carece de multihilo real, tiene un intérprete lento, etc. El separar el «cerebro» del algoritmo a otro programa libera de la atadura a esta plataforma y lenguaje.
- Una gran ventaja en el uso de AA.EE. es la paralelización. Si los programas usan instancias de PSS®e separadas, es posible evaluar individuos en paralelo; esto es posible si, nuevamente, el «cerebro» está separado.
- Permite correr los programas en máquinas separadas.

Se opta entonces por dejar en el programa controlador de PSS®e sólo lo estrictamente necesario para la corrida del caso. Por tanto, el programa controlador del algoritmo evolutivo no sólo debe correr el algoritmo sino además regular la interacción con el usuario.

7.3. Programa controlador del algoritmo evolutivo

Decidió usarse un motor de AA.EE. existente en lugar de implementar los algoritmos evolutivos como parte del proyecto. Esto se hace en línea con la directiva general de evitar «reinventar la rueda», tanto para evitar tiempos de desarrollo largos como para asegurar la continuidad del proyecto evitando implementaciones complejas. En particular, la implementación de un motor de AA.EE. sería particularmente onerosa.

Existe una amplia gama de motores de AA.EE. que se pueden usar para la computación genética; de entre ellos se optó por ECJ [17], que ofrece las siguientes ventajas:

- Madurez y completitud: es un motor altamente probado con veinte años de desarrollo que implementa muchos modelos básicos de individuo y provee bibliotecas de operadores genéticos implementados para éstos. Se evita así la necesidad de implementar los operadores como parte del proyecto, además de dejarse al futuro usuario

la flexibilidad de usar otros operadores o modificar sus parámetros de forma simple.

- **Amplia documentación:** ECJ ofrece un manual extenso, pero de referencia rápida [18].

También es ésta una ventaja para los implementadores futuros, quienes podrán inmiscuirse en el funcionamiento del motor sin necesidad de aprender las idiosincrasias de motor *ad hoc*.

ECJ es un motor implementado en la plataforma *Java*, por lo que la elección condicionó a escribir el resto del programa en la misma. Algunas ventajas que tiene esta plataforma, que se ponen de manifiesto en la solución propuesta, son:

Eficiencia *Java* resulta ideal para aplicaciones de cómputo; esto será beneficioso en particular para la ejecución de mínimos cuadrados no lineales en el ajuste de curva.

Robustez Los programas en *Java*, incluso en sus primeras iteraciones, son muy resistentes ante fallos o situaciones inesperadas. Esto es ventajoso para el tipo de desarrollo del presente proyecto, de tiempo limitado.

Amplia biblioteca de cómputo No sólo se hace uso de ECJ, sino también de la biblioteca matemática *Apache Commons Math*. Esto evita la necesidad de implementar algoritmos de cálculo para el ajuste de curva.

Biblioteca gráfica Facilidad para prototipar una interfaz gráfica.

Multiplataforma Permite independizarse de la plataforma de elección del usuario.

7.4. Arquitectura cliente-servidor

El requerimiento de que PSS[®]e ejecute desde su API *Python*, sumado a la elección de *Java* como plataforma para la ejecución del motor de AA.EE., plantean la necesidad de un protocolo de comunicación entre ambos extremos: el motor de AA.EE. corriendo bajo *Java* debe ser capaz de invocar procedimientos en la interfaz *Python* de PSS[®]e.

De entre las opciones posibles para la intercomunicación, se opta por una arquitectura cliente-servidor: los programas controladores de ECJ y PSS[®]e correrán concurrentemente, y se comunicarán mediante un modelo

donde el segundo esperará a la conexión del primero para sincronizar actividades.

Debido a que el motor de AA.EE. hace uso de PSS[®]e para evaluar la aptitud de los individuos, el modelo indica que el programa controlador de ECJ funcione como cliente y el controlador de PSS[®]e lo haga como servidor.

Para la comunicación entre los procesos fue necesario definir un protocolo de comunicación. Para esto existen varias opciones, siendo la principal pregunta si es conveniente hacer un protocolo *ad hoc* sobre TCP/IP, o utilizar una plataforma existente. En este caso, se optó por usar la biblioteca de comunicación XML-RPC.

XML-RPC [19] es una implementación de *Remote Procedure Call*, una metodología de computación distribuida donde un programa invoca un procedimiento en el espacio de memoria de otro¹.

El uso de esta biblioteca tiene las siguientes ventajas:

- Uso de TCP/IP como protocolo de transporte, lo que no sólo permite que cliente y servidor se encuentren corriendo en distintos nodos de una red; sino que, además, da lugar a que ejecuten en sistemas operativos diferentes.
- Implementación para *Java* [20] y soporte en la API de *Python*².
- Como consecuencia del punto anterior, pasaje directo de objetos de un espacio de memoria a otro, incluyendo tipos tan complejos como diccionarios y arreglos.

XML-RPC permite la definición de puntos de entrada en el servidor a los que accede el cliente. En la implementación del programa controlador de PSS[®]e se define sólo uno, `correr_caso`.

7.5. Conceptos de AA. EE. en la solución

Se ofrece en esta sección un mapeo de los conceptos teóricos desarrollados en el capítulo 4 a los elementos prácticos de la solución.

Individuo El *individuo* o *solución candidata* debe representar un modelo uninodal determinado, pasible de ser ejecutado en PSS[®]e. La descripción del modelo se hace en la sección 8.2.

¹La parte XML del nombre hace referencia a que está implementado usando ése formato de texto; sin embargo, para la mayoría de los casos —como para este programa— es un detalle de implementación irrelevante.

²*The Python Standard Library 2.7.17*, <https://docs.python.org/2/library/simplexmlrpcserver.html#module-SimpleXMLRPCServer>

Genoma Representación interna en el algoritmo evolutivo de los parámetros necesarios para ejecutar el modelo uninodal asociado que el cliente envía al servidor para su evaluación. Su contenido se describe en la sección 9.2.1.

Función de *fitness* Consiste en ejecutar una corrida en PSS[®] e usando los valores presentes en el genoma del individuo como parámetros del modelo uninodal, conformando junto a los parámetros de ejecución del algoritmo toda la información necesaria en el modelo uninodal. La ejecución producirá una *curva de salida* a partir de la cual se puede hacer la evaluación del *fitness* de ese individuo para esta ejecución particular.

Cruzamiento y mutación Estos operadores genéticos son implementados directamente por ECJ. La estructura y opciones de selección, cruzamiento y mutación se discuten en la sección 9.2.3.

Elitismo ECJ incorpora automáticamente el elitismo según una opción de configuración, pasando los mejores individuos a la siguiente generación. En la herramienta se usa un valor constante del 20 %³.

7.5.1. Generación inicial

Se mencionó en la sección 4.2 que garantizar que los individuos de la población inicial se encuentren «cerca» (en términos de búsqueda) del individuo ideal puede tener un efecto dramático sobre la velocidad de convergencia: en las primeras generaciones se pueden obtener valores muy buenos para algunas partes del genoma, limitando así el espacio de búsqueda. Es entonces esencial depositar la mayor cantidad posible de conocimiento sobre la solución buscada en la generación inicial.

De la observación empírica se conoce que la regulación primaria es el principal efecto observable en los primeros segundos de la respuesta de la frecuencia. Siguiendo lo desarrollado en la sección 3.5, se concluye que a partir de la curva de referencia del problema debería ser posible determinar, de forma aproximada, usando la técnica desarrollada en el capítulo 5, la *inercia total* del sistema $H^{(0)}$ y el *amortiguamiento total* (o *equivalente*) $D^{(0)}$. Si, además, el sistema tiene generación con regulación primaria, se comporta —de forma aproximada— como si hubiera un único generador con regulación primaria actuando; de esta forma también

³El valor seleccionado es un valor común en la práctica; sin embargo, sería deseable determinar si este valor es óptimo para el problema, tal como se hizo con otros parámetros (sección 10.1), véase sección 12.2.1.

Capítulo 7. Arquitectura de la solución

debe ser posible determinar un *tiempo de retardo de regulación equivalente* $T_g^{(0)}$ y un *estatismo equivalente* $R^{(0)}$.

Entonces, puede afirmarse que *para un modelo uninodal con un único generador cuyos parámetros sean estos valores equivalentes, su respuesta será aproximada a la observada en la curva de referencia*. Si bien ésta es la hipótesis de trabajo inicial, será puesta a prueba y confirmada en la sección 10.3.

Resta el problema de obtener estos parámetros equivalentes. A priori existen dos maneras, en base a lo desarrollado.

Obtención a partir de la red original

A partir de la red original, haciendo uso de la API de PSS^{®e}, se pueden obtener datos de un caso PSS^{®e}, incluidos los datos de modelos dinámicos. Recorriendo de forma programable los modelos de los generadores síncronos y extrayendo los valores de H —según visto en la sección 3.3— se puede estimar la inercia equivalente del sistema como $H \approx \sum_i H_i$. De la misma manera, se puede estimar el amortiguamiento equivalente como $D \approx \sum_i D_i$; sin embargo, no se tiene en cuenta el amortiguamiento introducido por las cargas.

En cuanto a los parámetros de regulación primaria, se puede estimar el estatismo también recorriendo las máquinas síncronas —según visto en la sección 3.5— como $R \approx \parallel_i R_i$.

Sin embargo, en la práctica, la estimación de un retardo de regulación equivalente T_g usando el mismo método no es una tarea realizable, dada la complejidad de los modelos de regulación primaria habitualmente presentes en una red original. No todos los modelos de máquina cuentan con un parámetro equivalente T_{gi} que unifique el retardo propio de la regulación, por lo que esto debería hacerse caso a caso para modelos arbitrariamente complejos —muchas veces las redes incluyen modelos especificados por el usuario—. No es posible entonces calcular un equivalente T_g en función de los tiempos de retardo de regulación obtenidos de las máquinas —si bien se dedujo una fórmula en la sección 3.5—.

Obtención a partir de la curva de entrada

Si bien estas funciones dan ecuaciones que no se pueden resolver explícitamente para los parámetros (excepto en el caso de $H^{(0)}$), sí es posible obtenerlos a partir de la curva de referencia usando un algoritmo iterativo: por ejemplo, mínimos cuadrados no lineales (usado en la solución presentada).

La convergencia de un algoritmo de ajuste de mínimos cuadrados, tanto absoluta en principio como rápida en segundo término, está condi-

cionada por los valores iniciales: se necesita una semilla que garantice la convergencia hacia el ajuste deseado.

Por tanto, para obtener los parámetros a partir de la curva de entrada es necesario contar con, al menos, una estimación inicial que se pueda obtener de puntos identificables de la curva.

Para ello se utiliza la teoría desarrollada en el capítulo 5 como parte de este trabajo: a partir de puntos identificables en la curva de entrada se obtienen buenas aproximaciones para los parámetros intermedios definidos en ese capítulo, los cuales se ajustan mediante mínimos cuadrados no lineales y desde cuyos valores ajustados es posible obtener una aproximación para $H^{(0)}$, $D^{(0)}$, $T_g^{(0)}$ y $R^{(0)}$.

Ruido gaussiano

Si bien la aproximación inicial para estos parámetros equivalentes proveerá una buena semilla para el algoritmo evolutivo, para asegurar la convergencia es necesario dotar de cierta entropía a la generación inicial, a fin de que tenga suficiente diversidad genética para evolucionar.

Para generar los individuos iniciales se toman las aproximaciones para $H^{(0)}$, $D^{(0)}$, $T_g^{(0)}$ y $R^{(0)}$, y se les superpone un ruido gaussiano con un σ variable (se determina un valor óptimo mediante experimentación, ver sección 10.1); de esta forma la generación inicial se encuentra *rodeando* a la solución (teóricamente) buscada, que de cumplirse la hipótesis arriba mencionada se parecerá a la dada por los parámetros equivalentes.

El significado de este σ variable es el siguiente: la desviación estándar esperada del ruido es la cantidad a la que se está aplicando el ruido ponderada por σ . En otras palabras, la componente H generada para un individuo i será

$$H_i = \mathcal{N}(H^{(0)}, \sigma H^{(0)}) = H^{(0)} (1 + \sigma \mathcal{N}(0, 1))$$

con el σ especificado.

Capítulo 8

Programa controlador de PSS^{®e}

El *programa controlador de PSS^{®e}* tiene como fin obtener el resultado de la simulación dinámica en PSS^{®e} de una consulta realizada por el *programa controlador del motor de AA.EE.*, dando como respuesta la curva que muestra la evolución en el tiempo de la velocidad del centro de inercia ante un desbalance de potencia.

Una consulta se define a partir de un conjunto de *parámetros de entrada*, con los cuales se especifican los datos de un modelo uninodal, la falta que se debe simular y las características de esta simulación.

8.1. Implementación

Se hizo uso de la versión 34 de PSS^{®e} *Xplore*, una versión de prueba de libre uso, disponible para uso educativo. Su uso se limita para redes con un máximo de 50 barras, lo cual no es una condición restrictiva para simulaciones en un modelo uninodal.

Usando la API de PSS^{®e} [21], se implementó en *Python* una serie de clases de forma de automatizar las simulaciones dinámicas de faltas con salida de servicio de cargas o generadores.

Vista también la necesidad de implementar funciones que permitieran trabajar con, además del modelo uninodal, modelos complejos de redes usadas como referencia (por ejemplo el modelo de la red interconectada uruguayo-argentina), dicha implementación fue de mucha utilidad.

Si bien estas funciones no forman parte de la herramienta principal, dieron lugar a la estructura del código desarrollado en el mismo por lo dicho anteriormente.

Clase Element: representa un elemento de la red que puede ser una carga, un generador convencional o un generador no convencional.

Capítulo 8. Programa controlador de PSS[®]e

Existen métodos que permiten cambiar el estado y los parámetros — tanto de las consignas de potencia como dinámicos— del elemento en PSS[®]e.

En caso de tratarse de un generador, el objeto tiene información del modelo dinámico de la máquina y del regulador primario si corresponde.

Clase Sep: se utiliza para el control de todo lo referido a un *caso PSS[®]e* y datos de modelos dinámicos.

Permite la realización de flujos de carga, obtención de datos del caso PSS[®]e como cantidad, ubicación y potencias de cada elemento (generadores y cargas) de la red y datos de modelos dinámicos (como inercia de los generadores convencionales).

Es posible también agregar nuevos elementos al sistema, lo cual será útil para la construcción del modelo uninodal.

Clase DynamicSim: permite el manejo de simulaciones dinámicas de eventos de salida de servicio de, en principio, un elemento saliente (ver sección 8.3.1). Se basa en los pasos para la realización de simulaciones dinámicas que se detallan en A.1 y debe estar asociado a un objeto Sep.

Se realiza el cálculo de la velocidad del centro de inercia, seleccionando las velocidades en los ejes de los generadores convencionales como los canales de salida.

Teniendo en cuenta que esto se utiliza también para redes grandes, como la red uruguayo-argentina, se define un máximo de veinte canales, con el fin de minimizar, dentro de las posibilidades, el tiempo de simulación. Para minimizar el error en la estimación, se eligen los generadores que mayor inercia aportan al sistema de forma de poder estimarla fielmente.

Las acciones que se describen en este capítulo se realizan haciendo uso de esta implementación.

8.2. Modelo uninodal

8.2.1. Parámetros de elementos

El modelo uninodal se define a partir de los datos que determinan las características de los elementos que son parte del mismo.

Estos elementos —generadores o cargas— se distinguen a partir de los modelos utilizados, de los parámetros que definen a estos modelos y del escenario de generación-demanda que se quiere representar en el mismo.

A continuación se listan los datos y parámetros necesarios para definir cada tipo de elemento que puede estar presente.

Generador convencional

- Datos de flujos de potencia
 - P : Potencia activa generada.
 - Q : Potencia reactiva generada.
 - $P^{(\text{máx})}$: Reservas de la máquina.
 - S_{base} : Potencia aparente que se usa como base en la máquina.
- Datos dinámicos de la máquina
Se utilizará el modelo GENCLS, en el que se deben definir las constantes:
 - H : Inercia de la máquina.
 - D : Constante de amortiguamiento. En el conjunto de generadores convencionales del modelo uninodal, esta constante representa el efecto de amortiguamiento equivalente del sistema original.
- Datos dinámicos del sistema de regulación.
En caso de que el generador convencional cuente con sistema de regulación de frecuencia, para éste se utiliza el modelo IEEEG2, donde se deben definir:
 - K : inverso del estatismo.
 - T_g : tiempo de governor. El modelo IEEEG2 en realidad cuenta con tres parámetros de tiempo (ver A.2). Se define $T_g = T_1 = T_2 = T_3$; de esta manera, el modelo se asemeja más al modelo teórico visto en la sección 3.5.2.
 - $P^{(\text{máx})}$: Máxima potencia que puede entregar la máquina primaria. Se define igual a la máxima potencia que puede entregar el generador síncrono.
 - T_w : Tiempo del agua. Este parámetro está limitado internamente de la siguiente manera: $T_w > 4T_{step}$. Es deseable minimizar el efecto de este retardo temporal, por lo que se busca minimizar el valor de T_w .

Capítulo 8. Programa controlador de PSS[®]e

Por otro lado, el tiempo T_{step} determina la cantidad de muestras de la curva de salida. Este valor se desea que sea lo más grande posible de forma de minimizar el tiempo de simulación y lo suficientemente pequeño como para no perder información de la misma curva.

El valor óptimo encontrado fue $T_w = 0,2$ s. Si en algún caso no se cumpliera la condición para el T_{step} definido desde la entrada, se redefine como: $T_{step} = \frac{1}{4,1} T_w$.

Generador no convencional

- Datos de flujos de potencia
 - P Potencia activa generada.
 - Q Potencia reactiva generada.
 - $P^{(máx)}$ Se toma $1,2 P$.
- Datos dinámicos de la máquina

Se utilizan los modelos WT4G2, para representar un generador eólico, y PVGU1, para representar un generador solar fotovoltaico, presentados en la sección A.2. Se utiliza para los parámetros de estos modelos valores típicos definidos en [22].

Carga

- Datos de flujos de potencia
 - P Potencia activa demandada.
 - Q Potencia reactiva demandada.
- Parámetros modelado de las cargas

Se utiliza el modelo polinomial con: $P_p = 0,7$, $P_l = 0,0$ y $P_z = 0,3$.

8.2.2. Parámetros de entrada

Son el conjunto de datos necesarios para la ejecución de una simulación en el modelo uninodal.

A efectos del problema, teniendo en cuenta el punto anterior, se pueden clasificar los parámetros necesarios para la ejecución de simulaciones distinguiendo a aquellos que forman parte del genoma del individuo de los que no.

Los parámetros de entrada se clasifican entonces como:

Parámetros del individuo Como se vio en el capítulo 3, son los parámetros dinámicos de los generadores convencionales y los parámetros dinámicos de los correspondientes reguladores de frecuencia —en particular R y T_g — los que determinan el comportamiento de la frecuencia en un sistema.

Si bien el escenario de generación-demanda y la falta que se simula son datos imprescindibles, la «identidad» del modelo se define con los parámetros mencionados.

Entonces, para cada generador convencional presente en el modelo uninodal a representarse se distinguen como parte del genoma de cada individuo:

- el valor de inercia del generador, H , y
- el valor de la constante de amortiguamiento D .

Además, para cada generador que tenga regulador de frecuencia,

- el tiempo de retardo del regulador, T_g , y
- el *setting* del estatismo R .

Según el caso de uso elegido (se define en 8.2.5), el genoma puede tomar largos variables, dado que es variable la cantidad de generadores que componen el modelo uninodal. El mismo es entonces un vector de vectores

$$\{H_i, D_i, T_{gi}, R_i\}$$

tales que $\#R_i = \#T_{gi} \leq \#H_i = \#D_i$.

Parámetros de corrida Son datos necesarios para la ejecución de la simulación pero que no forman parte del genoma del individuo. Por un lado, correspondientes a la red original:

- datos del flujo de potencia: vector de potencias, activa y reactiva, generada de los generadores P_i y Q_i ; potencia, activa y reactiva, demandada por la red: C_p, C_q
- datos de la falta: escalón de potencia ΔP , tipo de falta (salida de carga o generador) y
- datos de las reservas de generadores con regulación: vector de potencia máxima por tipo de generador $P_i^{\text{máx}}$.

Por otro lado, datos de ejecución, como:

- tiempo máximo de ejecución $t_{\text{máx}}$,
- tiempo de paso de ejecución t_{step} .

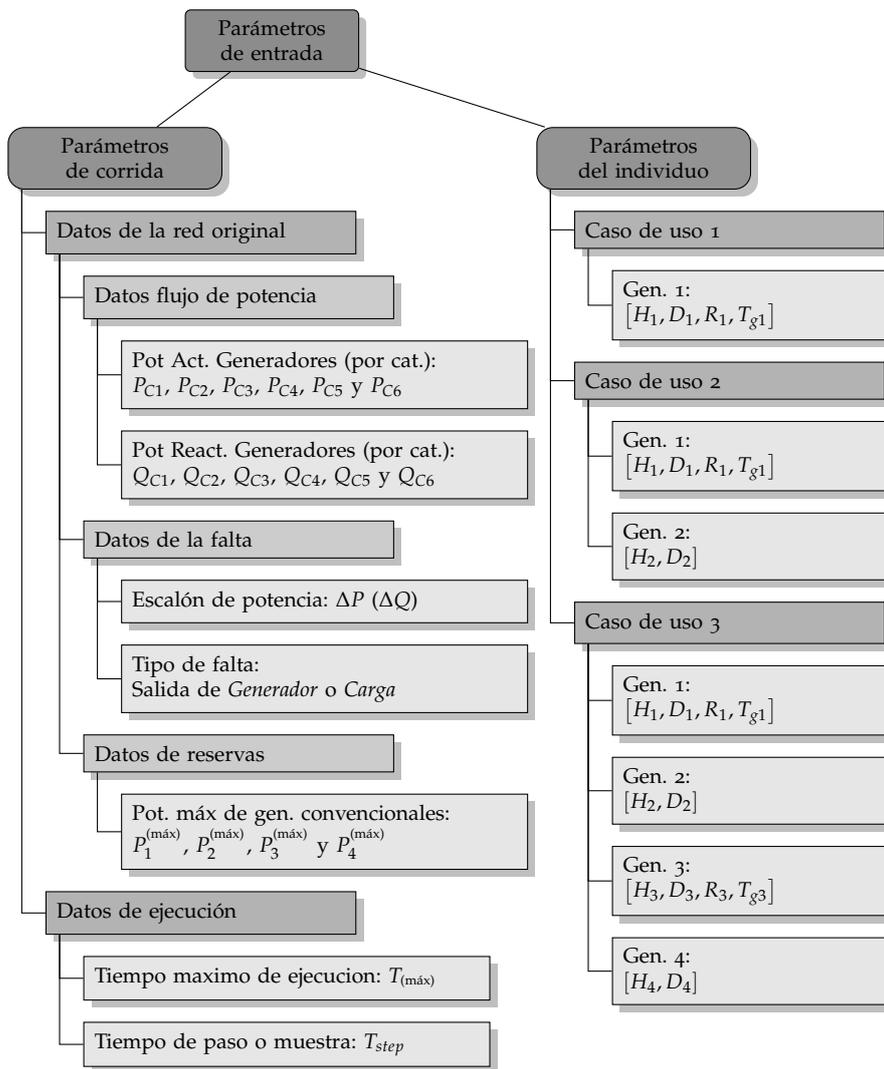


Figura 8.1: Resumen de la clasificación de los parámetros de entrada

8.2.3. Estructura de doble barra

Previo a una simulación dinámica, se debe realizar una simulación de flujo de cargas (ver sección A.1) para establecer un estado de régimen en el sistema previo a la falta.¹

Para ello el modelo uninodal debe contar con un generador y una barra que cumpla la función de barra flotante. El generador G1 cumplirá esta función (en todos los casos de uso que se describen en 8.2.5) y se definen a estos efectos dos barras: *barra flotante* y *barra principal*, tal como

¹Es importante tener en cuenta que esta simulación trabaja con las tensiones y potencias eléctricas como variables. Por tanto, las velocidades de las máquinas síncronas no estarán necesariamente en régimen previo a la falta.

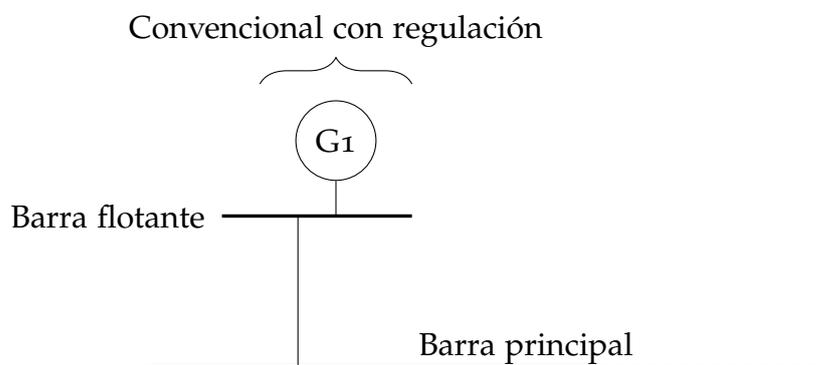


Figura 8.2: Conexión del generador G1 en modelo uninodal

se ve en la figura 8.2.

Si bien el modelo cuenta con dos barras, en los hechos continúa siendo un modelo de un solo nodo debido a que la línea que une la barra flotante y la barra principal es de impedancia nula.

Los demás generadores y las cargas se conectarán a la barra principal.

8.2.4. Modelo base

Se creó un *caso* pss^e como modelo base. El mismo contiene la estructura del modelo uninodal, es decir, el par *barra flotante* y *barra principal* y la línea de impedancia nula que las conecta.

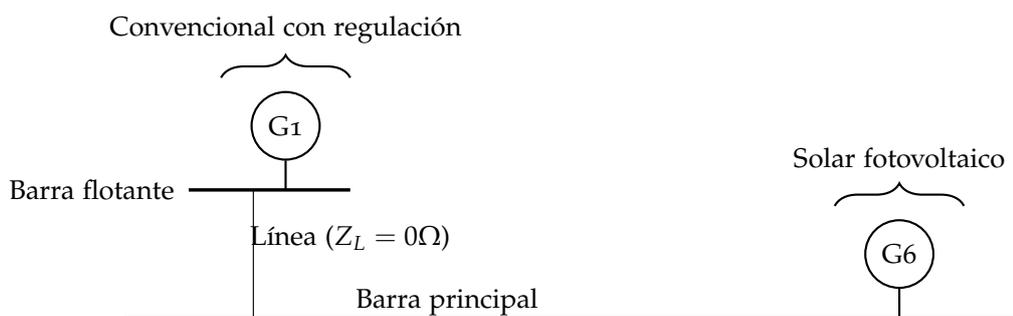


Figura 8.3: Modelo base

En la figura 8.3 se ve que el modelo base incluye también al generador G6, el cual representa la generación solar fotovoltaica. Como se vio en 8.2.1, se utiliza el modelo dinámico PVGU1. Si bien este modelo está incluido en la biblioteca de modelos dinámicos de pss^e , se define como «modelo de usuario», lo que implica que solo se puede usar agregándolo desde un archivo *.dyr* (ver sección A.1): por esto es necesaria su presencia en este modelo base.

En el caso de uso 3 (sección 8.2.5) el generador G6 se elimina, dado que para este caso este tipo de máquina no formará parte del modelo.

8.2.5. Descripción de los casos de uso

Con el fin de cumplir con las necesidades del usuario, se establecen los siguientes *casos de uso* para el modelo uninodal, siendo el objetivo establecer cómo se presentan los generadores en base a su tipo (convencionales y no convencionales) y fuente primaria (hidráulico, térmico, solar fotovoltaico o eólico).

Además, esto también será de utilidad para evaluar el desempeño del programa final cuando aumenta la dimensión del genoma del individuo.²

En todos los casos de uso, los generadores G5 y G6 representan la generación no convencional —eólica y solar fotovoltaica respectivamente— presentes en el sistema original.

Caso de uso 1

El modelo uninodal cuenta con tres generadores (figura 8.4) con las siguientes características:

- un generador convencional con regulación primaria mediante estatismo (G1),
- un generador eólico (G5) y
- un generador solar fotovoltaico (G6).

²Recordar que los parámetros del individuo son los parámetros dinámicos de los generadores convencionales, por lo cual la cantidad de estos determina el tamaño del genoma.

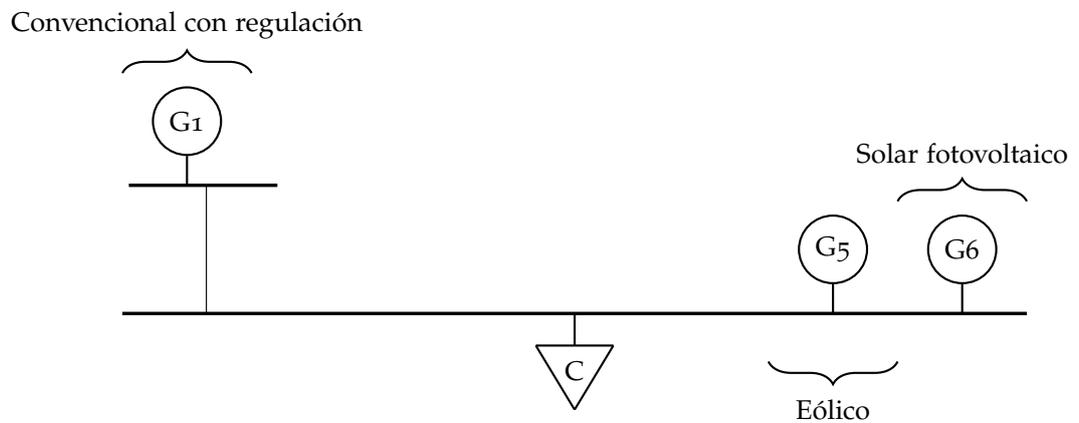


Figura 8.4: Caso de uso 1

El generador G_1 representa el total de generadores convencionales (con y sin regulación) y el total equivalente de la regulación primaria de frecuencia del sistema original.

