

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Título del trabajo

**PROGRAMACIÓN CUÁNTICA PARA ESTUDIO DE
CONFIABILIDAD A NIVEL JERÁRQUICO I POR MÉTODO
MONTE CARLO**

Previo la obtención del Título de:

INGENIERO EN ELECTRICIDAD

Presentado por:

Jorge Alexander Arteaga Quinto

Alexander Ricardo Bazán Zurita

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA 1

El presente proyecto lo dedico a mis padres, mis hermanos; a los docentes que he tenido a lo largo de mi vida académica y profesional.

Así también a mis compañeros del preuniversitario con los cuales siempre puedo contar y me han brindado su apoyo incondicional a lo largo de la carrera.

Alexander Bazán Zurita

DEDICATORIA 2

Dedico este proyecto de grado a un ser muy especial en mi vida, mi madre que siempre me acompaña a cada paso y es el pilar fundamental en todos mis logros, a mi hermana y familia. Que cada logro en mi vida siempre esté acompañado por su presencia.

De igual forma a mis amigos, compañeros y docentes que siempre estuvieron aportando con conocimientos, ayudas y experiencias.

Jorge Arteaga Quinto

AGRADECIMIENTO 1

Quiero agradecer a Dios por brindarme la vida. A mis padres por el esfuerzo que han hecho a lo largo de mis años y me han dado la formación y el conocimiento necesario para ser una gran persona y futuro profesional.

A mis hermanos por cada uno de sus consejos, enseñanzas y el amor incondicional que siempre me dan.

Mis más sinceros agradecimientos a mis tutores académicos por la orientación y guía en el desarrollo de este trabajo de titulación.

A mis compañeros de carrera, por sobre todo a Jorge y Ernesto que han sido un apoyo incondicional en estos últimos años de estudio y con los cuales viví gratos momentos en los momentos finales de la carrera.

Alexander Bazán Zurita

AGRADECIMIENTO 2

Agradezco a Dios por siempre brindarme sabiduría y guiarme en todo momento para tomar las decisiones correctas, a mi madre y familia por siempre estar pendientes y haberme enseñado e inculcado muchos valores que hoy en día representan el fruto de sus enseñanzas. También, quisiera agradecer a personas muy valiosas que conocí a lo largo de mi carrera y vida como son: Ing. Jhon Arreaga, Ing. Irving Tapia, Ing. Carlos Galarza, Ing. José Rivera y sin dejar a un lado a mi gran compañero Alexander Bazán por tomarme en cuenta en este majestuoso proyecto junto con el PhD. Manuel Álvarez que gracias a su ingenio y ayuda se logró culminar con éxito este trabajo.

¡Gracias! por formar parte de mi vida y haberme ayudado y motivado en todo este proceso.

Jorge Arteaga Quinto

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Jorge Arteaga y Alexander Bazán, 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”

Jorge Alexander
Arteaga Quinto

Alexander Ricardo
Bazán Zurita

EVALUADORES

.....
Msc. IVAN ENDARA

PROFESOR DE LA MATERIA

.....
PhD. MANUEL ALVAREZ

PROFESOR TUTOR

RESUMEN

El presente trabajo pretende demostrar la aplicación de una nueva alternativa para realizar estudios de confiabilidad mediante programación cuántica en sistemas de potencia a través del desarrollo de un algoritmo. En la actualidad gran parte de los métodos estocásticos tradicionales como las cadenas de Márkov, simulación de eventos aleatorios y el método de Montecarlo requieren de una gran carga computacional.

Ante esta necesidad se optó por el desarrollo de un algoritmo denominado Algoritmo Híbrido (AH) en el lenguaje de programación Python^{MT}, dentro del cual se empleó la librería Qiskit.org la cual permitió realizar el enlace y simulación en los aceleradores cuánticos.

La preparación de estos recursos fue necesaria para llevar a cabo un estudio de confiabilidad en nivel jerárquico 1 en dos distintos sistemas de potencia, mismos que permitieron evaluar su rendimiento comparando los resultados de los índices de confiabilidad LOLE, EENS, XLOL, LOLP del AH con el método tradicional de MonteCarlo. Finalmente, al comparar aspectos de precisión, exactitud, velocidad, aceleración y tiempo de simulación a través del “Pentágono de Características” se logra demostrar que el AH tiene un gran desempeño en el cálculo y estimación de índices pese al tiempo exagerado de simulación requerido.

Palabras Clave: Computación Cuántica, Estudio de Confiabilidad, Programación, Método de Montecarlo, Qubit.

ABSTRACT

This work aims to demonstrate the application of a new alternative to perform reliability studies using quantum programming in power systems through the development of an algorithm. At present traditional stochastic methods such as Markov chains, simulation of random events and the Monte Carlo method require a lot of computational resources.

Given this need, we opted for the development of an algorithm called Hybrid Algorithm (HA) in the programming language Python MT, within which we used the Qiskit.org library which allowed us to perform the link and simulation in quantum accelerators.

The preparation of these resources was necessary to carry out a reliability study at hierarchical level 1 in two different power systems, which allowed to evaluate their performance by comparing the results of the reliability indices like LOLE, EENS, XLOL, LOLP of AH with the traditional method of MonteCarlo.

Finally, when comparing aspects of precision, accuracy, speed, acceleration, and simulation time through the "Pentagon of Characteristics" it is possible to demonstrate that the HA has a great performance in the calculation and estimation of indices despite the exaggerated time of simulation required.

Keywords: Quantum Computing, Reliability Study, Programming, Monte Carlo Method, Qubit.

ÍNDICE GENERAL

EVALUADORES.....	7
RESUMEN.....	8
ABSTRACT	9
ÍNDICE GENERAL	10
ABREVIATURAS.....	13
SIMBOLOGÍA.....	14
ÍNDICE DE FIGURAS	15
ÍNDICE DE TABLAS.....	16
CAPÍTULO 1.....	17
1. INTRODUCCIÓN.....	17
1.1 DESCRIPCIÓN DEL PROBLEMA.....	18
1.2 JUSTIFICACIÓN DEL PROBLEMA.....	19
1.3 OBJETIVOS	20
1.3.1 Objetivo General.....	20
1.3.2 Objetivos Específicos.....	20
1.4 MARCO TEÓRICO.....	20
1.4.1 Confiabilidad	20
1.4.2 Sistemas eléctricos de potencia.....	22
• Zona funcional de generación.....	22
• Zona funcional de transmisión	22
• Zona funcional de distribución	22
1.4.3 Niveles Jerárquicos.....	22
• Nivel Jerárquico I	22
• Nivel Jerárquico II	23
• Nivel Jerárquico III	23

1.4.4	Índices de confiabilidad.....	23
1.4.5	Computadoras Cuánticas	24
1.4.6	Lenguajes de programación	25
1.4.7	Bit Cuántico.....	26
1.4.8	Compuertas cuánticas	27
1.4.9	Mediciones en diferentes bases a la computacional.....	30
1.4.10	Circuitos Cuánticos	30
1.4.11	Método de simulación de Monte Carlo.....	31
1.4.12	Método de Monte Carlo en estudios de confiabilidad	33
CAPÍTULO 2.....		35
2.	METODOLOGÍA PARA LA CONSTRUCCIÓN DEL ALGORITMO CUÁNTICO APLICADO A ESTUDIOS DE CONFIABILIDAD	35
2.1	NSMC TRADICIONAL.....	35
2.2	METODO HIBRIDO – ALGORITMO CUANTICO.....	39
2.2.1	Lenguaje de programación Q.....	40
2.2.2	Implementación del método de NSMC en el algoritmo cuántico.....	42
2.3	MÉTRICAS PARA EVALUACIÓN DE ALGORITMOS	45
2.3.1	Exactitud y precisión	45
2.3.2	Velocidad y aceleración	46
2.4	CASOS DE ESTUDIO	46
2.4.1	Caso de estudio 1	46
2.4.2	Caso de estudio 2	47
CAPÍTULO 3.....		50
3.	RESULTADOS Y ANÁLISIS	50
3.1	CASO DE ESTUDIO 1	50
3.1.1	Índices de confiabilidad.....	50
3.1.2	Precisión y exactitud	55

3.1.3	Velocidad y aceleración	56
3.1.4	Tiempo de simulación	57
3.2	CASO DE ESTUDIO 2	58
3.2.1	Índices de confiabilidad.....	58
3.2.2	Precisión y exactitud	63
3.2.3	Velocidad y aceleración	64
3.2.4	Tiempo de simulación	65
3.3	RENDIMIENTO GLOBAL PARA EVALUACIÓN DE CRITERIOS DE MEDICIÓN 66	
3.3.1	Caso de estudio I	67
3.3.2	Caso de estudio II	69
3.4	ANÁLISIS DE RESULTADOS	71
CAPÍTULO 4.....		74
4.	CONCLUSIONES Y RECOMENDACIONES.....	74
4.1	CONCLUSIONES.....	74
4.2	RECOMENDACIONES	75
BIBLIOGRAFÍA.....		76
ANEXOS.....		79
ANEXO A: TABLAS DE ERROR RELATIVO		79
ANEXO B: CÓDIGOS PYTHONTM- VISUAL CODETM.....		80

ABREVIATURAS

ESPOL Escuela Superior Politécnica del Litoral
SEP Sistema Eléctrico de Potencia
CNEL Corporación Nacional de Electricidad
LOLE Loss of Load Expectation
LOLP Loss of Load Probability
XLOL Exped Loss of Load
EENS Expected Energy Not Supply
NERC North American Electric Reliability Council
IBM International Business Machines
QCL Quantum Computation Language
SMC Montecarlo Secuencial
NSMC No Secuencial Montecarlo
MTTF Mean Time to Failure
MTTR Mean Time to Repair
RTS Reliability Test System
IEEE Institute of Electrical and Electronics Engineers

SIMBOLOGÍA

λ	Taza de falla
r	Duración media de falla
$ \psi\rangle$	Qubit
α	Alfa
β	Beta
$ 0\rangle$	Estado cero de un qubit
$ 1\rangle$	Estado uno de un qubit
Θ	Angulo Theta
Φ	Angulo Phi
X	Compuerta X de Pauli
Y	Compuerta Y de Pauli
Z	Compuerta Z de Pauli
H	Compuerta Hadamard
\otimes	Producto tensorial
ϵ	Épsilon
σ	Desviación Estándar
x	Valor de probabilidad de salida esperado
\bar{x}	Promedio de probabilidades
σ^2	Varianza
ϵ	Error aditivo
ui	Sistema
U_{exp}	Estados del sistema
P	Función de eventos de probabilidad
\bar{Q}	Indisponibilidad del sistema
U_i	Índice FOR
A_i	Disponibilidad del sistema
δ	Exactitud
φ	Precisión
ω	Velocidad
α	Aceleración
t	Tiempo de simulación

ÍNDICE DE FIGURAS

Figura 1-1 Clasificación en zonas funcionales y niveles jerárquicos	21
Figura 1-2 Primera computadora cuántica comercial [13]	25
Figura 1-3 Esfera de Bloch Representación de un qubit [14]	27
Figura 1-4 Compuerta CNOT	29
Figura 1-5 Circuito cuántico con dos qubits entrelazados	30
Figura 1-6 Bloque de medición en un qubit	31
Figura 1-7 Algoritmo de Monte Carlo [6].....	34
Figura 2-1 Modelo de margen de generación y demanda [6]	37
Figura 2-2 Diagrama de flujo de NSMC tradicional	38
Figura 2-3 Secuencia de instrucciones de programación cuántica [21].....	39
Figura 2-4 Instrucciones de secuencia de programación cuántica [21]	40
Figura 2-5 Distribución de probabilidad de funcionamiento de un generador.....	43
Figura 2-6 Diagrama de flujo del algoritmo híbrido	44
Figura 2-7: Perfil de carga Anual [26].....	47
Figura 2-8: Diagrama unifilar del RTS de 24 barras [27]	49
Figura 3-1: Índices de confiabilidad caso 1.....	52
Figura 3-2: Evolución de índice ENNS para NSMC híbrido caso 1	53
Figura 3-3: Evolución de índice ENNS para NSMC tradicional caso 1	54
Figura 3-4 Precisión, exactitud, velocidad y aceleración caso 1.....	57
Figura 3-5 Tiempo de simulación caso 1	58
Figura 3-6 Índices de confiabilidad caso 2.....	60
Figura 3-7 Evolución de índice ENNS para NSMC híbrido caso 2	61
Figura 3-8 Evolución de índice ENNS para NSMC tradicional caso 2.....	62
Figura 3-9 Precisión, exactitud, velocidad y aceleración caso 2.....	65
Figura 3-10 Tiempo de simulación caso 2.....	66
Figura 3-11 Técnica de optimización del rendimiento caso 1	69
Figura 3-12 Técnica de optimización del rendimiento en caso 2	71

ÍNDICE DE TABLAS

Tabla 1-1 Compuerta CNOT.....	29
Tabla 2-1: Parámetros de generadores para estudio de confiabilidad	46
Tabla 2-2: Información de barras del sistema [27].....	48
Tabla 2-3: Parámetros de generadores para estudio de confiabilidad	48
Tabla 3-1 Índices de confiabilidad caso 1	51
Tabla 3-2 Resultados de precisión y exactitud caso 1.....	55
Tabla 3-3 Resultados de velocidad y aceleración caso 1	56
Tabla 3-4 Tiempo de simulación caso 1	57
Tabla 3-5: Índices de confiabilidad caso 2.....	59
Tabla 3-6: Resultados de precisión y exactitud caso 2.....	63
Tabla 3-7: Resultados de velocidad y aceleración para el caso 2	64
Tabla 3-8 Tiempo de simulación caso 2	65
Tabla 3-9 Escala de datos para el tiempo de simulación en caso 1	67
Tabla 3-10 Escala de datos para el tiempo de simulación en caso 2	67

CAPÍTULO 1

1. INTRODUCCIÓN

Todos los sistemas eléctricos tienen la función principal de lograr proporcionar energía eléctrica a cada uno de los usuarios conectados a la red ya sea de transmisión, subtransmisión o distribución, mediante los diferentes centros de generación que pueden ser térmicos, hidráulicos, eólicos etc. Toda esa energía generada debe ser transportada a los diferentes centros específicos de consumo como lo son: poblados, ciudades, grandes centros industriales, centros comerciales, hospitales, etc. Con todos estos grandes centros de consumo cada una de las estaciones de generación deben de ser capaces de entregar y abastecer de energía de manera segura y confiable al consumidor final. [1]

Al hablarse de confiabilidad se entra en un enfoque muy importante hoy en día para la operación de un Sistema Eléctrico de Potencia (SEP), al obtenerse una respuesta óptima de la demanda por parte de la generación conlleva a grandes estudios del comportamiento del sistema y saber que operaciones realizar para que se logre respaldar la confiabilidad a través de la viabilidad de la carga y la mitigación de la demanda [2]. Una evaluación de confiabilidad del sistema se ocupa principalmente en la determinación de la disponibilidad o indisponibilidad del sistema ya sea en generación, transmisión y distribución para así lograr proporcionar un suministro óptimo a cada uno de los puntos de carga. Según Li, [3], para un estudio de confiabilidad existen dos métodos o pasos a seguir que son: por medio de métodos analíticos y simulación; los métodos analíticos eran usados principalmente para una toma rápida de decisiones en la planificación y diseño de pequeños sistemas; en la simulación se toma en cuenta todo tipo de sistema por medio de un proceso real y además se tiene un comportamiento aleatorio del sistema en estudio. Las técnicas de simulación conllevan a la operación, planificación y además del diseño del sistema considerando eventos aleatorios, variaciones en la carga, fuentes de energía y diferentes ambientes de operación.

Los dispositivos y todo el conjunto de componente que conforman un SEP se ven siempre involucrados en eventos de falla lo que provoca en el peor de los casos el colapso de grandes sistemas, por medio de índices se puede comprobar la calidad del servicio eléctrico suministrado. Para ello existen técnicas de evaluación de confiabilidad que son guiadas a poder determinar el comportamiento futuro de una red o sistema con

métodos de análisis y de simulación estocásticos. En los métodos estocásticos el más común es el de Monte Carlo y en los métodos de análisis se tiene en consideración los procesos de Markov [4] .

En el método de Monte Carlo se simula un gran número de eventos o situaciones que se generan de manera aleatoria que sirven para determinar los índices de confiabilidad por medio de distribuciones probabilísticas de dichos eventos. El método de Montecarlo es tomado como referencia para realizar el estudio de confiabilidad de un sistema en particular. Los eventos de Montecarlo se pueden generar por medio de computadoras cuánticas que simulan la aleatoriedad de cualquier evento por medio de probabilidades. Las computadoras cuánticas hoy en día son un gran aporte a la comunidad científica, por su gran capacidad de realizar múltiples tareas y se pueden obtener grandes resultados sin necesidad de mucho esfuerzo computacional [5].

1.1 DESCRIPCIÓN DEL PROBLEMA

Los sistemas de generación deben ser capaces de cubrir toda la demanda de una región o zona a la cual estén conectados y sobre todo de brindar una buena calidad de servicio eléctrico desde aguas arriba de la red teniendo en cuenta que cualquier tipo de falla que ocurra en el sistema repercutirá en interrupciones causando afectaciones a los usuarios finales conectados a la red [6].

Existen diferentes estudios de confiabilidad a los niveles jerárquicos por medio de técnicas e índices para así asegurar una mejor solución a los problemas que ocurren dentro de un SEP [7]. Uno de los métodos más comunes para realizar estudios de confiabilidad es el método estadístico de Monte Carlo que se debe emplear de acorde al tipo de estudio que se quiera efectuar, en este método se toma el diagnóstico de los índices de confiabilidad de acuerdo al comportamiento de las posibles fallas que puedan ocurrir en el sistema.

Hoy en día los investigadores de todo el mundo están buscando nuevas estrategias para poder reducir la carga computacional que se emplean con los diferentes modelos estadísticos, numéricos etc. Bajo esta necesidad surge la implementación de la computación cuántica como una potencial solución.

Toda la información generada por medio de computadoras clásicas es compilada mediante un conjunto de series de números binarios entre 0 y 1, estos bits son los pilares fundamentales de las computadoras. Al implementar computadoras

cuánticas se va a tener los qubits o también conocidos como bits cuánticos, a diferencia de los bits normales, los qubits no son binarios, es decir los qubits pueden estar el cualesquiera de los dos estados ya sea 0 o 1, o a su vez se puede encontrar en un estado de superposición y puede tomar simultáneamente 0,1. La particularidad de poder tomar simultáneamente dos valores es lo que hace a las computadoras cuánticas totalmente aleatorias [5].

Las diferentes empresas como Corporación Nacional de Electricidad (CNEL) o empresas dedicadas al estudio de confiabilidad de sistemas de potencia pueden implementar el uso de computadoras cuánticas para así poder disminuir la carga computacional que requiera la implementación de modelos matemáticos o estadísticos.

1.2 JUSTIFICACIÓN DEL PROBLEMA

Para brindar una buena calidad de energía y asegurar un óptimo funcionamiento de un SEP, se realiza un conjunto de análisis y estudios de confiabilidad para mitigar los posibles eventos de falla que pueden ocurrir en la red a su vez tratar de predecir con qué tipo de incurrencias pueden ocurrir estos eventos.

Por medio de la evaluación de confiabilidad probabilística se toma como referencia el método Monte Carlo; implementando qubits por medio de computadoras cuánticas. El método de Monte Carlo tradicional conlleva un gran número de experimentos para poder determinar cada uno de los índices y así asegurar un estudio óptimo [8]. Por otro lado, comúnmente los estudios de confiabilidad se los realizaba por medio de criterios deterministas conduciendo a grandes procesos de expansión, este tipo de criterios no son capaces de garantizar de manera equitativa el rendimiento en un sistema; dado que no pueden realizar un comportamiento totalmente aleatorio de los componentes ni de su operación. Por medio de los enfoques probabilísticos, de manera natural se puede obtener situaciones o eventos de manera aleatoria, por ejemplo, teniendo indisponibilidad de componentes o de variaciones en la carga. Todos estos efectos van a depender de manera proporcional a la probabilidad de ocurrencia, siendo posible determinar criterios o medidas de incurrencias de fallas.

Mediante el uso de computadoras cuánticas se pueden obtener eventos aleatorios, amenorando la carga computacional para el procesamiento de información mediante la implementación de algoritmos cuánticos, en donde cualquier modelo

de computación se puede simular mediante la máquina probabilística de Turing teniendo acceso a la aleatoriedad [5]

1.3 OBJETIVOS

1.3.1 Objetivo General

Proponer un modelo de programación cuántica para un estudio de confiabilidad a sistemas de nivel jerárquico uno (1) para la disminución de la carga computacional que conlleva la aplicación de modelos estadísticos.

1.3.2 Objetivos Específicos

- Implementar el uso de computadoras cuánticas o simuladores para la estimación de eventos de falla aleatorios.
- Realizar el estudio de confiabilidad a sistemas de generación de nivel HNI para el análisis de la operación del sistema en estudio.
- Cuantificar los índices de confiabilidad LOLE, LOLP, XLOL, EENS, una vez obtenida la simulación de eventos aleatorios por medio de programación cuántica.

1.4 MARCO TEÓRICO

En esta sección se detalla con precisión los conceptos fundamentales para el entendimiento de los términos y lenguaje técnico aplicado en el desarrollo del documento. Esta sección pretende explorar que es la confiabilidad, los tipos y su aplicación en los sistemas eléctricos de potencia. De igual forma introducir al lector sobre la programación cuántica, la unidad fundamental, el qubit, y las herramientas necesarias para construir circuitos cuánticos y aplicarlos en los estudios de confiabilidad.

1.4.1 Confiabilidad

El termino confiabilidad tiene su origen después de la primera guerra mundial en el sector de la aviación norteamericana. Las primeras experiencias con esta estrategia se centraron en medir la probabilidad de accidentes aéreos producidos por motores. En esta área se la definió como la probabilidad de un componente, equipo, red o sistema para que el mismo pueda llegar a

desempeñar su función durante un intervalo de tiempo ante condiciones de operación normales [9].

