

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

Ubicación Óptima de Generación Fotovoltaica y Eólica con Análisis de
Estabilidad de Pequeña Señal para la Integración a un Sistema de
Potencia

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Electricidad

Presentado por:

Rommel Miguel Gallegos Núñez

José Gabriel Rivera Mora

GUAYAQUIL - ECUADOR

Año: 2021

DEDICATORIA

Este trabajo lo dedico con mucho cariño a mi amada familia conformada por mi padre Wilmer Gallegos, mi madre Carmita Núñez; responsables de mis objetivos de vida y superación, mis hermanas María Belén y María Fernanda que son mi motivación para mejorar como persona y profesional.

Rommel Miguel Gallegos Núñez

Dedicado a mis padres, Gabriel Rivera y Gladys Mora, quienes han sido mi mayor soporte y que, con amor y aliento, me han permitido llegar hasta estas instancias finales de mi carrera. Este y todos mis triunfos siempre serán dedicados a ustedes. ¡Los amo!

José Gabriel Rivera Mora

AGRADECIMIENTOS

Expreso mi total agradecimiento a Dios que guía mi camino, me da fortaleza y está presente siempre cuidando de mí y mi familia.

A mis padres por su apoyo incondicional, por forjar en mí el carácter necesario para afrontar cualquier obstáculo, por mostrarme una filosofía de vida que radica en la fortaleza de una familia unida y trabajadora, por inculcarme valores como responsabilidad, equidad y justicia, por mostrarme que la perseverancia, paciencia y disciplina son virtudes que transforman los sueños en objetivos.

Al PhD. Manuel Álvarez Alvarado, MSc. Fernando Vaca Urbano y el MSc. Iván Endara Vélez por compartir su guía, experiencia y confianza en el desarrollo de este proyecto.

A mis tías Doris Gallegos, Maritza Gallegos, Diana Gallegos y Narcisa Núñez por apoyarme cuando más lo necesitaba y ser ejemplo de superación en lo personal y profesional.

A mis abuelos y tíos por mostrarme que el trabajo duro y honesto es bien recompensando.

Rommel Miguel Gallegos Núñez

AGRADECIMIENTOS

Mi infinito agradecimiento a Dios por la vida y por permitirme gozar de salud, familia y amigos.

A mis padres por permitirme crecer en un hogar lleno de amor, formándome con valores que me han permitido ser una mejor persona. Agradezco su eterno sacrificio, sus invaluable consejos y su apoyo incondicional. Son mi más grande fuente de inspiración y sin ustedes jamás lo habría logrado. Me siento orgulloso de tenerlos en mi vida.

A mis tutores, MSc. Iván Endara Vélez y especialmente al PhD. Manuel Álvarez Alvarado, quien con su ayuda, orientación y conocimientos supo guiarnos en el proceso de desarrollo del proyecto integrador para culminar con éxito nuestra carrera profesional.

A mis abuelos, quienes son un ejemplo de esfuerzo y lucha constante por el bienestar de la familia. Siempre serán mi mayor referente de amor y unión familiar.

A nuestro amigo Byron De Loor, que nos ha brindado su valioso apoyo durante el proceso de desarrollo del proyecto integrador.

José Gabriel Rivera Mora

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Rommel Miguel Gallegos Núñez y José Gabriel Rivera Mora damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Sr. Rommel Miguel Gallegos Núñez

AUTOR



Sr. José Gabriel Rivera Mora

AUTOR

EVALUADORES

MSc. Iván David Endara Vélez

PROFESOR TUTOR

PhD. Manuel Sebastián Álvarez Alvarado

PROFESOR TUTOR

MSc. Fernando Arturo Vaca Urbano

PROFESOR DE LA MATERIA

RESUMEN

El crecimiento de la demanda en el sector eléctrico tiene como efecto la instalación de mayor cantidad de fuentes de generación, que a su vez se deben considerar en lo posible sean sostenibles y de bajo impacto ambiental, para ello como opción se presentan las fuentes de energía renovables, tales como la energía eólica y energía fotovoltaica con gran impacto en la integración en los sistemas eléctricos de potencia en la actualidad. La ubicación de las fuentes de generación eléctrica antes mencionadas desempeñan un papel importante en la seguridad del sistema eléctrico de potencia en presencia de un evento de fallo de corta duración (pequeña señal), como lo es la desconexión de carga, misma que está sujeta al estudio en este trabajo. Para ello se desarrolla la programación de un método heurístico que optimice la ubicación de las fuentes en función del estudio de estabilidad en pequeña señal. El proyecto tiene como objetivo identificar la ubicación óptima de generación fotovoltaica y eólica que maximicen la seguridad en el sistema de potencia, considerando el punto de vista de estabilidad para análisis de pequeña señal. Dado que se desea obtener la ubicación del punto óptimo de instalación de las fuentes de generación eólica y fotovoltaica se procede a desarrollar un programa unificando entre el software DigSILENT-PowerFactory™ y Python™, que sirva de herramienta para determinar la solución del caso de estudio, de tal manera que en Python se desarrolle la codificación del método de optimización por enjambre de partículas (PSO) y este sirva de control maestro de la interface gráfica de DigSILENT-PowerFactory donde se realizara el estudio de estabilidad de pequeña señal, siendo Python una herramienta de programación especializada de libre acceso y DigSILENT-PowerFactory esencial para estudios de corto circuito, flujo de potencia, protecciones eléctricas, estabilidad, etc. Los resultados determinan la ubicación optima de las fuentes eólica y fotovoltaica permitiendo mejorar la estabilidad del sistema. Se concluye que DigSILENT-PowerFactory proporciona un sistema con herramientas especializadas en estudios eléctricos como estabilidad, en donde se permite el diseño de un sistema con generación eólica y fotovoltaica correspondiente al estudio que muestra el comportamiento de las señales de frecuencia y voltaje en un instante de tiempo de actuación de la perturbación, específicamente cuando se genera el evento de

desconexión de carga en la barra, mientras tanto Python agiliza el estudio mediante la automatización del proceso de la red con su programación. El algoritmo satisface la búsqueda de la mejor localización de fuentes eólicas y fotovoltaicas en caso de entrar en operación en la red eléctrica, claro está que existen más parámetros que considerar como por ejemplo confiabilidad, protecciones eléctricas y la incidencia del recurso renovable en la locación, pero para el caso planteado en este trabajo está en función de análisis de estabilidad. Finalmente se recomienda para que el enlace entre Python y DigSILENT-PowerFactory sea una realidad, estos deben mantener versiones compatibles.

Palabras Clave: Estabilidad de pequeña señal, Python, DigSILENT-PowerFactory, PSO, ubicación.

ABSTRACT

The growth of demand in the electricity sector has the effect of installing a greater number of generation sources, which in turn should be considered as sustainable and with low environmental impact, as an option, renewable energy sources such as wind power and photovoltaic power are presented with great impact in the integration in electrical systems today. The location of the sources of electrical generation play an important role in the safety of the electrical power system in the presence of a short-term failure event (small signal), such as load disconnection, which is subject to the study in this job. For this, the programming of a heuristic method is developed to optimize the location of the sources based on the study of stability in small signals. The project aims to identify the optimal location for photovoltaic and wind generation that maximize security in the power system, considering the point of view of stability for small signal analysis. Since it is desired to obtain the location of the optimal installation point of the wind and photovoltaic generation sources, a program unifying between the DigSILENT-PowerFactory™ and Python™ software is developed, which will serve as a tool to determine the solution of the study case with Python codifying the particle swarm optimization method (PSO) and serving as the master control of the DigSILENT-PowerFactory graphical interface where the small signal stability study will be carried out. The language Python is specialized programming tool of free access and DigSILENT-PowerFactory is essential for short circuit studies, power flow, electrical protections, stability, etc. It is concluded that DigSILENT-PowerFactory provides a system with specialized tools in studies such as stability, which allows the design of a system with wind and photovoltaic generation corresponding to the study that shows the behavior of frequency and voltage signals in an instant of time of the disturbance, specifically when the load disconnection event is generated in the bus, meanwhile Python speeds up the study by automating the network process with its programming.

Keywords: Small signal stability, Python, DigSILENT-PowerFactory, PSO, Location.

ÍNDICE GENERAL

EVALUADORES	6
RESUMEN	I
<i>ABSTRACT</i>	III
ÍNDICE GENERAL	IV
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	X
CAPÍTULO 1	1
1.1 Introducción	1
1.2 Descripción del problema	3
1.3 Justificación del problema.....	4
1.4 Objetivos.....	5
1.4.1 Objetivo general	5
1.4.2 Objetivos específicos	5
1.5 Limitaciones.....	5
CAPÍTULO 2.....	6
2.1 Marco teórico	6
2.2 Estabilidad en sistemas de potencia.....	6
2.3 Energías renovables.....	13
2.3.1 Energía solar fotovoltaica	17
2.3.2 Energía eólica	19
2.4 Método de optimización por enjambre de partículas (PSO).....	21
2.5 Control	27
2.5.1 Modelo PSS	27
2.5.2 Modelo AVR.....	27
2.6 Lenguaje de programación y software de estudios eléctricos.....	28

2.6.1	Python	28
2.6.2	DigSILENT-PowerFactory	29
2.6.3	Pycharm community edition	32
CAPÍTULO 3.....		34
3.1	Metodología	34
3.2	Estudio de estabilidad de pequeña señal en PowerFactory.	34
3.3	Estudio de estabilidad de pequeña señal para un sistema de 6 barras	39
3.3.1	Importación y activación del proyecto	40
3.3.2	Definición de eventos	42
3.3.3	Definición de variables para estudio de estabilidad.....	44
3.3.4	Cálculo de flujo de carga y condiciones iniciales	45
3.3.5	Simulación de los eventos programados.....	48
3.3.6	Graficación y exportación de resultados	50
3.4	Sistema de 6 barras con generación eólica y fotovoltaica	54
3.5	Enlace computacional de PowerFactory con Python.....	61
3.5.1	Compatibilidad entre programas	63
3.5.2	Módulo powerfactory	66
3.6	Ubicación óptima de fuentes eólicas y fotovoltaicas.....	67
3.6.1	Análisis matemático de estabilidad de pequeña señal	68
3.6.2	Proceso matemático del código fuente en Python	73
3.6.3	Caso de estudio	77
3.6.4	Sistema modificado IEEE de 39 barras.....	78
3.6.5	Integración de PSS y AVR	81
3.6.6	Índice de estabilidad.....	86
3.6.7	Modelamiento matemático	87
3.6.8	Diseño de código fuente en Python.....	88

CAPÍTULO 4	103
4.1 Resultados y análisis	103
4.2 Ubicación óptima para la generación eólica y fotovoltaica.....	103
4.3 Resultados proporcionados por Python	106
4.4 Respuestas del sistema frente a una perturbación de pequeña señal	108
4.4.1 Respuestas del generador Gen1 frente a evento en la carga Load39 .	108
4.4.2 Respuestas de generadores síncronos a la perturbación en Load39 ..	112
4.4.3 Respuestas de generadores síncronos a la perturbación en Load23 ..	115
4.5 Análisis de código fuente desarrollado en Python	118
4.6 Análisis de estudio de estabilidad de pequeña señal	121
CAPÍTULO 5.....	123
Conclusiones	123
Recomendaciones	125
BIBLIOGRAFÍA.....	126
APÉNDICES	129
APÉNDICE A: DESARROLLO MATEMÁTICO	130
APÉNDICE B: DATOS DINÁMICOS SÍNCRONOS	132
APÉNDICE C: COMANDOS DE PROGRAMACIÓN	135
APÉNDICE D: ESTÁNDARES DE OPERACIÓN	136

ÍNDICE DE FIGURAS

Figura 2.1. Clasificación de estabilidad en sistemas de potencia basado en [9]	8
Figura 2.2. Estado de equilibrio con tensión de campo fijo [9].....	10
Figura 2.3. Estado inestable no oscilatoria con tensión de campo fijo [9].....	11
Figura 2.4. Estado inestable oscilatorio con control de excitación [9].....	12
Figura 2.5. Consumo mundial de energía según el sector de ocupacional [12].....	15
Figura 2.6. Energía generada a nivel mundial según el tipo de combustible [12]	15
Figura 2.7. Incremento en la capacidad instalada de energías renovables [13]	16
Figura 2.8. Conexión a la red de un generador fotovoltaico típico [15]	18
Figura 2.9. Elementos conectados al sistema eólico [18]	20
Figura 2.10. Principales componentes de la turbina eólica de eje horizontal [17].....	21
Figura 2.11. Partículas en la búsqueda de su solución óptima [19].....	23
Figura 2.12. Diagrama de flujo del método convencional de PSO basado en [20]	26
Figura 3.1. Obtención de la mejor locación de generación eólica y fotovoltaica	36
Figura 3.2. Icono de escritorio de PowerFactory 2021 SP5	37
Figura 3.3. Icono de escritorio de PyCharm Community.....	39
Figura 3.4. Diagrama unifilar para el sistema de 6 barras	40
Figura 3.5. Importar el sistema de 6 barras al entorno de PowerFactory	41
Figura 3.6. Activación del proyecto que contiene el sistema de 6 barras	41
Figura 3.7. Definir un evento de switch en PowerFactory.....	43
Figura 3.8. Ventana de configuración para el evento de switch.....	43
Figura 3.9. Ventana de edición de eventos simulados.....	43
Figura 3.10. Definición de variables.....	44
Figura 3.11. Variables disponibles en el generador síncrono <i>Gen1</i>	45
Figura 3.12. Configuración para el caso de estudio de flujo de carga	46
Figura 3.13. Flujo de carga del sistema de 6 barras – primera parte.....	47
Figura 3.14. Flujo de carga del sistema de 6 barras – segunda parte	47
Figura 3.15. Ajuste de condiciones iniciales para el sistema de 6 barras	48
Figura 3.16. Configuración del tiempo de simulación del sistema de 6 barras	49
Figura 3.17. Simulación de apertura y cierre de switch para la carga <i>Load1</i>	49

Figura 3.18. Selección de <i>Plot page</i>	50
Figura 3.19. Tipo de gráfica seleccionada	51
Figura 3.20. Ventana de <i>Curve plot</i>	51
Figura 3.21. Gráfica de potencia activa para el generador <i>Gen1</i> en MW	52
Figura 3.22. Gráfica de Frecuencia Eléctrica del generador <i>Gen1</i> en Hz.....	53
Figura 3.23. Exportación de datos	53
Figura 3.24. Red de 6 barras con integración de generación eólica y fotovoltaica	54
Figura 3.25. Flujo de carga en el sistema luego de la perturbación.....	55
Figura 3.26. Señal de frecuencia eléctrica en [Hz].....	57
Figura 3.27. Frecuencia eléctrica en [Hz] con <i>tsim</i> = 200 [s]	58
Figura 3.28. Señal de Potencia Activa en [MW].....	59
Figura 3.29. Señal de voltaje terminal en [p.u.].....	60
Figura 3.30. Señal de voltaje terminal en [p.u.] con <i>tsim</i> = 200 [s].....	60
Figura 3.31. Secuencia para el enlace de PowerFactory con Python [5].....	62
Figura 3.32. Localización del módulo <i>powerfactory.pyd</i> en el ordenador	64
Figura 3.33. Versiones del módulo <i>powerfactory.pyd</i>	64
Figura 3.34. Módulo <i>powerfactory.pyd</i> para la versión de Python 3.8	64
Figura 3.35. Ajuste del IDE para que use el intérprete de Python 3.8	65
Figura 3.36. Sentencia para enlazar Python con PowerFactory	66
Figura 3.37. Sistema modificado IEEE de 39 barras [29]	79
Figura 3.38. Modelo compuesto de la unidad generadora <i>Gen2</i>	83
Figura 3.39. Marco compuesto SYM <i>Frame_no droop</i> de PowerFactory	84
Figura 3.40. Common Model para el AVR del <i>Gen2</i>	85
Figura 3.41. Common Model para el gobernador del <i>Gen2</i>	85
Figura 3.42. Common Model para el PSS del <i>Gen2</i>	86
Figura 3.43. Enlace de Python con PowerFactory	90
Figura 3.44. Importación de librerías	90
Figura 3.45. Activación del proyecto que contiene el sistema de 39 barras	90
Figura 3.46. Acceso a directorios de elementos del sistema de 39 barras	91
Figura 3.47. Función para digitar el número de partículas.....	91
Figura 3.48. Función para digitar el número de iteraciones	92
Figura 3.49. Función que determina si un número es par.....	92

Figura 3.50. Función de lectura de archivos de texto	93
Figura 3.51. Función que obtiene el Best Local por partícula	94
Figura 3.52. Función Best Global de todo el enjambre hasta el momento.....	94
Figura 3.53. Función Best Global óptimo al final del proceso	95
Figura 3.54. Función para crear la carpeta de eventos.....	95
Figura 3.55. Función para imprimir los resultados por iteración.....	96
Figura 3.56. Impresión del Best Global óptimo al final del proceso	97
Figura 3.57. Asignación de la función de eventos - Parte 1	97
Figura 3.58. Asignación de la función de eventos - Parte 2	98
Figura 3.59. Asignación de la función de eventos - Parte 3	99
Figura 3.60. Método metaheurístico PSO en Python.....	99
Figura 3.61. Cálculo de la nueva velocidad y posición de la partícula	99
Figura 3.62. Asignación de la función de eventos, Parte 4	100
Figura 3.63. Asignación de la función de eventos, Parte 5	100
Figura 3.64. Ejecuta la simulación de los eventos	101
Figura 3.65. Exportación de resultados en archivo de texto	101
Figura 3.66. Tiempo de cambio entre partículas del enjambre resultados.....	101
Figura 4.1. Esquema de ubicación óptima de generación eólica y fotovoltaica	104
Figura 4.2. Ubicación óptima de generación eólica y fotovoltaica en Python	107
Figura 4.3. Frecuencia eléctrica [Hz] del Gen1 frente a evento en Load39	109
Figura 4.4. Potencia activa total [MW] del Gen1 frente a evento en Load39	110
Figura 4.5. Voltaje terminal [p.u] del Gen1 frente a evento en Load39	111
Figura 4.6. Frecuencia [Hz] de generadores síncronos frente a evento en Load39	112
Figura 4.7. P. Activa [MW] de generadores síncronos frente a evento en Load39 ..	114
Figura 4.8. <i>Vterm</i> [p.u] de generadores síncronos frente a evento en Load39.....	115
Figura 4.9. Frecuencia [Hz] de generadores síncronos frente a evento en Load23	116
Figura 4.10. P. Activa [MW] de generadores síncronos frente a evento en Load23	117
Figura 4.11. <i>Vterm</i> [p.u] de generadores síncronos frente a evento en Load23.....	118
Figura 4.12. Acción del interruptor para el generador fotovoltaico PV_Sys1	120
Figura 4.13. Creación de eventos	120

ÍNDICE DE TABLAS

Tabla 2.1. Consumo Mundial de Energía [12].....	14
Tabla 2.2. Comparación de Variantes del Método PSO [20]	24
Tabla 3.1. Elementos de la Red Modificada IEEE 39 Barras	79
Tabla 3.2. Generadores Síncronos del Sistema Modificado IEEE de 39 Barras	80
Tabla 3.3. Demanda del Sistema Modificado IEEE de 39 Barras	81
Tabla 4.1. Generación Eólica y Fotovoltaica Activa en el Sistema	105
Tabla 4.2. Frecuencia Máxima Frente a Evento en Load39	113
Tabla 4.3. Frecuencia de Estabilización Frente a Evento en Load39	113
Tabla 4.4. Frecuencia Máxima Frente a Evento en Load23	116
Tabla 4.5. Frecuencia de Estabilización Frente a Evento en Load23	117
Tabla B.1. Datos de Máquina de la Red Modificada IEEE 39 Barras	132
Tabla B.2. Datos de Excitatriz de la Red Modificada IEEE 39 Barras.....	133
Tabla B.3. Datos del Gobernador en la Red Modificada IEEE 39 Barras	134
Tabla C.4. Funciones Implementadas en Código Fuente de Python	135
Tabla D.5. Estándar de Niveles de Tensión en Barras de Red de Transmisión	136
Tabla D.6. Normalización de Centrales de Generación por Control de Tensión.....	136
Tabla D.7. Limitantes de Frecuencia en Estado Estable	136
Tabla D.8. Requerimientos Mínimos de Estabilización en Estado Dinámico	137
Tabla D.9. Limitantes de Estabilidad y Seguridad del Sistema	137
Tabla D.10. Norma de Frecuencia con Integración de Generación Eólica	137
Tabla D.11. Estándar de Frecuencia en América del Norte	138

CAPÍTULO 1

1.1 Introducción

El sistema eléctrico de potencia (SEP) es el encargado de satisfacer la demanda diaria de distintos tipos de consumidores o usuarios que utilizan de manera cotidiana la energía eléctrica, que va desde encender artefactos eléctricos como refrigeradora, licuadora, plancha; hasta permitir la operación de maquinaria de perfil industrial como bandas automáticas, motores eléctricos y turbinas. La realidad del ser humano es dependiente de este recurso de tal manera que tiene que estar presente de forma continua las 24 horas del día. Es por lo que las empresas encargadas del sector eléctrico tienen la obligación de proporcionar un sistema seguro y confiable a los pequeños y grandes consumidores de energía eléctrica.

A lo largo de la historia, la energía eléctrica se ha expuesto a constantes estudios y pruebas, desde cargar una barra por medio de frotamiento y decidir si es factible la transferencia de corriente directa (CD) o corriente alterna (CA) en las redes eléctricas, hasta la actualidad en donde se encuentran diseños de sistemas complejos que están constituidos de elementos como lo son interruptores, disyuntores, relés automáticos y entre otros [1]. La innovación en función de proveer un buen servicio eléctrico, es decir, seguro y económico, esto constituye un trabajo impetuoso de investigación científica en materia de pérdidas por conducción, materiales, protecciones eléctricas, confiabilidad y estabilidad de sistemas de potencia.

Las fuentes renovables representan un mecanismo de generación de energía no convencional encargadas de proporcionar energía limpia y amigable con el ambiente. Las fuentes eólicas y fotovoltaicas están tomando más fuerza al momento de considerar generación asequible y no contaminante ya que aprovechan la energía del viento y la radiación solar respectivamente para luego transformarla en energía eléctrica. Dependiendo del porcentaje de fuentes eólicas y fotovoltaicas instaladas en los sistemas de potencia, estas se identifican como alta penetración y baja penetración [2].

Actualmente, el procesamiento de información es un campo de estudio en crecimiento e innovación debido a los datos que se recolectan y se almacenan. Para obtener una base de datos actualizada se debe generar de manera rutinaria una memoria técnica con datos como ingresos, utilidades, inversión y pérdidas. Para el caso del sector eléctrico, estos datos son mediciones de variables como voltaje, corriente, potencia y frecuencia en instantes de tiempo. La gran base de datos recopilada a lo largo de los años contribuye a la optimización o predicción del comportamiento de sistemas, esto a partir del modelamiento matemático de funciones y uso de técnicas metaheurísticas.

El método de optimización por enjambre de partículas, por sus siglas en inglés correspondiente a Particle Swarm Optimization (PSO) [3], es un método de optimización que busca una solución óptima para satisfacer una función objetivo específica. Para ello se debe de considerar el uso de un vector que indique a la partícula que elementos tiene a su alrededor, seguido actúa el vector que identifica la mejor solución de entre todas las direcciones posibles y guía a las partículas a dicha solución para que el ultimo vector, conocido como vector de inercia, sirva de indicador a las partículas para que busquen direcciones posibles un poco más lejos de la posición actual.

La simulación de eventos de perturbaciones de pequeña señal en los sistemas de potencia presentados en el estudio de este proyecto está dada por el software DigSILENT-PowerFactory™ con licencia de la Escuela Superior Politécnica del Litoral (ESPOL). Este proporciona una amplia librería con elementos necesarios para el diseño y estudio de sistemas eléctricos. Para el trabajo a realizar se considera el uso de elementos como fuentes eólicos y fotovoltaicos, generadores asíncronos, barras, líneas de transmisión, transformadores y para el estudio tenemos el análisis de flujo de carga, conexión y desconexión de carga y análisis de estabilidad.

El algoritmo de PSO, se desarrolla en el software Python™ de licencia libre que consta de un ambiente de programación fluido y eficaz, ya que proporciona librerías

de mucha utilidad para el análisis matemático como matrices, funciones matemáticas, etc. Este software es utilizado para la codificación de algoritmos en todo el mundo, siendo un facilitador a estudiantes, investigadores y desarrolladores, en la creación de herramientas que sirven de soporte o ayuden a corroborar la investigación [4].

1.2 Descripción del problema

El sistema eléctrico de potencia (SEP), encargado de proveer un servicio seguro, a lo largo del tiempo se ha visto expuesto a constantes estudios, esto se debe a la realidad que presenta en el transporte de energía respecto a pérdidas por conducción o fallas en el sistema. Las perturbaciones en sistemas eléctricos son muy comunes, un ejemplo de esta es la desconexión de carga, misma que se produce en instantes cortos de tiempo, que a su vez va acompañada de una pérdida de sincronismo de las máquinas que se encuentran conectados al sistema, tales como las fuentes generadoras.

La constante mejora del sistema en términos de seguridad e impacto ambiental implica que se realice la implantación de fuentes de energía no convencionales como eólica y fotovoltaica que proporcionan incidencia en la estabilidad del sistema frente a una perturbación. Para aquello se realiza la búsqueda de un método que ayuda a encontrar un mínimo posible de inestabilidad (solución óptima) en el sistema, considerando la ubicación de la fuentes eólicas y fotovoltaicas a partir del análisis de estabilidad de pequeña señal.

El estudio de estabilidad de pequeña señal contribuye con la seguridad del sistema, mismo que debe ser sostenible en el tiempo para servicio del usuario, ya que la continuidad del sistema no debe ser afectada junto con los parámetros de potencia, voltaje y frecuencia luego de una perturbación. El uso del software DigSILENT-PowerFactory con licencia de ESPOL para el estudio de estabilidad permite desarrollar o implementar la red sujeta a estudio, para luego utilizar su herramienta de análisis de pequeña señal que muestra el comportamiento del voltaje, potencia y frecuencia en función del tiempo [5]. Esto da paso al usuario que seleccione la mejor ubicación de la generación eólica y fotovoltaica, considerando antes la

optimización de una función objetivo que representa el comportamiento del sistema en términos de estabilidad mediante la aplicación del método de PSO, presentado como código fuente por Python (software libre).

1.3 Justificación del problema

La propuesta del proyecto conlleva a considerar la simulación de un sistema con alta incidencia de generación eólica y fotovoltaica, en donde se estudia la estabilidad de pequeña señal, con el fin de garantizar un sistema seguro y óptimo tanto para los consumidores del servicio de energía eléctrica como para las empresas inmersas en el sector eléctrico. Se debe utilizar el software DigSILENT-PowerFactory y Python de manera unificada, en donde se tome como referencia la ubicación de las barras del sistema. La ayuda proporcionada por el método de optimización por enjambre de partículas (PSO), constituye en determinar la mejor ubicación de la generación eólica y fotovoltaica a partir de la optimización de una función objetivo que representa la estabilidad en el sistema. De esta manera el usuario encargado del estudio verificará, según la red en particular implementada en DigSILENT-PowerFactory, si las posiciones establecidas de fuentes eólicas y fotovoltaicas son apropiadas. En un contexto real se debe considerar también si en la posición detectada existen recursos suficientes para la operación, ya sea flujo de aire para la fuentes eólicas y radiación solar para las fuentes fotovoltaicas.

La simulación y el análisis de estabilidad de pequeña señal son esenciales para la toma de decisión de la ubicación de fuentes eólicas y fotovoltaicas en la integración al sistema de potencia [6]. Python facilitaría la toma de decisiones a los usuarios encargados del estudio de estabilidad ya que por medio del código fuente se obtiene automáticamente el posicionamiento óptimo en función de los resultados del estudio de estabilidad de pequeña señal realizados a los generadores en DigSILENT-PowerFactory. Esta prueba se realiza cambiando la locación de las fuentes eólicas y fotovoltaicas en distintas barras del sistema hasta que se encuentre la óptima.

1.4 Objetivos

1.4.1 Objetivo general

Identificar la ubicación óptima de generación fotovoltaica y eólica que maximicen la seguridad en el sistema de potencia, considerando el punto de vista de estabilidad para análisis de pequeña señal.

1.4.2 Objetivos específicos

1. Utilizar el software Python™ en combinación de DIgSILENT-PowerFactory™ como herramienta que permita la simulación del estudio de estabilidad para pequeña señal.
2. Evaluar el algoritmo de optimización por enjambre de partículas (PSO) para definir el punto óptimo de ubicación de generación fotovoltaica y eólica.
3. Estudiar la estabilidad de un sistema de potencia bajo perturbaciones de pequeña señal en presencia de generación fotovoltaica y eólica.

1.5 Limitaciones

En la búsqueda del posicionamiento óptimo para la integración de generación eólica y fotovoltaica, se debe considerar que la ubicación de las fuentes de energía renovables eólicas y fotovoltaicas están en función del recurso natural, es decir gran incidencia de flujo de aire y radiación solar respectivamente conlleva a que exista mayor penetración en el sistema. Los niveles de integración de fuentes de energía renovable, como es el caso de las fuentes eólicas y fotovoltaica en el sistema se manifiestan en baja y alta penetración.

CAPÍTULO 2

2.1 Marco teórico

La comprensión del lector referente al proyecto se debe al conocimiento de las bases teóricas y prácticas fundamentales en los campos de estudios de estabilidad de sistemas de potencia, energías renovables, métodos metaheurísticos, herramientas computacionales especializadas en estudios eléctricos y programación aplicada a la ingeniería. El desarrollo del trabajo necesita del manejo o dominio de los conceptos mencionados previamente para una correcta y eficiente búsqueda de la solución a la problemática planteada.

2.2 Estabilidad en sistemas de potencia

La complejidad de los sistemas de potencia se debe a la mejora continua para proveer un servicio seguro. Esto se da mediante la integración de elementos encargados de disminuir las pérdidas de conducción y mantener la continuidad de operación en la red. La diversidad de elementos aporta a la variabilidad en tiempos de actuación y características de operación, que se ven reflejadas en la función de salida del sistema. Cuando la red se expone a una perturbación, la estabilidad se ve afectada porque su punto de operación en estado estable se mueve a otra posición para contrarrestar el cambio brusco producido. Esto se da porque el sistema necesita permanecer en equilibrio para operar correctamente y así mantener el sincronismo en las máquinas sincrónicas [7].

El sistema eléctrico de potencia, frente a perturbaciones, debe buscar mantener su estado de equilibrio siempre y cuando la interrupción de línea o la conexión y desconexión de carga en la red sea momentánea para que no exista impacto permanente en la potencia del sistema. El proceso de recuperación de sincronismo de las máquinas síncronas repercute en la inercia del rotor, considerando así también el uso de dispositivos que aseguren y controlen de manera óptima la operación de la máquina frente a fallas de red. De esta forma se mantiene en

equilibrio a los parámetros del sistema como lo son la frecuencia, voltaje, potencia, etc.

Las pequeñas perturbaciones pueden afectar al ángulo del rotor logrando que se pierda el sincronismo del sistema. La estabilidad busca contrarrestar esta perturbación manteniendo el sincronismo y el equilibrio del par de todas las máquinas síncronas interconectadas del sistema. Bajo planteamiento matemático, la estabilidad se logra linealizando las ecuaciones diferenciales no lineales que rigen el comportamiento del torque eléctrico estableciendo un nuevo punto de equilibrio antes las oscilaciones presentadas. En grandes perturbaciones (transitorias), las ecuaciones diferenciales no pueden linealizarse y tienen que resolverse numéricamente, volviéndose más complejas. Sin embargo, se puede proyectar mediante un enfoque gráfico conocido como el Criterio de Áreas Iguales que establece un análisis comparativo de los ángulos del rotor antes, durante y después de la falla [8].

Con la estabilidad del voltaje pasa prácticamente lo mismo, buscando mantener un valor aceptable en todos los buses del sistema y equilibrando la potencia reactiva de los generadores. Para pequeñas perturbaciones, el sistema requiere de un análisis de estado estable que estudia el comportamiento del voltaje y la potencia reactiva. Cuando la sensibilidad V-Q es positiva para cada barra entonces el sistema es estable, pero cuando la sensibilidad V-Q es negativa en al menos una barra de todo el sistema entonces es considerado un sistema inestable [7].

La estabilidad se ve afectada de forma directa por las perturbaciones, sin embargo, difieren en función del tiempo que permanece la falla y de la amplitud de su forma de onda. La operación del sistema es estable cuando se las asocia a fallas pequeñas de corto tiempo de duración, dinámica cuando es de pequeña señal con tiempo largo de duración y transitoria para perturbaciones de gran amplitud en su señal. A continuación, se presenta la **Figura 2.1** que muestra un esquema que clasifica la estabilidad del sistema de potencia en presencia de fallas como variaciones leves en los niveles de voltaje de la red, conexión o desconexión de carga o interrupción de conductor en el sistema.

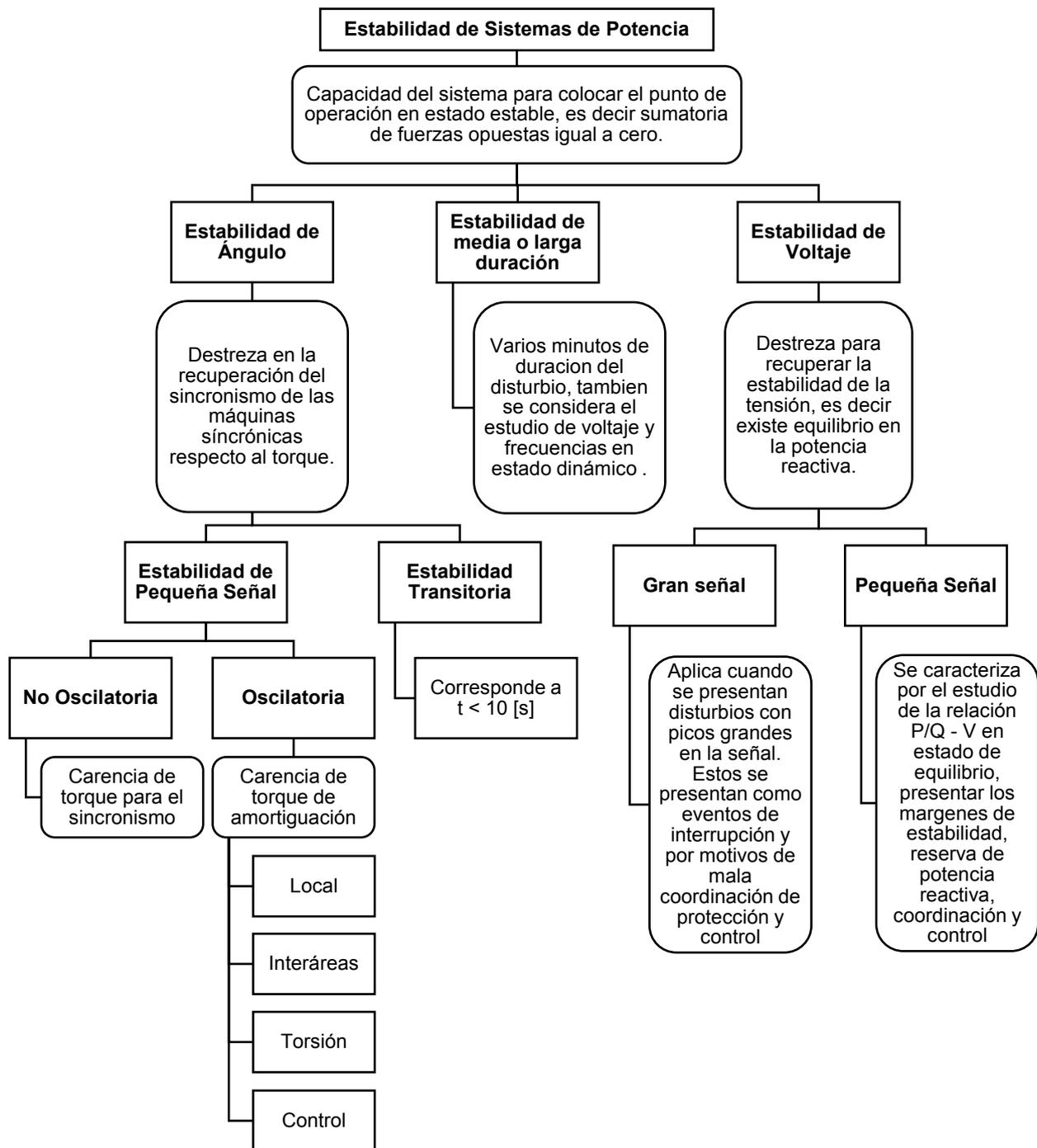


Figura 2.1. Clasificación de estabilidad en sistemas de potencia basado en [9]

La importancia de clasificar un fenómeno físico como la estabilidad está dada por las diversas formas en las que presenta al instante de generar un disturbio, dado que modifica la ubicación del punto de operación de las generadoras en el sistema. El tiempo de afectación y magnitud de amplitud del disturbio son parámetros que son estudiados y se ven reflejados por medio de gráficas de ángulo de rotor, velocidad, frecuencia, voltaje, potencia en función del tiempo. El diseño adecuado de sistemas de potencia es fundamental para librar de aperturas de elementos de protección, ya que este evento incide en el aislamiento de redes, mismas que en el peor de los casos generan sobrecargas, para posteriormente dejar sin servicio al consumidor y provocar daños en la infraestructura eléctrica [1].

Estabilidad de pequeña señal

Previamente se identificó el concepto de estabilidad como la habilidad del sistema de recuperar el estado estable de operación posterior a una perturbación. Esto es diferente para el caso de la inestabilidad, ya que se asocia a la deficiencia o carencia de sincronismo, que son vistas por las máquinas síncronas en el sistema. El tema en desarrollo de estudio para estas condiciones de operación en estado estable, constituyen la planificación de sistemas eléctricos de potencia a largo plazo, ya que se realiza la búsqueda de márgenes o cotas respecto a la carga conectada a las máquinas sincrónicas después de existir la carencia de sincronismo. El constante incremento de la demanda contribuye a la variación de ángulo de rotor de las máquinas sincrónicas, por ende, inciden en la inestabilidad del sistema.

La estabilidad de pequeña señal corresponde a perturbaciones con valores de amplitud pequeños que están presentes en el sistema frecuentemente y no aportan en su mayoría en la pérdida del sincronismo. Se debe considerar que, para que la estabilidad se vea afectada, debe de existir oscilaciones mínimas y amplitudes pequeñas, conociendo así a este estado como dinámicamente estable. Referente a lo que se menciona previamente, se considera que las oportunidades del sistema a encontrarse sometida a inestabilidad dinámica son mayores que las que existen en estado estable.

Una vez argumentado el comportamiento de las señales y los estados de operación que se presentan en los sistemas eléctricos de potencia, se define a la estabilidad de pequeña señal como la habilidad de los sistemas para lograr sostener el sincronismo frente a valores pequeños de oscilación y amplitud de la señal, característicos de la perturbación. Las pequeñas perturbaciones son proporcionadas comúnmente por variación de la carga o generados [6].

Existe el proceso de linealización de las ecuaciones que describen el sistema en el estado de afectación de una perturbación, es decir que condiciona al disturbio de tal forma que la amplitud sea pequeña y las oscilaciones cortas. Este tipo de fenómeno de inestabilidad, dada por disturbios de pequeña señal se presenta por condiciones de:

1. Déficit en el torque de amortiguamiento, dado que las oscilaciones muestran un aumento en la amplitud.
2. Déficit en el torque de sincronización, debido al crecimiento gradual del ángulo de rotor.

El estado de equilibrio visto en la **Figura 2.2**, es requerida por el sistema para una correcta operación de los elementos que la conforman, ya que este permite establecer valores de tensiones, frecuencia y potencia de manera teórica en una constante. Cabe mencionar que es de conocimiento general que la señal siempre presenta variaciones, pero son tan pequeñas que se aproximan a una constante.

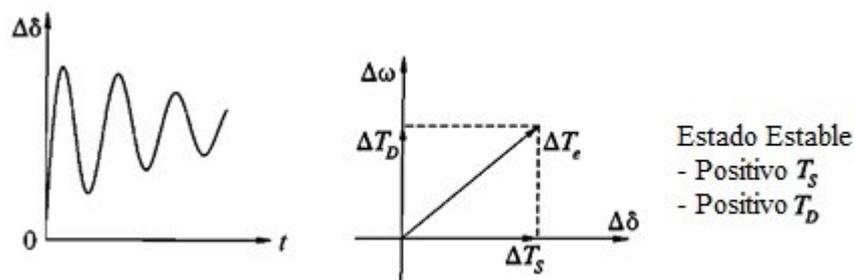


Figura 2.2. Estado de equilibrio con tensión de campo fijo [9]

El poco amortiguamiento que existe en las oscilaciones de las perturbaciones que afectan al sistema, comúnmente se relacionan con la inestabilidad de ángulo de rotor. Este proceso en donde se observa a la señal presentar irregularidad y constante variabilidad en la amplitud de su señal, a la vez predominante en el tiempo, puede ser regulado por medio de elementos de automatización o control de campo. En el caso de persistir el comportamiento inestable de la señal, este puede deberse a los márgenes establecidos en el campo que conllevan a la perturbación a superar los límites, generando así una inestabilidad no oscilatoria, este fenómeno se observa en la **Figura 2.3** [10].