Las reservas de G_1 representan el total de reservas de todos los generadores con regulación de frecuencia.

Caso de uso 2

Para este caso el modelo uninodal cuenta con cuatro generadores (figura 8.5):

- un generador convencional con regulación primaria mediante estatismo (G_1),
- un generador convencional sin regulación primaria (G_2),
- un generador eólico (G_5) y
- un generador solar fotovoltaico (G_6).

Convencional con regulación

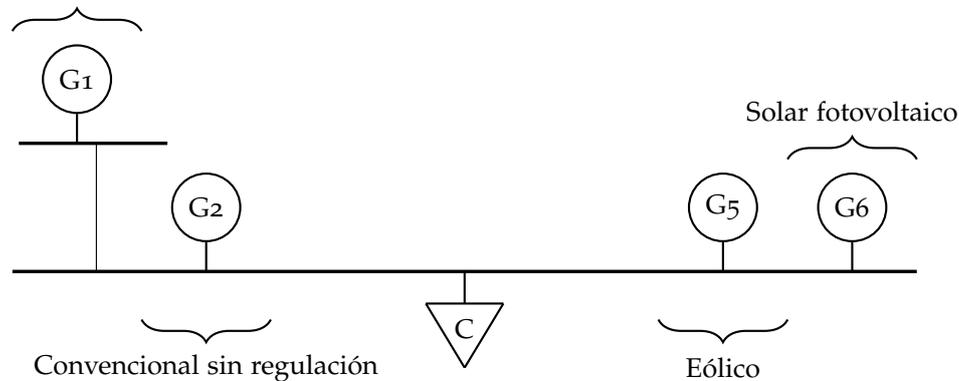


Figura 8.5: Caso de uso 2

El generador G1 representa el total de generadores que cuentan con regulación primaria de frecuencia, mientras que G2 el total que no cuenta con regulación.

Caso de uso 3

Se cuenta con cuatro generadores como se muestra en la figura 8.6, siendo:

- un generador convencional (hidráulico) con regulación primaria (G1),
- un generador convencional (hidráulico) sin regulación primaria (G2),
- un generador convencional (térmico) con regulación primaria (G3) y
- un generador convencional (térmico) sin regulación primaria (G4).

8.3. Ejecución del programa

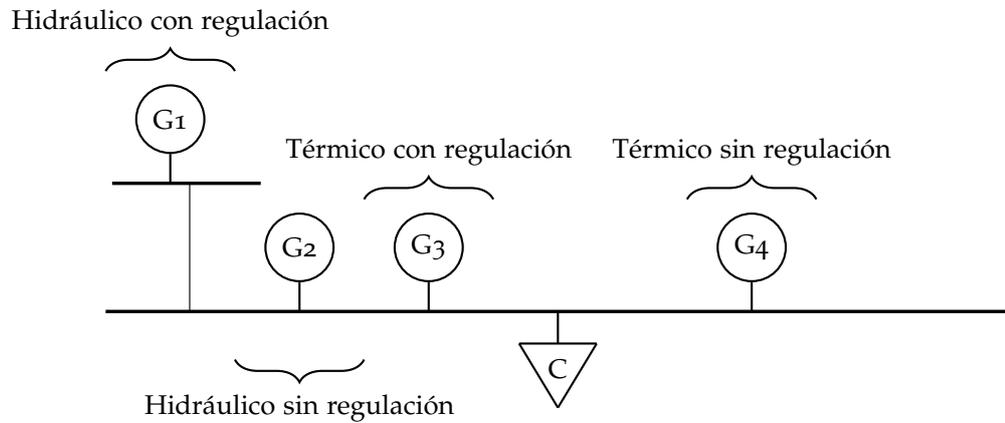


Figura 8.6: Caso de uso 3

Resumen de casos de uso

En la tabla 8.1 se muestra un resumen del rol que representa, en cada caso de uso, cada generador.

Generador	Caso de uso 1	Caso de uso 2	Caso de uso 3
G1	Convencional + Conv. con regulación	Conv. con regulación	Hidráulica con regulación
G2	—	Convencional	Hidráulica
G3	—	—	Térmica con regulación
G4	—	—	Térmica
G5	Eólica	Eólica	—
G6	Solar	Solar	—

Tabla 8.1: Representación de los generadores en modelo uninodal

8.3. Ejecución del programa

Para resolver la consulta del programa controlador del motor de AA.EE., el programa configura un servidor XML-RPC que exporta una función `correr_caso`, que recibe como argumentos los parámetros del individuo y los parámetros de corrida y realiza las siguientes acciones:

- construye el modelo uninodal especificado;
- realiza la simulación dinámica;
- devuelve el resultado de esa simulación.

8.3.1. Construcción del modelo uninodal

Se construye el modelo uninodal a partir del modelo base. A éste se incorporan el elemento saliente, los generadores que correspondan y la carga que representa la demanda. Para cada uno de estos elementos se deben agregar los datos especificados en 8.2.1 que vienen dados en los parámetros de entrada.

Elemento saliente

La potencia en los modelos dinámicos de generadores no convencionales no es una variable accesible durante una simulación dinámica, por tanto para simular una disminución de generación, por ejemplo eólica, se implementa agregando al modelo un generador de este tipo, cuya salida de servicio provocará el desbalance de potencia que se busca.

Para generalizar lo anterior, esto se realiza de la misma manera cuando el tipo de falta es un desbalance provocado por disminución de carga.

Por tanto, el elemento saliente se define —según el tipo de falta— como una carga que consume una potencia ΔP o como un generador no convencional que genera una potencia ΔP que se conecta a la barra principal (figura 8.2).

Modelo dinámico

Cuando el *elemento saliente* es un generador no convencional, es requisito para la ejecución de la simulación dinámica agregar un modelo dinámico (ver sección A.1). Se utiliza el mismo modelo definido en la sección 8.2.1, con los valores típicos para dicho modelo.

Generadores

Para cada generador en el modelo uninodal se deben agregar los datos de flujos de potencia y los datos dinámicos como se vio en la sección 8.2.1. Los parámetros del individuo son los datos dinámicos de los modelos de los generadores convencionales (H_i , D_i para cada uno) y de los modelos de reguladores primarios de frecuencia (T_{gi} y R_i para cada uno). Para los modelos dinámicos de los generadores no convencionales se utilizan valores típicos, extraídos de la guía de aplicación de PSS[®]e [22].

Carga

La carga representa toda la demanda de energía eléctrica del sistema original. Se utiliza un modelo polinomial para su representación, con la siguiente configuración: 70 % de potencia constante y 30 % de impedancia

constante para la potencia activa, y 100 % de impedancia constante para la potencia reactiva.

8.3.2. Simulación

Las simulaciones se realizan siguiendo los pasos que se especifican en la sección A.1. El elemento saliente se mantiene en servicio durante un segundo al inicio de la simulación; esto se realiza con el fin de establecer un estado estable en el sistema previo a la falta para las variables que no se estabilizaron en el flujo de cargas, como por ejemplo, la velocidad de las máquinas. Luego el elemento saliente cambia su estado a «fuera de servicio» provocando un desbalance de potencia $\pm\Delta P$ según corresponda.

La salida de la simulación son las curvas de las velocidades de las máquinas síncronas.

8.3.3. Respuesta

A partir de la salida de la simulación se obtiene la curva de la velocidad del centro de inercia del modelo uninodal, dada como serie temporal. Ésta se calcula según lo visto en 3.3, teniendo en cuenta a todos las máquinas convencionales presentes en el modelo.

8.4. Otras utilidades

También se implementaron otras herramientas con el fin de facilitar el desarrollo de pruebas. Como fue mencionado en este capítulo, estas no forman parte del programa principal.

Entre estas herramientas se encuentran funciones cuyo objeto es obtener datos de un caso PSS[®]e correspondiente a una red arbitraria, así como las curvas de referencia que se utilizan para la ejecución de la herramienta.

8.4.1. Obtención de curvas de red de referencia

Llamamos *red de referencia* u *original* al modelo del sistema para el cual se busca un equivalente dinámico uninodal para una determinada falta.

Para el desarrollo del programa se utilizaron modelos de diversos sistemas de potencia, entre ellos:

- Modelo de una red utilizada en ámbito educativo: *Red de pruebas "A"*, que cuenta con 25 nodos, adaptada para el presente trabajo incorporando modelos de generación ERNC.

- Modelo de la red interconectada uruguayo-argentina, que fue construido por CAMESSA y prestado por UTE para el desarrollo del presente trabajo.

Al igual que para las simulaciones en el modelo uninodal, la simulación se realiza mediante la desconexión de un elemento saliente. En este caso, se selecciona de entre los elementos del sistema.

8.4.2. Obtención de parámetros de la red original

Previo a la ejecución de la herramienta es necesario obtener los parámetros de la red necesarios para la evaluación de cada individuo.

Por un lado es necesario conocer la potencia generada por cada fuente, la potencia demandada y los datos de la falta simulada como ΔP ; mientras que por otro, para calcular la velocidad del centro de inercia es necesario obtener la inercia de las máquinas síncronas conectadas al sistema.

Se definen las siguientes categorías de generadores para poder discriminar y obtener los parámetros necesarios de cada tipo de máquina, y así cubrir las diferentes opciones que el usuario podrá elegir:

- C1** Categoría 1: generadores hidráulicos que cuentan con control de frecuencia mediante estatismo.
- C2** Categoría 2: generadores hidráulicos sin control de frecuencia.
- C3** Categoría 3: generadores térmicos que cuentan con control de frecuencia mediante estatismo.
- C4** Categoría 4: generadores térmicos sin control de frecuencia.
- C5** Categoría 5: generadores eólicos.
- C6** Categoría 6: generadores fotovoltaicos.

Finalmente los datos obtenidos se presentan, para cada uno de los casos de uso, calculados de la forma mostrada en la tabla 8.2.

Clasificación de generadores

La clasificación de los generadores presentes en el sistema para las categorías definidas anteriormente se realiza según el modelo dinámico que se utiliza para cada una de ellas.

Para el caso de los generadores convencionales, cuando estos cuentan con regulador de frecuencia, éste determina el tipo de turbina que se

8.5. Docker como motor múltiple para paralelismo

Generador	Caso de uso 1	Caso de uso 2	Caso de uso 3
G1	$P_{G1} = P_{C1} + P_{C2} + P_{C3} + P_{C4}$ $p_{G1}^{\text{máx}} = p_{C1}^{\text{máx}} + P_{C2} + p_{C3}^{\text{máx}} + P_{C4}$ $H_{G1} = H_{C1} + H_{C2} + H_{C3} + H_{C4}$	$P_{G1} = P_{C1} + P_{C3}$ $p_{G1}^{\text{máx}} = p_{C1}^{\text{máx}} p_{C3}^{\text{máx}}$ $H_{G1} = H_{C1} + H_{C3}$	$P_{G1} = P_{C1} + P_{C5} + P_{C6}$ $p_{G1}^{\text{máx}} = p_{C1}^{\text{máx}} + P_{C5} + P_{C6}$ $H_{G1} = H_{C1}$
G2	-	$P_{G2} = P_{C2} + P_{C4}$ $H_{G2} = H_{C2} + H_{C4}$	$P_{G2} = P_{C2}$ $H_{G2} = H_{C2}$
G3	-	-	$P_{G3} = P_{C3}$ $p_{G3}^{\text{máx}} = p_{C3}^{\text{máx}}$ $H_{G3} = H_{C3}$
G4	-	-	$P_{G4} = P_{C4}$ $H_{G4} = H_{C4}$
G5	$P_{G5} = P_{C5}$	$P_{G5} = P_{C5}$	-
G6	$P_{G6} = P_{C6}$	$P_{G6} = P_{C6}$	-

Tabla 8.2: Resumen distribución de datos por caso de uso

utiliza; sin embargo, cuando no están presentes, se deben diferenciar a partir del modelo dinámico de la máquina.

Por tanto, a partir del modelo de máquina se tiene:

- GENSAL y GENSAE para generadores hidráulicos
- GENROU y GENRAE para generadores térmicos.
- WT4G2 o WT4G1 para generadores eólicos.
- PVGU1 o WT4G1 para generadores solares fotovoltaicos.

8.5. Docker como motor múltiple para paralelismo

De no usarse paralelismo en cuanto a instalaciones del servidor, se estaría restringido por la cantidad de máquinas capaces de correrlo —cada una requiriendo una instancia de pss[®]e 34 instalada—. El paralelismo es la principal arma de los AA.EE., por lo que es de interés encontrar una forma de multiplicar esta capacidad.

A fin de no sólo poder correr múltiples instancias de pss[®]e en una misma máquina, sino también independizarse de *Windows* como plataforma requerida de trabajo, se creó una imagen de contenedor de *Docker*³, un sistema de contenedores (virtualización liviana) para la plataforma Linux, mediante el uso de *Wine*⁴ como emulador de un entorno *Windows*.

³<https://docs.docker.com/>, <https://www.redhat.com/es/topics/containers/what-is-docker>.

⁴<https://www.winehq.org/>

Capítulo 8. Programa controlador de PSS[®]e

En otras palabras, se instaló `pss®e` en un entorno *Wine* que corre dentro de un contenedor *Docker*. Es posible correr múltiples instancias de este contenedor en una máquina Linux, logrando así el paralelismo.

De esta manera, se lograron ejecutar pruebas masivas en instancias AWS, comprobándose que en una máquina `c5.2xlarge`⁵ pueden correrse unas treinta instancias del servidor concurrentemente, logrando una evaluación típica en un promedio de unos 12 s. Este tiempo puede bajar hasta unos pocos segundos si se usa una máquina mayor o se disminuye el paralelismo.

⁵<https://aws.amazon.com/es/ec2/pricing/on-demand/>

Capítulo 9

Programa controlador del motor de AA.EE.

En este capítulo se detallan las partes principales del programa controlador del ECJ o —pensando desde el punto de vista de la arquitectura cliente-servidor— *programa cliente*, que son las siguientes.

Identificador: se encarga del ajuste de una curva de salida, para extraer sus parámetros.

Controlador de ECJ: control de la biblioteca de AA.EE., incluido su comando y la implementación de las interfaces necesarias para definir los elementos del algoritmo evolutivo.

Comunicación con el servidor.

Bibliotecas auxiliares: incluyen una biblioteca de manipulación de series temporales y una de cálculo simbólico.

Interfaz: incluye dos programas, CLI y GUI.

Reporte: detalla la ejecución.

9.1. Identificador

Toma una serie temporal e identifica sus parámetros H , D , T_g , R , A y t_0 . Para ello ejecuta un ajuste de mínimos cuadrados no lineales con el algoritmo de Marquart-Levenberg, usando la implementación de este algoritmo de *Apache Commons Math* [23] —nuevamente con el criterio de evitar implementaciones *ad hoc* complejas y apostando por la confiabilidad de una implementación probada—.

Un algoritmo de mínimos cuadrados no lineales necesita una aproximación inicial a los parámetros a ajustar, para lo cual se usan las técnicas vistas en el capítulo 5.

9.1.1. Parámetros iniciales

Para obtener las aproximaciones iniciales H^* , D^* , etc., se ejecuta el siguiente algoritmo.

Identificación del valor previo A^* Se toma el primer valor de la serie temporal, asumiendo que durante los primeros instantes el sistema está equilibrado (aunque no necesariamente trabajando a la frecuencia nominal).

Identificación de nadir Se toma el punto más alto o más bajo de la serie (de entre estos dos, el que más diste de A^*), lo cual es adecuado para los dos tipos de respuesta posibles; su abscisa se toma como t_{∇} . El signo de la diferencia entre el valor de nadir y el valor previo $\widehat{\Delta\omega} - A^*$ indica el *signo* de la curva de respuesta, y por tanto del escalón de potencia ΔP de entrada, cuyo signo es opuesto. El algoritmo es válido para ambos signos.

Identificación de H^* y t_0^* Retrocediendo por la serie temporal desde el nadir, se busca el punto de derivada máxima. A partir de allí, se proyecta la tangente hasta la intersección con la ordenada del valor previo A^* , de cuya abscisa se toma t_0^* . Para H^* se toma la mitad del inverso del opuesto de esta derivada, que es el valor más cercano presente en la curva a H^* e ignora cualquier perturbación inicial.

Detección de la oscilación ω^* A partir del nadir, se buscan ceros de la derivada hacia adelante; de encontrarse, se considera que la respuesta es *oscilatoria*. El primer cero se toma como la aproximación al punto t_3 tal como se define en la sección 5.2.2. A partir de la diferencia $t_3 - t_{\nabla}$ se obtiene la estimación para la oscilación ω^* .

De no encontrarse, se considera la respuesta *no oscilatoria*.¹

Detección de p^* , q^* , r^* (no oscilatorio) Se identifican:

- el tiempo de levantamiento de 90 % $t_{90\%}$;
- el primer cero de la segunda derivada desde el nadir hacia adelante, identificado como t_2 , del que se mide su derivada D_2 ; y

¹Esto puede derivar en un ajuste no óptimo; véase sección 12.1.1.

- el punto cuya derivada es $\frac{1}{2,718}D_2$ hacia adelante de t_2 , que se toma como t_3 tal como se define en la sección 5.2.1.

Si la serie temporal termina antes de encontrar un punto con tal derivada, se toma el penúltimo punto de la serie como t_3 a fin de evitar efectos numéricos del corte de la serie.

Se siguen los pasos de la sección 5.5.2 para obtener las estimaciones p^* , q^* y r^* a partir de estos valores.

Detección de ϕ^* y q^* Se busca el punto de cuarto período $t_{1/4}$, intermedio entre t_{\blacktriangledown} y t_2 , del cual se toma su derivada. A partir de los valores identificados se siguen los pasos de la sección 5.5.3 para obtener ϕ^* y q^* .

Con esta aproximación inicial se puede ejecutar el algoritmo de Marquart-Levenberg, con el cual se ajustan los parámetros intermedios a la curva objetivo, representada por

$$F_{no}(A, t_0, H, p, q, r; t) = A - \frac{1}{2H} \frac{1}{2r} \left(\frac{p+r}{q+r} (1 - e^{-(q+r)(t-t_0)}) - \frac{p-r}{q-r} (1 - e^{-(q-r)(t-t_0)}) \right) \quad (9.1)$$

para el caso no oscilatorio, o por

$$F_o(A, t_0, H, \phi, q, \omega; t) = A - \frac{1}{2H} \frac{1}{q^2 + \omega^2} \left(\frac{q \sen \phi + \omega \cos \phi}{\sen \phi} - \frac{q \sen(\omega(t-t_0) + \phi) + \omega \cos(\omega(t-t_0) + \phi)}{\sen \phi} e^{-q(t-t_0)} \right) \quad (9.2)$$

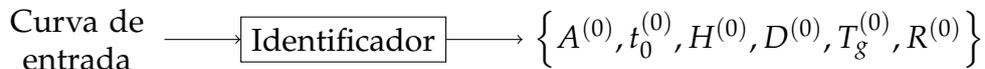
para el oscilatorio.

La ejecución del algoritmo requiere la evaluación de la curva objetivo para un conjunto de parámetros de prueba en cada punto de la serie temporal, así como la evaluación de su jacobiano también en cada punto de la serie. Este último cálculo requiere la derivación de las funciones F_{no} y F_o con respecto a cada uno de sus parámetros antes de la evaluación.

Para una serie temporal, calcular la derivada temporal de la curva que la representa no es difícil, siendo ésta aproximable por el cociente incremental. Sin embargo, el cálculo numérico de la derivada con respecto a las variables que se desea ajustar (no temporales) es ineficiente: si bien *Apache Commons* puede tomar incrementos pequeños y evaluar el cociente incremental, es mejor calcular la derivada *explícita* $\frac{\partial F}{\partial x_i}$ con respecto a cada uno de los parámetros x_i y evaluar en la función derivada.

Si bien las derivadas de F_{no} y F_o con respecto a A , H , etc., están bien definidas, es más económico en términos de tiempo de desarrollo resolver el *problema general* de la derivación y aplicarlo a una función simbólica que declarar en código fuente las doce derivadas necesarias. Para ello se usa la biblioteca de cálculo simbólico desarrollada en 9.4.2, que es capaz de calcular las funciones $\frac{\partial F}{\partial x_i}$ y expresarlas de forma simbólica, así como de evaluar la función a ajustar y las derivadas en los puntos de la serie temporal. Por tanto, se declaran (en el lenguaje definido por la biblioteca) las dos funciones F_{no} y F_o en el código fuente y se usan las entradas de la biblioteca para la evaluación.

El resultado de la ejecución del algoritmo de mínimos cuadrados son los parámetros intermedios de la curva, a partir de los cuales y siguiendo las ecuaciones de las secciones 5.5.2 y 5.5.3 se obtienen los parámetros $A^{(0)}$, $t_0^{(0)}$, $H^{(0)}$, $D^{(0)}$, $T_g^{(0)}$ y $R^{(0)}$ que la definen. Estos últimos cuatro representan los parámetros de inercia, amortiguamiento y regulación primaria equivalente del SEP y, como se postula en el capítulo 3 y se comprueba en la sección 10.3, son una aproximación a los parámetros del modelo unidodal final. Su cálculo es el rol del identificador, que se ejecuta al iniciar el programa con la curva de referencia como entrada.



El identificador se encuentra en el paquete `client.fit`. La clase que ejecuta el algoritmo de mínimos cuadrados es `DroopResponseFitting`, mientras que el algoritmo de identificación de parámetros iniciales es `DroopResponseIdentifier`.

9.2. Controlador de ECJ

ECJ ofrece dos modelos de trabajo: *ECJ en control*, donde la clase del programa principal es provista por ECJ y el programa usuario provee el comportamiento (incluyendo las clases necesarias) a través de un archivo de parámetros; o *control de ECJ externo*, donde una clase principal comanda y controla la ejecución (probablemente iterativa) de ECJ.²

Si bien el primer modelo es una buena herramienta de prototipado, el segundo ofrece mayores posibilidades de control y configuración dinámica, por lo que el programa implementado sigue este último modelo. En la

²Ver sección *Integrating ECJ with other Applications or Libraries* en [18].

sección 9.5 se detallan los programas principales implementados; ambos hacen uso de la componente detallada en esta sección.

El modelo de los AA.EE. es metaheurístico: provee un algoritmo abstracto que no entra en detalles de implementación. ECJ realiza este modelo haciendo uso de las clases abstractas e interfaces de la plataforma *Java*, para las cuales el programa cliente provee instancias de implementación. En el paquete `aaee.engine` se proveen las clases específicas necesarias para que ECJ pueda mantener, cruzar, mutar y evaluar a los individuos.

9.2.1. Individuos y especie

El «individuo» del problema es un vector de reales de largo variable según el caso de uso. ECJ provee la clase `DoubleVectorIndividual`, que es un modelo de genoma de vector de números de doble precisión. Su uso es ventajoso dado que ECJ provee bibliotecas de operadores genéticos (cruzamiento, mutación) para estos vectores, por lo que el esfuerzo de implementarlos se ahorra.

La clase se extiende en la clase abstracta `UninodeIndividual`, donde se agrega comportamiento particular usado en el programa. Luego ésta se extiende en `SingleHydroIndividual`, que conforma el caso 1. A su vez, `DualHydroIndividual` y `ThermalHydroIndividual` implementan los casos 2 y 3, formando una cadena de herencia. Estas tres forman parte del paquete `aaee.engine.individual`.

ECJ tiene el concepto de «especie», clase que utiliza como almacén de elementos comunes a todos los individuos de una clase particular. `DoubleVectorIndividual` está asociada a la especie provista por ECJ `FloatVectorSpecies`, que se extiende en la clase `Species` y se usa para seleccionar el largo correcto del genoma en el individuo según el caso (y seleccionar una de las tres clases de individuo).

Genoma

El genoma del individuo consiste en los parámetros necesarios para su ejecución, detallados en la sección 8.2.2:

- Para cada generador convencional sin regulación primaria se almacenan los parámetros H y D .
- Para cada generador convencional con regulación primaria se almacenan, además, los parámetros T_g y R .
- Los generadores no convencionales no aportan parámetros al genoma, dado que su modelo no tiene elementos particulares por individuo.

9.2.2. Evaluación

La evaluación se hace en la clase `Problem`, donde se implementa el algoritmo desarrollado en el capítulo 6.

Para obtener la curva de respuesta asociada al individuo el genoma es enviado a la componente *cliente* (sección 9.3), que devuelve una curva de respuesta luego de comunicarse con el servidor. Para la resincronización de ambas curvas se detectan t_0 y A utilizando el identificador, y se usa la biblioteca de series temporales (sección 9.4.1) para trasladar y remuestrear. Finalmente, esta última biblioteca también se usa para el cálculo de la distancia entre las curvas.

9.2.3. Cruzamiento y mutación

Debido a que los individuos son instancias de `DoubleVectorIndividual`, pueden usarse los mecanismos propios de ECJ para la producción de individuos nuevos. El *pipeline* de producción está configurado de la siguiente manera:

La *selección* se hace por torneo simple entre dos individuos al azar, y los individuos seleccionados se *cruzan* usando el método *line*: cada parámetro del individuo hijo es un punto al azar entre los parámetros de los padres. Luego, el genoma se muta de forma gaussiana con una probabilidad del 10% (este valor se justifica en la sección 10.1).

9.2.4. Ciclo de ejecución

El ciclo de ejecución se encapsula dentro de la clase `RunLoop`, y se ejecuta en un hilo de ejecución *Java*. Esto permite un mejor desacople del programa principal/controlador, pudiendo interactuar con el usuario mientras el algoritmo ejecuta. De no hacerse así, las evaluaciones bloquearían el ciclo principal, y (por ejemplo) un servidor no responsivo podría ocasionar el bloqueo del programa.

Los programas interfaz arrancan la clase `RunLoop` y pueden extraer información sobre la ejecución a través de un objeto compartido, generación a generación, para mostrar al usuario.

9.3. Comunicación con el servidor

El *cliente* propiamente dicho debe recibir un genoma y comunicarlo al programa servidor para que éste construya el modelo uninodal y devuelva su curva de respuesta. La componente «cliente» —en el paquete

`client` y cuya clase principal es `Client`— encapsula este comportamiento recibiendo un genoma y devolviendo una curva a través de su interfaz `query()`.

ECJ permite múltiples hilos de evaluación para aprovechar el paralelismo, por tanto se mantiene *una instancia de esta componente por servidor*: los servidores (identificados por una dirección IP y un puerto) son un parámetro del programa cliente, y al inicializarse el programa principal se crean tantas instancias de esta clase como servidores son especificados. Luego cada hilo de evaluación accede a una y sólo una de esas instancias.

Además del genoma, la interfaz `query()` acepta los *parámetros de ejecución* que son pasados desde la clase que evalúa los individuos. Tanto estos parámetros como el genoma deben pasarse a formatos conocidos por XML-RPC para su uso —proceso conocido como *serialización*—: tipos primitivos, diccionarios y arreglos. Dado que los parámetros de consulta son estándar en todos los casos, la serialización es hecha por la componente cliente. Sin embargo, debido a que el tamaño del genoma es variable según el caso, la serialización y deserialización de individuo a objeto XML-RPC y viceversa es delegada a las clases de individuo.

Ambos objetos son enviados al procedimiento remoto (del servidor) `correr_caso`, que devuelve una serie temporal en formato de par de arreglos, deserializada al recibirse al formato de serie temporal (delegado a la biblioteca de series temporales).

Además de retornar la serie temporal, `query()` ejecuta la identificación de parámetros de la serie. Esto se hace para descartar de forma temprana individuos que produzcan resultados inválidos; cuando esto pasa se devuelve una excepción.

En el caso de que no sea posible detectar nadir o punto de inflexión en la curva de respuesta (indicado en base a las excepciones lanzadas por el identificador), el cliente puede llegar a reintentar la consulta bajo el supuesto de que la ventana temporal ha sido demasiado corta. Para ello reenvía la consulta incrementando esta ventana repetidas veces hasta un largo de ventana máximo prefijado, luego de lo cual se asume inválido el resultado.

9.4. Bibliotecas auxiliares

9.4.1. Series temporales

El objeto *serie temporal*, definido en el capítulo 6, es central en el estudio. Encapsula la relación entre una función de respuesta *continua*, es-

Capítulo 9. Programa controlador del motor de AA.EE.

tudiada como respuesta del sistema en los capítulos 3 y 5, y su muestreo *discreto* hecho por `pss`[®]e. Existen diversas operaciones que se repiten a lo largo del programa sobre las series temporales, tanto para obtener información (por ejemplo para ajustar la curva de respuesta) como de transformación (remuestreo, recorte, traslación, etc.).

Estas operaciones se encapsulan en el objeto `Timeseries`, en el paquete `client.timeseries`, que representa dos sucesiones —temporal `t` y funcional `f`— del mismo tamaño, y que posee la siguiente interfaz.