Con el pasar de los años se aplicó a distintos ámbitos llegando así al sector eléctrico. En el sector eléctrico el estudio de confiabilidad busca como objetivo principal garantizar la proveer del servicio eléctrico de forma correcta y con calidad. La institución norteamericana, North American Electric Reliability Council (NERC) clasifica la confiabilidad en dos grupos funcionales: seguridad y la adecuación.

En lo que respecta a la seguridad se involucran aspectos como la capacidad de respuesta del sistema ante fenómenos transitorios o disturbios como cortocircuitos o pérdida de elementos de la red. Por otro lado, la confiabilidad basada en adecuación considera aspectos como la facilidad y capacidad del sistema para satisfacer la demanda ante las limitantes operacionales del sistema, esta última se divide en tres distintos niveles jerárquicos, cada uno relacionados con los niveles funcionales de generación, transmisión y distribución que constituyen el sistema eléctrico de potencia [10].

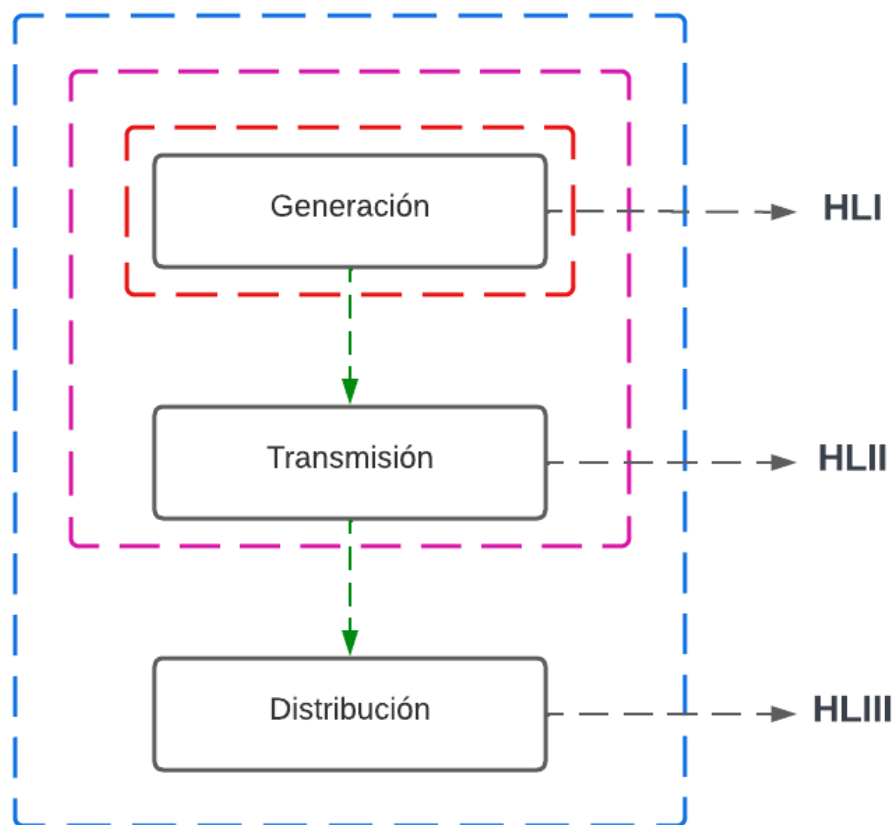


Figura 1-1 Clasificación en zonas funcionales y niveles jerárquicos

1.4.2 Sistemas eléctricos de potencia

Los sistemas eléctricos de potencia son un conjunto de elementos y equipos encargados de suministrar energía a distintos puntos de un área, región o país. Todo este conjunto de se encarga de llevar a cabo distintas funciones las cuales se clasifican en zonas funcionales y se detallan a continuación [11]:

- **Zona funcional de generación**

La zona funcional de generación tiene en su infraestructura equipos e instalaciones encargadas de generar energía eléctrica, la misma que puede ser producida a partir de fuentes convencionales o no convencionales.

- **Zona funcional de transmisión**

La zona funcional de transmisión está constituida por los elementos y estructuras que posibilitan la transportación de energía desde los puntos de generación hasta los puntos de consumo o transformación.

- **Zona funcional de distribución**

La zona funcional de distribución de forma similar al de transmisión presenta elementos y estructuras que transportan la energía a los usuarios finales que suelen ser pequeños consumidores.

1.4.3 Niveles Jerárquicos

Los estudios de confiabilidad aplicados a sistemas de potencia son una herramienta importante para conocer y predecir el estado del sistema, ayudando a garantizar una entrega del servicio continua. Estos estudios se clasifican en tres grandes grupos de acuerdo a la extensión o el área de aplicación en los sistemas de potencia los cuales son:

- **Nivel Jerárquico I**

El nivel jerárquico I tiene como fin realizar estudios a nivel de generación para determinar las facilidades del sistema para suplir la demanda total requerida y a su vez asegurar la capacidad necesaria que permitan llevar a cabo estrategias de mantenimiento a nivel de generación, en este nivel se

omiten aspectos como las limitantes de la red de transmisión- El modelo de red se presenta en la Figura 1-1.

- **Nivel Jerárquico II**

El nivel jerárquico II tiene como objetivo evaluar el conjunto generación transmisión y su habilidad para satisfacer la demanda requerida en los nodos de carga. Los estudios aplicados en este nivel pueden tener dos enfoques. El primero de adecuación. Donde se evalúa el sistema presente y sistemas planificados a futuro a nivel de generación y transmisión. El segundo, enfocado a la seguridad, valora la robustez del sistema ante perturbaciones o fallas.

- **Nivel Jerárquico III**

El nivel jerárquico III se evalúa la adecuación en los puntos de carga de cada uno de los circuitos principales. Es común subdividir el esquema de distribución en grupos por sectores debido a la gran extensión de ellos. De acorde a la topología de la red el método de estudio difiere para sistemas de tipo radial y para tipo mallado.

1.4.4 Índices de confiabilidad

Los índices básicos de confiabilidad en un estudio general se detallan a continuación:

- Frecuencia de falla: Expresado en [fallas/año] determina la proporción de fallas de un elemento o componente del sistema de estudio en un tiempo determinado.

$$\lambda = n_f/T \tag{1.1}$$

- Duración media de falla: Dado en [horas/falla] expresa el tiempo de indisponibilidad de un elemento o componente del sistema de estudio en ante la ocurrencia de una falla.

$$r = \frac{\sum_{i=1}^{n_f} ttr_i}{n_f} \tag{1.2}$$

Los índices de confiabilidad empleados a nivel de generación para cuantificar la calidad de servicio eléctrico que se puede presentar en un sistema son:

- LOLE:

Loss of Load Expectation (LOLE), se define como el promedio de días u hora en un tiempo de estudio donde la demanda pico sobrepasa la energía disponible de generación.

$$LOLE = \sum_{i \in S} p_i T \quad (1.3)$$

- LOLD

Loss of Load Duration (LOLD), este índice, similar al LOLE representa la tasa de duración de deficiencias o disturbios en la red.

$$LOLD = \frac{LOLE}{LOLF} \quad (1.4)$$

- EENS

Expected Energy Not Supply (EENS), representa la cantidad de energía que no ha sido despachada debido a que la demanda supera la energía generada disponible.

$$EENS = \sum_{i \in S} 8760 C_i p_i \quad (1.5)$$

1.4.5 Computadoras Cuánticas

Desde hace algunos años las computadoras han ido desarrollándose de manera en que su rendimiento es cada vez mayor dependiendo del tipo de uso que se vaya a emplear. Los circuitos de hoy en día contiene microcomponentes acompañados de transistores que hace se multiplique en gran magnitud su velocidad para el procesamiento de datos.

Hoy en día existen los ordenadores cuánticos, que es una de las tecnologías en donde se aprovecha en parte de la mecánica cuántica para poder resolver grandes problemas en los que una computadora normal con procesadores comunes no podría resolver [12]. Mediante International Business Machines (IBM), es posible la implementación de un hardware cuántico y que está disponible para miles de usuarios, el uso de estas supercomputadoras es

para el desarrollo de grandes problemas en los que intervienen un gran número de variables.

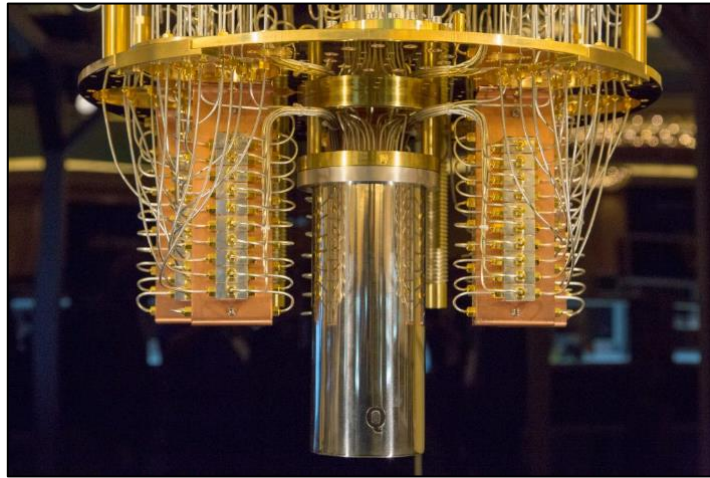


Figura 1-2 Primera computadora cuántica comercial [13]

Se conoce que las computadoras clásicas efectúan el uso de dígitos binarios también llamados bits, pero las computadoras cuánticas usan los qubits. Los qubits a diferencia de los bits clásicos de 1 y 0; toman ambos valores, es decir, son ambos valores a la vez o solo uno de ellos, esto se puede dar mediante la superposición de ambos estados a la que se atribuye a la mecánica cuántica, en donde se puede apreciar las propiedades de un objeto en particular que puede tener más de un valor al mismo tiempo [14]. La superposición en mecánica cuántica en donde existe o puede existir un registro de N qubits se pueda representar de la siguiente manera 2^N , tomando un conjunto de valores simultáneos donde todos estos registros producen 2^N resultados posibles, además de obtener un paralelismo exponencial, para obtener un resultado del estado de superposición o de cualquier qubit es necesario realizar la medición para así poder observar u obtener un resultado.

1.4.6 Lenguajes de programación

Los lenguajes de programación son un conjunto de instrucciones tanto de reglas, símbolos y demás caracteres que pueden permitir a una persona expresar de una manera lógica y detallada el procesamiento de un conjunto de datos y resultados que son parte del proceso de la programación, cada

lenguaje de programación posee un tipo de estructura propia que los diferencia entre sí. El proceso por el cual comienza a escritura de un programa es en base a algoritmos que permite la comunicación entre la persona y el computador [15].

Quantum Computation Lenguaje (QCL) es el primer lenguaje de programación cuántica que fue creado por Bernhar Omer, este tipo de lenguaje de programación se asemeja a los lenguajes de programación Pascal y C, además, contiene características clásicas como lo son con variables, condicionales, ciclos etc. También soporta la construcción de funciones y por ende trae en si funciones ya pre creadas, y cuenta con operadores aritméticos.

1.4.7 Bit Cuántico

El qubit es la unidad referencial de medida con respecto a la información cuántica. Se parte desde el concepto fundamental de la computación e información clásica. Es así como comúnmente un bit clásico puede tener un solo estado ya sea 1 o 0, el qubit puede tomar cualesquiera de los dos estados posibles y puede ser los dos al mismo tiempo $|0\rangle$ y $|1\rangle$. La notación $| \rangle$ es conocida como la notación de Dirac, es la notación estándar para los estados de la mecánica cuántica [16]. Entre los qubits también se pueden formar combinaciones lineales de estados que se conoce como estado de superposición:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (1.6)$$

Los números o expresiones α y β son números complejos, pero para fines de estudio u otros propósitos estos valores pueden ser vistos como número enteros. El estado del qubit en el espacio vectorial bidimensional es un vector complejo con los estados representativos de $|0\rangle$ y $|1\rangle$ que son estados bases computacionales y ambos forman un base ortonormal para ese espacio vectorial. Al momento de que se realiza la medición de un qubit para poder determinar su valor, se obtiene cero con probabilidad de $|\alpha|^2$ o se puede obtener el valor de 1 con probabilidad de $|\beta|^2$. Ya que ambos estados de manera natural tienen que formar un base ortonormal, se tiene

que cumplir que $|\alpha|^2 + |\beta|^2 = 1$, dado que ambas probabilidades deben sumar 1 [14].

Existe una manera más útil por la cual se puede representar un qubit y poder entender la esfera de Bloch que se presenta en la Figura 1-3.

Partiendo de $|\alpha|^2 + |\beta|^2 = 1$ podemos escribir la ecuación (1.6) como:

$$|\psi\rangle = e^{i\gamma} \left(\cos\frac{\theta}{2} |0\rangle + e^{i\varphi} \sin\frac{\theta}{2} |1\rangle \right) \quad (1.7)$$

Donde φ , Θ , y γ son números reales, el término $e^{i\gamma}$ se puede despreciar ya que no realiza mayor cambio o relevancia al momento de realizar alguna otra expresión y se puede reescribir como se muestra a continuación [14].

$$|\psi\rangle = \cos\frac{\theta}{2} |0\rangle + e^{i\varphi} \sin\frac{\theta}{2} |1\rangle \quad (1.8)$$

Los números φ , Θ constan de un punto en la esfera de Bloch Figura 1-3, en que se muestra un medio para poder ver o interpretar el estado de un solo qubit.

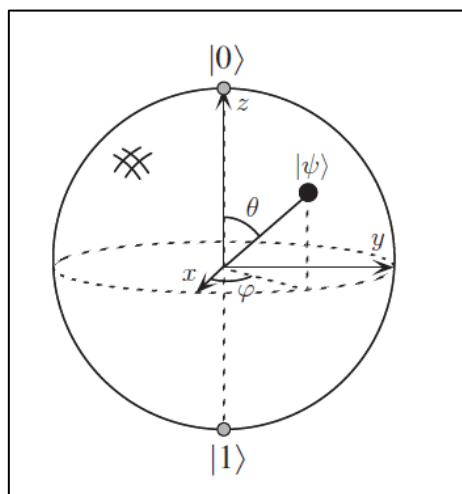


Figura 1-3 Esfera de Bloch Representación de un qubit [14]

Un qubit también puede ser representado de forma matricial mediante:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (1.9)$$

1.4.8 Puertas cuánticas

Mediante las puertas cuánticas se pueden realizar operaciones o cambios de estados entre los qubits, así como en la programación de los bits clásicos, los qubits también trabajan con puertas lógicas. A

continuación, se enlistan las compuertas más comunes usadas en algoritmos cuánticos:

- La puerta X

La compuerta X viene representada por medio de la matriz de Pauli-X:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0| \quad (1.10)$$

Para poder notar el efecto que tiene la compuerta en un qubit, simplemente basta con multiplicar el vector de los estados del qubit con la compuerta:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \quad (1.11)$$

- Puerta Y y Z

Al igual que la puerta X, las matrices de Pauli también actúan en las puertas Y y Z:

$$Y = \begin{bmatrix} 0 & -Y_0 \\ Y_0 & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (1.12)$$

$$Y = -Y_0|0\rangle\langle 1| + Y_0|1\rangle\langle 0| \quad Z = |0\rangle\langle 0| - |1\rangle\langle 1| \quad (1.13)$$

- Puerta Hadamard

La puerta Hadamard también conocida como puerta H es una de las principales compuertas cuánticas; su función es crear una superposición de los estados cuánticos para un qubit por ello se tiene la siguiente matriz:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1.14)$$

La puerta Hadamard se puede considerar como una rotación en el eje x y z o como una transformación del qubit entre ambas bases x y z:

$$\begin{aligned} H|0\rangle &= |+\rangle \\ H|1\rangle &= |-\rangle \end{aligned} \quad (1.15)$$

$$\begin{aligned} H|0\rangle &= |+\rangle \\ H|1\rangle &= |-\rangle \end{aligned} \tag{1.16}$$

Además, se puede comprobar que mediante la compuerta Hadamard llegamos a:

$$H^2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} * \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = I \tag{1.17}$$

- Compuerta CNOT para 2 qubits

La compuerta Cnot es también llamada como la compuerta de negación controlada. Este tipo de compuerta es común cuando se implementan dos qubits, lo que hace este tipo de compuerta es cambiar el valor del segundo qubit siempre y cuando el primer qubit sea igual a 1.

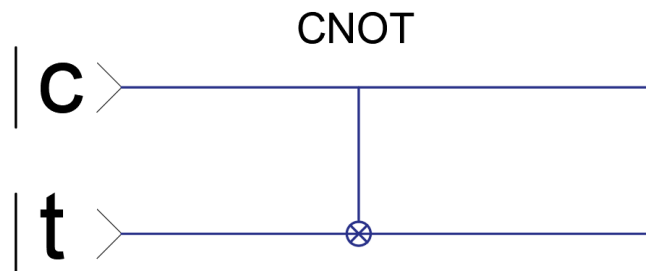


Figura 1-4 Compuerta CNOT

Teniendo la siguiente expresión de acorde a la Figura 1-4:

$$\begin{aligned} |c\rangle &= \alpha |0\rangle + \beta |1\rangle \\ |t\rangle &= \alpha' |0\rangle + \beta' |1\rangle \end{aligned} \tag{1.18}$$

El comportamiento de la compuerta CNOT se lo puede apreciar en la siguiente tabla:

Tabla 1-1 Compuerta CNOT

Entradas	 c>	 t>	Salidas
00>	0	0	00>
01>	0	1	01>
10>	1	0	10>
11>	1	1	11>

1.4.9 Mediciones en diferentes bases a la computacional

En las mediciones cuánticas de un solo qubit teniendo como estado $|\alpha\rangle + |\beta\rangle$ puede darse un resultado de 0 o 1 y dejando al qubit en unos de los estados correspondientes de $|0\rangle$ o $|1\rangle$; con sus respectivas probabilidades de $|\alpha|^2$ y $|\beta|^2$. Hay que tener en cuenta que los estados de $|0\rangle$ y $|1\rangle$ representan una de las muchas opciones de estados base de un qubit. En sí, la medición de un qubit representa la forma de transformar la información cuántica de un sistema y llevarla a un tipo de información clásica, uno de los principios centrales de la mecánica cuántica es que los resultados de la medición de un qubit son probabilísticos.

1.4.10 Circuitos Cuánticos

A menudo los algoritmos cuánticos se representan por medio de circuitos esquemáticos. En el circuito, los qubits se encuentran representados en líneas horizontales, en la misma línea se dibujan las puertas sobre las que actúan cada uno de los qubits. La secuencia en que se van colocando o dibujando las compuertas para realizar cualquier cambio de estado se lo hace de izquierda a derecha, con esto el estado inicial de cada uno de los qubits que forman el circuito se indica al inicio de las líneas. Cuando tenemos una expresión matemática para implementar en el circuito, las puertas se logran describir de derecha a izquierda dado el orden en que se aplican [17]. A continuación, se muestra un ejemplo de un circuito cuántico de dos qubits entrelazados que se denomina el estado de Bell de $|00\rangle$.

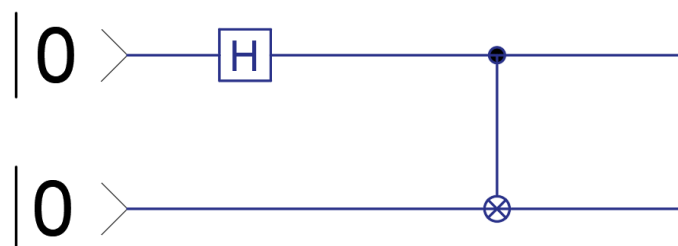


Figura 1-5 Circuito cuántico con dos qubits entrelazados

El circuito de la Figura 1-5 se puede reescribir mediante la siguiente ecuación:

$$CNOT_{12}(H \otimes I)|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad (1.19)$$

Como se puede observar en la Figura 1-5 al primer qubit se le aplica una compuerta cuántica que es la Hadamard este qubit está en estado $|0\rangle$ una vez que la compuerta hace su trabajo, la compuerta $CNOT_{12}$ actúa en ambos qubits, el primer punto en negrita significa que este qubit es el qubit de control para la compuerta $CNOT$, donde se encuentra el símbolo \oplus es el segundo qubit e implica que este qubit es el objetivo de control de la puerta NOT (Control realizado por el estado del qubit 1). Y el resultado es el que se muestra en la ecuación (1.19) una vez realizada la medición. La medición de un qubit también puede ser indicada en un circuito y significa que el qubit debe ser medido en la base computacional.

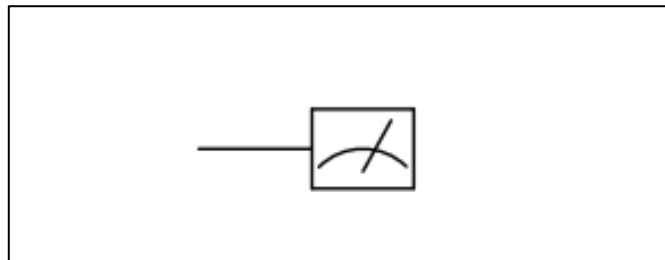


Figura 1-6 Bloque de medición en un qubit

1.4.11 Método de simulación de Monte Carlo

En los métodos de simulación de Monte Carlo se utilizan un conjunto de muestreo aleatorio para poder estimar cantidades numéricas que conlleven mayor complejidad y que, además, no sean sencillas de calcular de manera determinista. Uno de los ejemplos de uso de este método es en la física estadística para mezclar cadenas de Markov y poder calcular funciones de partición de forma aproximada. El método de Monte Carlo abarca muchas aplicaciones en los que se involucra complejos cálculos matemáticos, análisis estadísticos, estudios de confiabilidad, procesos estocásticos, etc. Uno de los algoritmos clásicos de Monte Carlo implica estimar el valor de salida esperado x de un algoritmo aleatorio A ; el algoritmo debe ser por

medio de n muestras a cada una de las correspondientes salidas de una ejecución independiente de A , para luego generar un promedio de \bar{x} que se aproxime a x . Al suponer que la varianza de la variable A que es aleatoria toma como máximo σ^2 por lo que la probabilidad de que el valor de salida del estimador esté lejos del valor correcto, se lo puede acotar utilizando la desigualdad de Chebyshev.

$$P[|\bar{x} - x| \geq \epsilon] \leq \frac{\sigma^2}{n\epsilon^2} \quad (1.20)$$

Para ello es suficiente tomar $n = O\left(\frac{\sigma^2}{\epsilon^2}\right)$ para estimar x hasta el error aditivo ϵ , con aproximadamente 99% de probabilidad de éxito [18].