El tema de alcanzar estabilidad en el sistema implica que se conozca los antecedentes de la señal, es decir las condiciones iniciales en su estado de operación, la confiabilidad y seguridad del sistema, y el control eléctrico en la excitatriz de las fuentes de energía. Esto se obtiene mediante el análisis del comportamiento de la señal de salida del sistema.

El análisis de estabilidad de pequeña señal cuenta con dificultades cuando no se garantiza el sostenimiento de las oscilaciones del sistema. Para evitar esto se considera el uso de reguladores de voltaje, que amortiguan la señal tal como se observa en la **Figura 2.4**. Se identifica el estudio de estabilidad de pequeña señal, en un rango de 10 a 20 segundos después de la aplicación de la perturbación [7]. La clasificación de estabilidad en la **Figura 2.1** muestra la variedad de ramas que surgen por la naturaleza de la perturbación aplicada al sistema.

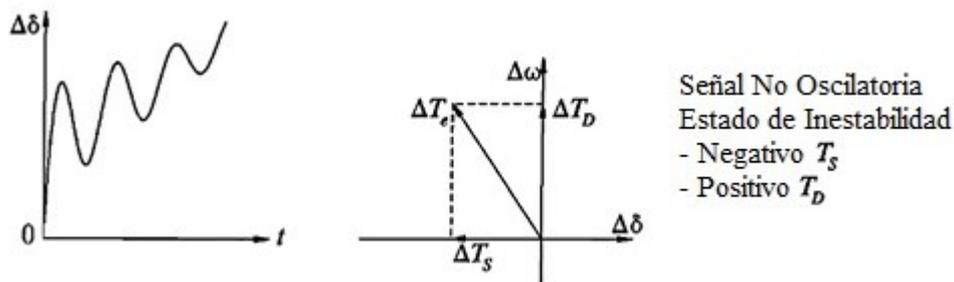


Figura 2.3. Estado inestable no oscilatoria con tensión de campo fijo [9]

La clasificación cuando existen inestabilidad provocadas por oscilaciones de amplitud pequeña se presenta a continuación:

1. Modo de oscilaciones de planta local: consideradas únicamente a las oscilaciones de centrales de generación, conociendo que se refiere con el termina local, debido a la especificación de la ubicación de la oscilación en la unidad de generación o en punto del sistema de potencia.
2. Modo interárea: Se identifica por las oscilaciones de un conjunto de generadores en un área oscilante, que repercuten con la interacción a otro grupo de generadoras localizadas en otra área. La característica principal está dada por el efecto que tiene la variación de carga en la estabilidad de este tipo.
3. Modo de control: relación existente de gobernadores de velocidad, convertidores HVDC y compensadores estáticos con las unidades generadoras, en términos de ser las causales de inestabilidad de este tipo.
4. Modo torsional: Se asocia a la parte móvil del generador, es decir los elementos que se encuentran en el eje del rotor, a este tipo de modo se le atribuye la inestabilidad debido a la acción mutua con los controles de excitación, gobernadores de velocidad, controles HVDC y compensadores en configuración serie [10].

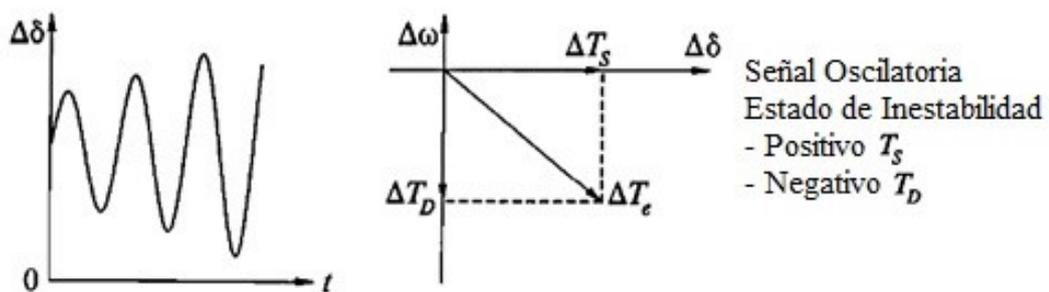


Figura 2.4. Estado inestable oscilatorio con control de excitación [9]

La estabilidad como referente de seguridad en la transmisión de la energía eléctrica constituye la regulación y control de las señales de salida, siendo esta la frecuencia, voltaje, potencia, etc. La amplitud, oscilación y tiempo de estabilización son los principales parámetros para considerar establecer una clasificación de la estabilidad.

2.3 Energías renovables

A lo largo de la historia, la vida cotidiana de la población mundial ha estado inmersa en la realización de actividades como trasladarse de un lugar a otro, alimentarse, hidratarse y muchas más, conscientes de la pérdida de energía que se necesita para sustituir, conlleva a la búsqueda de fuentes que le permitan satisfacer su necesidad de prevalecer sin limitarse de las tareas en su diario vivir. El proceso de desarrollo del ser humano, mediante la colonización, industrialización, boom tecnológico y la actualidad ha hecho el mundo más cómodo para el hombre en la realización de tareas, por otro lado, este crecimiento implica mayor consumo de recursos primarios para el funcionamiento y operación.

La crecimiento tecnológico y poblacional implica también incremento en fuentes de energía primaria, por ello a nivel mundial los principales aportadores de energía son los combustibles fósiles, que no son amigables con el ambiente, ya que son recursos no renovables, es decir que el periodo de tiempo de creación natural de dicho recurso es muy largo. Para exponer en términos de porcentaje el requerimiento total de energía por la población mundial, considerando variables como sector ocupacional del recurso y el tipo de combustible, esto se presenta en la **Tabla 2.1**.

Las cantidades encontradas deben considerarlas en unidades de Quad Btu equivalentes a billones de Btu, también el siguiente factor de conversión de unidades de Btu a kJ: 1 cuatrillón de Btu = 1×10^{15} Btu = 0.95×10^{15} kJ [11].

La **Tabla 2.1** indica que el tipo de combustible proveniente de energías renovables ocupa el 25% del aporte total en la generación eléctrica en el mundo, solo por detrás del carbón que tiene un aporte del 40% del total. A partir de estos datos ya podemos inducir un auge a futuro de las energías renovables dentro de la matriz energética.

Se observa en la **Figura 2.5** que el sector industrial del 2017 representa en términos de consumo de energía a nivel mundial más del 50%, definitivamente el acelerado crecimiento de la población requiere la mecanización y automatización de procesos para obtener alimentos en grandes cantidades, siendo esta es la raíz de la industrialización. Eventualmente el crecimiento en la demanda de recurso energéticos es eminente. El sector de transporte en segundo lugar con un consumo aproximado a la cuarta parte (25%) de la energía total, en tercer lugar, al sector residencial con un valor aproximado del 7% y finalmente en el cuarto lugar a el sector comercial con un 13%.

La búsqueda de nuevas fuentes de energía y procesos que minimicen el consumo de recursos no renovables radica en la investigación, optimización de procesos e inculcar un pensamiento verde, en donde prevalezca el bienestar y cuidado ambiental sobre toda nueva forma de desarrollo social en el mundo.

Tabla 2.1. Consumo Mundial de Energía [12]

Combustible	Total	Electricidad	Residencial	Comercial	Industrial	Transportación	Todos los sectores de uso final
Petróleo	196.7	6.7	8.5	3.9	69.7	106.7	188.8
Gas natural	130.7	44.0	20.6	8.8	53.2	4.1	86.6
Carbón	159.8	90.5	4.3	1.6	63.5	0.0	69.4
Nuclear	26.9	26.9	-	-	-	-	-
Electricidad	-	-	21.0	16.8	35.1	1.6	74.5
Renovables	74.9	55.5	1.3	0.2	17.9	-	19.4
Total	589.0	223.6	55.7	31.3	239.4	112.4	438.8

En la **Figura 2.6** se presenta el porcentaje de suministro mundial de energía del 2017 en función del tipo de combustible, considerando el petróleo con un 33% como el mayor recurso primario consumido de manera global, en segundo puesto tenemos el carbón con el 27%, tercer lugar el gas natural con el 22%, el cuarto lugar es para las energías renovables con el 13% y finalmente el 5% para la energía nuclear.

La incidencia de las energías renovables a nivel mundial se encuentra en crecimiento, debido a la problemática global del cambio climático. Se considera que las fuentes de energías renovables son el camino para disminuir el exceso de CO₂ y gases de efecto invernadero que causa grietas en la capa de ozono, principal mecanismo de defensa de la radiación solar.

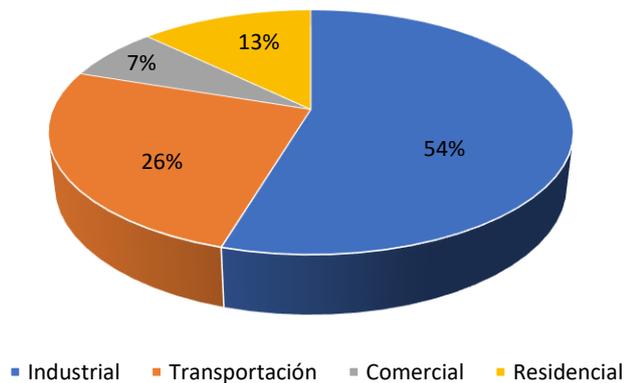


Figura 2.5. Consumo mundial de energía según el sector de ocupacional [12]

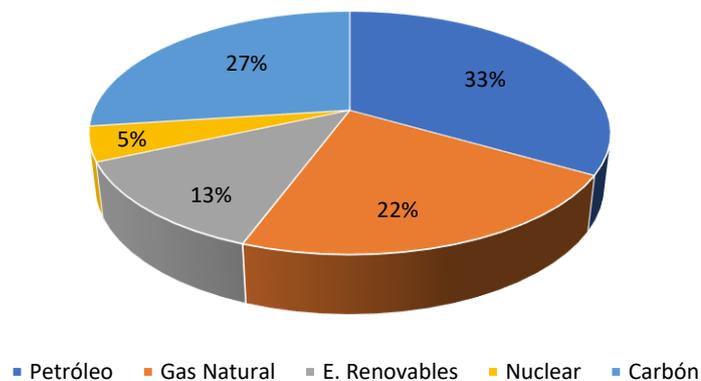


Figura 2.6. Energía generada a nivel mundial según el tipo de combustible [12]

El proceso de instalación del mecanismo o infraestructura de las fuentes renovables considera, niveles o concurrencia de vientos para el caso de una central eólica, mareas en una central mareomotriz, caudal y reservorio en una central hidroeléctrica, etc. Estos factores, así como el económico y despacho de las generadoras son esenciales en la toma de decisión para la ejecución de proyectos. Es por ello, que se dificulta en muchos países aplicar esta medida de contingencia frente al uso excesivo de los recursos no renovables.

La **Figura 2.7** muestra el incremento anual en la capacidad instalada de energía renovables a nivel mundial, esto considerando el periodo de 2015 al 2019, esta información es proporcionada por la “International Renewable Energy Agency” (IRENA).

Es importante identificar que existen causales de las energías renovables que conocemos en la actualidad, es decir si obtenemos energía térmica y fotovoltaica, es a partir del sol, ya que este emite radiación en forma de calor que se aprovecha y transforma en las antes mencionadas formas de energía. En el caso de la energía eólica, también el sol incide en los flujos de vientos, siendo el caso de energía hidroeléctrica el recurso es el agua, pero este se obtiene mediante el ciclo del agua, en donde el sol es principal responsable del cumplimiento del ciclo.

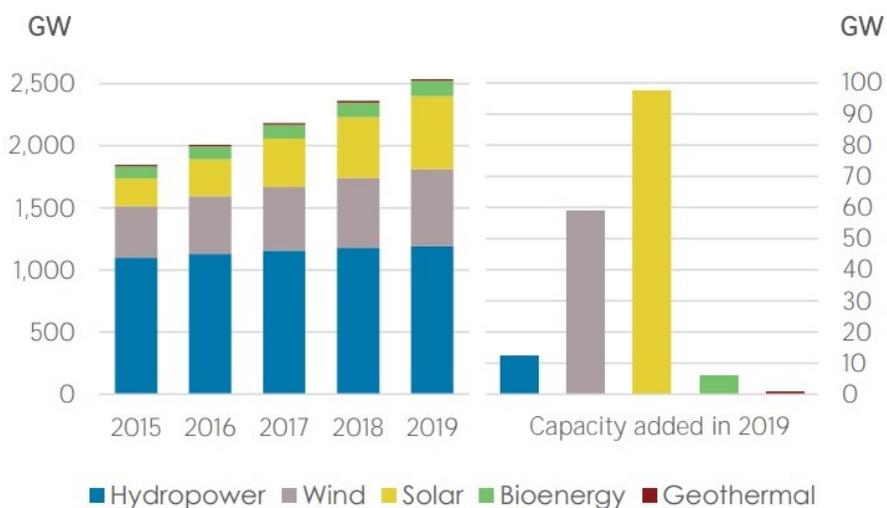


Figura 2.7. Incremento en la capacidad instalada de energías renovables [13]

La biomasa se la obtiene mediante la descomposición de la materia orgánica, que así mismo el sol es el responsable del ciclo de vida y fotosíntesis. Es evidente que el sol contribuye en su mayoría en la generación de las fuentes renovables, pero no es la única, esto se puede observar para el caso de la energía mareomotriz proveniente de las olas, la misma está dada por la fuerza gravitatoria existente de la luna y sol sobre la tierra y finalmente tenemos a la energía geotérmica brindada por el calor del núcleo interno de la tierra.

2.3.1 Energía solar fotovoltaica

La energía que emite el sol es esencial para la existencia, cumplimiento del ciclo de la vida y la sostenibilidad entre los sistemas naturales, ya que estos entornos se establecen debido a la incidencia del sol de manera positiva en una temperatura apropiada en la superficie de la tierra. Cabe mencionar que el sol es tan apropiado que emite el 95% de su energía en estado de luz visible, con respecto a otras estrellas que liberan su energía en su estado de rayos X.

La transformación de la luz solar que irradia la superficie de la tierra en electricidad se da mediante la intervención de un dispositivo conocido como célula fotovoltaica. Los fotones de energía arriban en grupos que son transportados por la irradiación de luz solar que viaja desde el sol hacia la tierra. La importancia de obtener estos grupos de fotones está dada en el aprovechamiento de sus características, siempre y cuando interactúe con una célula fotovoltaica mediante reflexión, absorción o traspaso. La obtención de electricidad a partir de la luz solar se da cuando las células fotovoltaicas absorben fotones [14].

La integración de los sistemas fotovoltaicos en los sistemas eléctricos es bien visto por la comunidad mundial, por su aporte al cuidado ambiental. El modelo de un sistema fotovoltaica parte desde la operación del generador fotovoltaico, considerado como un elemento que transforma la tensión que ingresa de corriente continua (CC) a una tensión de corriente alterna (CA) de una fuente síncrona permitiendo satisfacer la demanda en el sistema con potencia activa (P) y reactiva (Q). La señal de voltaje que se da como resultado es regulable en parámetros de la frecuencia fundamental como lo son el valor de la amplitud y el ángulo de fase.

Es importante recalcar que la energía proporcionada por la luz solar que incide en los paneles fotovoltaicos es variable, pero su penetración va en incremento en el día, ya que existe mayor interacción del sol con la tierra. El generador eléctrico repercute en la estabilidad y seguridad de la red eléctrica, esto ocurre por el incremento en la penetración de generación fotovoltaica. Para ello es razón suficiente para establecer lugares propicios en donde se minimice este efecto [15].

A continuación, se describen las componentes esenciales para la correcta operación de un sistema fotovoltaico, mismos que se observan en la **Figura 2.8**.

1. Arreglo fotovoltaico: se considera a la conexión de un grupo de módulos fotovoltaicos, que, al enlazar el cátodo de un panel con el ánodo de otro panel, se establece la configuración serie, contribuyendo así a sumar las tensiones.
2. Los inversores: se encargan de transformar la energía de corriente (CC) proporcionada por las baterías a energía de corriente alterna (CA), que es utilizada para satisfacer la demanda del sistema. La variedad de inversores se da en función de la magnitud de carga a suministrar.
3. Baterías: es necesaria para la reserva de energía solar irradiada en el día, y así aprovecharla en horas de la noche, comúnmente la batería de plomo-acido es utilizada por el tipo de consumidor residencial [16].

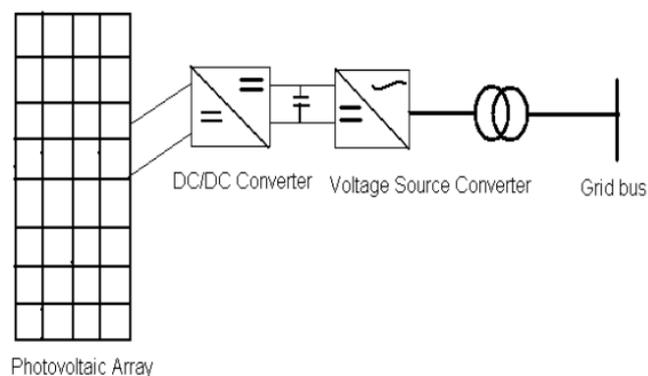


Figura 2.8. Conexión a la red de un generador fotovoltaico típico [15]

4. Regulador de carga: este evita que exista sobrecarga en las baterías, considerando los niveles de carga que recepta los módulos fotovoltaicos. Los reguladores permiten ajustarse con respecto a la variación de carga en las baterías, es decir el caso en un día por la noche el regulador permite suministrar energía a la carga, por otro lado, en el día este recibe energía y la almacena en la batería de tal manera que se mantenga siempre cargada.

La energía solar fotovoltaica es una fuente limpia de energía, debido a su bajo impacto ambiental, específicamente en disminuir la emisión de gases de efecto invernadero, esto cuando se realiza el proceso de generación de la energía eléctrica. Se debe considerar que los elementos pueden ser usados de manera particular, es decir la adquisición del sistema de conversión de energía solar a eléctrica también está dirigida a personas naturales. El tipo de conexión que es posible realizar a la tecnología modular se presenta integrada a la red de distribución general y tipo isla (aislado de la red).

2.3.2 Energía eólica

La energía del viento a lo largo de la historia fue de utilidad para las comunidades o particulares en actividades como, moler granos, impulsar navíos o bombear agua. La implantación de aerogeneradores surge a partir de la escasez de recursos energéticos de 1970 [11]. El proceso de obtención de energía eléctrica a partir de vientos se da mediante la transformación de energía cinética proporcionada por el flujo de viento en los alerones del aerogenerador a energía eléctrica mediante la inducción y conmutación del generador.

La diferencia del aerogenerador con respecto a un molino de viento radica en la transformación de energía, ya que el molino de viento a partir de la energía cinética obtiene trabajo mecánico. Los sistemas eléctricos en la actualidad cuentan con alta y pequeña penetración de fuentes de energías renovables, en gran parte es energía eólica y fotovoltaica, claro está que esto es dependiente de la geografía a nivel mundial dado la afluencia de recurso, es decir existe países en donde el flujo de viento es más frecuente y de mayor magnitud, así también la

energía fotovoltaica, zonas como oriente medio son el caso. La implantación de esta tecnología en los sistemas eléctricos es común encontrarlos en capacidades menores o iguales a 10 kW. Para el caso de generación en grandes magnitudes tenemos un rango de 1.5 a 5 MW [17].

La turbina eólica HAWT de eje horizontal, dado que el eje es paralelo en referencia al suelo. En la actualidad es la más utilizada alrededor del mundo cuando se habla de generación eólica. Los rotores tipo HAWT se clasifican en función de la orientación del rotor, diseño de la caja, sistema de control del rotor, número de palas y desplazamiento de estas. La **Figura 2.9** representa un esquema general de red con generación eólica.

A continuación, se presenta los principales elementos que conforman el conjunto de un aerogenerador típico, que se presentan en la **Figura 2.10**.

1. El rotor está integrado por las palas y la caja.
2. El tren de transmisión está integrado por ejes para el giro, control de velocidad. Acople, freno.
3. La góndola y el bastidor principal, en donde se observa conjuntos como la carcasa, bancada y el sistema de guiñada.
4. La torre y los cimientos.

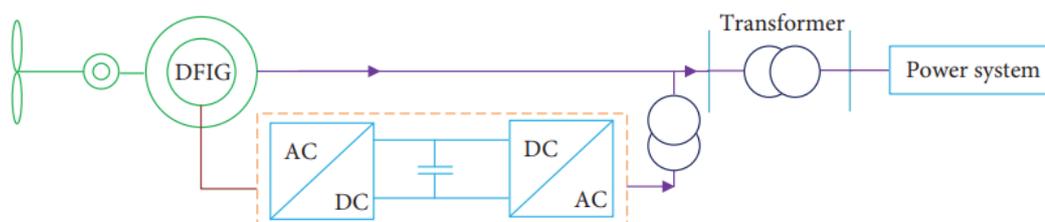


Figura 2.9. Elementos conectados al sistema eólico [18]

5. La máquina controla.
6. El equilibrio del sistema eléctrico está integrado por incluidos cables, paramenta, transformadores y convertidores de potencia en la mayoría de los casos.

La energía eólica presenta mayor penetración en los sistemas de potencia que la energía solar fotovoltaica, esto debe al área de ocupación. El sistema eólico cuenta con la facilidad de retomar su estado natural, cuando se desea retirar el sistema de generación de la locación.

2.4 Método de optimización por enjambre de partículas (PSO)

Kennedy y Eberhart en el año de 1995 desarrollaron el método de optimización por enjambre de partículas, mediante el estudio del comportamiento social de individuos en una población específica, dicho de manera general. La sincronización y la estética de las aves en su vuelo fueron inspiradores y detonantes en el interés sobre encontrar una relación matemática que implique tomar decisiones en forma conjunta. El método se considera global y heurístico presentado como algoritmo de optimización [3].

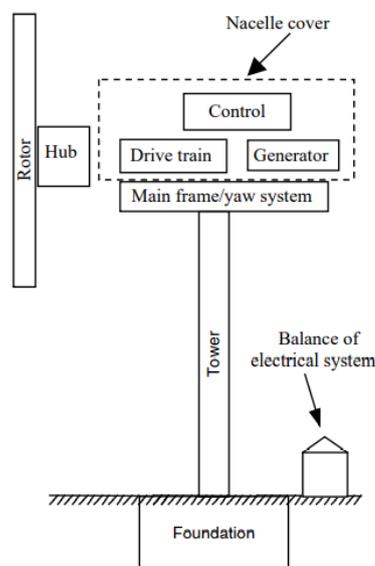


Figura 2.10. Principales componentes de la turbina eólica de eje horizontal [17]

El método PSO se encuentra en constante avance investigativo, debido a su naturaleza y la efectividad al momento de determinar la mejor solución, cabe mencionar que es aplicable a cualquier campo de estudio en donde surge la necesidad de optimizar. Los campos de aplicación del método radican con mayor frecuencia en diseño e inteligencia computacional. En la actualidad la búsqueda de mejora y potenciación del método, proporcionan al menos veinte y cuatro opciones de PSO, sin contar también los algoritmos híbridos que relacionan al método PSO con otros algoritmos de optimización que se encuentran sujetos a investigación [19].

El proceso que utiliza el método de PSO para encontrar la mejor solución se enfoca el vector de velocidad de la partícula, en donde se verifica en primera instancia soluciones locales (solución débil) alrededor de la partícula, seguido procede a tomar la decisión para la selección de la mejor solución local (solución débil), para así poder comunicar el mensaje a las demás partículas, es decir las partículas ven lo que las otras han obtenido al desplazarse. Finalmente, como tercer suceso en el camino a la mejor solución se tiene la inercia de la partícula, que indica a cada una, que proceda a buscar más allá de una buena solución. La utilidad del método de PSO es frecuente en diferentes campos de estudio, esto se debe a la simplicidad de cálculo, la eficacia en la búsqueda y no está inmersa en cálculos de superposición y transformación. La posición de cada partícula se ve modificada en función de lo que ve cada partícula en la búsqueda de la mejor solución [2].

$$V_i^{k+1} = \omega V_i^k + a_1 r_1 \times (P_{bi}^k - X_i^k) + a_2 r_2 \times (G_{bi}^k - X_i^k) \quad (2.1)$$

Se definen las variables de la **Ecuación 2.1** de la siguiente manera:

P_{bi} : mejor partícula local,

G_{bi} : mejor partícula global,

ω : función de peso de inercia,

a_1, a_2 : constantes de aceleración y

r_1, r_2 : valores aleatorios entre 0 y 1.

La **Figura 2.11** representa el comportamiento de una partícula, considerada en el método PSO. El desplazamiento que se observa se manifiesta a partir de la

identificación de la mejor posición actual global (g^*) y a la vez a su mejor posición históricamente (x_i^*).

La **Ecuación 2.2** representa la actualización tanto de la partícula como la posición, en función de la memoria de las partículas cercanas. El espacio en el cual actúan las partículas en la búsqueda de la mejor solución es de origen multidimensional [2].

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2.2)$$

El conjunto limitado para la búsqueda D-dimensional, cuenta con la integración de partículas (posibles soluciones) que describen la actuación del enjambre o individuos interactuando de manera colectiva. El proceso de actualización de una partícula se da mediante la recopilación de nueva información que se presenta en la búsqueda por medio de las variables de posición en la d-ésima dimensión de la i-ésima partícula, representada como X_i^d , vista en la **Ecuación 2.3** y la variable de velocidad en la d-ésima dimensión de la i-ésima partícula, representada como V_i^d vista en la **Ecuación 2.4**. A continuación, se describe el modelo matemático de actualización de las variables [20].

$$X_i^d \leftarrow X_i^d + V_i^d \quad (2.3)$$

$$V_i^d \leftarrow V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (gbest^d - X_i^d) \quad (2.4)$$

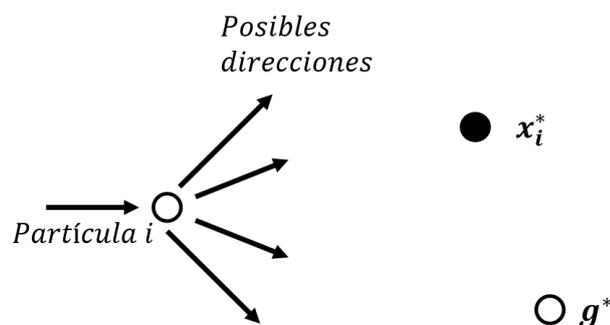


Figura 2.11. Partículas en la búsqueda de su solución óptima [19]

De manera general se describen las variables en las ecuaciones como:

$X_i = (X_i^1, X_i^2, \dots, X_i^D)$: Posición de la i -ésima partícula;

$V_i = (V_i^1, V_i^2, \dots, V_i^D)$: Velocidad de la partícula i ;

$pbest_i = (pbest_i^1, pbest_i^2, \dots, pbest_i^D)$: Mejor posición anterior que genera el mejor valor para la i -ésima partícula;

$gbest = (gbest^1, gbest^2, \dots, gbest^D)$: Mejor posición encontrada por la población.

c_1 y c_2 : Las constantes de aceleración que reflejan la ponderación de los términos de aceleración estocástica que muestran de manera individual a las partículas que deben dirigirse hacia $pbest$ y $gbest$ posiciones, respectivamente.

$rand1_i^d$ y $rand2_i^d$: Valores aleatorios en el intervalo de $[0, 1]$.

Los valores aleatorios que se generan en el intervalo de $[0, 1]$ difieren en función de la actualización de la variable de velocidad, mediante la **Ecuación 2.4**. Es frecuente encontrar la aplicación de la **Ecuación 2.5** en el campo científico para la actualización [20].

$$V_i^d \leftarrow V_i^d + c_1 * rand1_i * (pbest_i^d - X_i^d) + c_2 * rand2_i * (gbest^d - X_i^d) \quad (2.5)$$

La **Ecuación 2.4** y **Ecuación 2.5** son diferentes, considerando que tiene el mismo objetivo de actualizar información concedida durante la búsqueda de la mejor solución. Los aspectos diferenciales son planteados en la **Tabla 2.2**.

Tabla 2.2. Comparación de Variantes del Método PSO [20]

Ecuación 2.4	Ecuación 2.5
<ul style="list-style-type: none"> ▪ La dimensión se actualiza independientemente, por ende, la Ecuación 2.4 considera un espacio mayor para la búsqueda. ▪ La eficiencia cuando se trata de problemas no rotados es considerable. 	<ul style="list-style-type: none"> ▪ Los valores aleatorios son los mismos para todas las dimensiones, dado que está ligado a la dimensión y un espacio pequeño de búsqueda. ▪ La coherencia y robustes en la aplicación a problemas rotados y no rotados es buena.

La **Ecuación 2.4** presenta características de espacio de búsqueda y confiabilidad como agente de actualización de partículas, se recomienda para el uso en la optimización de problemas. A continuación, en la figura se muestra el diagrama de flujo del método PSO, considerando el modelo matemático descrito en la **Ecuación 2.4** [20].

Se definen las variables que se encuentran en el diagrama de flujo presentado en la **Figura 2.12**.

ps : tamaño de población;

k : contador de generación desde 1 a max_gen : (generación máxima);

d : dimensión;

w : Peso de inercia ($w_0 = 0.9$, $w_1 = 0.4$);

max_gen : generación máxima;

i : contador de id de partículas de 1 a ps ;

x_i^d : i^{th} : valor de las dimensiones de las partículas

$c_1 = c_2 = 2$

El análisis del espacio de búsqueda dentro de la función objetivo puede llegar a ser muy extensa a la hora de encontrar la solución óptima. Por esto también se puede trabajar con regiones específicas del espacio de manera que se agilice la búsqueda completa dentro del dominio de la función. Este proceso viene dado por el método de Optimización de multi-enjambre, que trabaja con sub-enjambres de partículas enfocados en una región específica del espacio de búsqueda. La principal diferencia con el único enjambre de partículas utilizado en el método PSO es que los sub-enjambres actúan a la orden del método de diversificación que ordena a cada subconjunto de partículas cuando y donde trabajar en su búsqueda. La Optimización por multi-enjambres puede agilizar la búsqueda de la solución siempre y cuando se conozca el conjunto del espacio en el cual se encuentra nuestra función objetivo. Esta variante del método PSO se vuelve más efectiva a la hora de trabajar con problemas multi-nodales ya que puede llegar a encontrar múltiples soluciones óptimas locales en las diferentes regiones del espacio de manera más rápida.

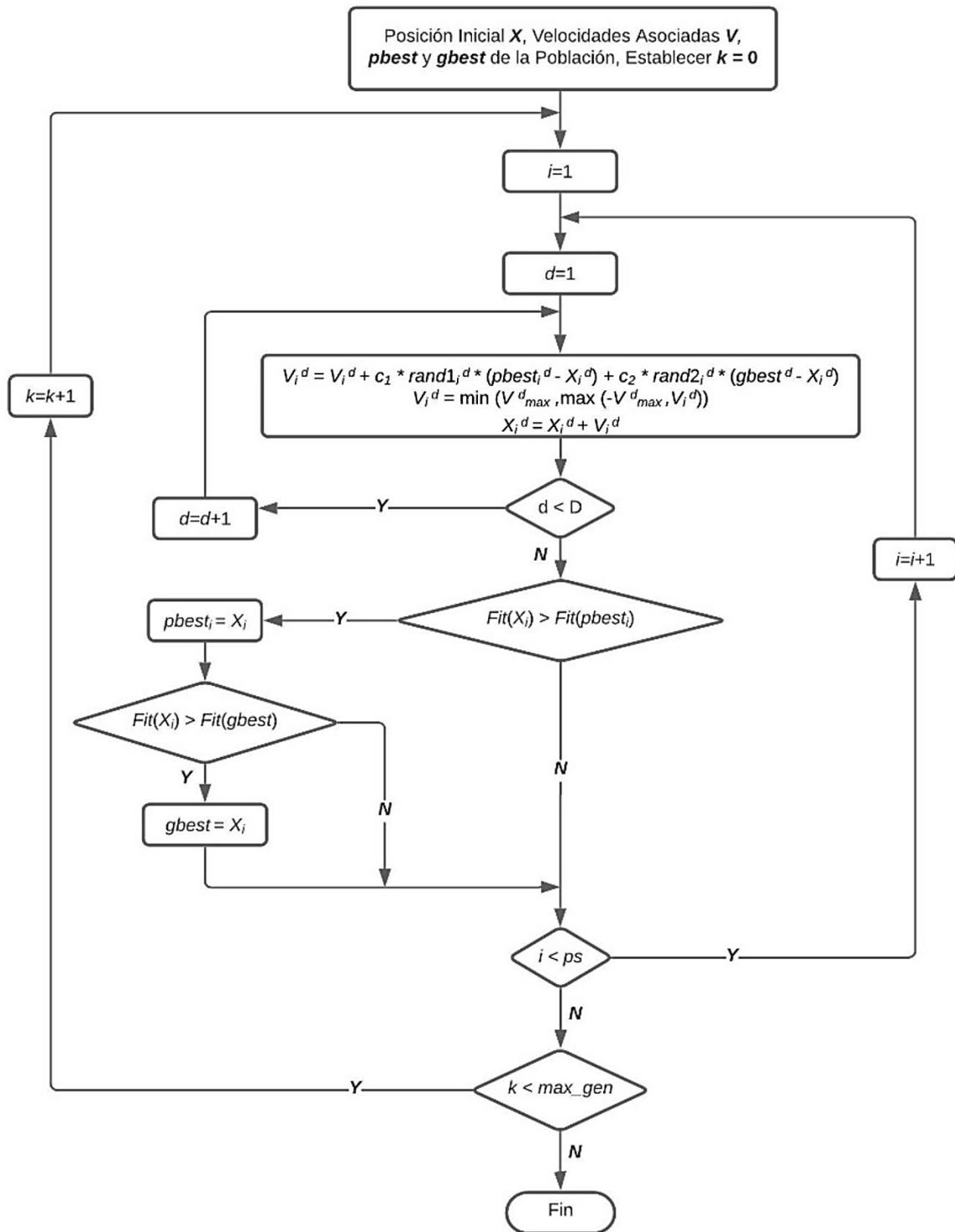


Figura 2.12. Diagrama de flujo del método convencional de PSO basado en [20]

2.5 Control

La seguridad del sistema eléctrico de potencia es esencial para el aprovechamiento óptimo de la energía eléctrica en los lugares de consumo, para ello se presenta mecanismos de control de excitación para los generadores síncronos, mismos que presentan inestabilidad luego de exponerse a perturbaciones.

2.5.1 Modelo PSS

El Estabilizador de Sistemas de Potencia (PSS por sus siglas en inglés) es un controlador de retroalimentación, parte del sistema de control de un generador síncrono, que actúa a través del sistema de excitación, agregando una señal para modular el voltaje de campo. La función principal del PSS es amortiguar las oscilaciones electromecánicas del rotor del generador, que se encuentran en una amplia gama de frecuencias en el sistema de potencia. La amortiguación del PSS se produce mediante la adición de par eléctrico, en fase con las desviaciones de la velocidad del rotor, que contrarreste la dinámica mecánica del generador. Para lograr esto, el PSS debe compensar la diferencia de fase entre la entrada del sistema de excitación y el par eléctrico [21].

2.5.2 Modelo AVR

El regulador Automático de Voltaje (o AVR por sus siglas en inglés) es un controlador que se encarga de mantener la magnitud de voltaje terminal del generador síncrono en un nivel específico para asegurar la estabilidad del sistema. El AVR opera en el campo del excitador, nivelando el voltaje terminal del excitador y la corriente de campo, permitiendo el control de la potencia reactiva del generador síncrono. La nivelación de voltaje se logra mediante la medición de tensión en los bornes del generador, que se filtra para poder obtener una señal de corriente que mantenga el suministro de energía constante y confiable. El voltaje rectificado es comparado con un valor de referencia y la diferencia es utilizada como alimentación para la excitación del generador. Para lograr la estabilidad, el sistema de control de voltaje debe tomar la decisión de control basándose en el error de voltaje solo si no hay desviaciones de velocidad del rotor. En el caso de que existan oscilaciones en la velocidad del rotor, el sistema de control de voltaje

debe producir una señal de control basada en el error de voltaje y en una señal adicional del PSS.

2.6 Lenguaje de programación y software de estudios eléctricos

La interpretación del computador, en referencia a los eventos en el mundo real son codificados mediante el lenguaje de programación. El proceso consiste en conocer comandos o funciones básicas que sirvan de herramientas para la conexión hombre – máquina. Los algoritmos representan el medio lógico, a partir de instrucciones en secuencia lógica para el desarrollo de programas que automaticen y controlen procesos. El software representa el resultado de la programación mediante un lenguaje de programación. El caso específico de DigSILENT-PowerFactory enfocado en la simulación de estudios eléctricos y actividades de pruebas referente al sector eléctrico, presenta un entorno de interpretación gráfico de ingreso de datos y selección de opciones que permiten la realización de una actividad específica requerida por el usuario.

2.6.1 Python

Python es un lenguaje de programación que se define como código legible para cualquier usuario con un mínimo conocimiento en programación. Considerado como uno de los lenguajes de programación más importante, Python es administrado por Python Software Foundation y fue desarrollado en el año 1991 por informático Guido van Rossum. Es un lenguaje de código abierto, es decir, gratuito en su totalidad, convirtiéndolo en el lenguaje más popular del medio de la informática y la programación.

Python es compatible con cualquier versión de Windows o Linux debido a que tiene muchos intérpretes disponibles. No presenta limitaciones ya que su uso se extiende en diferentes áreas, permitiendo el desarrollo de aplicaciones de todo tipo, inteligencia artificial, método de optimización de sistemas informáticos, manejo de variables, etc.

Entre sus principales funciones se encuentran [22], [4]:

1. Big Data, manteniendo una amplia gama de librerías que constantemente se están actualizando, convierten a Python en el lenguaje más ideal a la hora de analizar variables y datos en tiempo real.
2. Almacenamiento de información que permiten al usuario analizar el comportamiento de datos y así poder predecir sucesos futuros. Usado en el área de la optimización de procesos, para analizar y gestionar las variables de manera que se obtengan los datos de salida deseados.
3. Avances en la Inteligencia Artificial debido a su sencillez de escritura, que permiten plantear ideas complejas en pocas líneas de código.
4. Organización, manipulación y estructuras de datos como listas, conjuntos y tuplas.
5. Basta gama de módulos de programación que cubren necesidades básicas como también específicas del programador.
6. Implementación de máquinas virtuales en entornos de desarrollo integrado para la creación y desarrollo de aplicaciones. Ejemplos populares como IDLE (Integrated Development Environment) y Eclipse que mantienen la sencillez de la programación sin dejar de lado el desarrollo de códigos inteligentes.

Además de su libre acceso y uso, también se caracteriza por no tener restricciones de copyleft, es decir que se pueden producir programas sin la necesidad de liberar el código. Esto conlleva a que pueda usarse tanto para desarrollar softwares de libre acceso como también de acceso privados para las instituciones o usuarios que lo requieran [22], [4].

2.6.2 DigSILENT-PowerFactory

PowerFactory es un software de diseño especializado en el campo de la ingeniería eléctrica, proveniente de la empresa DigSILENT GmbH, que brinda un servicio

aplicativo en el análisis de la transmisión, generación y distribución de energía, así como también en todo lo referente a plantas industriales y energías renovables. Su diseño consta de una interfaz avanzada e interactiva que permite al usuario planificar y optimizar sistemas de potencia mediante simulaciones en tiempo real.

Considerado como una de las herramientas más completas en el campo de la ingeniería eléctrica, DigSILENT-PowerFactory atiende y cubre todas las necesidades que poseen los sistemas eléctricos, permitiendo modelar estructuras complejas con parámetros exactos como generadores, líneas de transmisión, transformadores, fuentes de energías renovables, cargas, etc. También se permite medir y controlar la naturaleza compleja de dichas estructuras ya que tiene la opción de simular flujos de carga, eventos de fallas y demás simulaciones en tiempo real. Cada evento definido en el software se almacena en una base de datos que después puede utilizarse para realizar gráficas, analizar el historial de cambios del sistema, predecir el comportamiento de variables o incluso importar y exportar datos.

Características

DigSILENT-PowerFactory incluye un paquete muy amplio de funciones integradas de ingeniería como lo son [23], [24].:

1. Diseño y modificación de diferentes componentes incluidos en sistemas eléctricos.
2. Gráfica y manejo de eventos, de manera integrada e interactiva.
3. Almacenamiento de base de datos de simulaciones en tiempo real, así como la importación y exportación de datos en formato texto o archivos Excel para poder visualizar el historial de cambios.
4. Cálculos de parámetros de los elementos, modificación y variación de datos, flexibilidad en la creación y uso de hojas de datos.

5. Control sobre sistemas eléctricos, visualización de salidas y adaptación automática de modelos.
6. Plantear diferentes eventos en tiempos definidos. Eventos de fallas múltiples, apertura y cierre de switches, aumento y disminución de carga, etc.

Simulación y funciones gráficas

El DigSILENT-PowerFactory permite definir diferentes estudios de casos en la simulación de sistemas eléctricos. Los datos que se obtengan de estos escenarios se almacenarán en una base de datos conformada por carpetas jerárquicas que se mantienen constantemente actualizadas. Esto significa que cuando se cambia algún dato específico, automáticamente se verá reflejado en todos los proyectos y datos asociados, permitiendo una simulación en tiempo real del sistema.

La base de datos posee características consistentes, pero a la vez flexibles, por lo que se puede almacenar prácticamente todo sin presentar pérdidas de control. De esa manera, todos los proyectos asociados pueden hacer uso de esta base de datos sin la necesidad de tener que acceder a los ajustes del sistema nuevamente.