`load()` y `save()` Carga y persistencia de series temporales desde y hacia archivos en disco, representados en forma de archivo de texto con dos columnas (series temporal y funcional).

`length()` Devuelve el largo de la serie.

`diff()` Devuelve una serie temporal con el vector de cocientes incrementales de la serie representada, equivalente a la primera derivada discreta. La serie temporal devuelta tiene un largo una unidad inferior a la representada.

`movingavg()` Para muchas aplicaciones, es de interés suavizar la curva (filtro pasabajos) para eliminar puntos irregulares. Devuelve una serie temporal que es el resultado de tomar promedios de ventana móvil con el ancho de ventana especificado.

`downsample()` En ocasiones, es de interés disminuir la granularidad de una serie para su trabajo más eficiente en ajustes. Devuelve una serie interpolada a partir de la original con un largo especificado.

`resample()` Interpola los valores de la serie funcional para los valores de una serie temporal especificada. Devuelve una nueva serie temporal con la serie temporal especificada y la funcional interpolada.

`findF()` Busca el índice temporal para el cual la curva es más próxima al valor *funcional* especificado, posiblemente a partir de cierto índice y con una dirección específica. Es usada por los algoritmos de ajuste para encontrar ceros, etc.

El valor funcional es especificado en forma de condición: se encuentra el primer valor que cumple la condición especificada.

`findT()` Busca el índice temporal más cercano al valor *temporal* especificado.

`sub()` Recorta una serie entre dos índices temporales.

`cut()` Recorta una serie entre dos tiempos, o entre los tiempos extremos de otra serie especificada.

`timeshift()` y `fshift()` Traslada la serie en los ejes temporal y funcional el valor especificado.

`distance()` Calcula la distancia entre curvas como definida en la sección 6.3.2. Asume que cubren el mismo intervalo temporal y que están sincronizadas entre sí en cuanto a tiempo de falta.

9.4.2. Cálculo simbólico

Es usada para el ajuste de mínimos cuadrados no lineales, tanto para la evaluación de la función como para el cálculo del jacobiano (derivación analítica) en diferentes puntos temporales. Sirve también como puente entre el mundo continuo de los jacobianos necesarios para el ajuste y el mundo discreto de la serie temporal, usada para el ajuste.

Su objeto de trabajo es la *función temporal*, que representa una función tanto del tiempo como de una cantidad arbitraria de parámetros reales. Permite la creación, orientada a objetos con una interfaz funcional, de funciones arbitrariamente complejas.

Su implementación es una de las pocas instancias donde se eligió reinventar la rueda: si bien existen otras implementaciones de cálculo simbólico, se vio que adoptar una de las existentes (por lo general demasiado genéricas) hubiera resultado más trabajoso, con dudosas ganancias.

Composición de funciones

La forma de crear objetos es mediante la composición a través de tipos básicos, que son:

`Time` Representa el tiempo t , parámetro de la función.

`Constant` Representa una constante, pasada como parámetro.

`Parameter` Representa un parámetro de la función respecto del cual es posible, por ejemplo, derivarla. El parámetro viene dado por un índice —parámetro 0, 1, etc., la función tiene tantos parámetros como el mayor índice especificado—, es responsabilidad del usuario de la biblioteca asociar los índices a los parámetros específicos al usarla.

A partir de ellos se pueden componer funciones. Todo objeto funcional —incluidos los básicos— tiene una interfaz que incluye por ejemplo `sum()`, aceptando otro objeto funcional y devolviendo la suma de ambos; `sqrt()`, devolviendo un objeto que representa la raíz cuadrada; etc.

A modo de ejemplo, se muestra la implementación de la ecuación (9.2):

```
TimeFunction
  A = TimeFunction.param(0),
  H = TimeFunction.param(1),
  phi = TimeFunction.param(2),
  q = TimeFunction.param(3),
  w = TimeFunction.param(4),
  t0 = TimeFunction.param(5),
  time = TimeFunction.t().minus(t0);

TimeFunction
  term1 = q.times(TimeFunction.sin(phi)).plus(
    w.times(TimeFunction.cos(phi))),
  term2 = TimeFunction.exp(q.times(time).negative()).times(
    q.times(TimeFunction.sin(w.times(time).plus(phi))).plus(
    w.times(TimeFunction.cos(w.times(time).plus(phi)))));
oscillatingFunction = term2.minus(term1). // Simplificando el menos exterior
  over(q.square().plus(w.square())).
  over(TimeFunction.sin(phi)).
  over(TimeFunction.constant(2).times(H)).times(
    TimeFunction.heaviside(5)).plus(A);
```

La evaluación y derivación se ejecutan mediante las interfaces:

`derive()` Acepta un índice de parámetro y devuelve la función derivada con respecto a ese parámetro.

`eval()` Acepta una serie temporal y valores para los parámetros, evalúa la función para esos valores de parámetros en los puntos especificados y devuelve una serie funcional.

`jacobian()` Acepta una serie temporal y valores para los parámetros, deriva la función para cada parámetro con `derive()` y luego evalúa las derivadas con `eval()`. Devuelve un arreglo de series funcionales —una por parámetro—.

9.5. Interfaz

Se ofrece una interfaz de línea de comandos (CLI) y una visual (GUI). Como parte del cliente, son programas *Java* y como tales requieren JRE³

³*Java Runtime Environment*, <https://www.oracle.com/technetwork/java/javase/overview/index.html>.

instalado para funcionar.

Ambos programas son multiplataforma y están empaquetados de forma que se puedan ejecutar sin la instalación de software adicional: los archivos *jar* que se suministran incluyen todas las bibliotecas necesarias, incluyendo gráficas, para la ejecución inmediata. Estos son `uninode-cli.jar` y `uninode-gui.jar`.

Las clases principales de los programas se implementan en el paquete `aaee`, mientras que las específicas a la interfaz GUI se encuentran en el paquete `aaee.gui`.

9.5.1. Parámetros del algoritmo

Una corrida del algoritmo evolutivo, ya sea por CLI o GUI, requiere una serie de parámetros obligatorios y otros opcionales. Ambas interfaces tienen diferentes formas de especificarlos.

Servidores Obligatorio; son los servidores que se usarán para evaluar los individuos, en forma de lista de direcciones IP y puertos.

Curva de entrada Obligatorio; necesario para definir un *fitness*. Viene dado en forma de archivo de serie temporal, tal como aceptado por la interfaz `load()` de la biblioteca de series temporales.

Escalón de entrada Especifica en p.u. el tamaño de ΔP que definió la curva de entrada. De no especificarse, se considera unitario.

S_{base} La base de potencia, sólo relevante a efectos de interpretar el escalón de entrada junto a los parámetros de curva.

Caso 1, 2 ó 3. Especifica el caso de modelo uninodal a correr.

Vector P Vector de (máximo) seis elementos que indican la potencia entregada por cada generador del modelo uninodal en el momento previo a la falta. El tamaño de la carga se hace inicialmente igual a la suma total.

Vectores H , D , T_g y R iniciales Si bien por defecto la generación inicial se toma desde la curva de entrada, es posible, para poder correr cierto tipo de pruebas, comenzar con valores forzados de los mismos (a los que se le agrega un ruido gaussiano) para la generación inicial.

Cantidad de individuos y generaciones

$t^{(\text{máx})}$ Límite para la curva de entrada; permite ignorar efectos luego de cierto tiempo en la misma.

9.5.2. CLI

La interfaz de línea de comandos es ideal para trabajos por lotes, o simple de usar cuando no se necesita una visualización. A los parámetros necesarios para la corrida del algoritmo se los especifica en un archivo `uninode.params` que debe estar presente en el directorio activo donde se ejecuta el programa. Su formato es el aceptado por los archivos de parámetros de ECJ (sección *The Parameter Database* en [18]). Es posible aquí redefinir cualquier comportamiento permitido por ECJ, por lo que —comparada con la GUI— otorga bastante más poder al usuario.

La curva de entrada se pasa como único parámetro de entrada. Los servidores se pasan mediante el archivo de parámetros, bajo la llave `init.servers`; mientras que el número de generaciones e individuos pueden especificarse con las llaves `generations` y `pop.subpop.0.size` respectivamente.

9.5.3. GUI

La interfaz gráfica permite una mayor interacción con los datos de entrada y es más amigable para familiarizarse con el trabajo en el programa.

En la ventana principal (figura 9.1) se pueden especificar parámetros generales: tamaño del escalón ΔP , cantidad de generaciones e individuos, así como el caso de uso. También se permite seleccionar si los datos para la generación inicial se deben tomar desde los datos detectados de curva (comportamiento por defecto) o son especificados por el usuario —para lo cual se provee un área donde, según el caso, se habilitan opciones—. El área también incluye el reparto del vector P de generación en el momento previo a la falta. Los servidores se especifican en una pestaña para este fin.

Cuenta con un área gráfica donde se muestra la curva de entrada contrastada con la curva del ajuste inicial, además de los parámetros de curva hallados en la identificación (figura 9.1). A lo largo de la ejecución se muestra también el mejor individuo hallado hasta el momento (figura 9.2).

Biblioteca gráfica multiplataforma

Para la biblioteca gráfica se usó *Eclipse SWT*⁴, que permite la ejecución en diferentes entornos gráficos, incluidas las plataformas más conocidas *Windows*, *Mac OS X* y la interfaz gráfica *X* usada por, entre otros sistemas compatibles, *GNU/Linux*.

La biblioteca para estas tres plataformas se empaquetó dentro de la distribución `uninode-gui.jar`, por lo que el archivo `jar` de la GUI debe

⁴<https://www.eclipse.org/swt/>.

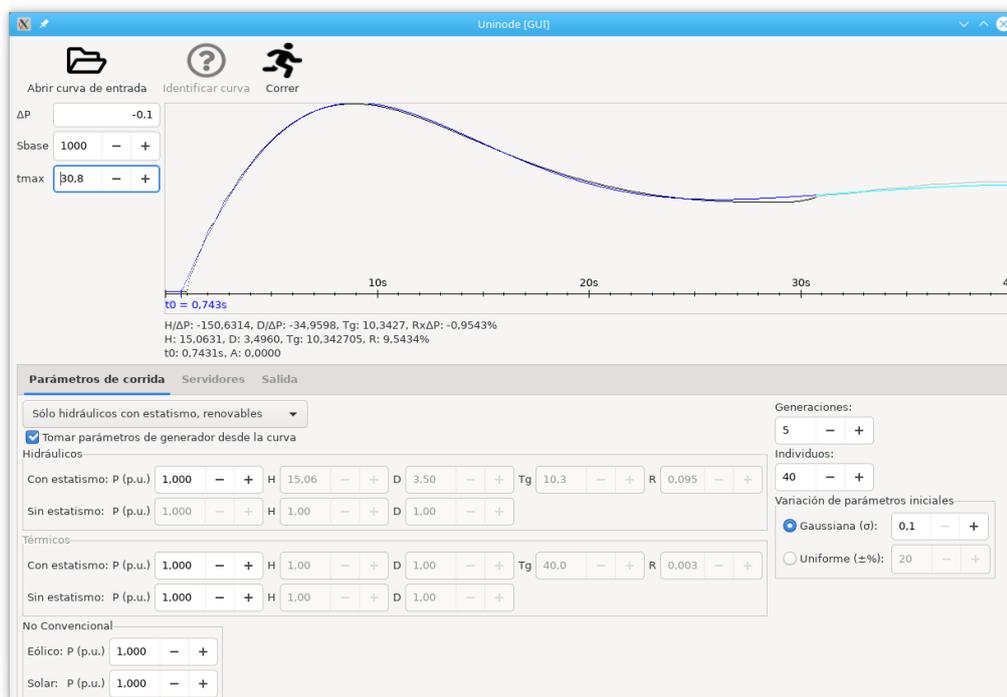


Figura 9.1: Interfaz GUI corriendo bajo GNU/Linux

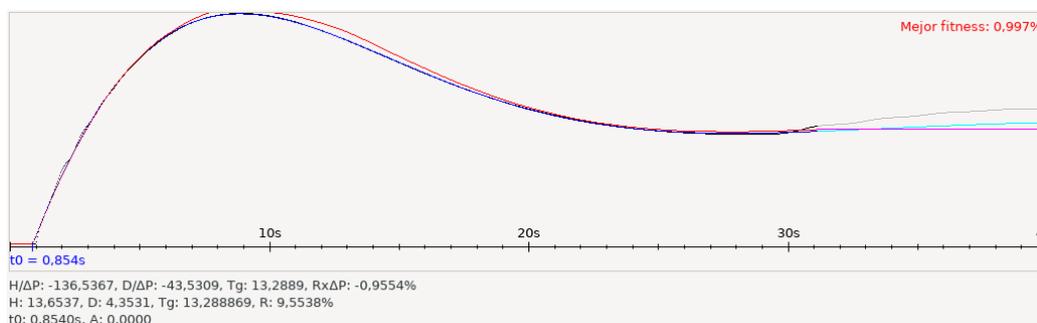


Figura 9.2: Área gráfica de la interfaz GUI durante la ejecución

funcionar sin cambios en cualquiera de ellas. En los hechos, se ejecutaron pruebas funcionales sobre *Windows 10*, *Mac OS X 10.15* y *Fedora 30*, lo que asegura una amplia cobertura en sistemas modernos sin cambio alguno.

Íconos

Para los íconos de la GUI (ver por ejemplo la barra de herramientas en la figura 9.1) se utilizó la biblioteca gráfica *Font Awesome*⁵. Los íconos se usan bajo la licencia *Creative Commons Atribución 4.0 Internacional* [24],

⁵<https://github.com/FontAwesome/Font-Awesome>.

que permite su uso libre en un trabajo de esta categoría.

9.6. Reporte

La ejecución del algoritmo genera un reporte en formato \LaTeX orientado tanto al usuario del programa como al desarrollador.

Cuenta con un resumen de ejecución que muestra los parámetros iniciales obtenidos de la curva de entrada y los parámetros de ejecución del algoritmo (cantidad de generaciones e individuos, servidores, etc.), y un gráfico mostrando la evolución del *fitness* a través de las generaciones (figura 9.3).

Para cada generación muestra los individuos ordenados por *fitness*, pudiendo optarse por una versión resumida donde se listan los mejores diez o la generación completa (figura 9.4). Esto también incluye los individuos cuya evaluación no produjo un *fitness*, y se muestra la causa en cada caso.

El reporte está implementado en el paquete `aaee.report`. Éste incluye algunos archivos `.tex` como encabezados, pie de página, etc., pero, en su mayor parte, el reporte se genera con una interfaz orientada a objetos que conoce y acepta los tipos presentes en el programa (individuos, series temporales, etc.). Esto trae ramificaciones a otros paquetes: por ejemplo, dado que el listado de individuos incluye todo su genoma, la generación de la tabla es delegada a la clase individuo correspondiente (de forma similar a la serialización para XML-RPC).

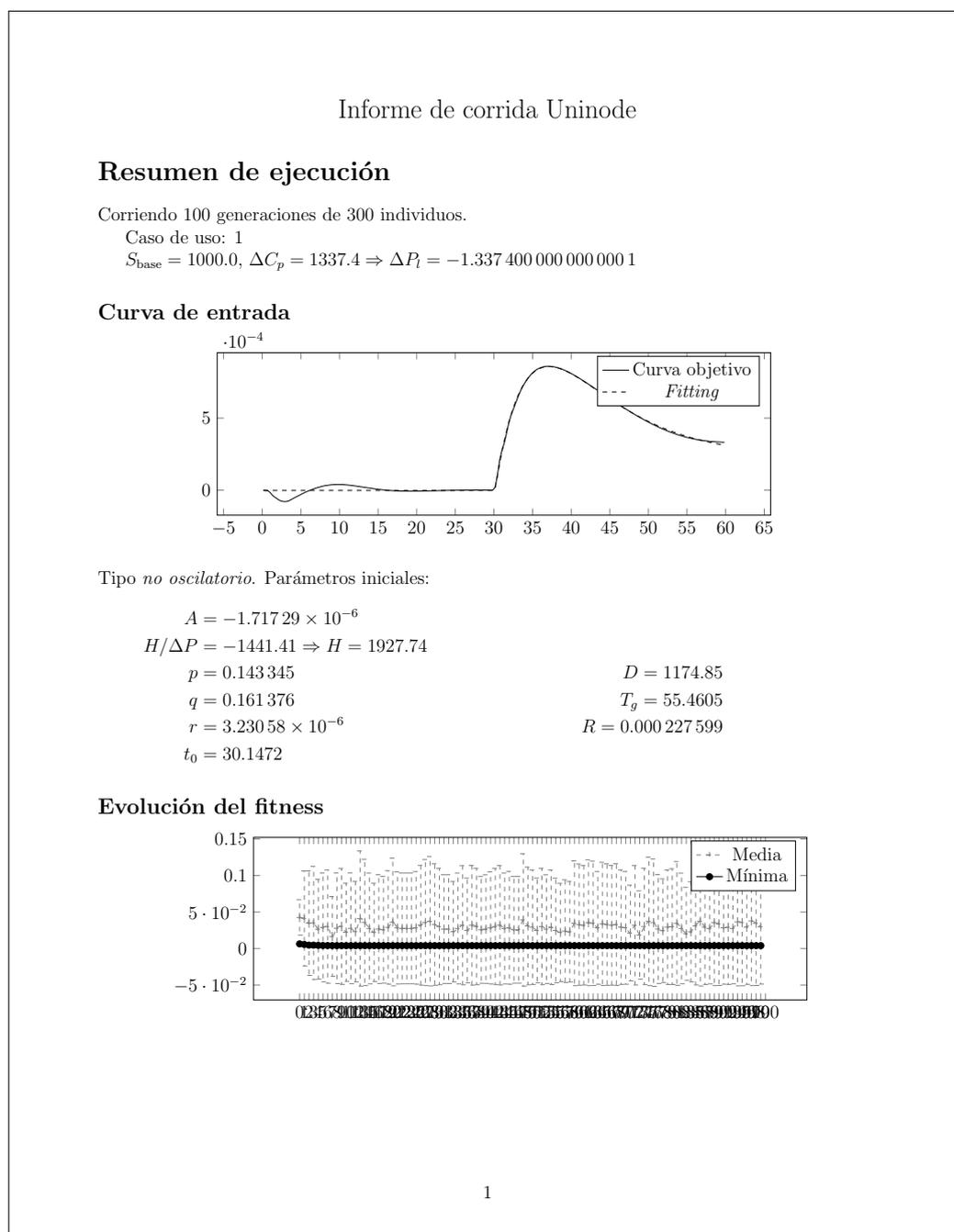


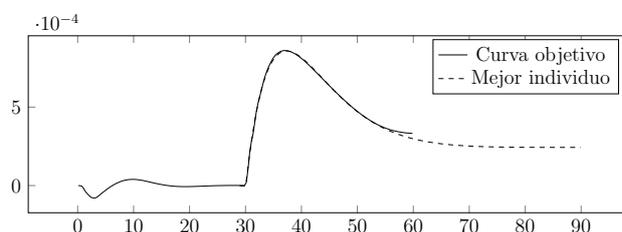
Figura 9.3: Resumen de ejecución en el reporte

Detalle de generaciones

Generación 0

Mejor individuo de la generación, fitness 0.00630173

$$H = 2027.18 \quad D = 1144.62 \quad T_g = 51.4666 \quad R = 0.0239372\%$$



No.	H	D	T_g	R (%)	Fitness
1	2027.178	1144.623	51.467	0.024	0.006302
2	1991.784	1162.864	59.320	0.022	0.007608
3	1832.971	1193.121	53.312	0.025	0.007745
4	1958.793	1167.467	54.847	0.023	0.008944
5	1955.538	1180.292	60.277	0.021	0.009853
6	1967.353	1128.673	56.235	0.022	0.011235
7	1956.552	1212.044	57.901	0.023	0.011245
8	2021.947	1127.490	56.722	0.022	0.011616
9	1948.142	1205.320	51.796	0.024	0.011643
10	1867.715	1221.573	53.588	0.025	0.012043

290 individuos con *fitness* mayor no mostrados.
Fitness promedio no mostrados: 0.0436186 ± 0.0239030

Generación 1

Mejor individuo de la generación, fitness 0.00550097

$$H = 1971.19 \quad D = 1172.08 \quad T_g = 54.9035 \quad R = 0.0232311\%$$

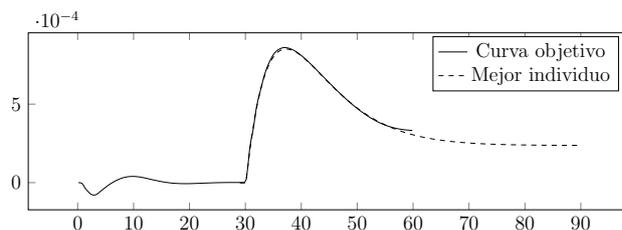


Figura 9.4: Resumen de generación en el reporte

Parte IV

Resultados y conclusiones

Capítulo 10

Evaluaciones y resultados

10.1. Parámetros de exploración aleatoria

Los algoritmos evolutivos tienen diversos parámetros que es necesario ajustar para lograr que haya un buen balance entre la *exploración* del espacio de soluciones y la *explotación* apropiada de zonas fértiles de dicho espacio.

Los parámetros ideales dependerán del problema particular, por lo que es imposible a priori saber qué valores resultarán en una convergencia aprobada o rápida. Sin embargo es posible, con el debido conocimiento del problema, hacer una búsqueda guiada de estos parámetros.

En esta sección se da cuenta de pruebas hechas a fin de determinar tres parámetros que influyen en la exploración del espacio de soluciones: la variabilidad de las soluciones de la generación inicial, la probabilidad de mutación y la variabilidad de esta última.

La metodología será estimar un rango donde los parámetros puedan ubicarse, y ejecutar sucesivas pruebas con estos parámetros; para finalmente con los resultados tabulados concluir qué parámetros obtuvieron mejor resultado.

10.1.1. Parámetros a ajustar

Variabilidad de las soluciones de la generación inicial σ_0 : La generación inicial se crea a partir de los parámetros teóricos obtenidos a partir del ajuste de la curva de entrada (sección 7.5.1), a los cuales se les agrega un ruido gaussiano (sección 7.5.1). La amplitud de este ruido determina qué tan grande es el espacio de búsqueda inicial.

Ensachar el espacio de búsqueda sólo será beneficioso si la estimación inicial no es buena; aunque incluso siendo ésta buena, puede

Capítulo 10. Evaluaciones y resultados

ser beneficioso cierto ancho mínimo para la buena explotación de la zona.

Se prueba una amplia gama de valores: 1; 0,5; 0,2 y 0,1.

Probabilidad y variabilidad de mutación p_M y σ_M : Tradicionalmente, en AA.EE. se usa una probabilidad baja de mutaciones, esperando que los cruzamientos conduzcan a soluciones óptimas con una mínima intervención de cambios aleatorios de este tipo.

Sin embargo, al ser las variables presentes en el problema de tipo continuo, es de esperarse que pequeñas mutaciones de tipo gaussiano alrededor de los valores del genoma tengan chances de mejorar ligeramente la aproximación.

Para la probabilidad se prueba nuevamente una amplia gama: 0,1 %, 0,5 % y 1 % que se encuentran dentro de los valores típicamente usados; y 5 % y 10 % que se consideran valores extremadamente grandes.

Para la variabilidad se prueban sólo dos valores, 0,1 y 1, dado que se presume que la chance de mutación tendrá una influencia mucho mayor.

Se ejecuta un número de pruebas en el cubo de combinaciones definido por esos tres parámetros, *ceteris paribus*.

10.1.2. Condiciones de ejecución

Curva de pruebas

La curva de pruebas se muestra en la figura 10.1; ésta se obtiene a partir del centro de inercias de una red mediana, con una salida de carga. El algoritmo se ejecuta con un tiempo máximo de 40 segundos.

Generaciones, individuos y parámetros

Se corren ejecuciones de 300 generaciones con una población de 200 individuos.

El esquema de evolución es cruzamiento tipo *line* [18] con una extensión del 25 %; a la cual se superpone una mutación gaussiana con los parámetros pasados en la prueba. Se usa un elitismo del 20 %.

10.1. Parámetros de exploración aleatoria

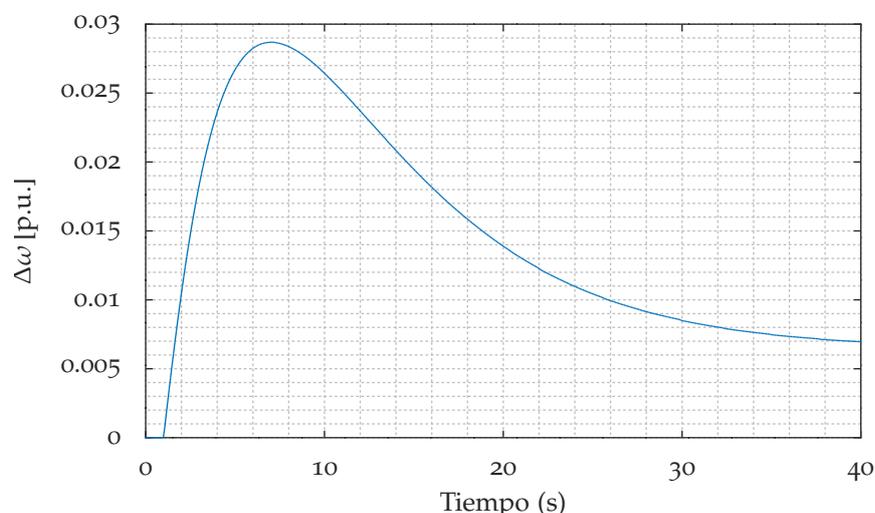


Figura 10.1: Curva de pruebas

10.1.3. Resultados de las corridas

Se ejecutan siete series de pruebas cruzando las tres dimensiones de variables a probar, resultando en $7 \times 2 \times 4 \times 5 = 280$ corridas. Una de las corridas falla por lo que se obtienen $N = 279$ resultados.

La mediana del *fitness* final de todas las corridas es de 0,215 %, mientras que el percentil 90 (más bajo) está dado por 0,209 %.

Los resultados se agrupan según las variables probadas para su análisis, que se hace por separado. Para cada valor se muestran el *fitness* final promedio (con su desviación estándar), el porcentaje de corridas cuyo *fitness* final sea *a*) menor que la mediana y *b*) menor al percentil 90, la generación promedio en la que el *fitness* pasó por debajo de *a*) la mediana y *b*) el percentil 90, y la cantidad de corridas.

Variabilidad de la generación inicial

σ_0	Fit. prom.	< med.	#gen < med.	< p. 90	#gen < p. 90	N
0,1	0,21 ± 0,01 %	94 %	19,4	20 %	82,6	70
0,2	0,23 ± 0,04 %	61 %	26,6	10 %	90,6	70
0,5	0,30 ± 0,12 %	30 %	36,3	7,1 %	93,2	70
1	0,44 ± 0,25 %	13 %	45,1	0 %	—	69

Para este parámetro se ve cómo una menor variabilidad conduce a una amplia mayoría de resultados (94 %) cuyo *fitness* está por debajo de la mediana —es decir, más cerca del óptimo—, además de alcanzar un *fitness* promedio menor. Se ve también que hay una convergencia un poco más rápida, decreciendo el número de generaciones necesarias para ubicarse el *fitness* promedio por debajo de la mediana.

Capítulo 10. Evaluaciones y resultados

Se concluye que para la generación inicial, el mejor valor de entre los probados para la variabilidad inicial es el de $\sigma_0 = 0,1$.

Probabilidad de mutación

p_M	<i>Fit. prom.</i>	< med.	#gen < med.	< p. 90	#gen < p. 90	<i>N</i>
0,1 %	$0,34 \pm 0,21 \%$	34 %	16,9	7,1 %	36	56
0,5 %	$0,30 \pm 0,20 \%$	41 %	36,1	3,6 %	60,5	56
1 %	$0,29 \pm 0,14 \%$	46 %	26,5	7,1 %	61,2	56
5 %	$0,28 \pm 0,14 \%$	55 %	24,1	3,6 %	170	55
10 %	$0,25 \pm 0,09 \%$	73 %	25,1	25 %	100	56

Se observa cómo una mayor probabilidad de mutación conduce a una mayor cantidad de corridas que ubican su promedio de *fitness* por debajo de la mediana, incluyendo un *fitness* promedio menor.

Si bien parece existir una convergencia hacia valores por debajo de la mediana o por debajo del percentil 90, esto puede deberse a que mayor probabilidad de mutación produce una mayor cantidad de individuos que se alejan de la zona de buen *fitness*, produciendo un área de búsqueda mayor.

De todas formas, se concluye que es beneficioso tener una probabilidad mayor de mutación, en este caso $p_M = 10 \%$.

Variabilidad de mutación

σ_M	<i>Fit. prom.</i>	< med.	#gen < med.	< p. 90	#gen < p. 90	<i>N</i>
1	$0,28 \pm 0,15 \%$	49 %	31,7	7,2 %	104	139
0,1	$0,30 \pm 0,18 \%$	51 %	20,2	11 %	76,1	140

La variabilidad en la mutación no parece tener efectos muy significativos —ubicándose el *fitness* promedio en valores similares—, aunque con una mejor cantidad de corridas ubicadas por debajo del percentil 90 a la versión de $\sigma_M = 1$.

Se concluye que si bien el parámetro no tiene mucho efecto, se debe optar por una variabilidad de $\sigma_M = 1$ de entre las probadas.