La simulación de Montecarlo funciona haciendo un muestro de estado del sistema basado en la distribución de probabilidad conjunta. Es decir, sus componentes pueden considerarse independientes, cada uno de los estados del sistema se puede determinar mediante el muestreo de las funciones de probabilidad que consta en cada uno de los componentes.

En comparación con las habituales técnicas analíticas, el método de Monte Carlo se usa principalmente para la simulación de redes complejas, en donde cada uno de sus componentes deben estar existentes en la red de estudio. Existen dos técnicas básicas utilizadas en el método de Monte Carlo que son el método secuencial (SMC) y el método no secuencial (NSMC) [19].

- Método de Montecarlo Secuencial

En el método SMC, la simulación sigue cronológicamente con el cambio de estado de uno o de varios componentes de un sistema. Los estados del sistema son registrados en una base en donde se pueden calcular los índices de confiabilidad luego de realizar la simulación durante un tiempo muy prolongado. El modelo secuencial replica el mismo proceso de evolución preservando las características de los eventos cronológicos, dependiente y causales.

- Método de Montecarlo No Secuencial

En el método de NSMC, los estados son muestreados del espacio proporcional al de sus probabilidades. El tipo de enfoque del NSMC es la simulación de estados aleatorios a lo largo de su vida escogiendo estados

al azar, en esta simulación la variable o concepto de tiempo no aparece y el periodo en el que se estima el servicio del sistema debe ser muy alto para así obtener promedios estadísticamente confiables. Los resultados o estados del sistema deben ser simulados por separado y dependerán de la combinación de los estados de los componentes [20].

1.4.12 Método de Monte Carlo en estudios de confiabilidad

Uno de los puntos fundamentales para la evaluación de confiabilidad por el método de Monte Carlo es tener una expectativa o rasgo matemático de un índice previamente establecido, se puede trabajar por medio de la indisponibilidad Q dado una probabilidad de falla en un sistema u_i es una variable que puede tomar dos estados bajo la siguiente regla de correspondencia:

$$\begin{aligned} u_i &= 1 ; \text{ Sistema en funcionamiento} \\ u_i &= 0 ; \text{ Sistema no está en funcionamiento} \end{aligned} \quad (1.21)$$

$$U_i = \left\{ \begin{array}{ll} 1, & \text{Operando} \\ 0, & \text{No operando} \end{array} \right\} \quad (1.22)$$

La simulación de Monte Carlo es aplicada a los sistemas de potencia para poder determinar cada uno de los estados de los componentes que lo conforman, se trata de empezar generando un número aleatorio entre uno y cero para cada componente tal y como se muestra en (1.21).

Cada uno de los estados del sistema se pueden expresar mediante $U_{exp} = \{u_1, u_2, \dots, u_{NC}\}$. Luego cada uno de los estados del sistema se pueden representar mediante:

$$EXS_{exp} = EXF(U_{exp})P(U_{exp}) \quad (1.23)$$

EXF representa a la función de los experimentos y P la función de eventos de probabilidades. Los cálculos se repiten para un número (NE) de experimentos, donde se registra E_{exp} . Y dicho procedimiento termina en la convergencia de la determinación del valor medio de la muestra mediante:

$$\widehat{E\bar{X}S} = \frac{1}{NE} \sum_{exp=1}^{NE} EXS_{exp} \quad (1.24)$$

Algoritmo 1 Método de Monte Carlo

- 1: **Procedimiento** de Monte Carlo
- 2: **for** $exp = 1$ al total de experimentos (NE) **do**
- 3: *aleatorizar los estados para todos los componentes* : $U_{expect} = (U_1, U_2, \dots, U_{NC})$
- 4: *definir EXF y P*
- 5: *evaluar EXF y P en U_{exp}*
- 6: *determinar EXS_{exp} usando(1.23)*
- 7: **end for**
- 8: *Determinar $\overline{E\bar{X}S}$ usando(1.24)*

Figura 1-7 Algoritmo de Monte Carlo [6]

La indisponibilidad del sistema puede estar dada por:

$$\bar{Q} = \frac{1}{N} \sum_{i=1}^N u_i \quad (1.25)$$

Es N el número total de muestras en un sistema.

En retrospectiva, el método de Monte Carlo tiende a presentar un mejor resultado a diferencia de los métodos analíticos bajo los siguientes escenarios de estudio:

- Estudios dependientes del tiempo.
- Se modela ciclo de trabajo de las unidades.
- Las distribuciones de carácter no exponencial de los componentes son consideradas.
- Se necesitan de distribuciones de índices de confiabilidad

La evaluación de adecuación en los sistemas de generación se divide en problemas de una área o múltiples áreas.

CAPÍTULO 2

2. METODOLOGÍA PARA LA CONSTRUCCIÓN DEL ALGORITMO CUÁNTICO APLICADO A ESTUDIOS DE CONFIABILIDAD

En este capítulo se explica el proceso para la construcción del algoritmo tradicional de Monte Carlo No Secuencial empleando una computadora clásica y el algoritmo Híbrido mediante un acelerador cuántico, haciendo uso del lenguaje de programación Python™ teniendo acceso a un acelerador cuántico para la prueba de los algoritmos. Finalmente, los algoritmos desarrollados fueron aplicados en dos distintos casos de estudio de confiabilidad de nivel jerárquico I donde se utilizaron índices como EENS, LOLP, XLOL para validar la eficiencia de estos.

2.1 NSMC TRADICIONAL

El método de Monte Carlo No Secuencial, denominado NSMC tradicional en este documento; es un método iterativo y estocástico donde se realiza una serie de experimentos donde se obtiene el comportamiento de una muestra bajo ciertas restricciones y condiciones de acuerdo con el caso de estudio.

El algoritmo inicia con la carga de una base de datos de un sistema de potencia sobre el cual se aplicó el estudio de nivel jerárquico I. Entre los datos necesarios se incluyen:

- La demanda total del sistema, fundamental para establecer los requerimientos del sistema y en conjunto con la generación evaluar los índices de confiabilidad de acorde al perfil de carga.
- Los parámetros de cada una unidad generadora que interviene en el caso de estudio, entre estos están la generación máxima, el tiempo medio de falla (MTTF) y el tiempo medio de reparación (MTTR). El MTTR da referencia al tiempo necesario para poder reparar un componente y ser capaz de devolverlo al estado de funcionamiento normal, mientras que el MTTF se refiere al tiempo en que un sistema está en funcionamiento entre periodos o cortes.

Estos datos son necesarios para pasar a la siguiente etapa del proceso y realizar los experimentos y el cálculo de los índices de confiabilidad.

En la siguiente parte se inicializan los datos previos a la ejecución de los experimentos. Se calcula el índice FOR, obteniendo también la disponibilidad A_i mediante [6]:

$$U_i = \frac{MTTF}{MTTF + MTTR} \quad (2.1)$$

$$A_i = 1 - \frac{MTTF}{MTTF + MTTR} \quad (2.2)$$

Parte también de la inicialización del algoritmo fue definir el número total de experimentos NE y conocer el número de generadores que son parte del sistema de estudio. Una vez que se establecieron todos los parámetros de entrada se generó un número aleatorio representativo de cada generador, para cada muestreo de intervalo en nuestro caso de estudio se definió un tiempo de análisis de 8736 horas para ambos algoritmos.

La elección del estado de los generadores en cada hora del experimento se realiza considerando el valor del FOR el cual fue comparado con un número aleatorio tomado para cada uno de los generadores existentes, y si este es mayor el componente pasa a estar en estado operativo, en caso de que ocurra lo contrario el componente pasa a estar en fuera de servicio [7].

$$u_i = \begin{cases} 0; FOR_i \leq x_i \\ 1; FOR_i > x_i \end{cases} \quad x_i \in X$$

u_i : Estado del sistema en funcionamiento

X : Variable aleatoria con distribución normal

(2.3)

El siguiente paso fue el calcular la generación de margen o reserva de energía, tomando la diferencia entre la generación disponible por hora y la demanda horaria. La suma de margen negativo es una de las áreas que se muestran en la Figura 2-1 que nos sirve para determinar la energía no suministrada (EENS), mientras que el periodo de tiempo que ocurre la falla o fenómeno se determina la duración de la pérdida de la carga (LOLE).

Los valores son almacenados durante cada iteración y finalmente se obtiene el promedio de estos. Los índices se pueden determinar de acorde a las siguientes ecuaciones [6]:

$$LOLE = \frac{1}{NE} \sum_{exp=1}^{NE} LLD_{exp} \quad (2.4)$$

$$LOLP = \frac{LOLE}{NE} \quad (2.5)$$

$$EENS_{exp} = Area_1 + Area_2 + Area_3 \dots \dots \quad (2.6)$$

$$XLOL = \frac{EENS_{exp}}{LOLE}; LOLE \neq 0 \quad (2.7)$$

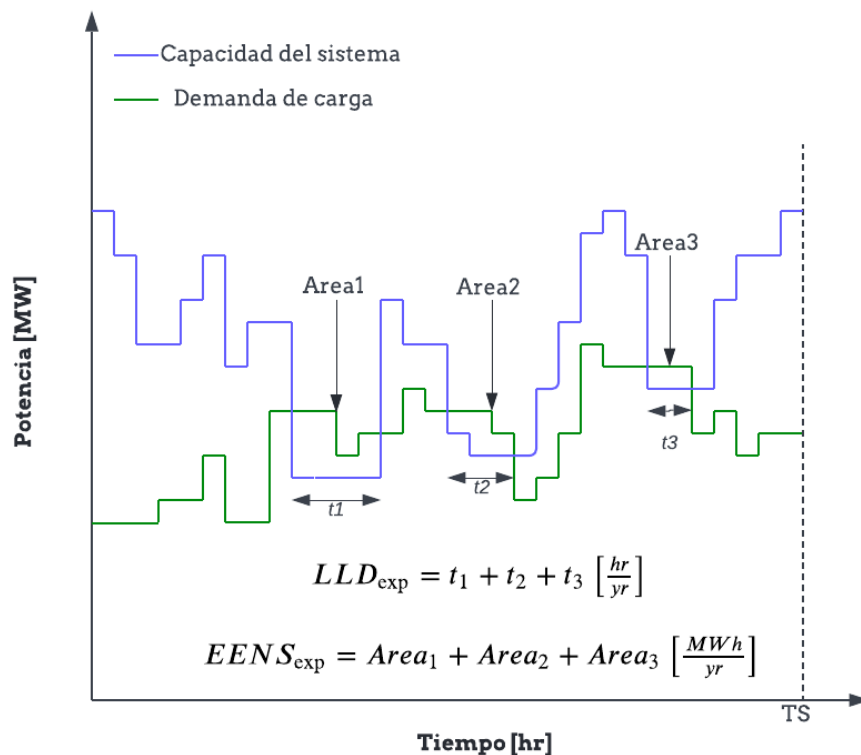


Figura 2-1 Modelo de margen de generación y demanda [6]

La siguiente figura presenta un diagrama de flujo que detalla la secuencia en el que fue implementado el algoritmo de NSMC tradicional.

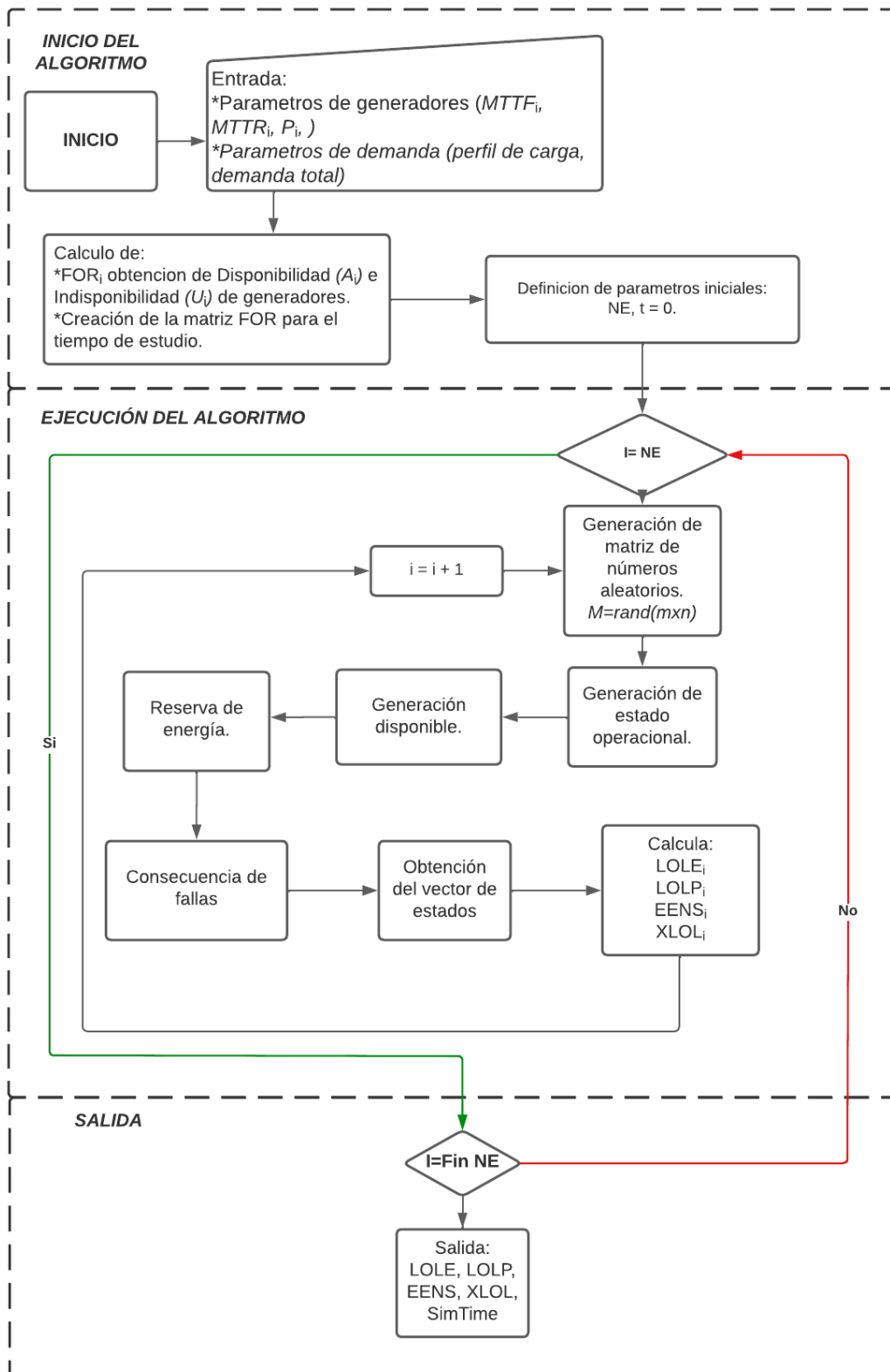


Figura 2-2 Diagrama de flujo de NSMC tradicional

2.2 METODO HIBRIDO – ALGORITMO CUANTICO

Para la construcción del algoritmo cuántico se tomó en cuenta un conjunto de procesos que partieron desde el concepto básico de computadoras, compuertas y mediciones cuánticas. Para nuestro caso en particular se implementó el uso de simuladores cuánticos que brinda las características similares al funcionamiento del computador cuántico. El uso de qubits en nuestro algoritmo cuántico estuvo delimitado por el número de generadores usados en el sistema, es decir, para cada generador fue necesario implementar un qubit individual.

El algoritmo fue implementado en el lenguaje de programación PythonTM debido a que por medio de este software se puede realizar la conexión con las computadoras y simuladores cuánticos por medio de librerías. El tipo de lenguaje de programación que se usó fue Q que se detalla en la sección 1.4.6.

Para la implementación del algoritmo y circuito cuántico se requirió el uso de la librería Qiskit.org, cuya programación cumple la secuencia de instrucciones al momento de programar.

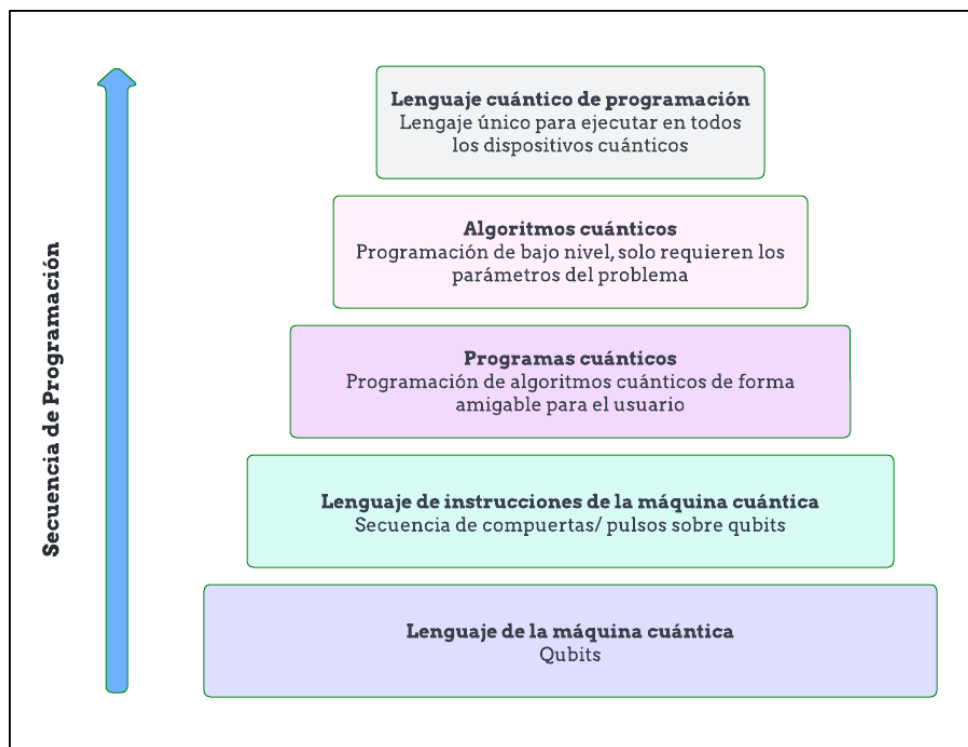


Figura 2-3 Secuencia de instrucciones de programación cuántica [21]

2.2.1 Lenguaje de programación Q

El lenguaje de programación Q se cuenta enlazado con un kit de desarrollo que se puede usar en conjunto con herramientas además de lenguajes comunes para el desarrollo de aplicaciones cuánticas pudiéndose usar diferentes entornos. Un programa con lenguaje de programación Q puede compilarse independientemente desde un host debidamente escrito en Python. Al momento de que se compila, se crea una sentencia de simulador cuántico. El simulador que se use implementa este lenguaje Q para la creación de qubits y realizar las diferentes tareas, luego los resultados y operaciones efectuadas son devueltas al programa. El aislamiento del código Q en cualquier simulador cuántico se asegura de que los algoritmos encaminen las leyes de la física cuántica para poder ser ejecutados en cualquier instancia en los computadores cuánticos.

A continuación, se muestra un esquema que sigue la secuencia en la que se implementó el algoritmo cuántico [22].

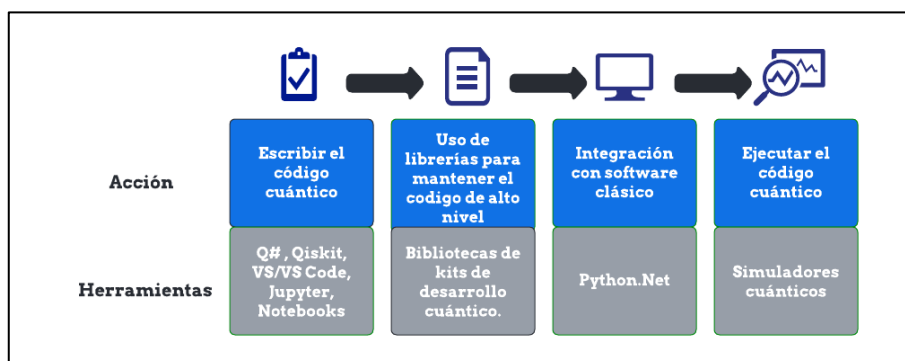


Figura 2-4 Instrucciones de secuencia de programación cuántica [21]

Simuladores cuánticos implementados

IBM Quantum posee además de computadoras cuánticas a los diferentes usuarios suscritos, dispone de simuladores de alto rendimiento para la creación y familiarización de algoritmos, los simuladores son capaces de realizar el comportamiento similar al de las computadoras cuánticas, lo cual lo hace bajo modelos realistas de ruido.

Cada uno de los simuladores disponibles por medio de la IBM Quantum tienen un límite de tiempo para procesar los trabajos alrededor de los 10.000 segundos, aproximadamente 2.75 horas.

A continuación, se detallan los siguientes simuladores que fueron implementados para la construcción del algoritmo híbrido para el estudio de confiabilidad.

Simulador Stabilizer

Es un tipo de simulador eficiente de circuitos Clifford. Capaz de simular operadores de ruido en caso de que sean implementadas las puertas de Clifford. Este simulador nos ofrece los siguientes parámetros [23]:

# Qubits disponibles	Modelado de ruido	Puertas/Instrucciones disponibles
5000	Solo Clifford	['cx', 'cy', 'cz', 'id', 'x', 'y', 'z', 'h', 's', 'sdg', 'sx', 'swap', 'delay', 'roerror']

Simulador Qasm

El simulador Qasm es de propósito general para la simulación de circuitos cuánticos de manera ideal o pueden estar sujetos a ruidos. La manera en que se produce la simulación va de acorde a los parámetros en entrada de los circuitos. El simulador nos ofrece los siguientes requerimientos:

# Qubits disponibles	Modelado de ruido	Puertas/Instrucciones disponibles
32	Si	['u1', 'u2', 'u3', 'u', 'p', 'r', 'rx', 'ry', 'rz', 'id', 'x', 'y', 'z', 'h', 's', 'sdg', 'sx', 't', 'tdg', 'swap', 'cx', 'cy', 'cz', 'csx', 'cp', 'cu1', 'cu2', 'cu3', 'rxx', 'ryy', 'rzz', 'rzx', 'ccx', 'cswap', 'mcx', 'mcy', 'mcz', 'mcsx', 'mcp', 'mcu1', 'mcu2', 'mcu3', 'mcrx', 'mcry', 'mcrz', 'mcr', 'mcswap', 'unitary', 'diagonal', 'multiplexer', 'initialize', 'kraus', 'roerror', 'delay']

Simulador State Vector

State vector simula un circuito cuántico capaz de calcular la función de onda del vector de estado del que se encuentra el qubit en función al tipo de compuertas implementadas, también es capaz de generar el modelado de ruido [15].