La base de datos de DigSILENT-PowerFactory puede usarse como objeto gráfico, permitiendo al usuario visualizar el comportamiento de variables a estudiarse. Los gráficos permiten observar las estaciones topológicas y diferentes resultados de cálculos suscitados en el sistema simulado, en donde se permite modificar y plantear nuevos eventos acorde a los resultados obtenidos para mejorar o realizar diferentes clases de pruebas. Mediante la sección de Instrumentos Virtuales, el usuario puede acceder al diseño de gráficos para analizar resultados de estabilidad, desgramas de bloques en circuitos lógicos, caja de valores, etc. [23], [24].

2.6.3 Pycharm community edition

PyCharm es un entorno de desarrollo integrado (IDE) con una amplia gama de herramientas para el lenguaje de programación Python. Desarrollada por la empresa JetBrains, la versión de código abierto de PyCharm fue lanzada en octubre de 2013 como PyCharm Community Edition. Proporciona un entorno conveniente para los desarrolladores de Python con ventajas en la finalización del código inteligente, inspecciones del código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automática y completas funcionalidades de navegación [25].

Características

PyCharm funciona como un asistente inteligente de Python de manera que facilite y mejore el entorno de trabajo de los desarrolladores de Python. Entre sus características se encuentran [25]:

- Asistencia en el análisis de código fuente, identificación de errores y corrección de la sintaxis.
- Navegación de código, vista estructurada, búsqueda rápida dentro del código y herramientas de refactorización.
- Interfaz de usuario y esquemas de teclado personalizable, accesos rápidos a diferentes funciones de PyCharm.
- Rápida navegación entre proyectos, visualizar y trabajar con varios archivos de manera simultánea, enlace entre proyectos y códigos.
- Depurador visual, inspección de variables locales, definición de condiciones de coincidencia.
- Multiplataformas; trabaja con versiones de Windows, macOS y Linux.
- Ofrece la versión de código abierto PyCharm Community Edition.

Licencia y complementos

El proyecto se realizará con la versión gratuita PyCharm Community Edition, distribuida bajo la licencia Apache 2, con la disponibilidad de acceder al código fuente completo en GitHub [26].

PyCharm otorga libertad a los desarrolladores para crear sus propios complementos y de esa manera se logre ampliar las funciones de PyCharm. Actualmente, Pycharm cuenta con más de 1000 complementos disponibles, incluida la integración con herramientas y marcos de trabajo variado, acceso a archivos CSV/TSV/PSV y compatibilidad con otras IDE desarrolladas por JetBrains [27].

CAPÍTULO 3

3.1 Metodología

El Capítulo 3 describe las etapas de desarrollo del proyecto, mediante las herramientas computacionales tanto de DigSILENT-PowerFactory como software especializado en estudios eléctricos y Python para la programación. La solución parte del estudio de pequeña señal realizado a un sistema eléctrico de 6 barras implementado en DigSILENT-PowerFactory, seguido se establece la relación que existe entre DigSILENT-PowerFactory y Python en términos de código fuente para la programación, luego se procede a determinar el lugar óptimo de la generación fotovoltaica mediante la aplicación del método metaheurístico PSO. La automatización de todo el proceso antes mencionado se implementa para el caso del sistema eléctrico modificado IEEE de 39 barras, considerando que la herramienta programada debe servir para n barras en el sistema.

3.2 Estudio de estabilidad de pequeña señal en PowerFactory.

El proceso de búsqueda para la mejor opción de ubicación de generación eólica y fotovoltaica en el sistema propuesto para estudio eléctrico se da mediante la apertura de la aplicación DigSILENT-PowerFactory, en donde se importa el archivo del sistema eléctrico sujeto a evaluación para luego activar el proyecto en el programa, cabe mencionar este proceso es previo al uso o ejecución del código fuente en Python.

La importancia de ajustar previamente valores o caracteres, que son generales de la red o estándares de operación, siendo estos los nombres de los generadores eólicos y fotovoltaicos como “PV_Sys#” y “StaGen#”, tales que, el símbolo numeral representa el número asignado desde el 1 a n elementos de sus respectivas clases, también ocurre para la frecuencia a la que se desea operar el sistema, esto depende de la región a nivel mundial en donde se ejecute el proyecto.

El siguiente paso una vez guardado el proyecto, se procede a cerrar DigSILENT-PowerFactory para luego abrir el entorno de programación integrado con sus siglas

en inglés (IDE). El desarrollo del código fuente está dado en PyCharm Community, mismo que corresponde a uno de un sin número de IDE's de Python versión 3.8, cabe mencionar que tanto PyCharm Community y Python son softwares de código libre. El código fuente hace referencia a los algoritmos implementados para realizar un conjunto de operaciones, que para el caso planteado en el proyecto se enlaza las funciones, casos de estudio, elementos de tal manera que se automatiza el proceso de identificación de las barras, en donde se desea tentativamente ejecutar el proyecto de construcción de centrales de generación eólica u fotovoltaica. Las limitaciones están dadas por la afluencia del recurso en el lugar, por tal razón no es recomendable colocar fuentes en todas las barras del sistema.

La identificación de la mejor posición se da mediante la integración del método PSO en el código fuente, este dirige a las partículas agregadas en la búsqueda de la ubicación óptima (Best Global). Finalmente ejecutado el código fuente en Python se obtiene la mejor configuración de generadores tanto eólicos como fotovoltaicos en el sistema y en sus respectivas barras, a partir de la estabilidad analizada.

El valor agregado está dado por la representación del comportamiento de las señales de voltaje, frecuencia y potencia una vez determinado la mejor solución, esto para que el usuario estudie los resultados. La **Figura 3.1** muestra un diagrama de flujo que sintetiza el proceso a realizar por el usuario de la herramienta desarrollada en Python para optimizar la ubicación de generación eólica y fotovoltaica.

La secuencia representa el proceso que debe seguir el usuario de la herramienta, cuando se requiera el análisis de ubicación en cualquier red y sin importar las cantidades de barras, generadores eólicos o fotovoltaicos. A continuación, se describen los puntos identificados en el diagrama de flujo.

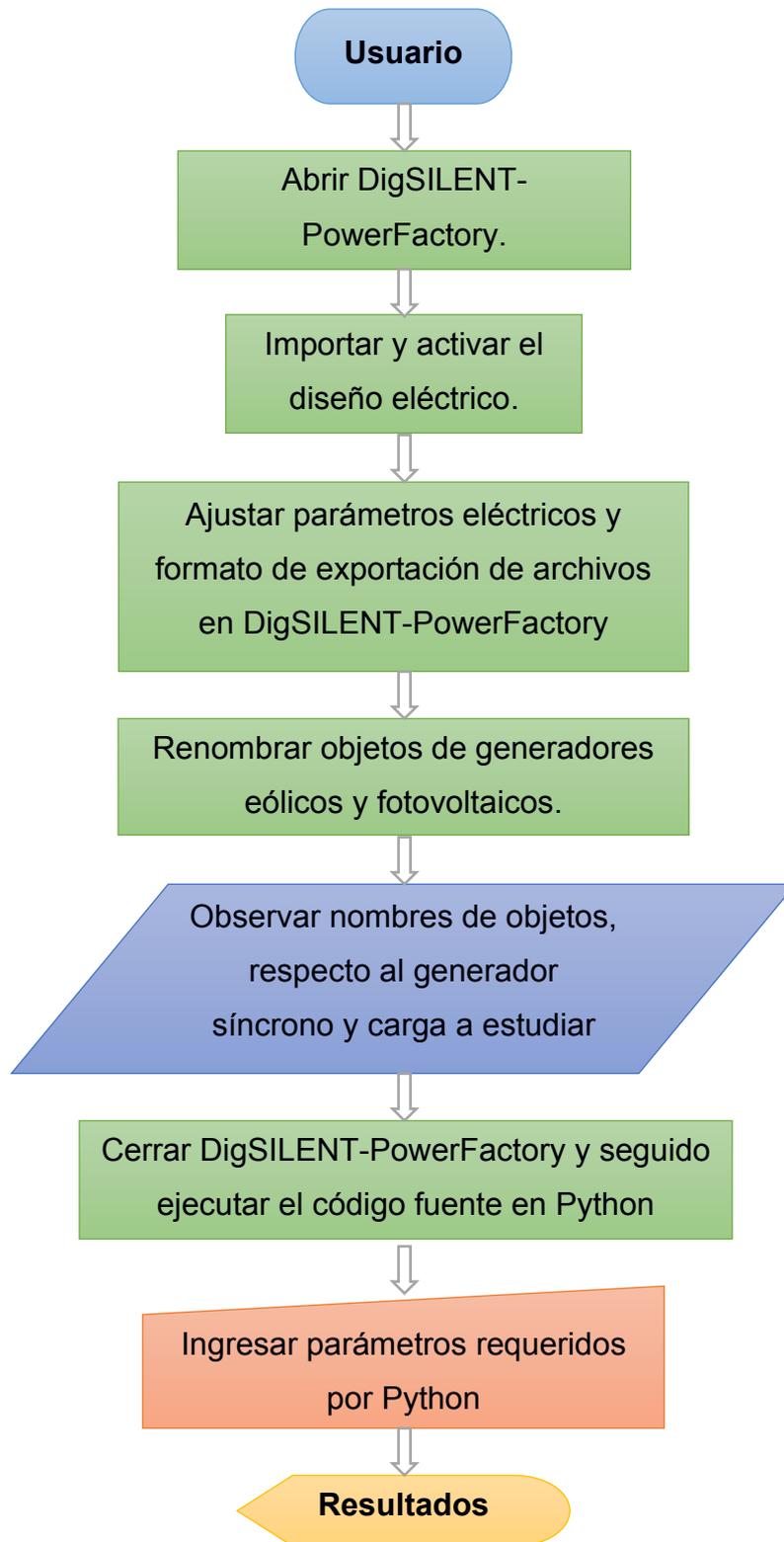


Figura 3.1. Obtención de la mejor locación de generación eólica y fotovoltaica

Usuario: Se define de manera general, al personal encargado de la operación, planificación y diseñador de proyectos de construcción vinculados a una entidad inmersa en el sector eléctrico, cabe mencionar que esto sucede en el supuesto que se use la herramienta (código fuente desarrollado en Python) como un servicio. Referente al proyecto los usuarios identificados son dos estudiantes de la Escuela Superior Politécnica del Litoral (ESPOL), que utilizan la herramienta para determinar la mejor ubicación de la generación fotovoltaica y eólica a partir del estudio de estabilidad de pequeña señal.

Abrir el software DigSILENT-PowerFactory: es la acción de ingresar mediante el icono de escritorio que se muestra en la **Figura 3.2** al ambiente de una herramienta especializada en estudios eléctricos, se puede ejecutar una vez dentro estudios de flujo de potencia, análisis de contingencia, estudios de corto circuito, en general estudios de sistemas estáticos y dinámicos.

Importar y activar el diseño eléctrico: La opción de importar se da cuando se ha desarrollado el diseño eléctrico en otro computador, es decir se tiene el archivo con extensión “.pfd”, correspondiente a DigSILENT-PowerFactory. La visualización del esquema de sistema de potencia a estudiar se da después de haber importado el archivo y activado el proyecto, con la opción “Activate Project”, que se encuentra en el menú desplegable de la pestaña “File”.

Ajustar parámetros de estándares eléctricos y formato de exportación de archivos en DigSILENT-PowerFactory: La frecuencia eléctrica natural del sistema, dependiente de la locación a nivel mundial del diseño, por ejemplo, en Europa comúnmente los sistemas funcionan a 50 Hz, caso contrario en América del Norte y América del Sur que operan a frecuencia de 60 Hz.



Figura 3.2. Icono de escritorio de PowerFactory 2021 SP5

El formato de exportación para el cual se desarrolló el código fuente es “.txt” o extensión de archivos de texto, recomendable si el usuario desea obtener y tratar con datos correspondiente a señales de frecuencia, voltaje, potencia y otros más de manera externa.

Renombrar objetos de generadores eólicos y fotovoltaicos: En el desarrollo del código fuente se emplea para lograr un accionamiento de generadores eólicos y fotovoltaicos la lectura del atributo del objeto definido para sistema fotovoltaico “PV_Sys#” y para el generador eólico “StaGen#”, de tal manera que el # representa el número de elementos de generación fotovoltaica en el sistema, es decir se itera desde 1 hasta el valor n que está definido en base al tamaño de filas en el directorio de generadores fotovoltaicos.

Observar nombres de objetos, respecto al generador síncrono y carga a estudiar: El análisis de pequeña señal requiere que se simule un evento de falla de desconexión y conexión al sistema. El programa desarrollado necesita leer el atributo (nombre del elemento de tipo carga), por ello se debe especificar el nombre de la carga. El caso del generador síncrono es similar, solo que este representa el elemento que ve los efectos de la estabilidad en el sistema, debido a la pérdida del sincronismo generada por la perturbación en la carga.

Cerrar DigSILENT-PowerFactory y seguido ejecutar el código fuente en Python: Previo al cierre del programa se debe verificar que los comandos de edición de variables (Edit Result Variables) y el comando de edición de eventos de simulación (Edit Simulation Events) vacíos, esto para evitar la acumulación de eventos. La automatización del proceso de estudio de pequeña señal realizados en las barras del sistema se da a partir de la ejecución “Run” en el editor de Python, cuyo icono de escritorio se muestra en la **Figura 3.3**, este procede a establecer la conexión con la interfaz de DigSILENT-PowerFactory mediante el módulo “powerfactory”.

Ingresar parámetros requeridos por Python: El desarrollo de código fuente establece parámetros que necesariamente son datos de entrada que depende del sistema a estudiar y que pueden ser ingresados segundos después de haber compilado el código fuente. Los parámetros corresponden al número de partículas, nombre de la carga en donde se simula el evento de falla.

Resultados: Representa el valor óptimo (Best Global) o arreglo que satisface la condición de mejor situación en donde los generadores se ubican en el sistema, esto considerando el índice de estabilidad óptimo. Una vez identificado la mejor ubicación se presentan información de las señales de frecuencia, voltaje y potencia generadas para dicha situación.

3.3 Estudio de estabilidad de pequeña señal para un sistema de 6 barras

DigSILENT-PowerFactory tiene una amplia gama de funcionalidades para el análisis de sistemas de energía. El software permite obtener diferentes módulos de análisis como flujos de carga y eventos de fallas programables con una librería muy amplia de equipos de potencia para adecuarse a las necesidades del usuario. Su aplicación dentro del estudio de estabilidad comprende desde el modelamiento de componentes del sistema hasta análisis de datos mediante su interfaz gráfica para la visualización de diferentes variables de estudio. Para ejemplificar las funcionalidades del software DigSILENT-PowerFactory trabajaremos con un sistema de 6 barras y se explicará el proceso respectivo para el análisis de estabilidad de pequeña señal.



Figura 3.3. Icono de escritorio de PyCharm Community

3.3.1 Importación y activación del proyecto

Previamente se realizó el diseño de un sistema de 6 barras visto en la **Figura 3.4** que consta de dos generadores síncronos de 210 MVA y 255 MVA respectivamente, un transformador reductor de relación de transformación de 220/60 kV y dos cargas de 50 MW. Para el modelamiento de los componentes del sistema se usaron datos de la librería de DigSILENT-PowerFactory y se configuró a una frecuencia nominal de operación de 60 Hz como corresponde al mercado eléctrico ecuatoriano.

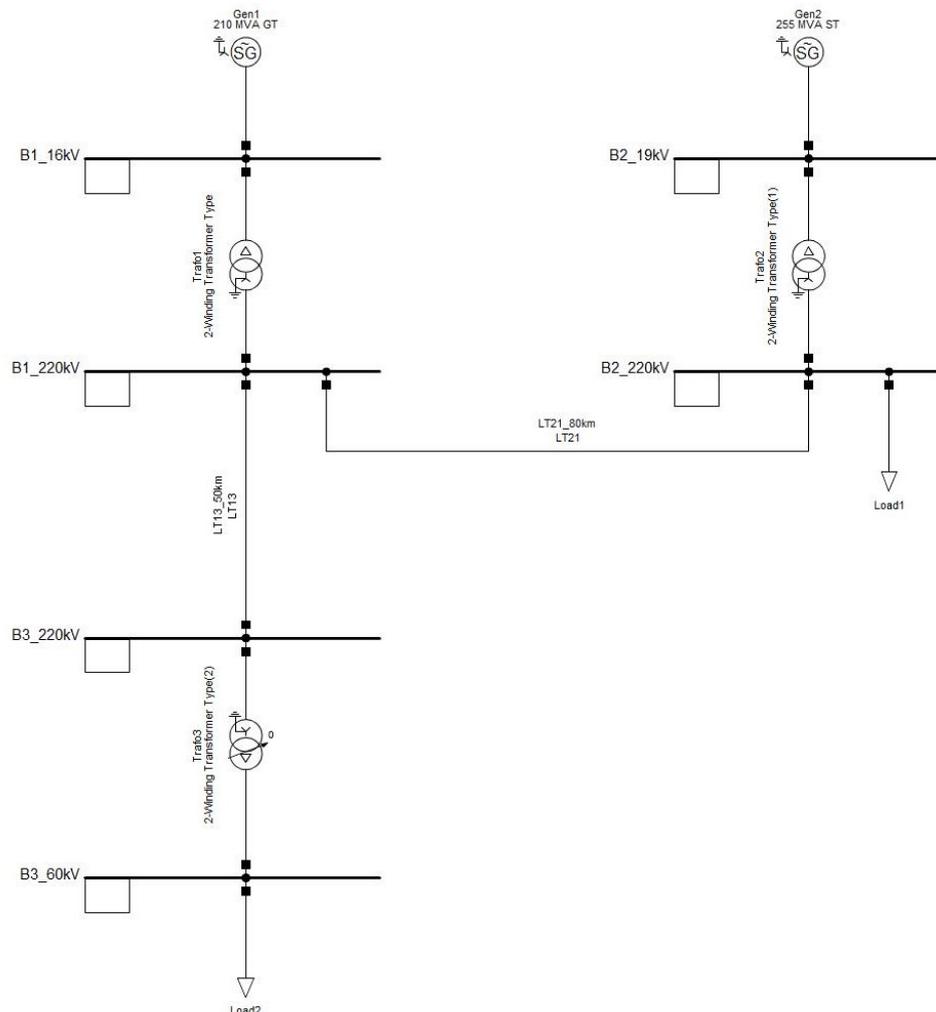


Figura 3.4. Diagrama unifilar para el sistema de 6 barras

Se debe de importar el sistema al entorno de DigSILENT-PowerFactory para poder realizar el estudio de estabilidad de pequeña señal. El archivo con extensión “.pfd” debe de encontrarse dentro de las carpetas de nuestro ordenador. Nos ubicamos en *File* y luego bajamos hasta la opción de *Import* para poder seleccionar *Data(*.pfd;*.dz;*.dle)* y buscar nuestro proyecto que contenga el sistema de 6 barras como se muestra en la **Figura 3.5**.

Una vez que al archivo se encuentra en el entorno de DigSILENT-PowerFactory se debe realizar la activación para poder visualizarlo y manipularlo. Para esto nos situamos nuevamente en *File*, luego vamos a la opción de *Activate Project* y seleccionamos nuestro proyecto. En la **Figura 3.6** se muestra la activación del proyecto *SB6_SinPV_SinEolico* que contiene al sistema de 6 barras.

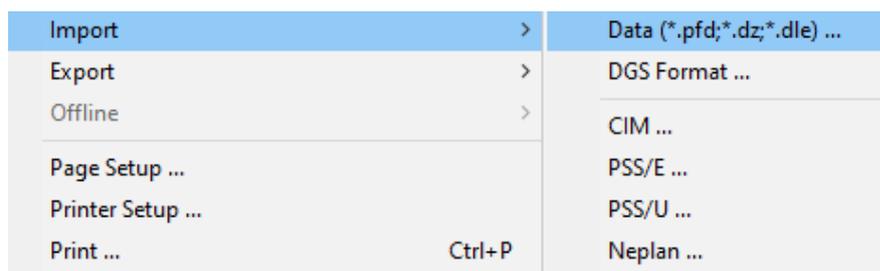


Figura 3.5. Importar el sistema de 6 barras al entorno de PowerFactory

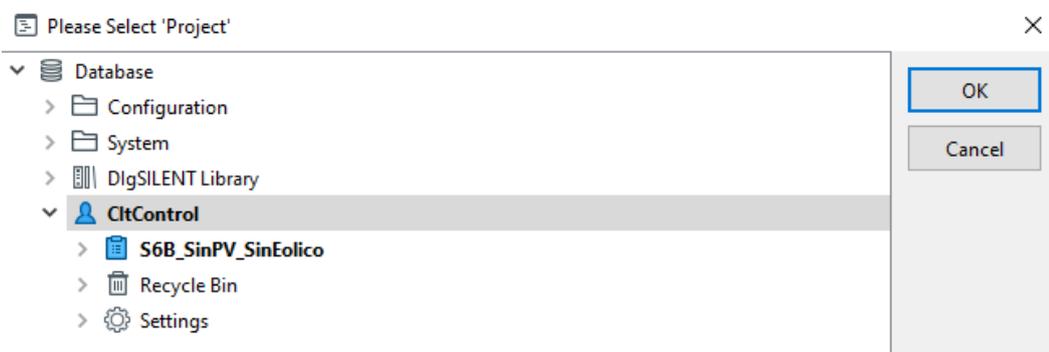


Figura 3.6. Activación del proyecto que contiene el sistema de 6 barras

Una vez activado el proyecto ya podremos visualizar nuestro sistema en DigSILENT-PowerFactory y así poder continuar con nuestro estudio de estabilidad de pequeña señal. La importancia de conocer el manejo de la herramienta especializada en estudios de eléctricos radica en la correcta evaluación de parámetros y sección de opción dependientes de las condiciones del sistema planteado.

3.3.2 Definición de eventos

DigSILENT-PowerFactory permite al usuario definir una serie de eventos como por ejemplo la apertura de interruptores, eventos de fallas, desconexión de generadores, etc., que pueden ser configurados para tener lugar en cualquier instante de tiempo durante el lapso simulado. Los eventos pueden definirse para cualquier componente del sistema eléctrico, incluyendo líneas, transformadores, cargas y barras, permitiendo al usuario obtener un análisis gráfico de los resultados del proceso.

Para nuestro sistema de 6 barras se definirán dos eventos: uno para la apertura y otro para el cierre del interruptor de la carga *Load1*. Esto permitirá simular una falla del tipo conexión y desconexión de carga para poder analizar el comportamiento del sistema antes, durante y después de la perturbación con el fin de realizar un estudio de estabilidad de pequeña señal.

Debemos de ubicar el cursor sobre el elemento sobre el cual queremos realizar la definición de un evento, que para nuestro caso de estudio será la carga *Load1*, dar clic derecho, situar el cursor en *Define* y seleccionar la opción de *Switch Event*.

Dentro de la ventana de *Switch Event* vista en la **Figura 3.7** se configura el evento de interruptor para que suceda en cualquier instante. Se recomienda que el evento se sitúe dentro del periodo de simulación de modo que se pueda analizar el comportamiento gráfico de las variables del sistema, del cual se explicará más adelante.

En la **Figura 3.8** se observa el tipo de configuración que se le puede dar al evento de switch. Definiremos un evento de apertura “Open” en 0.25 segundos y otro evento de cierre “Close” en 0.50 segundos en el interruptor de la carga *Load1* para luego presentarse en la **Figura 3.9** los eventos creados.

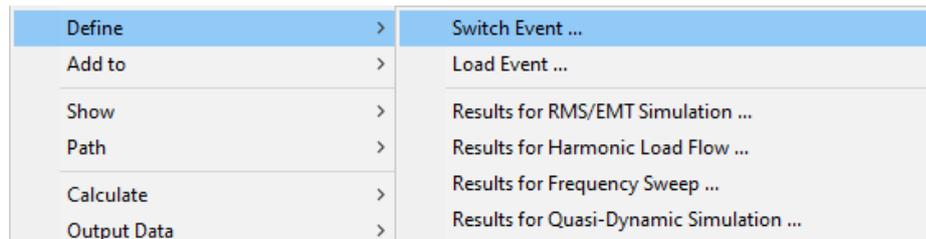


Figura 3.7. Definir un evento de switch en PowerFactory

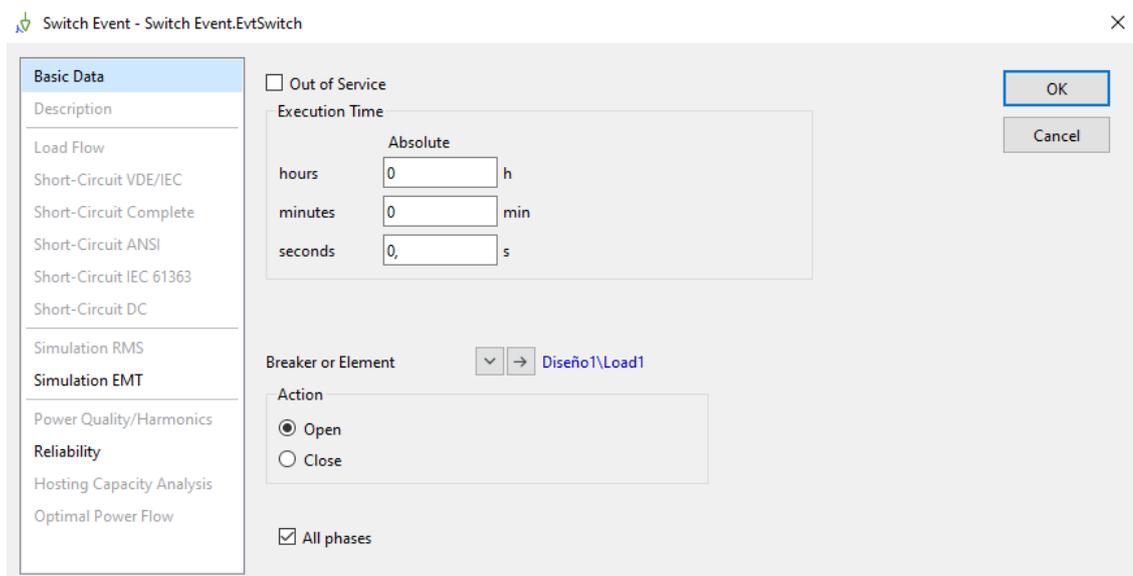


Figura 3.8. Ventana de configuración para el evento de switch

The image shows the 'Simulation Events/Fault' window with a table of simulation events. The table has the following columns: Name, Time, Object, Out of Service, and Object modified.

Name	Time	Object	Out of Service	Object modified
Desconexion de Carga	0,25	Load1	<input type="checkbox"/>	17/12/2021 2:42:42
Conexion de Carga	0,5	Load1	<input type="checkbox"/>	17/12/2021 2:43:02

Figura 3.9. Ventana de edición de eventos simulados

Para poder estar seguros de que nuestros eventos fueron creados podemos abrir la ventana de *Edit Simulations Events* ubicada en el listado superior de *RMS/EMT Simulations* y verificar los dos eventos programados. Los eventos que se pueden crear pueden ser de varios tipos como de interrupción, corto circuito y con la variante de la ubicación de la perturbación, es decir existen muchos escenarios.

3.3.3 Definición de variables para estudio de estabilidad

Las opciones gráficas de DigSILENT-PowerFactory permiten al usuario realizar un análisis gráfico de cualquier objeto creado y almacenado en la base de datos del software. Como nuestra finalidad es analizar la estabilidad del sistema, definiremos las variables de Frecuencia y Potencia Activa del generador síncrono *Gen1* que trabaja a una potencia aparente nominal de 210 MVA.

Ubicaremos el cursor sobre el generador síncrono *Gen1* y daremos clic derecho. Se desplegarán el listado de opciones que se muestra en la **Figura 3.10** en donde nos situaremos sobre *Define* y seleccionaremos la opción de *Results for RMS/EMT Simulations*.

Nos aparecerá la ventana de resultados correspondiente a la **Figura 3.11** en que se mostrará el objeto sobre el cual queremos definir las variables. Daremos clic sobre *Gen1* y continuaremos con la selección de variables.

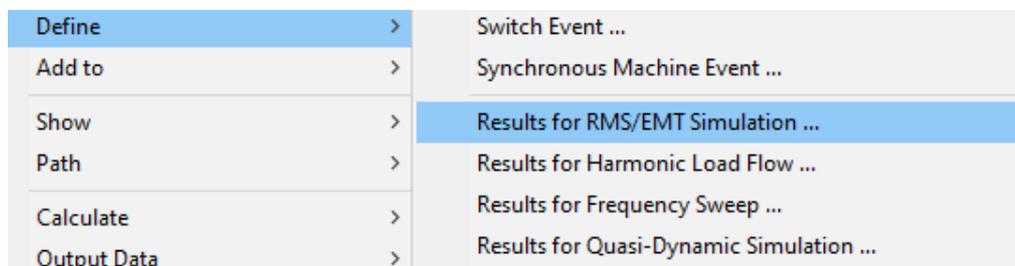


Figura 3.10. Definición de variables

El listado de variables disponibles para el generador seleccionado es muy amplio. Para estudios de estabilidad nos enfocaremos en tres variables, *fehz:bus1* que corresponde a la frecuencia eléctrica en Hz, *Psum:bus1* que corresponde a la potencia activa en MW y el *ut* para el voltaje terminal en p.u. Las añadimos al listado de variables seleccionadas dando clic sobre el cuadro de la variable correspondiente y cerramos la ventana pulsando *OK*.

3.3.4 Cálculo de flujo de carga y condiciones iniciales

El análisis de flujo de carga nos permite conocer los datos nodales necesarios para determinar los flujos en las líneas de transmisión y demás elementos del sistema, considerando los niveles de demanda y generación. Mediante estos datos se pueden realizar numerosos cálculos para determinar el funcionamiento del sistema como cálculos de fallas, cálculos de contingencia, cálculos de aumento de demanda y/o generación, etc. Entonces, el análisis de flujo de carga se vuelve fundamental para obtener un estudio predictivo y de esa manera evitar fallos que afecten a la seguridad de cualquier sistema eléctrico. DigSILENT-PowerFactory nos brinda la herramienta para realizar el cálculo de flujo de carga para nuestro sistema.

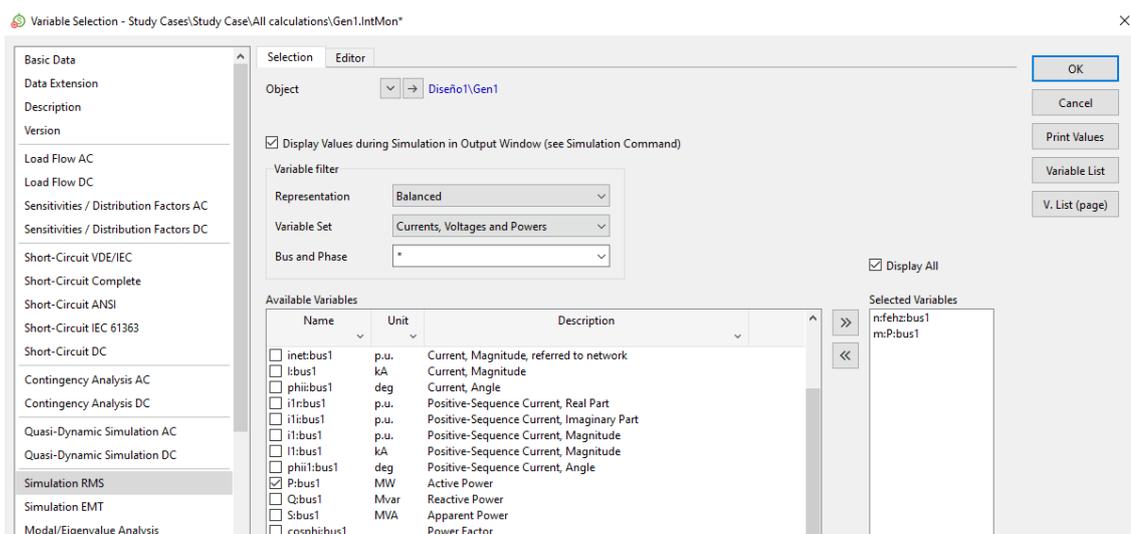


Figura 3.11. Variables disponibles en el generador síncrono Gen1

El cálculo de flujo de carga emplea el método de Newton-Raphson, analizando las condiciones del sistema en estado estacionario y sin ningún evento de falla simulado. Para esto debemos dirigir el cursor a la opción *Calculation*, ubicada en la barra de menú principal, y seleccionar *Load Flow*. En la ventana de configuración presentada en la **Figura 3.12** seleccionaremos el método de cálculo balanceado, considerando la secuencia positiva del sistema y ejecutamos. Es importante tener en cuenta que primero se debe de realizar un análisis de flujo de carga para que se pueda habilitar la función del cálculo de las condiciones iniciales del sistema.

Cuando se ejecute el cálculo de flujo de carga nos mostrarán los resultados respectivos en cada elemento de nuestro sistema como se observa en la **Figura 3.13** y **Figura 3.14**. Para las barras se mostrarán los datos de voltaje (*kV*, *p.u.* y ángulo de desfase *phi*) y para los demás elementos del sistema se mostrarán los datos de flujo de potencia activa, potencia reactiva y corriente.

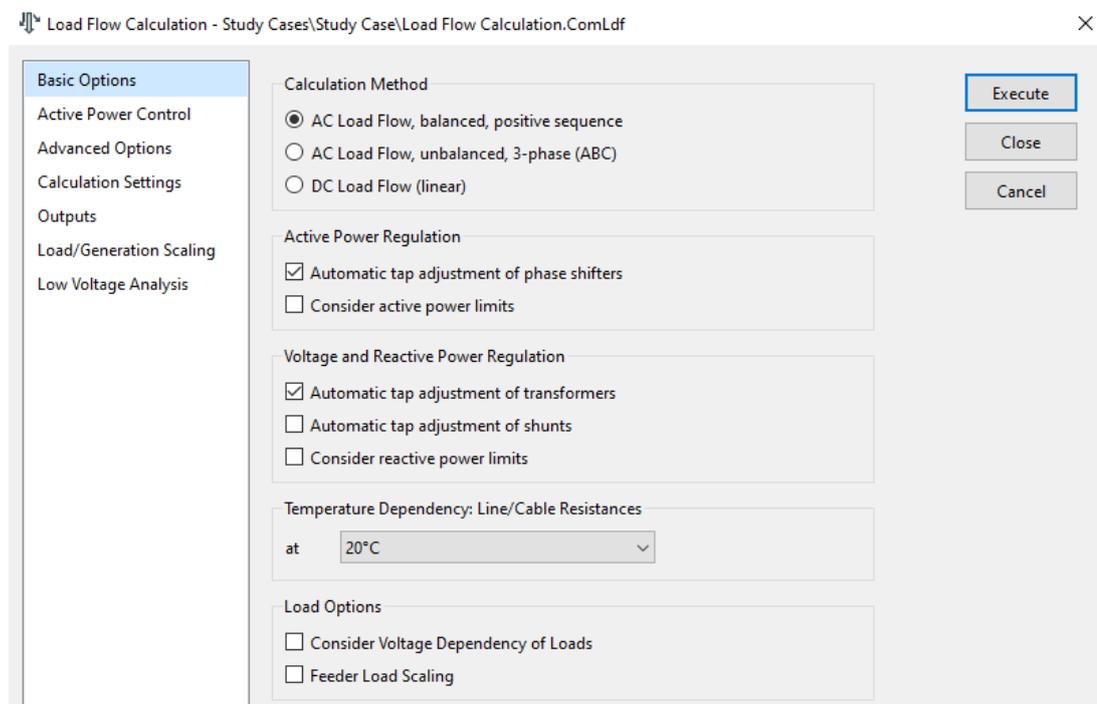


Figura 3.12. Configuración para el caso de estudio de flujo de carga

Ahora se procede a realizar el cálculo de las condiciones iniciales que corresponde a la configuración de la ventana mostrada en la **Figura 3.15**. Es de suma importancia ya que contribuye al conocimiento del estado interno de los elementos del sistema. El mismo se realizará mediante el método RMS ya que estamos trabajando con una red eléctrica balanceada.

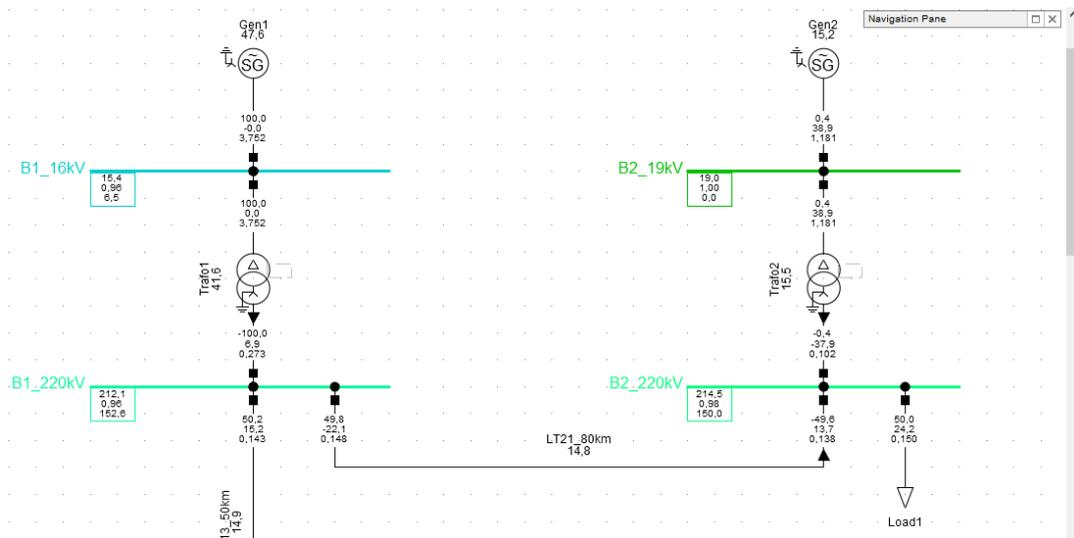


Figura 3.13. Flujo de carga del sistema de 6 barras – primera parte

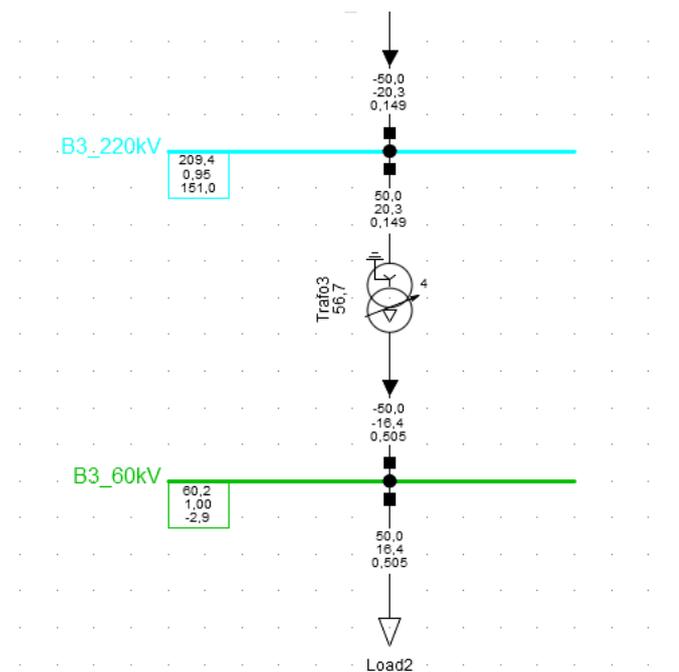


Figura 3.14. Flujo de carga del sistema de 6 barras – segunda parte

Para acceder al cálculo de las condiciones iniciales debemos de dirigirnos a *Calculations*, ubicar el curso en *Simulations RMS/EMT* y seleccionar *Initial Conditions*. Los resultados de las condiciones iniciales nos sirven para conocer el estado en el que se encontraba nuestro sistema previo a cualquier evento de falla. Ahora se puede proceder con la simulación de eventos de falla programados para el sistema.

3.3.5 Simulación de los eventos programados

Previamente se realizó la definición de eventos que corresponden a la desconexión y conexión de la carga *Load1*. Los eventos se activan cuando se realiza la simulación del sistema en un tiempo establecido, donde se simulan todos los eventos en los tiempos programados y se obtiene el estado de los componentes del sistema luego de la perturbación. Para acceder a esta función debemos de dirigirnos a *Calculations*, luego a *Simulations RMS/EMT* y seleccionar *Start Simulation*. Nos aparecerá la ventana que se presenta en la **Figura 3.16** en donde se permitirá establecer el tiempo de simulación que requiera el usuario, que para caso de estudio de este trabajo es de 20 segundos.