10.1.4. Conclusión

Se recomiendan los siguientes valores para los parámetros probados:

σ_0 : 0,1

p_M : 10 %

σ_M : 1

Estos valores serán utilizados en el resto de las pruebas.

10.2. Evaluación del modelo uninodal como equivalente dinámico de una red

10.2. Evaluación del modelo uninodal como equivalente dinámico de una red

El objetivo de esta esta sección es evaluar la calidad del modelo uninodal obtenido a partir de la ejecución del algoritmo evolutivo como equivalente de la red a la que modela, verificando si distintas faltas ocurridas en uno y otro resultan en respuestas aproximadas entre sí.

10.2.1. Motivación y objetivo

Se sabe que el modelo uninodal resultado es una buena representación para la falta que se produjo en la red original utilizada como *curva de referencia* para la ejecución del algoritmo; se quiere además evaluar cómo se comporta frente a otras faltas.

Ante salidas de carga de la misma dimensión, se espera una respuesta de la frecuencia similar en ambos modelos —el de la red original y el uninodal—. Aquí «similar» debe ser evaluado en términos relativos, por ejemplo utilizando la noción de *distancia entre campos escalares* desarrollada en el capítulo 6 que se usa como *fitness*.

Dado que el *fitness* del individuo óptimo obtenido con el algoritmo es la distancia a la curva de referencia obtenida desde la red original, se espera que la distancia entre curvas obtenidas para diferentes faltas en ambas redes esté relativamente cerca de este *fitness* —o igual o ligeramente mayor—, dado que el individuo obtenido está optimizado para la falta original.

10.2.2. Metodología

1. Se obtiene una *curva de referencia* de una red original, sacando de servicio un elemento para generar un desequilibrio de potencia.
2. A partir de dicha curva y los datos de la falta, se ejecuta una corrida de la herramienta. Para simplificar la ejecución se realiza para el caso de uso 1. Se obtiene así un modelo uninodal equivalente.
3. Para un conjunto de elementos (cargas y generadores ERNC) de la red original:
 - a) Se simula un desbalance de potencia ΔP_e en la red original, sacando de servicio el elemento e , obteniendo la curva $C_{\Delta P_e}^{(\text{orig})}$.
 - b) Se simula el mismo desbalance de potencia ΔP_e en el modelo uninodal, obteniendo la curva $C_{\Delta P_e}^{(\text{uninodal})}$.

Capítulo 10. Evaluaciones y resultados

- c) Se calcula la distancia entre las curvas $C_{\Delta P_e}^{(\text{orig})}$ y $C_{\Delta P_e}^{(\text{uninodal})}$ usando la distancia definida para el cálculo del *fitness*.

Redes de prueba

Se prueba en dos redes:

- la red de prueba "A", cuyo modelo se incluye en el anexo C; y
- el modelo de red uruguayo-argentina, utilizado por UTE y actualizado a 2018.

Se realizan diferentes tipos y cantidad de pruebas para ambas, dado que las pruebas en la red de prueba "A" son simples y permiten una mayor cantidad en un tiempo razonable, comparado a la red interconectada uruguayo-argentina cuyas simulaciones son costosas.

10.2.3. Evaluación de red de prueba "A"

Datos de la red

La red cuenta fuente cuenta con:

- 15 cargas, con una demanda total de 2140 MW;
- 5 generadores convencionales, con una potencia generada total de 1317 MW: hidráulica 817 MW y térmica 500 MW; y
- 5 generadores no convencionales, con una potencia generada total de 850 MW: 700 MW de tipo eólico y 150 MW fotovoltaico.

Pruebas realizadas y modelos uninodales resultantes

Se realizan tres pruebas con diferentes tipos de falta; para cada una de estas pruebas se sigue el procedimiento delineado en la sección 10.2.2. La curva de referencia para cada prueba se obtiene a partir de las siguientes simulaciones de faltas:

Prueba 1: salida de carga de 280 MW (elemento «1»).

Prueba 2: salida de generador eólico de 150 MW (elemento «17»).

Prueba 3: salida de carga de 235 MW (elemento «12»).

10.2. Evaluación del modelo uninodal como equivalente dinámico de una red

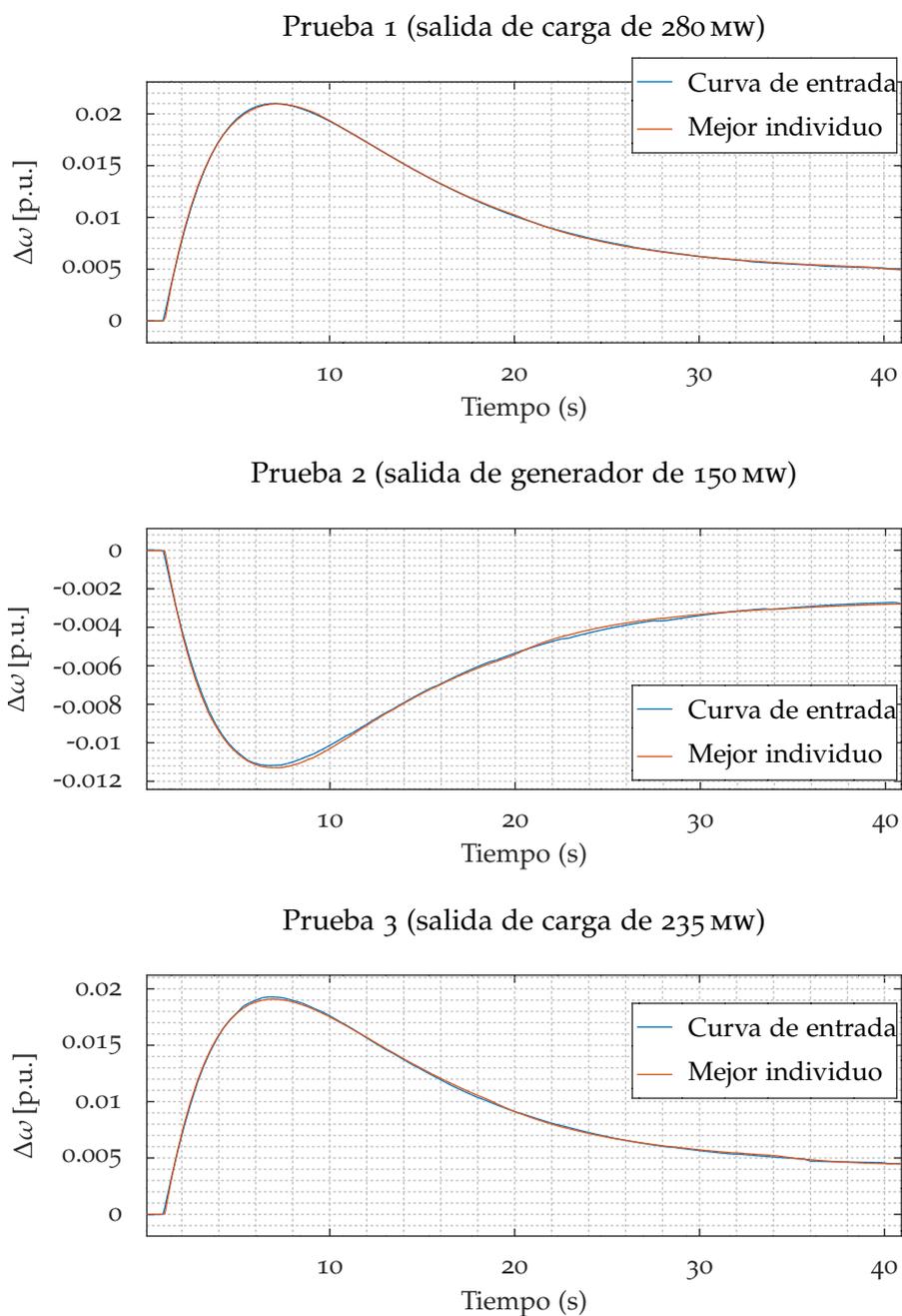


Figura 10.2: Curvas de referencia (azul) y curvas de salida del mejor individuo para cada prueba (rojo)

Capítulo 10. Evaluaciones y resultados

A partir de las curvas de referencia se obtienen los modelos equivalentes uninodales. En la figura 10.2 se muestran las tres curvas de referencia junto a la curva de salida del mejor individuo obtenido en cada prueba.

A continuación, se ejecutan las simulaciones de salidas de veinte distintos elementos («1» a «20»), tanto en la red original como en los tres modelos uninodales obtenidos.

Resultados y discusión

En la tabla 10.1 se muestran los resultados de las pruebas. En la fila superior se destaca el *fitness* obtenido por el mejor individuo en el cálculo de los tres modelos uninodales; mientras que entre los veinte casos, para cada prueba, se resalta en negrita la prueba correspondiente a la falta que originó esa prueba.

El orden de los elementos en la tabla, ascendente con respecto a la prueba 1, se corresponde de forma aproximada con una noción de «distancia geográfica» dentro del modelo en términos de nodos intermedios y largo de líneas.

<i>Fitness</i> del mejor individuo:				0,268 %	0,586 %	0,277 %			
Elemento saliente	Tipo	Nodo	Pot. (mw)	Prueba 1		Prueba 2		Prueba 3	
				Distancia	NI [†]	Distancia	NI	Distancia	NI
1	Carga	201	280	0,268 %	0	0,704 %	6	4,073 %	8
2	Carga	201	120	0,889 %	0	0,790 %	6	3,902 %	8
6	Carga	801	380	0,309 %	3	0,700 %	3	3,901 %	5
3	Carga	401	80	1,726 %	3	1,622 %	5	3,830 %	9
4	Carga	501	50	1,980 %	3	1,978 %	3	2,603 %	7
10	Carga	1201	200	0,591 %	4	0,775 %	4	3,685 %	4
5	Carga	601	100	1,038 %	4	1,035 %	6	4,074 %	7
9	Carga	1101	115	0,826 %	5	0,907 %	3	4,040 %	5
16	ERNC	901	200	1,065 %	5	1,326 %	1	4,465 %	6
7	Carga	901	80	1,707 %	5	1,580 %	1	3,410 %	6
8	Carga	901	50	2,038 %	5	2,034 %	3	2,495 %	6
17	ERNC	1001	150	0,718 %	6	0,589 %	0	3,500 %	7
18	ERNC	1001	100	0,991 %	6	0,80 %	2	3,094 %	8
11	Carga	1401	145	2,978 %	6	2,702 %	5	1,242 %	2
19	ERNC	1501	150	0,741 %	7	0,696 %	7	3,048 %	1
20	ERNC	2001	250	1,427 %	7	1,650 %	7	4,978 %	3
14	Carga	1901	120	3,767 %	8	3,546 %	7	0,615 %	3
15	Carga	1901	105	4,127 %	8	3,745 %	7	0,659 %	3
12	Carga	1601	235	4,337 %	8	3,972 %	7	0,277 %	0
13	Carga	1701	80	4,430 %	8	4,243 %	7	0,905 %	2

[†] NI: cantidad de nodos intermedios

Tabla 10.1: Comparativa para cada elemento saliente en la red de prueba y el uninodal

10.2. Evaluación del modelo uninodal como equivalente dinámico de una red

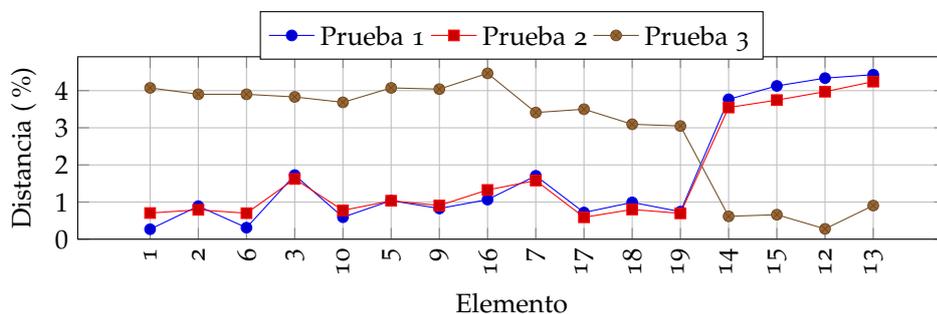


Figura 10.3: Distancia entre curvas según elemento saliente

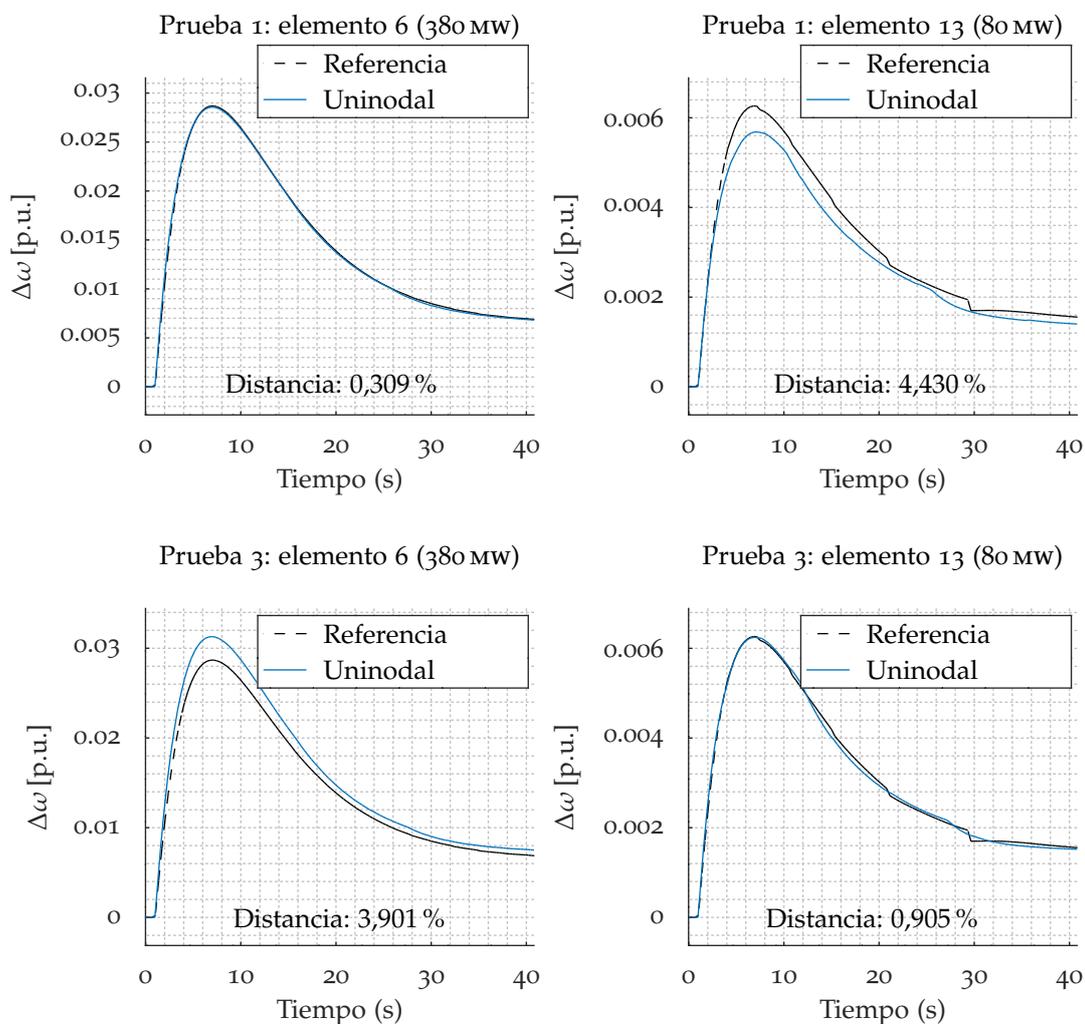


Figura 10.4: Salidas de distintos escalones en red de referencia y uninodal

Se observa que para los elementos más «alejados» —en términos de nodos intermedios— al lugar donde se produce la falta original cuya curva se usa como referencia, la distancia entre las curvas de salida de la red original y el modelo uninodal es mayor (figura 10.3).

En los dos diagramas superiores de la figura 10.4 se presentan las curvas comparativas para algunos casos críticos de la prueba 1: un elemento vecino («6») y el elemento más alejado («13»); mientras que en los dos inferiores se presentan los mismos casos para la prueba 3 donde se da la situación inversa: el elemento «6» es uno de los más alejados y el «13» el más cercano.

Al mismo tiempo, para ΔP pequeños se observan efectos que repercuten en el cálculo de la distancia; esto es visible en las curvas para el elemento «13» en la misma figura.

10.2.4. Prueba en red interconectada uruguayo-argentina

Datos de la red y modelo uninodal

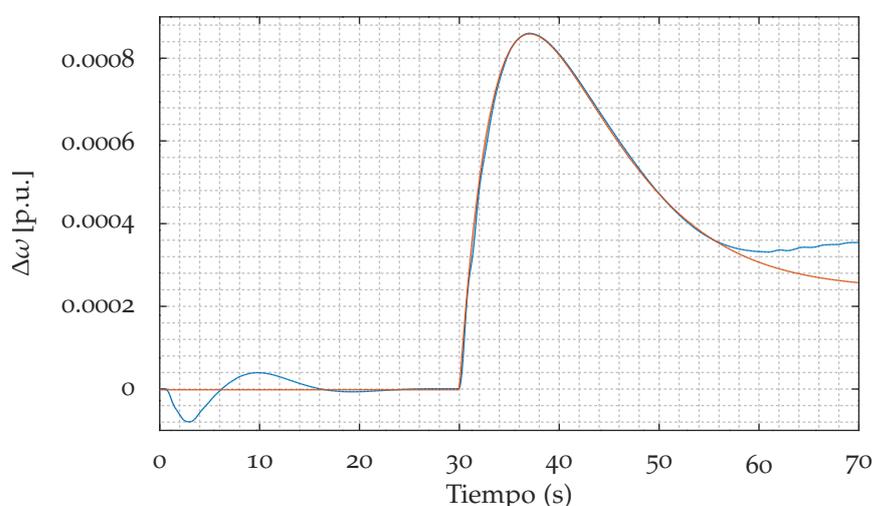


Figura 10.5: Curva fuente (azul) y mejor individuo uninodal obtenido (rojo)

La prueba se realizó en un escenario de potencia generada de 16 566 MW, siendo la falta una salida de carga de 1 337,4 MW (figura 10.5).

Para obtener el modelo uninodal equivalente, se corrieron 100 generaciones de 300 individuos usando los parámetros aleatorios obtenidos en la sección 10.1. Para evitar efectos de regulación posterior —visibles en la gráfica—, se ejecutó con un parámetro de tiempo máximo de 60 s (es decir, 30 s desde la falta).

10.2. Evaluación del modelo uninodal como equivalente dinámico de una red

La respuesta del mejor individuo obtenido se gráfica en la figura 10.5 junto a la curva original y tiene un *fitness* (medido entre el momento de falta y el tiempo máximo, es decir, $t = 60$ s en la figura) de 0,391 %.

Pruebas realizadas

La dimensión de los elementos individuales de la red es despreciable respecto a la dimensión del sistema, por tanto para las pruebas se ejecutaron salidas de distintos grupos de carga.

Grupos de carga en diversos puntos Se ejecutaron seis pruebas quitando diferentes grupos de carga en la red original (en distintas posiciones geográficas) y comparando las curvas resultantes con faltas de equivalente potencia en el modelo uninodal.

Diferentes potencias en mismo punto Se ejecutaron además cuatro pruebas de diferente escalón de potencia para el mismo grupo de carga que ocasionó la falta original, comparándose con pruebas a la misma potencia en el modelo uninodal.

Diferentes potencias en mismo punto sin conversión de cargas Se repite la prueba anterior utilizando en el modelo de la red uruguayo-argentina, esta vez usando cargas de potencia constante.

Resultados: grupos de carga en diversos puntos

Caso	Grupo	Cant. Elementos	Falta (MW)	Distancia (%)
1	22	49	641,41	3,95
2	23	47	1004,31	0,793
3	32	47	1337,36	1,01
4	42	54	983,41	1,28
5	52	41	752,80	3,74
6	947	11	44,86	4,45
7	950	28	254,04	8,56

Tabla 10.2: Distancia obtenida para salidas de distintos grupos de carga

Para cada grupo, se muestra en la tabla 10.2 su nombre en el modelo de la red, la potencia demandada por estos elementos (es decir, $-\Delta P$ para cada falta), y la distancia entre la curva obtenida entre la respuesta en la red original y la obtenida en el modelo uninodal. El caso 3 corresponde

Capítulo 10. Evaluaciones y resultados

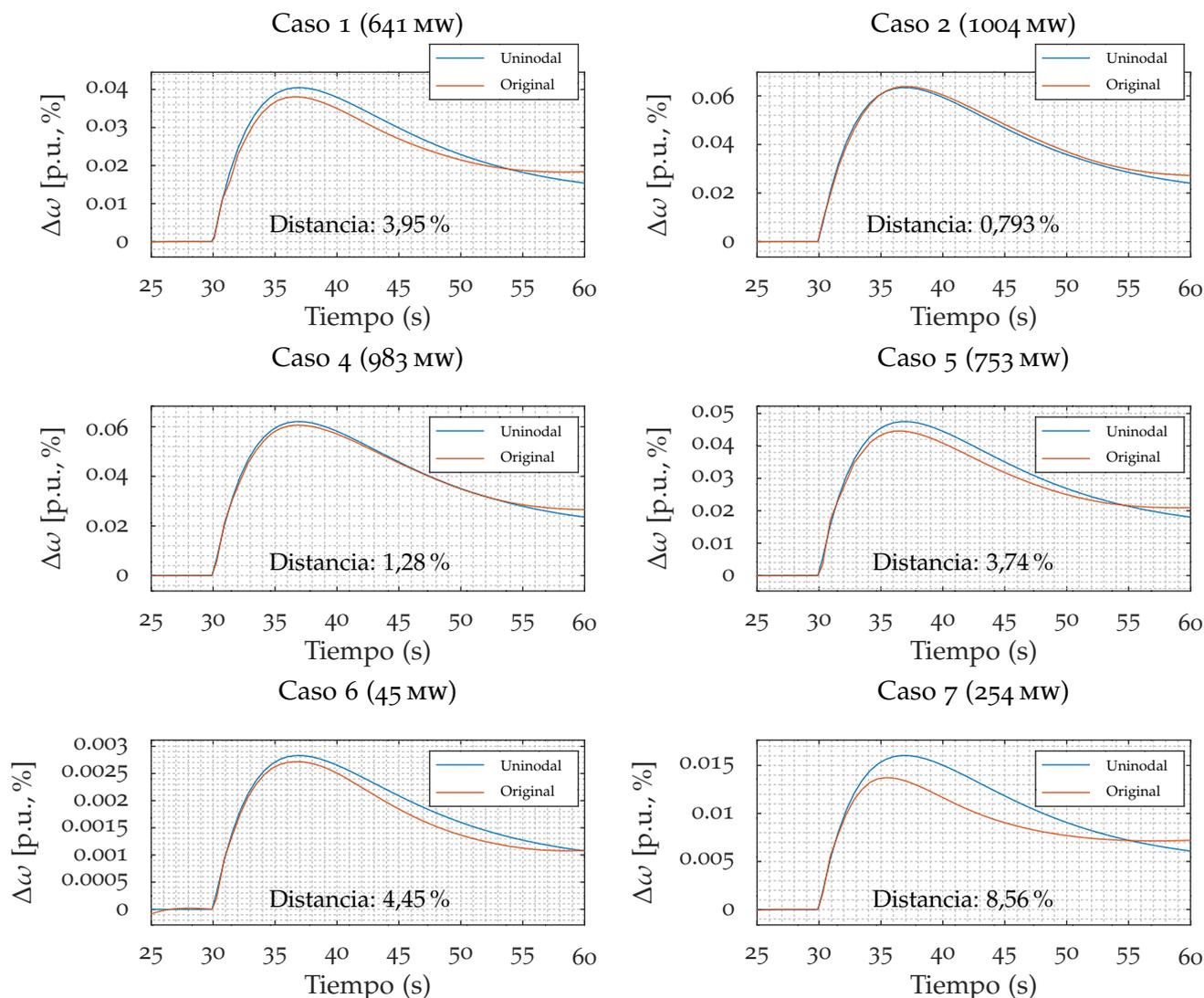


Figura 10.6: Comparación entre curvas de salida en modelo original (rojo) y uninodal (azul) para iguales faltas.

a la misma falta que originó la curva de referencia usada para hallar el modelo uninodal.

Las curvas de salida se grafican en la figura 10.6. Para la distancia entre curvas, siempre se compara entre el tiempo de falta y el límite de 60 s.

Al igual que en el caso de la red de pruebas "A", se observa que mientras más alejada la falta ejecutada de la falta original, mayor distancia hay entre las curvas.

10.2. Evaluación del modelo uninodal como equivalente dinámico de una red

Resultados: diferentes potencias en mismo punto

En la tabla 10.3 se muestran las pruebas realizadas variando la potencia del grupo 32, siendo la prueba 4 equivalente a la prueba original. Las curvas de salida se grafican en la figura 10.7.

Caso	Elementos	Falta (MW)	Distancia (%)
1	12	331,96	5,70
2	24	638,28	5,12
3	36	1023,15	3,10
4	47	1337,36	2,39

Tabla 10.3: Distancia obtenida para salidas de carga de distinta potencia

Resultados: diferentes potencias sin conversión de carga

Se realiza esta prueba para la red uruguayo-argentina sin la conversión de cargas; es decir, éstas se modelaron como cargas de potencia — activa y reactiva— constante. De esta forma se elimina el efecto que tiene la variación de tensión en el momento de introducir una falta en la simulación (ver discusión en sección 10.2.5).

En la figura 10.8 se muestran los resultados de la prueba para las mismas faltas presentes en la tabla 10.3.

10.2.5. Discusión

Cuando se introduce un escalón de potencia ΔP de las formas presentes en este estudio —deslastre de carga o generación en un punto particular de la red—, además del propio ΔP del elemento saliente también están presentes otros efectos que no entran en consideración en el modelo uninodal.

Algunos de estos efectos son no lineales: por ejemplo, la banda muerta. A continuación se describen algunos de estos efectos y sus consecuencias.

Efecto de la banda muerta sobre pequeños escalones

En un sistema real, en todo momento están presentes pequeñas variaciones de potencia tanto de generación como de demanda que provocarían la actuación constante de los reguladores de frecuencia ante cambios muy rápidos y pequeños de velocidad en las máquinas. Para evitar desgaste en las partes mecánicas de los reguladores ante estas variaciones, es

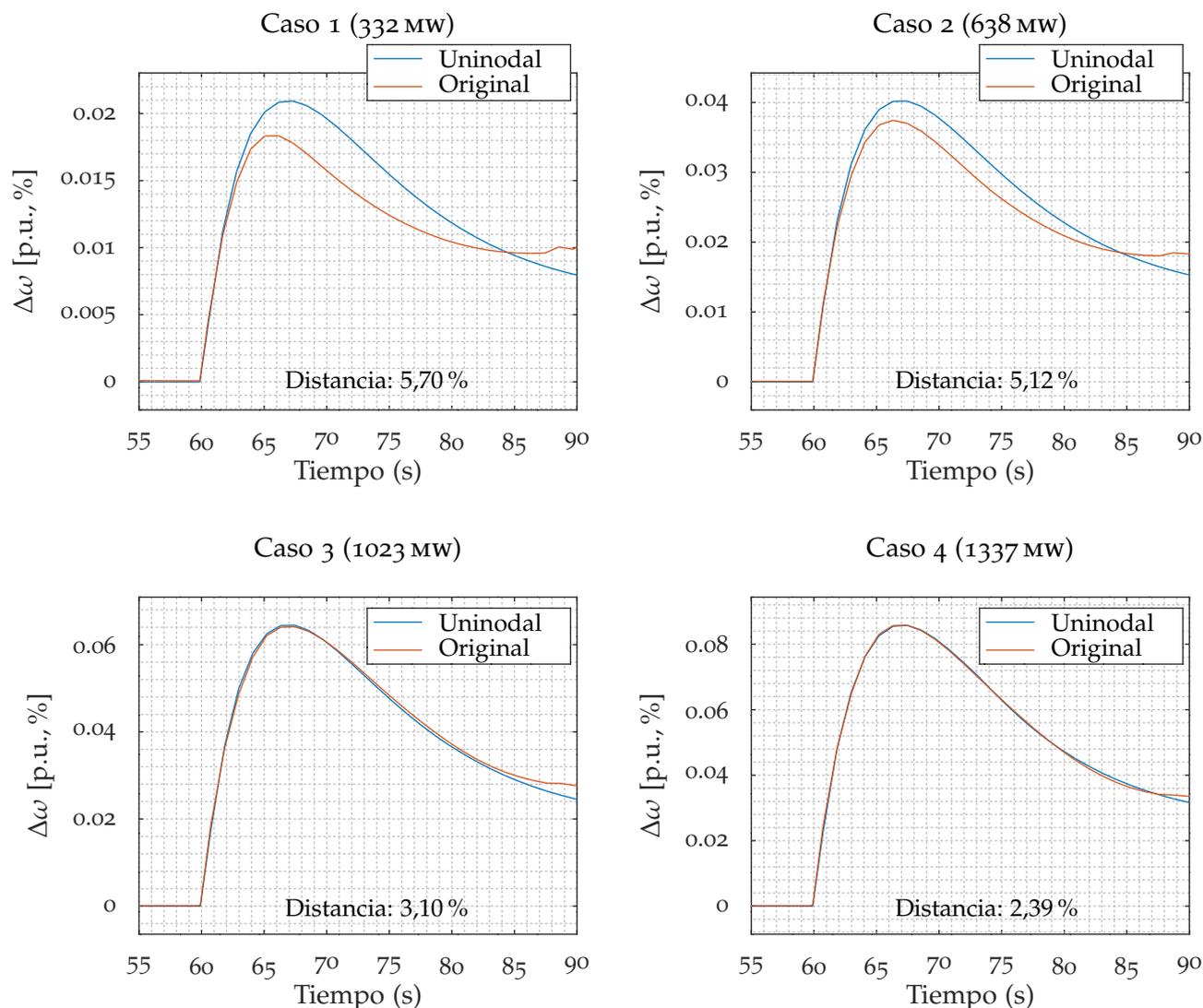


Figura 10.7: Comparación entre curvas de salida en modelo original (rojo) y uninodal (azul) para iguales faltas.

habitual que éstos cuenten con una *banda muerta* $\Delta\omega^{(DB)}$ para la frecuencia, definida de manera que mientras se cumpla $|\Delta\omega| < \Delta\omega^{(DB)}$ no actúe la regulación de frecuencia.

