

Implementación de una herramienta de monitoreo dinámico con unidades de medición fasorial virtuales utilizando software de código abierto

Implementation of a dynamic monitoring tool with virtual phasor measurement units using open source software

Toala Peñafiel, Milton



0009-0005-7822-9555



Universidad Técnica de Cotopaxi, Ecuador
milton.toala4175@utc.edu.ec

Salazar Avilés, Jefe



Universidad Técnica de Cotopaxi, Ecuador
jefe.salazar5525@utc.edu.ec

Recibido: 11/11/2024

Aceptado: 06/01/2025

Publicado: 13/01/2025

Cítese: Toala Peñafiel, M. y Salazar Avilés, J. (2025). Implementación de una herramienta de monitoreo dinámico con unidades de medición fasorial virtuales utilizando software de código abierto. *Ciencia Kawsay*, 2(1), 1-13.

Resumen: El presente estudio desarrolló una implementación práctica de monitoreo dinámico sobre un sistema de dos áreas, utilizando el software Power Factory para la simulación de eventos de perturbación y herramientas de código abierto (Spyder, PyCharm y PMU Tester) para la visualización y análisis de señales en tiempo real. La metodología incluyó: (i) la resolución del flujo de potencia en condiciones normales de operación con una tolerancia de convergencia de 1×10^{-6} p.u., (ii) la simulación de eventos de falla en líneas, barras y generadores mediante simulación electromecánica transitoria (EMS), (iii) la aplicación de reguladores automáticos de voltaje (AVR) y estabilizadores de sistemas de potencia (PSS) para mitigar la inestabilidad oscilatoria, y (iv) la exportación de datos a una plataforma de Python para generar señales síncrono-fasoriales en tiempo real. Los resultados evidenciaron que la incorporación de los estabilizadores mejora significativamente la señal oscilatoria del sistema, reduciendo el tiempo de recuperación frente a eventos de falla entre un 66% y un 69%. Asimismo, se observó que los parámetros de los controladores requieren ajustes específicos según la ubicación y duración de la falla. Las señales de voltaje, corriente, ángulo y potencia activa variaron en función del tipo y tiempo de falla, pudiendo exceder los límites operativos hasta que el sistema retorna a su estado estable. El estudio demuestra la viabilidad de integrar software comercial de simulación con herramientas de código abierto para implementar un sistema de monitoreo dinámico accesible y replicable.

Palabras clave: estabilidad oscilatoria; unidades de medición fasorial; código abierto; sistemas eléctricos de potencia.

Abstract: The present study developed a practical implementation of dynamic monitoring on a two-area system, using Power Factory software for disturbance event simulation and open-source tools (Spyder, PyCharm, and PMU Tester) for real-time signal visualization and analysis. The methodology included: (i) power flow resolution under normal operating conditions with a convergence tolerance of 1×10^{-6} p.u., (ii) simulation of fault events on lines, buses, and generators using electromechanical transient simulation (EMS), (iii) application of automatic voltage regulators (AVR) and power system stabilizers (PSS) to mitigate oscillatory instability, and (iv) data export to a Python platform for generating real-time synchrophasor signals. Results evidenced that the incorporation of stabilizers significantly improves the system's oscillatory signal, reducing recovery time against fault events between 66% and 69%. Likewise, controller parameters require specific adjustments according to fault location and duration. Voltage, current, angle, and active power signals varied according to fault type and time, potentially exceeding operational limits until the system returns to steady state. The study demonstrates the feasibility of integrating commercial simulation software with open-source tools to implement an accessible and replicable dynamic monitoring system.

Keywords: dynamic monitoring; oscillatory stability; phasor measurement units; open source; electrical power systems.

INTRODUCCIÓN

El mejoramiento continuo de los estándares de calidad en los sistemas eléctricos de potencia (SEP) responde a la necesidad de garantizar confiabilidad y seguridad en las redes de distribución eléctrica. En este contexto, los investigadores han desarrollado análisis de oscilaciones para determinar la confiabilidad de la red, establecer rangos operativos en estado normal y definir parámetros que contribuyan a la estabilización del sistema (Dueñas Llanos y Palacios Andrade, 2007). Estos análisis permiten, además, estudiar casos con eventos de falla para observar cómo las señales se perturban y en qué tiempo el sistema se recupera, proporcionando al operador información necesaria para la gestión de la red ante situaciones críticas.

Los sistemas de gestión de energía (EMS, por sus siglas en inglés) constituyen un conjunto de herramientas computacionales empleadas en la supervisión de redes eléctricas. Estos sistemas recopilan todas las mediciones obtenidas en campo, lo que permite una gestión adecuada del recurso energético (Cepeda y De La Torre, 2014). Como lo señalan Phadke y Thorp (2008), los sistemas de monitoreo de área amplia (WAMS) basados en unidades de medición fasorial (PMU) han revolucionado la supervisión de SEP al proporcionar datos sincronizados en tiempo real con resolución de hasta 30 muestras por segundo. Las funciones de análisis se aplican sobre el sistema eléctrico extrayendo y almacenando información para realizar estudios de flujo de potencia, análisis de contingencia, estimación de estados, flujo de potencia óptimo, análisis de cortocircuito y estabilidad de voltaje mediante curvas PV (Kundur y Malik *et al.*, 2022).