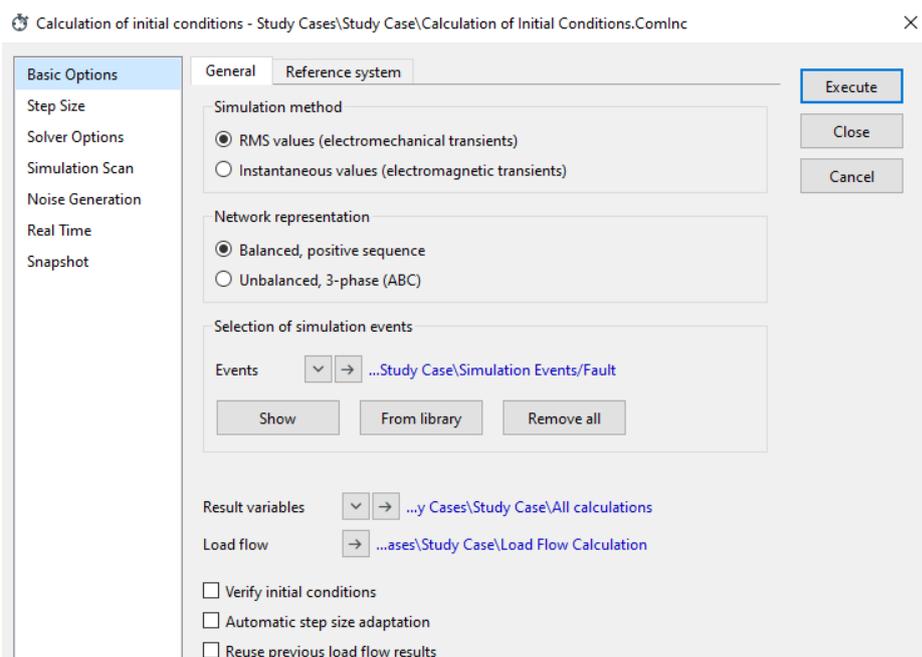


Figura 3.15. Ajuste de condiciones iniciales para el sistema de 6 barras

Se recuerda que el tiempo de simulación debe establecerse según los tiempos de los eventos programados para que corran en el lapso definido. Cuando los eventos programados tuvieron lugar en nuestra simulación, los cambios vienen señalados en nuestro sistema. La **Figura 3.17** se muestra la visualización cuando la perturbación actúa sobre la carga Load, siendo esta un evento de desconexión y conexión de carga *Load1*.

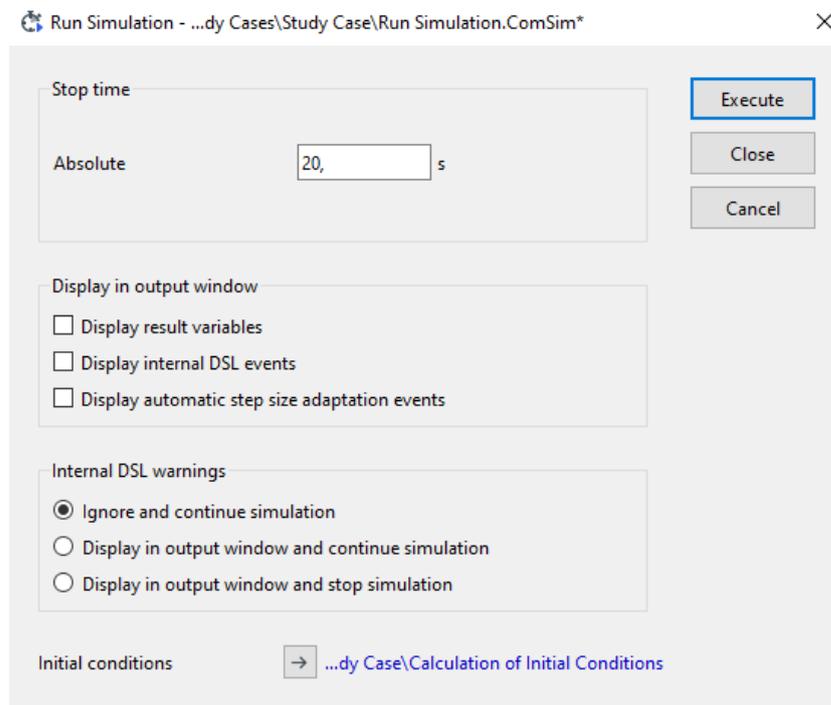


Figura 3.16. Configuración del tiempo de simulación del sistema de 6 barras

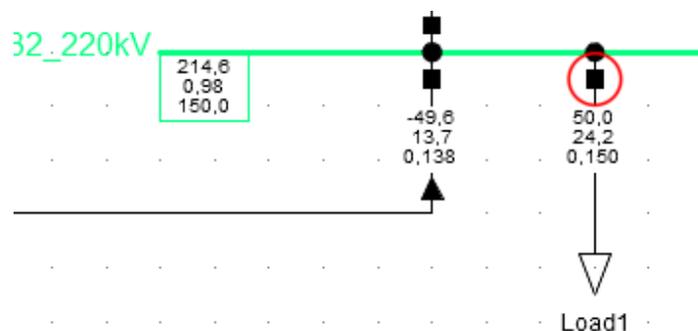


Figura 3.17. Simulación de apertura y cierre de switch para la carga *Load1*

Observamos que el evento del interruptor correspondiente a la carga *Load1* tuvo lugar en nuestra simulación y actualmente se encuentra conectado a la red, ya que la desconexión y conexión ocurre en un intervalo corto de tiempo de 0.25 s.

3.3.6 Graficación y exportación de resultados

Para el estudio de estabilidad de pequeña señal analizaremos el comportamiento de la frecuencia y la potencia activa del generador antes, durante y después de la perturbación. DigSILENT-PowerFactory nos permite obtener un análisis gráfico de nuestras variables definidas previamente. Los límites de las gráficas estarán determinados por la configuración del tiempo de simulación y se podrá observar el comportamiento de la señal siempre que los eventos programados hayan tenido lugar dentro de los 20 segundos simulados.

Para acceder a las funciones gráficas de DigSILENT-PowerFactory debemos de crear una nueva página de trazado en nuestro entorno de trabajo. Se selecciona *Plot Page* y se ejecuta, este proceso se muestra en la **Figura 3.18**. Se presentan diferentes opciones de gráficas, que son vistas en la **Figura 3.19** para necesidades específicas, como se trabajó con simulaciones RMS para analizar el comportamiento transitorio del generador, se escogerá la opción de *Simulation RMS/EMT* y *Curve plot* para visualizar nuestras variables en el tiempo simulado.

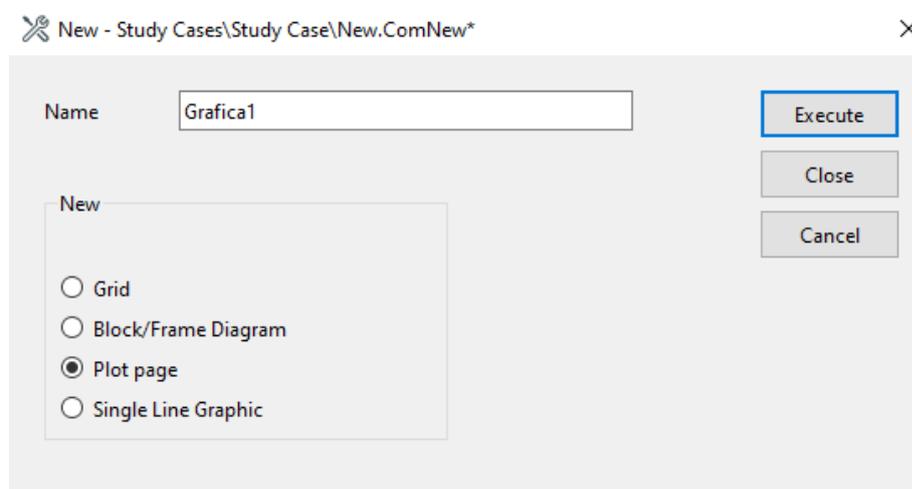


Figura 3.18. Selección de *Plot page*

Agregamos la variable que queremos graficar con doble clic sobre la página de la ventana de gráfica. En la ventana de *Curve plot*, vista en la **Figura 3.20** seleccionaremos el elemento del cual se realizará el estudio, la variable (frecuencia o potencia) y las características de la curva.

El estudio de estabilidad de pequeña señal comprende el análisis del régimen transitorio de la frecuencia eléctrica, potencia activa y hasta incluso el voltaje terminal del generador en cuestión.

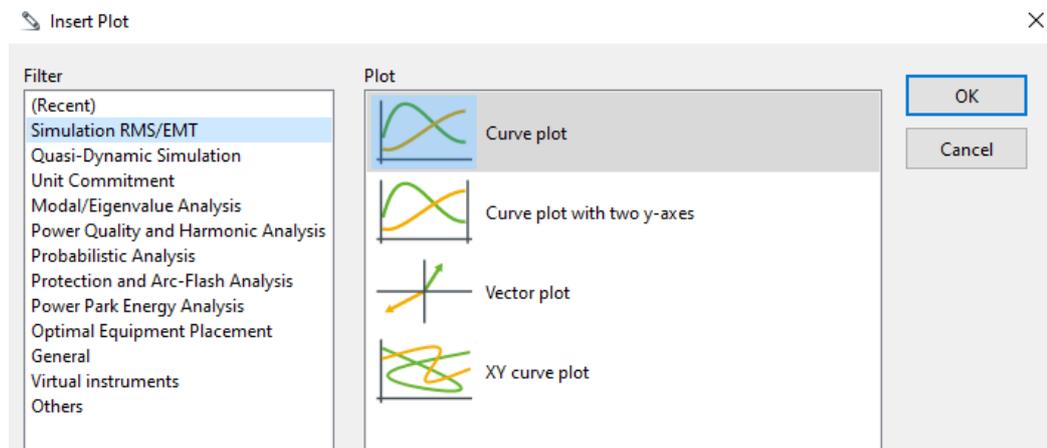


Figura 3.19. Tipo de gráfica seleccionada

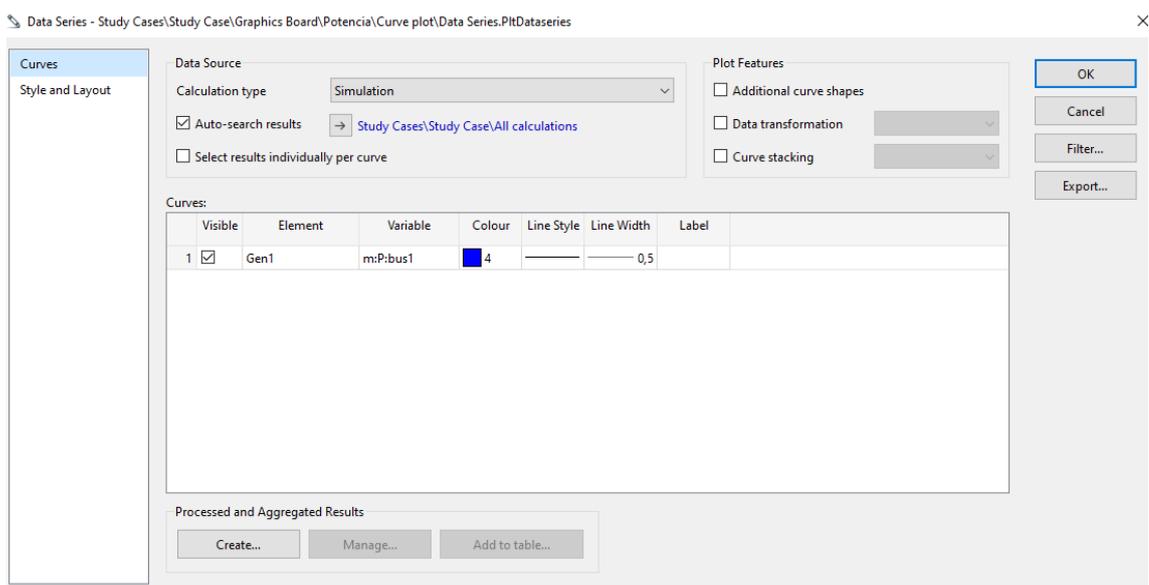


Figura 3.20. Ventana de *Curve plot*

Observamos que el comportamiento de las variables durante la perturbación hasta que se extingue en el tiempo y vuelve a estabilizarse en un nuevo punto óptimo. Los datos de la gráfica que se muestran en la **Figura 3.21** y **Figura 3.22** tiene como opciones de exportación el tipo de archivos con extensión “.txt” o “.csv” de manera que puedan observarse los cambios en frecuencia y potencia con mayor precisión. Se realiza dando clic derecho a la gráfica y seleccionando la opción *Export*.

Una vez en la ventana de *Result Export* vista en la **Figura 3.23**, ubicamos el nombre con el que queremos guardar el archivo, la dirección donde se guardará dentro del ordenador donde se esté trabajando y la extensión. En el caso que no se seleccione un lugar de destino, el archivo se guardará por default en la carpeta donde se encuentra el caso de estudio en “.pfd” y como archivo de texto.

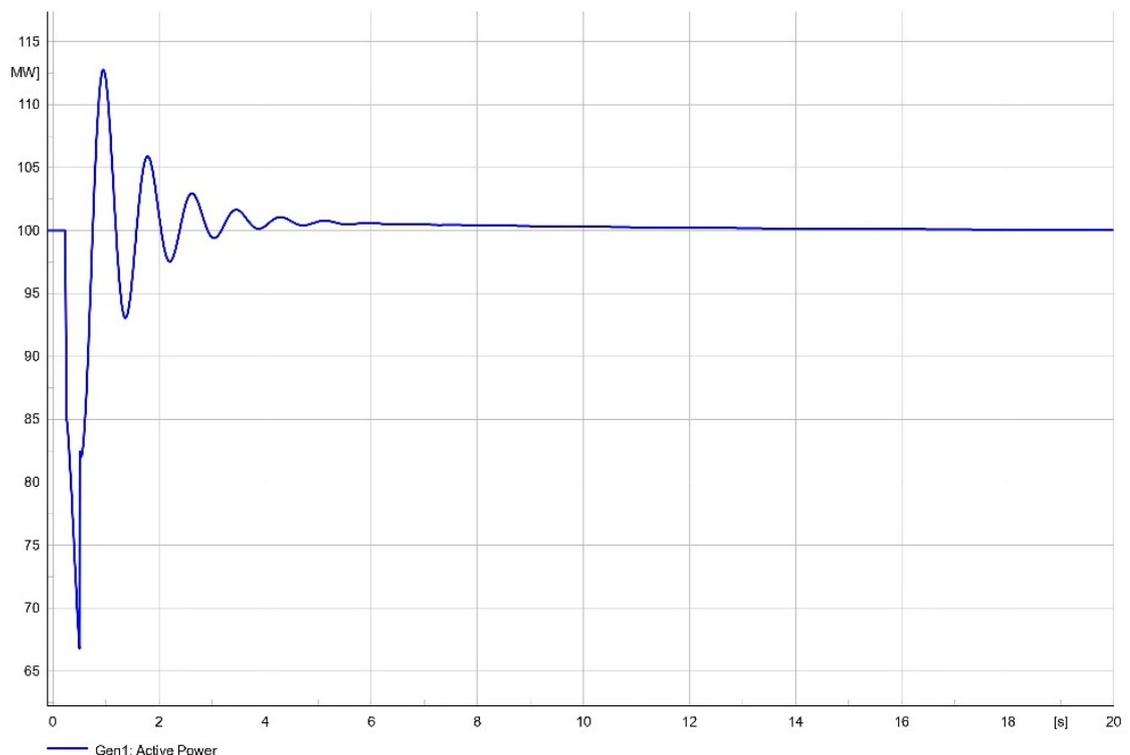


Figura 3.21. Gráfica de potencia activa para el generador Gen1 en MW

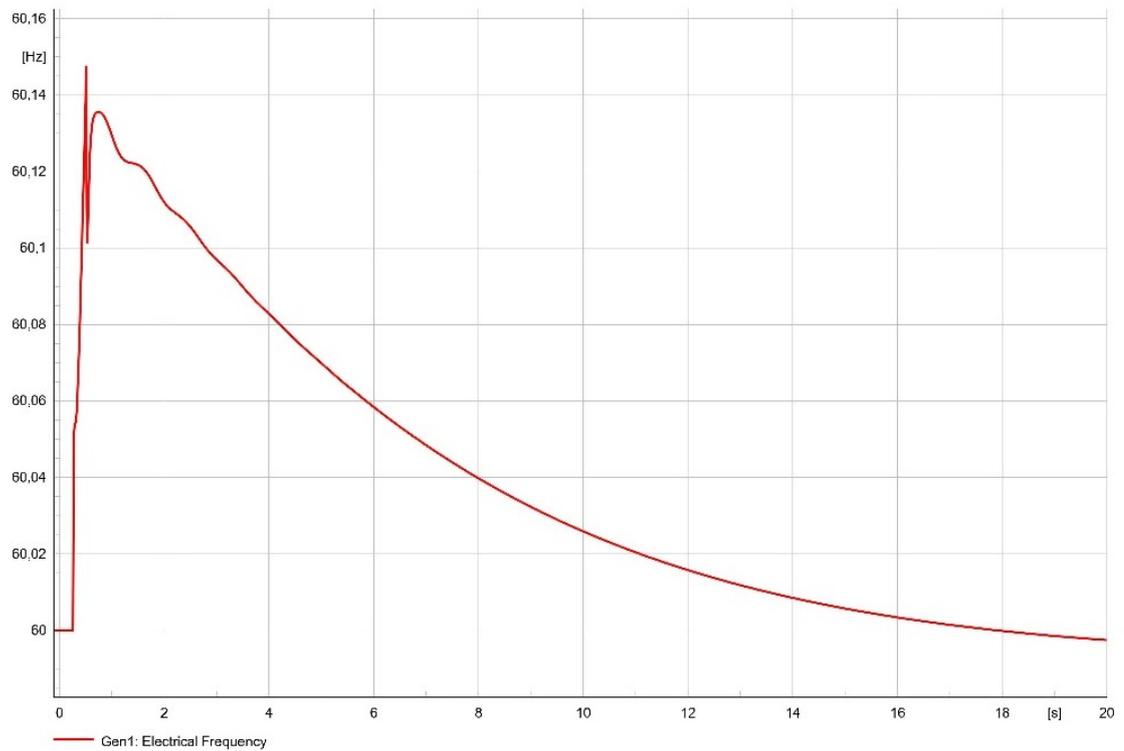


Figura 3.22. Gráfica de Frecuencia Eléctrica del generador *Gen1* en Hz

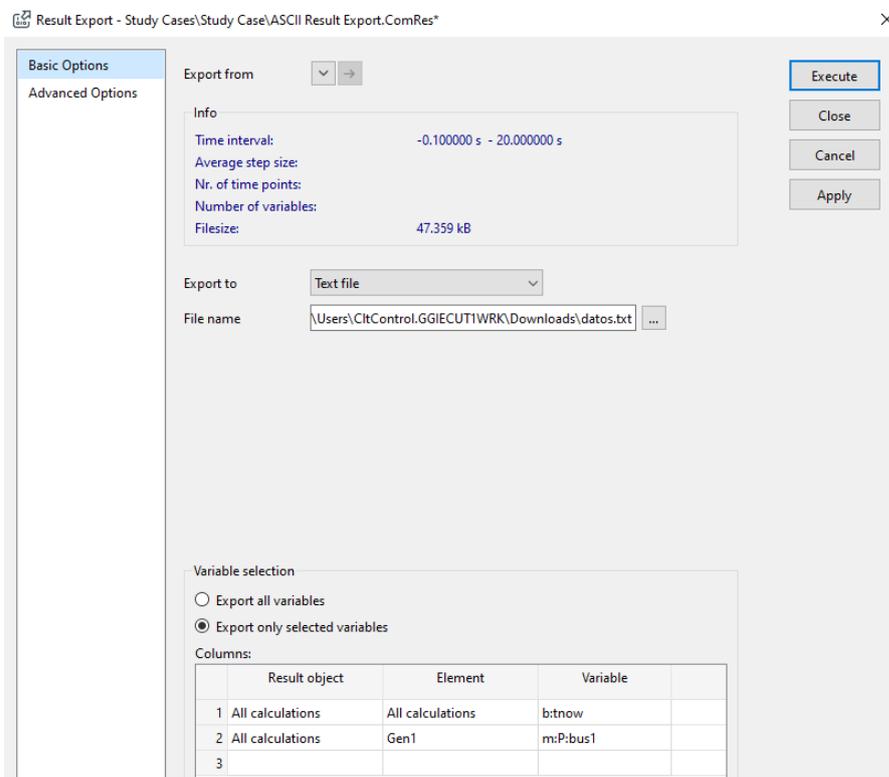


Figura 3.23. Exportación de datos

Los datos pueden exportarse con todas las variables en un mismo archivo. Como requerimos un estudio de la variable en el tiempo, se exportan los valores de tiempo con los valores de la variable asignada en dicho tiempo, formándose dos columnas.

3.4 Sistema de 6 barras con generación eólica y fotovoltaica

Para efectos prácticos, se realizó la conexión de sistemas fotovoltaicos y eólicos en todas las barras del sistema que se muestra en la **Figura 3.24** para poder seleccionar de manera aleatoria los que estarán activados durante la simulación.

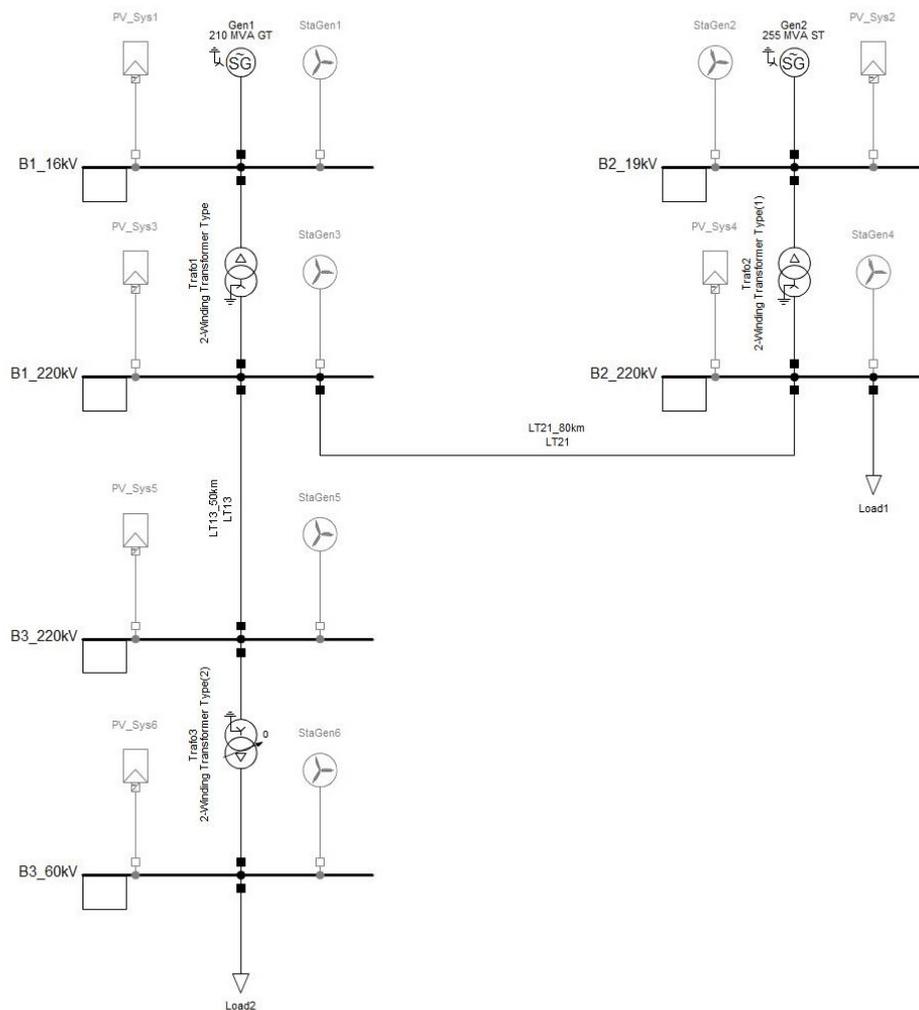


Figura 3.24. Red de 6 barras con integración de generación eólica y fotovoltaica

La perturbación tendrá lugar en la carga *Load1*, en donde se programará la desconexión de carga en 0.25 segundos y la conexión de carga en 0.50 segundos. Se simulará en un tiempo de 20 segundos de tal manera que podamos apreciar la perturbación en las gráficas y el comportamiento de las curvas de frecuencia, potencia activa y voltaje terminal del generador *Gen1*. La **Figura 3.25** muestra el flujo de carga al final de los 20 segundos simulados, se observa una mayor incidencia de generación eólica a comparación de la fotovoltaica.

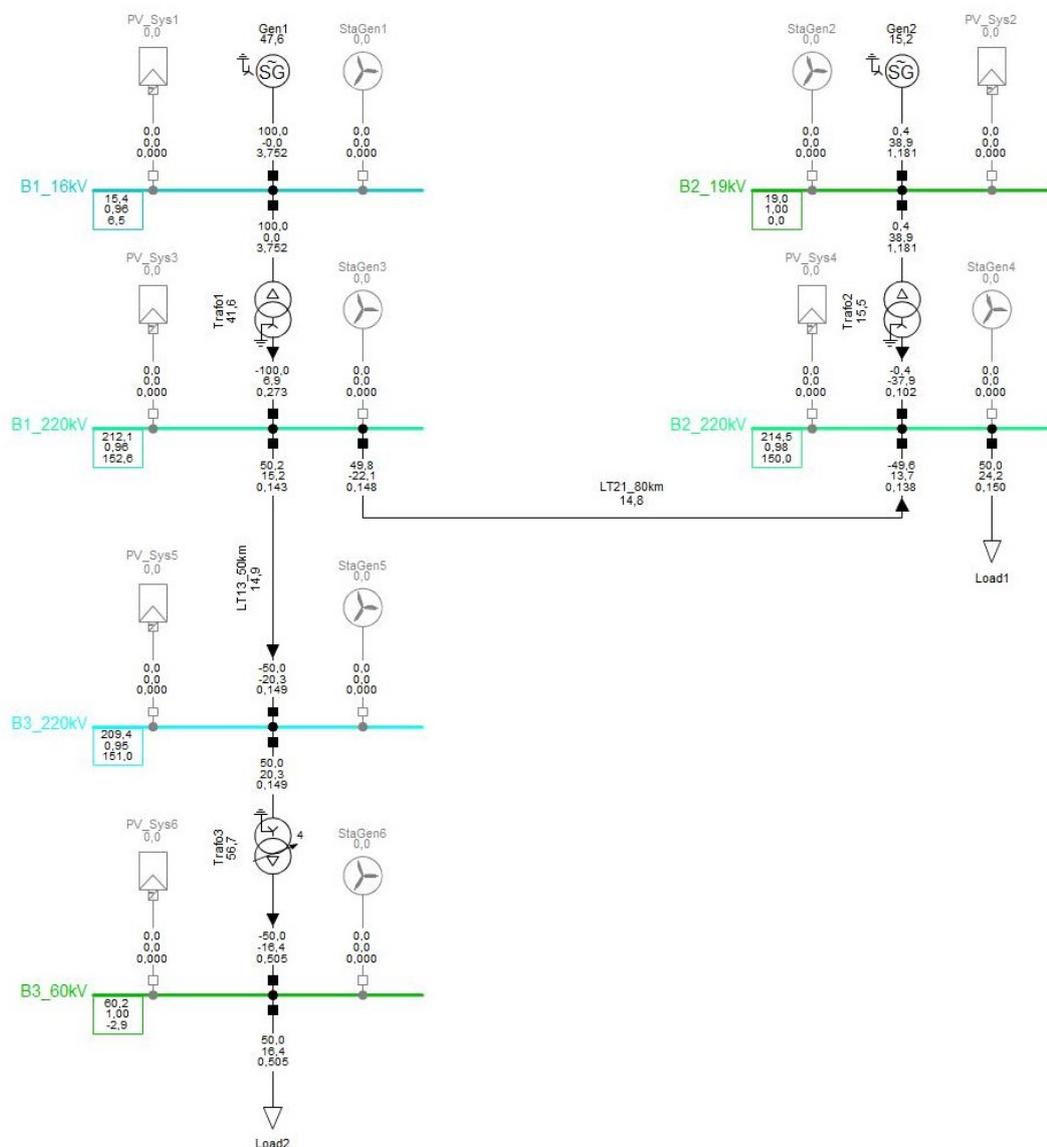


Figura 3.25. Flujo de carga en el sistema luego de la perturbación

De manera aleatoria se eligió la siguiente configuración para la generación eólica y fotovoltaica:

- **Ambas:** barra 2 y barra 6.
- **Únicamente eólica:** barra 3 y barra 5.
- **Únicamente fotovoltaica:** ninguna barra.
- **Ninguna:** barra 1 y barra 4.

Previamente se definieron las variables de frecuencia en *Hz*, potencia activa en *MW* y voltaje terminal en *p.u.* en el generador *Gen1* de manera que se puedan visualizar las curvas en 3 pestañas diferentes.

Cuando se realiza el estudio de estabilidad de un sistema eléctrico, lo que se analiza es la capacidad del sistema de mantener los niveles de frecuencia, potencia y voltaje en estado de equilibrio después del evento transitorio.

Cualquier cambio en la carga provoca que el ángulo del rotor del generador comience a buscar el punto óptimo en donde se suscite la condición de equilibrio. Esto se logra siempre que exista la suficiente fuerza restauradora que permita a la máquina acelerarse o desacelerarse con respecto a la referencia inicial.

En la **Figura 3.26** podemos observar como la frecuencia se dispara en el momento exacto que sucede la desconexión de carga, lo que se traduce en un giro sobre acelerado del rotor. Para poder observar con mayor detenimiento el comportamiento de la frecuencia, se ha realizado la simulación en un tiempo de 200 segundos.

Como resultado inmediato del desequilibrio entre la generación y la carga, la frecuencia del sistema empieza a desviarse de su valor nominal. Se observa en la **Figura 3.27** como la frecuencia disminuye su valor inmediato se reconecta la carga, sin embargo, luego vuelve a elevarse y se comienzan a generar pequeñas ondas que van decreciendo. Comparando con la frecuencia del sistema sin generación eólica ni fotovoltaica, podemos ver como la frecuencia disminuye sin ningún tipo de

amortiguamiento y alcanza valores mucho menores de frecuencia aun cuando existe mayor incidencia de generación eólica que fotovoltaica, lo cual se considera como un mejor escenario para la estabilidad.

Observamos con mayor claridad el comportamiento de la frecuencia mucho después de la perturbación debido a que el sistema no posee ningún tipo de control y protección que permita y facilite llegar a valores estables de operación, se observa que toma mucho tiempo llegar al valor nominal. Incluso se puede observar que entre los segundos 20 y 120 existen valores de frecuencia por debajo del valor nominal estandarizado de 60 Hz a causa de la insuficiente reserva de generación (activa y reactiva) en el sistema.

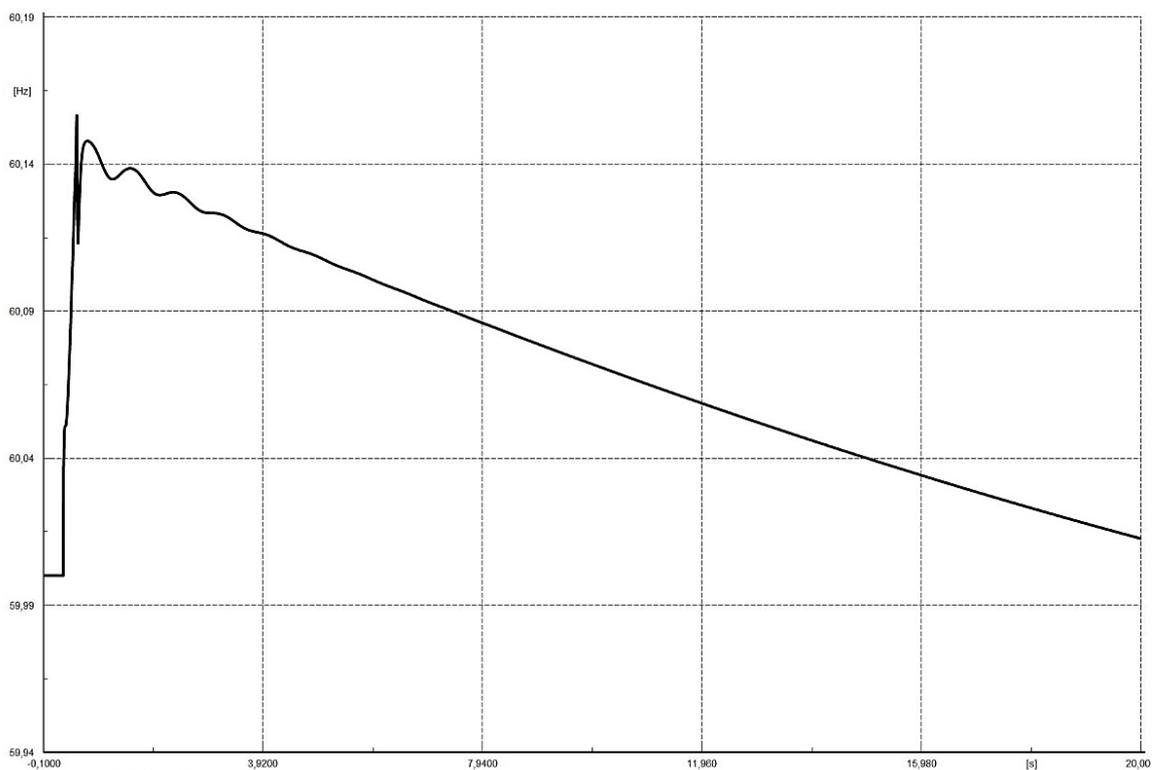


Figura 3.26. Señal de frecuencia eléctrica en [Hz]

La potencia activa tiene una relación proporcional con el par eléctrico del generador. Esto deriva a que, cuando en el sistema disminuya la generación también se disminuirá la velocidad angular del rotor. En la **Figura 3.28** se observa, en el momento que sucede la desconexión de carga, que el generador Gen1 suministra menos potencia activa debido a que no existe la demanda inicial del sistema, pero cuando se reconecta la carga, la potencia activa se dispara a valores mayores al nominal y comienza el periodo transitorio de oscilación. La oscilación de la potencia activa se da durante el periodo en que el generador logre equilibrar el torque electromagnético y mecánico en el sistema. Para suplir este aumento de carga, se transfiere la carga del generador más lento hacia la máquina más rápida, lo que genera una desaceleración de este.

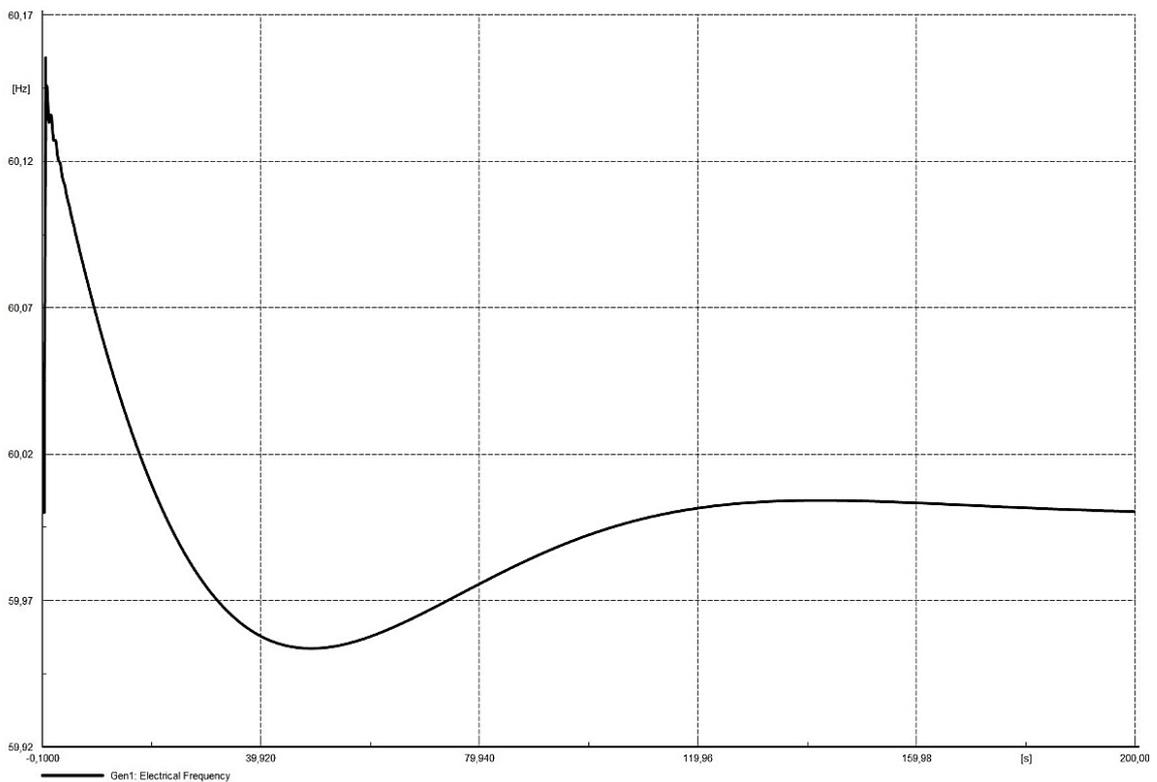


Figura 3.27. Frecuencia eléctrica en [Hz] con $t_{sim} = 200$ [s]

La **Figura 3.29** muestra un comportamiento similar del voltaje terminal con respecto a las otras señales. Al desconectarse la carga, el voltaje terminal se dispara a valores superiores al 1 p.u. Cuando el generador pierde el sincronismo con el sistema, el rotor comienza a girar con la mayor velocidad necesaria para generar el voltaje con respecto a la frecuencia del sistema que, como vimos en la **Figura 3.26**, también se dispara. En el instante preciso en el que se reconecta la carga del sistema, el voltaje disminuye y comienza el periodo oscilatorio. Observamos como el voltaje va tomando valores menores a medida que la velocidad del rotor del generador disminuye. Lo que está sucediendo es una inyección de potencia reactiva por parte del generador considerando la condición operativa estable del voltaje antes de la perturbación. [28]

Finalmente, en la **Figura 3.30** observamos la señal de voltaje terminal del generador en un periodo de 200 segundos donde se aprecia de mejor manera como se va estabilizando a su valor nominal de operación inicial del sistema.

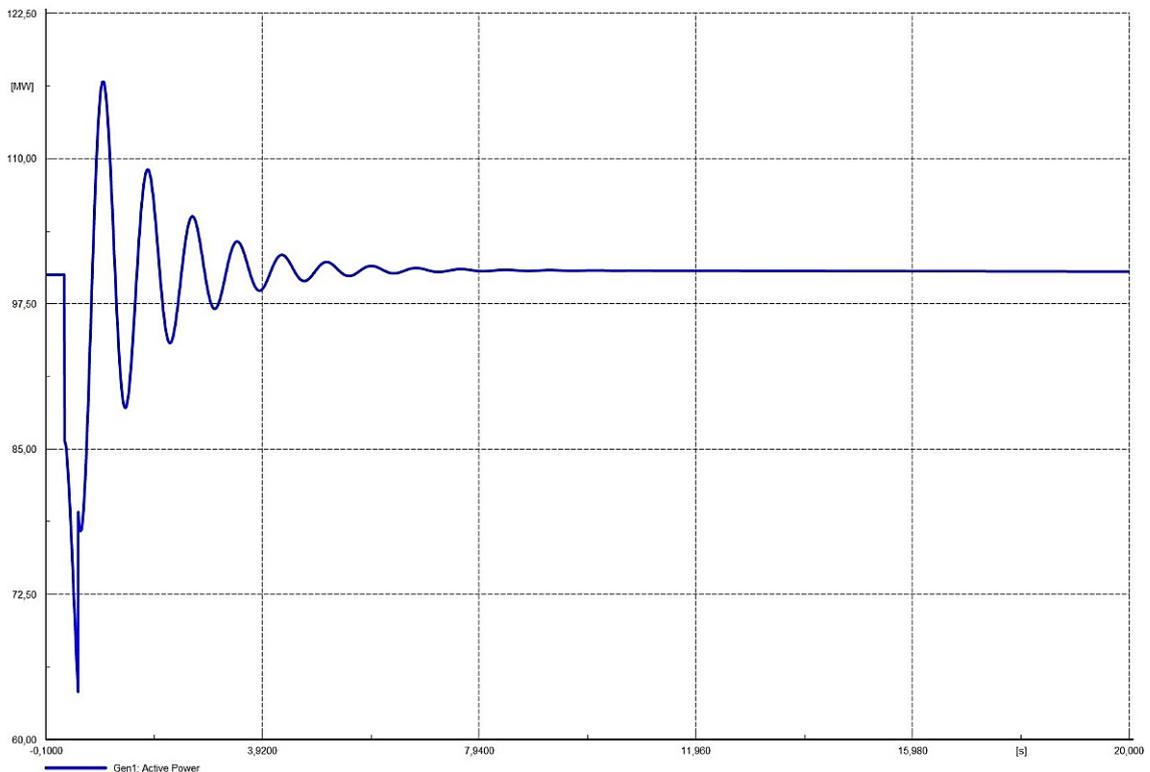


Figura 3.28. Señal de Potencia Activa en [MW]

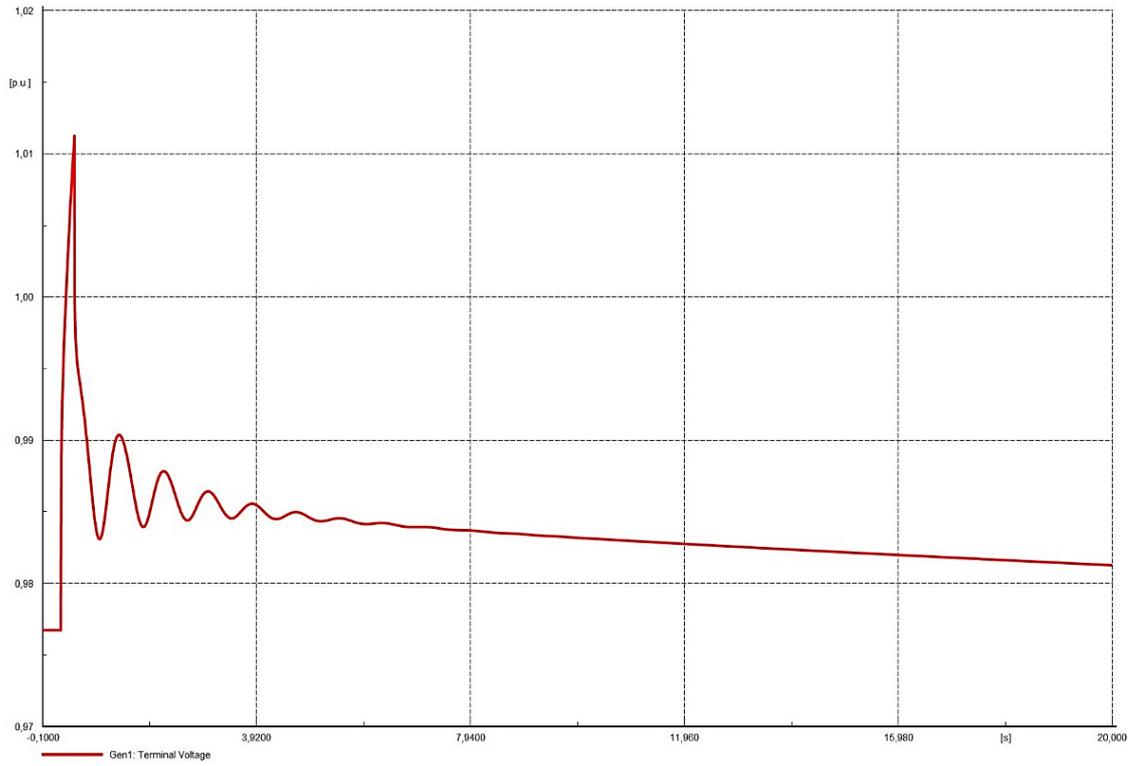


Figura 3.29. Señal de voltaje terminal en [p.u.]