Esto se traduce en un retardo $\tau^{(DB)}$ en la realimentación desde el instante en que se produce la falta hasta el tiempo en que $|\Delta\omega|$ supera la zona de la banda muerta $\Delta\omega^{(DB)}$; en consecuencia, la respuesta del sistema hasta el tiempo $\tau^{(DB)}$ es tal que actúa como si no tuviera regulación —esto no afecta la derivada inicial de la respuesta, y por ende tampoco

10.2. Evaluación del modelo uninodal como equivalente dinámico de una red

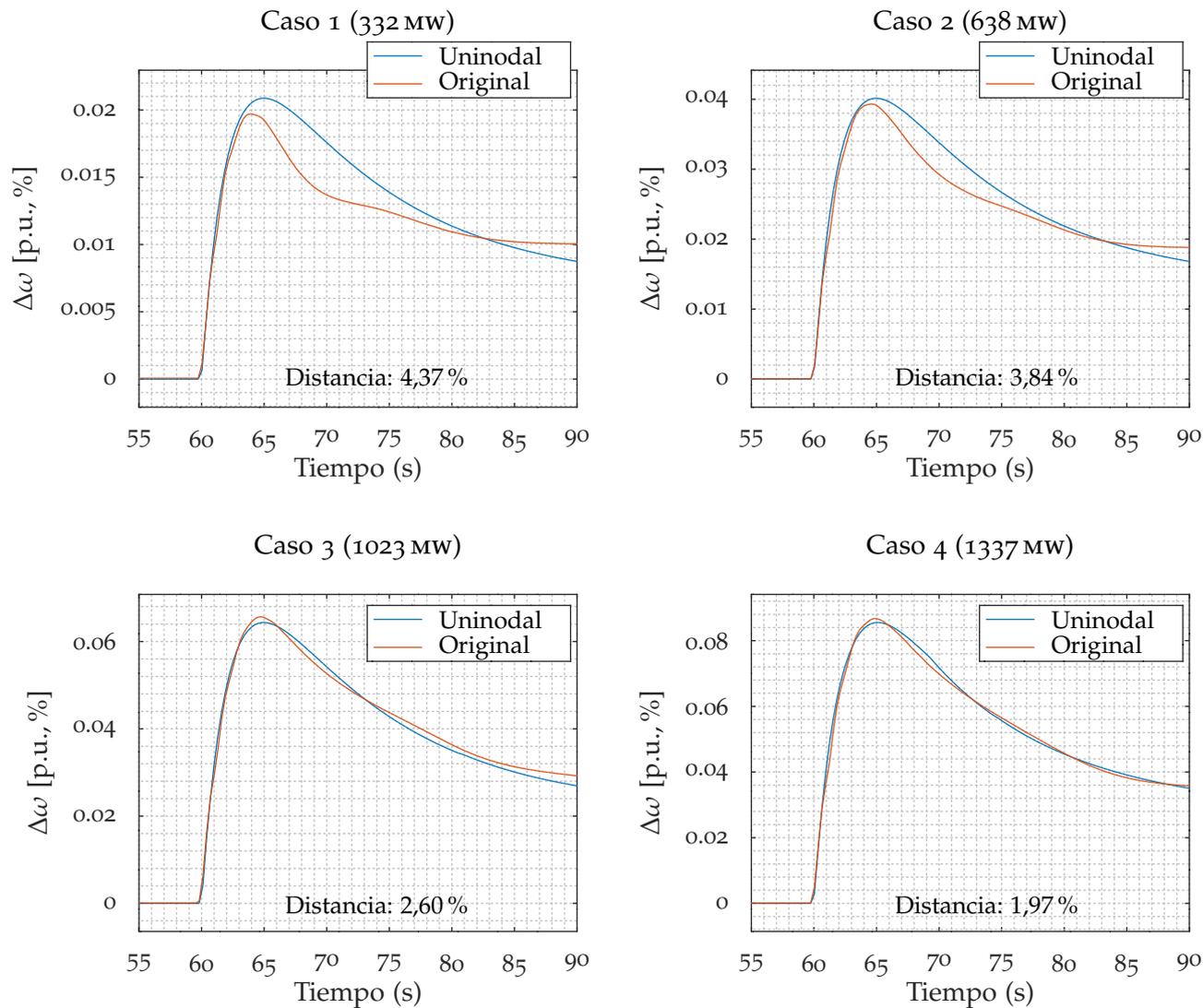


Figura 10.8: Pruebas ejecutadas sin conversión de carga; modelo original (rojo) y uninodal (azul) para iguales faltas.

el cálculo de H —.

Linealizando la derivada en el momento inicial,

$$\Delta\dot{\omega} = \frac{\Delta P}{2H} \Rightarrow \Delta\omega = \frac{\Delta P}{2H}t. \quad (10.1)$$

Si la regulación actúa a partir de $\Delta\omega^{(DB)} \simeq \frac{\Delta P}{2H}\tau^{(DB)}$, el tiempo de retardo resulta

$$\tau^{(DB)} = \frac{2H}{\Delta P}\Delta\omega^{(DB)}, \quad (10.2)$$

donde estamos asumiendo que a partir de cada $\Delta\omega_i^{(DB)}$ de cada regulador i se puede obtener una estimación para el sistema y encontrar un $\Delta\omega^{(DB)}$ equivalente.

La ecuación (10.2) indica que al ser más pequeño el ΔP de la falla, es mayor el tiempo en que no actúa el regulador. Esto resulta en que, para ΔP pequeños, el efecto no lineal sea más notorio.

Efecto de la tensión y prueba adicional

La salida de servicio de —por ejemplo— una carga resulta en una disminución también en la potencia reactiva demandada por el sistema, que provoca a su vez variaciones en las tensiones de las barras —principalmente— cercanas a la falta. Dado que para las cargas se utiliza un modelo ZIP, estas variaciones de tensión resultan en cambios en la demanda de potencia activa.

$$Q^{\text{dem}} \downarrow \Rightarrow V^{\text{prom}} \uparrow \Rightarrow P^{\text{dem}} \uparrow \quad (10.3)$$

Una falta localizada en una zona cambia la configuración de flujos de potencia en toda la red. Por tanto, el desbalance de potencia ΔP introducido inicialmente no es exactamente el desbalance «efectivo» que ve el sistema, sino que además se produce una variación de potencia activa debido a las variaciones de tensión, que no son fáciles de estimar. Este efecto se ve reflejado en la curva de respuesta de la frecuencia de referencia, y por tanto en el uninodal que se obtiene a partir de esa curva.

Se intenta reducir este efecto con la realización de una prueba en la cual no se hace la conversión de carga a un modelo polinomial. Los resultados pueden verse en la figura 10.8; comparándolos con la figura 10.7 se puede observar, en forma cualitativa, una mayor linealidad en esta prueba final.

10.2.6. Conclusiones

Se concluye que para obtener mayor precisión en la búsqueda de modelos uninodales que sean equivalentes dinámicos, en cuanto a la respuesta frecuencial de un SEP y en particular para «pequeños» desequilibrios, es necesario enriquecer dicho modelo introduciendo el efecto de banda muerta de los reguladores de frecuencia.

Modelar las cargas con un modelo de potencia constante mostró que es importante lograr estimar correctamente cual es el ΔP visto por el sistema cuando se produce una falta.

10.3. Evaluación del ajuste como semilla para la generación inicial

Respuesta inercial

El análisis cualitativo de las curvas en todas las pruebas muestra que, aún en los casos más patológicos, las salidas del modelo original y el uninodal sólo empiezan a alejarse en instantes cercanos al nadir y posteriores: durante la zona de respuesta inercial, el modelo uninodal tiene un comportamiento aproximado a la respuesta del modelo original.

Esto permite concluir que el modelo uninodal es una herramienta que permite caracterizar la respuesta inercial de forma satisfactoria, aún en casos donde la falta es diferente de la usada para originarlo.

10.3. Evaluación del ajuste como semilla para la generación inicial

10.3.1. Motivación

En el capítulo 4 se postula la relevancia de la calidad de las soluciones iniciales para asegurar la rápida convergencia hacia las soluciones, en el capítulo 5 se desarrolla un método para obtener información analítica sobre la solución buscada a partir de la curva de entrada, y en la sección 7.5.1 se detalla cómo a partir de esta información se generan soluciones iniciales.

Es de interés relevar cuánto ayudan estas soluciones iniciales a la convergencia del algoritmo, si es que lo hacen. Las soluciones están basadas en los parámetros de curva obtenidos en el ajuste, por lo que una forma posible de someter su influencia a prueba es proveer al algoritmo de números iniciales inexactos.

10.3.2. Método

Para una curva no oscilatoria se detectan —haciendo uso del identificador— los parámetros semilla $H^{(0)}$, $p^{(0)}$, $q^{(0)}$ y $r^{(0)}$, y se ejecuta un número de pruebas introduciendo un error en uno de ellos. Para cada prueba se miden el *fitness* final —mínimo— y el número de generación para el cual el algoritmo logró acercarse a la solución esperada. Se ejecuta una prueba sin variar ningún parámetro para conocer cuál es esta solución.

La prueba se repite luego para el caso no oscilatorio, ajustando $H^{(0)}$, $\phi^{(0)}$, $q^{(0)}$ y $\omega^{(0)}$.

Parámetros intermedios

Las razones por las que se escoje variar los parámetros intermedios en lugar de los parámetros del genoma H , D , T_g y R son:

- Existe cierta interdependencia entre los parámetros de la curva que hace que variar un parámetro de forma individual lleve a curvas de características muy diferentes de la original: por ejemplo, la locación de los polos depende no de H sino de la razón D/H ; variar sólo H no sólo alterará la derivada inicial —como puede esperarse— sino que también puede introducir o quitar características oscilatorias, mover los puntos clave, etcétera.

Los parámetros intermedios permiten alterar la curva de una forma más orgánica, alterando sólo una característica pero manteniendo la forma general de la curva.

- Los parámetros intermedios son de hecho los que se detectan en la identificación de curva, y a través de ellos se pasa a los parámetros finales de curva. Si la detección introduce un error en uno de ellos, este tipo de prueba arrojará con una precisión mayor el tipo de problemas que pueden verse.

La desventaja de este método es que el operador no maneja estos parámetros directamente: por ejemplo, al introducir estimaciones, lo hace para los parámetros finales.

10.3.3. Curvas de entrada y parámetros iniciales

Caso no oscilatorio

La curva de entrada es una de las curvas de prueba, `NoOscilatorio.txt`; para una falta de $\Delta P = -0,28$ (salida de carga) el ajuste ofrece:

$$\begin{array}{l|l} H^{(0)} & 9,773 \\ p^{(0)} & 0,7510 \\ q^{(0)} & 0,8167 \\ r^{(0)} & 0,7112 \\ A^{(0)} & 1,1 \times 10^{-4} \\ t_0^{(0)} & 10,46 \end{array} \left\| \begin{array}{l|l} D^{(0)} & 30,64 \\ T_g^{(0)} & 15,22 \\ R^{(0)} & 5,774\% \end{array} \right.$$

Caso oscilatorio

Se usa otra curva de prueba, `Oscilatorio.txt`; para una falta de $\Delta P = -0,28$ (salida de carga) el ajuste ofrece:

10.3. Evaluación del ajuste como semilla para la generación inicial

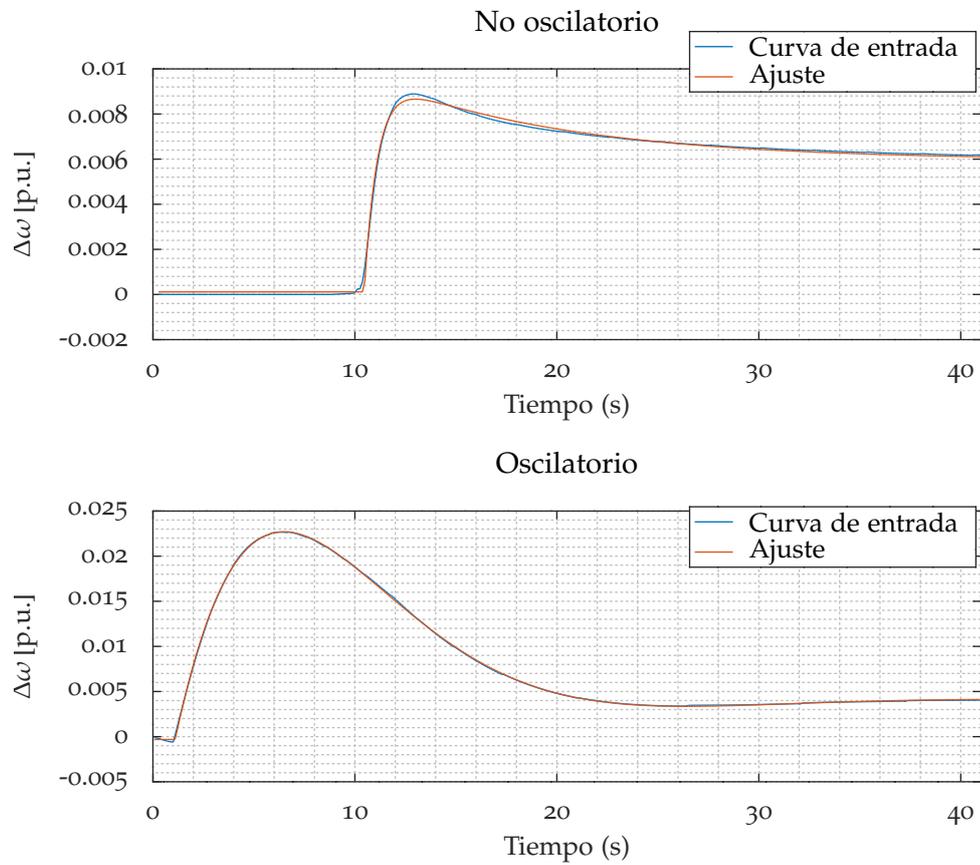


Figura 10.9: Curvas de entrada

$$\begin{array}{l|l}
 H^{(0)} & 13,25 \\
 \phi^{(0)} & 2,292 \\
 q^{(0)} & 0,1601 \\
 \omega^{(0)} & 0,1579 \\
 A^{(0)} & -3,1 \times 10^{-4} \\
 t_0^{(0)} & 1,105
 \end{array}
 \left\| \begin{array}{l|l}
 D^{(0)} & 7,919 \\
 T_g^{(0)} & 47,10 \\
 R^{(0)} & 1,813 \%
 \end{array} \right.$$

10.3.4. Parámetros que varían

Se hacen pruebas con diez valores por cada parámetro; cinco por encima y cinco por debajo del valor detectado. Algunos parámetros tienen restricciones en los valores; se intenta cubrir un rango representativo que no viole las mismas.

Capítulo 10. Evaluaciones y resultados

Caso no oscilatorio

H: Se toman los múltiplos $2H^{(0)}$, $10H^{(0)}$, $20H^{(0)}$, $100H^{(0)}$ y $1000H^{(0)}$; y los respectivos submúltiplos ($2^{-1}H^{(0)}$, $10^{-1}H^{(0)}$, etc.).

r: Debe cumplirse la restricción $r < p$. Para ello, para los valores superiores al detectado se toman $\frac{1}{k}(r^{(0)} + (k-1)p^{(0)})$ para $k = 2, 10, 20, 100$ y 1000 (mismos multiplicadores que los múltiplos de *H*). Esto tiene el efecto de producir un valor a un medio de distancia de $p^{(0)}$ con respecto a $r^{(0)}$, a un décimo de distancia, etc..

Para los valores inferiores se toman los mismos submúltiplos que para *H*.

p: Debe cumplirse la restricción $r < p < q$. Para ello se toman valores tanto inferiores como superiores con las fórmulas $\frac{1}{k}(p^{(0)} + (k-1)r^{(0)})$ y $\frac{1}{k}(p^{(0)} + (k-1)q^{(0)})$.

q: En principio $q > p$ es la única restricción; sin embargo, al obtener mayores valores *q*, el punto de inflexión se acerca al punto de nadir (ecuaciones (5.16) y (5.17)):

$$\lim_{q \rightarrow \infty} t_2 = \frac{1}{2r} \log \frac{p+r}{p-r} \frac{q+r}{q-r} = \frac{1}{2r} \log \frac{p+r}{p-r} = t_{\blacktriangledown},$$

y esto dificulta la detección para el algoritmo de sincronización.

Se opta por imponer un límite de $t_{\blacktriangledown} - t_0 < 2(t_2 - t_0)$; que luego de operar conduce a $q < \frac{p}{r}$ por lo que se toma como restricción $p < q < \frac{p}{r}$.

Se toman múltiplos y submúltiplos con la misma estrategia que con *p*.

Caso oscilatorio

H: Se toman los mismos múltiplos que en el caso no oscilatorio.

ϕ : El desfase debe ser mayor a $\frac{\pi}{2}$, además de que debe cumplirse $q + \omega \cot \phi > 0$, por lo que el rango válido para ϕ es $\frac{\pi}{2} < \phi < \text{arccot} \frac{-q^{(0)}}{\omega^{(0)}} = \frac{\pi}{2} + \arctan \frac{q^{(0)}}{\omega^{(0)}}$.

Por tanto para valores inferiores se toman los valores $\frac{1}{k}(\phi^{(0)} + (k-1)\frac{\pi}{2})$ y superiores $\frac{1}{k}(\phi^{(0)} + (k-1)(\frac{\pi}{2} + \arctan \frac{q^{(0)}}{\omega^{(0)}}))$.

10.3. Evaluación del ajuste como semilla para la generación inicial

q : Debe cumplirse $q > \omega \cot \phi$, por lo que para los valores inferiores se toman valores $\frac{1}{k}(q^{(0)} + (k-1)\omega^{(0)} \cot \phi^{(0)})$.

Nuevamente, a priori no hay límite teórico superior; sin embargo, si se atenúa demasiado rápidamente (q es para el caso oscilatorio la constante de atenuación), se dificulta la detección en el pico t_3 . Se opta entonces por imponer que desde el nadir hasta el pico haya una atenuación menor al 10%; dado que q es la constante de tiempo de esta atenuación, su inverso indica el tiempo en el que la atenuación será de aproximadamente $e^{-1} \approx 37\%$; por lo que se toma un valor máximo de 3 veces esta atenuación:

$$q < \frac{3}{(t_3 - t_{\blacktriangledown})}.$$

Ahora $t_3 - t_{\blacktriangledown} = \frac{\pi}{\omega}$, por lo que se llega a la restricción

$$q < \frac{3\omega}{\pi},$$

por lo que se toman valores $\frac{1}{k}(q^{(0)} + (k-1)\frac{3\omega^{(0)}}{\pi})$.

ω : Cumpliendo la misma restricción,

$$\omega < -\frac{q^{(0)}}{\cot \phi^{(0)}} = -q^{(0)} \tan \phi^{(0)}.$$

A priori no hay más restricciones; sin embargo un ω demasiado alto impone una granularidad mínima muy alta para la correcta detección; y uno bajo hace que no se pueda detectar t_3 de forma correcta.

Se opta por poner frecuencias mínima y máxima correspondientes a medio período de 2 s y 30 s respectivamente; que aseguran la convergencia para una granularidad de 200 puntos y un $t^{(\text{máx})}$ de 40 s. Para la máxima, se toma la menor de entre ésta y la anteriormente definida.

10.3.5. Resultados: caso no oscilatorio

Se hacen ocho corridas por cada valor de 100 individuos y 100 generaciones; y se obtienen para cada valor el mejor *fitness* alcanzado promedio, para qué generación se llegó a un *fitness* menor al 2%, y para cuál a uno menor al 0,603% en promedio.

Capítulo 10. Evaluaciones y resultados

Corrida con parámetros ajustados

Se ejecutan ocho corridas de 100 individuos y 100 generaciones con los parámetros ajustados para obtener un valor de referencia. Se obtiene un *fitness* promedio de $0,503 \pm 0,003 \%$.

A fin de comparar la rapidez de aproximación al *fitness* final de las corridas variando parámetros contra la referencia, se miden la generaciones para las cuales el *fitness* alcanza un valor menor al 2% y al 0,603% (0,1% de distancia del valor de referencia), además de la cantidad de corridas (un número de 0 a 8) para las cuales tales límites fueron alcanzados.

Los valores para la corrida de referencia se muestran en todas las tablas de las corridas para fácil comparación.

Parámetro H

El parámetro H inicial se usa para el cálculo de los parámetros D y R iniciales también (ecuaciones (5.18) y (5.20), por lo que la introducción de un error en H también los afecta.

Los resultados al introducir un parámetro inicial $H \neq H^{(0)}$, *ceteris paribus*, son:

$H^{(0)}: 9,773$						Gen. <2%		Gen. <0,6%	
$H/H^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,001	0,009773	0,03064	15,22	5774	$95,4 \pm 0,5$	0	—	0	—
0,01	0,09773	0,3064	15,22	577,4	$14,5 \pm 28,0$	0	—	0	—
0,05	0,4887	1,532	15,22	115,5	$4,8 \pm 1,7$	0	—	0	—
0,1	0,9773	3,064	15,22	57,74	$5,1 \pm 1,5$	0	—	0	—
0,5	4,887	15,32	15,22	11,55	$1,7 \pm 0,6$	6	$82,3 \pm 12,8$	0	—
1	9,773	30,64	15,22	5,774	$0,5 \pm 0,0$	8	$1,0 \pm 0,0$	8	$5,9 \pm 5,6$
2	19,55	61,29	15,22	2,887	$1,7 \pm 0,3$	7	$81,1 \pm 15,3$	0	—
10	97,73	306,4	15,22	0,5774	$62,9 \pm 1,6$	0	—	0	—
20	195,5	612,9	15,22	0,2887	$79,8 \pm 2,8$	0	—	0	—
100	977,3	3064	15,22	0,05774	$93,0 \pm 0,5$	0	—	0	—
1000	9773	$3,064 \times 10^4$	15,22	0,005774	$96,3 \pm 0,0$	0	—	0	—

Parámetro p

Tanto p como q determinan el cálculo de los parámetros D , T_g y R (ecuaciones (5.18), (5.19) y (5.20)), por lo que el único que no se ve afectado es H .

10.3. Evaluación del ajuste como semilla para la generación inicial

$p^{(0)}: 0,751$						Gen. <2%		Gen. <0,6%	
$p/p^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,9471	9,773	29,87	9,484	9528	4,5±0,0	0	—	0	—
0,9475	9,773	29,87	9,516	949,3	2,5±0,3	0	—	0	—
0,9497	9,773	29,9	9,662	186,8	1,0±0,3	8	48,8±23,4	1	94
0,9523	9,773	29,94	9,852	91,47	1,3±0,4	8	45,2±26,7	0	—
0,9735	9,773	30,25	11,68	15,25	0,7±0,3	8	14,8±15,1	4	38,2±33,6
1	9,773	30,64	15,22	5,774	0,5±0,0	8	1,0±0,0	8	5,9±5,6
1,044	9,773	31,29	30,45	1,547	1,2±0,1	8	4,9±2,7	0	—
1,079	9,773	31,8	152,2	0,2233	1,7±0,1	8	12,5±11,4	0	—
1,083	9,773	31,86	304,5	0,1078	1,8±0,1	8	30,1±24,5	0	—
1,087	9,773	31,92	1522	0,020 99	2,0±0,4	6	22,8±20,4	0	—
1,087	9,773	31,93	1,522×10 ⁴	0,002 086	3,4±0,3	0	—	0	—

Parámetro q

$q^{(0)}: 0,8167$						Gen. <2%		Gen. <0,6%	
$q/q^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,9197	9,773	29,36	1,522×10 ⁴	0,005 774	4,5±0,8	0	—	0	—
0,9204	9,773	29,37	1522	0,057 74	4,0±1,5	1	50	0	—
0,9236	9,773	29,42	304,5	0,2887	2,3±0,7	4	64,8±8,3	0	—
0,9276	9,773	29,49	152,2	0,5774	1,8±0,1	7	51,9±26,0	0	—
0,9598	9,773	30	30,45	2,887	1,4±0,1	8	6,8±4,5	0	—
1	9,773	30,64	15,22	5,774	0,5±0,0	8	1,0±0,0	8	5,9±5,6
1,146	9,773	32,98	5,396	16,29	1,1±0,3	8	28,0±9,6	1	97
1,264	9,773	34,85	3,558	24,7	1,3±0,3	8	61,1±19,0	0	—
1,278	9,773	35,09	3,413	25,75	1,4±0,4	8	64,5±15,6	0	—
1,29	9,773	35,27	3,305	26,6	1,4±0,3	8	66,1±8,1	0	—
1,293	9,773	35,32	3,282	26,79	1,3±0,3	8	50,9±10,5	0	—

Capítulo 10. Evaluaciones y resultados

Parámetro r

El parámetro r sólo aparece en los cálculos para R , por tanto es el único al que afecta este cambio.

$r^{(0)}: 0,7112$						Gen. <2%		Gen. <0,6%	
$r/r^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,001	9,773	30,64	15,22	0,5958	0,5±0,0	8	5,1±1,5	8	17,2±7,1
0,01	9,773	30,64	15,22	0,5958	0,5±0,0	8	5,6±1,1	7	22,4±17,2
0,05	9,773	30,64	15,22	0,5971	0,5±0,0	8	5,1±1,5	8	34,8±23,9
0,1	9,773	30,64	15,22	0,6012	0,5±0,0	8	3,5±1,2	8	19,4±12,2
0,5	9,773	30,64	15,22	0,768	0,5±0,0	8	4,9±1,1	8	30,9±19,9
1	9,773	30,64	15,22	5,774	0,5±0,0	8	1,0±0,0	8	5,9±5,6
1,028	9,773	30,64	15,22	11,39	0,9±0,2	8	2,6±1,5	1	39
1,05	9,773	30,64	15,22	56,36	1,1±0,5	7	17,6±12,1	1	20
1,053	9,773	30,64	15,22	112,6	1,0±0,4	8	25,8±22,8	3	56,3±27,9
1,055	9,773	30,64	15,22	562,2	1,8±0,3	7	84,9±7,2	0	—
1,056	9,773	30,64	15,22	5621	4,5±0,0	0	—	0	—

10.3.6. Resultados: caso oscilatorio

Se repite la misma cantidad de pruebas que para el caso no oscilatorio; mostrando para qué generación se llegó a un *fitness* menor al 2%, y para cuál a uno menor al 0,535% en promedio.

Corrida con parámetros ajustados

Se ejecutan ocho corridas de 100 individuos y 100 generaciones con los parámetros ajustados para obtener un valor de referencia. Se obtiene un *fitness* promedio de $0,435 \pm 0,002$.

Nuevamente se miden la generaciones para las cuales el *fitness* alcanza un valor menor al 2% y, para este caso, al 0,535% (0,1% de distancia del valor de referencia), y nuevamente la cantidad de corridas para las cuales tales límites fueron alcanzados.

Corridas variando parámetros

Nuevamente se hacen ocho corridas por cada valor de 100 individuos y 100 generaciones.

10.3. Evaluación del ajuste como semilla para la generación inicial

Parámetro H

Valen las mismas consideraciones que para el caso no oscilatorio; el parámetro H afecta a los parámetros D y R también (ecuaciones (5.28) y (5.30)).

$H^{(0)}: 13,25$						Gen. <2 %		Gen. <0,53 %	
$H/H^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,001	0,013 25	0,007 919	47,1	18,13	89,2±2,7	o	—	o	—
0,01	0,1325	0,079 19	47,1	1,813	3,5±0,6	o	—	o	—
0,05	0,6624	0,3959	47,1	0,3627	3,6±0,6	o	—	o	—
0,1	1,325	0,7919	47,1	0,1813	3,5±0,4	o	—	o	—
0,5	6,624	3,959	47,1	0,036 27	1,7±0,7	6	52,8±13,5	o	—
1	13,25	7,919	47,1	0,018 13	0,4±0,0	8	1,0±0,0	8	5,6±2,3
2	26,49	15,84	47,1	0,009 067	3,0±3,5	5	68,0±13,3	o	—
10	132,5	79,19	47,1	0,001 813	33,7±2,4	o	—	o	—
20	264,9	158,4	47,1	0,000 906 7	52,3±2,4	o	—	o	—
100	1325	791,9	47,1	0,000 181 3	84,2±0,7	o	—	o	—
1000	1,325×10 ⁴	7919	47,1	1,813×10 ⁻⁵	91,6±0,1	o	—	o	—

Parámetro ϕ

Tanto ϕ como q y ω determinan el cálculo de los parámetros D , T_g y R (ecuaciones (5.28), (5.29) y (5.30)), por lo que el único que no se ve afectado es H .