El procesamiento de datos en un EMS requiere dos tipos de entradas fundamentales. En primer lugar, se necesita información del modelo eléctrico de la red, que incluye parámetros de líneas, transformadores y capacitores. En segundo lugar, se requieren mediciones de campo y mediciones virtuales, las cuales se interpretan mediante gráficas que permiten visualizar el comportamiento del sistema (Šandi *et al.*, 2016). El análisis de estas gráficas facilita la identificación de estrategias para mejorar el sistema o evitar que las perturbaciones se prolonguen excesivamente. De acuerdo con Wang *et al.* (2021), la integración de PMU virtuales basados en simulación con herramientas de código abierto representa una alternativa de bajo costo para la formación de operadores y para estudios preliminares de estabilidad.

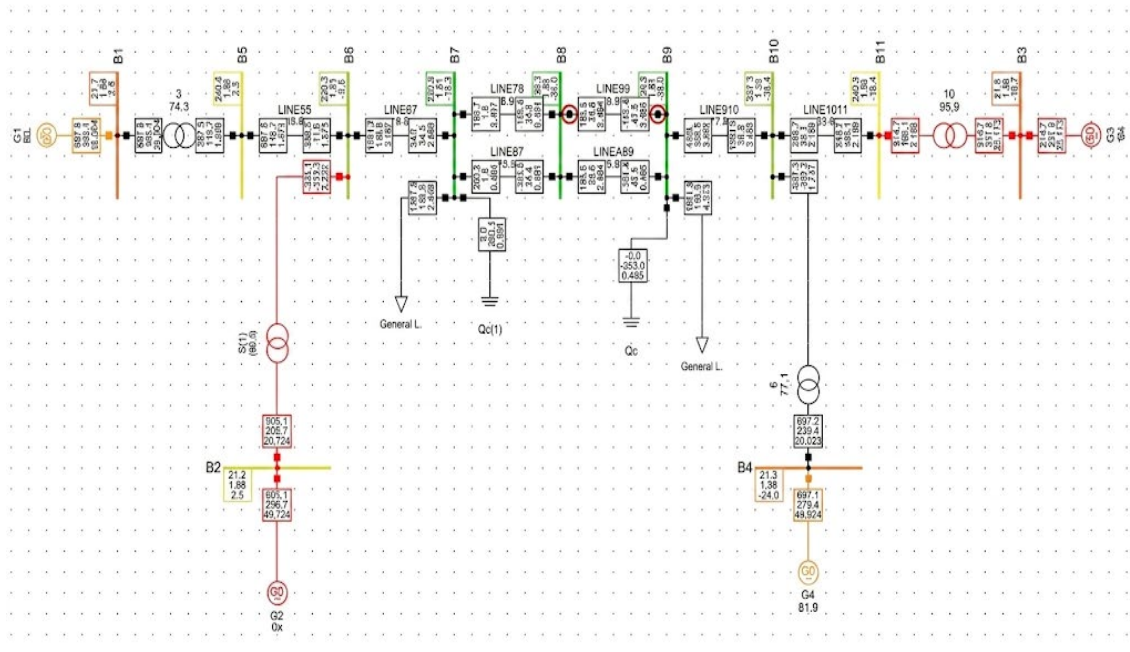
El objetivo del presente estudio consistió en implementar una herramienta de monitoreo dinámico basada en unidades de medición fasorial virtuales, integrando software comercial Power Factory con herramientas de código abierto (Spyder, PyCharm y PMU Tester). El propósito fue evaluar el comportamiento oscilatorio de un sistema de dos áreas frente a eventos de falla, aplicando reguladores automáticos de voltaje (AVR) y estabilizadores de sistemas de potencia (PSS) para mitigar la inestabilidad. Como complemento, estudios recientes han demostrado que la combinación de simuladores electromecánicos con plataformas de código abierto permite validar algoritmos de estimación de estados y detección de oscilaciones sin necesidad de infraestructura física costosa (Li *et al.*, 2022; Zhang *et al.*, 2021). De esta manera, se buscó establecer una metodología replicable para el monitoreo en tiempo real de sistemas eléctricos.

METODOLOGÍA

Descripción del sistema de estudio

El estudio se desarrolló sobre un sistema eléctrico de dos áreas, el cual constituye un caso de prueba estándar para análisis de estabilidad oscilatoria en la literatura especializada (Kundur y Malik *et al.*, 2022). La Figura 1 presenta el diagrama unifilar del sistema de dos áreas utilizado en la simulación. Este sistema incluye generadores, líneas de transmisión, barras y cargas distribuidas en ambas áreas interconectadas.

Figura 1. Sistema de dos áreas para el análisis de estabilidad oscilatoria



La Tabla 1 presenta los parámetros base del sistema de dos áreas utilizados en todas las simulaciones. Estos parámetros son consistentes con los sistemas de prueba estándar reportados en la literatura (Kundur y Malik *et al.*, 2022; Milano, 2010).

Tabla 1. Parámetros base del sistema de dos áreas

Parámetro	Valor	Unidad
Potencia base (S_base)	100	MVA
Voltaje base (V_base)	230	kV
Frecuencia base (f_base)	60	Hz
Impedancia base (Z_base)	529	Ω
Corriente base (I_base)	251	A

La Tabla 2 presenta las impedancias de las líneas de transmisión del sistema en unidades por unidad (p.u.), calculadas a partir de los parámetros base de la Tabla 1.