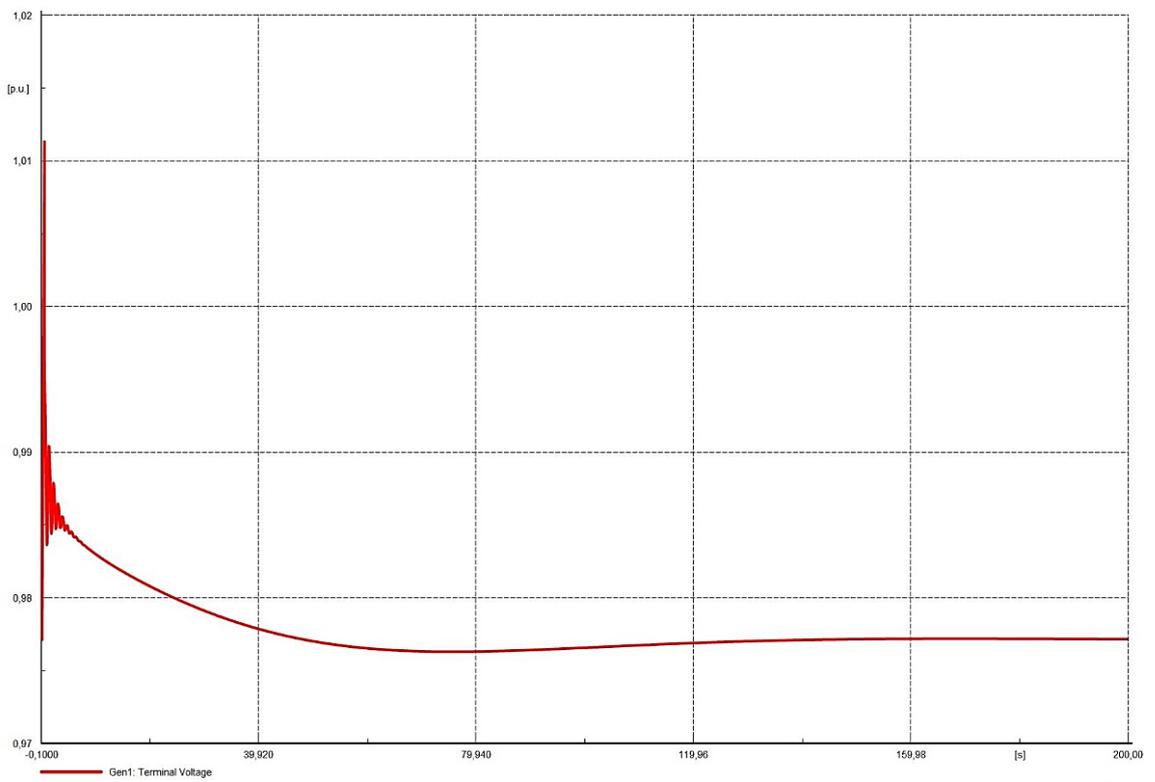


Figura 3.30. Señal de voltaje terminal en [p.u.] con $t_{sim} = 200$ [s]

El análisis de señales proporcionada por la generación síncrona del sistema permite evaluar el estado de la red en presencia de perturbaciones que afectan a la estabilidad de este. La observación como instrumento de investigación mitiga en el comportamiento de los datos de frecuencia eléctrica, voltaje terminal y potencia activa para tomar medidas correctivas con el objetivo de brindar mayor seguridad en el sistema.

3.5 Enlace computacional de PowerFactory con Python

La naturaleza en la existencia de un puente entre el software especializado en estudios eléctricos DigSILENT-PowerFactory y el software de código abierto Python se da partir de la dificultad de obtener datos en procesos repetitivos que se presentan cuando se realizan estudios en el sistema eléctrico o proyecto cargado en DigSILENT-PowerFactory.

Python como facilitador en la ejecución de procesos, este busca automatizar cualquier secuencia repetitiva, por ejemplo, la creación de eventos de fallas como desconexión de carga, corto circuito, interrupción de línea y un sin número de procesos que requieran de tratamiento de muchos datos. Los datos que se obtienen en DigSILENT-PowerFactory están dados en matrices de datos con numero de columna y filas en específico, esos datos pueden ser tratados de manera externa mediante la lectura de archivos, para que luego sean tratados en modelos matemáticos, dependiendo el requerimiento del usuario. El software Python al ser de código abierto y muy amigable para los desarrolladores en la actualidad, es implantado como base en la estructura de programación y tratamiento de datos de otros softwares, ya que ofrece muchas librerías que sintetizan la programación.

Python proporciona al usuario eficiencia y una plataforma amigable para el desarrollo de código fuente, una vez dada la integración con DigSILENT-PowerFactory, este adquiere las características que se mencionan a continuación.

- Brindar un ambiente para desarrolladoras de diferentes especializaciones de programación, considerando el manejo de lenguaje de programación de alto nivel;
- La estructura de programación es apropiada por su claridad y enfocada en la comprensión del desarrollador;
- Software de código abierto;
- Es utilizado como lenguaje de programación en un sin número de campos de estudio;
- La diversidad de desarrolladores que son usuarios de Python proporciona módulos y librerías potentes para la ejecución de procesos, claro ejemplo es numpy, matplotlib, random, time, bases de datos externas, servicios web, aplicaciones análogas a Microsoft Office, etc.

El enlace entre Python y DigSILENT-PowerFactory se establece considerando los pasos que se muestran en la **Figura 3.31**, que corresponden a la instalación de uno de los tantos editores de Python, seguido del módulo proporcionado por PowerFactory y finalmente llamar a PowerFactory mediante el código fuente de comandos de enlace.

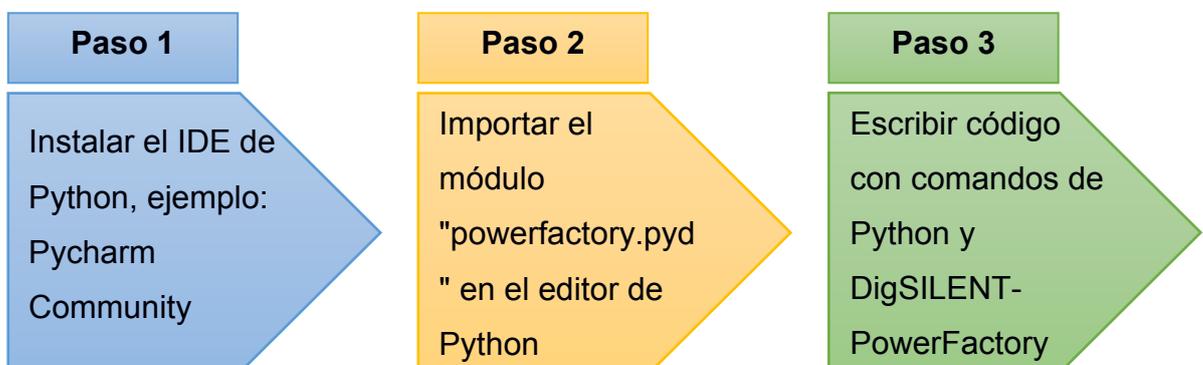


Figura 3.31. Secuencia para el enlace de PowerFactory con Python [5]

Los pasos descritos previamente representan un punto de vista general, pero más allá se debe considerar factores como configuraciones computacionales acerca de permisos de administrador, arquitectura del sistema (Windows 32 bits, Windows 64 bits), sistemas operativos (versiones de: Windows, iOS, Linux, etc.) o versiones de software (DigSILENT-PowerFactory, Python, Pycharm).

3.5.1 Compatibilidad entre programas

La instalación de DigSILENT-PowerFactory es posible mediante la licencia proporcionada a la ESPOL, cabe mencionar que para el proyecto integrador se utiliza las computadoras del Laboratorio de la Facultad de Ingeniería en Electricidad y Computación (FIEC). El laboratorio de computación cuenta con un sin número de herramientas para uso estudiantil con sus respectivas licencias, una de estas es DigSILENT-PowerFactory 2020. El software Python en su versión 3.8 con su respectivo interprete o IDE Pycharm Community de igual versión, estas herramientas se encuentran incluidas en el escritorio del computador.

En consecuencia, se debe establecer relación entre la versión de Python y Pycharm, también esto debe cumplirse para DigSILENT-PowerFactory, pero es mucho más complejo ya que la selección de la versión del Python proporcionado por DigSILENT-PowerFactory debe hacerse desde la carpeta donde se instaló. La instalación debe hacerse secuencial, es decir primero DigSILENT-PowerFactory, para seguido ir a la carpeta de instalación y tomar la decisión de la versión de Python con la que se desea desarrollar la programación, una vez identificado se instala el intérprete de Python denominado Pycharm Community, en donde se lo ajusta como ambiente de programación a la versión de Python seleccionada previamente. A continuación, se presenta en la **Figura 3.32** la carpeta en donde se almacenan las versiones de Python en DigSILENT-PowerFactory que contienen el módulo “powerfactory.pyd”.

C:\Program Files\DlgSILENT\PowerFactory 2021 SP5\Python\3.8 es la dirección que se escribe en el código fuente para la selección de la versión de Python, para el estudio realizado se selecciona la versión 3.8, mismo que se observa en la **Figura 3.33** y **Figura 3.34**.

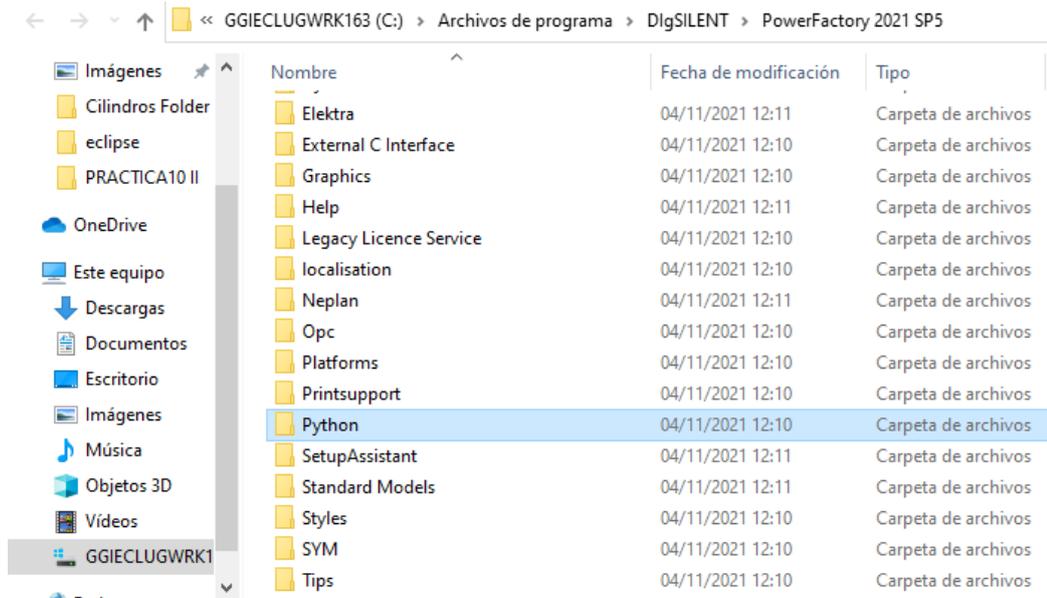


Figura 3.32. Localización del módulo powerfactory.pyd en el ordenador

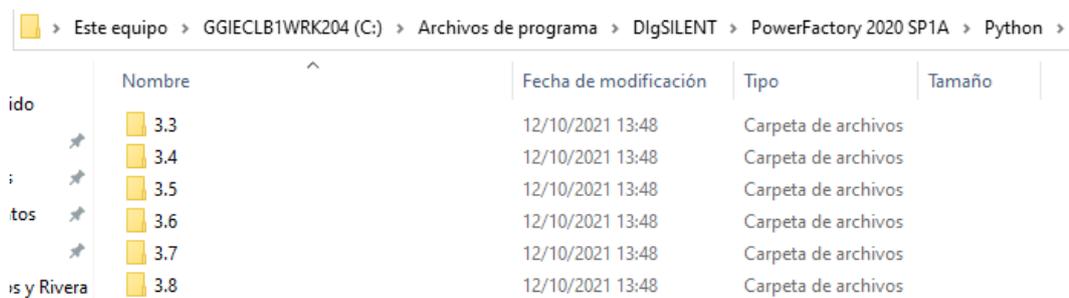


Figura 3.33. Versiones del módulo powerfactory.pyd

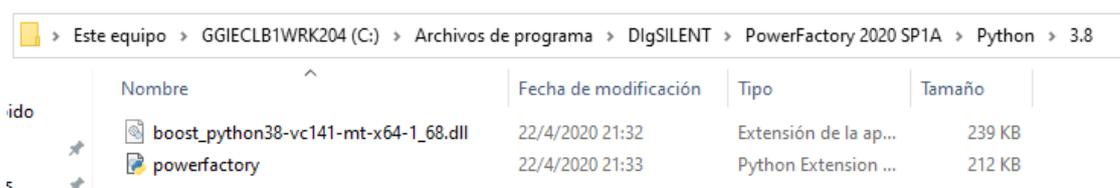


Figura 3.34. Módulo powerfactory.pyd para la versión de Python 3.8

Se debe mencionar que las versiones deben considerar las arquitecturas del sistema operativo, es decir:

- DigSILENT-PowerFactory de 32 bits debe considerar un intérprete de Python de 32 bits;
- DigSILENT-PowerFactory de 64 bits debe considerar un intérprete de Python de 64 bits.

El IDE de Python 3.8 es Pycharm Community, este es un editor de código fuente que también debe considerar la arquitectura, ya que la selección de la versión de intérprete de Python es interna, es decir ya en el programa, en la **Figura 3.35** se muestra el ajuste realizado en Pycharm Community para que opere de manera correcta con el intérprete de Python 3.8.

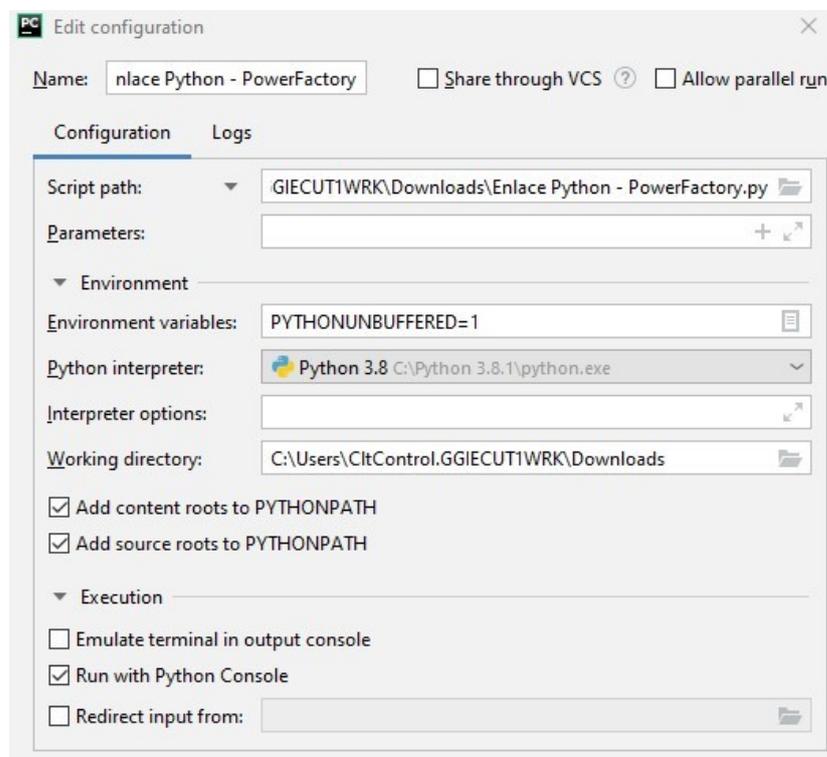


Figura 3.35. Ajuste del IDE para que use el intérprete de Python 3.8

La compatibilidad de herramientas tanto en versiones como en arquitectura del sistema operativo a utilizar representa limitantes para un enlace efectivo, arrojando errores de acoplamiento mostrados por el sistema operativo que el usuario este utilizando.

3.5.2 Módulo powerfactory

La interacción con la interfaz de programación de aplicación (API) de DigSILENT-PowerFactory que establece Python se manifiesta cuando se llama al módulo de simulación dinámica de Python denominado “powerfactory”. El enlace que se da al llamar al módulo desde el editor de texto (Script) de Pycharm Community, brinda información relacionada con la fábrica del entorno Power, tales como:

- Objetos;
- Atributos (datos del elemento, datos del tipo, resultados);
- Comandos (estudio de flujo de carga, estudio de cortocircuito, etc.);
- Funciones integradas especiales en un su gran parte, estas son consideradas funciones DPL [5].

Para obtener las funcionalidades de Python sobre DigSILENT-PowerFactory se importa la librería o almacenamiento de funciones destinadas para la interacción con la API de DigSILENT-PowerFactory, este proceso se da mediante la ejecución de la línea de código que se muestra en la **Figura 3.36**.

```
# Importar la aplicación  
import powerfactory as pf
```

Figura 3.36. Sentencia para enlazar Python con PowerFactory

El comando “import” cumple la función de traer al entorno todas las funciones que proporciona la librería o modulo especializado en la interacción de la API de DigSILENT-PowerFactory.

3.6 Ubicación óptima de fuentes eólicas y fotovoltaicas.

El comportamiento del sistema ante la inestabilidad de pequeña señal es de fácil estudio cuando se procede a realizar el respectivo análisis matemático, en donde se sustenta por medio de ecuaciones la relación existente entre variables. Las relaciones que se establecen en el proyecto consideran el amortiguamiento o ubicación del punto de operación posterior a la conexión y desconexión de una carga en el sistema. Es indispensable conocer cuando un sistema está en la capacidad de estabilizarse, así mismo los afectos que se presentan en los parámetros eléctricos del sistema como, la frecuencia, voltaje, potencia o ángulo de rotor. La ubicación de energías renovables está en función del análisis de estabilidad de pequeña señal realizado en el sistema. El sistema modificado IEEE de 39 barras, con distintos puntos donde es posible integrar la generación eólica y fotovoltaica, presenta variación en la estabilidad dependiendo del posicionamiento de la generación eólica y fotovoltaica, razón de ser para desarrollar una herramienta que realice el estudio para un generador definido y a la vez creando un evento de desconexión y conexión de carga, considerando la iteración por cada una de las barras del sistema para integrar la generación renovable no convencional. Se define la ubicación óptima a las barras en donde se conecta los generadores eólicos y fotovoltaicas y la vez se produce la mejor estabilidad en el sistema, por ello se establece un índice de estabilidad a maximizar con sus respectivas restricciones del caso. Finalmente, la simulación mediante el control maestro de Python y el modo esclavo de DigSILENT-PowerFactory en donde se determine la mejor posición para integrar fuentes eólicas y fotovoltaicas a partir del análisis de estabilidad de pequeña señal.

3.6.1 Modelo matemático de estabilidad de pequeña señal

El sistema de potencia, siendo el conjunto de varios elementos, es complejo de tal manera que el comportamiento se lo modela según la identidad dinámica que representa.

La incorporación de una perturbación pequeña de ángulo se la modela mediante la **Ecuación 3.1** de oscilación.

$$\delta = \delta_0 + \Delta\delta \quad (3.1)$$

Donde:

δ : perturbación pequeña;

δ_0 : perturbación pequeña inicial;

$\Delta\delta$: tasa de cambio de la perturbación pequeña.

A partir de la **Ecuación 3.2** de Swing, misma que está en función del modelo de perturbación mencionado como **Ecuación 3.1** [1], [8].

$$\left(\frac{2H}{\omega_{rot}}\right)\left(\frac{d^2(\delta_0 + \Delta\delta)}{dt^2}\right) = P_{m(pu)} - P_{e.max(pu)} \sin(\delta_0 + \Delta\delta) \quad (3.2)$$

Aplicando la identidad trigonométrica de la suma al modelo de perturbación se dividen los términos de la perturbación como se ve en la **Ecuación 3.3**.

$$\sin(\delta_0 + \Delta\delta) = \sin(\delta_0) \cos(\Delta\delta) + \cos(\delta_0) \sin(\Delta\delta) \quad (3.3)$$

Evaluando en la **Ecuación 3.2** en la **Ecuación 3.3** se tiene que:

$$\left(\frac{2H}{\omega_{rot}}\right)\left(\frac{d^2(\delta_0 + \Delta\delta)}{dt^2}\right) = P_{m(pu)} - P_{e.max(pu)}(\sin(\delta_0) \cos(\Delta\delta) + \cos(\delta_0) \sin(\Delta\delta)) \quad (3.4)$$

El estudio está en función de pequeñas perturbaciones, por lo que es posible realizar las siguientes aproximaciones:

$$\cos(\Delta\delta) \cong 1 \quad (3.5)$$

$$\sin(\Delta\delta) \cong \Delta\delta \quad (3.6)$$

Aplicando las aproximaciones correspondientes a la **Ecuación 3.5** y **Ecuación 3.6** en la **Ecuación 3.4** se obtiene la siguiente representación algebraica:

$$\left(\frac{2H}{\omega_{rot}}\right)\left(\frac{d^2(\delta_0)}{dt^2}\right) + \left(\frac{2H}{\omega_{rot}}\right)\left(\frac{d^2(\Delta\delta)}{dt^2}\right) = P_{m(pu)} - P_{e.max(pu)}(\sin(\delta_0) + \cos(\delta_0)\Delta\delta) \quad (3.7)$$

En la **Ecuación 3.7** se observa la siguiente relación:

$$\left(\frac{2H}{\omega_{rot}}\right)\left(\frac{d^2(\delta_0)}{dt^2}\right) = P_{m(pu)} - P_{e.max(pu)}(\sin(\delta_0)) \quad (3.8)$$

Evaluando la **Ecuación 3.8** en la **Ecuación 3.7** y su vez separando términos se presenta la siguiente expresión matemática:

$$P_{m(pu)} - P_{e.max(pu)}(\sin(\delta_0)) + \left(\frac{2H}{\omega_{rot}}\right)\left(\frac{d^2(\Delta\delta)}{dt^2}\right) = P_{m(pu)}(\sin(\delta_0)) - P_{e.max(pu)}(\cos(\delta_0)\Delta\delta) \quad (3.9)$$

Despejando e igualando a cero la **Ecuación 3.9** finalmente obtenemos la **Ecuación 3.10** representativa en función de la potencia de sincronización [1], [8]:

$$\left(\frac{2H}{\omega_{rot}}\right)\left(\frac{d^2(\Delta\delta)}{dt^2}\right) + P_{e.max(pu)}(\cos(\delta_0)\Delta\delta) = 0 \quad (3.10)$$

Donde:

$P_{e.max(pu)}(\cos(\delta_0))$: pendiente de la curva del ángulo de potencia en δ_0

Se la representa también de la siguiente manera:

$$P_s = \left.\frac{dP}{d\delta}\right|_{\delta_0} = P_{e.max(pu)}(\cos(\delta_0)) \quad (3.11)$$

Donde:

P_s : potencia de sincronización

Caso 1: La ecuación de oscilación con perturbación pequeña de ángulo considera el amortiguamiento.

La diferencia que existe en las velocidades angulares en la maquina síncrona, se ve afectada frente la presencia de perturbaciones, ya que la velocidad angular del rotor y el campo giratorio que se forma en el entrehierro disminuyen el torque neto del rotor, a este se lo conoce también como torque de amortiguamiento (D), las unidades están en por unidad (pu). En consecuencia, al desarrollo matemático se define la potencia de amortiguación P_d representado de la siguiente manera:

$$P_d = D \frac{d\delta}{dt} \quad (3.12)$$

Donde:

P_d : Potencia de amortiguación;

D : Torque de amortiguamiento.

A partir de la integración del concepto de potencia de sincronización y amortiguamiento en la **Ecuación 3.2** de Swing [1], [8], obtenemos la siguiente expresión:

$$\frac{d^2(\Delta\delta)}{dt^2} + \frac{\pi f_0}{H} \left(D \frac{d\Delta\delta}{dt} \right) + \frac{\pi f_0}{H} P_s \Delta\delta = 0 \quad (3.13)$$

Donde:

H : Constante de inercia normalizada [J/VA].

La solución de la **Ecuación 3.13** se presenta de manera rápida, pero si desea ver el proceso de desarrollo sobre la ecuación diferencial de orden dos puede dirigirse a la sección de **Anexo 1** [1], [8].

$$\Delta\delta = \frac{\Delta\delta_0}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta) \quad (3.14)$$

Donde:

La frecuencia natural es representada mediante la siguiente ecuación:

$$\omega_n = \sqrt{\frac{\pi f_0}{H} P_s} \quad (3.15)$$

El coeficiente de amortiguamiento:

$$\zeta = \frac{D}{2} \sqrt{\frac{\pi f_0}{H P_s}} < 1 \quad (3.16)$$

La frecuencia de amortiguamiento:

$$\omega_d = \omega_n \sqrt{1-\zeta^2} \quad (3.17)$$

El ángulo de amortiguamiento:

$$\theta = \cos^{-1}(\zeta) \quad (3.18)$$

Dinámica de la máquina sincrónica ante pequeñas perturbaciones de ángulo

La cuantificación de la perturbación pequeña de ángulo representa la apertura para definir el comportamiento dinámico del sistema mediante las siguientes ecuaciones, cabe mencionar que esto es entorno a la máquina síncrona [1], [8].

$$\delta(t) = \delta_0 + \frac{\Delta\delta_0}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta) \quad (3.19)$$

$$\omega(t) = \omega_0 + \frac{\omega_n \Delta\delta_0}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t) \quad (3.20)$$

$$\tau = \frac{1}{\zeta\omega_n} \quad (3.21)$$

Caso 2: La ecuación de oscilación con perturbación pequeña de ángulo considera el amortiguamiento y la variación de potencia.

El comportamiento dinámico del sistema en función de la variación de la potencia sugiere cambios en la forma de las respuestas, esto en comparación al supuesto que se plantea en el caso 1 en donde se considera que la perturbación no genera cambios en la potencia. Es decir, si ocurre que la potencia de entrada incrementa en una pequeña cantidad (ΔP), la ecuación de oscilación linealizada que modela el comportamiento del sistema se presenta de la siguiente manera [1], [8]:

$$\frac{H}{\pi f_0} \left(\frac{d^2(\Delta\delta)}{dt^2} \right) + D \frac{d\Delta\delta}{dt} + P_s \Delta\delta = \Delta P \quad (3.22)$$

$$\frac{d^2(\Delta\delta)}{dt^2} + \frac{\pi f_0}{H} \left(D \frac{d\Delta\delta}{dt} \right) + \frac{\pi f_0}{H} (P_s \Delta\delta) = \frac{\pi f_0}{H} (\Delta P) \quad (3.23)$$

La solución de la **Ecuación 3.23** se presenta de manera rápida, pero si desea ver el proceso de desarrollo es recomendable dirigirse al **Apéndice A** [1], [8].

$$\Delta\delta = \frac{\Delta u}{\omega_n^2} \left[1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta) \right] \quad (3.24)$$

Luego:

$$\Delta\omega = \frac{d\Delta\delta}{dt} = -\frac{\pi f_0 \Delta P}{H \omega_n \sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t) \quad (3.25)$$

El modelamiento del comportamiento de las respuestas del sistema frente perturbaciones permiten la realización de estudios de estabilidad de pequeña señal, y estos a su vez contribuyen con la mejora de la confiabilidad, calidad

energía y seguridad del sistema para proveer de energía eléctrica a la demanda diaria requerida.

3.6.2 Proceso matemático del código fuente en Python

A continuación, se describirá el proceso matemático que realiza el código fuente en Python tomando como referencia el sistema de 6 barras con generación eólica y fotovoltaica que se presentó en la **Figura 3.24**.

Para este ejemplo, se considerará un estudio de 2 partículas con el fin de observar los cálculos internos que realiza el código fuente y su respectiva descripción para un mejor entendimiento del proceso PSO aplicado a arreglos.

Inicialmente todos los generadores se encuentran desconectados y, desde Python, interpretan secuencias de números para su operación, siendo 0 para mantenerse desconectados y 1 para conectarse a la barra correspondiente. Cada número de la partícula controla el accionamiento de cada generador respectivamente. Como se está trabajando con 6 barras, la partícula será un arreglo de 12 números, en donde a cada par de números le corresponde trabajar con una barra siguiendo el orden con el que fueron nombradas. A ese par de números, el primer elemento controla el generador fotovoltaico y el segundo elemento el generador eólico de dicha barra en particular.

Las ecuaciones de velocidad y posición, siguiendo el proceso del método PSO, serán las siguientes:

$$v_j(i + 1) = w * v_j(i) + c1r1(BL_j - x_j(i)) + c2r2(BG - x_j(i)) \quad (3.26)$$

$$x_j(i + 1) = x_j(i) + v_j(i + 1) \quad (3.27)$$

Donde:

i : número de iteración.

j : número de partícula.

w : factor de inercia en rango de 0.4 a 0.9.

c_1 y c_2 : constantes de atracción al mejor personal. Ambas serán iguales a 2.

r_1 y r_2 : números aleatorios entre 1 y la dimensión de la partícula, que para este caso será 12.

BL_j : mejor posición en la que ha estado la partícula j hasta el momento.

BG : mejor posición de todo el enjambre hasta el momento.

Cada configuración proporcionada por la partícula dará como resultado un archivo *.txt* que mostrará los valores de frecuencia del generador a lo largo del tiempo de simulación. A partir de este archivo se realizará el cálculo del índice.

$$Indice = I_j(i + 1) = \frac{60}{Promedio\ de\ Frecuencias\ de\ la\ posición\ x_j(i + 1)} \quad (3.28)$$

El Código Fuente se encarga de leer los archivos correspondientes a cada partícula y se obtendrá el promedio de frecuencias de dicha configuración. Mediante la **Ecuación 3.28** se calculará el índice que se usará en el análisis del método PSO.

Una vez descritas las ecuaciones que se implementaron dentro del código fuente, se continuará con el ejemplo.

$i = 0$

Inicialmente, el algoritmo creará valores aleatorios para el arreglo de cada partícula y se calculará el índice de la configuración.

- *Partícula 1* = $x_1(1) = [1,1,1,1,0,0,0,0,1,1,1,1] \rightarrow I_1(1)$
- *Partícula 2* = $x_2(1) = [0,0,1,1,0,0,1,1,0,0,1,1] \rightarrow I_2(1)$

Como ninguna partícula se ha movido, la velocidad inicial para ambas será 0.

- $v_1(1) = [0,0,0,0,0,0,0,0,0,0,0,0]$
- $v_2(1) = [0,0,0,0,0,0,0,0,0,0,0,0]$

De igual manera, como no hay otra posición con la que comparar, cada partícula tendrá un Best Local de 0 inicialmente.

- $BL_1 = [0,0,0,0,0,0,0,0,0,0,0,0]$
- $BL_2 = [0,0,0,0,0,0,0,0,0,0,0,0]$

La elección del Best Global realiza comparando los índices de cada partícula a lo largo de todas las iteraciones y eligiendo al mejor. El índice se considera mejor cuanto más cercano se encuentre del valor de 1.

Asumiendo que $I_1(1) > I_2(1)$, el Best Global será:

$$BG = x_1(1) = [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1]$$

$i = 1$

Se utilizará la **Ecuación 3.26** y la **Ecuación 3.27** para calcular la velocidad y la posición de cada partícula respectivamente. Para métodos prácticos haremos uso de las siguientes asunciones:

$$w = 0.9$$

$$r1 = r2 = 1$$

Las velocidades son:

$$\begin{aligned} \bullet \quad v_1(2) &= w * v_1(1) + c1r1(BL_1 - x_1(1)) + c2r2(BG - x_1(1)) \\ &= 0.9 * [0,0,0,0,0,0,0,0,0,0,0,0] + 2([0,0,0,0,0,0,0,0,0,0,0,0] - [1,1,1,1,0,0,0,0,1,1,1,1]) \\ &\quad + 2([1,1,1,1,0,0,0,0,1,1,1,1] - [1,1,1,1,0,0,0,0,1,1,1,1]) \\ &= [-2, -2, -2, -2, 0, 0, 0, 0, -2, -2, -2, -2] \end{aligned}$$

Para que se pueda leer la configuración en DigSILENT-PowerFactory, los valores se convierten a binario.

$$= [-2, -2, -2, -2, 0, 0, 0, 0, -2, -2, -2, -2] \equiv [0,0,0,0,0,0,0,0,0,0,0,0]$$

- $$v_2(2) = w * v_2(1) + c1r1(BL_2 - x_2(1)) + c2r2(BG - x_2(1))$$

$$= 0.9 * [0,0,0,0,0,0,0,0,0,0,0,0] + 2([0,0,0,0,0,0,0,0,0,0,0,0] - [0,0,1,1,0,0,1,1,0,0,1,1])$$

$$+ 2([1,1,1,1,0,0,0,0,1,1,1,1] - [0,0,1,1,0,0,1,1,0,0,1,1])$$

$$= [2,2, -2, -2,0,0, -4, -4,2,2, -2, -2] \equiv [1,1,0,0,0,0,0,0,1,1,0,0]$$

Las posiciones son:

- $$x_1(2) = x_1(1) + v_1(2) = [1,1,1,1,0,0,0,0,1,1,1,1] + [0,0,0,0,0,0,0,0,0,0,0,0] =$$

$$[1,1,1,1,0,0,0,0,1,1,1,1] \rightarrow I_1(2)$$
- $$x_2(2) = x_2(1) + v_2(2) = [0,0,1,1,0,0,1,1,0,0,1,1] + [1,1,0,0,0,0,0,0,1,1,0,0] =$$

$$[1,1,1,1,0,0,1,1,1,1,1,1] \rightarrow I_2(2)$$

Se eligen los Best Local comparando con las posiciones anteriores de cada partícula.

- Como $I_1(1) = I_1(2)$

$$Best\ Local_1 = x_1(1) = x_1(2) = [1,1,1,1,0,0,0,0,1,1,1,1]$$
- Asumiendo que $I_2(1) < I_2(2)$

$$Best\ Local_2 = x_2(2) = [1,1,1,1,0,0,1,1,1,1,1,1]$$

En la iteración previa se eligió el $BG = x_1(1)$ dado que $I_1(1) > I_2(1)$.

Como $I_1(1) = I_1(2)$ y asumiendo que $I_1(2) > I_2(2)$, el Best Global será:

$$BG = x_1(2) = [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1]$$

Para este caso en particular, el Best Global no cambia.

El proceso matemático que maneja el código fuente es el mismo para cada iteración. Al final se obtiene el Best Global óptimo con el índice de estabilidad más alto.

3.6.3 Caso de estudio

El sector eléctrico identifica problemas en la pérdida del sincronismo en las máquinas síncronas, esto se refleja en la estabilidad del sistema en presencia de perturbaciones. La obtención de señales de frecuencia, potencia y voltaje son de los parámetros que en lo ideal después de verse afectado por una perturbación deben amortiguarse de tal manera que encuentren el punto de estado estable, considerando estándares aceptables de operación del sistema. La clasificación de la estabilidad de sistemas de potencia presentada en la **Figura 2.1** establece eventos de pequeña señal, motivo de estudio para este proyecto en función de la desconexión y conexión de carga en el sistema.

La iniciativa de pensamiento verde, la necesidad de establecer una relación directa entre el hombre y la naturaleza como seres únicos y dependientes entre sí, es el requerimiento a nivel mundial debido al cambio climático y las consecuencias que este representa para las generaciones futuras. Las medidas frente a estos eventos, es la sustitución de fuentes de energía no renovables, mediante la implementación de sistemas que generen cantidad de energía en misma magnitud, pero sin causar daño a nivel medio ambiental. Las energías renovables son una solución ya que toman en cuenta recursos primarios como el sol, viento, marea, materia orgánica, etc.

La implementación de fuentes de generación eólica y fotovoltaica están tomando fuerza en las organizaciones gubernamentales a nivel mundial, ya que sustituye fuentes de generación que requieran de combustibles fósiles como las centrales térmicas. Es importante considerar que lo ideal debe ser que, la gran parte de la demanda del sector eléctrico o promedio sea cubierta por generación eólica y fotovoltaica y que la central térmica opera en horas de demanda pico, dado que la característica principal de esta central es su rápido ingreso de operación.

La integración de fuentes de generación renovables no convencionales, específicamente hablando de generación eólica y fotovoltaica, influye en la estabilidad de pequeña señal del sistema, esto se debe a la variación que existe

en la dinámica del sistema. El efecto hace referencia a la conexión y desconexión de carga, ya que este repercute en el punto de operación original, colocando segundos después en estado estable, pero en otro nivel de operación. El efecto provocado en la estabilidad de pequeña señal debe ser consecuente con la penetración en el sistema, ya que la penetración depende del porcentaje de demanda que se satisface por las fuentes de generación mencionadas anteriormente.

El caso de estudio presentado en este proyecto se enfoca en el análisis de estabilidad de pequeña señal como mecanismo de indicador, para la integración de fuentes no convencionales, específicamente hablando de generación eólica y fotovoltaica. El sistema de prueba utilizado para la realización del estudio es el sistema modificado IEEE de 39 barras [29].

La ubicación se establece por las barras, es decir existen 39 posibles posiciones en donde se podría colocar tanto la generación eólica como fotovoltaica, pero en la realidad este proceso se restringe por la afluencia del recurso en la locación. Para ello se establece barras en el sistema donde no será posible integrar generación renovable no convencional. La obtención del mejor escenario posible debe ser estudiado mediante las respuestas del sistema cuando se realiza el estudio de flujo de potencia, análisis transitorio de las señales de frecuencia, voltaje y potencia activa en el dominio del tiempo.

3.6.4 Sistema modificado IEEE de 39 barras

El sistema de prueba modificado IEEE de 39 barras, es un modelo de sistema eléctrico de potencia basado en la red eléctrica de alta tensión de Nueva Inglaterra que se encuentra localizado en los Estados Unidos (US), cabe señalar que el sistema de IEEE de 39 barras sin modificar es simplificado con respecto al real. En resumen, los elementos que conforman la red eléctrica son los descritos en la **Tabla 3.1**.

El sistema modificado IEEE de 39 barras cuenta con el equipamiento de todos los generadores con un excitador IEEE tipo 1 y un regulador de turbina simple, esto se restringe solo para el generador 39, ya que este representa una incorporación de una gran cantidad de generadores, motivo por el cual no le corresponde un gobernador. La **Figura 3.37** muestra el sistema de prueba a simular en el software especializado en cálculos y estudios eléctricos DigSILENT-PowerFactory.