$\phi^{(0)}: 2,292$						Gen. <2 %		Gen. <0,53 %	
$\phi/\phi^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,6856	13,25	4,244	6,252	0,2422	10,4±1,3	o	—	o	—
0,6885	13,25	4,271	6,293	0,2407	12,0±1,0	o	—	o	—
0,701	13,25	4,391	6,479	0,2335	10,5±1,1	o	—	o	—
0,7168	13,25	4,543	6,727	0,224	10,7±1,0	o	—	o	—
0,8427	13,25	5,818	9,949	0,1333	10,1±0,9	o	—	o	—
1	13,25	7,919	47,1	0,018 13	0,4±0,0	8	1,0±0,0	8	5,6±2,3
1,015	13,25	8,19	91,14	0,008 784	3,9±0,2	o	—	o	—
1,028	13,25	8,421	443,2	0,001 71	7,6±0,7	o	—	o	—
1,029	13,25	8,451	883	0,000 851 9	8,3±1,3	o	—	o	—
1,031	13,25	8,475	4396	0,000 170 2	9,0±0,9	o	—	o	—
1,031	13,25	8,48	4,32×10 ⁴	1,729×10 ⁻⁵	10,0±1,0	o	—	o	—

Capítulo 10. Evaluaciones y resultados

Parámetro q

$q^{(0)}: 0,1601$						Gen. <2%		Gen. <0,53%	
$q/q^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,8675	13,25	7,357	$4,71 \times 10^4$	$1,813 \times 10^{-5}$	$8,9 \pm 0,5$	o	—	o	—
0,8687	13,25	7,362	4710	0,000 181 3	$8,3 \pm 0,5$	o	—	o	—
0,874	13,25	7,384	942	0,000 906 7	$8,2 \pm 1,0$	o	—	o	—
0,8806	13,25	7,412	471	0,001 813	$7,2 \pm 0,2$	o	—	o	—
0,9337	13,25	7,637	94,2	0,009 067	$3,9 \pm 0,2$	o	—	o	—
1	13,25	7,919	47,1	0,018 13	$0,4 \pm 0,0$	8	$1,0 \pm 0,0$	8	$5,6 \pm 2,3$
1,145	13,25	8,535	22,47	0,038 02	$5,1 \pm 0,8$	o	—	o	—
1,262	13,25	9,029	15,84	0,053 92	$8,1 \pm 1,0$	o	—	o	—
1,276	13,25	9,09	15,28	0,055 91	$8,7 \pm 0,7$	o	—	o	—
1,288	13,25	9,14	14,85	0,0575	$8,7 \pm 1,0$	o	—	o	—
1,291	13,25	9,151	14,76	0,057 86	$9,0 \pm 1,2$	o	—	o	—

Parámetro ω

$\omega^{(0)}: 0,1579$						Gen. <2%		Gen. <0,53%	
$\omega/\omega^{(0)}$	H	D	T_g	R (%)	$Fit.$ (%)	N	No. gen	N	No. gen
0,6637	13,25	6,682	14,72	0,1317	$9,1 \pm 0,7$	o	—	o	—
0,6667	13,25	6,693	14,82	0,1297	$8,2 \pm 1,4$	o	—	o	—
0,6802	13,25	6,742	15,24	0,1211	$8,3 \pm 1,4$	o	—	o	—
0,697	13,25	6,804	15,8	0,1113	$8,0 \pm 1,1$	o	—	o	—
0,8317	13,25	7,3	22,42	0,055 07	$5,8 \pm 0,9$	o	—	o	—
1	13,25	7,919	47,1	0,018 13	$0,4 \pm 0,0$	8	$1,0 \pm 0,0$	8	$5,6 \pm 2,3$
1,076	13,25	8,2	94,2	0,007 825	$4,3 \pm 0,2$	o	—	o	—
1,138	13,25	8,425	471	0,001 401	$8,6 \pm 1,4$	o	—	o	—
1,145	13,25	8,453	942	0,000 691 2	$9,8 \pm 2,1$	o	—	o	—
1,151	13,25	8,475	4710	0,000 136 8	$11,2 \pm 1,8$	o	—	o	—
1,153	13,25	8,481	$4,71 \times 10^4$	$1,365 \times 10^{-5}$	$13,9 \pm 2,1$	o	—	o	—

10.3.7. Discusión

En todos los casos de prueba variando parámetros se observa un *fitness* final promedio mayor (menos óptimo) que en el caso de referencia. Esto se da para ambos modelos de ajuste, oscilatorio y no oscilatorio.

Excepto en excepciones puntuales, además, se aprecia que un mayor alejamiento del valor de referencia produce un resultado más alejado del óptimo, lo cual apunta a que el corrimiento del valor de referencia tiene un efecto *continuo* —contrastado a un efecto binario o *discreto* donde, por ejemplo, luego de introducido un error no habría mayor variación en los resultados—.

La conclusión inmediata a extraer es que el correcto ajuste de parámetros, generando soluciones iniciales con buena calidad genética, ayuda a

10.3. Evaluación del ajuste como semilla para la generación inicial

la convergencia y permite para un número moderado de generaciones e individuos obtener resultados de mejor calidad.

Si bien el *fitness* medio ya habla de la ventaja de una generación inicial próxima al ajuste, los resultados del número de generaciones para la convergencia al 2 % y al 0,1 % del valor de referencia muestran que un valor próximo al de referencia acelera la convergencia.

Para la convergencia al 2 %, en muchos casos se observa nuevamente el comportamiento continuo: al alejarse más del valor de referencia, hay una entrada más lenta a la zona de *fitness* menor —notoriamente en los parámetros p y q de la curva no oscilatoria—. En la gran mayoría de los casos se observa que un alejamiento extremo resulta en la no convergencia al 2 % antes de la generación 100.

Sólo en un caso de entre los probados (parámetro $p = 1,044 \times p^{(0)}$ en el tipo no oscilatorio) se observa un caso en el cual hay una entrada más rápida a la zona de 2 % que el promedio del caso de referencia. Sin embargo, *a*) es un sólo caso convergente, contra cuatro de la referencia, y *b*) el valor queda incluido dentro de la desviación estándar de la media de la referencia; se concluye que es un caso fortuito aislado.

Para el caso de la convergencia al 0,1 % del valor de referencia, se observa que sólo para un parámetro (r en el tipo no oscilatorio, ver discusión en la siguiente sección) hay un acercamiento antes de la generación 100: en los otros casos o bien no se llega para ningún caso, o sólo se lo hace para los desplazamientos más pequeños.

Como conclusión general, un buen ajuste inicial es pilar fundamental de una convergencia rápida; si bien hay un eventual acercamiento, el ahorro en generaciones hace esencial la buena detección de parámetros.

Caso no oscilatorio

El análisis del efecto de desplazar cada parámetro por separado permite extraer conclusiones específicas relativas a cada uno de ellos, y la importancia de su correcta detección como conclusión global.

H: Dado que H afecta el cálculo también de D y R , era esperable ver que la introducción de un error en su detección podía tener graves consecuencias para la convergencia. En efecto, un alejamiento dentro del orden $(H^{(0)}/10 < H < H^{(0)} \times 10)$ tiene como resultado una entrada a la zona de *fitness* de 2 % sólo luego de la generación 80 en sólo dos corridas de las ocho probadas, mientras que para el valor de referencia se entra en tal zona en la generación (promedio) 5,5. Para alejamientos más allá de es orden no se vuelve a entrar, y para alejamientos catastróficos no se logra un *fitness* inferior al 90 %.

Bajo esta luz es notoria la importancia del ajuste para H desarrollado en la sección 5.7.1. Si se tomase la derivada de la primer perturbación como referencia para H y t_0 ; el valor para H podría estar completamente fuera de sintonía con el valor «real» necesario para garantizar la convergencia del ajuste.

p y q : Tanto en cuanto a *fitness* final como a velocidad de convergencia, puede verse que la introducción de errores no tiene un resultado catastrófico, como sí fue posible ver en H . Sin embargo, debe tenerse en cuenta que la variación de los mismos fue en los rangos determinados por las restricciones desarrolladas en la sección 10.3.4, lo cual puede tener un efecto amortiguatorio sobre los resultados.

A pesar de esto, es posible notar una tendencia a una convergencia más lenta cuando se introduce un error en cualquiera de ellos.

r : Tiene la mayor tolerancia a la introducción de un error de entre las pruebas realizadas.

Sin embargo, es interesante notar que el comportamiento es asimétrico: cuando $r \rightarrow 0$ —es decir, cuando los polos se acercan entre ellos—, no se introducen mayores problemas en la convergencia y es posible una convergencia relativamente rápida; mientras que cuando $r \rightarrow p$ el error tiende a afectar de forma marcada la convergencia, como en los otros parámetros.

Puede concluirse que sería posible en el ajuste no oscilatorio, a falta de mejores estimaciones, comenzar con $r \approx 0$ ($r \ll p$).

Caso oscilatorio

H : Las observaciones se corresponden con las hechas en el caso no oscilatorio.

ϕ , q y ω : Para todos los parámetros se observa un comportamiento similar: si bien el *fitness* promedio final se mantiene lejos de los valores catastróficos vistos para H , pequeños desplazamientos ya retrasan la entrada a la zona de 2% más allá de la centésima generación, cuando para el caso de referencia se ve una entrada extremadamente rápida (generación 2,8 en promedio).

10.4. Distribución de potencia en generadores y parámetros del individuo

10.4. Distribución de potencia en generadores y parámetros del individuo

Según visto en la sección 3.3, se puede afirmar que un generador síncrono aporta su inercia H y amortiguamiento D a la red, independientemente de la potencia que se encuentre inyectando a la misma.

Se busca comparar el efecto que tiene para un mismo *individuo* modificar las potencias generadas por cada generador. De corroborarse que esto no tiene efecto sobre los resultados, se podrá ratificar la decisión de no incluirlas en el genoma.

10.4.1. Prueba

La prueba se realiza para el resultado de una ejecución de la herramienta. Se realizan 3 simulaciones de desconexión de carga de 280 MW, en un escenario con 2 167 MW de potencia demandada. En cada una de ellas se mantienen fijos los parámetros de individuo —en otras palabras, el individuo es el mismo— y se cambia la distribución de potencia de generación en cada generador presente en el modelo, sin modificar el escenario inicial de generación total de 2 167 MW.

Se elige el caso de uso 2 dado que se cuenta con la mayor variedad de fuentes y tipos de generación, incluyendo ERNC.

La distribución de generación para cada simulación se muestra en la tabla 10.4.

	Simulación 1 (MW)	Simulación 2 (MW)	Simulación 3 (MW)
G1	541,82	2167,27	400,0
G2	541,82	0	0
G5	541,82	0	1767,27
G6	541,82	0	0

Tabla 10.4

Dado que se simula una salida de carga, el sistema debe ser capaz de disminuir la potencia generada mediante regulación de frecuencia —para lograr la comparación que se quiere hacer—. Es por eso que el generador G1 —que representa la generación convencional con de regulación de frecuencia— tiene una potencia generada no nula en la simulación 3 (ver sección 3.4).¹

¹Si G1 tuviese P nula, la regulación no tendría margen para disminuir la potencia mecánica que entrega la turbina al generador.

10.4.2. Resultados y discusión

En la imagen 10.10 se comparan las curvas obtenidas como resultados de las simulaciones.

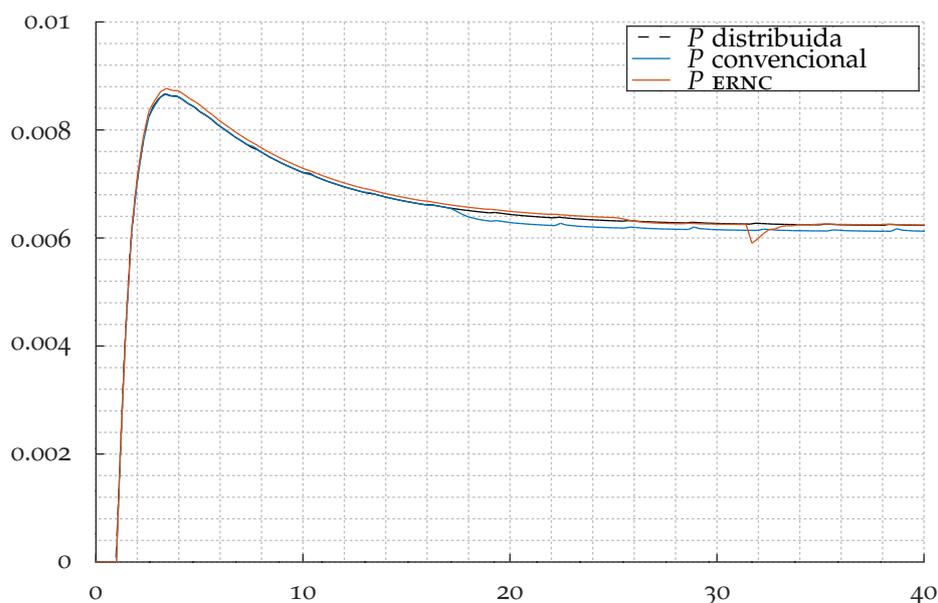


Figura 10.10: Comparación de las curvas de cada simulación

Se pueden observar ligeras diferencias entre las simulaciones. En particular, cuando las ERNC representan la mayor parte de la generación, se observa un mayor valor máximo de frecuencia en el nadir.

En líneas generales, se concluye que si bien existe una dependencia con la potencia en la respuesta de la frecuencia, ésta no es determinante, dado que las variaciones realizadas en P_i entre una simulación y otra —considerables, pasando de 100 % de generación convencional a 82 % de generación ERNC— apenas afectaron la respuesta.

10.5. Dimensión del genoma y convergencia

El objetivo es evaluar el desempeño del AA.EE. para cada caso de uso. En particular, encontrar la relación entre la velocidad de convergencia y el tamaño del genoma.

El fin es determinar el costo que podría tener la incorporación de nuevos elementos al modelo uninodal en aplicaciones de estudio futuras:

10.5. Dimensión del genoma y convergencia

esto implicaría nuevos modelos de generador y por ende nuevas variables candidatas a formar parte del genoma.

Por otro lado, con base en lo visto en la sección 3.5.2, se busca comprobar que es innecesaria la presencia de más de un generador convencional en el modelo uninodal para encontrar el equivalente del sistema.

10.5.1. Metodología

Se realizan pruebas para dos curvas de referencia —oscilatoria y no oscilatoria—. Cada prueba consiste en obtener un modelo uninodal para los tres casos de uso mediante la ejecución de la herramienta, usando una de las curvas de referencia.

Se realizaron ejecuciones de 100 generaciones de 100 individuos utilizando las variables de exploración sugeridas en 10.1. Para cada caso de uso se realizan ocho pruebas para la curva de referencia oscilatoria y ocho para la no oscilatoria.

10.5.2. Resultados y discusión

Interesa conocer la calidad y la velocidad de convergencia en función del caso de uso, por lo que para cada caso se tabulan:

- el mejor *fitness* (mínimo);
- el *fitness* medio;
- la cantidad de veces y la generación media para la cual se alcanza un *fitness* de 0,5 %;
- la cantidad de veces y la generación media para la cual se alcanza un *fitness* cercano al mejor caso —0,475 %, elegido de forma arbitraria—, como medida de las diferencias en la convergencia.

Resultados en curva no oscilatoria

Se muestran los resultados obtenidos en la tabla 10.5 para las pruebas realizadas con la curva no oscilatoria.

Caso	Fit. mín (%)	Fit. med. (%)	Gen. <0,5 %		Gen. <0,475 %	
			N	No. gen.	N	No. gen.
1	0,472	0,474±0,002	8	5,0±4,2	6	31,3±19,3
2	0,412	0,452±0,016	8	7,5±3,7	8	17,1±11,1
3	0,397	0,409±0,007	8	1,2±0,4	8	1,8±0,7

Tabla 10.5: Resultados de prueba de casos de uso en referencia no oscilatoria

Capítulo 10. Evaluaciones y resultados

En la figura 10.11 se muestra el mejor modelo uninodal para cada caso de uso obtenido para la curva de referencia no oscilatoria.

En primera instancia, a partir de la tabla 10.5 parece ser que el caso de uso 3 es la opción que mejor aproxima a la referencia. Sin embargo, observando la figura 10.11 se ve cualitativamente que las diferencias son casi imperceptibles.

Resultados en curva oscilatoria

En la tabla 10.6 se muestran los resultados para las pruebas realizadas con la curva oscilatoria.

Caso	Fit. mín (%)	Fit. med. (%)	Gen. <0,5 %		Gen. <0,435 %	
			N	No. gen.	N	No. gen.
1	0,386	0,392±0,012	8	6,4±1,4	8	11,6±9,6
2	0,368	0,377±0,010	8	6,1±2,0	8	8,9±3,3
3	0,322	0,325±0,003	8	6,1±1,7	8	7,8±2,1

Tabla 10.6: Resultados de prueba de casos de uso en referencia oscilatoria

En la figura 10.12 se muestran, en este caso, los peores resultados obtenidos para cada caso de uso siendo el *fitness* obtenido:

- caso de uso 1: 0,422 %;
- caso de uso 2: 0,402 %;
- caso de uso 3: 0,332 %.

Nuevamente se ve que, aún comparando los peores casos, las desviaciones están dentro de lo aceptable.

Inercia total y análisis de las soluciones

Según lo visto en el capítulo 3, para un SEP existe un modelo con parámetros equivalentes, por lo que para el modelo uninodal debe ser posible hallar estos parámetros. A continuación se hará énfasis en el cálculo de la inercia total del sistema, siendo el análisis para los demás parámetros análogo.

En las tablas 10.7 se presenta el promedio del cálculo de inercia total, para cada prueba con las curvas de referencia no oscilatoria y oscilatoria.

10.5. Dimensión del genoma y convergencia

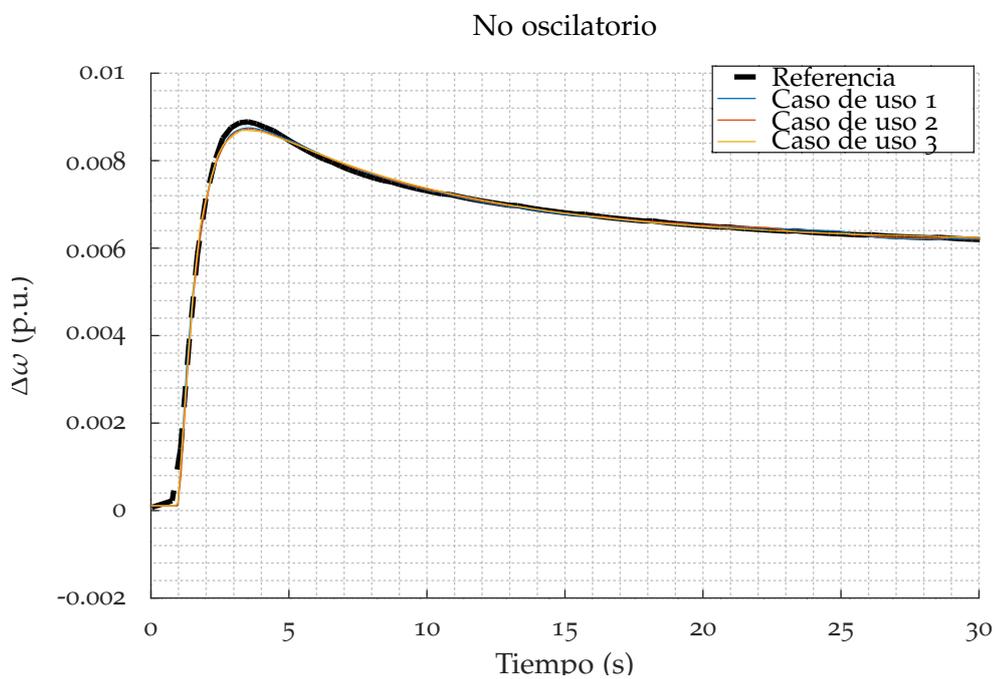


Figura 10.11: Curvas comparativas de mejores resultados para caso no oscilatorio

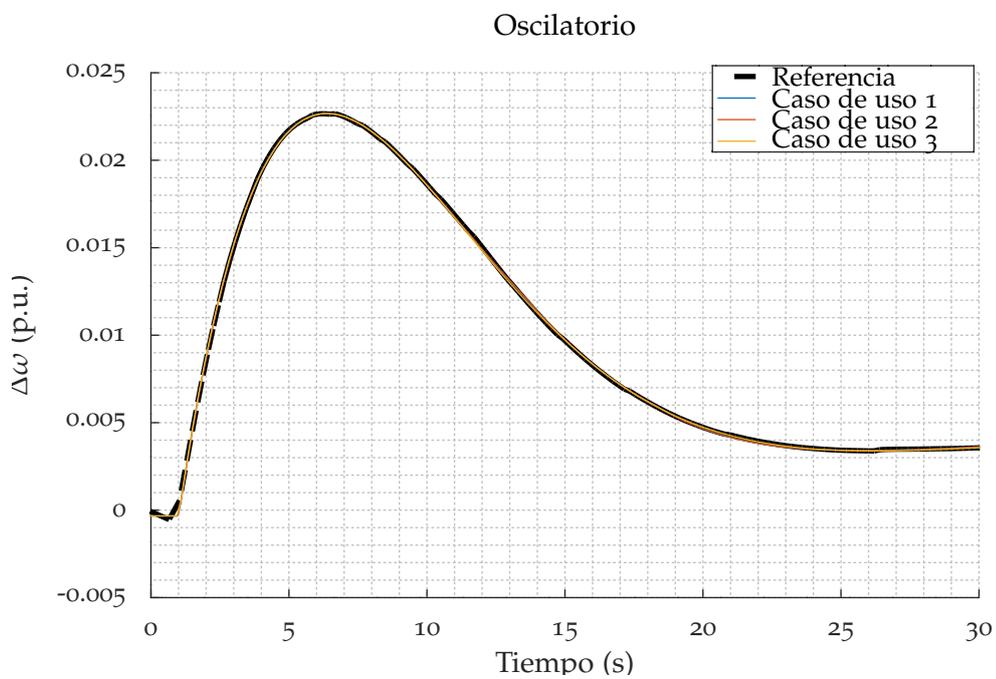


Figura 10.12: Curvas comparativas de peores resultados para caso oscilatorio

Capítulo 10. Evaluaciones y resultados

Caso	No oscilatorio H_{total} (s)	Oscilatorio H_{total} (s)	Cálculo de H
1	10,055±0,084	12,614±0,063	$H_{total} = H_1$
2	10,110±0,130	12,600±0,064	$H_{total} = H_1 + H_2$
3	10,011±0,082	12,922±0,078	$H_{total} = H_1 + H_2 + H_3 + H_4$
Media	10,062±0,121	12,712±0,288	

Tabla 10.7: Calculo de inercia total en modelo uninodal

Se comprueba así que la inercia total o equivalente tiende a un valor cercano para cada caso de uso.

En las figuras 10.13 y 10.14, se muestra para los casos de uso 2 y 3 respectivamente, cómo quedó distribuida la inercia en cada prueba.

Se puede ver en las figuras que la solución óptima se alcanza en todos los casos con una inercia total aproximada H_{total} , sin importar la distribución particular de inercia entre las máquinas.

Se presentan en las tablas 10.8 y 10.9 el cálculo de la media de los demás parámetros equivalentes, para cada prueba con las curvas de referencia no oscilatoria y oscilatoria respectivamente.

Caso	$D_{eq} (\sum_i D_i)$	$T_{g,eq} (\ _i \frac{R_i}{R_{eq}} T_{g,i})$	$R_{eq} (\ _i R_i)$
1	29,214±0,244	10,62±0,21	6,615±0,095
2	29,058±0,381	10,16±0,51	6,704±0,076
3	30,565±0,176	12,23±0,22	6,364±0,067
Media	29,612±1,056	11,00±1,44	6,552±0,282

Tabla 10.8: Cálculo promedio por caso de uso de los parámetros equivalentes

Caso	$D_{eq} (\sum_i D_i)$	$T_{g,eq} (\ _i \frac{R_i}{R_{eq}} T_{g,i})$	$R_{eq} (\ _i R_i)$
1	7,950±0,157	47,25±0,80	1,879±0,121
2	7,956±0,012	47,19±0,56	1,879±0,015
3	8,141±0,873	47,17±0,46	1,830±0,081
Media	8,016±0,748	47,21±0,76	1,863±0,044

Tabla 10.9: Cálculo promedio por caso de uso de los parámetros equivalentes

Conclusiones

La herramienta encuentra una solución óptima dentro de un espacio de soluciones equivalentemente óptimas. Si para una curva de referencia dada se tiene que el modelo equivalente que representa esa curva tiene

10.5. Dimensión del genoma y convergencia

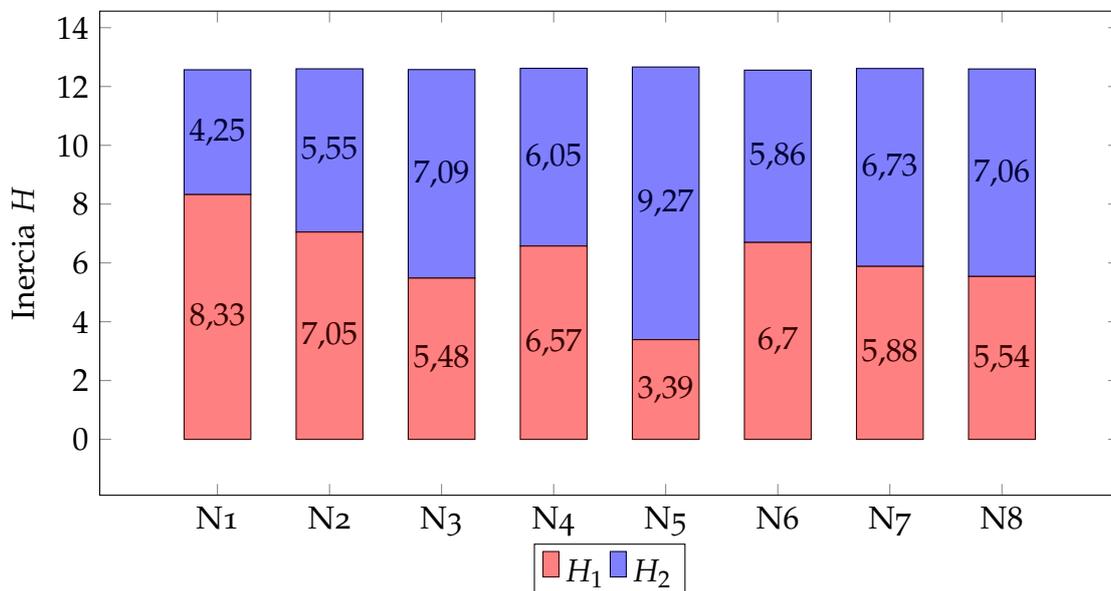


Figura 10.13: Inercia total en uninodal con caso de uso 2

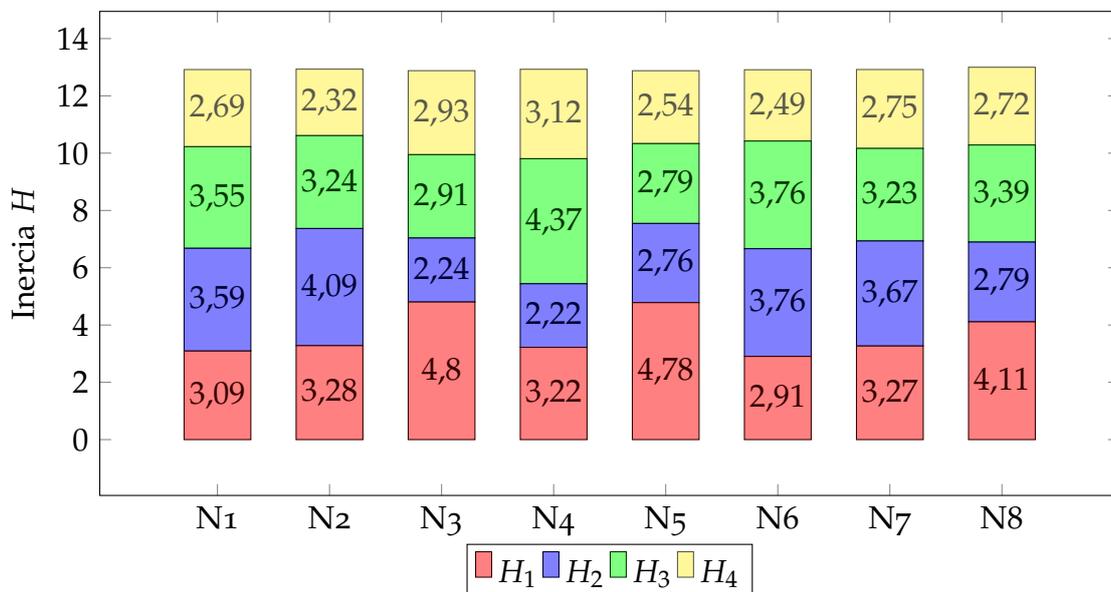


Figura 10.14: Inercia total en uninodal con caso de uso 3

Capítulo 10. Evaluaciones y resultados

inercia H_{total} , entonces habrán múltiples soluciones óptimas que cumplen $H_{total} = \sum_i H_i$, siendo H_i los parámetros de individuos correspondientes a la inercia de las máquinas.