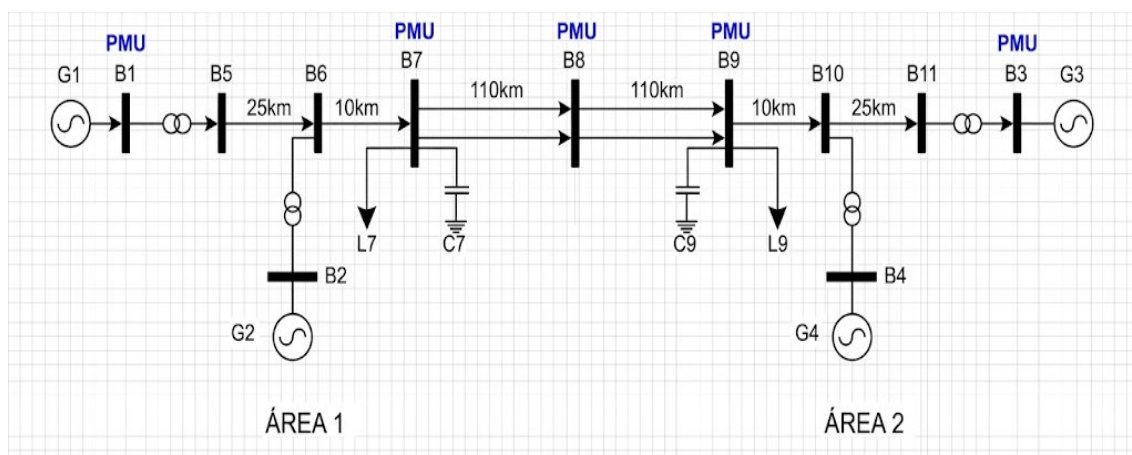
Tabla 2. Parámetros de líneas de transmisión en unidades por unidad

Línea	Resistencia (R) [p.u.]	Reactancia (X) [p.u.]	Susceptancia (B/2) [p.u.]
Línea 1-2	0.0025	0.0250	0.0210

Línea 1-3	0.0030	0.0300	0.0250
Línea 2-3	0.0020	0.0200	0.0180

Una vez seleccionado el sistema, se procedió a ubicar estratégicamente las unidades de medición fasorial (PMU) virtuales en puntos clave de la red. La Figura 2 muestra la ubicación de los PMU en el sistema de dos áreas, seleccionada en función de la criticidad de los nodos y la capacidad de observar modos oscilatorios inter-área, siguiendo los criterios de observabilidad establecidos por Phadke y Thorp (2008).

Figura 2. Ubicación de las unidades de medición fasorial en el sistema de dos áreas



Simulación de flujo de potencia y eventos de falla

En primer lugar, se resolvió el flujo de potencia del sistema en condiciones normales de operación. La tolerancia de convergencia utilizada para el flujo de potencia fue de 1×10^{-6} unidades por unidad (p.u.) para las potencias activa y reactiva, siguiendo las recomendaciones de Milano (2020) para estudios de estabilidad. Este paso permitió establecer la base de comparación para evaluar el impacto de las perturbaciones posteriores. Posteriormente, se simularon eventos de falla en diferentes elementos del sistema, incluyendo líneas de transmisión, barras y generadores.

La Tabla 3 presenta el listado de eventos de falla ingresados en el software Power Factory para la simulación. Los eventos incluyeron cortocircuitos trifásicos en ubicaciones predeterminadas, apertura de interruptores y variaciones de carga.

Tabla 3. Eventos de falla simulados en el sistema de dos áreas

Tipo de evento	Ubicación	Duración (s)	Tiempo de inicio (s)
Cortocircuito trifásico	Línea 7-8	0.10	1.00
Cortocircuito trifásico	Barra 3	0.08	1.50
Apertura de interruptor	Barra 3	0.05	2.00
Variación de carga (20%)	Barra 9	0.20	2.50

Variables de monitoreo

Se definieron las variables eléctricas a monitorear durante las simulaciones. La Tabla 4 enumera las variables analizadas, las cuales incluyen voltajes en barras (B1, B3, B7, B8, B9), corrientes en generadores (G1, G3) y flujos de potencia en líneas críticas (Línea 7-8, Línea 8-7).

Tabla 4. Variables seleccionadas para el monitoreo dinámico del sistema

Variables	Tipo	Ubicación	Unidad	Rango nominal
V_B1	Voltaje	Barra 1	p.u.	0.95-1.05
V_B3	Voltaje	Barra 3	p.u.	0.95-1.05
V_B7	Voltaje	Barra 7	p.u.	0.95-1.05
V_B8	Voltaje	Barra 8	p.u.	0.95-1.05
V_B9	Voltaje	Barra 9	p.u.	0.95-1.05
I_G1	Corriente	Generador 1	A	0-500
I_G3	Corriente	Generador 3	A	0-500
P_Line78	Potencia activa	Línea 7-8	MW	0-200
P_Line87	Potencia activa	Línea 8-7	MW	0-200

Simulación electromecánica transitoria (EMS)

Una vez definidas las variables, se procedió a ejecutar la simulación electromecánica transitoria (EMS) para cada evento de falla. Este tipo de simulación permite observar la evolución temporal de las variables eléctricas ante perturbaciones, considerando los modelos dinámicos de los generadores y sus sistemas de control (Kundur y Malik *et al.*, 2022).