# Qubits disponibles	Modelado de ruido	Puertas/Instrucciones disponibles
5000	Solo Clifford	['u1', 'u2', 'u3', 'u', 'p', 'r', 'rx', 'ry', 'rz', 'id', 'x', 'y', 'z', 'h', 's', 'sdg', 'sx', 't', 'tdg', 'swap', 'cx', 'cy', 'cz', 'csx', 'cp', 'cu1', 'cu2', 'cu3', 'rxx', 'ryy', 'rzz', 'rzx', 'ccx', 'cswap', 'mcsx', 'mcy', 'mcsz', 'mcsx', 'mcp', 'mcsu1', 'mcsu2', 'mcsu3', 'mcrx', 'mcry', 'mcrz', 'mcr', 'mcswap', 'unitary', 'diagonal', 'multiplexer', 'initialize', 'kraus', 'roerror', 'delay']

2.2.2 Implementación del método de NSMC en el algoritmo cuántico

El algoritmo implementado se denomina “Algoritmo-Hibrido” debido a que este emplea una computadora clásica para operaciones básicas y un acelerador cuántico para la carga computacional de mayor relevancia como lo es la obtención de los datos aleatorios. El algoritmo está constituido en 3 etapas generales, la inicialización de datos, el procesamiento y la salida de datos, a continuación, se detallan cada una de estas partes.

Inicialización

En esta etapa se realiza la carga de datos del caso de estudio a analizar, considerando un sistema de nivel jerárquico I. Los datos necesarios son los de generación y de la carga o demanda. En el caso del generador se solicitan datos del generador como MTTF, MTTR, FOR, Potencia, por parte de la carga es necesaria una curva de demanda y la potencia máxima requerida. Subido los datos al sistema se proceden a calcular los porcentajes de disponibilidad e indisponibilidad empleando las ecuaciones 2.1 y 2.2, estos porcentajes son necesarios para la preparación y el tratamiento de los qubits.

La rotación del qubit necesaria para representar cada generador en el acelerador cuántico se calculó utilizando 2.8 donde A_i es el porcentaje de disponibilidad de las unidades de generación.

$$\theta_i = 2 \arcsin(A_i) \quad (2.8)$$

Siendo este ángulo el requerido para definir la rotación generada sobre el qubit a través de la compuerta Ry. Concluido este cálculo se da paso al proceso o ejecución del algoritmo.

Ejecución del algoritmo

Esta etapa se inició definiendo el número de experimentos NE, número de generadores NG, número de disparos 'SHOTS', donde este último valor dependerá de la duración del perfil de carga.

Posterior a ello se procedió con la ejecución del código. Donde por cada experimento se realiza la preparación de qubit en estado cero y este a través de la compuerta RY modifica su probabilidad para el estado 0 y 1.

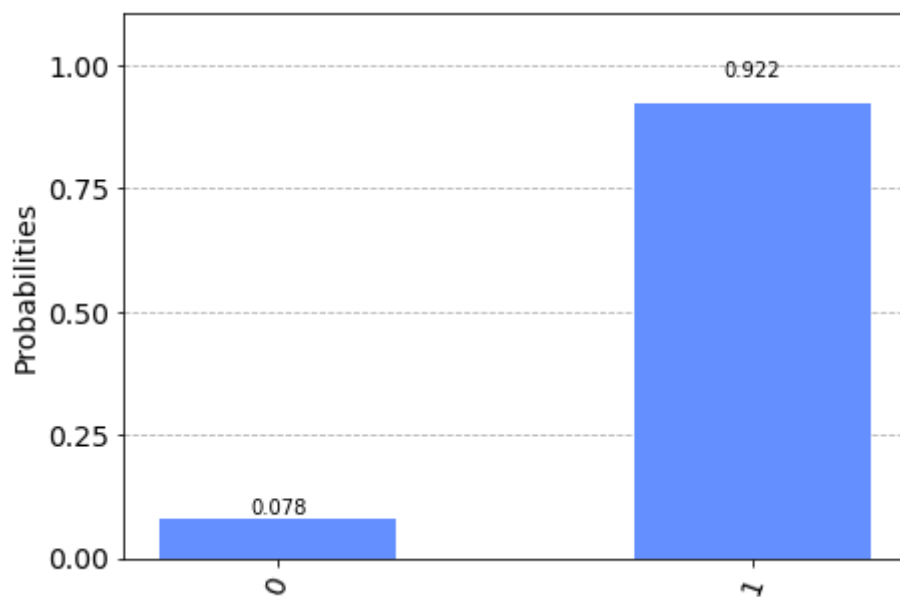


Figura 2-5 Distribución de probabilidad de funcionamiento de un generador

Una vez ajustado el qubit se realiza la simulación para la cantidad de disparos definidos y se realizan las respectivas mediciones.

Este proceso se realizó para cada generador por cada experimento lo cual ayuda a construir la matriz de estados y con ella obtener los índices LOLP, EENS, XLOL y LOLE dados en las ecuaciones (2.4), (2.5), (2.6) y (2.7).

Salida de datos

Una vez finalizado los experimentos se pasa a la etapa final donde se calcularon los índices del estudio de confiabilidad planteado para su análisis, en la siguiente figura se presenta un diagrama de flujo de la ejecución del algoritmo híbrido.

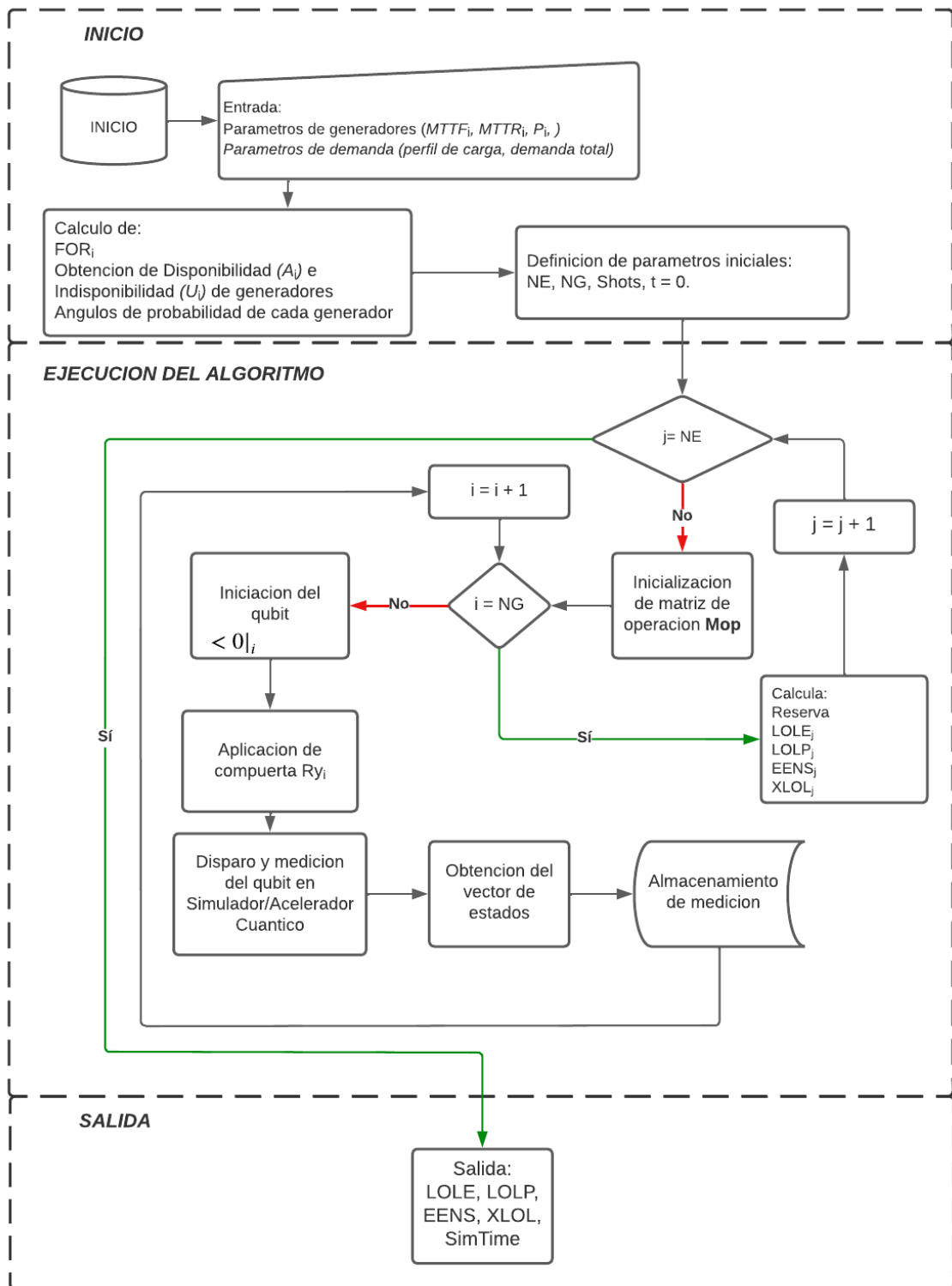


Figura 2-6 Diagrama de flujo del algoritmo hibrido

Algoritmo 2 NSMC Híbrido

```
1: Procedimiento NSMC Híbrido
2: Import librarys
3: Carga de datos: Curva de demanda
4: Carga de datos del sistema IEEE
5: Cálculo de índice FOR mediante (2.1) y (2.2)
6: Definir  $\theta$  para cada uno de los generadores mediante (2.8)
7: Inicio:  $NE = 100$ 
8: for  $i = 1$  al total de experimentos ( $NE$ ) do
9:    $Def.\#Gen$ 
10:  for  $j = 1$  al total de # Gen ( $Ng$ ) do
11:     $Uso\ Aer\ Simulator$ 
12:     $Def\ QuantumRegister\ y\ ClassicalRegister$ 
13:     $Creación\ de\ matriz\ de\ números\ aleatorios\ MOP$ 
14:     $Gen\ disponible = MOP * DataGenPot$ 
15:     $Reserva\ de\ energía\ para\ suplir\ la\ demanda; ReservePot.$ 
16:     $Cálculo\ de\ índices\ de\ confiabilidad\ mediante\ (2.4), (2.5), (2.6), (2.7)$ 
17:     $Almacenamiento\ de\ los\ índices\ LOLE, LOLP, EENS, XLOL$ 
18:  end for
19:   $Prom : LOLE, LOLP, EENS, XLOL$ 
20: end for
Salida:  $LOLE, LOLP, EENS, XLOL$ 
```

2.3 MÉTRICAS PARA EVALUACIÓN DE ALGORITMOS

Finalmente; una vez que se culmina la ejecución de los algoritmos en el estudio fue necesario emplear métodos estadísticos para la evaluación de los resultados y el rendimiento de los algoritmos en términos de explotación: exactitud, precisión y en términos de exploración mediante: velocidad de búsqueda y aceleración [24].

2.3.1 Exactitud y precisión

La explotación hace referencia a la capacidad de búsqueda alrededor de un área local o región de estudio. Se puede cuantificar dejando en función de dos métricas: exactitud (δ) y precisión (φ). El término (δ) está definido mediante el valor absoluto de la diferencia entre el valor promedio y el valor verdadero de referencia de la cantidad de estudio, se trata de las cercanías de las medidas al valor óptimo o verdadero. Mientras que el término precisión indica la cercanía que existe entre sí de las unidades. Cada uno de estos términos se pueden obtener matemáticamente usando (x_{op}) , (\bar{x}) y la desviación estándar (σ), cada uno de estos términos se indican en las siguientes ecuaciones [25]:

$$\delta = |x_{op} - \bar{x}| \quad (2.9)$$

$$\varphi = |\sigma - \bar{x}| \quad (2.10)$$

2.3.2 Velocidad y aceleración

La exploración es definida como la capacidad de poder examinar todas las áreas del espacio de búsqueda de una manera amplia y se encuentra relacionada con la convergencia de un proceso o metodología que presentan ciertos patrones de congruencia. Para ello se calculó la velocidad de búsqueda y aceleración promedio de los dos algoritmos planteados por medio de las varianzas de Allan, el primer índice mide la velocidad de búsqueda por medio de la variación de la distancia del mejor agente de búsqueda E , que se logra describir matemáticamente mediante las siguientes ecuaciones [25]:

$$\omega = \left| \frac{\sum_{k=1}^{k_{max}} E(k+1) - E(k)}{\Delta k} \right| \quad (2.11)$$

$$\alpha = \left| \frac{\sum_{k=1}^{k_{max}} \omega(k+1) - \omega(k)}{\Delta k} \right| \quad (2.12)$$

2.4 CASOS DE ESTUDIO

Para poder realizar las diferentes pruebas al AH fue necesario implementar dos casos de estudios que permitieron evaluar y examinar el rendimiento de ambos algoritmos y poder realizar las respectivas comparaciones.

2.4.1 Caso de estudio 1

El modelo de red consiste en un sistema de 3 generadores y una carga máxima de 360 [MW] la cual sigue un perfil de demanda dado por la Figura 2-7. Los parámetros de los generadores para este caso de estudio se definen en la Tabla 2-1, estos son los datos mínimos requeridos para ejecutar la simulación del algoritmo.

Tabla 2-1: Parámetros de generadores para estudio de confiabilidad

Tipo	Capacidad (MW)	# Unidades	Forced Outage Rate (FOR)	Tasa de Falla	Tasa de Reparación
1	100	1	0.0004577	2.289377×10^{-04}	0.5
2	150	1	0,0011	4.578754×10^{-04}	0.4
3	200	1	0,0015	5.151098×10^{-04}	0.333

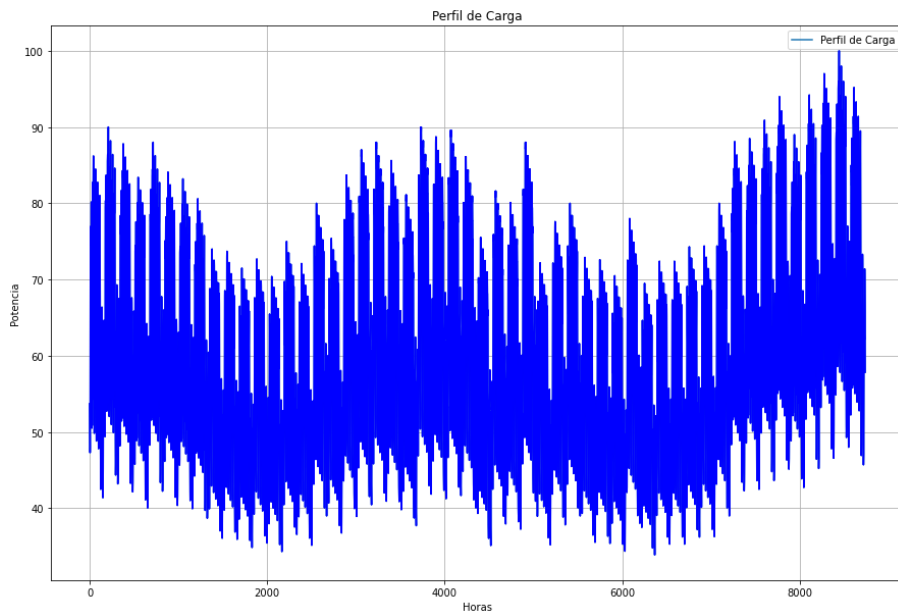


Figura 2-7: Perfil de carga Anual [26]

2.4.2 Caso de estudio 2

El modelo de red utilizado para la ejecución del algoritmo cuántico fue el IEEE Reliability Test System (RTS), el cual está constituido por 32 generadores con capacidades desde los 12 [MW] hasta los 400 [MW]. La Tabla 2-2 presenta los datos de las barras y la demanda siendo una demanda máxima de 2850 MW, el diagrama unifilar con las interconexiones del sistema se tiene en la Figura 2-8. A su vez este sistema se ajusta a un perfil de carga anual en un tiempo de estudio de 8736 horas, empleándose el mismo del caso de estudio 1 en la Figura 2-7.

Tabla 2-2: Información de barras del sistema [27]

Área 1 / Bus Numero	%\ de carga del Sistema	Carga MW	Carga MVar
101	3,8	108	22
102	3,4	97	20
103	6,3	180	37
104	2,6	74	15
105	2,5	71	14
106	4,8	136	28
107	4,4	125	25
108	6	171	35
109	6,1	175	36
110	6,8	195	40
113	9,3	265	54
114	6,8	194	39
115	11,1	317	64
116	3,5	100	20
118	11,7	333	68
119	6,4	181	37
120	4,5	128	26
Total	100	2850	580

Los datos de los generadores se detallan en la Tabla 2-3, se tienen datos como la cantidad de unidades por tipo, índices como la tasa de salidas forzadas y los tiempos medios de reparación y falla para cada unidad [27].

Tabla 2-3: Parámetros de generadores para estudio de confiabilidad

Tipo	Capacidad (MW)	# Unidades	Grupo	Forced Outage Rate (FOR)	MTTF (Horas)	MTTR (Horas)
U12	12	5	Petroleo/Vapor	0,02	2940	60
U20	20	4	Petroleo/CT	0,1	450	50
U50	50	6	Hydro	0,01	1980	20
U76	76	4	Carbón/Vapor	0,02	1960	40
U100	100	3	Petroleo/Vapor	0,04	1200	50
U155	155	4	Carbón/Vapor	0,04	960	40
U197	197	3	Petroleo/Vapor	0,05	950	50
U350	350	1	Carbón/Vapor	0,08	1150	100
U400	400	2	Nuclear	0,12	1100	150

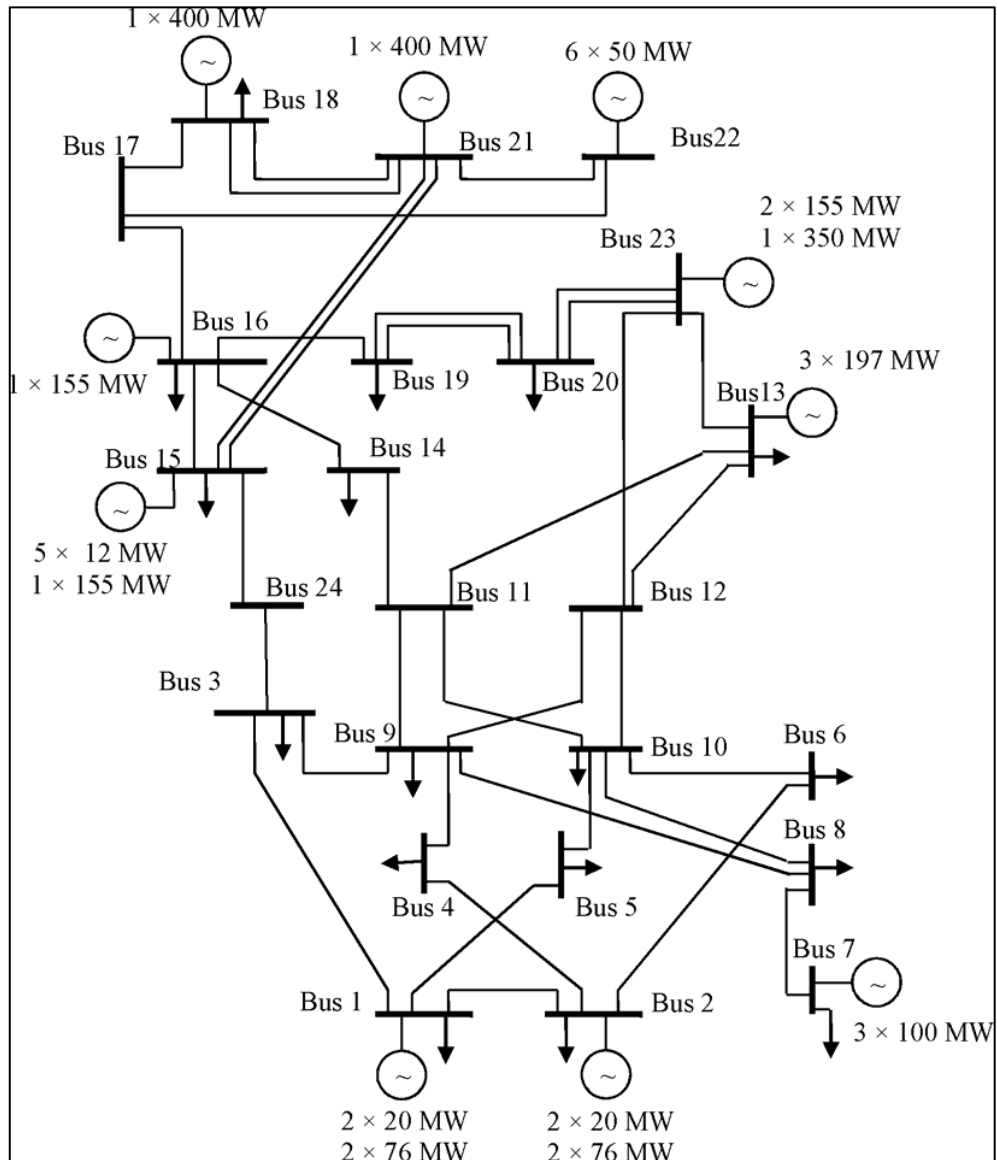


Figura 2-8: Diagrama unifilar del RTS de 24 barras [27]

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

Para la evaluación y muestra de resultados se tomaron en cuenta métodos estadísticos donde se involucra la exactitud, precisión, velocidad de convergencia y aceleración de convergencia. Todos los datos para análisis fueron obtenidos mediante la simulación con el lenguaje de programación Python™ empleando el editor de código Visual Studio Code™ para mantener un seguimiento de cada parte del código en ejecución.