Tabla 3.1. Elementos de la Red Modificada IEEE 39 Barras

Elementos	Cantidad
Barras	49
Generadores	10
Transformadores	24
Líneas de transmisión	32
Cargas	19

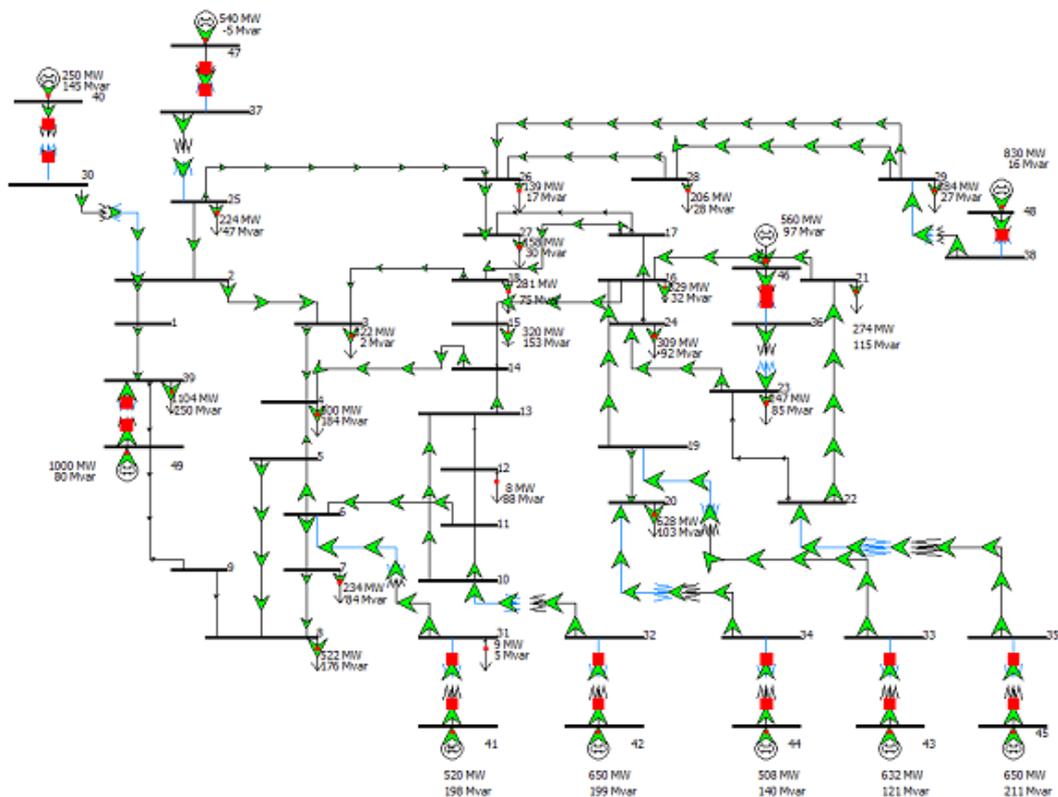


Figura 3.37. Sistema modificado IEEE de 39 barras [29]

La información de ajuste de parámetros para los elementos que conforman el sistema modificado IEEE de 39 barras se presenta en la **Tabla 3.2** y **Tabla 3.3** que corresponden a los generadores síncronos y las cargas o demanda respectivamente, que conforman el sistema eléctrico de potencia modificado proporcionado por la IEEE.

La **Tabla 3.2** muestra que la potencia activa de generación es de 5366 MW, así también que no existe potencia reactiva generada, finalmente que la potencia nominal aparente de generación es de 8257 MVA.

El sistema modificado IEEE de 39 barras cuenta con 9 barras de generación (PV) y una barra de referencia o denominada Slack, esta corresponde al generador número 2 (Gen2), ubicada en la barra 41 (Bus41).

Los ajustes mencionados anteriormente son realizados en el diseño eléctrico importado y activado en el software DigSILENT-PowerFactory.

Tabla 3.2. Generadores Síncronos del Sistema Modificado IEEE de 39 Barras

Nombre del generador	Nombre de la barra	Tipo de barra	Voltaje nominal [kV]	Potencia aparente nominal [MVA]	Potencia activa [MW]	Potencia reactiva [Mvar]	Factor de potencia	Número de unidades en paralelo
Gen1	Bus49	PV	26	911	1000	0	0.9	1
Gen2	Bus41	SL	20	835	0	0	0.9	1
Gen3	Bus42	PV	20	835	650	0	0.9	1
Gen4	Bus43	PV	20	835	632	0	0.9	1
Gen5	Bus44	PV	20	835	254	0	0.9	2
Gen6	Bus45	PV	20	835	650	0	0.9	1
Gen7	Bus46	PV	20	835	560	0	0.9	1
Gen8	Bus47	PV	20	835	540	0	0.9	1
Gen9	Bus48	PV	26	911	830	0	0.95	1
Gen10	Bus40	PV	22	590	250	0	0.95	1

En la **Tabla 3.3** se observa que la potencia activa total demanda por el sistema es 6097.1 MW y para la potencia reactiva y 1408.8 Mvar. Los aspectos constructivos de las máquinas que conforman el sistema eléctrico se encuentran detalladas en el **Apéndice B**.

3.6.5 Integración de PSS y AVR

Dentro de la librería de DigSILENT-PowerFactory se encuentran un sin números de modelos y diagramas de bloques que pueden servir de control para nuestro sistema.

Tabla 3.3. Demanda del Sistema Modificado IEEE de 39 Barras

Nombre de la carga	Nombre de la barra	Potencia activa [MW]	Potencia reactiva [Mvar]
Load3	Bus3	322.0	2.4
Load4	Bus4	500.0	184.0
Load7	Bus7	233.8	84.0
Load8	Bus8	522.0	176.0
Load12	Bus12	7.5	88.0
Load15	Bus15	320.0	153.0
Load16	Bus16	329.0	32.3
Load18	Bus18	158.0	30.0
Load20	Bus20	628.0	103.0
Load21	Bus21	274.0	115.0
Load23	Bus23	247.5	84.6
Load24	Bus24	308.6	-92.2
Load25	Bus25	224.0	47.2
Load26	Bus26	139.0	17.0
Load27	Bus27	281.0	75.5
Load28	Bus28	206.0	27.6
Load29	Bus29	283.5	26.9
Load31	Bus31	9.2	4.6
Load39	Bus39	1104.0	250.0

La integración de PSS y AVR se ha realizado mediante la implementación del modelo “SYM Frame_no droop” que consta de los dos tipos de controles. Previamente deberán definirse parámetros dentro del **Composite Model**, **Composite Frame** y **Common Model** para que el diagrama de bloques lea las entradas/salidas del sistema.

Modelo Compuesto

Un modelo compuesto (Composite Model) es una estructura de datos asociada a una unidad generadora que contiene todos los modelos y elementos relacionados con la máquina. Cada elemento controla un apartado específico de la unidad generadora, los cuales pueden ser seleccionados de la librería de DigSILENT-PowerFactory o crearse según las necesidades del usuario. [30]

DigSILENT-PowerFactory permite configurar las ranuras del modelo compuesto para que sea controlado por un elemento que constituye una función específica. Para nuestro proyecto se asociarán las siguientes ranuras, estas se observan en la **Figura 3.38**.

- Sym Slot: se ingresa la unidad generadora con la que se desea trabajar.
- Avr Slot: se ingresa el modelo del regulador automático de voltaje (AVR) de la unidad generadora.
- Gov Slot: se ingresa el modelo del gobernador de velocidad de la unidad generadora.
- Pss Slot: se ingresa el estabilizador de potencia (PSS) de la unidad generadora.

Este control permite optimizar el amortiguamiento de las señales del generador eléctrico logrando una mejor estabilización frente a cualquier perturbación.

Marco Compuesto

Un marco compuesto (Composite Frame) es una estructura de conexiones formada por sistemas de control que se encargan de definir los elementos asociados en el modelo compuesto. Se representa como un diagrama de bloques con ranuras a las cuales se asocian señales de entrada y salida. [31]

Nuestro proyecto usará un marco compuesto de la librería de DigSILENT-PowerFactory denominado "SYM Frame_no droop" que se muestra a continuación en la **Figura 3.39**, que contiene slots para incorporar sistemas AVR y PSS.

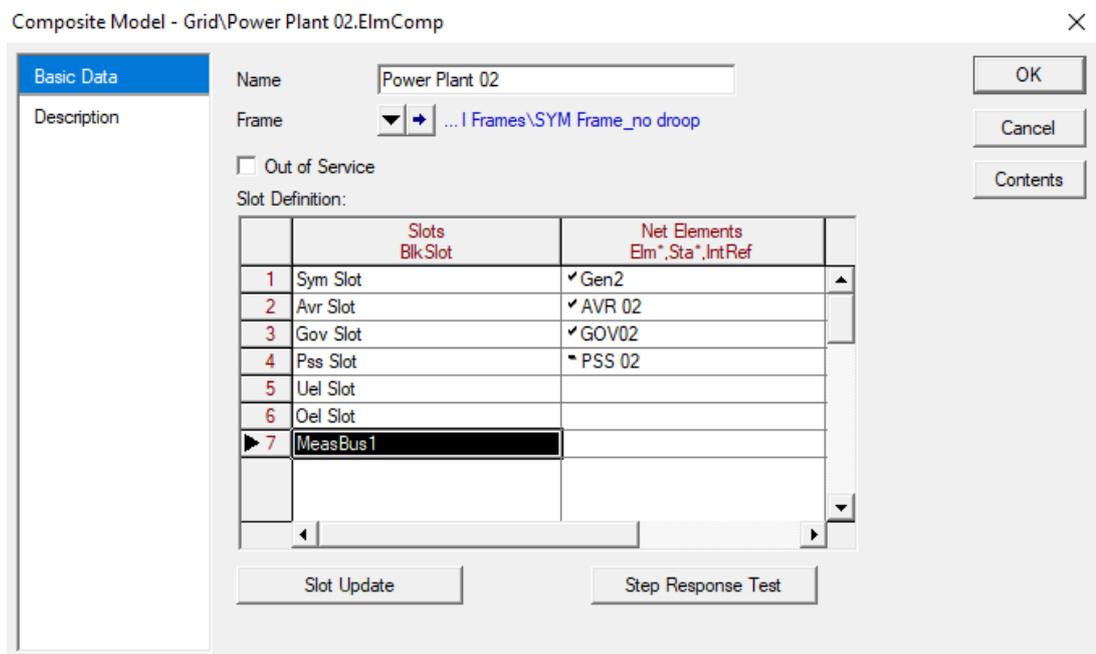


Figura 3.38. Modelo compuesto de la unidad generadora Gen2

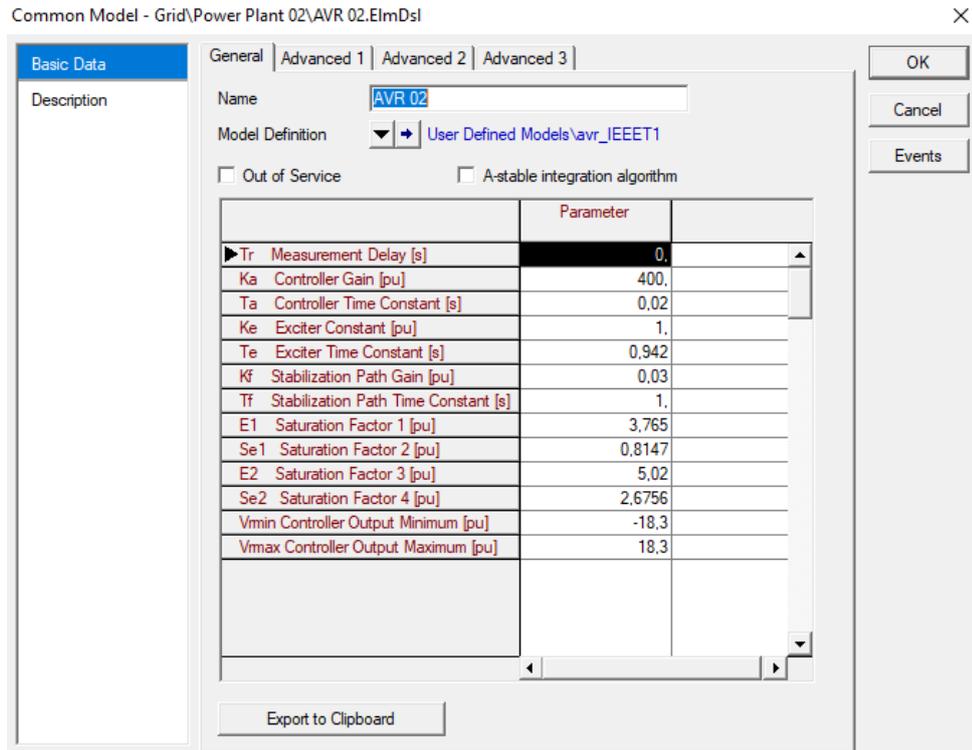


Figura 3.40. Common Model para el AVR del Gen2

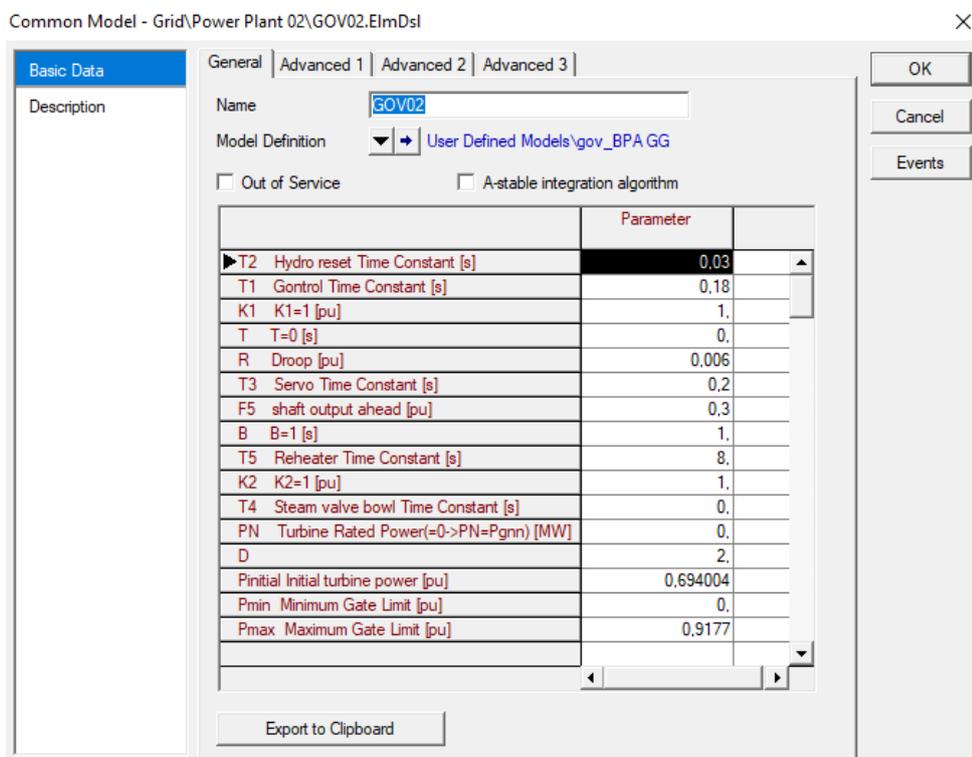


Figura 3.41. Common Model para el gobernador del Gen2

Se muestran los modelos generales para cada elemento del generador *Gen2*. Dichos parámetros fueron configurados automáticamente dentro del modelo IEEE 39-bus que consta en los sistemas de prueba dinámicos de la IEEE. Las demás unidades generadoras, a excepción del *Gen1*, fueron configurados con los mismos parámetros bajo el marco compuesto descrito con anterioridad.

3.6.6 Índice de estabilidad

El índice de estabilidad (z) dado por la **Ecuación 3.29**, es una función de datos tabulados en un archivo de texto con extensión “.txt”, estos datos representan la frecuencia de oscilación del sistema, considerando la perturbación cuando se conecta y desconecta una carga del sistema y a la vez se integra generación eólica y fotovoltaica al sistema modificado IEEE de 39 barras. Los datos se obtienen mediante el estudio de estabilidad de pequeña señal realizado con ayuda del software especializado en cálculos y estudios eléctricos DigSILENT-PowerFactory. La tabulación de datos está formada por valores de frecuencia (f), tomados en un instante de tiempo (t), durante un tiempo de simulación (t_{sim}).

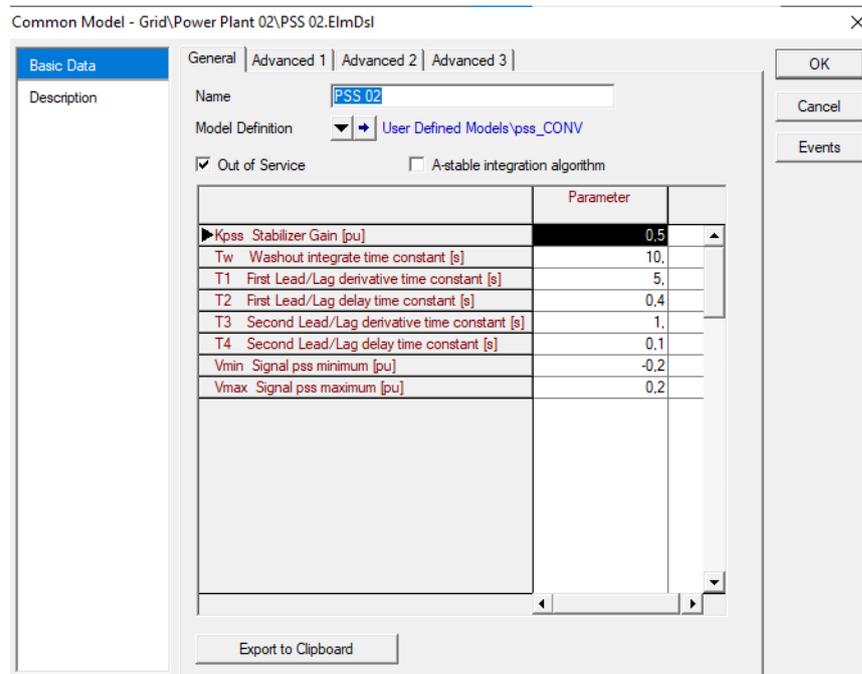


Figura 3.42. Common Model para el PSS del Gen2

El índice de estabilidad considera el valor promedio de las (n) mediciones, dividido entre el valor de frecuencia estándar (f) en la región en donde está sujeto el sistema. El índice (z) toma valores entre 0 y 1.

3.6.7 Modelamiento matemático

La **Ecuación 3.29** representa el modelo matemático de índice de estabilidad, este corresponde a la función objetivo a maximizar, ya que se desea obtener la mayor estabilidad en función de la frecuencia de oscilación en el sistema. Las **desigualdades 3.31 – 3.33** corresponden a las restricciones de no negatividad.

Maximizar:

$$z = \frac{f}{\sum_{i=1}^n \frac{x_i}{n}} \quad (3.29)$$

Donde:

z : índice de estabilidad;

x_i : representación de la medición de frecuencia en un instante t en Hz;

n : número de mediciones realizadas durante el tiempo de simulación t_{sim} ;

f : frecuencia estándar en Hz.

Sujeto a:

Restricciones de función objetivo: La **Restricción 3.30** representa el rango en el cual se debe encontrar el índice de estabilidad en cada iteración por partícula analizada en el sistema eléctrico, de la misma manera se establece las **Restricciones 3.31 – 3.33** como no negatividad, limitando así también que el denominador de la **Ecuación 3.29** tome el valor de 0, que sería un problema ya que infiere una indeterminación matemática.

$$0 \leq z \leq 1 \quad (3.30)$$

$$x_i > 0, \quad i = 1 \dots n \quad (3.31)$$

$$n > 0 \quad (3.32)$$

$$f \geq 0 \quad (3.33)$$

Restricciones del estudio de flujo de potencia: La **Restricción 3.34** representa el equilibrio de potencia activa (P) que debe existir en el sistema, de igual manera para la **Restricción 3.35** pero en términos de la potencia reactiva (Q). Se debe considerar que i es el número de barra sobre el cual se ejecuta el estudio y n el número total de barras en el sistema. La **Restricción 3.36** considera los valores de límites de tensión en el sistema, de igual forma la **Restricción 3.37** con respecto a la variable eléctrica de corriente [32].

$$P_{red} - \sum_{i=1}^n P_{pérdidas}(i) = \sum_{i=1}^n P_{carga}(i), \quad i = 1, 2, \dots, n \quad (3.34)$$

$$Q_{red} - \sum_{i=1}^n Q_{pérdidas}(i) = \sum_{i=1}^n Q_{carga}(i), \quad i = 1, 2, \dots, n \quad (3.35)$$

$$V_{p.u. \min} \leq |V_{p.u.}(i)| \leq V_{p.u. \max}, \quad i = 1, 2, \dots, n \quad (3.36)$$

$$0 \leq |I_{p.u.}(i)| \leq I_{p.u. \max}, \quad i = 1, 2, \dots, n \quad (3.37)$$

El modelo matemático establecido, es el mecanismo de búsqueda de la mejor posición en donde se debe integrar generación eólica y fotovoltaica en el sistema de potencia sobre el cual se realiza el estudio.

3.6.8 Diseño de código fuente en Python

Para el diseño del código fuente, se utilizó el lenguaje de programación Python en su versión 3.8 y el IDE gratuito Pycharm Community Edition. El código fuente contiene los procesos matemáticos del método PSO aplicado a arreglos, en donde se establecerá su operación para n partículas y n iteraciones de modo que el usuario tenga control sobre el estudio que se realizará. Mediante el código fuente, se realizará el vínculo con DigSILENT-PowerFactory para establecer un control maestro y ocupar todas las herramientas que permita el enlace.

A continuación, se realizará una descripción del diseño del código fuente de Python para establecer un control maestro sobre nuestro sistema de 39 barras implementando en DigSILENT-PowerFactory.

Primero se debe de vincular Python con DigSILENT-PowerFactory. Para esto debemos de verificar la carpeta en la que se encuentra el software DigSILENT-PowerFactory dentro del ordenador en el cual estamos trabajando, que para nuestro caso será un ordenador de ESPOL y su directorio recurrente es *C:\Program Files\DigSILENT\PowerFactory 2020 SP1A*. En la **Figura 3.43** se muestra el enlace e importación de DigSILENT-PowerFactory hacia Python. El módulo “os” expone el ambiente de trabajo del programa mientras que el módulo “sys” extrae la carpeta que vincula a DigSILENT-PowerFactory con Python. Por último, se importa el programa y se le asigna su variable como “*pf*” para poder llamarlo en las siguientes líneas del código.

Python permite el acceso a una infinidad de funciones, sin embargo, muchas de ellas se encuentran dentro de módulos los cuales requieren de una importación previa para poder trabajarlos. De esa manera, la importación de librerías en nuestro entorno de trabajo es de suma importancia para poder acceder a estas funciones complementarias que faciliten el desarrollo del código fuente.

En la **Figura 3.44** se realiza la importación de las librerías *Pandas*, *Random* y *Numpy* con las variables respectivas para nombrarlas. La librería *Time* se importó para poder acceder a un control del tiempo en el proceso del código fuente que será detallado más adelante.

En la **Figura 3.45** se realiza la activación del proyecto de 39 barras en DigSILENT-PowerFactory, el cual se encuentra bajo el nombre de “*simulate39bus*”. Es importante aclarar que se debe realizar la importación previa del proyecto al entorno de DigSILENT-PowerFactory, caso contrario no se podrá activar ya que no lo encontrará en la lista de archivos importados.

Todos los elementos del sistema de 39 barras fueron nombrados bajo el mismo nombre según su tipo, solo cambiando con un número al final que identifica al elemento en particular. Esto se realizó con el fin de establecer un orden entre elementos del mismo tipo y poder identificarlos de una manera más sencilla al momento de trabajar con ellos.

El acceso a los directorios de DigSILENT-PowerFactory permite identificar a cualquier elemento del sistema y poder ordenarlos en una lista según su tipo, tal como se muestra en la **Figura 3.46**. De esa manera se agrupan todos los elementos de un mismo tipo y para acceder a ellos, solo basta con nombrar su posición dentro de la lista. Como se mencionó anteriormente, los elementos del sistema fueron configurados bajo el mismo nombre según su tipo para facilitar la búsqueda dentro de la lista y poder manejar un proceso ordenado.

```
# Dirección de DigSILENT-PowerFactory en el ordenador de ESPOL:
import os
os.environ['PATH']=r'C:\Program Files\DIGSILENT\PowerFactory 2020 SP1A'+os.environ['PATH']

# Vincular python con DigSILENT-PowerFactory
import sys
sys.path.append(r'C:\Program Files\DIGSILENT\PowerFactory 2020 SP1A\Python\3.8')

# Importar la aplicación
import powerfactory as pf
```

Figura 3.43. Enlace de Python con PowerFactory

```
# Importar la librería de pandas
import pandas as pd

# Importar la librería de manejo de tiempo o temporizador de procesos
import time

# Importar la librería de valores aleatorios
import random as rd

# Importar la librería de arreglos
import numpy as np
```

Figura 3.44. Importación de librerías

```
# Activación del proyecto
app = pf.GetApplication() # Llama a DigSILENT-PowerFactory

app.Show() # Mostrar aplicativo de DigSILENT-PowerFactory

user = app.GetCurrentUser()
project = app.ActivateProject('simulate39bus') # Buscar proyecto por nombre
prj = app.GetActiveProject() # Activación del proyecto ya una vez cargado a DigSILENT-PowerFactory
```

Figura 3.45. Activación del proyecto que contiene el sistema de 39 barras

El código fuente fue diseñado para que el usuario sea el que establezca la cantidad de partículas que se desea analizar y en cuantas iteraciones se quiere manejar el proceso del método PSO. Las **Figura 3.47** y **Figura 3.48** muestran la función para el ingreso por teclado de estos parámetros, aceptando solamente números enteros y de no ser así, solicitar un nuevo ingreso.

```

# Acceso de los directorios del proyecto
# Directorio de generadores fotovoltaicos
pv_dict = {}
pvs = app.GetCalcRelevantObjects('*.ElmPvsys') # Obtenemos lista de generadores fotovoltaicos
for i in pvs: # Obtienes cada elemento en la lista del directorio
    pv_dict[i.loc_name] = i

# Directorio de generadores eólicos
genst_dict = {}
gs = app.GetCalcRelevantObjects('*.ElmGenstat') # Obtenemos lista de generadores eólicos
for i in gs: # Obtienes cada elemento en la lista del directorio
    genst_dict[i.loc_name] = i

# Directorio de cargas
load_dict = {}
loads = app.GetCalcRelevantObjects('*.ElmLod') # Obtenemos lista de cargas
for i in loads: # Obtienes cada elemento en la lista del directorio
    load_dict[i.loc_name] = i

# Directorio de generadores síncronos
gen_dict = {}
gens = app.GetCalcRelevantObjects('*.ElmSym') # Obtenemos lista de generadores síncronos
for i in gens: # Obtienes cada elemento en la lista del directorio
    gen_dict[i.loc_name] = i

```

Figura 3.46. Acceso a directorios de elementos del sistema de 39 barras

```

# Función para el ingreso de partículas a implementarse
def ingreso_particulas(): #Devuelve el número ingresado por teclado, verificando si es entero
    flagParticulas = False
    while not flagParticulas:
        num_p = input('Ingrese número de partículas a analizar: ')
        if num_p.isnumeric():
            flagParticulas = True
        else:
            print("No ha ingresado un número entero. Intente nuevamente")
    return int(num_p)

```

Figura 3.47. Función para digitar el número de partículas

Como trabajaremos con partículas que contienen el accionamiento de generadores fotovoltaicos y eólicos, es importante identificar un orden en la configuración. Previamente identificamos que las posiciones pares de cada partícula trabajan con los generadores eólicos mientras que las posiciones impares trabajan con los generadores fotovoltaicos. Por eso es importante definir una función como la que se muestra en la **Figura 3.49**, que establece un filtro de número pares, de manera que podamos segmentar las posiciones de la partícula entre pares e impares.

La función *ObtenerDatos* permite acceder a los archivos de texto que contienen los resultados de frecuencia en el tiempo simulado. Estos archivos tienen que ser exportado desde DigSILENT-PowerFactory hasta la carpeta en la cual se encuentra el código fuente. La función permite agrupar en una lista todos los datos de frecuencia en el orden de tiempo respectivo de manera que se pueda realizar la operación matemática del índice de frecuencia, el cual se usará en el proceso del método PSO. La **Figura 3.50** muestra el proceso inmerso en la función, con los datos que recibe y la respuesta del índice.

```
# Función para el ingreso de iteraciones
def ingreso_iteraciones(): #Devuelve el número ingresado por teclado, verificando si es entero
    flagIteraciones = False
    while not flagIteraciones:
        num_p = input('Ingrese número de iteraciones que desea realizar: ')
        if num_p.isnumeric():
            flagIteraciones = True
        else:
            print("No ha ingresado un número entero. Intente nuevamente")
    return int(num_p)
```

Figura 3.48. Función para digitar el número de iteraciones

```
# Función para determinar si el número es par
def es_par(numero):
    if numero%2 == 0:
        par = True
    else:
        par = False
    return par
```

Figura 3.49. Función que determina si un número es par

La función *obtenerBL*, mostrada en la **Figura 3.51**, recibe la lista de iteraciones por partícula y entrega una lista con los Best Local de cada partícula ordenado en el proceso. La función *obtenerBG* crea una lista vacía y va añadiendo todos los Best Global de cada iteración en el proceso. La **Figura 3.52** muestra el proceso interno que realiza la función. Podemos observar cómo se itera mediante el lazo *for*, la lista que contiene la configuración de los Best Global en todos los tiempos. El retorno de la función es una lista llamada *ListaG* que contiene los datos de los Best Global segmentado de la siguiente manera: la posición 1 del primer elemento corresponde a la partícula de ese Best Global, la posición 2 del primer elemento corresponde a la configuración de la partícula y la posición 3 del primer elemento corresponde al índice de ese Best Global. Eso mismo ocurre con los demás Best Global anexados a la lista *ListaG*.

Una vez que se agrupe en una lista a los Best Global de cada iteración, se debe de buscar el que tenga el mejor índice para establecerlo como el óptimo. En la **Figura 3.53** se define la función encargada de comparar el índice de todos los Best Global a lo largo del tiempo y, a partir del índice, retornar con el mejor.

```
# Función para leer los archivos .txt y obtener el índice
def obtenerDatos(nombre, dfT, i_p, i_ite):
    cod=str(i_p)+str(i_ite)
    tiempo='tiempo'+cod
    freq='freq'+cod

    df=pd.read_csv(nombre, delimiter = "\t", skiprows=1)
    df.columns=[tiempo, freq]
    df[freq] = df[freq].astype(float)
    #En el caso que el software exporte los resultados con "," se cambia a "."
    #df[freq]=df[freq].str.replace(",",".").astype(float)

    indice=60/df[freq].mean()

    dfT=pd.concat([dfT, df], axis=1)
    return indice, dfT
```

Figura 3.50. Función de lectura de archivos de texto

Debemos definir los eventos de apertura y cierre de interruptores correspondientes. En nuestro caso de estudio, como son 15 barras con generación eólica y fotovoltaica, deben de crearse 30 eventos respectivamente por los dos tipos de generadores. Adicional se debe de crear el evento de apertura y el evento de reconexión de carga.

```
# Función para obtener el Best Local por partícula
def obtenerBL(dc, strParticula, i_ite):
    listaTemp=['P0', 0, 0]

    for i in range(1, i_ite):
        indice=dc[strParticula][i]['indice']

        if(indice>listaTemp[2]):
            listaTemp[0]=strParticula
            listaTemp[1]=i
            listaTemp[2]=indice
    return dc[listaTemp[0]][listaTemp[1]]['config']
```

Figura 3.51. Función que obtiene el Best Local por partícula

```
# Función para obtener el Best Global de todo el enjambre de partículas
def obtenerBG(dc, p, ite):
    listaG=['P0', 0, 0]

    for i in range(1, p+1):
        for j in range(1, ite+1):
            s1="P"+str(i)
            indice=dc[s1][j]['indice']
            if(indice>listaG[2]):
                listaG[0]=s1
                listaG[1]=j
                listaG[2]=indice
    return dc[listaG[0]][listaG[1]]['config']
```

Figura 3.52. Función Best Global de todo el enjambre hasta el momento

La **Figura 3.54** muestra la creación de 4 eventos que son: desconexión de carga, conexión de carga, conexión del generador fotovoltaico PV_Sys1 y conexión del generador eólico StaGen1. En el caso que se desee trabajar con sistemas que posean más barras con generación eólica y fotovoltaica, deberán de digitar los eventos adicionales en las líneas de código. Como tal, los eventos creados solo son eventos vacíos a los cuales aún no se les ha asignado una función y solo se les configuró un nombre para poder identificarlos en la lista de eventos.

```
# Función para obtener el Best Global óptimo entre todas las iteraciones
def obtenerBT(dc,p,ite):
    listaG=['P0',0,0,0]
    for i in range(1,p+1):
        listaTemp=['P0',0,0,0]

        for j in range(1,ite+1):
            s1="P"+str(i)
            indice=dc[s1][j]['indice']

            if(indice>listaTemp[1]):
                listaTemp[0]=s1
                listaTemp[1]=j
                listaTemp[2]=indice
                listaTemp[3]=dc[listaTemp[0]][listaTemp[1]]['config']
            if(indice>listaG[2]):
                listaG[0]=s1
                listaG[1]=j
                listaG[2]=indice
                listaG[3]=dc[listaG[0]][listaG[1]]['config']

    return listaG
```

Figura 3.53. Función Best Global óptimo al final del proceso

```
# Creación de eventos
def shcfolder(p,i):
    shc_folder = app.GetFromStudyCase('IntEvt') # Abrir la carpeta para crear eventos
    cod=str(p)+str(i)
    shc_folder.CreateObject('EvtSwitch', 'Evento_switch_on'+cod) # Evento de switcheo apertura de carga
    shc_folder.CreateObject('EvtSwitch', 'Evento_switch_off'+cod) # Evento de switcheo cierre de carga

    shc_folder.CreateObject('EvtSwitch', '1Evento_switch_Pvsys'+cod) # Evento de switch PV_Sys 1
    shc_folder.CreateObject('EvtSwitch', '1Evento_switch_GenSt'+cod) # Evento de switch StaGen 1
```

Figura 3.54. Función para crear la carpeta de eventos

También se define una función que imprima la información de cada partícula por iteración como se muestra en la **Figura 3.55**. Esto se realiza para que el usuario pueda visualizar el proceso iterativo del algoritmo y tenga una idea de cómo funciona y hacia donde está enfocado.

De la misma forma, se imprime el resultado del mejor Best Global al final del proceso. La **Figura 3.56** indica el proceso de impresión del Best Global óptimo, seguido de la impresión de su índice. También se lee la configuración de este Best Global óptimo y se determina cuáles son los generadores, tanto eólicos como fotovoltaicos, que deben de accionarse.

Mediante lazos *for* se van iterando todos los eventos creados anteriormente y se les va asignando una función en particular según como fueron nombrados, tal como se muestra en la **Figura 3.57**. Se muestra cómo se asigna la función de los eventos de desconexión y conexión de carga para la primera iteración. En este caso en particular, se realizarán estos eventos sobre la carga *Load39*.

```
# Impresión de resultados por iteración
def resultados(dicc,iteraciones):
    print("\nRESULTADOS:\n")

    for j in range(1,iteraciones+1):
        print(f"Iteracion {j}:")
        linea1=""
        linea2=""
        linea3=""
        for i in dicc.keys():

            config=str(dicc[i][j]['config'])
            indice=str(round(dicc[i][j]['indice'],11))

            indice+=" "*(11-len(indice.split('.')[1]))

            linea1=linea1+"Particula "+str(i[1])+" "*37+" "*54 # Número de partícula
            linea2=linea2+"Posicion: "+config+" " # Posición de esa partícula
            linea3=linea3+"Indice: "+indice+" "*27+" "*54 # Índice de esa partícula

        print(linea1)
        print(linea2)
        print(linea3+"\n")
```

Figura 3.55. Función para imprimir los resultados por iteración

En la **Figura 3.58** se muestra cómo hacemos uso de la función “es_par” para verificar que estamos trabajando con una posición par de la partícula en cuestión. Esto se realiza ya que se les asignará una función a los eventos de los generadores fotovoltaicos.

```
# Impresión del Best Global más óptimo entre todo el enjambre de partículas hasta el final de las iteraciones
listaG=obtenerBT(dicc,particulas,iteraciones)
print("El best global del enjambre de partículas es: ",listaG[3])
print("Con un índice de: ",listaG[2])
print("\n")
print("La configuración óptima resulta cuando se conectan los siguientes generadores:\n")
listaFotovoltaicos=[]
listaEolicos=[]
print(len(listaG[3]))
for i in range(0,len(listaG[3])//2):

    if listaG[3][i*2]==1:
        listaFotovoltaicos.append(((i*2)//2)+1)

    if listaG[3][i*2+1]==1:
        listaEolicos.append(((i*2)//2)+1)

print("Generadores fotovoltaicos: ",listaFotovoltaicos) # Imprime los generadores fotovoltaicos que se encienden
print("Generadores eólicos: ",listaEolicos)           # Imprime los generadores eólicos que se encienden
print("\n")
```

Figura 3.56. Impresión del Best Global óptimo al final del proceso

```
# Se asigna la función de los eventos creados
dfT=pd.DataFrame()
dc = {}
for i_ite in range(1,iteraciones+1):
    for i_p in range(1,particulas+1):
        shc_folder=shcfolder(p=i_p,i=i_ite)
        eventos = shc_folder.GetContents() # Obtener la lista de eventos

        strParticula = "P"+str(i_p)

        # Se asignan lo eventos de la primera iteración
        if i_ite==1:
            dc[strParticula]={}
            dc[strParticula][i_ite]={}
            # modificar parámetros de evento de desconexión de carga
            eventos[0].time = 0.25 # tiempo de apertura a t = 0.25 s
            eventos[0].i_switch = 0 # apertura de sw de carga
            eventos[0].p_target = load_dict['Load39'] # Se llama a carga Load39
            # modificar parámetros de evento de conexión de carga
            eventos[1].time = 0.5 # Tiempo de cierre a t = 0.50 s
            eventos[1].i_switch = 1 # cierre de sw de carga
            eventos[1].p_target = load_dict['Load39'] # Se llama a carga Load39
```

Figura 3.57. Asignación de la función de eventos - Parte 1

Para los demás elementos que corresponden a las posiciones impares de la partícula, se le fueron asignados sus trabajos sobre los generadores eólicos del sistema. Las dos últimas líneas de código mostradas en la **Figura 3.59** corresponden a la identificación de la posición y velocidad de la partícula con la cual se está trabajando dentro de la primera iteración.

El proceso matemático inmerso en el algoritmo PSO se indica en las líneas de código de la **Figura 3.60**. Aquí se opera bajo los datos de la primera iteración (aleatorios) y se realizan los cálculos para obtener la posición de la partícula en la siguiente iteración. Podemos ver cómo hacemos uso de la librería *Random* para asignar valores aleatorios a $r1$ y $r2$, y con la librería *Numpy* para operar con los arreglos de partículas.

Una vez realizados los cálculos, se obtiene la velocidad y posición de la partícula en la siguiente iteración. Como estamos trabajando con número binarios, la **Figura 3.61** muestra la conversión de los elementos de la velocidad y posición convertidos a valores de 0 y 1.

En la parte del código mostrada en la **Figura 3.62**, se crean nuevamente los eventos de conexión y desconexión de carga que servirán para todas las iteraciones después de la primera. A partir de la configuración de la posición de la partícula en la siguiente iteración, se asigna la función de los eventos para la generación fotovoltaica y eólica, tal como se muestra en la **Figura 3.63**.

```
nPv=1
nGen=1
listaTempX = []
listaTempV = []
for i in range(2,2*n_barras+2):
    if es_par(i):
        var1_pv = rd.randrange(0, 2)
        listaTempX.append(var1_pv)
        listaTempV.append(0)
        eventos[i].time = 0
        eventos[i].i_switch = var1_pv
        eventos[i].p_target = pv_dict['PV_Sys'+str(nPv)]
        nPv+=1
# tiempo de cierre del sw t = 0 s
# pv 1 conetar y 0 apagar
```

Figura 3.58. Asignación de la función de eventos - Parte 2

```

else:
    var1_gen = rd.randrange(0,2)

    listaTempX.append(var1_gen)
    listaTempV.append(0)
    eventos[i].time = 0 # tiempo de cierre del sw t = 0 s
    eventos[i].i_switch = var1_gen # eólico 1 conetar y 0 apagar
    eventos[i].p_target = genst_dict['StaGen'+str(nGen)]
    nGen+=1
dc[strParticula][i_ite]['config']=listaTempX
dc[strParticula][i_ite]['vel']=listaTempV

```

Figura 3.59. Asignación de la función de eventos - Parte 3

```

# Se asignan los eventos luego de la primera iteración. Inicio del método PSO
else:
    dc[strParticula][i_ite]={}
    # Parámetros del método PSO
    w=0.4 # Factor de inercia
    vant=dc[strParticula][i_ite-1]['vel'] # Velocidad anterior de la partícula
    xant=dc[strParticula][i_ite-1]['config'] # Posición anterior de la partícula
    c1=c2=2 # Constantes de atracción
    r1=rd.randrange(1,2*n_barras+1) # Número aleatorio
    r2=rd.randrange(1,2*n_barras+1) # Número aleatorio
    bestL=obtenerBL(dc, strParticula, i_ite) # Best Local de la partícula hasta el momento
    # Cálculo de la velocidad
    b1= [i * w for i in vant]
    con1=c1*r1
    l1=list(np.array(bestL) - np.array(xant))
    b2 = [i * con1 for i in l1]
    con2=c2*r2
    l2=list(np.array(bestG) - np.array(xant))
    b3 = [i * con2 for i in l2]

```

Figura 3.60. Método metaheurístico PSO en Python

```

velocidad = list(np.array(b1)+np.array(b2)+np.array(b3)) # Se obtiene la nueva Velocidad
for i in range(len(velocidad)):
    if velocidad[i]>0:
        velocidad[i]=1
    else:
        velocidad[i]=0
posicion=list(np.array(xant) + np.array(velocidad)) # Se obtiene la nueva Posición
for i in range(len(posicion)):
    posicion[i] = int(posicion[i])
    if posicion[i]>0:
        posicion[i]=1
    else:
        posicion[i]=0

```

Figura 3.61. Cálculo de la nueva velocidad y posición de la partícula

La simulación se realiza cada vez que se cambie la configuración de la posición de la partícula. En la **Figura 3.64** se muestra el proceso de simulado mediante la codificación en Python, en donde asignamos un tiempo de 20 segundos ejecutado para todos los casos de estudio de cada partícula y en cada iteración.