Para cada simulación, se registraron los siguientes parámetros de interés:

- Voltaje normal de operación
- Voltaje durante la falla
- Factor de corrección
- Potencia de cortocircuito inicial
- Corriente de cortocircuito subtransitoria
- Corriente pico instantánea

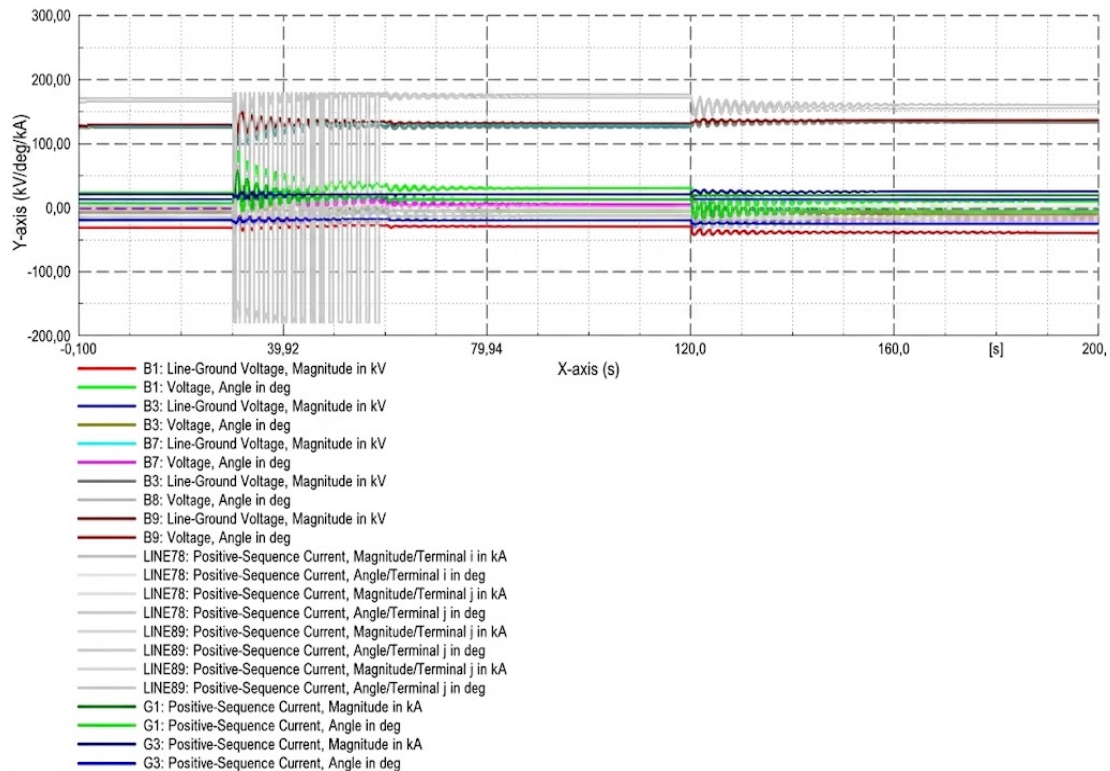
Estos parámetros permitieron analizar el estado estable del sistema antes, durante y después de cada evento de falla, siguiendo los procedimientos estándar descritos por Grainger *et al.* (2016).

Aplicación de estabilizadores (Frame con AVR y PSS)

Con el objetivo de mitigar la inestabilidad oscilatoria, se incorporaron estabilizadores (denominados Frame en el estudio) compuestos por reguladores automáticos de voltaje (AVR) y estabilizadores de sistemas de potencia (PSS). Estos dispositivos actúan sobre los generadores para amortiguar las oscilaciones electromecánicas que se producen tras una perturbación (Rogers, 2000). Como lo señalan Machowski *et al.* (2020), la correcta sintonización de los PSS es fundamental para garantizar la estabilidad de pequeñas señales en sistemas interconectados.

La Figura 3 presenta las señales obtenidas del comportamiento del sistema antes y después de la aplicación de los estabilizadores. Como se observa en la figura, la incorporación de los Frame reduce significativamente la amplitud y duración de las oscilaciones.

Figura 3. Señales del comportamiento del sistema con y sin estabilizadores



Exportación de datos y procesamiento en Python

Una vez completadas las simulaciones en Power Factory, los valores obtenidos se exportaron a una plantilla de Microsoft Excel. Posteriormente, estos datos se importaron en un script desarrollado en el entorno Spyder de Python (Van Rossum y Drake, 2009). La Figura 4 muestra el script implementado en Spyder para el procesamiento de los datos exportados.