Los algoritmos utilizados (híbrido y tradicional) fueron puesta a prueba para los casos de estudios planteados. El primer caso de estudio es un sistema corto conformado por tres generadores cuyos datos se encuentran en la sección 2.4.1 este sistema de menor escala ayudó en la validación de los resultados de los índices de confiabilidad y comprender el comportamiento del algoritmo híbrido, una vez comprobado el funcionamiento del algoritmo se cargó el segundo caso de estudio de la IEEE conformado por 32 generadores, los datos se pueden observar en la sección 2.4.2. Cada uno de los resultados fueron evaluados empleando los resultados obtenidos a través del software Matlab^R para la comprobación de los resultados finales.

3.1 CASO DE ESTUDIO 1

En esta sección se presentan los resultados obtenidos de la simulación llevada a cabo para el caso de estudio de tres generadores, los mismos que corresponden a los índices de confiabilidad y las características de precisión, exactitud, aceleración, velocidad y tiempo de cada simulación para 100, 200, 300, 400, 500, 1000 y 5000 experimentos.

3.1.1 Índices de confiabilidad

El cálculo de los índices de confiabilidad se obtuvo empleando la metodología descrita en las secciones 2.1 y 2.2; la Tabla 3-1 evidencia el valor de los índices de confiabilidad obtenidos tras la ejecución de los dos algoritmos. En esta tabla se evidencian el valor de los índices de confiabilidad obtenidos para 100, 200, 300, 400, 500, 1000 y 5000

experimentos. Lográndose observar que tras el incremento de experimentos el resultado logra acercarse al valor esperado.

Tabla 3-1 Índices de confiabilidad caso 1

NE	Índices	NSMC_Qubits	NSMC_Tradicional
100	LOLE	4.8989	4.42
	LOLP	0.056	0.0505
	EENS	166.4019	138.723
	XLOL	33.5807	30.399
200	LOLE	4.8737	4.7738
	LOLP	0.0557	0.0546
	EENS	153.922	150.472
	XLOL	32.349	32.0143
300	LOLE	4.722	4.7391
	LOLP	0.054	0.05424
	EENS	148.788	146.857
	XLOL	31.1033	31.3046
400	LOLE	4.9119	4.8387
	LOLP	0.056	0.0553
	EENS	155.833	153.323
	XLOL	31.629	31.471
500	LOLE	4.7838	4.798
	LOLP	0.547	0.0549
	EENS	152.5029	151.26
	XLOL	31.672	31.219
1000	LOLE	4.75	4.6764
	LOLP	0.0543	0.05353
	EENS	146.0509	145.897
	XLOL	30.6868	31.0262
5000	LOLE	4.8518	4.7816
	LOLP	0.055	0.05473
	EENS	150.7978	151.12055
	XLOL	31.083	31.672

La Figura 3-1 evidencia la tendencia de los índices de confiabilidad. Se identificó que la convergencia de los métodos es muy similar. Además, se visualizó que el algoritmo híbrido genera valores más cercanos al valor esperado de cada uno de los índices, que el método tradicional con pocas iteraciones. Se analizó cada uno de los índices de confiabilidad donde el índice LOLE muestra una aproximación del resultado esperado alrededor de 4.75-4.85 horas; para el índice LOLP se tiene una aproximación de 0.50-0.60; para el índice EENS se tiene un valor aproximado de 155-165 [MW]; para el índice XLOL se tiene un valor aproximado de 0.30-0.35.

Como se mencionó el algoritmo híbrido presenta una mejor cercanía y estabilización desde experimentos muy bajos, como es de costumbre en estudios de confiabilidad a medida que aumentamos el número de experimentos tendremos una mayor cercanía al valor esperado, lo que le ocurre al algoritmo tradicional y podemos evidenciarlo en la Figura 3-1 y en la evolución del índice EENS de la Figura 3-2.

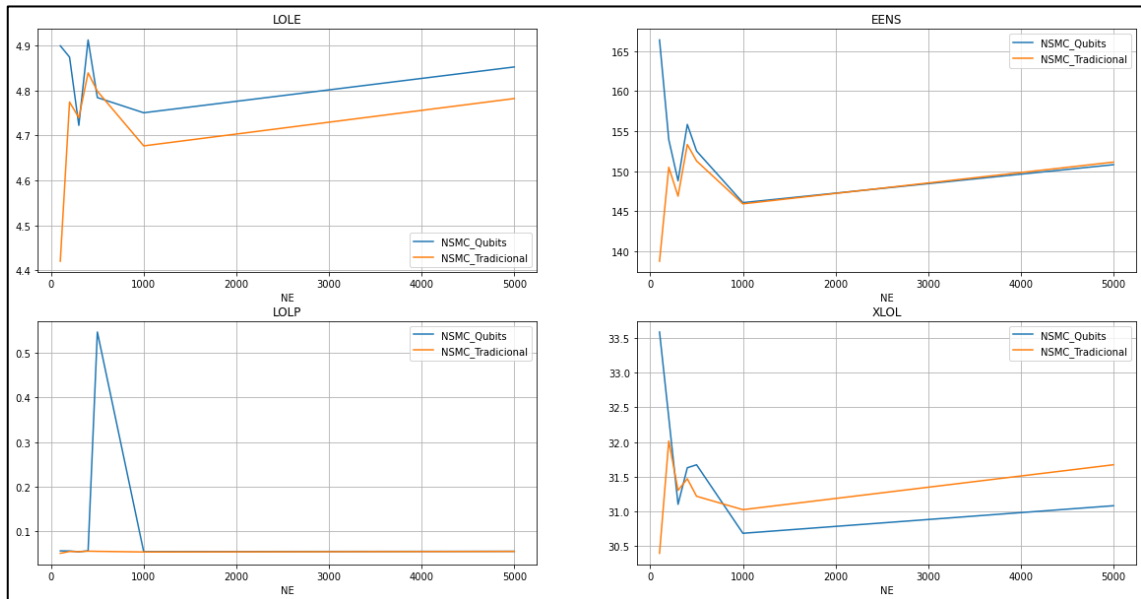
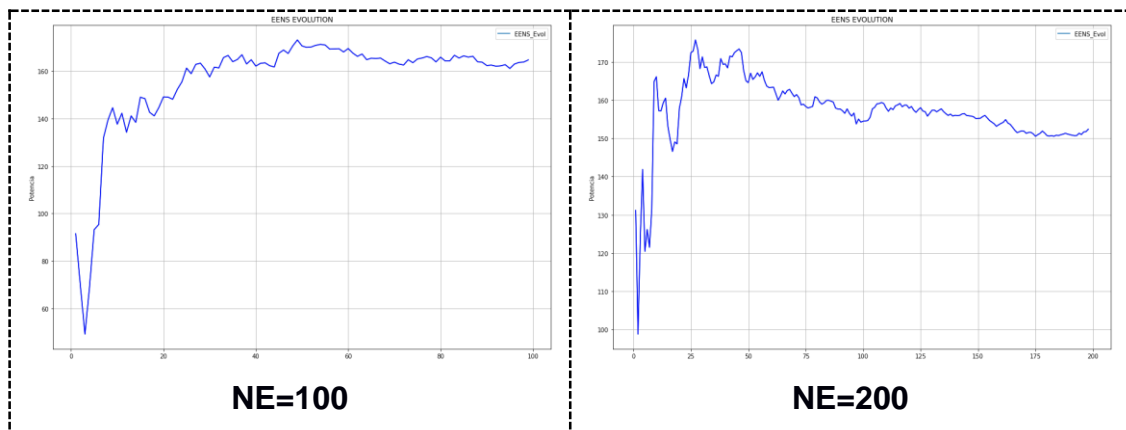


Figura 3-1: Índices de confiabilidad caso 1

Para comprobar la convergencia del nuevo algoritmo, se emplea la gráfica del índice EENS, esto considerando que es uno de los indicadores con la convergencia más lenta entre los índices de estudio.



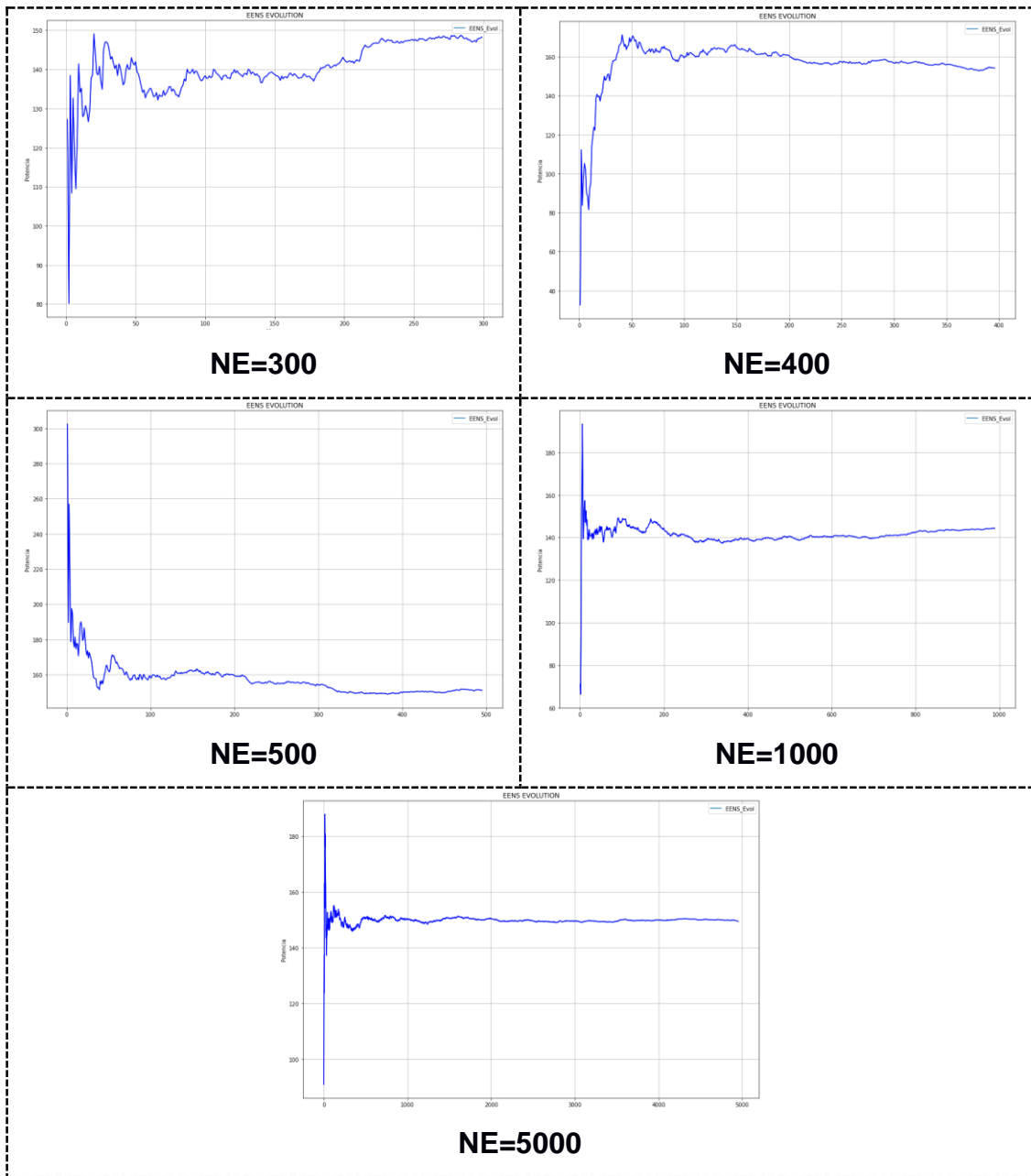
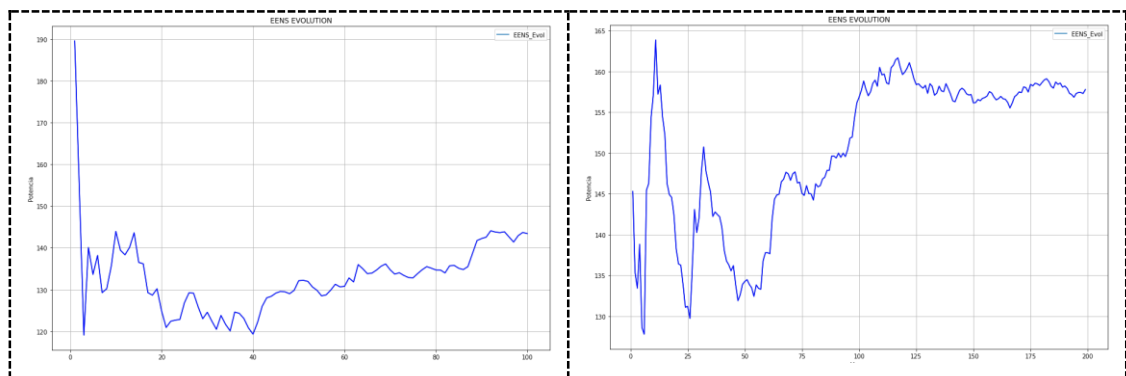


Figura 3-2: Evolución de índice EENS para NSMC híbrido caso 1



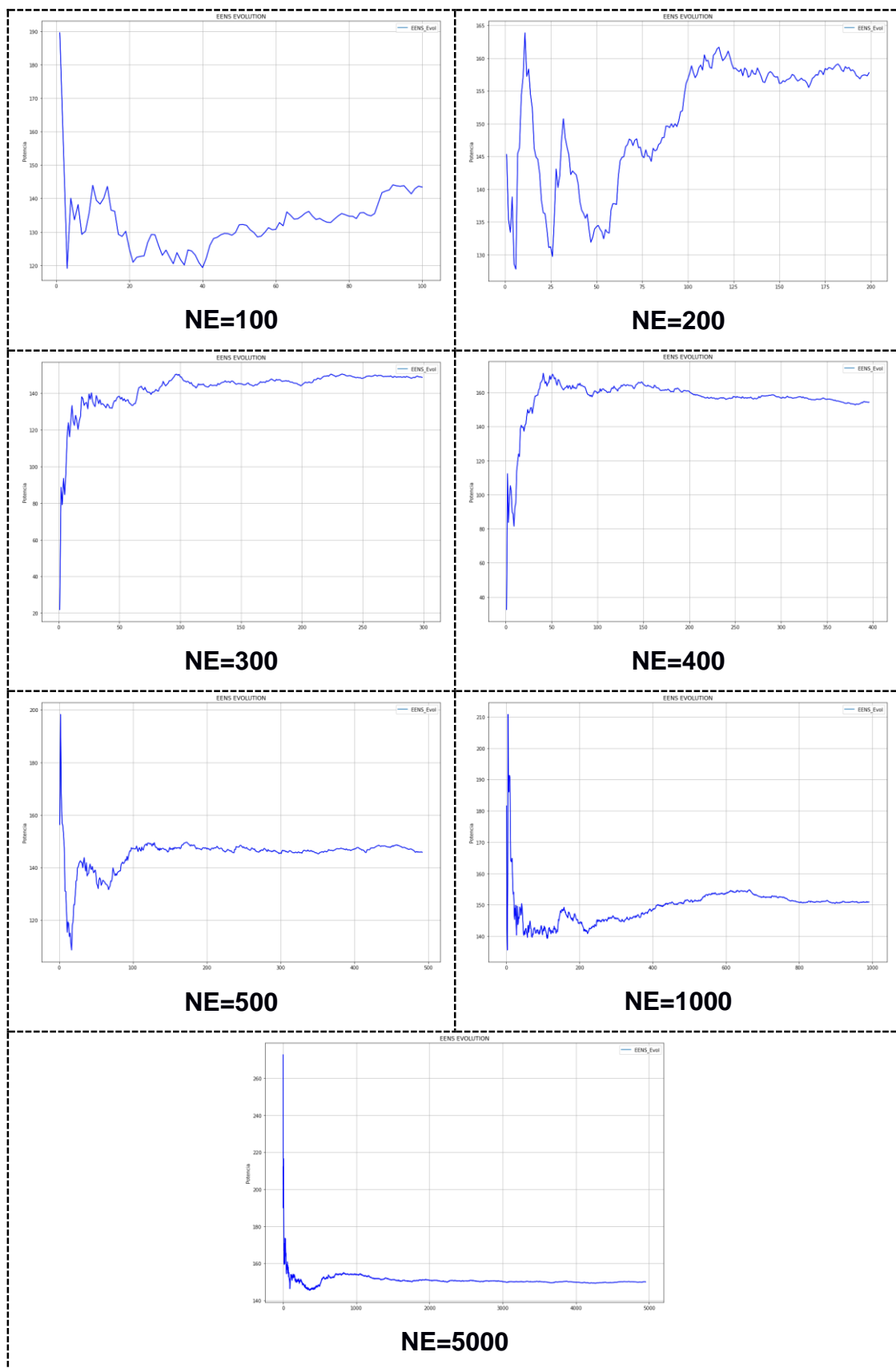


Figura 3-3: Evolución de índice ENNS para NSMC tradicional caso 1

La Figura 3-2 para cada uno de los experimentos del método híbrido se determinó que desde la iteración más baja NE=100 se tiene una estabilidad en la evolución de este índice de confiabilidad, es decir, para un tiempo corto determinado se estabiliza dentro de un rango de valores, este índice nos dice la energía no suministrada en nuestro sistema durante el tiempo de estudio.

En la Figura 3-3 pudimos notar que en la evolución del índice EENS la estabilización se logra a partir del NE=300, es necesario que el algoritmo tradicional aumente su número de experimentos para poder obtener los resultados deseados. A medida que se aumentó el número de experimentos el algoritmo tradicional mostraba una mejor convergencia.

3.1.2 Precisión y exactitud

La exactitud y precisión de los datos se obtuvieron evaluando las ecuaciones (2.9) y (2.10); estos indicadores nos permitieron relacionar la cercanía y dispersión de los datos, respecto al valor esperado para cada caso de estudio. Para el caso de estudio 1 conformado por 3 generadores se tiene para cada uno de los experimentos los resultados que se muestran en la Tabla 3-2.

Tabla 3-2 Resultados de precisión y exactitud caso 1

NE	Métricas	NSMC_Qubits	NSMC_Tradicional
100	δ	4.8989	10.266
	Φ	3.1487	2
200	δ	6.886	6.2871
	Φ	0.5484	0.3014
300	δ	9.7255	0.361
	Φ	0.3347	0.2481
400	δ	6.2966	2.35
	Φ	1.135	0.2049
500	δ	6.9337	0.77099
	Φ	0.785	0.3428
1000	δ	8.6845	2.60712
	Φ	0.1666	0.5475
5000	δ	0.4392	1.244
	Φ	0.02584	0.1038

3.1.3 Velocidad y aceleración

Para las mediciones de velocidad y aceleración de convergencia se emplearon las ecuaciones (2.11) y (2.12) para el cálculo de la velocidad se tomaron los datos de la evolución del índice de confiabilidad EENS, en la ecuación (2.11) se aplica el concepto de derivada para poder aproximar la convergencia de cada uno de los métodos analizados, posteriormente los datos arrojados al aplicar la ecuación (2.11) se vuelve al mismo criterio para dar lugar a la ecuación (2.12). Los datos arrojados son mostrados en la Tabla 3-3 y graficados en la Figura 3-4 para el Caso 1.

Tabla 3-3 Resultados de velocidad y aceleración caso 1

NE	Métricas	NSMC_Qubits	NSMC_Tradicional
100	ω	73.319	26.5484
	α	22.658	27.4975
200	ω	21.22934	11.086
	α	32.998	16.09
300	ω	21.64	2.30427
	α	47.2712	62.7038
400	ω	121.6822	8.2073
	α	79.6706	65.4671
500	ω	151.4242	41.54053
	α	1124932	41.5508
1000	ω	73.4863	248.8214
	α	4.4863	159.186
5000	ω	58.5303	159.416
	α	23.3	83.121

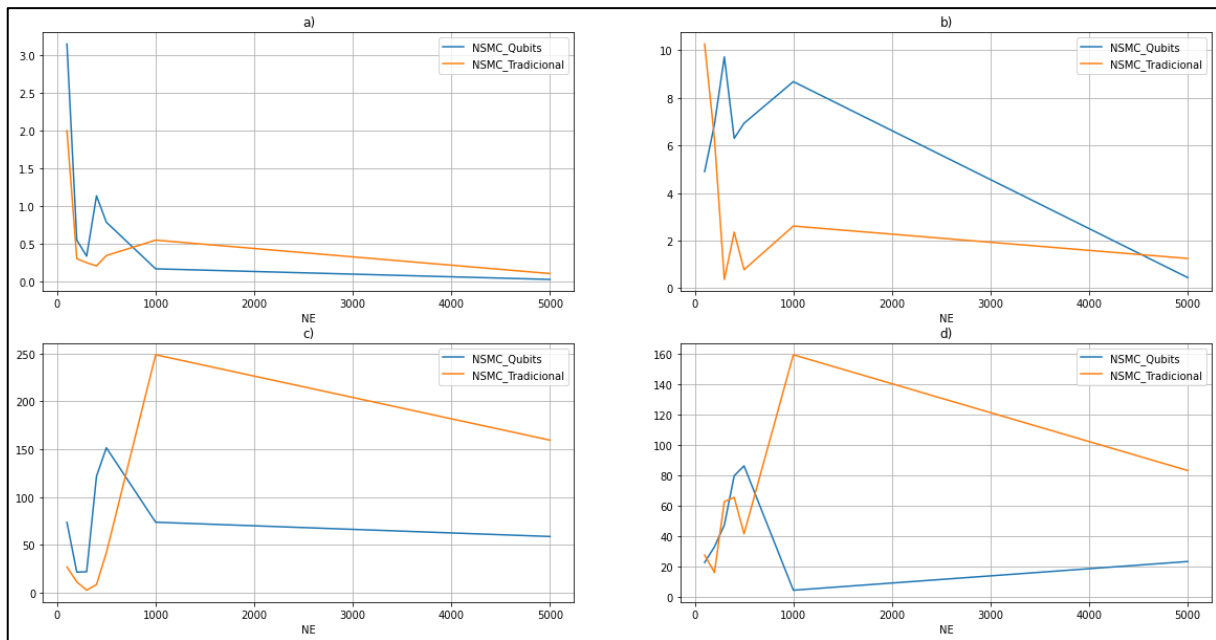


Figura 3-4 Precisión, exactitud, velocidad y aceleración caso 1

3.1.4 Tiempo de simulación

La medida del tiempo de simulación de cada uno de los algoritmos fue efectuada, implementando la librería Time de Python™, estableciendo un punto de inicio y punto final de medición. El punto final se estableció en el momento de que cada uno de los algoritmos converge a los resultados de los índices de confiabilidad, es decir, en la etapa donde se evidencia el rendimiento de cada algoritmo. Los datos para cada iteración se muestran en la Tabla 3-4 y Figura 3-5 para el Caso 1.