Por último, la **Figura 3.65** muestra las líneas de código que permiten la exportación de resultados en formato “.txt” a la carpeta la cual se encuentra el código fuente. El archivo es nombrado *partícula* seguido del número de partícula a la cual corresponde y del número de iteración, como, por ejemplo, *partícula24.txt* que corresponde a los datos de frecuencia de la partícula 2 en la iteración 4.

```

dc[strParticula][i_ite]['config'] = posicion
dc[strParticula][i_ite]['vel'] = velocidad
# Desconexion de carga
eventos[0].time = 0.25 # tiempo de apertura a t = 0.25 s
eventos[0].i_switch = 0 # apertura de sw de carga
eventos[0].p_target = load_dict['Load39'] # Se llama a carga Load39
# Conexion de carga
eventos[1].time = 0.5 # tiempo de cierre a t = 0.25 s
eventos[1].i_switch = 1 # cierre de sw de carga
eventos[1].p_target = load_dict['Load39'] # Se llama a carga Load39

```

Figura 3.62. Asignación de la función de eventos, Parte 4

```

nPv=1
nGen=1
for i in range(2,2*n_barras+2):

    if es_par(i):
        eventos[i].time = 0 # tiempo de cierre del sw t = 0 s
        eventos[i].i_switch = posicion[i-2] # pv 1 conetar y 0 apagar
        eventos[i].p_target = pv_dict['PV_Sys'+str(nPv)]
        nPv+=1

    else:
        eventos[i].time = 0 # tiempo de cierre del sw t = 0 s
        eventos[i].i_switch = posicion[i-2] # eólico 1 conetar y 0 apagar
        eventos[i].p_target = genst_dict['StaGen'+str(nGen)]
        nGen+=1

```

Figura 3.63. Asignación de la función de eventos, Parte 5

Las últimas líneas del código fuente, mostradas en la **Figura 3.66**, sirven para establecer el tiempo que existe entre los cambios de eventos correspondientes a la configuración de cada partícula. Esto se realiza mediante la librería *Time* que contiene las funciones relacionadas con la medición de tiempos de procesos, en este caso de iteraciones. Para conocer más acerca de los comandos utilizados en el código fuente se guía al lector al **Apéndice C** de este documento.

```
# Ejecución y simulación del caso de estudio
# Reset de los cálculos
#app.ResetCalculation()

# Leer y modificar archivos de resultado
eLmres = app.GetFromStudyCase('All calculations.ElMRes') # Llamar al archivo de resultados

# Condiciones iniciales
ini = app.GetFromStudyCase('ComInc') # Cálculo de las condiciones iniciales

ini.Execute() # Ejecución del caso de estudio

# Simulación Dinámica
sm = app.GetFromStudyCase('ComSim') # Activación de la simulación dinámica

sm.tstop = 20 # tiempo de simulación t = 20s

sm.Execute() # Ejecución del caso de estudio
```

Figura 3.64. Ejecuta la simulación de los eventos

```
# Exporta los resultados desde DigSILENT-PowerFactory en archivo .txt hacia la carpeta del ordenador
comres = app.GetFromStudyCase('ComRes')
comres.iopt_csel = 0
comres.iopt_tsel = 1
comres.iopt_locn = 2
comres.ciopt_head = 1
comres.pResult = eLmres # desde donde se va a exportar el archivo de resultados
comres.iopt_exp = 4
comres.f_name = r'C:\Users\CltControl\Desktop\PI Gallegos y Rivera\particula' + str(i_p)+str(i_ite) + '.txt'
comres.Execute()
```

Figura 3.65. Exportación de resultados en archivo de texto

```
time.sleep(1) # Tiempo de cambio entre configuraciones
eLmres.Clear()
nombreA='particula'+str(i_p)+str(i_ite)+".txt"

indice,dfT=obtenerDatos(nombreA,dfT,i_p,i_ite) #Se obtiene apartir del txt de resultados el indice y el dfTotal
dc[strParticula][i_ite]['indice']=indice
#shc_folder.Delete()
bestG=obtenerBG(dc,particulas,i_ite)
```

Figura 3.66. Tiempo de cambio entre partículas del enjambre resultados

El estudio de estabilidad de pequeña señal a realizar en el sistema implica que la frecuencia difiera en función de las barras en donde se localiza la generación eólica y fotovoltaica. La cantidad de arreglos de posibles posicionamientos es grande, ya que depende del número de generadores a integrar y barras que presente el sistema. La solución a este proceso es automatizar el desarrollo del estudio mediante el control por código fuente a través de Python sobre DigSILENT PowerFactory. El algoritmo de búsqueda radica en el método metaheurístico denominado Particle Swarm Optimization (PSO).

CAPÍTULO 4

4.1 Resultados y análisis

El Capítulo 4 presenta los resultados obtenidos del estudio de estabilidad de pequeña señal realizado en PowerFactory (modo esclavo), a partir de la automatización del proceso en la búsqueda de la solución, cabe señalar que esto es desarrollado previamente en Python (modo maestro). Se analizará el comportamiento de la señal de frecuencia frente a un evento de desconexión y conexión de carga, para posteriormente establecer el afecto en la estabilidad en el sistema en función a la ubicación de la generación eólica y fotovoltaica.

4.2 Ubicación óptima para la generación eólica y fotovoltaica

La **Figura 4.1** muestra la integración de los generadores eólicos y fotovoltaicos en las barras (locación) de la red, considerada la ubicación óptima de tal manera que satisfaga el índice de estabilidad para pequeñas perturbaciones establecido en el Capítulo 3. El sistema modificado IEEE de 39 barras, cuenta con alta penetración de energía renovables, en este se detalla que en total existen 15 barras, cada una cuenta con la posibilidad de conexión de un generador eólico y otro fotovoltaico, siendo así en total 15 generadores eólicos y 15 fotovoltaicos con la posibilidad de integrarse al sistema. El algoritmo de búsqueda implementado en Python que sigue el método metaheurístico PSO, establece como mejor opción al arreglo de generadores eólicos y fotovoltaicos en distintas locaciones del sistema que mejor aproxime al valor ideal de índice de estabilidad, este valor es 1. Es importante mencionar que, como restricciones o limitantes se considera la afluencia del recurso, siendo la cantidad de viento entorno a generación eólica e incidencia de radiación solar correspondiente a la generación fotovoltaica, se establece que desde la barra 16 hasta la barra 49, los niveles de viento y radiación solar no son adecuados para la construcción de proyectos de generación eléctrica a partir de fuentes eólicas o fotovoltaicas.

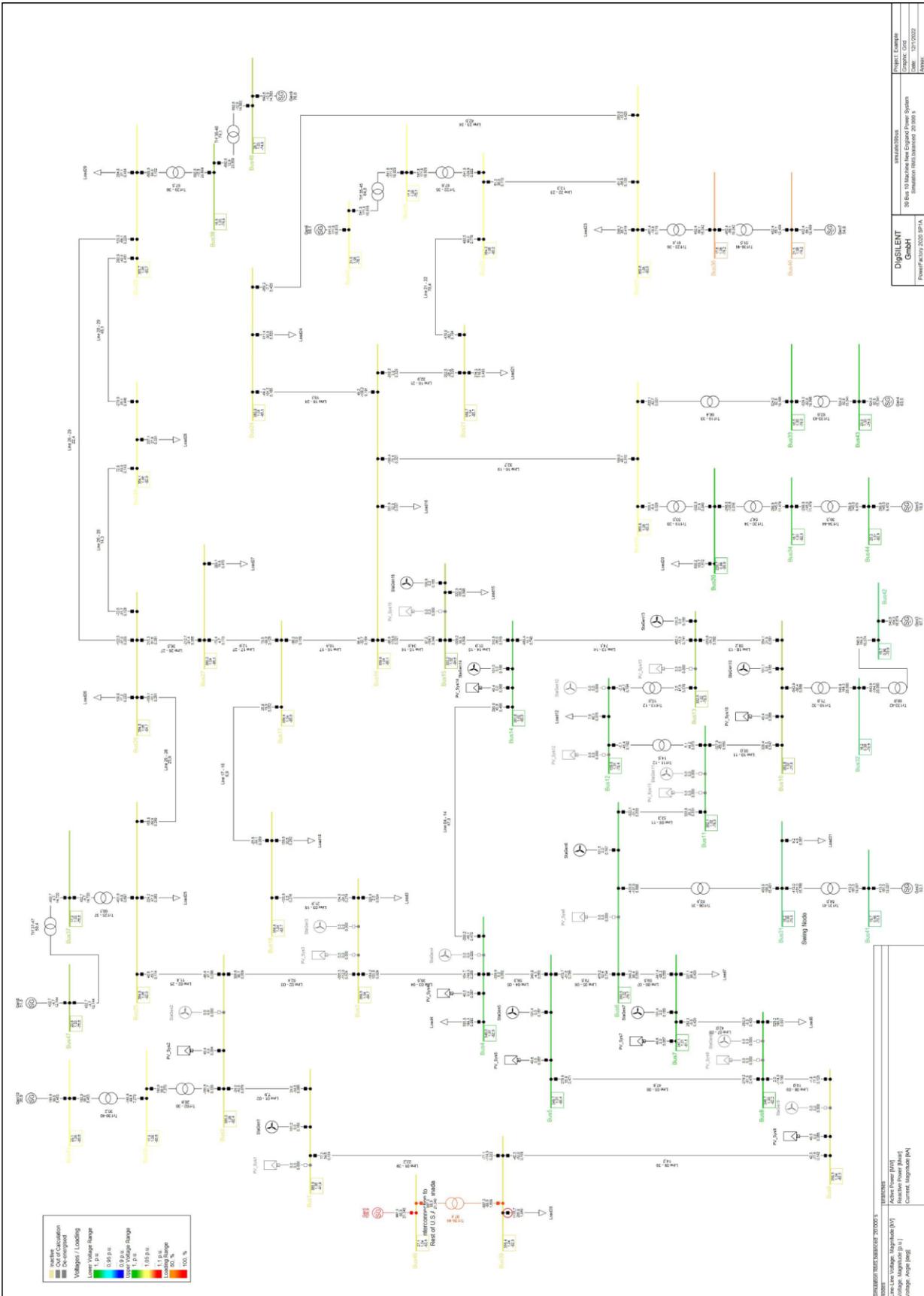


Figura 4.1. Esquema de ubicación óptima de generación eólica y fotovoltaica

Los generadores eólicos y fotovoltaicos que se integran a la red, considerando la mejor ubicación en función del estudio de estabilidad de pequeña señal, específicamente de la respuesta de frecuencia eléctrica se muestra a continuación.

La **Tabla 4.1** muestra a los generadores tanto eólicos como fotovoltaicos conectados al sistema modificado IEEE de 39 barras. La cantidad de generadores eólicos integrados, en comparación al total de posiciones (barras) en donde se podría ubicar generación eólica representa el 53,3%, mientras que, para el caso de generación fotovoltaica, los generadores conectados representan el 46.6% de las posibles ubicaciones de generación fotovoltaica en el sistema.

El total de unidades de generación fotovoltaicas son 7, tales que proporcionan 40 MW de potencia activa cada una, es decir que el sistema eléctrico de potencia recibe un suministro total de 280 MW de energía eléctrica, con la finalidad de satisfacer la demanda industrial, comercial, residencial y sus estratos de usuarios derivados. El total de unidades de generación eólica son 8, estas contribuyen con 100 MW de potencia activa cada una, es decir que el sistema eléctrico de potencia recibe un suministro total de 800 MW de energía eléctrica, con la finalidad de satisfacer la demanda requerida por la red estudiada.

El suministro total que recibe el sistema eléctrico de potencia modificado de la IEEE de 39 barras corresponde a la suma del total proporcionado por la generación fotovoltaica y la generación eólica, es lo mismo decir 1080 MW, este valor representa calidad de energía, confiabilidad y seguridad en el sistema de potencia, ya que se obtiene a partir del mejor caso de estabilidad de pequeña señal frente a una perturbación.

Tabla 4.1. Generación Eólica y Fotovoltaica Activa en el Sistema

Generación Eólica	StaGen1	StaGen5	StaGen6	StaGen7	StaGen10	StaGen13	Stagen14	StaGen15
Generación fotovoltaica	PV_Sys2	PV_Sys4	PV_Sys5	PV_Sys7	PV_Sys9	PV_Sys10	PV_Sys14	

La energía suministrada por las fuentes renovables no convencionales que son referentes al tema de este proyecto, son limpias y son sinónimo de la disminución de la huella de carbono en el planeta, así como un incentivo a las nuevas generaciones al desarrollo de tecnologías amigables con el ambiente que satisfagan el incremento anual de la demanda energética.

La generación suministrada por los generadores eólicos y fotovoltaicos satisface el 17.7 % de la demanda requerida por el sistema, ya que se conoce que la potencia activa de la carga total es 6097.1 MW, misma que se describe con detalle en la **Tabla 3.3**.

Suponiendo el caso de sustituir la generación síncrona proporcionada por los generadores: Gen1, Gen2, ..., Gen10; la generación eólica y fotovoltaica representaría un valor aproximado del 20 %, conociendo que previamente se determinó que la potencia total generada por los generadores sincrónicos es de 5366 MW, esto puede ser visto en detalle en la **Tabla 3.2**.

4.3 Resultados proporcionados por Python

Mediante el código fuente se realiza la búsqueda de la conexión de generación eólica y fotovoltaica que presenta el escenario óptimo de estabilidad frente a una perturbación de pequeña señal en el sistema modificado IEEE de 39 barras. El proceso iterativo es realizado mediante el método metaheurístico PSO aplicado a arreglos. La interfaz del código fuente permite al usuario ingresar la cantidad de iteraciones que desea en el proceso y la cantidad de partículas a analizar por iteración. Una vez finalizado el proceso, se imprime el resultado de la conexión óptima representado por el Best Global del enjambre de partículas.

Cada partícula comienza con una posición aleatoria en la primera iteración y después se da inicio el proceso matemático del método PSO para obtener la configuración de la nueva posición en las siguientes iteraciones. Así mismo, cada configuración iterada va a traer consigo un nuevo índice obtenido de los resultados de frecuencia exportados de DigSILENT-PowerFactory.

El objetivo fundamental de una partícula del enjambre es buscar la mejor posición entre todas las demás y no moverse a menos que alguna otra partícula la alcance o supere. Esto es lo que sucede con la partícula 1, en donde el método PSO muestra una inclinación hacia ella dado que tiene el mejor índice desde el inicio. Las demás partículas del enjambre presentan cambios debido a la búsqueda constante en la que se encuentran para obtener una mejor configuración de posición que refleje un índice mayor al de la partícula 1. En los cambios se ve reflejado el acercamiento que tienen unas a las otras con el fin de superarse entre ellas y poder alcanzar a la partícula 1. Mientras más iteraciones se planteen en el proceso, mayor será el acercamiento de estas partículas hacia la partícula 1.

A continuación, se presentarán los resultados obtenidos de la ventana de comandos de Python para un escenario de 6 partículas y un proceso de 10 iteraciones.

La **Figura 4.2** muestra el resultado final del proceso en donde se imprime el Best Global del enjambre de partículas con su índice, y la representación de dicha configuración con respecto a la conexión de los generadores eólicos y fotovoltaicos. El escenario planteado indica que a cuando se instalan los generadores fotovoltaicos [2, 4, 5, 7, 9, 10, 14] y los generadores eólicos [1, 5, 6, 7, 10, 13, 14, 15], se obtiene un mayor índice con respecto a todas las configuraciones del enjambre de partículas. Esto se traduce en el mejor esquema de conexiones del sistema modificado IEEE de 39 barras considerando la estabilidad del parámetro de frecuencia frente a una perturbación de pequeña señal.

```
El Best Global del enjambre de partículas es: [0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1]
Con un índice de: 0.9987836486586442

La configuración más óptima sucede cuando se conectan los siguientes generadores:

Generadores fotovoltaicos: [2, 4, 5, 7, 9, 10, 14]
Generadores eólicos: [1, 5, 6, 7, 10, 13, 14, 15]
```

Figura 4.2. Ubicación óptima de generación eólica y fotovoltaica en Python

4.4 Respuestas del sistema frente a una perturbación de pequeña señal

La red eléctrica frente a perturbaciones pequeñas tiende a modificar la inercia que proporcionan los generadores sincrónicos debido también a la integración de generación eólica y fotovoltaica, este fenómeno se manifiesta al presentarse un desbalance entre la generación y la carga. La seguridad en la generación de energía eléctrica depende de la rapidez de actuación de los dispositivos frente a un evento de falla y la recuperación del estado de equilibrio después de haberse sometido a la misma.

4.4.1 Respuestas del generador Gen1 frente a evento en la carga Load39

Luego de realizar la simulación con las conexiones óptimas del sistema, mediante DigSILENT-PowerFactory podemos observar los parámetros de frecuencia, potencia y voltaje del generador del Generador 1 (Gen1), esto cuando el sistema se ve afectado por una perturbación en la carga 39 (Load39), cabe mencionar que: Gen1 y Load39 representan las etiquetas de los elementos en el sistema presentado en el entorno de DigSILENT-PowerFactory, a la vez son los nombres como los reconoce el código fuente desarrollado en Python.

La **Figura 4.3** muestra la respuesta de frecuencia eléctrica en hertz en función del tiempo de simulación en segundos, como resultado se obtiene la estabilización o la recuperación del sincronismo en la maquina síncrona (Gen1) aproximadamente a los 12.101666 segundos después de haber ocurrido la desconexión de la carga (Load39) en 0.25 segundos, para el instante de tiempo mencionado le corresponde a la frecuencia un valor de 60.05097 hertz.

Los valores de tiempo de estabilización y frecuencia del generador síncrono que le corresponde al mismo, se los obtiene mediante el análisis gráfico, colocando el cursor sobre donde se amortigüe completamente la señal, pero como segunda opción más exacta al parecer es exportar los datos de frecuencia eléctrica y tiempo de simulación en un archivo de extensión “.csv” para observarlos en más detalle y permitirse aplicar funciones que proporciona Excel en el tratamiento de datos.

$$Error = \frac{|Valor\ experimental - Valor\ teórico|}{Valor\ teórico} \quad (4.1)$$

$$Error = \frac{|60.05097 - 60|}{60} * 100\% = 0.08\%$$

La **ecuación 4.1** permite estimar el error de la variable de frecuencia eléctrica, considerando que se toma la decisión de escoger un valor de frecuencia que presente 3 decimales que permanezcan constantes. El valor de 0.08% representa un valor muy bueno de estabilización considerando que para brindar mayor confiabilidad y seguridad al sistema se estableció el control previo por PSS y AVR en cada uno de los generadores síncronos del sistema.

El valor máximo de frecuencia eléctrica de un evento de desconexión en $t_{sim} = 0.25\ seg$ y conexión en $t_{sim} = 0.5\ seg$ de la carga (Load39) corresponde a $f = 60.856627$ para un $t_{sim} = 0.421667\ seg$.

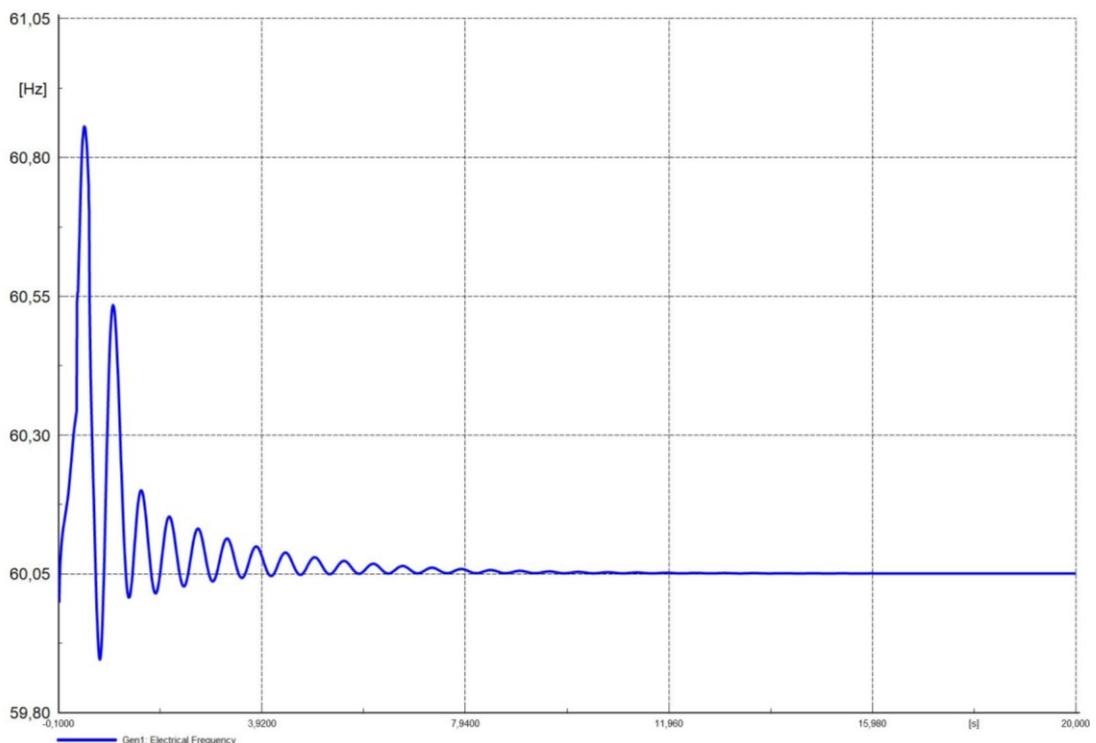


Figura 4.3. Frecuencia eléctrica [Hz] del Gen1 frente a evento en Load39

La **Figura 4.4** muestra la respuesta de potencia activa total del sistema en función del tiempo de simulación, como resultado se observa que la estabilización de la máquina síncrona (Gen1) está dada aproximadamente a los 11.531666 segundos después de haber ocurrido la desconexión de la carga (Load39) en 0.25 segundos, para el instante de tiempo mencionado le corresponde un valor de potencia activa de 997.969645 MW.

La señal de potencia activa que se obtiene permite verificar el nivel de recuperación del sistema a un punto de operación estable, de tal manera que el estudio garantice niveles aceptables de seguridad y confiabilidad. La **ecuación 4.1** permite estimar el error de la variable de potencia activa. El valor de 0.2% representa la disminución de la potencia activa de generación del generador (Gen1), dicho de otro modo, la recuperación del sistema frente a un disturbio es considerable.

$$Error = \frac{|997.969645 - 1000|}{1000} * 100\% = 0.2\%$$

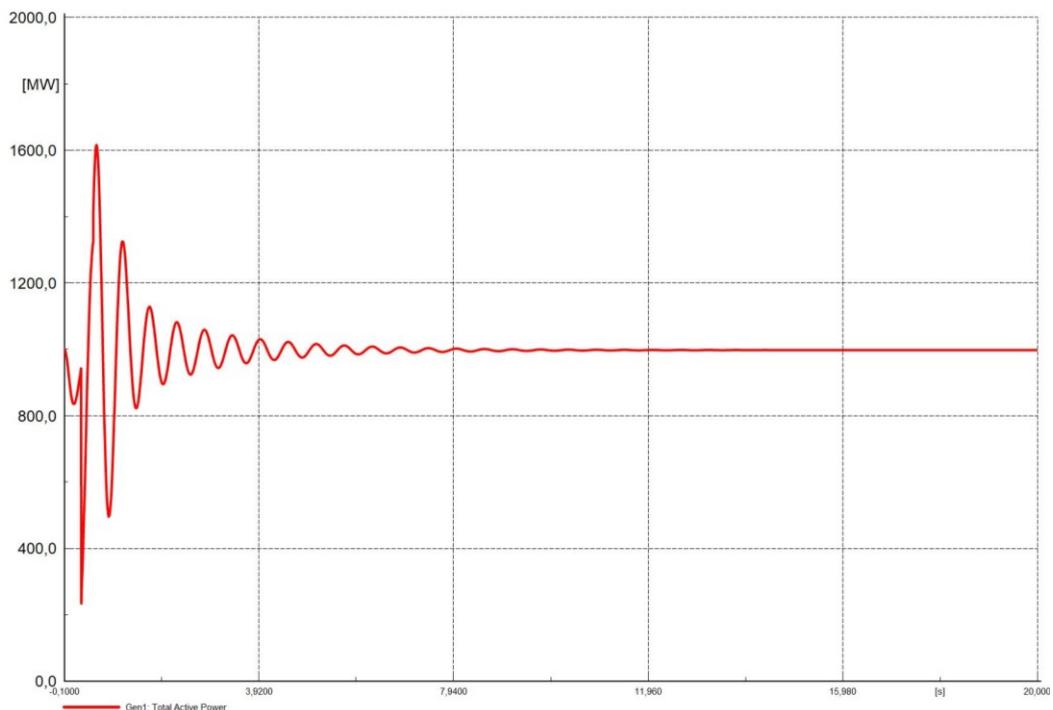


Figura 4.4. Potencia activa total [MW] del Gen1 frente a evento en Load39

El valor máximo de potencia activa cuando se presenta un evento de desconexión en $t_{sim} = 0.25 \text{ seg}$ y conexión en $t_{sim} = 0.5 \text{ seg}$ de la carga (Load39) corresponde a $P = 1615.547302$ para un $t_{sim} = 0.571667 \text{ seg}$.

La **Figura 4.5** muestra la respuesta del voltaje terminal en función del tiempo de simulación, como resultado se observa que la estabilización de la máquina síncrona (Gen1) se aproxima a los 7.591667 segundos después de haber ocurrido la desconexión de la carga (Load39) en 0.25 segundos, para el instante de tiempo mencionado le corresponde un valor de tensión terminal de 1.041903 p.u.

La respuesta de voltaje terminal se amortigua mucho más rápido que las repuestas de frecuencia eléctrica y potencia activa, esto se debe en gran parte al AVR implementado en los generadores sincrónicos del sistema. El proceso realizado anteriormente para identificar el nivel de seguridad sobre el cual actúa el sistema, se calcula el error de voltaje terminal.

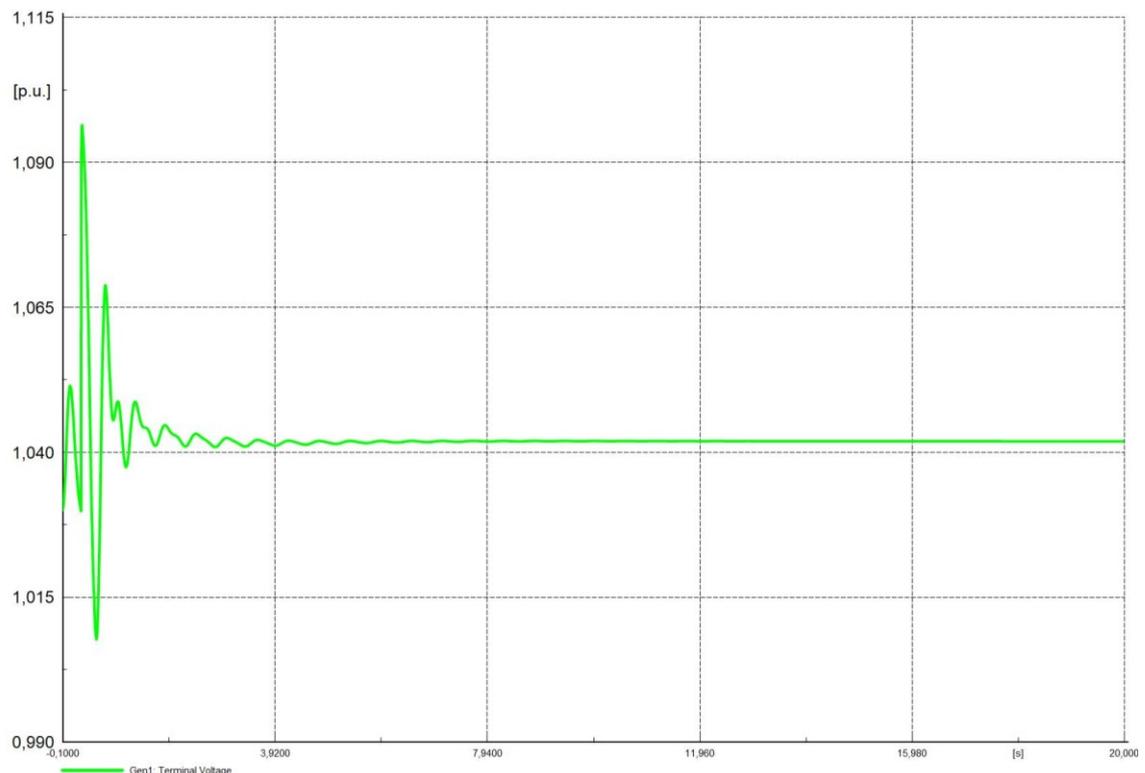


Figura 4.5. Voltaje terminal [p.u] del Gen1 frente a evento en Load39

$$Error = \frac{|1.041903 - 1|}{1} * 100\% = 4\%$$

El 4% error representa la caída del voltaje terminal respecto al valor nominal de voltaje terminal. El valor máximo de voltaje terminal cuando se presenta un evento de desconexión en $t_{sim} = 0.25 \text{ seg}$ y conexión en $t_{sim} = 0.5 \text{ seg}$ de la carga (Load39) corresponde a $V_{term} = 1.096412 \text{ p.u}$ para un $t_{sim} = 0.271667 \text{ seg}$.

4.4.2 Respuestas de generadores síncronos a la perturbación en Load39

Los generadores síncronos Gen1, Gen2, ..., Gen10 se ven afectados por el evento de desconexión y conexión de la carga (Load39) considerando la distancia que los separa entre sí. La **Figura 4.6** muestra el incremento en la magnitud del valor máximo de frecuencia eléctrica y la frecuencia de estabilización respecto al tiempo de cada uno de los generadores síncronos del sistema.

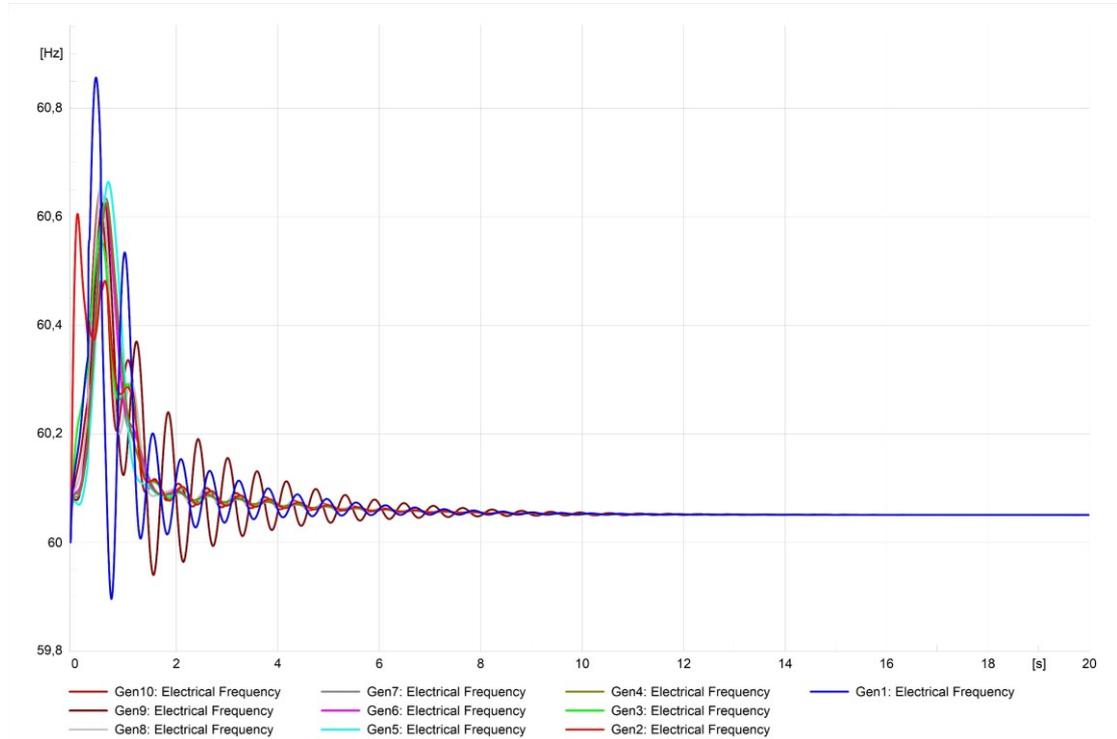


Figura 4.6. Frecuencia [Hz] de generadores síncronos frente a evento en Load39

El escenario establece valores distintos de estabilización para cada generador en el sistema, por lo que se permite contrastar para definir una relación entre la distancia de aplicación de perturbación con respecto generación síncrona que está integrada a la red.

En la **Tabla 4.2** se presenta el valor pico que alcanzan los generadores síncronos que se encuentran en el sistema modificado IEEE de 39 barras implementado en DigSILENT-Powerfactory, así mismo el tiempo de simulación en donde ocurre este suceso. El valor máximo de frecuencia proporciona el nivel de afección de la perturbación sobre el sincronismo de los generadores en la red.

En la **Tabla 4.3** se define la frecuencia de estabilización, considerando para este estudio, y con el objetivo de obtener un error de medición de frecuencia bajo, estableciendo un valor con 3 decimales no variables.

La recuperación del punto de operación de los generadores síncronos en un intervalo de 10 a 20 segundos es representativa de la rapidez, confiabilidad y seguridad al sistema después de haberse visto afectada por una perturbación.

Tabla 4.2. Frecuencia Máxima Frente a Evento en Load39

	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6	Gen7	Gen8	Gen9	Gen10
t_{sim} [s]	0.42166	0.06	0.51166	0.62166	0.66166	0.61166	0.62166	0.51166	0.57166	0.51166
f_{max} [Hz]	60.8566	60.6052	60.5834	60.6256	60.6641	60.6240	60.6336	60.6679	60.6261	60.6553

Tabla 4.3. Frecuencia de Estabilización Frente a Evento en Load39

	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6	Gen7	Gen8	Gen9	Gen10
t_{sim} [s]	11.5416	12.6916	12.7016	12.5616	12.0916	12.6316	12.6316	12.6016	12.4316	12.6016
f_{sta} [Hz]	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509	60.0510	60.0509

La respuesta que se muestra en la **Figura 4.7** representa la potencia activa total que suministra cada generador al sistema, misma que no se ve afectada considerablemente ya que al momento de la desconexión el sistema se libera en un porcentaje mínimo de la potencia total del sistema, el pico que se observa en las gráficas es referente el arranque de cada generador, es por lo que alcanza picos de sobrecarga antes que se manifieste el evento de desconexión y conexión de la carga (Load39).

El comportamiento de las señales de voltajes terminales de cada generador se muestra en la **Figura 4.8**, donde las tensiones pico no se presentan necesariamente luego de la perturbación, como es el caso del *Gen3* y *Gen7*. En este los valores máximos de voltaje en *p.u* ocurren en $t_{sim} = 0.06 \text{ seg}$ y $t_{sim} = 0.04 \text{ seg}$ respectivamente. Por otro lado, las señales muestran un comportamiento similar ya que se estabilizan, aproximadamente, luego de los **10 seg** de simulación.

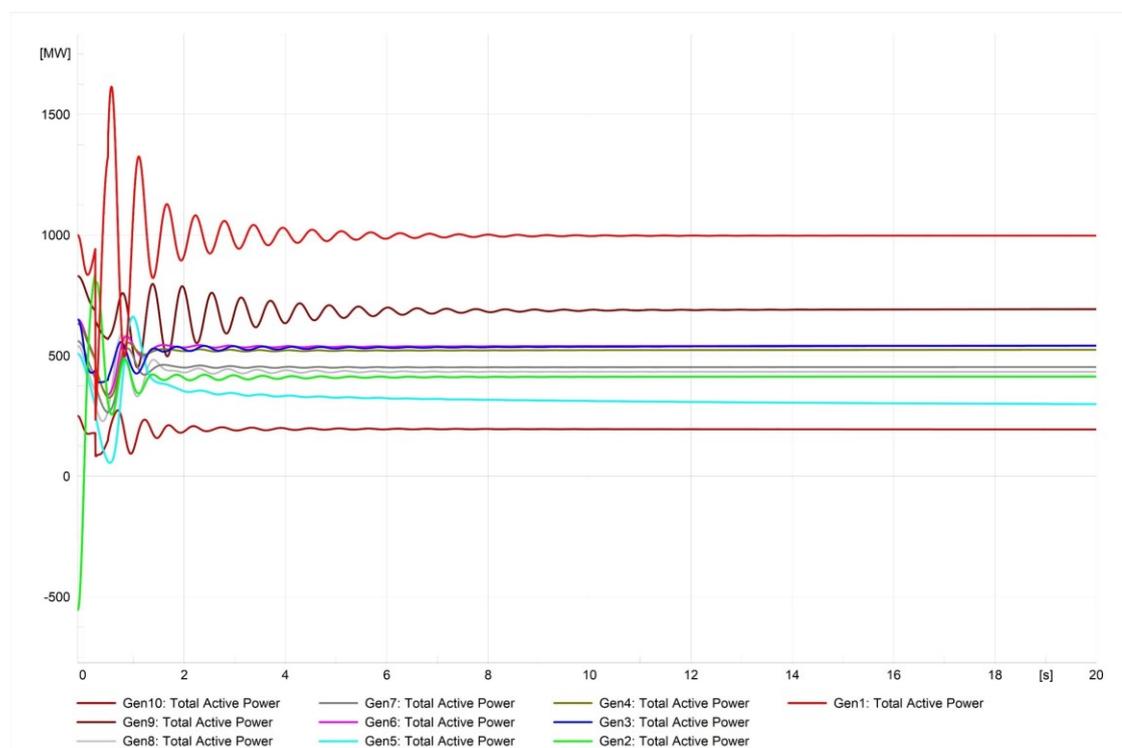


Figura 4.7. P. Activa [MW] de generadores síncronos frente a evento en Load39

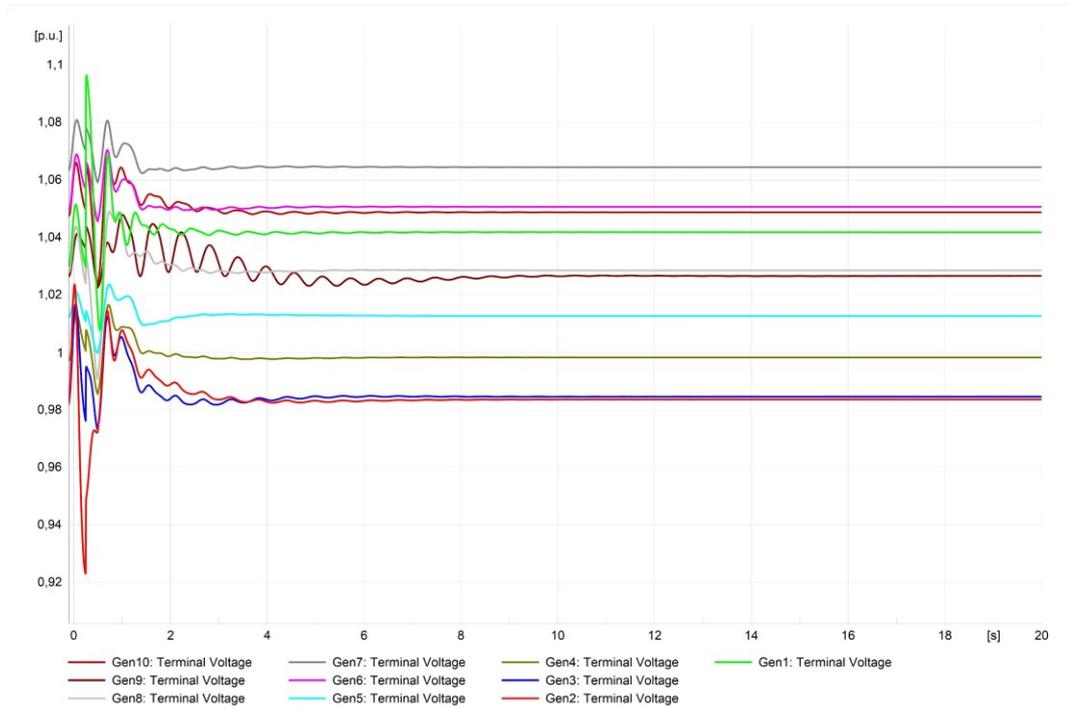


Figura 4.8. V_{term} [p.u.] de generadores síncronos frente a evento en Load39

Las señales de salida de potencia activa modelan el estado dinámico del sistema en presencia de la perturbación en la carga *Load39* del sistema modificado IEEE de 39 barras, en el que se aprecia su rapidez de actuación y búsqueda del equilibrio del sistema posteriormente. La respuesta permite contrastar tiempos y magnitudes, siendo estos indicadores cuantificables de la afcción de la perturbación en el sincronismo del sistema.

4.4.3 Respuestas de generadores síncronos a la perturbación en Load23

La localización de la falla es un factor que afecta directamente a la estabilidad de la unidad de generación. Para determinar el impacto de la localización de la falla en el comportamiento de los generadores síncronos, se ha programado el evento de desconexión y conexión en la carga *Load23* situado al otro extremo del sistema. En la **Figura 4.9** se muestra los cambios de frecuencia que experimenta cada generador síncrono cuando se programa la perturbación en la carga *Load23*.

Se puede observar cambios en los tiempos de estabilización de frecuencia causados a partir de la nueva localización de falla y los nuevos valores máximos de frecuencia que experimenta cada unidad generadora en el tiempo simulado.

En la **Tabla 4.4** se muestran los nuevos valores picos de frecuencia de los generadores síncronos, que se suscitan luego de presentarse la perturbación de desconexión de carga en $t_{sim} = 0.25 \text{ seg}$, los cuales varían en distintos tiempos.