Por esta razón, a pesar de contar con un genoma de mayor dimensión, tanto el caso de uso 2 como el caso de uso 3 tienen una convergencia similar a la del caso de uso 1 —tanto en velocidad como en calidad—, sin importar el tamaño del genoma.

Capítulo 11

Conclusiones

11.1. Solución presentada

El resultado principal del proyecto es una herramienta capaz de analizar la respuesta de la frecuencia en una simulación dinámica ante una falta de tipo escalón de potencia de un modelo de SEP y de producir un modelo uninodal cuya respuesta inercial es, de forma aproximada, equivalente a la respuesta original.

El modelo tiene las siguientes características:

- La curva de respuesta de la simulación de un escalón ΔP de igual magnitud al que originó la curva de entrada en el modelo original es aproximadamente igual a la propia curva de entrada; mientras que para escalones de diferente magnitud la curva de respuesta es aproximada —dentro de un margen aceptable— a la curva producida por la simulación de ese escalón en el modelo original.

En la sección 10.2 se realiza un análisis para distintas faltas y se concluye que la respuesta se mantiene dentro de un margen de distancia del 5% (según el *fitness* propuesto en el capítulo 6), incluso para las faltas de otro orden de magnitud u ocurriendo en otro nodo de la red de prueba.

- Para escalones de potencia de diferente magnitud en el mismo nodo, la respuesta comienza a alejarse de forma más pronunciada cuando la magnitud de la banda muerta en la red se equipara con la falta.
- Para respuestas en lugares geográficos alejados, se observa un alejamiento incremental con la distancia —en términos de nodos y líneas intermedias—.

11.2. Modelo de regulación primaria para el SEP

El modelo uninodal presentado hace uso de regulación primaria en algunos de sus generadores síncronos, e incorpora sus parámetros (estatismo y retardo en la regulación) al genoma.

Se decidió incorporar esta característica a pesar de que la zona de interés de la curva a efectos del estudio de la inercia son los primeros instantes de la respuesta, a partir de la observación experimental de que

- la respuesta de la frecuencia en un SEP durante los primeros segundos está en gran medida gobernada por la regulación primaria, siendo éste el principal efecto observable; y que
- a pesar de la regulación primaria distribuida en múltiples generadores, el comportamiento de la frecuencia en un SEP como un todo es similar al comportamiento de la misma en un único generador con regulación primaria. Esto se apoya también en las conclusiones de [7], donde se consiguieron buenos resultados en las simulaciones al incluir la regulación primaria.

Con base en ello, se desarrolló en forma teórica la respuesta de la frecuencia en un SEP con múltiples reguladores primarios y, con la ayuda de ciertas simplificaciones necesarias (pero apoyadas en los resultados), se postuló la existencia de parámetros de inercia, amortiguamiento y estatismo equivalentes en la red (sección 3.5). Con esta hipótesis de trabajo se buscó una forma de aproximar estos parámetros, para lo cual se hicieron dos esfuerzos en paralelo:

- La construcción de herramientas que pudieran trabajar sobre el modelo PSS[®] de la red original y extraer estos parámetros de allí utilizando las relaciones encontradas.
- El desarrollo de la identificación, a partir de la curva de respuesta de la red original, de estos parámetros equivalentes.

En relación con el primer enfoque, si bien fue posible obtener resultados para la inercia y el estatismo en redes pequeñas, se concluyó que tanto el cálculo del amortiguamiento equivalente como el del retardo de la regulación presentaban problemas quizás insalvables: en muchos modelos de generadores y reguladores presentes en el sistema (que, contando con los modelos de usuario, pueden ser arbitrariamente diferentes entre sí) no existe un equivalente de estos parámetros, o de haberlo, es necesario adaptarlo de otros parámetros especificados en los generadores.

El segundo método probó ser altamente efectivo. Para ello se desarrolló una teoría de identificación para ambos tipos de respuesta vistos en la

11.3. Identificación de parámetros de inercia y estatismo

práctica —oscilatorios y no oscilatorios— y se corroboró que a partir de las estimaciones encontradas para los parámetros (capítulo 5), la aproximación por mínimos cuadrados no lineales producía los parámetros de la curva con fidelidad.

Tomando este segundo enfoque, y con la hipótesis de trabajo de que los parámetros ajustados de la curva debían ser buenas aproximaciones a los parámetros buscados de inercia y estatismo en el modelo uninodal, se proveyó al algoritmo evolutivo con soluciones iniciales aproximadas a este ajuste.

11.2.1. Influencia de la generación inicial

La aproximación de las soluciones iniciales probó ser fundamental para asegurar la convergencia (y su rapidez) del algoritmo evolutivo: valiéndose de estos parámetros iniciales, se obtiene una convergencia a menos del 1 % en las primeras pocas (1 a 5) generaciones, mientras que un error en la generación inicial invariablemente aumenta el tiempo de convergencia, si no la evita por completo (sección 10.3). Este resultado validó tanto la inclusión de la regulación primaria en el modelo, como los desarrollos teóricos del capítulo 3 y el ajuste desarrollado en el capítulo 5.

11.3. Identificación de parámetros de inercia y estatismo

Una parte no menor —e integral, dados los resultados obtenidos— del proyecto fue el desarrollo de las herramientas de ajuste e identificación de parámetros de inercia, amortiguamiento y estatismo (H , D , T_g y R equivalentes de la red). Éste fue impulsado por no encontrarse en la bibliografía un compendio completo en donde se estudiase la respuesta de la frecuencia (en su fase de regulación primaria) con el enfoque necesario.

Los resultados está plasmados en el capítulo 5, donde desde un planteo teórico —fundado en lo desarrollado en el capítulo 3— se logra la identificación de los parámetros del sistema a partir de su respuesta de la frecuencia a una entrada escalón de potencia.

El desarrollo teórico se complementó con una comprobación práctica a lo largo del proyecto, donde se ajustó la técnica a las consideraciones prácticas necesarias: técnicas adicionales para el filtrado de la señal, detección de parámetros donde hay ruido superpuesto a la señal —en particular, la detección de H y t_0 en una respuesta donde se carece de una derivada inicial «limpia»—.

El resultado es un estudio sobre la respuesta de la frecuencia ante un

desbalance de potencia con regulación primaria en un SEP que, en cuanto a la identificación de los parámetros de inercia y regulación como trabajo autocontenido, es inédito en la bibliografía consultada.

Este resultado se deja como conclusión adicional del presente trabajo.

11.3.1. Algoritmo de ajuste por mínimos cuadrados

La identificación de parámetros desarrollada es usada como semilla de un ajuste por mínimos cuadrados no lineales que es capaz de corregir el error introducido por el muestreo en la identificación, así como descartar el ruido presente. Como en todo ajuste no lineal, la cercanía de la semilla es fundamental tanto para la convergencia como para la velocidad de la misma.

El resultado final asegura la convergencia en la inmensa mayoría de los casos para las señales típicas, produciendo muy pocos individuos cuyos parámetros no se puedan detectar: menos del 5% para los parámetros usados. Esta convergencia también es *rápida* —se logra en pocas iteraciones—: el promedio de tiempo de ajuste para algunas pruebas realizadas se encuentra en pocas decenas de milisegundos; en los casos más rápidos por debajo del milisegundo.

Este valor es dependiente de la granularidad (cantidad de muestras por segundo) con la que se dispone de la respuesta de la frecuencia: si bien la convergencia será más rápida si ésta es menor, se corre más riesgo de una falla en la convergencia; mientras que una mayor granularidad puede asegurar la convergencia a coste de mayor tiempo de cómputo. Hay espacio para ejecutar pruebas al respecto y detectar parámetros óptimos (sección 12.1.2).

Una pieza importante del ajuste es la introducción de los parámetros intermedios para el estudio de la curva, junto a la decisión de ejecutar el ajuste por mínimos cuadrados para los mismos en lugar de los parámetros finales. Esto no sólo simplifica los cálculos del ajuste sino que permite tratar las curvas con cantidades que están relacionadas de forma más intuitiva con la curva de respuesta y sus puntos clave.

11.3.2. Detección de la inercia en base a la respuesta inercial

Una consecuencia del algoritmo introducido es la obtención de H —o, más específicamente, $H/\Delta P$ — para la red a partir de la respuesta de la frecuencia ante una falta. De conocerse el escalón de potencia inicial para el cual se sabe la respuesta, el ajuste otorga una medida de la inercia total en la red. Se logra extraer esta información incluso si no se dispone de una medida de la velocidad del centro de inercias, o si existen perturbaciones

11.3. Identificación de parámetros de inercia y estatismo

al inicio de la medida que impiden la medición directa de la pendiente en el momento inicial.

Esto es debido a que se considera la curva más allá de la respuesta inercial, al incluirse el modelo de regulación primaria en el cálculo: la medida de H se realiza teniendo en cuenta la respuesta más allá del nadir —donde, a pesar de estar actuando otros efectos, también influye la inercia—.

Este es un resultado que podría tener aplicación en otras áreas de este campo de estudios, más allá de este estudio particular.

Capítulo 12

Trabajos futuros

El alcance del proyecto —como es típico en este tipo de trabajo— estuvo limitado por varios factores, entre ellos el tiempo máximo de trabajo, el tiempo empleado en el aprendizaje de la materia, etc. Se dejan en este capítulo algunas ideas que estaban en los planes pero fueron dejadas de lado, no sólo a fin de documentarlas sino para dar pie a futuros posibles desarrollos.

12.1. Ajuste

12.1.1. Detección oscilatoria/no oscilatoria

El ajuste del programa cliente basa su detección oscilatoria/no oscilatoria en la presencia (o no) de un pico a continuación del nadir; de faltar este pico se considera una respuesta no oscilatoria (sección 9.1.1).

En casos oscilatorios donde *a*) falla la detección del pico o *b*) la simulación termina antes de que el pico se manifieste (respuesta lenta), el algoritmo de mínimos cuadrados intentará ajustar sus parámetros intermedios hacia una respuesta oscilatoria (dado que es lo que mejor ajusta); esto se manifiesta en forma de r negativo —en los hechos sería complejo una vez atravesado el 0, dado que $j\omega = r$ —. Sin embargo, como el algoritmo impide esos valores, simplemente obtiene un r muy pequeño, terminando con un ajuste no óptimo.

Es de interés que el algoritmo de ajuste detecte esta situación a través de la validación de parámetros, y cambie el tipo de ajuste a oscilatorio en caso de que se detecte.

Puede ser deseable intentar también detectar el caso contrario; por ejemplo, cuando hay una detección espuria de pico en una respuesta no oscilatoria.

12.1.2. Granularidad

En la sección 11.3 se hace referencia a la *granularidad* de la respuesta de la frecuencia devuelta por PSS[®]e, entendida como las muestras por segundo que se usan para el ajuste. Empíricamente se vio que aumentando la granularidad, se evita la no convergencia en algunos casos, a costo de aumentar el tiempo de convergencia.

No se hicieron pruebas formales sobre un valor óptimo (más allá de usarse un valor que empíricamente dio resultados). Sería interesante poder hacer un ajuste dinámico de la granularidad para lograr ejecutar las simulaciones con *la menor granularidad posible —a fin de que tanto el ajuste como las propias simulaciones sean lo más rápidas posibles— que mantenga la cantidad de individuos no convergentes por debajo de cierto margen*, por ejemplo 95 % o 90 %, que son cifras aceptables de trabajo si otorgan más rapidez para la convergencia.

12.2. Algoritmo evolutivo

12.2.1. Elitismo

Es de interés encontrar si el valor usado para el elitismo es correcto para el problema, al igual que se hizo con otros parámetros aleatorios (sección 10.1). Se sugieren hacer pruebas con diferentes opciones y evaluar tanto la velocidad de convergencia como el valor final, dado que el elitismo puede afectar la exploración (sección 4.3).

12.2.2. Criterio de corte

Actualmente no hay un criterio de corte en el algoritmo evolutivo: éste se detiene al terminar el número prefijado de iteraciones. Es de interés tener la capacidad de definir un *fitness* mínimo para el cual, al superarse, se da por finalizado el algoritmo.

Típicamente, se ha visto que una vez llegado a un mínimo dado para una curva, el *fitness* de todos los individuos de la generación converge a ese mínimo y el promedio de *fitness* deja de bajar. Sería de interés detectar esta situación y suspender la ejecución al detectarlo, bajo el supuesto de que no se ganará más terreno más allá de cierto límite.

12.3. GUI

Si bien se proveyó una interfaz gráfica como parte del proyecto, ésta fue concebida en las últimas etapas y no pasa de ser, aunque funcional,

una *prueba de concepto*. Es un campo fértil donde se puede mejorar en muchas áreas, algunas ideas son:

- Actualmente la interfaz es muy sensible a errores en los servidores, interrumpiendo su ejecución ante una falla. Sería de interés detectar esto y actuar de forma más robusta, por ejemplo mostrando un error.
- Mostrar en la pestaña servidores si están en estado operativo, por ejemplo ejecutando un procedimiento remoto de tipo *ping*.
- Agregar la opción de ejecutar faltas en modelos uninodales especificados por el usuario: la interfaz tiene acceso a los servidores para la ejecución, por lo que simplemente puede enviar una consulta con datos entrados directamente. Esto puede servirle al operador para ganar intuición sobre valores típicos en el uninodal.
- Agregar a la herramienta la posibilidad de ejecutar modelos arbitrarios en pss[®]e para obtener curvas de referencia, para que el usuario no deba ejecutar ese paso antes de usar la herramienta (para modelos grandes, esto podría requerir una instancia local de pss[®]e con licencia completa).

12.4. Reporte

El reporte, si bien es una herramienta muy útil para el análisis durante el desarrollo del programa, está también en un estado de *prueba de concepto* y hay mucho lugar para mejoras. Algunas ideas:

- Mejorar el aspecto de las gráficas cuando hay muchas generaciones. Debido a que actualmente se grafican todas las generaciones, los datos son ilegibles si hay más de un número relativamente pequeño (decenas). Es de interés mostrar sólo un muestreo de las mismas.

12.5. Modelo uninodal

Incorporar al modelo reguladores que sean capaces de modelar el efecto de banda muerta visto en 10.2.5. Tener en cuenta este efecto podría enriquecer al modelo como equivalente dinámico de un SEP.

Anexos

Apéndice A

Software PSS[®]e

El software *Power System Simulator for Engineering* (PSS[®]e, también estilado PSS/e) [25] es una herramienta de simulación y análisis de SEP para operaciones y planificación de transmisión de energía.

Permite realizar una amplia variedad de estudios: flujo de cargas, análisis dinámicos, cortocircuito, análisis de contingencia, flujo potencia óptimo, estabilidad transitoria; entre otras.

PSS[®]e ofrece automatización y personalización a través de sus API con gran flexibilidad, incluyendo soporte para *Python*.

A.1. Simulaciones dinámicas en PSS[®]e

Una *simulación dinámica* permite estudiar la evolución de variables atadas a fenómenos físicos con un profundo nivel de detalle. Por ejemplo (y es el objeto de estudio de esta tesis), permite, ante una falta en el sistema, estudiar la evolución de la velocidad de los rotores de los generadores, voltajes instantáneos en todas las barras, etcétera.

La base de una simulación detallada es un modelo completo de la red. Para cada máquina conectada, es necesaria una descripción de todas sus variables físicas: la exactitud del modelado dependerá de la complejidad de los modelos con que se disponga.

Se detallan los pasos a seguir para la realización de simulaciones dinámicas en PSS[®]e.

- 1. Cargar caso** Un *caso* PSS[®]e es un conjunto de datos que representa un SEP: define la estructura de la red (líneas, barras, etc.) y los elementos que la componen (generadores, cargas, transformadores, compensadores, etc.), con sus parámetros y las *consignas de generación* (P , Q) y tensión V .

Los datos del sistema pueden estar dados como archivos con extensión `.sav` o `.raw`. El primero es un archivo binario sólo editable por PSS®e, mientras que el segundo es un archivo de texto plano, fácil de editar con cualquier editor de texto.

También existe la posibilidad de crear un *caso* PSS®e utilizando la interfaz de PSS®e o desde la API de *Python*.

2. Realizar un flujo de cargas Una simulación de *flujo de cargas* tiene como objetivo establecer un estado de régimen en el sistema, donde en cada barra quedan definidas:

- V_i , tensión en barra i y el ángulo de la tensión φ_{v_i} , y
- potencia activa P_i y potencia reactiva Q_i de la barra i .

Se distinguen tres tipos de barras en un flujo de cargas:

Barra	en PSS®e	Incógnitas	Datos fijos
barra PQ	tipo 1	V y φ_v	P y Q
barra PV	tipo 2	Q y φ_v	P y V
barra flotante o <i>swing</i>	tipo 3	P y Q	V y φ_v

Tabla A.1: Tipos de barras para cálculo de flujo de cargas

En la *barra flotante*, al tener P como incógnita, es donde se define el balance de potencia de la red cuando se simula un flujo de cargas: se le asigna un P tal que la potencia generada y demandada se encuentran en equilibrio.

Por lo anterior, es un requerimiento de esta simulación que siempre exista una –y sólo una– barra flotante.

Con esta simulación se obtiene como resultado una representación de un escenario de generación/demanda en un SEP o *caso base*¹ como pueden ser:

- escenario de máxima demanda en invierno;
- escenario de mínima demanda en verano;
- escenario de máxima generación eólica.

3. Agregar datos dinámicos del sistema. Un *caso base* no cuenta con la información suficiente para la realización de simulaciones dinámicas: es necesario agregar modelos que describan el comportamiento

¹Normalmente se denomina así debido a que este estudio da lugar a otros análisis como estudios de estabilidad, análisis de cortocircuitos, etc.

transitorio de las variables del sistema y de sus elementos. En particular, los generadores² deben contar con un modelo dinámico.

Los datos dinámicos se pueden obtener a partir de archivos de texto con la extensión `.dyr`; también pueden ser ingresados desde la interfaz de PSS®e o desde la API *Python*³.

Los modelos se definen indicando:

- el nombre del mismo, que en la biblioteca se vincula a un diagrama de bloques o sistema de ecuaciones diferenciales; y
- los parámetros que caracterizan dicho diagrama. Es importante tener en cuenta que estos parámetros tienen un rango de valores válidos que, en caso de no respetarse, pueden presentarse errores en la simulación.

4. Convertir los generadores a un equivalente Norton

En simulaciones de flujo de carga los generadores se representan como fuentes de tensión, y es como se representan por defecto en el caso PSS®e.

En una simulación dinámica se representan como fuentes de corriente controladas por tensión; por lo que se deben convertir los mismos a un modelo equivalente de Norton previo a la simulación.

5. Convertir las cargas a un modelo polinomial

Convertir las cargas tiene como objetivo representar de una forma más realista la dependencia de la potencia consumida en función de la tensión. PSS®e permite adoptar un modelo en el que las cargas dejan de comportarse únicamente como objetos que consumen una potencia constante. En la sección 3.2 se describen los llamados *modelos estáticos* de cargas.

Como se mencionó en el paso 3, se pueden utilizar *modelos dinámicos* para las cargas, con los que es posible simular con mayor precisión el comportamiento de cada una de las cargas.

6. Especificar archivo de almacenamiento y canales de salida

Se deben también especificar las variables que se desea observar. Existe la posibilidad de seleccionar variables referidas a los distintos elementos de la red como *canales* de muestra; a priori tantos como se desee, aunque se debe tener en cuenta que la cantidad de canales influye negativamente en el tiempo de ejecución.

²Para las cargas no es necesario utilizar modelos dinámicos. Los modelos algebraicos o *estáticos* son suficientes para las simulaciones dinámicas.

³Los modelos definidos como modelos de usuarios solo se pueden agregar desde un archivo `.dyr`, como es el caso del modelo de generador fotovoltaico PVGU1.

Por ejemplo, una simulación de 20 segundos en la red interconectada uruguayo-argentina para 200 canales tomó 40 minutos, mientras que la misma simulación tomando 20 canales tomó 10 minutos.

El resultado de la simulación se guarda en otro archivo binario de extensión `.out` o `.outx`, según la versión, dada en forma de matriz de datos donde:

- cada fila corresponde a una muestra;
- la primera columna corresponde a los datos de tiempo y
- para cada canal de muestra, corresponde una columna.

7. Correr simulación A pesar de haber definido los valores iniciales de régimen, mediante la simulación de *flujo de carga* puede persistir un desequilibrio en el sistema en las variables dinámicas. Es necesario esperar un tiempo para establecer un estado estable para las variables, previo a los eventos a simular.

Por ejemplo, un segundo en el modelo uninodal es suficiente, dado que es el sistema más simple. En el caso de la red uruguayo-argentina, llegar al régimen puede tomar decenas de segundos de simulación.

Luego, para una serie de eventos, se procede a:

- simular el evento de falla, por ejemplo: la desconexión de un generador, un cortocircuito en una línea, etc.;
- definir los tiempos de finalización de la simulación del evento t_{max} y de paso t_{step} . Para t_{step} se debe tener en cuenta el rango válido de los parámetros de los modelos dinámicos mencionado en el paso 3.

A.2. Modelos

La biblioteca de PSS®e cuenta con una amplia variedad de modelos dinámicos, tanto para máquinas convencionales —con sus correspondientes controles de excitación y frecuencia— como para generadores renovables no convencionales también con sus respectivos controles. A su vez, cuenta con modelos dinámicos para líneas, cargas, transformadores, compensadores, etc. A ello se suma la posibilidad de crear bibliotecas de modelos por parte del usuario.

A continuación se presentan las características de algunos de los modelos más comunes, relevantes en el análisis de la respuesta frecuencial.

A.2.1. Generadores convencionales

Modelos GENROE y GENROU

Máquina síncrona con rotor de polos lisos; se utiliza principalmente para plantas de generación térmica.

En la figura A.1 se pueden ver los datos de entrada de este modelo: potencia mecánica, corriente de excitación y tensión en la barra a la cual se conecta. La salida es el grupo formado por la velocidad de la máquina, la corriente del equivalente Norton, la tensión en sus terminales y el ángulo interno de la máquina.

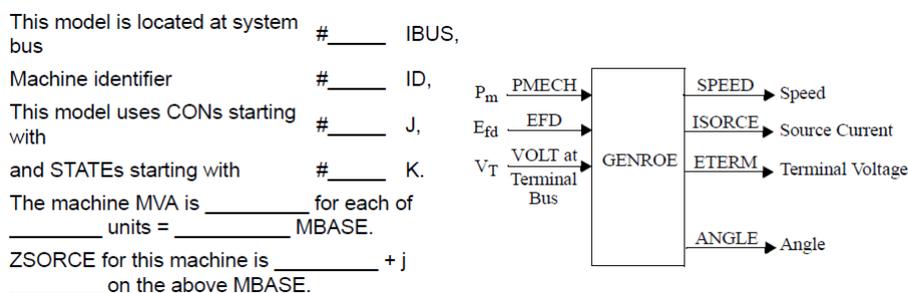


Figura A.1: GENROE

Los parámetros dinámicos necesarios para definir este par de modelos son:

- | | |
|--|--|
| T'_{do} constante de tiempo transitoria en el eje directo. | X_q reactancia síncrona en el eje de cuadratura. |
| T''_{do} constante de tiempo subtransitoria en el eje directo. | X'_d reactancia transitoria en el eje directo. |
| T'_{qo} constante de tiempo transitoria de cuadratura. | X'_q reactancia transitoria en el eje de cuadratura. |
| T''_{qo} constante de tiempo subtransitoria en el eje de cuadratura. | $X''_d = X''_q$ reactancias subtransitoria. |
| H constante de inercia de la máquina. | X_1 reactancia del inductor. |
| D speed damping, constante de los efectos de amortiguamiento. | $S_{1,0}$ factor de saturación 1,0 de la curva. |
| X_d reactancia síncrona directa. | $S_{1,2}$ factor de saturación 1,2 de la curva. |

Apéndice A. Software PSS®e

El modelo GENROU se diferencia del modelo GENROE en que el primero modela la curva de saturación de las reactancias síncronas utilizando una curva cuadrática, mientras que el segundo utiliza una curva exponencial.

Modelos GENSAL y GENSAE

Máquina síncrona con rotor de polos salientes; se utilizan principalmente para plantas de generación hidráulica.

En la figura A.2 se pueden ver los datos de entrada y salida para este modelo, similares al modelo GENSAL.

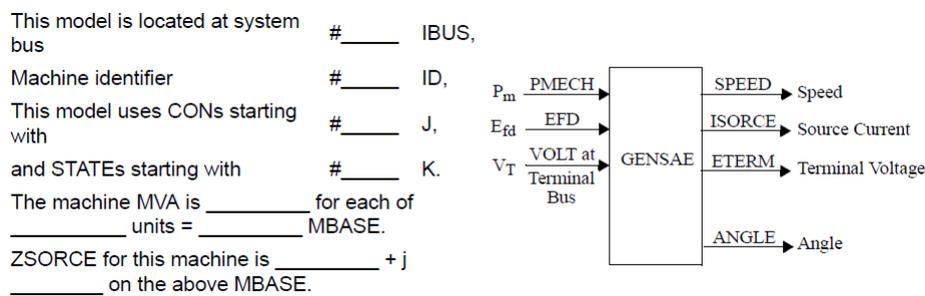


Figura A.2: GENSAE

Al igual que en el par GENROU y GENROE; GENSAL modela la saturación con una curva cuadrática y GENSAE con una exponencial.

Este modelo se define con los siguientes parámetros:

T'_{do} constante de tiempo transitoria en el eje directo.	X_q reactancia síncrona de cuadratura.
T''_{do} constante de tiempo subtransitoria en el eje directo.	X'_d reactancia transitoria en el eje directo.
T''_{qo} constante de tiempo subtransitoria en el eje de cuadratura.	$X''_d = X''_q$ reactancias subtransitoria.
H constante de inercia de la máquina.	X_1 reactancia del inductor.
D speed damping, constante de los efectos de amortiguamiento.	$S_{1,0}$ factor de saturación 1,0 de la curva.
X_d reactancia síncrona directa.	$S_{1,2}$ factor de saturación 1,2 de la curva.

Si bien estos modelos son los más comunes, tienen cierta complejidad en ámbitos que no son de nuestro interés.

El siguiente modelo que se presenta es más simple y modela específicamente lo que queremos representar en nuestro modelo uninodal.

Modelo GENCLS

Modelo clásico de generador síncrono. No se observa muy a menudo dada su simplicidad; sin embargo, puede representar fielmente el comportamiento de una máquina de este tipo para el estudio de la frecuencia y la velocidad de las máquinas.

Las entradas y salidas para este modelo son las mismas que para los casos anteriores, como se puede ver en la figura A.3.

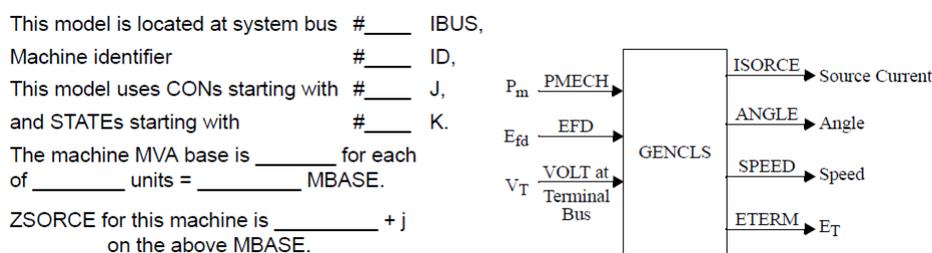


Figura A.3: GENCLS

Se define únicamente a partir de dos parámetros:

H constante de inercia de la maquina.

D constante de amortiguamiento.

El modelo GENCLS se utilizará para representar los generadores convencionales en el modelo uninodal, por simplicidad. Los modelos GENROU (GENROE) y GENSAL (GENSAE) se tendrán en cuenta a la hora de discriminar en un sistema genérico la generación térmica e hidráulica respectivamente.

A.2.2. Reguladores mediante estatismo

La biblioteca de pss[®]e provee un total de cincuenta y dos modelos de reguladores primarios de frecuencia, o *governors (turbine-governor model)*. En la tabla A.2 se listan los más comunes.

Apéndice A. Software PSS[®]e

Modelo	Descripción
DEGOV (DEGOVIDU)	Woodward diesel governor model (with speed deadband)
GAST (GASTDU)	Gas turbine-governor (with speed deadband)
GGOV1 (GGOV1DU)	GE general governor/turbine model (with speed deadband)
HGOV (HYGOVDU)	Hydro turbine-governor model (with speed deadband)
IEEEG1 (IEEEG1SDU)	IEEE type 1 speed-governing model (with speed deadband)
IEEEG2	1981 IEEE type 2 turbine-governor model
IEEEG3 (IEEEG3DU)	1981 IEEE type 3 speed-governing model (with speed deadband)
IEESGO (IEESGODU)	1973 IEEE standard turbine-governor model (with speed deadband)
PIDGOV (PIDGOVDU)	PID Hydro turbine-governor model (with speed deadband)
TGOV1 (TGOV1DU)	Steam turbine-governor model (with speed deadband)

Tabla A.2: Modelos de governors de la librería PSS[®]e

Para representar los sistemas de regulación primaria de frecuencia, en el uninodal se utilizará el modelo IEEEG2. Al igual que el modelo GENCLS, se hará uso de éste debido a su simplicidad; se evita así simular fenómenos físicos que están presentes en los otros modelos, pero que a nivel global no tienen relevancia.