Tabla 3-4 Tiempo de simulación caso 1

Tiempo de simulación [s]		
NE	NSMC_Qubits	NSMC_Tradicional
100	12,98	1,86
200	23,02	3,52
300	35,42	5,71
400	46,68	7,64
500	57,80	9,60
1000	111,63	17,96
5000	551,96	91,29

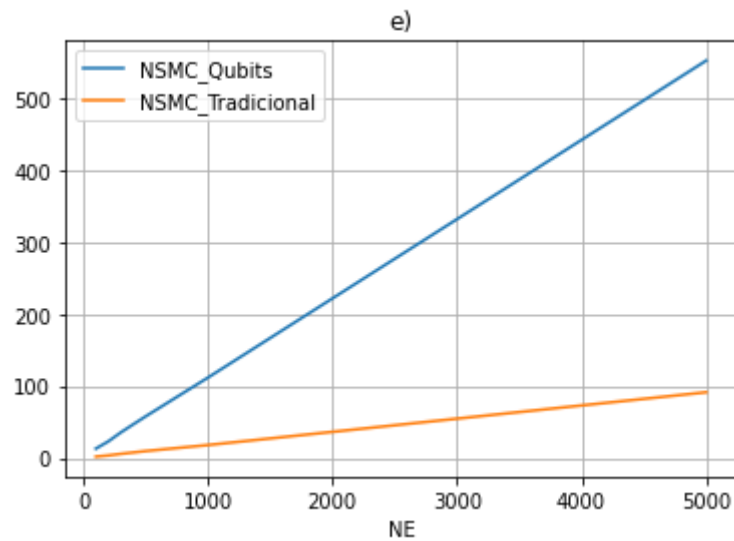


Figura 3-5 Tiempo de simulación caso 1

3.2 CASO DE ESTUDIO 2

En esta sección se presentan los resultados obtenidos de la simulación llevada a cabo para el caso de estudio IEEE de 32 generadores, al igual que el caso anterior se presentan los mismos índices e indicadores para evaluar el rendimiento del algoritmo para el mismo número de experimentos.

3.2.1 Índices de confiabilidad

El cálculo de los índices de confiabilidad se obtuvo empleando la metodología descrita en las secciones 2.1 y 2.2; La aplicación tanto del algoritmo híbrido, así como el método tradicional de Monte Carlo sobre el caso de estudio IEEE RTS permitieron obtener una mejor apreciación sobre la aproximación de los índices de confiabilidad. Como se esperó, a medida que el número de experimentos fue incrementando, los índices de confiabilidad (LOLE, LOLP, EENS y XLOL) fueron acercándose lo más posible al resultado esperado tal como se evidencia en la Tabla 3-5 y la Figura 3-6.

Tabla 3-5: Índices de confiabilidad caso 2

NE	Índices	NSMC_Qubits	NSMC_Tradicional
100	LOLE	9,31	9,61
	LOLP	0,1065	0,11
	EENS	1142,8983	1219,13
	XLOL	124,7676	126,3046
200	LOLE	9,605	9,86
	LOLP	0,1099	0,1128
	EENS	1224,30168	1231,81
	XLOL	125,8454	125,94
300	LOLE	9,5266	9,5766
	LOLP	0,109	0,1096
	EENS	1186,2054	1223,1831
	XLOL	125,051	127,3254
400	LOLE	9,47	9,505
	LOLP	0,1084	0,1088
	EENS	1178,3688	1179,4493
	XLOL	124,164	125,3455
500	LOLE	9,4226	9,6336
	LOLP	0,1078	0,1103
	EENS	1139,8394	1234,4132
	XLOL	119,6831	126,6
1000	LOLE	9,402	9,401
	LOLP	0,1076	0,1076
	EENS	1197,4608	1155,5712
	XLOL	126,7029	122,5525
5000	LOLE	9,446	9,4188
	LOLP	0,1081	0,1078
	EENS	1189,6735	1183,1144
	XLOL	126,511	125,193

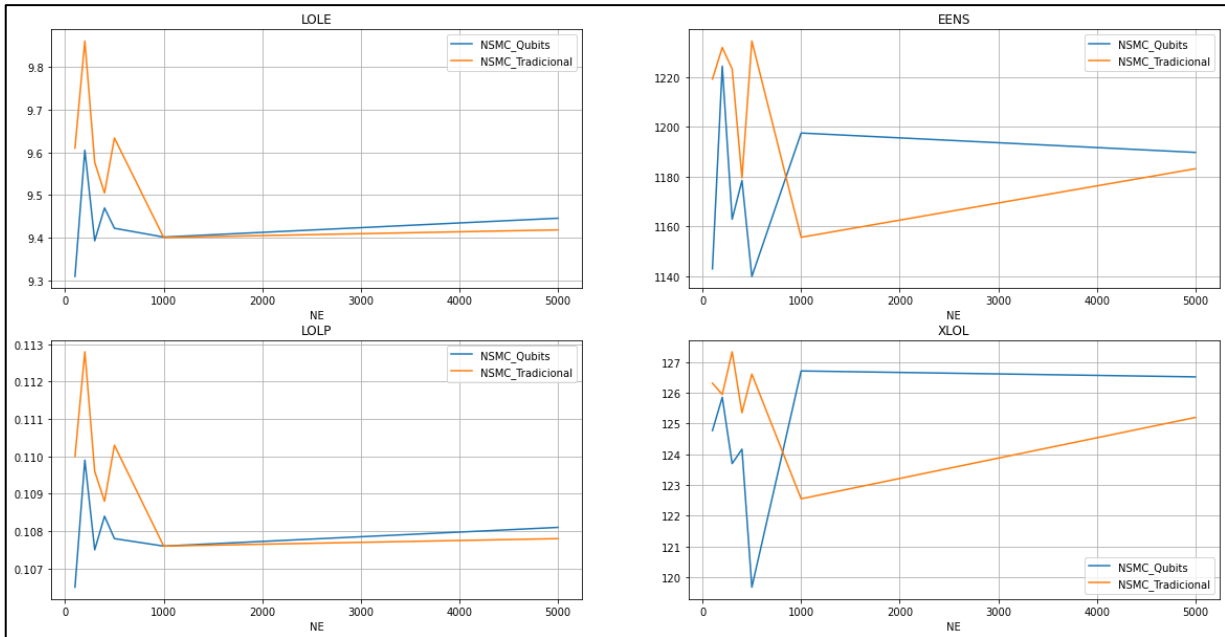


Figura 3-6 Índices de confiabilidad caso 2

En la Tabla 3-5 del caso de estudio 2, se incrementó el número de generadores en el sistema para poder comprobar el rendimiento del algoritmo híbrido en un sistema más complejo, obtuvimos que, para el índice de confiabilidad LOLE tenemos un valor aproximado de 9.30 a 9.45 [h]; para el índice LOLP tenemos un valor aproximado entre 0.10-0.12; para el índice EENS tenemos un valor aproximado de 1140-1190 [MW]; para el índice XLOL tenemos un valor entre 1.20 y 1.26. Nuevamente nos percatamos que para el algoritmo híbrido mantiene su estabilidad a pesar de que se cambió el sistema de estudio, es decir, desde las primeras iteraciones los valores de cada uno de los índices se mantienen oscilando en un rango no muy disperso. Mientras que, el algoritmo tradicional mantiene su mismo comportamiento, que a medida que el número de experimentos comienza a incrementarse su estabilidad mejora y se acerca al valor de aproximación correcto. El comportamiento de estabilidad de uno de los índices de confiabilidad lo podemos notar en la Figura 3-7 y Figura 3-8 que muestra la evolución del índice EENS para cada una de las iteraciones en el caso de estudio 2.

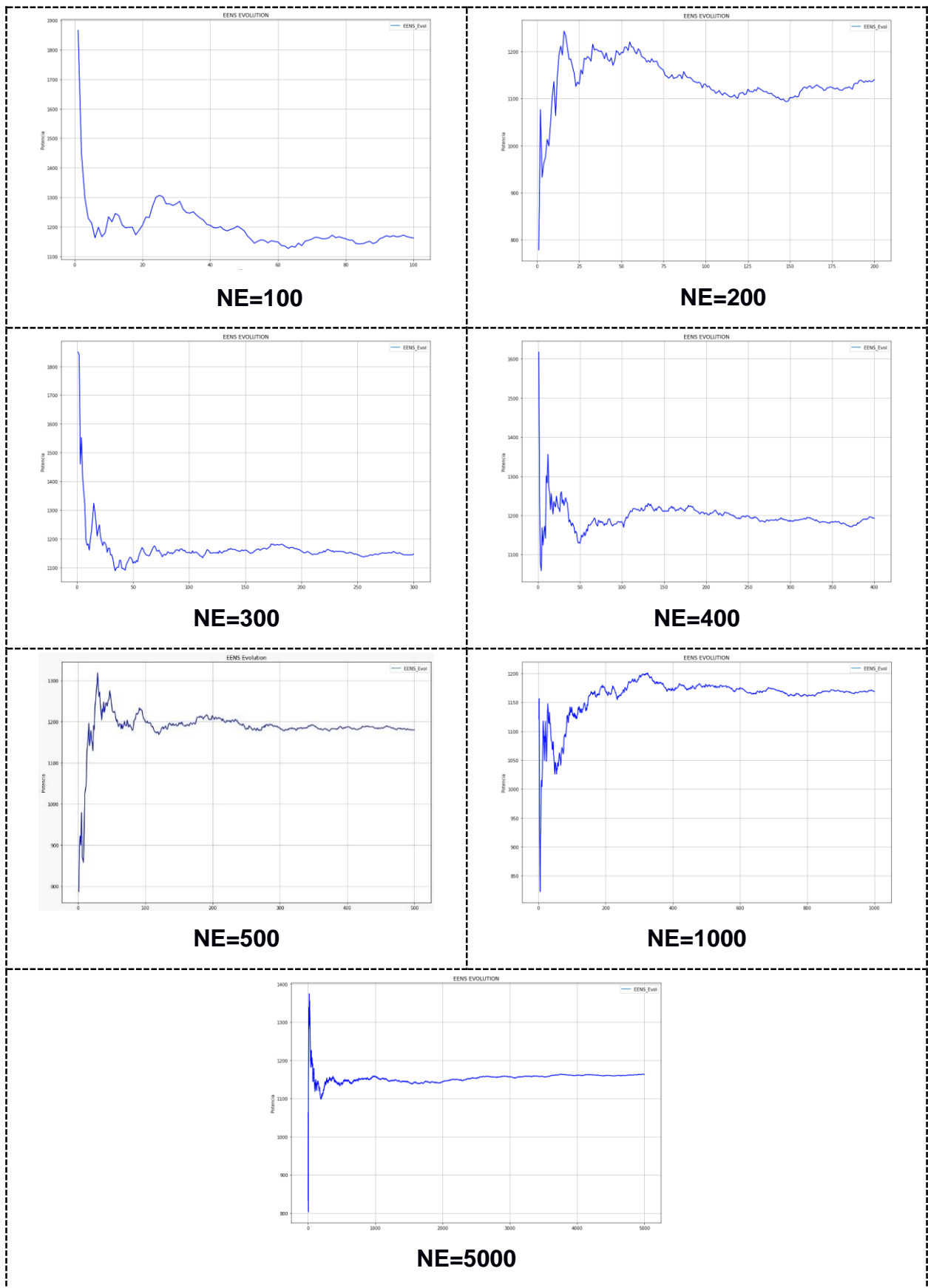


Figura 3-7 Evolución de índice ENNS para NSMC híbrido caso 2

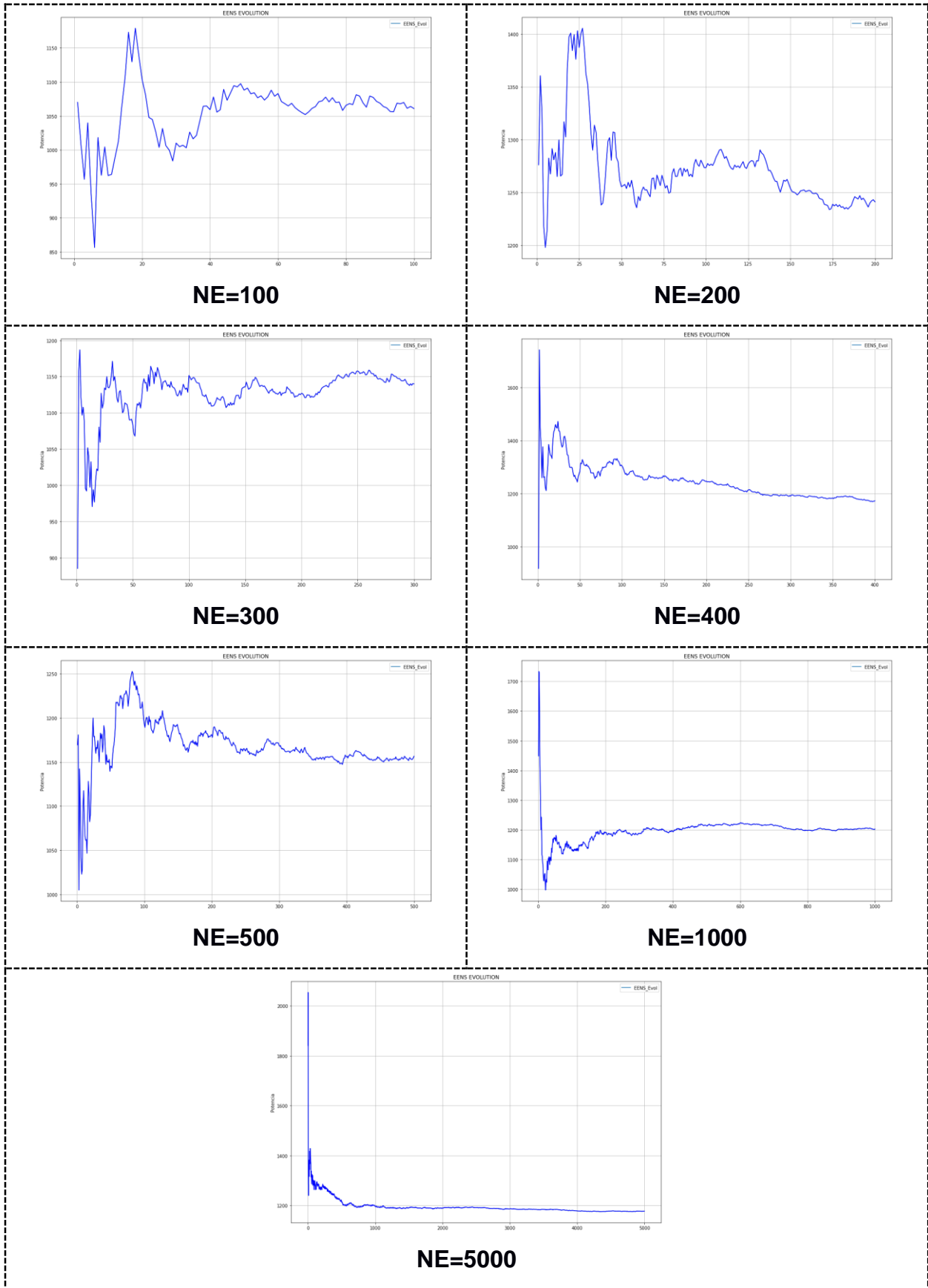


Figura 3-8 Evolución de índice ENNS para NSMC tradicional caso 2

La Figura 3-7 y Figura 3-8 de la evolución del índice de confiabilidad tanto para el método híbrido como para el método tradicional nos mostraron que presentan un comportamiento muy similar al del Caso 1 revisado en la sección 3.1.1; es decir, el AH mantiene su estabilidad desde los primeros experimentos pero como es de esperarse que al incrementarse el número de experimentos su estabilidad mejora similar al método tradicional pero con la diferencia de que el método tradicional en sus primeras iteraciones no presenta estabilidad en ningún periodo de horas de estudio, es muy inestable y recién para el NE=1000 comienza a estabilizarse. En el caso de estudio 1 al tener un sistema más pequeño conformado solamente por tres generadores, el método tradicional presentaba una estabilidad desde el NE=300 pero al tener un sistema más complejo su eficacia mejora desde un número de experimento mayor; mientras que el algoritmo híbrido mantiene su comportamiento en ambos casos tanto para un sistema pequeño como para un sistema con mayor número de generadores.

3.2.2 Precisión y exactitud

La exactitud y precisión de los datos se obtuvieron evaluando las ecuaciones (2.9) y (2.10); estos indicadores nos permitieron relacionar la cercanía y dispersión de los datos, respecto al valor esperado para cada caso de estudio. Para el caso de estudio 2 de la IEEE conformado por 32 generadores se tiene para cada uno de los experimentos la Tabla 3-6 que muestra la exactitud y precisión de cada algoritmo.

Tabla 3-6: Resultados de precisión y exactitud caso 2

NE	Métricas	NSCM_Qubits	NSMC_Tradicional
100	δ	26,6169	53,87177
	Φ	4,8228	0,6267
200	δ	82,955	24,178
	Φ	2,6986	12,6309
300	δ	27,783	39,4686
	Φ	0,8857	3,4991
400	δ	15,6952	49,8303
	Φ	1,8099	5,4886
500	δ	24,3559	35,8012
	Φ	2,9159	1,436
1000	δ	24,844	27,5889
	Φ	0,3298	0,8085
5000	δ	23,5984	14,5043
	Φ	0,4574	0,2314

3.2.3 Velocidad y aceleración

Para las mediciones de velocidad y aceleración de convergencia se emplearon las ecuaciones (2.11) y (2.12); para el cálculo de la velocidad se tomaron los datos de la evolución del índice de confiabilidad ENS, en la ecuación (2.11) se aplica el concepto de derivada para poder aproximar la convergencia de cada uno de los métodos analizados, posteriormente los datos arrojados al aplicar la ecuación (2.11) se vuelve al mismo criterio para dar lugar a la ecuación (2.12). Los datos arrojados son mostrados en la Tabla 3-7: Resultados de velocidad y aceleración para el caso 2 y Figura 3-9 para el Caso 2.

Tabla 3-7: Resultados de velocidad y aceleración para el caso 2

NE	Métricas	NSMC_Qubits	NSMC_Tradicional
100	ω	483,61498	39,244
	α	337,626	37,5518
200	ω	180,159	1089,767
	α	257,8386	378,9732
300	ω	101,5023	589,8612
	α	296,217	286,8892
400	ω	531,1726	443,9699
	α	131,9	584,9315
500	ω	105,4285	194,918
	α	685,3821	249,599
1000	ω	176,033	586,17
	α	70,4027	96,3046
5000	ω	539,485	64,5521
	α	484,2071	224,7683

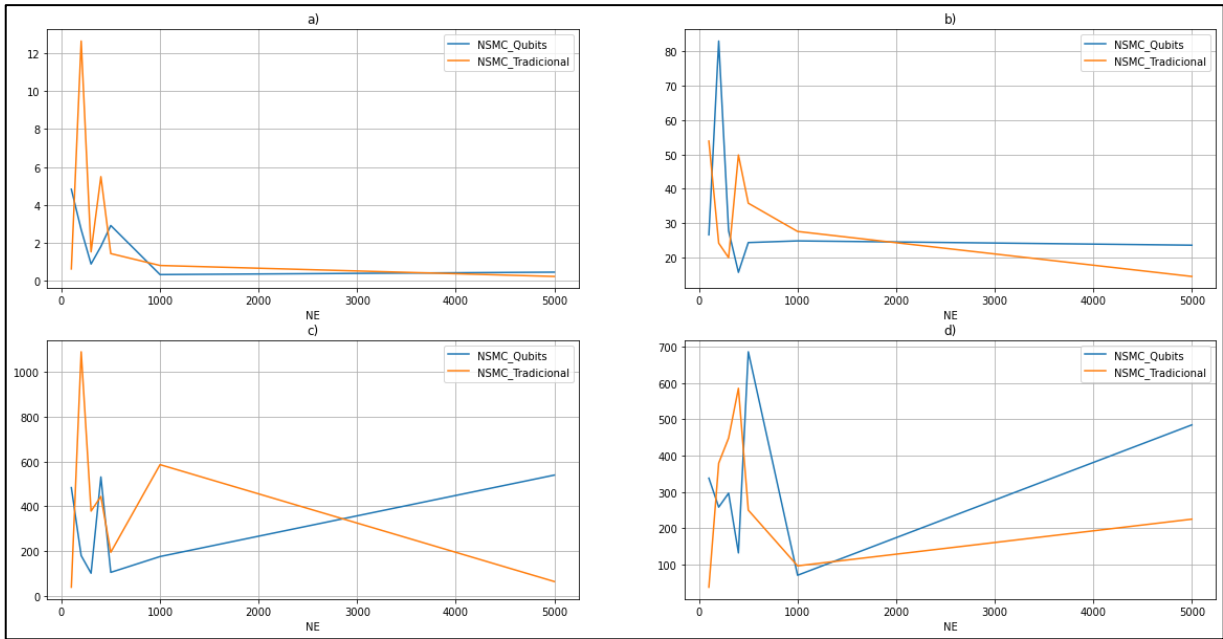


Figura 3-9 Precisión, exactitud, velocidad y aceleración caso 2

3.2.4 Tiempo de simulación

La medida del tiempo de simulación de cada uno de los algoritmos fue efectuada, implementando la librería Time de Python™, estableciendo un punto de inicio y punto final de medición. El punto final se estableció en el momento de que cada uno de los algoritmos converge a los resultados de los índices de confiabilidad es decir en la etapa donde se evidencia el rendimiento de cada algoritmo. Los datos para cada iteración se muestran en la Tabla 3-8 y Figura 3-10 para el caso 2.