Los tiempos de estabilización y magnitud de frecuencias varían según las características y localización de los generadores síncronos del sistema, tal como se muestra en la **Tabla 4.5**.

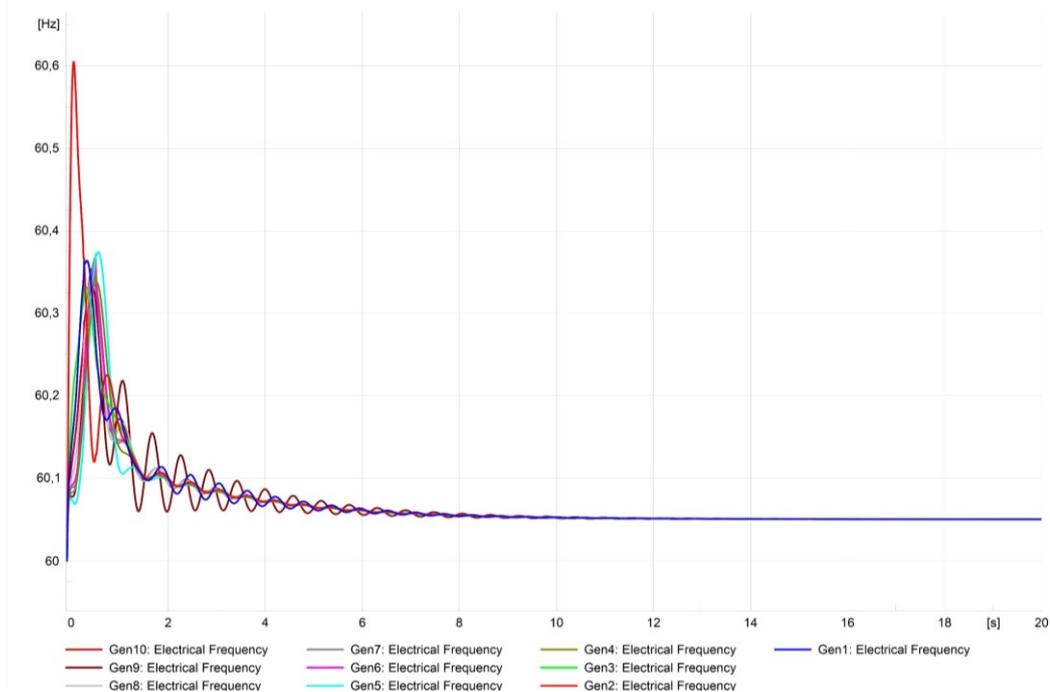


Figura 4.9. Frecuencia [Hz] de generadores síncronos frente a evento en Load23

Tabla 4.4. Frecuencia Máxima Frente a Evento en Load23

	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6	Gen7	Gen8	Gen9	Gen10
t_{sim} [s]	0.3316	0.06	0.2716	0.5116	0.5616	0.5	0.5	0.4516	0.4416	0.4216
f_{max} [Hz]	60.3640	60.6052	60.3334	60.3476	60.3745	60.3552	60.3676	60.3497	60.3549	60.3349

Para las potencias activas totales de cada generador, se observa en la **Figura 4.10** que al principio se manifiestan oscilaciones y luego se estabilizan en un valor constante según su valor nominal de operación. La perturbación genera que dichas oscilaciones se magnifiquen por el efecto de una sobre generación al no presentarse la misma carga inicial del sistema, sin embargo, ya existían valores picos mucho mayores debido al arranque de los generadores.

La **Figura 4.11** muestra el comportamiento transitorio de cada generador síncrono, apreciando el tiempo de estabilización de cada señal.

Tabla 4.5. Frecuencia de Estabilización Frente a Evento en Load23

	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6	Gen7	Gen8	Gen9	Gen10
t_{sim} [s]	12.5616	13.0816	13.0916	12.9716	12.6116	13.0316	13.0316	13.0216	13.4316	13.0116
f_{sta} [Hz]	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509	60.0509

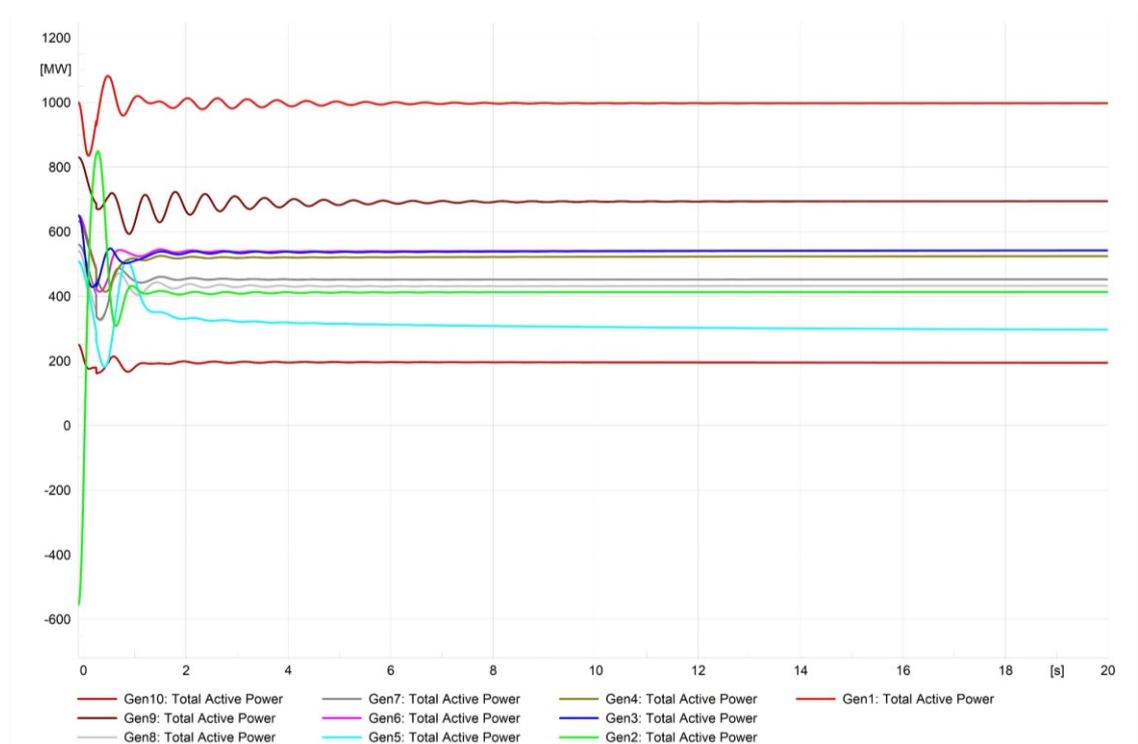


Figura 4.10. P. Activa [MW] de generadores síncronos frente a evento en Load23

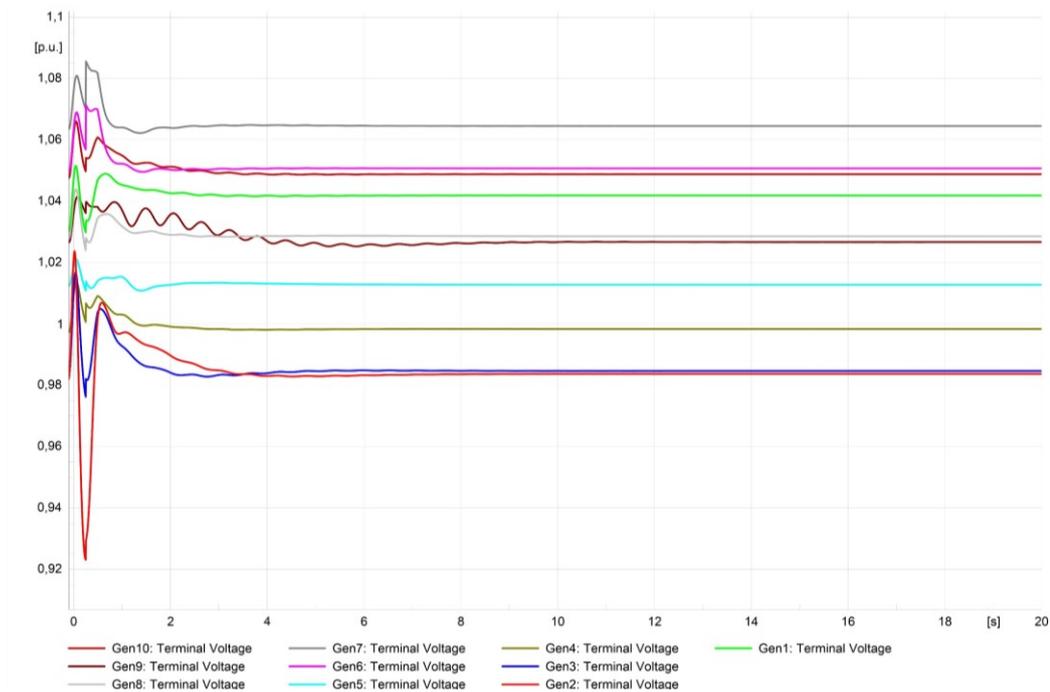


Figura 4.11. V_{term} [p.u] de generadores síncronos frente a evento en Load23

Similar a la potencia activa total, el voltaje terminal de las unidades generadoras presenta un comportamiento similar, en donde las señales se encuentran oscilando desde el inicio y posterior a la perturbación, presenta mayores oscilaciones hasta encontrar la estabilidad.

4.5 Análisis de código fuente desarrollado en Python

Las funciones creadas en el código fuente permiten optimizar el estudio de estabilidad de pequeña señal para el sistema modificado IEEE de 39 barras ya que agilizan el proceso de búsqueda de la mejor configuración de conexiones realizando diferentes pruebas representadas por partículas. En función a la cantidad de partículas que ingresan al sistema, el código realiza una comparativa de los resultados de respuesta de frecuencia y selecciona al Best Global del enjambre. Esto automatiza la secuencia de realizar las pruebas de conexiones manualmente, exportar los resultados de frecuencia de los generadores síncronos, obtener un índice mediante cálculos y realizar la comparación histórica de los resultados. Incluso quedan guardados los resultados de las diferentes partículas en

las iteraciones para poder realizar una comprobación y poder visualizar los cambios de frecuencia con mayor exactitud durante el tiempo simulado.

No solo el estudio de estabilidad se ve optimizado, sino también los cálculos que vienen inmersos en el método metaheurístico PSO. Por medio de lazos *for*, se realiza el proceso matemático para la obtención de la velocidad y, posteriormente, la posición de cada partícula en la siguiente iteración sin la necesidad de realizar cálculos escritos. Las líneas de código se encargan incluso de definir los números aleatorios ($r1$ y $r2$) por medio de la librería *Random* y utilizando la dimensión de las partículas dada por la cantidad de generadores, tanto eólicos como fotovoltaicos, que tiene el sistema. Cuando se realizan los cálculos de velocidad y posición, los arreglos pueden venir con diferentes valores, esto se soluciona mediante estructuras de control *if* y *else*, que se encargan de condicionar a los arreglos para que solo se muestren como binario. La condición se valúa de la siguiente manera:

- *if* → Los números mayores a 0 serán 1.
- *else* → Los números menores a 0 serán 0.

Esto permite que se evalúen las partículas como configuración de conexiones en el sistema y se vayan obteniendo los índices. A partir de los datos históricos de la partícula, se obtienen los diferentes Best Locals que servirán para la selección del Best Global.

Las partículas mandan la señal de acción de los generadores para el evento de switch al que serán programados, las cuales pueden ser acción de apertura (*Open* \equiv 0) o acción de cierre (*Close* \equiv 1), tal como se muestra en la **Figura 4.12**.

Las acciones se asignan a los eventos de la partícula a la que corresponden dentro de la carpeta de eventos previamente definida. Los eventos de esta carpeta tienen que digitarse en orden, en donde el número de eventos depende de la cantidad de generación eólica y fotovoltaica que posea el sistema. En el sistema modificado IEEE de 39 barras se instaló generación eólica y fotovoltaica a 15 barras, lo que se traduce a 30 eventos para cada tipo de generador. La generación renovable no se

instala a todas las barras puesto que estamos considerando aspectos limitantes de la realidad como el poco flujo de aire o la poca irradiación solar que podría existir en ciertas áreas del sistema.

En la **Figura 4.13** se muestra los eventos agregados en *shfolder*, que es la carpeta de eventos que almacena los eventos por el número y tipo de generador al que corresponde.

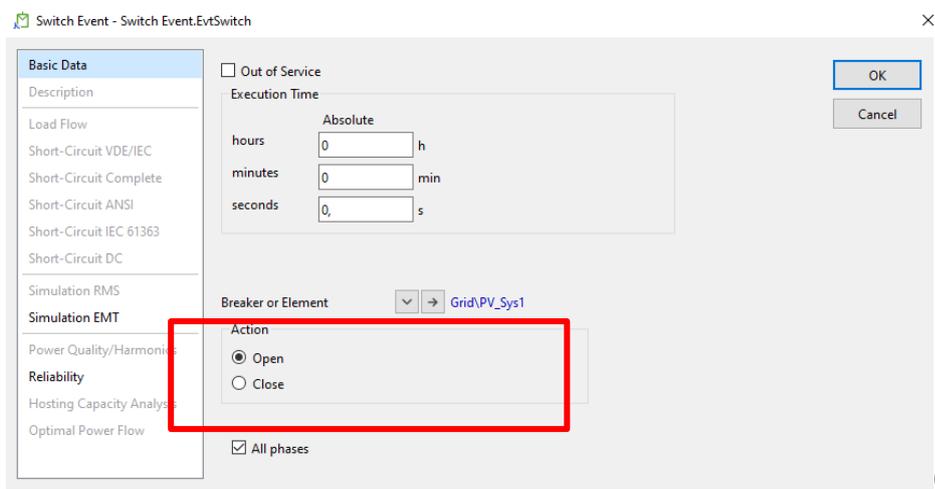


Figura 4.12. Acción del interruptor para el generador fotovoltaico PV_Sys1

```
# Creación de eventos
def shcfolder(p,i):
    shc_folder = app.GetFromStudyCase('IntEvt') # Abrir la carpeta para crear eventos
    cod=str(p)+str(i)
    shc_folder.CreateObject('EvtSwitch', 'Evento_switch_on'+cod) # Evento de switcheo apertura de carga
    shc_folder.CreateObject('EvtSwitch', 'Evento_switch_off'+cod) # Evento de switcheo cierre de carga

    shc_folder.CreateObject('EvtSwitch', '1Evento_switch_Pvsys'+cod) # Evento de switch PV_Sys 1
    shc_folder.CreateObject('EvtSwitch', '1Evento_switch_GenSt'+cod) # Evento de switch StaGen 1

    shc_folder.CreateObject('EvtSwitch', '2Evento_switch_Pvsys'+cod) # Evento de switch PV_Sys 2
    shc_folder.CreateObject('EvtSwitch', '2Evento_switch_GenSt'+cod) # Evento de switch StaGen 2

    shc_folder.CreateObject('EvtSwitch', '3Evento_switch_Pvsys'+cod) # Evento de switch PV_Sys 3
    shc_folder.CreateObject('EvtSwitch', '3Evento_switch_GenSt'+cod) # Evento de switch StaGen 3

    shc_folder.CreateObject('EvtSwitch', '4Evento_switch_Pvsys'+cod) # Evento de switch PV_Sys 4
    shc_folder.CreateObject('EvtSwitch', '4Evento_switch_GenSt'+cod) # Evento de switch StaGen 4

    shc_folder.CreateObject('EvtSwitch', '5Evento_switch_Pvsys'+cod) # Evento de switch PV_Sys 5
    shc_folder.CreateObject('EvtSwitch', '5Evento_switch_GenSt'+cod) # Evento de switch StaGen 5

    shc_folder.CreateObject('EvtSwitch', '6Evento_switch_Pvsys'+cod) # Evento de switch PV_Sys 6
    shc_folder.CreateObject('EvtSwitch', '6Evento_switch_GenSt'+cod) # Evento de switch StaGen 6
```

Figura 4.13. Creación de eventos

Se mencionó con anterioridad que los eventos tienen que digitarse textualmente en las líneas de código para que se puedan crear como eventos vacíos y se encuentren en la espera de su acción por parte de la partícula. Al intentar crear los eventos mediante lazos *for*, pudimos notar que no se actualizaba la carpeta y, por consiguiente, no creaban los nuevos eventos para la siguiente partícula. De esa manera, queda en los usuarios digitar la cantidad de eventos que sean necesarios para las características particulares del sistema con el cual deseen trabajar.

4.6 Análisis de estudio de estabilidad de pequeña señal

Mediante un análisis comparativo, se propone determinar la estabilidad del sistema modificado IEEE de 39 barras, con la configuración óptima de generación fotovoltaica y eólica proporcionada por el código fuente en Python, para perturbaciones de pequeña señal, una en *Load39* y otra en *Load23*, localizadas en extremos de la red.

Analizando el comportamiento de las señales del generador *Gen1* para el caso de una falla en *Load39*, se observa que el sistema posee la suficiente fuerza restauradora para desacelerar la máquina luego de la desconexión y conexión de carga, teniendo como resultado un valor porcentual de 0.08% que representa la diferencia entre la referencia inicial de frecuencia y el nuevo valor de frecuencia una vez alcanzada la estabilidad. El error porcentual para la potencia activa total es de 0.2%, en donde la mayor oscilación ocurre en el momento de la perturbación debido a que se deja de suministrar potencia al no existir la carga inicial del sistema, mientras que cuando se conecta la carga, la potencia se dispara para suplir esta demanda. El periodo de inestabilidad dura aproximadamente 11 segundos hasta que el generador logra equilibrar el torque electromagnético y mecánico del sistema, y alcance la estabilidad en 997.969645 MW. El comportamiento del voltaje terminal presenta un mejor escenario, ya que se amortigua en menor tiempo que las señales previas. El disparo que posee la señal de voltaje al momento de la desconexión de carga se debe al aumento de velocidad de giro del rotor por el aumento de frecuencia. Al reconectarse la carga, la señal presenta oscilaciones cada vez menores debido a la inyección de potencia reactiva por parte del

generador para suplir este desequilibrio de carga, logrando estabilizarse en un tiempo de 7.591667 segundos y a un valor de 1.041903 p.u, que representa un error porcentual del 4% con respecto a la referencia.

Como resultado del evento de falla 1 (Load39) y el evento de falla 2 (Load23), se obtienen las gráficas de frecuencia, potencia activa total y voltaje terminal de los 10 generadores síncronos del sistema. Tomando como referencia el generador *Gen1*, se observa que al presentarse una falla en la carga *Load39*, el valor pico de frecuencia es mayor que cuando la falla se presenta en *Load23*. Esto se debe a que la carga *Load39* se encuentra más cercana a la unidad generadora, lo que provoca un mayor impacto al sincronismo de esta. Esto se comprueba con la frecuencia de estabilidad, ya que con el evento de falla 1 se aprecia un tiempo de estabilidad menor debido a la cercanía que tiene con el generador, siendo de $t_{sim} = 11.541666 \text{ seg}$ a comparación de $t_{sim} = 12.561666 \text{ seg}$ que corresponde al tiempo en el que la frecuencia alcanza su estado estable luego del evento de falla 2.

Al analizar el comportamiento de la potencia activa total de los generadores síncronos, podemos observar que existe mayor oscilación para aquellos generadores cercanos a la perturbación, ya sea en el evento de falla 1 o el evento de falla 2, sin embargo, no se perdió el sincronismo ya que se alcanza el valor estable según la respuesta del gobernador de la máquina síncrona. Cuando ocurre la desconexión, la potencia de los generadores disminuye bruscamente mientras que, en la conexión de carga, la potencia aumenta, lo que deriva un aumento de frecuencia según este exceso. El voltaje terminal de los generadores presenta un comportamiento similar, obteniendo la estabilización de señal en un tiempo menor a los 10 segundos. Esto se debe al sistema de control que incorpora PSS y AVR al sistema, logrando mantener los valores adecuados de voltaje mediante el control de la potencia reactiva que se inyecta al sistema para suplir el desequilibrio de carga. La correcta parametrización de los sistemas de control permite garantizar la estabilidad del voltaje terminal en un menor tiempo, comprobando la confiabilidad y seguridad de la red. La normalización referente a estabilidad se la puede apreciar en mayor detalle en la sección de **Apéndice D**.

CAPÍTULO 5

Conclusiones

- El algoritmo propuesto basado en el método de PSO desarrollado en Python, permite determinar la mejor ubicación de la generación eólica y fotovoltaica frente a una perturbación de desconexión y conexión de carga, considerando el estudio de estabilidad de pequeña señal en función del parámetro de frecuencia eléctrica. Los elementos activos de generación eólica y fotovoltaica integrados al sistema para el escenario de ubicación óptima se muestran en la **Figura 4.1** cuando la perturbación ocurre en la carga denominada Load39.
- El control maestro establecido por el código fuente desarrollado en Python permite automatizar el proceso de búsqueda de la configuración de conexiones óptimas para la generación eólica y fotovoltaica de cualquier sistema eléctrico de potencia, basando el proceso en el estudio de estabilidad de pequeña señal que se realiza con los datos obtenidos de frecuencia eléctrica [Hz], potencia activa de generación [MW], voltaje terminal [p.u] tabulados en formato de texto “.txt” exportados por medio de DigSILENT-PowerFactory.
- La ubicación de generación eólica y fotovoltaica considerada en el sistema modificado IEEE de 39 barras presenta un comportamiento óptimo de estabilidad, alcanzando la amortiguación en promedio a los 12.101666 segundos para un evento de desconexión en 0.25 segundos y conexión en 0.5 segundos en la carga Load39, es decir el tiempo de recuperación del sincronismo del sistema es menor y está dentro del rango máximo establecido de 15 – 20 segundos para la realización de estudios de estabilidad de pequeña señal después de la manifestación del disturbio en el sistema.
- Los generadores síncronos que se encuentren más cercanos a la carga perturbada presentan oscilaciones más severas y un mayor tiempo de estabilización a comparación de los que se encuentran más alejados, pero generalmente, todos los generadores presentan una rápida amortiguación debido a la inclusión de los controladores de PSS y AVR.

- Los índices que se han obtenido en el estudio de estabilidad del sistema en función del parámetro de frecuencia eléctrica [Hz], permiten comparar diferentes configuraciones de conexiones determinando el punto óptimo de ubicación eólica y fotovoltaica para alcanzar la estabilidad en menor tiempo frente a una perturbación de pequeña señal, considerando como un sistema ideal cuando el índice presente un valor de uno, el índice de estabilidad está representado por la **ecuación 3.29**.

Recomendaciones

- Implementar distintos tipos de eventos en el sistema planteado en el proyecto tales como: cortocircuito en la línea, cortocircuito en la barra, sobretensión, sobrecorriente, desconexión de líneas y también reemplazar la generación síncrona por generación eólica y fotovoltaica.
- Identificar como puntos críticos del sistema a los generadores que se encuentran más cercanos a la perturbación, ya que el impacto del evento de falla es mayor, esto se observa en el incremento de la magnitud de la frecuencia eléctrica, potencia activa y voltaje terminal en el intervalo de duración de la perturbación en el sistema, por ende, se genera un incremento en la afección del sincronismo de las generadoras con respecto a la generación síncrona más alejada del evento de falla.
- Verificar la versión de Python que se está utilizando como interprete, ya que DigSILENT-PowerFactory permite enlazar desde la versión 3.3 a la versión 3.8, considerando que el módulo "powerfactory.pyd" es distinto para cada versión.
- Implementar variaciones del método metaheurístico PSO para comparar los resultados de arreglos denotados como óptimos en términos de ubicación y verificar su eficiencia referente a estabilidad.

BIBLIOGRAFÍA

- [1] H. Saadat, *Power System Analysis*, vol. 19, McGraw-Hill Higher Education, 2009, pp. 192-198.
- [2] T. K. Renuka, P. Reji y S. Sreedharan, «An enhanced particle swarm optimization algorithm for improving the renewable energy penetration and small signal stability in power system,» *Renewables: Wind, Water, and Solar*, vol. 5, nº 1, pp. 1-17, 2018.
- [3] Q. Bai, «Analysis of particle swarm optimization algorithm,» *Computer and information science*, vol. 3, nº 1, pp. 180-184, 2010.
- [4] Python, «Introducing,» *Python*, 2019.
- [5] DigSILENT PowerFactory, «DigSILENT PowerFactory 15 User Manual,» DigSILENT GmbH, Gomaringen, 2014.
- [6] M. C. Zapata Cevallos, Artist, *Estabilidad de pequeña señal en sistemas de energía eléctrica con alta penetración de generación renovable*. [Art]. Universidad Nacional de Colombia, 2020.
- [7] M. Ramírez Castaño y L. A. Duque Torres, *Análisis de Estabilidad de Pequeña Señal en el Sistema Eléctrico de Pereira*, vol. 9, Pereira, 2009, pp. 166-180.
- [8] J. D. Glover, M. S. Sarma y T. J. Overbye, *Power System Analysis and Design*, Boston: Cengage Learning, 2015.
- [9] P. Kundur, *Power System Stability and Control*, California, Holguín: McGraw-Hill, 1994.
- [10] L. J. Arcos Guerrón y I. S. Flores Gómez, Artists, *Análisis de Estabilidad Transitorio del Sistema Nacional Interconectado del Ecuador*. [Art]. Escuela Politécnica Nacional, 2007.
- [11] K. Mehmet, Y. A. Çengel y J. M. Cimbala, *Fundamentals and applications of renewable energy*, vol. 19, McGraw-Hill Education, 2020, pp. 917-937.
- [12] EIA, *U.S. Energy Information Administration, Annual Energy Outlook*, Washington, DC, 2018, pp. 27-50.
- [13] IRENA, «Renewable Capacity Highlights,» 2020. [En línea]. Available: <https://www.irena.org/publications>.
- [14] G. Cook, L. Billman y R. Adcock, *Photovoltaic fundamentals*, vol. VI, Washington, DC: Solar Energy Research Institute, 1991, pp. 348-360.

- [15] R. Shah, N. Mithulananthan, A. Sode-Yome y K. Y. Lee, «Impact of large-scale PV penetration on power system oscillatory stability,» *IEEE PES general meeting*, pp. 1-7, 2010.
- [16] E. Franklin, «Solar Photovoltaic (PV) System Components,» *The University of Arizona College of Agriculture and Life Sciences*, pp. 1-8, 2018.
- [17] J. F. Manwell, J. G. McGowan y A. L. Rogers, *Wind energy explained: theory, design and application*, John Wiley & Sons, 2010.
- [18] T. He, S. Li, S. Wu y K. Li, «Small-Signal Stability Analysis for Power System Frequency Regulation with Renewable Energy Participation,» *Mathematical Problems in Engineering*, vol. 2021, pp. 1-13, 2021.
- [19] . X. S. Yang, *Nature-inspired optimization algorithms*, Academic Press., 2020.
- [20] J. J. Liang, A. K. Qin, P. N. Suganthan y S. Baskar, «Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,» *IEEE transactions on evolutionary computation*, vol. 10, nº 3, pp. 281-295, 2006.
- [21] A. Zea, A., Artist, *Power System Stabilizers for The Synchronous Generator*. [Art]. Chalmers University of Technology, 2013.
- [22] I. Challenger Pérez, Y. Díaz Ricardo y R. A. Becerra García, «El lenguaje de programación Python,» *Ciencias Holguín*, vol. 20, nº 2, pp. 1-13, 214.
- [23] F. Gonzalez Longatt, «Manual de usuario de DigSILENT PowerFactory 12.0 en Español,» 2004.
- [24] J. D. Arcila, «DigSILENT Básico,» Instituto de Estudios Bursátiles, 2015.
- [25] JetBrains Corporation, «Funcionalidades - PyCharm,» JetBrains Corporation, 2021. [En línea]. Available: <https://www.jetbrains.com/es-es/pycharm/features/>. [Último acceso: 23 12 2021].
- [26] JetBrains Corporation, «Announcing General Availability of PyCharm 4 | The PyCharm Blog,» JetBrains Blog, 18 11 2014. [En línea]. Available: <https://blog.jetbrains.com/pycharm/2014/11/announcing-general-availability-of-pycharm-4/>. [Último acceso: 23 12 2021].

- [27] JetBrains Corporation, «JetBrains Plugin Repository,» JetBrains Marketplace, [En línea]. Available: <https://plugins.jetbrains.com/pycharm>. [Último acceso: 23 12 2021].
- [28] C. W. Aguirre Cárdenas, *Estudios Eléctricos de Sistemas de Potencia para la Carrera de Ingeniería Eléctrica Utilizando el Software PowerFactory 13.1 de DigSILENT*, Quito: Escuela Politécnica Nacional, 2008.
- [29] S. L. Z. Aun, . M. B. Marsadek y A. K. Ramasamy, «Small signal stability analysis of grid connected photovoltaic,» *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 6, nº 3, pp. 553-562, 2017.
- [30] IEEE Power & Energy Society Reports, «Dynamic Models For Turbine-Governors in Power System Studies,» Technical Reports PES-TR1, Enero, 2013.
- [31] P. Cruz Romero y A. de la Villa Jaén, *Mejora de la estabilidad transitoria en líneas de gran longitud mediante el uso de Facts.*, Sevilla: ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA, 2016.
- [32] M. S. Alvarez-Alvarado, C. D. Rodríguez-Gallegos and D. Jayaweera, “Optimal planning and operation of static VAR compensators in a distribution system with non-linear loads,” *IET Generation, Transmission & Distribution*, vol. 12, no. 15, pp. 3726-3735, 2018.
- [33] K. W. Arnaud , Artist, *Simulación dinámica para distintos escenarios de alta penetración eólica en el SIC*. [Art]. Universidad de Chile, 2008.
- [34] B. J. Kirby, J. Dyer, C. Martinez, R. A. Shoureshi , R. Guttromson y J. Dagle, «Frequency control concerns in the North American electric power system,» *United States. Department of Energy*, pp. 1949-3053, 2003.

APÉNDICES

APÉNDICE A: DESARROLLO MATEMÁTICO

Solución de la ecuación swing con amortiguación y pequeña perturbación.

Se conoce la ecuación swing:

$$\frac{d^2(\Delta\delta)}{dt^2} + \frac{\pi f_0}{H} \left(D \frac{d\Delta\delta}{dt} \right) + \frac{\pi f_0}{H} P_s \Delta\delta = 0$$

$$\omega_n = \sqrt{\frac{\pi f_0}{H} P_s}$$

$$\zeta = \frac{D}{2} \sqrt{\frac{\pi f_0}{H P_s}} < 1$$

Donde:

ω_n : La frecuencia natural;

ζ : coeficiente de amortiguamiento

Aplicando la transformada de Laplace

$$\mathcal{L} \left\{ \frac{d^2(\Delta\delta)}{dt^2} + \frac{\pi f_0}{H} \left(D \frac{d\Delta\delta}{dt} \right) + \frac{\pi f_0}{H} P_s \Delta\delta = 0 \right\}$$

$$(s^2 \Delta\delta(s) - s \Delta\delta_0 - \Delta\delta'(0)) + 2\zeta\omega_n(s\Delta\delta(s) - \Delta\delta_0) + \omega_n^2 \Delta\delta(s) = 0$$

$$\Delta\delta'(0) = 0$$

$$\mathcal{L}^{-1} \left\{ \Delta\delta(s) = \frac{(s + 2\zeta\omega_n)\Delta\delta_0}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right\}$$

L.Q.Q.D [1], [8]:

$$\Delta\delta = \frac{\Delta\delta_0}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta)$$

Solución de la ecuación swing con amortiguación, pequeña perturbaciones y variaciones de potencia

Se conoce la ecuación swing considerando ahora una variación de potencia (ΔP):

$$\frac{d^2(\Delta\delta)}{dt^2} + \frac{\pi f_0}{H} \left(D \frac{d\Delta\delta}{dt} \right) + \left(\frac{\pi f_0}{H} \right) P_s \Delta\delta = \left(\frac{\pi f_0}{H} \right) \Delta P$$

$$\omega_n = \sqrt{\frac{\pi f_0}{H} P_s}$$

$$\zeta = \frac{D}{2} \sqrt{\frac{\pi f_0}{H P_s}} < 1$$

Donde:

ω_n : La frecuencia natural;

ζ : coeficiente de amortiguamiento

Aplicando la transformada de Laplace

$$\mathcal{L} \left\{ \frac{d^2(\Delta\delta)}{dt^2} + \frac{\pi f_0}{H} \left(D \frac{d\Delta\delta}{dt} \right) + \frac{\pi f_0}{H} P_s \Delta\delta = 0 \right\}$$

$$(s^2 \Delta\delta(s) - s\Delta\delta_0 - \Delta\delta'(0)) + 2\zeta\omega_n(s\Delta\delta(s) - \Delta\delta_0) + \omega_n^2 \Delta\delta(s) = \Delta u$$

$$\Delta\delta'(0) = 0$$

$$\mathcal{L}^{-1} \left\{ \Delta\delta(s) = \frac{\Delta u - (s + 2\zeta\omega_n)\Delta\delta_0}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right\}$$

L.Q.Q.D [1], [8]:

$$\Delta\delta = \frac{\Delta u}{\omega_n^2} \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta) \right]$$

APÉNDICE B: DATOS DINÁMICOS SÍNCRONOS

Datos de máquina

Tabla B.1. Datos de Máquina de la Red Modificada IEEE 39 Barras

Tipo de operación Default Unit no. (New Unit no.)	GENROU Sync. Gen. 30(40)	GENROU Sync. Gen. 31(41), 32(42), 33(43), 34(44), 35(45), 36(46), 37(47)	GENROU Sync. Gen. 38(48), 39(49)
Potencia aparente nominal (MVA)	590	835	911
Voltaje nominal (kV)	22	20	26
Nominal pf	0.95	0.9	0.9
H(s)	2.3186	2.6419	2.4862
D	2.00	2.00	2.00
$r_a(p.u)$	0.0046	0.0019	0.0010
$x_d(p.u)$	2.110	2.183	2.040
$x_q(p.u)$	2.020	2.157	1.960
$x'_d(p.u)$	0.280	0.413	0.266
$x'_q(p.u)$	0.490	1.285	0.262
$x''_d(p.u)$	0.215	0.339	0.193
$x''_q(p.u)$	0.215	0.339	0.193
x_i o $x_p(p.u)$	0.155	0.246	0.154
$T'_{d0}(s)$	4.200	5.690	6.000
$T'_{q0}(s)$	0.565	1.500	0.900
$T''_{d0}(s)$	0.032	0.041	0.004
$T''_{q0}(s)$	0.062	0.144	0.004
S(1.0)	0.079	0.134	0.340
S(1.2)	0.349	0.617	1.120

Datos de excitatriz

Tabla B.2. Datos de Excitatriz de la Red Modificada IEEE 39 Barras

Tipo	IEEE TI	IEEE TI	IEEE TI
Default Unit no. (New Unit no.)	30(40)	31(41), 32(42), 33(43), 34(44), 35(45), 36(46), 37(47)	38(48)
Potencia aparente nominal (MVA)	590	835	911
Voltaje nominal (kV)	22	20	26
$T_r(s)$	0.000	0.000	0.000
$K_a(p.u)$	200	400	50
$T_a(s)$	0.3575	0.020	0.060
$V_{Rmax}(p.u)$	5.730	18.300	1.000
$V_{Rmin}(p.u)$	-5.730	-18.300	-1.00
$K_e(p.u)$	1.000	1.000	-0.0393
$T_e(s)$	0.004	0.942	0.440
$K_f(p.u)$	0.0529	0.030	0.070
T_f	1.000	1.000	1.000
E_1	4.2975	3.765	3.375
$SE(E_1)$	0.000	0.8147	0.0644
E_2	5.730	5.020	4.5
$SE(E_2)$	0.000	2.6756	0.2363

Datos de gobernador

Tabla B.3. Datos del Gobernador en la Red Modificada IEEE 39 Barras

Tipo	BPA_GG	BPA_GG	BPA_GG
Default Unit no. (New Unit no.)	30(40)	31(41), 32(42), 33(43), 34(44), 35(45), 36(46), 37(47)	38(48)
Potencia aparente nominal (MVA)	590	835	911
Voltaje nominal (kV)	22	20	26
$P_{max}(p.u)$	0.9373	0.9177	0.9001
$R(p.u)$	0.0085	0.006	0.00548
$T_1(s)$	0.080	0.180	0.100
$T_2(s)$	0.000	0.030	0.000
$T_3(s)$	0.150	0.200	0.200
$T_4(s)$	0.050	0.000	0.100
$T_5(s)$	10.000	8.000	8.720
F	0.280	0.300	0.300

APÉNDICE C: COMANDOS DE PROGRAMACIÓN

Comandos utilizados en el código fuente

Tabla C.4. Funciones Implementadas en Código Fuente de Python

Comando	Descripción
<code>GetApplication()</code>	Devuelve el programa que se está llamando.
<code>GetCurrentUser()</code>	Devuelve el dominio o nombre de usuario que usa el programa.
<code>GetActiveProject()</code>	Activa el proyecto del programa.
<code>GetCalcRelevantObjects()</code>	Filtra los elementos del proyecto por nombre y tipo.
<code>isnumeric()</code>	Devuelve True si todos los caracteres de una cadena son numéricos. Caso contrario, devuelve False.
<code>read_csv()</code>	Función de pandas para leer archivos (tipo CSV o TXT) y realizar operaciones en él.
<code>astype()</code>	Función de pandas para convertir un tipo de datos de columna particular en otro tipo de datos.
<code>mean()</code>	Función para calcular la media/promedio de una lista dada de números.
<code>concat()</code>	Función para unir dos o más arreglos
<code>GetFromStudyCase()</code>	Accede a la carpeta de elementos del proyecto y selecciona un caso de estudio. De no existir el caso de estudio, la función lo creará.
<code>CreateObject()</code>	Se crea un objeto dentro de la carpeta del caso de estudio.
<code>append()</code>	Añade un elemento al final de la lista.
<code>keys()</code>	Devuelve un objeto de vista.
<code>split()</code>	Divide una cadena en una lista.
<code>DataFrame()</code>	Devuelve una estructura de datos bidimensional con columnas de diferentes tipos.
<code>GetContents()</code>	Accede al contenido de una carpeta.
<code>randrange()</code>	Devuelve un número aleatorio dentro de un rango específico.
<code>array()</code>	Devuelve el elemento en formato de arreglo.
<code>Execute()</code>	Inicia la ejecución del caso de estudio.

APÉNDICE D: ESTÁNDARES DE OPERACIÓN [33]

Normativa de Chile

Tabla D.5. Estándar de Niveles de Tensión en Barras de Red de Transmisión

Nivel de tensión [kV]	Condición normal		Condición de alerta		Condición de emergencia	
	Valor mínimo	Valor máximo	Valor mínimo	Valor máximo	Valor mínimo	Valor máximo
[500, +∞)	0.97	1.03	0.96	1.04	0.95	1.05
[200, 500)	0.95	1.05	0.93	1.07	0.90	1.10
(-∞, 200]	0.93	1.07	0.91	1.09	0.90	1.10

Tabla D.6. Normalización de Centrales de Generación por Control de Tensión

Parámetro eléctrico	Condición normal		Condición de alerta		Condición de emergencia	
	Valor mínimo	Valor máximo	Valor mínimo	Valor máximo	Valor mínimo	Valor máximo
Tensión	0.95	1.05	0.95	1.05	0.95	1.05
Potencia reactiva	0.9 capacitivo	0.9 inductivo	1 capacitivo	1 inductivo	1 capacitivo	1 inductivo

Tabla D.7. Limitantes de Frecuencia en Estado Estable

Condición de frecuencia	Valor mínimo [Hz]	Valor máximo [Hz]
Normal	49.8	50.2
Alerta	49.8	50.2
Emergencia	49.5	50.5

Tabla D.8. Requerimientos Mínimos de Estabilización en Estado Dinámico

Tipo de sistema	Valor mínimo de frecuencia [Hz]	Valor máximo de permanencia [s]
Troncal	48.3	0
Subtransmisión	48.3	0.2

Tabla D.9. Limitantes de Estabilidad y Seguridad del Sistema

Parámetro eléctrico	Condición Normal	Condición de alerta	Condición de emergencia
Tensión	0.2	0.2	0.1
Angular	0.3	0.3	0.3

Normativa internacional europea

Tabla D.10. Norma de Frecuencia con Integración de Generación Eólica

Frecuencia eléctrica [Hz]	Tiempo de operación mínimo en Escocia [s]	Tiempo de operación mínimo en Irlanda [min]	Tiempo de operación mínimo en Dinamarca [min]	Tiempo de operación mínimo en Alemania [min]	Tiempo de operación mínimo en Inglaterra [s]
[52, 53]	Desconectado	Desconectado	3	Desconectado	Desconectado
[51.5, 52]	Conectado	60	30	Desconectado	Conectado
[51, 51.5]	Conectado	60	30	Conectado	Conectado
[50.5, 51]	Conectado	60	30	Conectado	Conectado
[49.5, 50.5]	Conectado	Conectado	Conectado	Conectado	Conectado
[47.5, 49.5]	Conectado	60	30	Conectado	Conectado
[47, 47.5]	20	0.33 = 20 s	3	Desconectado	20
$(-\infty, 47)$	20	0.33 = 20 s	Desconectado	Desconectado	20

Normativa internacional en américa del norte [34]

Tabla D.11. Estándar de Frecuencia en América del Norte

Características	Frecuencia
Banda de control en condición de operación normal	± 0.05 Hz
Frecuencia de resonancia	$< 5\%$ Si $f = 60$ Hz $< 5\%f = 3$ Hz