Este modelo tiene como dato de entrada la velocidad del rotor de la máquina sobre la que actúa, siendo su salida la potencia mecánica que la turbina entrega a la máquina síncrona.

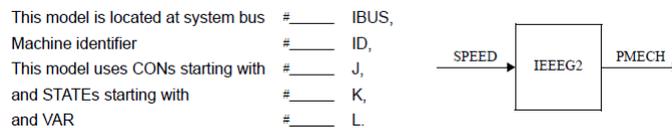


Figura A.4: IEEEG2 in-out

El modelo se define con los siguientes parámetros:

k constante que representa el inverso del estatismo permanente: $k = \frac{1}{R}$.

T_1 Constante de tiempo del *governor*.

T_4 Constante de tiempo del «agua» (también aparece como T_W).

P_{max} Máxima potencia mecánica, en p.u., que puede entregar la turbina al generador.

P_{min} Mínima potencia mecánica, en p.u., que puede entregar la turbina al generador.

T_W debe cumplir $T_W \geq 4T_{step}$, ser al menos 4 veces mayor al tiempo de paso. Es importante poder seleccionar correctamente T_{step} , de lo contrario la simulación produce errores.

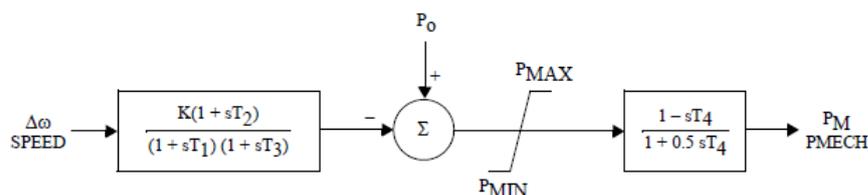


Figura A.5: Diagrama de bloques del modelo de governor IEEE2

Si comparamos el diagrama de la figura A.5 definiendo $T_1 = T_2 = T_3 = T_g$, y el diagrama de la figura 3.7, podemos ver que existe cierta equivalencia entre ambos modelos. El bloque de salida modela el comportamiento del fluido de trabajo. Tiene como entrada el cambio de consigna de potencia mecánica P_m y como salida la potencia mecánica de salida de la turbina.

A.2.3. Generadores no convencionales

En la tabla A.3 vemos los principales modelos de generadores renovables no convencionales con los que cuenta la biblioteca de pss[®]e.

Modelo	Descripción
PVGU1	User written generator model to represent photo-voltaic (PV) systems
REGCA1	Renewable Energy Generator/Converter Model
WT1G1	Direct connected (Type 1) generator
WT2G1	Induction generator with controlled external rotor resistor (Type 2)
WT3G1	Doubly-fed induction generator (Type 3)
WT3G2	Doubly-fed induction generator (Type 3), version 2
WT4G1	Wind generator model with power converter (Type 4)
WT4G2	Wind generator model with power converter (Type 4), version 2

Tabla A.3: Modelos de generadores renovables no convencionales

En los sistemas utilizados como ejemplo de *sistema original* se presentan los modelos WT4G1 y WT4G2 como generadores eólicos, aunque el primero aparece como generador solar fotovoltaico en algunos casos. Existe, además, un modelo definido como modelo de usuario derivado del modelo WT4G1 que es adaptable a un generador del tipo solar fotovoltaico: el PVGU1.

Se utilizarán en el uninodal los modelos PVGU1 y WT4G2.

Apéndice A. Software PSS®e

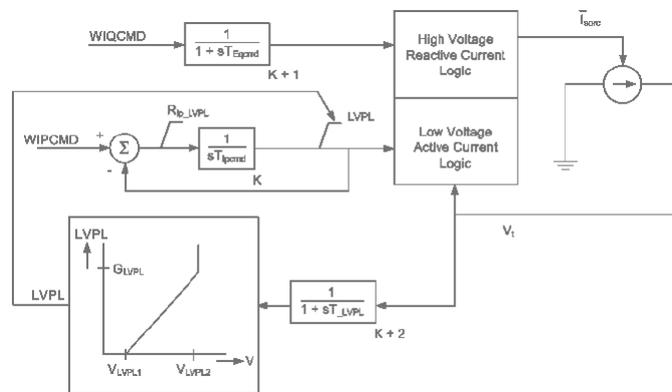


Figura A.6: Diagrama de bloque de los Modelos WT4G1 y WT4G2

Los modelos WT4G1, WT4G2 y PVGU1 se definen a partir de los siguientes parámetros:

T_{IqCmd} Converter time constant for IQcmd, second

T_{IpCmd} Converter time constant for IPcmd, second

V_{LVPL1} Low voltage power logic (LVPL) voltage 1, p.u.

V_{LVPL2} LVPL voltage 2, pu

G_{LVPL} Gain

V_{HVRCR} High voltage reactive current (HVRC) logic voltage, p.u.

CUR_{HVRCR} max. reactive current at VHVRRCR, pu

Rip_{LVPL} Rate of LVACR active current change

T_{LVPL} voltage sensor for LVACR time constants

Apéndice B

Guía del usuario

B.1. Arquitectura general

La herramienta consta de dos partes fundamentales, *programa controlador de pss^{®e}* (o «programa servidor») y el *programa controlador de motor de AA.EE.* («programa cliente»), que se conecta al primero y es la interfaz con el usuario.

Para ejecutar el programa cliente satisfactoriamente, éste debe poder conectarse a al menos un servidor a través de la red.

B.2. Programa servidor

El programa servidor hace uso de la interfaz *Python* de pss^{®e} y por ende debe instalarse en una computadora que tenga una versión de pss^{®e} instalada.

Con una instalación de la versión estudiantil pss^{®e} explore 34, el programa debe funcionar sin cambio alguno; aunque dependiendo de la versión instalada o la ubicación de pss^{®e}, las rutas incorporadas en `server.py` pueden necesitar modificaciones.

B.2.1. Instalación

Al estar escrito en *Python*, la instalación consiste en copiar el directorio `server/` del código fuente a la máquina destino.

B.2.2. Ejecución

Debe ejecutarse `server.py` con una versión de *Python 2.7* —compatible con la interfaz de pss^{®e}—. Para la versión estudiantil basta con ejecutarla

con la versión instalada. El programa servidor escucha por defecto en el puerto 8000.

B.2.3. Versión *Docker*

La versión *Docker* provista permite la ejecución en un entorno preempaquetado, que incluye la versión estudiantil de PSS[®]e que corre bajo *Wine*. Para ejecutarlo usando esta versión basta exportar el puerto 8000 del contenedor.

B.3. Programa cliente

Para la ejecución del programa cliente se debe disponer de:

- la curva de entrada, correspondiente a la falta que se desea aproximar, y
- una o más direcciones (dirección IP + puerto) de servidores a los que conectarse.

El programa cliente es un programa *Java* y por lo tanto es un requerimiento para su ejecución tener JRE instalado.

B.3.1. Versión CLI

La versión CLI es adecuada para procesamiento por lotes y ejecución automatizada. Se debe correr el programa *Java* `uninode-cli.jar` pasando como argumento el archivo de la curva de entrada. En el directorio de ejecución debe existir un archivo `uninode.params`, donde se deben especificar, al menos, las siguientes entradas:

```
init.server = servidor:puerto
generations = #generaciones
pop.subpop.0.size = #individuos
```

Las entradas se describen en la sección 9.5.1.

B.3.2. Versión GUI

Correr el programa *Java* `uninodal-gui.jar`. Funciona correctamente en Windows, Mac OS x y Linux, e incluye las bibliotecas creadas en el `.jar`, por lo que debe ser capaz de funcionar en cualquier versión moderna de JRE.

En la figura B.1 se muestra la interfaz de inicio.

La ejecución debe realizarse siguiendo los siguientes pasos:

B.3. Programa cliente

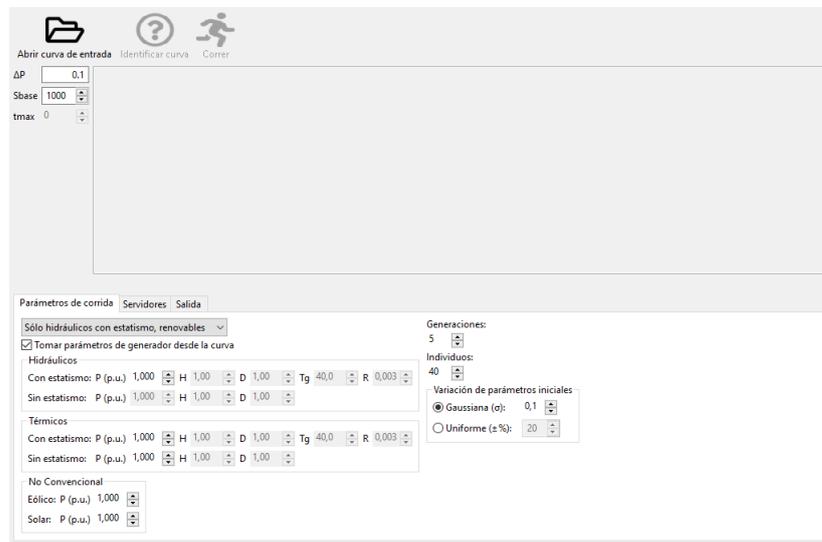


Figura B.1: Interfaz de inicio de la GUI

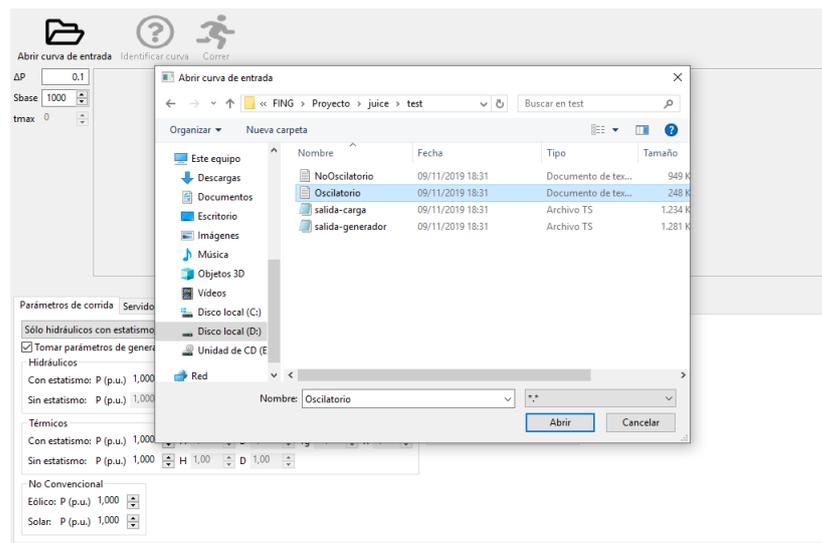


Figura B.2: Abrir curva

1. Cargar la curva de referencia. Con el botón «Abrir» se despliega un cuadro de diálogo (ver figura B.2), donde debe especificarse el archivo que contiene la curva.

Luego de seleccionar la curva, la misma se muestra en la *área de gráfico* como se ve en la figura B.3.

Apéndice B. Guía del usuario

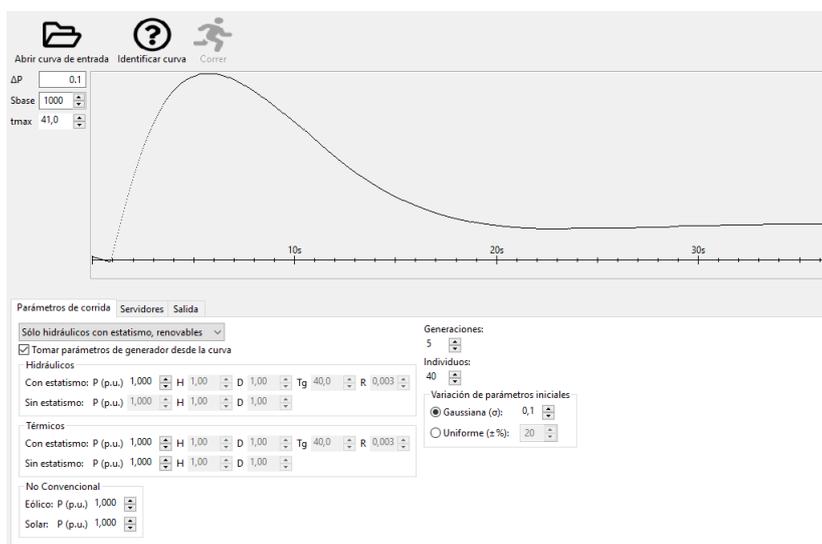


Figura B.3: Área de gráfico mostrando la curva abierta

2. Luego se debe identificar la curva. Para ello, previamente se deberán especificar algunos parámetros:

a) En la área de gráficos (a la izquierda de la figura B.4) se ingresan los datos del escalón de potencia: ΔP y S_{base} .

Es importante especificarlos antes del comando «Identificar Curva», dado que los parámetros resultantes de la identificación están dados en función de estos datos.

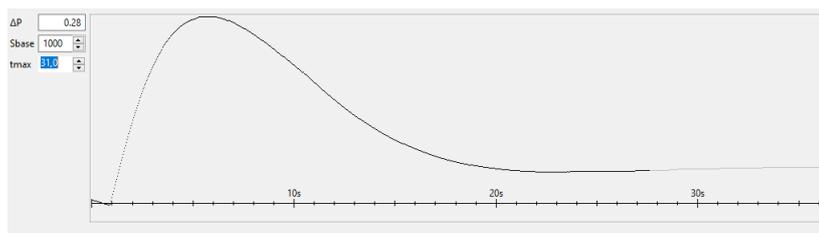


Figura B.4: Ingreso de datos del escalón de potencia

b) Luego se incorporan los datos de los generadores. Primero se define el caso de uso (ver figura B.5) y después los datos de cada generador (figura B.6)

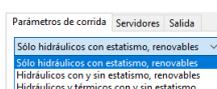


Figura B.5: Selección de caso de uso

B.3. Programa cliente

Hidráulicos con y sin estatismo, renovables
 Tomar parámetros de generador desde la curva

Hidráulicos:
Con estatismo: P (p.u.) 0,650 H 1,00 D 1,00 Tg 40,0 R 0,003
Sin estatismo: P (p.u.) 0,232 H 1,00 D 1,00

Térmicos:
Con estatismo: P (p.u.) 1,000 H 1,00 D 1,00 Tg 40,0 R 0,003
Sin estatismo: P (p.u.) 1,000 H 1,00 D 1,00

No Convencional:
Eólico: P (p.u.) 0,157
Solar: P (p.u.) 0,000

Figura B.6: Ingreso de datos de generadores

- c) Finalmente, en la área de configuración se ingresan los datos de corrida: *generaciones e individuos por generación*.

Generaciones:
50
Individuos:
100
Variación de parámetros iniciales:
 Gaussiana (σ): 0,1
 Uniforme ($\pm\%$): 20

Figura B.7

Finalmente, con el botón «Identificar Curva» se realiza la identificación de la curva cargada. Esto mostrará los parámetros detectados con el ajuste y los agregará al área de datos (figura B.8).

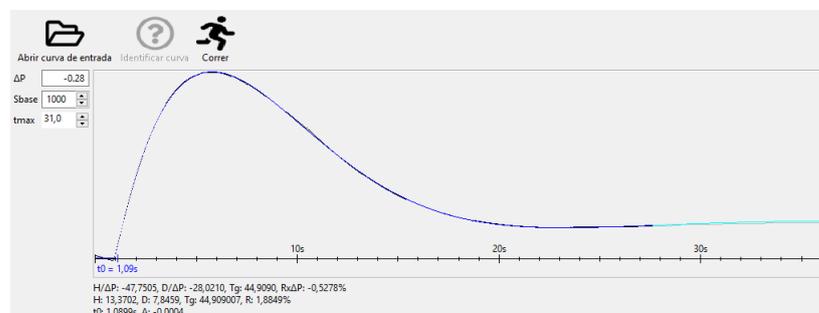


Figura B.8: Datos de identificación de curva

3. En la pestaña «Servidores», se especifican las direcciones de los servidores a los cuales se debe conectar (figura B.9).

Apéndice B. Guía del usuario

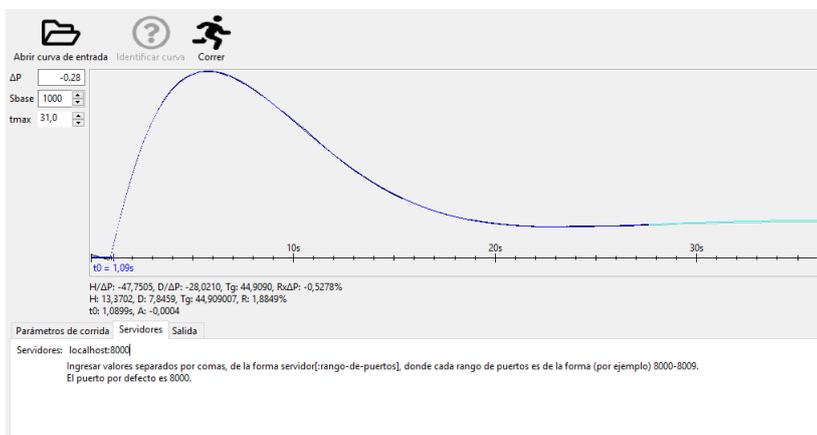


Figura B.9: Pestaña «Servidores»

4. Luego de seleccionadas las opciones se puede correr el algoritmo, cuya ejecución se inicia con el botón «Correr». En la pestaña «Salida» se mostrará el estado generación a generación (figuras B.10, B.11 y B.12).

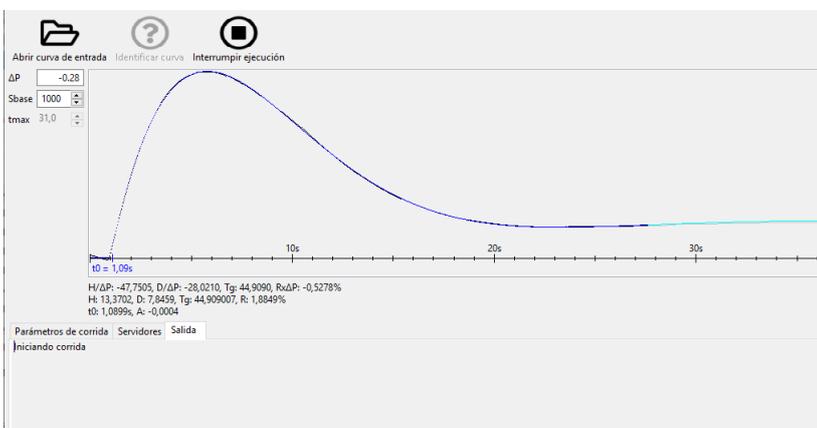


Figura B.10: Inicio de corrida

5. Al finalizar, es posible guardar el reporte de ejecución.

B.3. Programa cliente

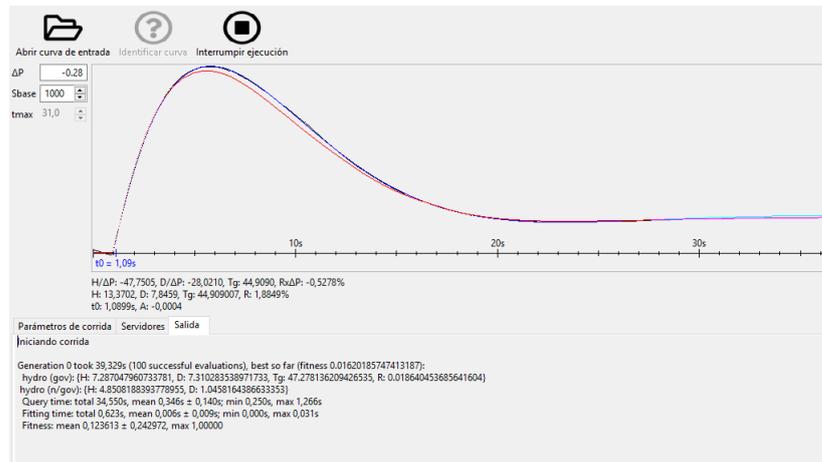


Figura B.11: Salida luego de algunas generaciones

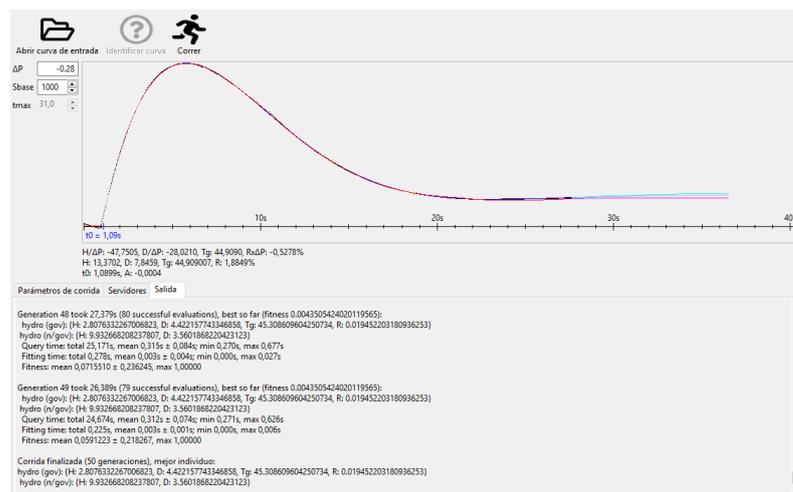
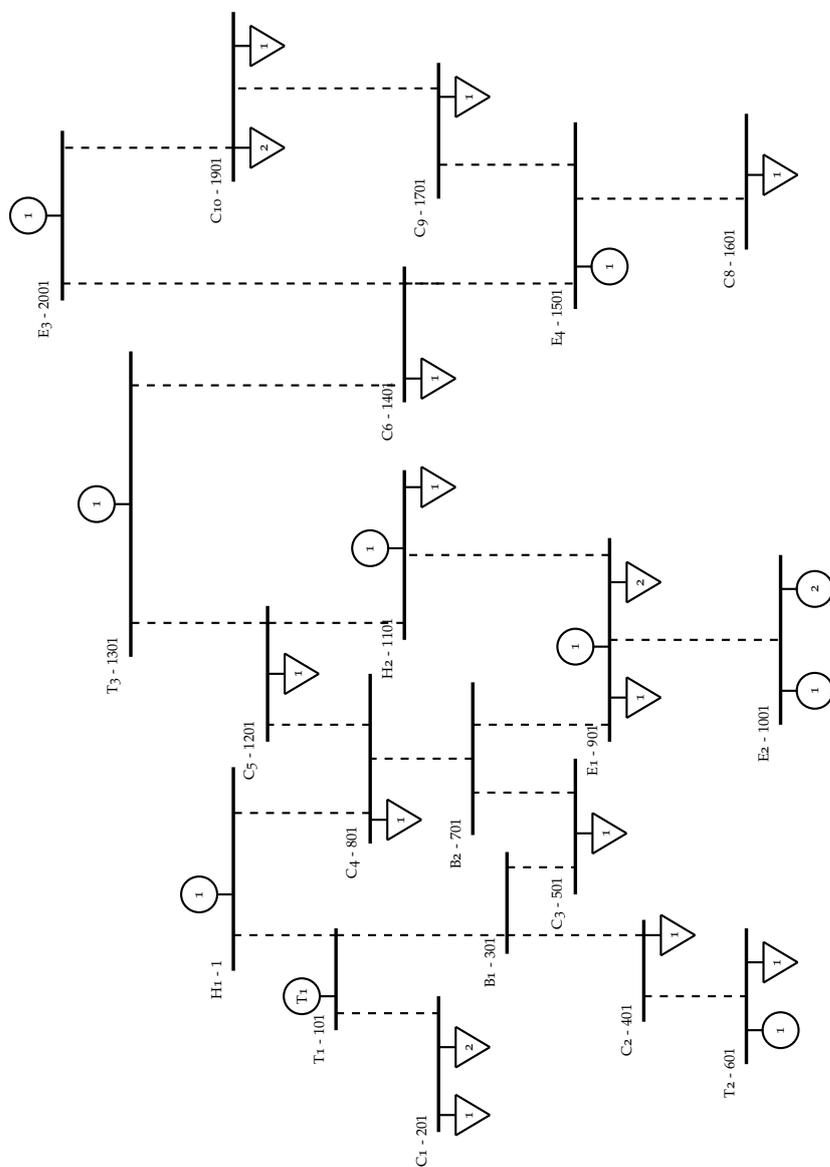


Figura B.12: Finalización de la corrida

Apéndice C

Esquema red de pruebas "A"



Referencias

- [1] *Consulta Geográfica de Fuentes de Generación*. UTE. 2018.
Disponible *online*: <https://portal.ute.com.uy/institucional/infraestructura/fuentes-de-generacion> (visitado 13-1-2020).
- [2] Argentina.gob.ar. *La generación de energía de fuentes renovables alcanzó a cubrir el 7,9 % de la demanda eléctrica durante el mes de septiembre, alcanzando un nuevo hito para el sector*. 22 de octubre de 2019.
Disponible *online*: <https://www.argentina.gob.ar/noticias/la-generacion-de-energia-de-fuentes-renovables-alcanzo-cubrir-el-79-de-la-demanda-electrica> (visitado 3-12-2019).
- [3] Secretaría de Gobierno de Energía, Ministerio de Hacienda, Presidencia de la Nación (Argentina). *Convocatoria Plan RenovAr*. 14 de noviembre de 2018.
Disponible *online*: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/315000-319999/316407/norma.htm> (visitado 3-12-2019).
- [4] *Informe de demanda correspondiente al 28/02/2019*. UTE. 2019.
Disponible *online*: https://portal.ute.com.uy/sites/default/files/files-cuerpo-paginas/informeDemanda28_02_2019.pdf (visitado 3-12-2019).
- [5] Frequency Response Technical Sub Group. *Frequency Response Technical Sub-Group Report*. Informe técnico. Grid Code Review Panel and Balancing Services Standing Group, 15 de noviembre de 2011.
- [6] R. E. LARREA MORAGA. "Análisis de la respuesta en frecuencia en sistemas de potencia con altos niveles de generación variable sin inercia". Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería Eléctrica, 2015.
- [7] A. OFTEBRO. "Analysis of Slow Frequency Variations in the Nordic Power System". Tesis de maestría. Norwegian University of Science and Technology, Department of Electric Power Engineering, 2014.
Disponible *online*: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/257950> (visitado 19-11-2019).

Referencias

- [8] P. M. ASHTON, C. S. SAUNDERS, G. A. TAYLOR, A. M. CARTER, M. E. BRADLEY. "Inertia Estimation of the GB Power System Using Synchrophasor Measurements". En: *IEEE Transactions on Power Systems* 30.2 (marzo de 2015).
- [9] F. PESCE, I. CIBILS, F. MOTTA. "Equivalente dinámico calculado mediante algoritmos genéticos". Tesis de grado. Universidad de la República (Uruguay), Facultad de Ingeniería, Instituto de Ingeniería Eléctrica, 2015.
- [10] PRABHA KUNDUR. *Power System Stability And Control*. McGraw-Hill, 1993.
- [11] Á. GIUSTO. "Seminario sobre Inercia, RoCoF y centro de inercia". 1 de abril de 2019.
- [12] P. LEDESMA. *Regulación de frecuencia y potencia*. Universidad Carlos III de Madrid. 2008.
- [13] UTE. Práctica extendida en estudios del sistema.
- [14] Melanie Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, 1999.
- [15] E. ALBA, M. TOMASSINI. "Parallelism and Evolutionary Algorithms". En: *IEEE Transactions on Evolutionary Computation* 6.5 (octubre de 2002).
- [16] J. L. GONZÁLEZ-SANTANDER, G. M. GONZÁLEZ. "Relative distance between two scalar fields. Application to mathematical modelling approximation". En: *Mathematical Methods in the Applied Sciences* 2014 37 (2013), páginas 2906-2922.
- [17] S. LUKE et al. ECJ. George Mason University, 2019. URL: <https://cs.gmu.edu/~eclab/projects/ecj/> (visitado 30-11-2019).
- [18] S. LUKE. *The ECJ Owner's Manual*. George Mason University. 2019.
- [19] D. WIENER. *XML-RPC Specification*. 15 de junio de 1999. Disponible online: <http://xmlrpc.scripting.com/spec.html> (visitado 8-12-2019).
- [20] *Apache XML-RPC*. Apache Software Foundation, 2001. URL: <https://ws.apache.org/xmlrpc/> (visitado 8-12-2019).
- [21] *Application Program Interface (API), pss^{®e} 34.3.1*. Octubre de 2017.
- [22] *pss^{®e} 34.2 Program Application Guide Vol. 2*. Abril de 2017.
- [23] *Least Squares, Apache Commons Math*. Apache Software Foundation, 2016. URL: <https://commons.apache.org/proper/commons-math/userguide/leastquares.html> (visitado 6-12-2019).

- [24] *Atribución 4.0 Internacional (CC BY 4.0)*. Creative Commons. Disponible *online*: <https://creativecommons.org/licenses/by/4.0/deed.es> (visitado 13-1-2020).
- [25] PSS[®]e. Siemens. URL: <https://new.siemens.com/global/en/products/energy/services/transmission-distribution-smart-grid/consulting-and-planning/pss-software/pss-e.html> (visitado 1-11-2019).

Esta es la última página.
Compilado el sábado 15 febrero, 2020.
<http://iie.fing.edu.uy/>