Tabla 3-8 Tiempo de simulación caso 2

Tiempo de simulación [s]		
NE	NSMC_Qubits	NSMC_Tradicional
100	103.00	14.00
200	215.00	27.00
300	361.00	47.00
400	435.00	56.00
500	566.00	73.00
1000	1085.00	110.00
5000	5670.00	461.00

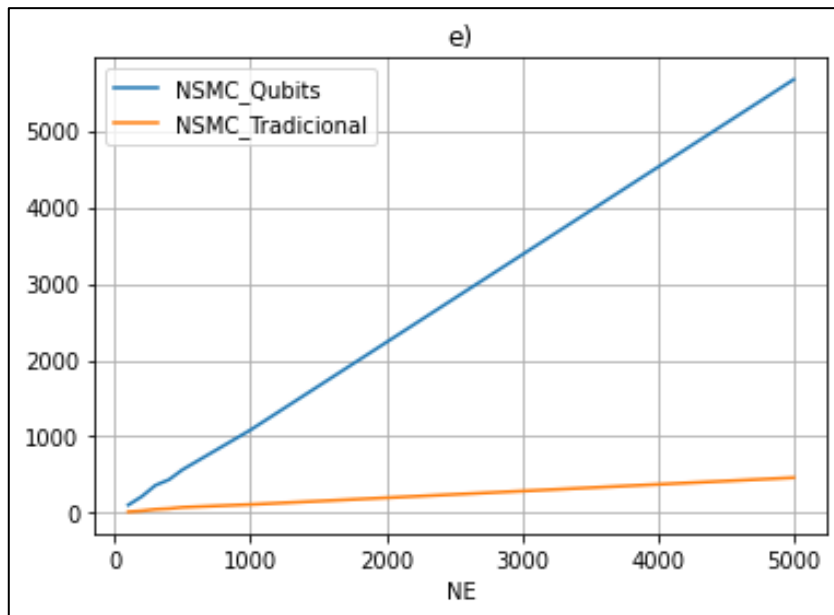


Figura 3-10 Tiempo de simulación caso 2

3.3 RENDIMIENTO GLOBAL PARA EVALUACIÓN DE CRITERIOS DE MEDICIÓN

Para un mejor análisis de desempeño de los criterios de medición propuestos: exactitud, precisión, velocidad de búsqueda, aceleración de búsqueda y el tiempo de simulación fueron definidas por reglas de grado o reglas de correspondencia. Las reglas fueron desarrolladas para poder facilitar una mejor comprensión y comparación de ambos algoritmos propuestos dentro de los parámetros; para las reglas de correspondencia tenemos: '3' para un excelente rendimiento, '2' para un buen rendimiento, '1' para un regular rendimiento, '0' para bajo rendimiento [25]. Cabe recalcar que para poder estimar una regla de correspondencia en los valores de tiempo de simulación fue necesario efectuar una conversión para tratar de que los resultados mostrados tanto del algoritmo híbrido como del tradicional no se tornen dispersos y poder hacer una agrupación adecuada. Para ello fue necesario tomar los datos y efectuar la siguiente conversión:

$$\tau = \text{Log}(t) \tag{3.1}$$

Donde:

τ es el nuevo valor obtenido

t es el tiempo de simulación en [s]

A continuación, se muestran los nuevos valores obtenidos en los tiempos de simulación aplicando la ecuación (3.1)

Tabla 3-9 Escala de datos para el tiempo de simulación en caso 1

τ caso I		
NE	NSMC_Qubits	NSMC_Tradicional
100	1.11	0.27
200	1.36	0.55
300	1.55	0.76
400	1.67	0.88
500	1.76	0.98
1000	2.05	1.25
5000	2.74	1.96

Tabla 3-10 Escala de datos para el tiempo de simulación en caso 2

τ caso II		
NE	NSMC_Qubits	NSMC_Tradicional
100	2.013	1.146
200	2.332	1.431
300	2.558	1.672
400	2.638	1.748
500	2.753	1.863
1000	3.035	2.041
5000	3.754	2.664

3.3.1 Caso de estudio I

Para el primer caso de estudio conformado por el sistema pequeño de 3 generadores tenemos la siguiente regla de correspondencia para cada uno de los criterios de medición:

$$regla \delta = \begin{cases} 3 & si \quad \delta < 1 \\ 2 & si \quad 1 \leq \delta < 5 \\ 1 & si \quad 5 \leq \delta < 10 \\ 0 & si \quad \delta \geq 10 \end{cases} \quad (3.2)$$

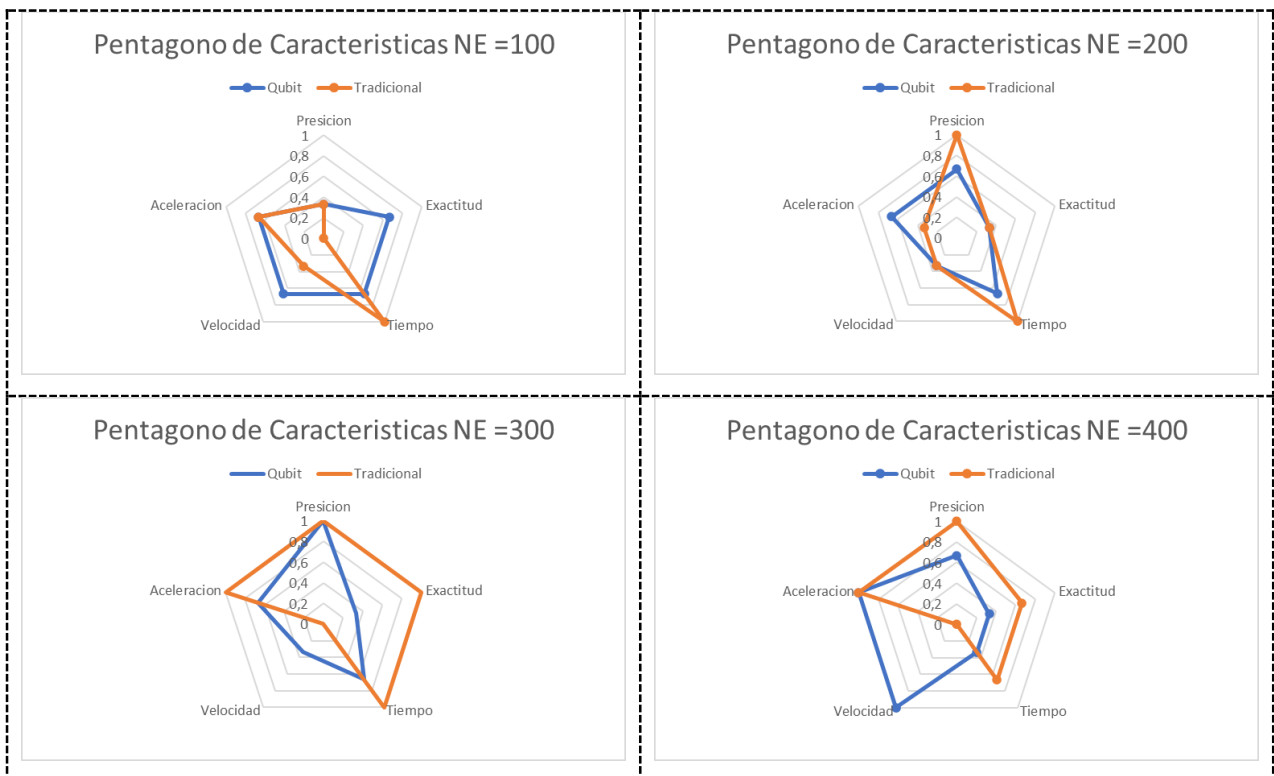
$$regla \varphi = \begin{cases} 3 & si \quad \varphi < 0.5 \\ 2 & si \quad 0.5 \leq \varphi < 1.5 \\ 1 & si \quad 1.5 \leq \varphi < 3.5 \\ 0 & si \quad \varphi \geq 3.5 \end{cases} \quad (3.3)$$

$$regla \omega = \begin{cases} 3 & si \quad \omega \geq 100 \\ 2 & si \quad 50 \leq \omega < 100 \\ 1 & si \quad 10 < \omega < 50 \\ 0 & si \quad \omega \leq 10 \end{cases} \quad (3.4)$$

$$regla \alpha = \begin{cases} 3 & si \quad \alpha \geq 50 \\ 2 & si \quad 20 \leq \alpha < 50 \\ 1 & si \quad 5 < \alpha < 20 \\ 0 & si \quad \alpha \leq 5 \end{cases} \quad (3.5)$$

$$regla \tau = \begin{cases} 3 & si \quad \tau < 0.82 \\ 2 & si \quad 0.82 \leq \tau < 1.65 \\ 1 & si \quad 1.65 \leq \tau < 2.47 \\ 0 & si \quad \tau \geq 2.47 \end{cases} \quad (3.6)$$

..



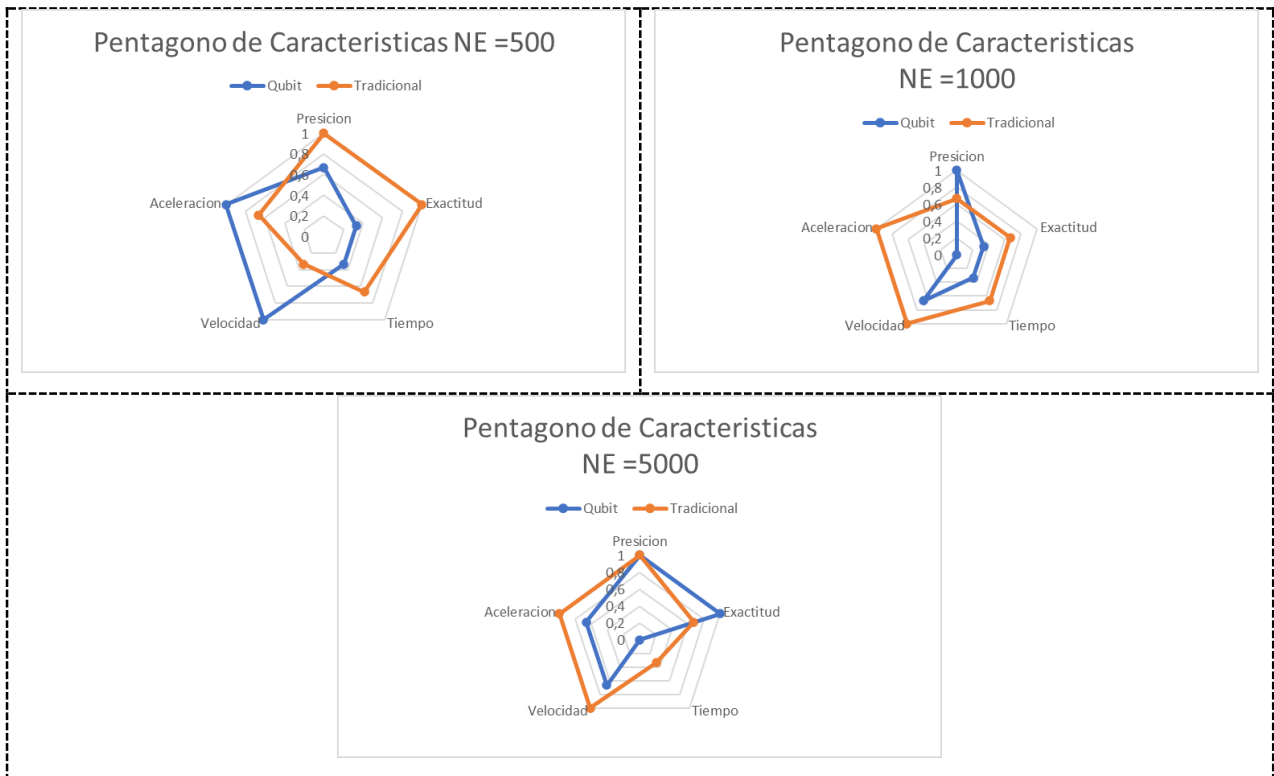


Figura 3-11 Técnica de optimización del rendimiento caso 1

3.3.2 Caso de estudio II

El segundo caso de estudio conformado por el sistema IEEE de 32 generadores se establecen las siguientes reglas de correspondencias para los diferentes criterios de medición:

$$regla \delta = \begin{cases} 3 & si \quad \delta < 15 \\ 2 & si \quad 15 \leq \delta < 25 \\ 1 & si \quad 25 \leq \delta < 30 \\ 0 & si \quad \delta \geq 30 \end{cases} \quad (3.7)$$

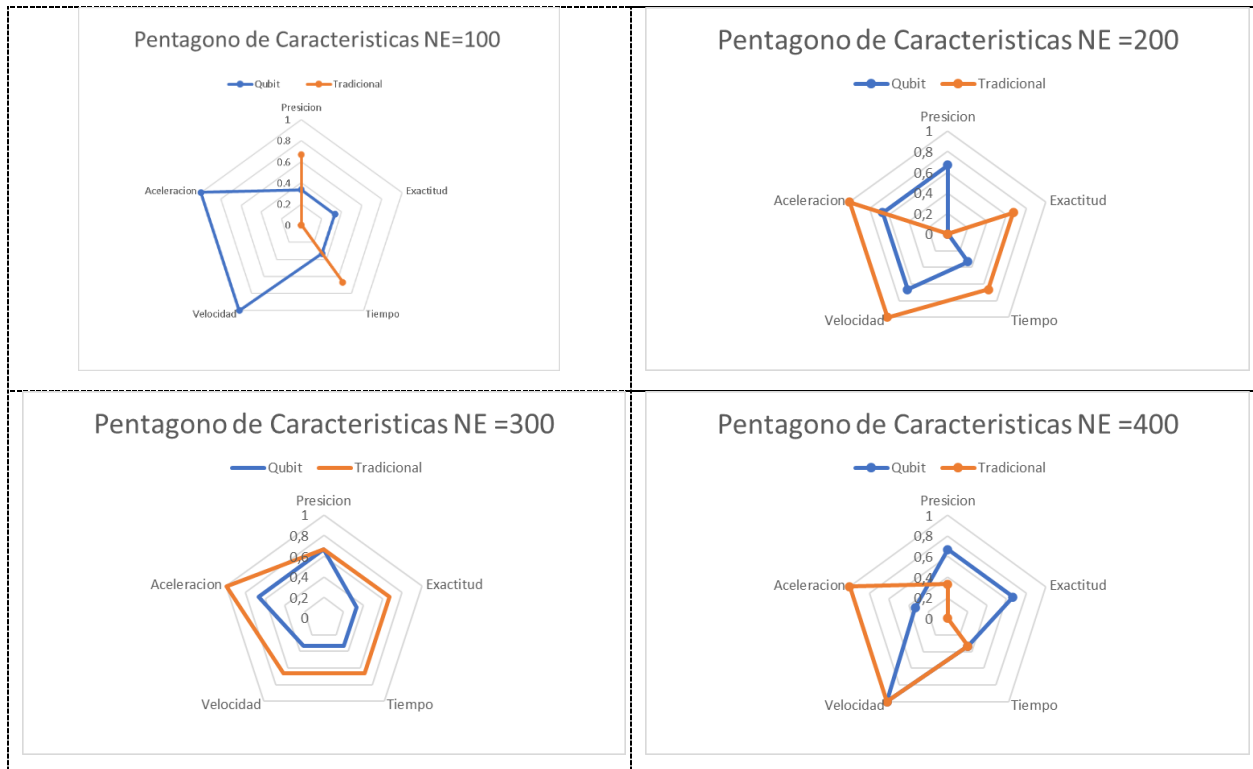
$$regla \varphi = \begin{cases} 3 & si \quad \varphi < 0.5 \\ 2 & si \quad 0.5 \leq \varphi < 3 \\ 1 & si \quad 3 \leq \varphi < 10 \\ 0 & si \quad \varphi \geq 10 \end{cases} \quad (3.8)$$

$$regla \omega = \begin{cases} 3 & si \quad \omega \geq 400 \\ 2 & si \quad 150 \leq \omega < 400 \\ 1 & si \quad 50 < \omega < 150 \\ 0 & si \quad \omega \leq 50 \end{cases} \quad (3.9)$$

$$regla \alpha = \begin{cases} 3 & si \quad \alpha \geq 300 \\ 2 & si \quad 200 \leq \alpha < 300 \\ 1 & si \quad 50 < \alpha < 200 \\ 0 & si \quad \alpha \leq 50 \end{cases} \quad (3.10)$$

$$regla \tau = \begin{cases} 3 & si \quad \tau < 1.04 \\ 2 & si \quad 1.04 \leq \tau < 1.7 \\ 1 & si \quad 1.7 \leq \tau < 2.8 \\ 0 & si \quad \tau \geq 2.8 \end{cases} \quad (3.11)$$

Una vez que se establecieron las reglas de correspondencias para cada criterio de medición se formó para ambos experimentos un pentágono de características para poder establecer un rango de lectura se toma el valor máximo de los datos obtenidos por medio de la regla de correspondencia para obtener valores en un rango de 0 y 1.



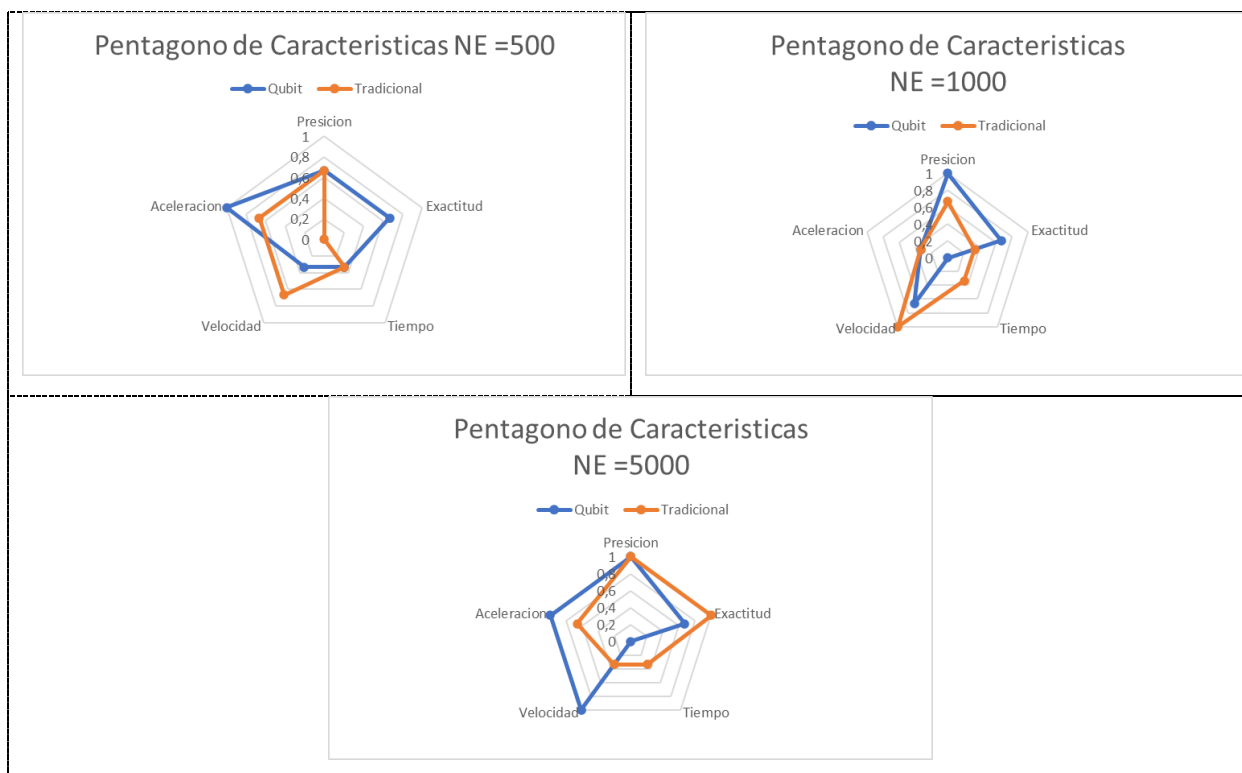


Figura 3-12 Técnica de optimización del rendimiento en caso 2

3.4 ANÁLISIS DE RESULTADOS

Los resultados mostrados en la Tabla 3-2 y Tabla 3-6 presentan la exactitud y precisión de ambos algoritmos para los diferentes sistemas de estudio. La evaluación de ambos parámetros se realiza con respecto al algoritmo que presente menor valor de exactitud y precisión, es decir el que presente menor dispersión en los datos obtenidos. Ambos criterios son evaluados mediante la evolución del índice de confiabilidad EENS; en el caso de estudio 1 el algoritmo híbrido para la exactitud presenta pequeñas desventajas con respecto al algoritmo tradicional. Sin embargo, a medida que aumenta el número de experimentos el algoritmo híbrido presenta mayor ventaja con respecto a la exactitud para las iteraciones 100 y 5000 debido al aumento del número de muestras. Para el caso de estudio 2 donde el sistema aumenta su tamaño el AH presenta mayor ventaja con respecto al tradicional, recordemos que en las secciones 3.1.1 y 3.2.1 donde se evidenció el comportamiento de los índices de confiabilidad, el AH mostraba igual rendimiento en comparación al método tradicional con tan solo 200 experimento, cabe recalcar que al tener un sistema más complejo el método tradicional requiere mayor número

de iteraciones para tener un resultado aceptable. Finalmente se tiene que ambos algoritmos al llegar al mayor número de experimentos aumentan su precisión y exactitud.

Los resultados presentados en la Tabla 3-3 y Tabla 3-7. Este indicador expresa la rapidez y aceleración con la cual el algoritmo tiende a aproximar el valor esperado del EENS para cada uno de los casos de estudio. Como se logró constatar, el AH con tan solo 100 experimentos presenta una velocidad y aceleración relativamente alta a diferencia del algoritmo tradicional. Sin embargo, es importante rescatar que el método tradicional también logra igualar y en ocasiones superar al AH con el aumento de experimentos, lo cual a su vez requiere mayor cantidad de recursos computacionales dejando al AH como la opción más viable para la optimización del proceso de cómputo. Finalmente es importante resaltar que si se consideran tanto estos indicadores como los de precisión y exactitud se logra constatar que el AH garantiza una convergencia de resultados acelerador con una gran precisión y mínima dispersión en los resultados asegurando así buena fidelidad en estos.