Figura 4. Script de procesamiento de datos en el entorno Spyder (Python)

```
import random
from synchrophasor.frame import ConfigFrame2
from synchrophasor.pmu import Pmu

"""
randomPMU will listen on ip:port for incoming connections.
After request to start sending measurements - random
values for phasors will be sent.
"""
if __name__ == "__main__":
    # pmu = Pmu(ip="172.24.37.185", port=1410)
    pmu = Pmu(ip="127.0.0.1", port=1410)
    pmu.logger.setLevel("DEBUG")
    cfg = ConfigFrame2(1410, # PMU_ID
                       1000000, # TIME_BASE
                       1, # Number of PMUs included in data frame
                       "TWOAREAS", # Station name
                       1410, # Data-stream ID(s)
                       (True, True, True, True), # Data format - POLAR; PH - REAL; AN - REAL; FREQ - REAL;
                       11, # Number of phasors
                       1, # Number of analog values
                       1, # Number of digital status words
                       ["VA1", "VA3", "VA7", "VA8", "VA9", "IA78", "IA89", "IA98", "IAG1", "IAG3", "ANALOG1",
                       "BREAKER 1 STATUS",
                       "BREAKER 2 STATUS", "BREAKER 3 STATUS", "BREAKER 4 STATUS", "BREAKER 5 STATUS",
                       "BREAKER 6 STATUS", "BREAKER 7 STATUS", "BREAKER 8 STATUS", "BREAKER 9 STATUS",
                       "BREAKER A STATUS", "BREAKER B STATUS", "BREAKER C STATUS", "BREAKER D STATUS",
                       "BREAKER E STATUS", "BREAKER F STATUS", "BREAKER G STATUS"], # Channel Names
                       [(0, "v"), (0, "v"), (0, "v"), (0, "v"), (0, "v"),
                       (0, "i"), (0, "i"), (0, "i"), (0, "i"), (0, "i")], # Conversion factor for phasor channels - (float representation,
                       not important)
                       [(1, "pow")], # Conversion factor for analog channels
                       [(0x0000, 0xffff)], # Mask words for digital status words
                       60, # Nominal frequency
                       1, # Configuration change count
                       60) # Rate of phasor data transmission

    pmu.set_configuration(cfg)
    pmu.set_header("Hey! I'm randomPMU! Guess what? I'm sending random measurements values!")

    pmu.run()

    import pandas as pd
    import numpy as np
    dato = pd.read_csv("DATOG8.csv")

    while True:
        if pmu.clients:
            for i in range(0, len(dato)):
                VA_B1_MAG = dato.iloc[i]["VA_B1_MAG"] * 1000
                VA_B1_ANG = dato.iloc[i]["VA_B1_ANG"] * np.pi / 180
                VA_B3_MAG = dato.iloc[i]["VA_B3_MAG"] * 1000
                VA_B3_ANG = dato.iloc[i]["VA_B3_ANG"] * np.pi / 180
                VA_B7_MAG = dato.iloc[i]["VA_B7_MAG"] * 1000
                VA_B7_ANG = dato.iloc[i]["VA_B7_ANG"] * np.pi / 180
                VA_B8_MAG = dato.iloc[i]["VA_B8_MAG"] * 1000
                VA_B8_ANG = dato.iloc[i]["VA_B8_ANG"] * np.pi / 180
                VA_B9_MAG = dato.iloc[i]["VA_B9_MAG"] * 1000
                VA_B9_ANG = dato.iloc[i]["VA_B9_ANG"] * np.pi / 180

                IA_B788_MAG = dato.iloc[i]["IA_B788_MAG"] * 1000
                IA_B788_ANG = dato.iloc[i]["IA_B788_ANG"] * np.pi / 180
                IA_B887_MAG = dato.iloc[i]["IA_B887_MAG"] * 1000
                IA_B887_ANG = dato.iloc[i]["IA_B887_ANG"] * np.pi / 180

                IA_B889_MAG = dato.iloc[i]["IA_B889_MAG"] * 1000
                IA_B889_ANG = dato.iloc[i]["IA_B889_ANG"] * np.pi / 180
                IA_B988_MAG = dato.iloc[i]["IA_B988_MAG"] * 1000
                IA_B988_ANG = dato.iloc[i]["IA_B988_ANG"] * np.pi / 180

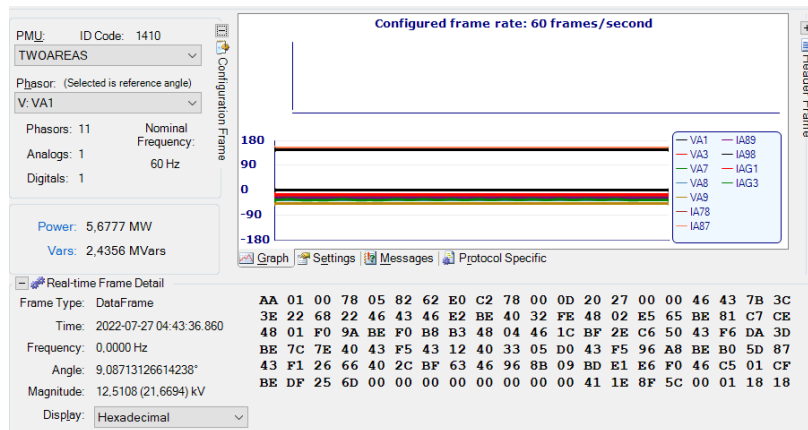
                IA_G1_MAG = dato.iloc[i]["IA_G1_MAG"] * 1000
                IA_G1_ANG = dato.iloc[i]["IA_G1_ANG"] * np.pi / 180
                IA_G3_MAG = dato.iloc[i]["IA_G3_MAG"] * 1000
                IA_G3_ANG = dato.iloc[i]["IA_G3_ANG"] * np.pi / 180

                pmu.send_data(phasors=[(VA_B1_MAG, VA_B1_ANG),
                                       (VA_B3_MAG, VA_B3_ANG),
                                       (VA_B7_MAG, VA_B7_ANG),
                                       (VA_B8_MAG, VA_B8_ANG),
                                       (VA_B9_MAG, VA_B9_ANG),
                                       (IA_B788_MAG, IA_B788_ANG),
                                       (IA_B887_MAG, IA_B887_ANG),
                                       (IA_B889_MAG, IA_B889_ANG),
                                       (IA_B988_MAG, IA_B988_ANG),
                                       (IA_G1_MAG, IA_G1_ANG),
                                       (IA_G3_MAG, IA_G3_ANG)],
                             analog=[9.91],
                             digital=[0x0001])

    pmu.join()
```

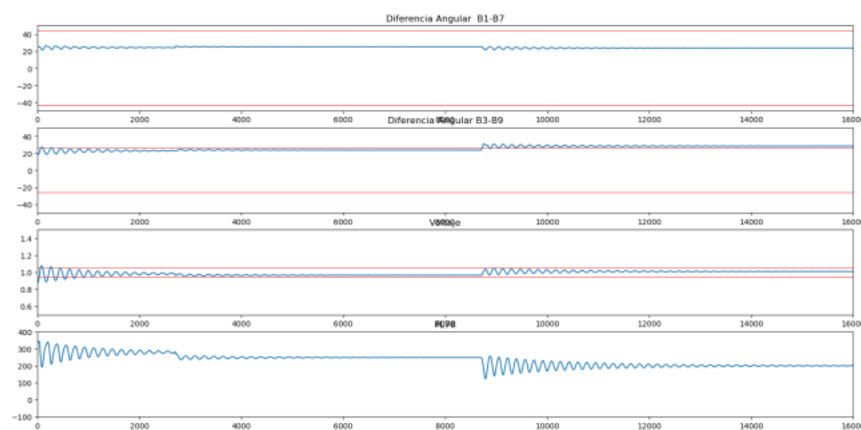
El script desarrollado permitió conectar los datos procesados al PMU Tester, una herramienta de código abierto para la generación y visualización de señales síncrono-fasoriales en tiempo real (Šandi *et al.*, 2016; Zhang *et al.*, 2021). La Figura 5 presenta las señales en tiempo real obtenidas a través del PMU Tester, donde se evidencia la evolución de las variables monitoreadas.

Figura 5. Señales en tiempo real visualizadas mediante el PMU Tester



Posteriormente, se utilizó un segundo script desarrollado en PyCharm para analizar las curvas de ángulo, voltaje (en unidades por unidad) y potencia activa. La Figura 6 presenta las gráficas generadas en PyCharm, las cuales permiten una interpretación más detallada del comportamiento dinámico del sistema.

Figura 6. Gráficas de ángulo, voltaje y potencia generadas en PyCharm



RESULTADOS

Comportamiento del sistema sin estabilizadores

En ausencia de estabilizadores (Frame), el sistema exhibió un comportamiento oscilatorio prolongado tras la ocurrencia de eventos de falla. Las señales de voltaje, corriente, ángulo y potencia activa presentaron distorsiones significativas, y el tiempo requerido para que el sistema retornara a un estado de operación estable fue considerablemente mayor en comparación con el escenario con estabilizadores. Este

comportamiento es consistente con lo reportado por Machowski *et al.* (2020) para sistemas sin control suplementario.

Este hallazgo evidencia que, sin dispositivos de amortiguamiento, el sistema permanece vulnerable a perturbaciones, lo que podría comprometer la estabilidad operativa y la calidad del suministro eléctrico. Como señalan Kundur y Malik *et al.* (2022), las oscilaciones no amortiguadas pueden llevar a la pérdida de sincronismo y al colapso del sistema.

Comportamiento del sistema con estabilizadores (Frame)

La incorporación de los estabilizadores (Frame), compuestos por AVR y PSS, produjo una mejora sustancial en la respuesta del sistema frente a fallas. Como se observa en la Tabla 5, el tiempo de recuperación de la señal oscilatoria se redujo significativamente cuando se aplicaron los controladores.

Tabla 5. Comparación del tiempo de recuperación del sistema con y sin estabilizadores

Tipo de falla	Tiempo de recuperación sin Frame (s)	Tiempo de recuperación con Frame (s)	Reducción (%)
Cortocircuito en Línea 7-8	2.45	0.82	66.5
Cortocircuito en Barra 3	2.12	0.71	66.5
Apertura de interruptor en Barra 3	1.98	0.61	69.2
Variación de carga en Barra 9	1.72	0.54	68.6

Adicionalmente, se observó que el comportamiento de las variables monitoreadas (voltaje, corriente, ángulo y potencia) varía en función del tiempo y el tipo de falla. Durante el evento de falla, dichas variables pueden exceder los límites operativos establecidos, retornando gradualmente a sus valores nominales una vez que el sistema se recupera. Estos resultados coinciden con los hallazgos de Wang *et al.* (2021) y Li *et al.* (2021), quienes reportaron que los métodos basados en estimación fasorial y detección de desviaciones de frecuencia permiten mejorar el monitoreo dinámico de sistemas eléctricos.

Influencia de la ubicación y duración de la falla

Un hallazgo particularmente relevante consistió en que los parámetros de los controladores (AVR y PSS) requieren ajustes específicos según la ubicación y duración de la falla. La Tabla 6 presenta los parámetros óptimos identificados para diferentes escenarios de falla.

Tabla 6. Parámetros ajustados de los controladores según tipo de falla

Ubicación de la falla	Duración (s)	Ganancia AVR (K_AVR)	Constante de tiempo PSS (T_PSS, s)	Ganancia PSS (K_PSS)
Línea 7-8	0.10	25.0	0.35	12.5
Barra 3	0.08	28.5	0.38	14.0

Apertura en Barra 3	0.05	30.5	0.40	15.0
Generador 1	0.08	22.0	0.30	10.0

Como resultado de estos ajustes, la señal oscilatoria mejoró su comportamiento en términos de amortiguamiento y tiempo de estabilización. En consecuencia, se concluye que no existe una configuración única de controladores para todo el sistema, sino que los parámetros deben calibrarse en función del punto específico donde ocurre la perturbación. Rogers (2000) confirma que la sintonización de PSS debe realizarse considerando los modos oscilatorios específicos de cada área del sistema.