Los resultados obtenidos en las tablas Tabla 3-4 y Tabla 3-8 para ambos casos de estudio nos permitieron evaluar el rendimiento en cuanto a tiempo de simulación del algoritmo híbrido y el método tradicional. Es importante recalcar que el comportamiento del tiempo de simulación para el AH es de tipo cubico, mientras el del método tradicional es cuadrático para el rango de estudio, lo que nos permite concluir que el tiempo de simulación del método propuesto no es el mejor y es muy lento en su convergencia cuando se elevan el número de experimentos.

Para la evaluación de las técnicas de optimización en las mediciones de exactitud, precisión, velocidad y aceleración de convergencia se efectuó para cada caso de estudio un pentágono de características, para la formación del pentágono fue necesario agrupar los datos de cada variable y establecer una escala de medición para que todos los resultados se encuentren dentro de un rango de valores predeterminados para su respectiva interpretación, se establecieron reglas de correspondencia para el Caso 1 mostradas en 3.3.1 y para el Caso 2 se establecieron reglas de correspondencia que se observan en 3.3.2. Con los datos obtenidos en las reglas de correspondencias se procedió a realizar el pentágono

de características para cada número de experimentos como se aprecia en Figura 3-11. Según la apreciación del pentágono de características presenta mayor ventaja en velocidad y aceleración de convergencia con rangos de 0.5 en adelante. El algoritmo tradicional siempre lleva la ventaja en tiempo de simulación dado que el AH contempla diferentes librerías e implementa un simulador cuántico que efectúa el mismo comportamiento que las computadoras cuánticas de IBM, por eso se establece el mismo ruido al tratar de programar un circuito cuántico, para las mediciones de exactitud y precisión el algoritmo híbrido presenta un incremento en la iteración 100 y 500; al evidenciarse que para la iteración 100 el AH presenta un buen rendimiento a comparación del algoritmo tradicional decimo que nuestro algoritmo es más eficiente para pocas iteraciones sin necesidad de incrementar el número experimentos. Para el caso de estudio 2 la evaluación de las mediciones por medio de las técnicas de optimización fue realizado por medio de las reglas de correspondencia que se mostraron en (3.7)-(3.11); con los resultados obtenidos que se observaron en la Figura 3-12 notamos que empezando en la iteración 100 el AH tiene mayor rendimiento en velocidad, aceleración y exactitud; en los siguientes experimentos 200, 300 el rendimiento del AH se ve superado por el algoritmo tradicional, pero mantiene en ciertos experimentos la ventaja en velocidad y aceleración de convergencia. Para los experimentos restantes se observó que el AH tiene un mejor rendimiento en aceleración, velocidad, exactitud y precisión; la única desventaja que presenta es con respecto al tiempo de simulación.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Se implementó un algoritmo cuántico híbrido que toma como referencia la técnica de muestreo de Monte Carlo. El algoritmo propuesto presenta un buen rendimiento para cualquier tipo de sistema y permite determinar los índices de confiabilidad a nivel HLI. La eficacia ha sido validada hasta el punto de que el AH requiere un menor número de experimentos en comparación con un algoritmo tradicional basado en Monte Carlo, el cual requiere un gran número de experimentos para lograr aproxima el resultado esperado. Con todo lo anterior es correcto afirmar que el AH representa una herramienta que ayuda a reducir la carga computacional que necesitan métodos tradicionales como Monte Carlo.
- El algoritmo es capaz de compilar ambos sistemas propuestos para el nivel HLI que consta de un sistema de 3 generadores y fue necesario para validar el comportamiento del algoritmo propuesto, además del sistema de la IEEE RTS de 32 generadores. En ambos casos de estudios el algoritmo presenta mayor ventaja en velocidad y aceleración de convergencia, teniendo pequeñas desventajas en precesión y exactitud debido a que el método tradicional a medida que se va aumentando su número de experimentos comienza a estabilizarse y sus resultados a converger. Dada la estructura del algoritmo híbrido se puede acoplar cualquier tipo de sistema de nivel HNI además de ampliar el estudio de probabilidades de fallas de un sistema con la implementación de nuevos índices de confiabilidad a diferencia del LOLE, LOLP, EENS, LOLP.
- El cálculo de los índices de confiabilidad permitió validar el funcionamiento del AH a través de la comparación de los resultados con el método tradicional de Monte Carlos con lo cual se evidenció la similitud que existen entre los dos algoritmos propuestos en la convergencia de los datos.
- El tiempo de simulación en el algoritmo híbrido aumenta de manera lineal a mayor número de experimentos; debido a la simulación de ruidos e interferencias similares a los de una computadora cuántica real, con el uso del

simulador cuántico Aer_simulation correspondiente a uno de los simuladores principales donde se incluye QAS, disminuyó el tiempo de simulación en el AH en comparación al simulador State Vector y Stabilizer que presentaban mayor tiempo de simulación.

4.2 RECOMENDACIONES

- Para poder mejorar el tiempo de simulación del AH se tiene que realizar más estudios respecto a el uso de computadoras y programación cuántica con el objetivo de tratar de priorizar el número de qubits en todo el sistema, al disminuir el uso de qubits el algoritmo sería capaz de procesar en menor tiempo la convergencia de los resultados.
- Por medio de la librería Qiskit.org escoger el simulador que más se asemeje a los requerimientos del algoritmo que se propongo, cada simulador está establecido para diferentes tareas, compuertas y método de convergencia. Dependiendo del simulador se obtendrá una diferencia con el tiempo de simulación, las características de cada simulador se encuentran en la sección 2.2.1.
- Investigar acerca del uso de compuertas cuánticas que permitan la optimización del circuito, el uso de computadoras cuánticas va en crecimiento y se puede implementar en cualquier área de estudio para optimizar grandes procesos.

BIBLIOGRAFÍA

- [1] S. R. Castaño, Redes de Distribución de Energía, Manizales : Universidad Nacional de Colombia .
- [2] K. a. K. R. Moslehi, A Reliability Perspective of the Smart Grid, IEEE Transactions on Smart Grid, 2010.
- [3] W. a. o. Li, Reliability assessment of electric power systems using Monte Carlo methods, New York: Springer Science & Business Media, 2013.
- [4] A. G. A. Mass, «Evaluación de Confiabilidad en Sistemas Eléctricos de Distribución,» Pontificia Universidad Católica de Chile, Santiago de Chile , 1994.
- [5] A. Luongo, «Algoritmos cuánticos para el análisis de datos,» 24 04 2022. [En línea]. Available: <https://quantumalgorithms.org/>. [Último acceso: 20 05 2022].
- [6] M. S. A. Alvarado, «Power System Reliability Enhancement with Reactive Power Compensation and Operational Risk Assessment with Smart Maintenance for Power Generators,» The University of Birmingham, Birmingham, 2020.
- [7] N. S. S. Yapo, «Evaluación de la Confiabilidad del Sistema de Generación y Transmisión Considerando Fallas de Causa Común en Líneas,» Universidad Politécnica Salesiana de Quito, Quito, 2021.
- [8] A. M. a. S. M. T. Rej, «Reliability Assessment of the Brazilian Power System Using Enumeration and Monte Carlo,» IEEE Transactions on Power Systems, 2008.
- [9] M. Rausand, y A. Hoyland, System reliability theory: models, statistical methods, and applications, vol. 396, John Wiley & Sons., 2003.
- [10] R. Billinton y R. N. Allan, Reliability Evaluation of Power Systems, Segunda ed., New York: Plenum Press, 1994.
- [11] N. R. Melchor, «Universidad de Salamanca Escuela Técnica Superior de Ingeniería Industrial,» 2021. [En línea]. Available: http://stsproyectos.com/U/S/SIST_01.pdf.
- [12] M. Saraceno, «Computación Cuántica,» UBA-CNEA-CITEFA , 1997.
- [13] A. Montoya, «Colaborativo (IBM presenta la primer computadora cuántica comercial),» [En línea]. Available: <https://colaborativo.net/todo/ciencia-y-tecnologia/ibmq/>. [Último acceso: 05 06 2022].

- [14] I. L. C. Michael A. Nielsen, «Quantum Computation and Quantum Information,» CAMBRIDGE, New York, 2010.
- [15] B. Prieto, «Lenguajes de Programación Cuántico,» 2018.
- [16] J. W. B. Juan Pablo Rúa, «Quantum computing's state of the art,» Escuela de Sistemas e Informática, Medellín, 2009.
- [17] A. a. A. J. a. A. P. a. B. A. a. C. W. a. C. G. a. C. C. a. D. H. a. G. D. a. K. S. a. o. Adedoyin, «Quantum algorithm implementations for beginners,» Los Alamos National Laboratory, Nueva Mexico, 2018.
- [18] A. Montaro, «Quantum speedup of Monte Carlo methods,» Department of Computer Science, University of Bristol,, 2015.
- [19] A. S. N. a. Ž. R. Huda, «Distribution system reliability assessment using sequential multilevel Monte Carlo method,» 2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia, 2016.
- [20] N. N. R. B. a. F. N. F. B. M. a. R. M. I. M. Roslan, «Sequential and Nonsequential Monte Carlo in Assessing Reliability Performance of Distribution Network,» 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), 2020.
- [21] I. QUANTUM, «IBM QUANTUM PROGRAMS,» [En línea]. Available: <https://quantum-computing.ibm.com/services/programs/docs/runtime/programs/start>. [Último acceso: 20 06 2022].
- [22] B. a. S. M. a. M. S. a. G. C. a. R. M. a. G. A. a. T. M. a. S. K. Heim, «Quantum programming languages,» Nature Publishing Group, 2020.
- [23] I. a. N. F. Buluta, «Quantum simulators,» American Association for the Advancement of Science, 2009.
- [24] H. a. L. S.-y. a. W. L.-d. a. Y. Z.-j. a. W. X.-d. Liu, «An optimized algorithm and the verification methods for improving the volumetric error modeling accuracy of precision machine tools,» Springer, 2021.
- [25] M. S. a. A.-C. F. E. a. L.-R. E. A. a. R.-G. C. D. a. V. W. Alvarez-Alvarado, «Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields,» Nature Publishing Group, 2021.

- [26] M. S. a. J. D. Alvarez-Alvarado, «Aging Reliability Model for Generation Adequacy,» 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), 2018.
- [27] C. a. W. P. a. A. P. a. A. R. a. B. M. a. B. R. a. C. Q. a. F. C. a. H. S. a. K. S. a. o. Grigg, «The IEEE reliability test system-1996. A report prepared by the reliability test system task force of the application of probability methods subcommittee,» IEEE, 1999.

ANEXOS

ANEXO A: TABLAS DE ERROR RELATIVO

Tabla A- 1: Resultados referenciales obtenidos de Matlab

Resultados de referencia obtenidos de Matlab	
Índices	NSMC_Matlab
LOLE	9.4444
LOLP	0.1081
EENS	1173.33
XLOL	124.9331

Tabla A- 2: Porcentaje de error del AH con 100 experimentos

Porcentaje de error en los índices de confiabilidad 100 experimentos			
Índices	NSMC_Qubits	NSMC_Matlab	% Error
LOLE	9.31	9.4444	1.423066
LOLP	0.1065	0.1081	1.480111
EENS	1142.8983	1173.33	2.593618
XLOL	124.7676	124.9331	0.132471

Tabla A- 3: Porcentaje de error del algoritmo tradicional con 100 experimentos

Porcentaje de error en los índices de confiabilidad 100 experimentos			
Índices	NSMC_Tradicional	NSMC_Matlab	% Error
LOLE	9.61	9.4444	1.753420
LOLP	0.11	0.1081	1.757632
EENS	1219.13	1173.33	3.903420
XLOL	126.3046	124.9331	1.097788

Tabla A- 4: Porcentaje de error del AH con 5000 experimentos

Porcentaje de error en los índices de confiabilidad 5000 experimentos			
Índices	NSMC_Qubits	NSMC_Matlab	% Error
LOLE	9.446	9.4444	0.016941
LOLP	0.1081	0.1081	0.000000
EENS	1189.6735	1173.33	1.392916
XLOL	126.511	124.9331	1.262996

Tabla A- 5: Porcentaje de error del método tradicional con 5000 experimentos

Porcentaje de error en los índices de confiabilidad 5000 experimentos			
Índices	NSMC_Tradicional	NSMC_Matlab	% Error
LOLE	9.4188	9.4444	0.271060
LOLP	0.1078	0.1081	0.277521
EENS	1183.1144	1173.33	0.833900
XLOL	125.193	124.9331	0.208031

ANEXO B: CÓDIGOS PYTHON™- VISUAL CODE™

```
#CARGA DE DATOS : Gráfica de la curva de demanda
datos=loadmat('loadprofile.mat')
dat_load=datos['loadprofile']
fc.grafic(datos)

#CARGA DE DATOS : DATOS DE LOS GENERADORES
#DATOS TABLA 6
RowName06 = [
    'Unit Type',
    'Unit Size (MW)',
    'Unit Group',
    'Forced Outage Rate',
    'MTTF (Hour)',
    'MTTR (Hour)',
    'Scheduled Maintenance Weeks per year']
Df06 = pd.read_excel(
    "C:\\Users\\Mónica\\Desktop\\TESIS\\Table-06.xls",
    sheet_name='Table-06',
    header = None,
    names = RowName06)

#Extraccion de datos de índices de confiabilidad
GenData06 = Df06.loc[
    :,
    ['Unit Type',
    'Unit Size (MW)',
    'Forced Outage Rate']]

#DATOS TABLA 7
RowName07 = [
    'Bus ID',
    'Unit Type',
    'ID #',
    'PG MW',
    'QG MVAR',
    'Qmax MVAR',
    'Qmin MVAR',
    'vs pu']

Df07 = pd.read_excel(
    "C:\\Users\\Mónica\\Desktop\\TESIS\\Table-07.xls",
    sheet_name='Table-07',
    header = None,
    names = RowName07)
GenData07 = Df07.loc[(Df07['Bus ID'] < 200),['Unit Type','PG MW','Bus ID']]
GenData07 = GenData07.loc[(GenData07['Unit Type'] != 'Sync Cond')]
GenData = pd.merge(
    GenData06,
    GenData07,
    how="right",
    on="Unit Type")
```



```

GenData = GenData.to_numpy()
#Potencia de Generadores
GenPot = GenData[:,1]
#Numero de Generadores
Ng = len(GenData)
#FOR de Generadores
GenFOR = GenData[:,2]
#Disponibilidad de Generador
GenAvi = 1 - GenFOR
#Potencia Total
PotTot=sum(GenPot)
#Perfil de Carga
PCarga = 2850
Pload=(PCarga*np.array(dat_load))/100

```

```

#Ángulos para la compuerta RY
Theta = []
for i in range(0,Ng):
    thetai = (math.asin((math.sqrt(GenAvi[i]))))*2
    Theta.append(thetai)

```

```

#Matriz de estados operacionales
Mop=np.array(Mop)
#Generación disponible
MPot=fc.mult(Mop,GenPot)
Mresult=np.sum(MPot,axis=0)
#Reserva de energía
Reserve_P=Mresult-Pload
#Consecuencia de fallas
ReservePos=fc.diferencia(Reserve_P)
ReservePot=fc.difPot(ReservePos,Reserve_P)
#Condición para comprobar fallas en el sistema
if ReservePos!=0:
    lole_i=len(ReservePos)
    lole.append(lole_i)
    lolp_i=(lole_i/disparos)*100
    lolp.append(lolp_i)
    eens_i=abs(np.sum(ReservePot,axis=0))
    eens.append(eens_i)
    eens_evoli=sum(eens)/(i+1)
    eens_evol.append(eens_evoli)
    xlol_i=eens_i/lole_i
    xlol.append(xlol_i)
lole=np.array(lole)
lolp= np.array(lolp)
eens=np.array(eens)
xlol=np.array(xlol)
LOLE=lole.mean()
LOLP=lolp.mean()
EENS=eens.mean()
XLOL=xlol.mean()
fin=time.time()

```

```

###Detalles del circuito cuántico y probabilidades##
circuit.draw('mpl')
plot_histogram(counts)

```

```

#Medición del tiempo de simulación
time_seconds=fin_inicio;timeTot=datetime.timedelta(seconds=time_seconds)
##### Análisis de resultados #####

#Primer caso EMMS Evol
eensEvol=np.array(eens_evolution);varEvol=var(eensEvol)
promEvol=np.sum(eens_evolution)/len(eens_evolution)
precision=abs(varEvol/promEvol);eens_op=1173.33;accuracy=abs(eens_op-promEvol)
# Speed
speed,result_speed=fc.derivate(eens_evolution)
#Acceleration
aceleration,result_aceleration=fc.derivate(speed)
dataresult={'LOLE':LOLE, 'LOLP':LOLP,'EENS':EENS,
'XL0L':XL0L,'Tiempo':time_seconds,'Exactitud':accuracy,'Precisión':precision,
'velocidad':result_speed,'Aceleración':result_aceleration
}
#Impresión de resultados
for data, result in dataresult.items():
    print(f'{data:15}=> {result:10}')

fc.evol(eens_evolution)

```

Figura B - 1: Algoritmo Híbrido

```

#CARGA DE DATOS : DATOS DE LOS GENERADORES
#DATOS TABLA 6
RowName06 = [
    'Unit Type',
    'Unit Size (MW)',
    'Unit Group',
    'Forced Outage Rate',
    'MTTF (Hour)',
    'MTTR (Hour)',
    'Scheduled Maintenance Weeks per year']
Df06 = pd.read_excel(
    "C:\\Users\\Mónica\\Desktop\\TESIS\\Table-06.xls",
    sheet_name='Table-06',
    header = None,
    names = RowName06)
#Extraccion de datos de indices de confiabilidad
GenData06 = Df06.loc[
    :,
    ['Unit Type',
    'Unit Size (MW)',
    'Forced Outage Rate']]

#DATOS TABLA 7
RowName07 = [
    'Bus ID',
    'Unit Type',
    'ID #',
    'PG MW',
    'QG MVAR',
    'Qmax MVAR',
    'Qmin MVAR',
    'vs pu']
Df07 = pd.read_excel(
    "C:\\Users\\Mónica\\Desktop\\TESIS\\Table-07.xls",
    sheet_name='Table-07',
    header = None,
    names = RowName07)
GenData07 = Df07.loc[(Df07['Bus ID'] < 200),['Unit Type','PG MW','Bus ID']]
GenData07 = GenData07.loc[(GenData07['Unit Type'] != 'Sync Cond')]
GenData = pd.merge(
    GenData06,
    GenData07,
    how="right",
    on="Unit Type")

```

```

#Exportación de datos
GenData = GenData.to_numpy()
#Potencia de Generadores
GenPot = GenData[:,1]

#Numero de Generadores
Ng = len(GenData)

#FOR de Generadores
GenFOR = GenData[:,2]

#Disponibilidad de Generador
GenAvi = 1 - GenFOR

#Potencia Total
PotTot=sum(GenPot)

#Perfil de Carga
PCarga = 2850
Pload=(PCarga*np.array(dat_load))/100

```

```

#Monte Carlo no Secuencial
inicio=time.time()
lole=[]
lolp=[]
eens=[]
eens_evol=[]
xlol=[]
ne=5000;horas=8736 # Número de experimentos y horas de estudio
#Matriz For para el tiempo de estudio
FOR=fc.indexación(GenFOR,horas);
f,c=np.shape(FOR) #obtención del tamaño de la matriz
for i in range(ne):
    #Generación de números aleatorios de probabilidades
    OperacionR=fc.rand(f,horas)
    #Generación de estado operacional
    Operation_g=OperacionR>FOR
    #Generación disponible de acorde a la matriz operacional
    OperationP=fc.Multiplicacion(Operation_g,GenPot)
    OperationTot=np.sum(OperationP,axis=0)
    #Reserva de energia
    OperationResult=OperationTot-Pload
    #Consecuencia de fallas
    ReservPos=fc.diferencia(OperationResult)
    #Diferencia de potencia para suplir la demanda
    ReservPot=fc.difPot(ReservPos,OperationResult)
    #Condición para comprobar fallas
    if ReservPos!=0:
        #Loss of Load Expectation
        lole_i=len(ReservPos)
        lole.append(lole_i)
        #Loss of Load Probability
        lolp_i=(lole_i/horas)*100
        lolp.append(lolp_i)

```

```

#Expected Energy no supplied
eens_i=abs(np.sum(ReservPot,axis=0))
eens.append(eens_i)
eens_evoli=sum(eens)/(i+1)
eens_evol.append(eens_evoli)
#Expected Loss of Load
xlol_i=eens_i/lole_i
xlol.append(xlol_i)
lole=np.array(lole)
lolp= np.array(lolp)
eens=np.array(eens)
xlol=np.array(xlol)
LOLE=lole.mean()
LOLP=lolp.mean()
EENS=eens.mean()
XLOL=xlol.mean()
fin=time.time()

```

```

time_seconds=fin-inicio;timeTot=datetime.timedelta(seconds=time_seconds)
##### Análisis de resultados #####

#Primer caso EENS Evol
eensEvol=np.array(eens_evolution);varEvol=var(eensEvol)
promEvol=np.sum(eens_evolution)/len(eens_evolution)
precision=abs(varEvol/promEvol);eens_op=1173.33;accuracy=abs(eens_op-promEvol)
# Speed
speed,result_speed=fc.derivate(eens_evolution)
#Acceleration
acceleration,result_acceleration=fc.derivate(speed)
dataresult={'LOLE':LOLE, 'LOLP':LOLP, 'EENS':EENS,
'XL0L':XL0L, 'Tiempo':time_seconds, 'Exactitud':accuracy, 'Precisión':precision,
'Velocidad':result_speed, 'Aceleración':result_acceleration
}
for data, result in dataresult.items():
    print(f'{data:15}>=> {result:10}')

fc.evol(eens_evolution)

```

Figura B - 2: Método Tradicional