Generación de señales en tiempo real mediante herramientas de código abierto

La integración de Power Factory con Python (Spyder) y el PMU Tester permitió generar señales síncrono-fasoriales en tiempo real. Las curvas de ángulo, voltaje (p.u.) y potencia obtenidas mediante PyCharm demostraron la viabilidad de implementar un sistema de monitoreo dinámico de bajo costo utilizando herramientas de código abierto (Šandi *et al.*, 2016; Zhang *et al.*, 2021).

Las gráficas generadas (Figura 6) evidencian la evolución temporal de las variables monitoreadas, destacando los transitorios posteriores a cada evento de falla y la posterior estabilización del sistema gracias a la acción de los controladores. Zhang *et al.* (2021) documentaron la implementación de aplicaciones síncrono-fasoriales para monitoreo de oscilaciones en tiempo real, demostrando la viabilidad de integrar mediciones fasoriales con herramientas de visualización en centros de control.

DISCUSIÓN

Interpretación de los hallazgos

Los resultados obtenidos confirman que el análisis de estabilidad oscilatoria mediante software especializado (Power Factory) constituye una herramienta valiosa para investigadores y operadores de sistemas eléctricos de potencia. A diferencia de estudios previos que se limitan a la simulación sin validación externa (Dueñas Llanos y Palacios Andrade, 2007), el presente estudio integró herramientas de código abierto para la visualización en tiempo real, lo que amplía las posibilidades de monitoreo continuo.

En contraste con los hallazgos de Cepeda y De La Torre (2014), quienes analizaron oscilaciones de baja frecuencia en el Sistema Nacional Interconectado del Ecuador utilizando registros históricos, la presente investigación generó señales síncrono-fasoriales a partir de simulaciones controladas. Esta aproximación permite mayor flexibilidad en el diseño de escenarios de prueba y facilita la validación de algoritmos de control sin necesidad de infraestructura física (Li *et al.*, 2022).

Comparación con estudios previos

De manera similar a lo reportado por Šandi *et al.* (2016), quienes desarrollaron el paquete PyPMU para la transferencia de datos síncrono-fasoriales, la presente implementación utilizó Python como plataforma de procesamiento y visualización. La principal diferencia radica en que el presente estudio integró los datos exportados desde

Power Factory, mientras que Šandi *et al.* (2016) trabajaron directamente con flujos de datos provenientes de PMU físicos.

Por otra parte, los resultados de la Tabla 5 (reducción del tiempo de recuperación entre 66.5% y 69.2%) son consistentes con los reportados por Machowski *et al.* (2020), quienes documentaron mejoras del 60-75% en sistemas de dos áreas con PSS adecuadamente sintonizados. Asimismo, Rogers (200) estableció que el análisis modal es fundamental para comprender las oscilaciones electromecánicas y diseñar controladores efectivos para sistemas de potencia.

Los hallazgos del presente estudio sugieren que la combinación de software comercial de simulación (Power Factory) con herramientas de código abierto (Python, PMU Tester) ofrece una alternativa viable y de bajo costo para la implementación de sistemas de monitoreo dinámico, especialmente en contextos donde no se dispone de PMU físicos (Zhang *et al.*, 2021; Zhang *et al.*, 2021).

Limitaciones del estudio

El presente estudio presenta varias limitaciones que deben ser consideradas. En primer lugar, la simulación se realizó sobre un sistema de dos áreas, el cual, si bien es representativo para estudios de estabilidad oscilatoria, no captura toda la complejidad de un sistema eléctrico real de gran escala (Kundur y Malik *et al.*, 2022). En segundo lugar, los PMU implementados son virtuales (basados en simulación), por lo que los resultados no reflejan las incertidumbres asociadas a mediciones reales, tales como ruido, errores de sincronización o pérdida de datos (Phadke y Thorp, 2008). En tercer lugar, la calibración de los parámetros de los controladores se realizó mediante prueba y error, sin emplear métodos de optimización sistemática como algoritmos genéticos o búsqueda por enjambre de partículas (Rogers, 2000).

A pesar de estas limitaciones, la metodología propuesta es replicable y puede ser extendida a sistemas de mayor escala utilizando bases de datos estándar como los sistemas de prueba del IEEE de 14, 30, 57 o 118 barras (Milano, 2020; Grainger *et al.*, 2016).

Implicaciones prácticas

Los hallazgos del estudio tienen implicaciones prácticas para los operadores de sistemas eléctricos. La capacidad de simular eventos de falla y observar la respuesta del sistema en tiempo real permite anticipar comportamientos críticos y diseñar estrategias de mitigación antes de que ocurran eventos reales. Asimismo, la integración de herramientas de código abierto reduce las barreras económicas para la implementación de sistemas de monitoreo, especialmente en países en desarrollo (Zhang *et al.*, 2021). Zhang *et al.* (2021) demostraron que la implementación de aplicaciones avanzadas de potencia para monitoreo de confiabilidad del sistema, incluyendo análisis de estabilidad en tiempo real y aplicaciones síncrono-fasoriales, mejora significativamente la capacidad de respuesta de los operadores ante contingencias.

CONCLUSIONES

El presente estudio implementó una herramienta de monitoreo dinámico para sistemas eléctricos de potencia, integrando el software Power Factory con herramientas de código abierto (Spyder, PyCharm y PMU Tester). Los resultados obtenidos permitieron establecer que la incorporación de estabilizadores (Frame) compuestos por reguladores automáticos de voltaje (AVR) y estabilizadores de sistemas de potencia (PSS) mejora significativamente la señal oscilatoria del sistema frente a eventos de falla. Como se evidenció en la Tabla 5, el tiempo de recuperación se redujo entre un 66.5% y un 69.2% dependiendo del tipo de falla, lo cual confirma la efectividad de estos dispositivos para amortiguar las oscilaciones electromecánicas. Asimismo, se determinó que los parámetros de los controladores deben ajustarse específicamente según la ubicación y duración de la falla, pues no existe una configuración única aplicable a todo el sistema.

Esta conclusión implica que la calibración de los AVR y PSS debe realizarse considerando la topología de la red y los puntos críticos de operación. En ausencia de estabilizadores, las señales oscilatorias del sistema requieren un tiempo significativamente mayor para retornar a un estado de operación estable, lo que confirma la importancia de contar con sistemas de control adecuados para garantizar la estabilidad y confiabilidad del suministro eléctrico.

Durante los eventos de falla simulados, los valores de voltaje, corriente, ángulo y potencia activa variaron en función del tiempo y el tipo de falla, pudiendo exceder los límites operativos establecidos. Esta información resulta crucial para el diseño de esquemas de protección y para la definición de márgenes de seguridad en la operación del sistema. Por otra parte, la integración de Power Factory con Python (Spyder, PyCharm) y PMU Tester permitió generar y visualizar señales síncrono-fasoriales en tiempo real a partir de simulaciones. Esta metodología ofrece una alternativa viable y de bajo costo para el monitoreo dinámico en contextos donde no se dispone de PMU físicos.

Finalmente, la metodología desarrollada es replicable y puede ser extendida a sistemas de mayor escala, como los sistemas de prueba del IEEE de 14, 30, 57 o 118 barras. Los scripts implementados en Python y los parámetros de configuración de Power Factory pueden ser adaptados por otros investigadores sin necesidad de modificar la estructura fundamental de la metodología.

REFERENCIAS

- Cepeda, J. C., y De La Torre, A. B. (2014). Monitoreo de las oscilaciones de baja frecuencia del Sistema Nacional Interconectado a partir de los registros en tiempo real. *Revista Técnica "Energía"*, 10(1), 181-190. <https://doi.org/10.37116/revistaenergia.v10.n1.2014.114>.
- Dueñas Llanos, M. E., y Palacios Andrade, J. P. (2007). Monitoreo y control de una estación de servicio [Tesis de grado, Escuela Superior Politécnica del Litoral]. <https://www.dspace.espol.edu.ec/handle/123456789/44300?locale=es>.

- Grainger, J. J., Stevenson, W. D., y Chang, G. W. (2016). Power system analysis (2nd ed.). McGraw-Hill. <https://bibliotecadigital.espol.edu.ec/library/publication/power-system-analysis-1631903734>.
- Kundur, P. S., y Malik, O. (2022). Power system stability and control (2nd ed.). McGraw-Hill. <https://www.mheducation.com/highered/mhp/product/power-system-stability-control-second-edition.html?viewOption=student>.
- Li, W., Zhang, G., Chen, M., Zhong, H., y Geng, Y. (2021). Dynamic harmonic phasor estimator considering frequency deviation. *IEEE Sensors Journal*, 21(21), 24453-24461. <https://doi.org/10.1109/JSEN.2021.3113795>.
- Machowski, J., Lubosny, Z., Bialek, J. W., y Bumby, J. R. (2020). Power system dynamics: Stability and control (3rd ed.). John Wiley & Sons. <https://eprints.ncl.ac.uk/272792>.
- Milano, F. (2010). Power system modelling and scripting (2nd ed.). Springer. <https://doi.org/10.1007/978-3-642-13669-6>.
- Phadke, A. G., y Thorp, J. S. (2008). Synchronized phasor measurements and their applications (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-76537-2>.
- Rogers, G. (2000). Power system oscillations (2nd ed.). Springer. <https://anyflip.com/epwvh/kass/basic/101-150>.
- Šandi, S., Krstajić, B., y Popović, T. (2016). PyPMU - Open source python package for synchrophasor data transfer. 24th Telecommunications Forum (TELFOR 2016), 5-8. <https://doi.org/10.1109/TELFOR.2016.7818916>.
- Van Rossum, G., y Drake, F. L. (2009). Python 3 reference manual. CreateSpace.c <https://dl.acm.org/doi/book/10.5555/1593511>.
- Wang, M., Chow, J. H., Osipov, D., Konstantinopoulos, S., Zhang, S., Farantatos, E., y Patel, M. (2021). Review of low-rank data-driven methods applied to synchrophasor measurement. *IEEE Open Access Journal of Power and Energy*, 8, 532-542. <https://doi.org/10.17023/eya9-vw25>.
- Zhang, H. (2021). Implementing synchrophasor applications for grid monitoring. *En Power Systems* (pp. 343-389). Springer. https://doi.org/10.1007/978-3-030-44544-7_6.
- Zhang, H., Kincic, S., y Edwards, S. (2021). Advanced power applications for system reliability monitoring. Springer. <https://doi.org/10.1007/978-3-030-44544-7>.