



Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado

Implementación en Tiempo Real de un Algoritmo de Prony con Mediciones Sincrofasoriales para Identificación Paramétrica Modal

Tesis

Que para obtener el grado de:

Maestro en Sistemas Eléctricos de Potencia

Presenta:

Emanuel Rosas Hernández

Asesor:

Dr. Juan Segundo Ramírez

Coasesor:

Dr. Emilio Barocio Espejo



Agradecimientos

A Dios por darme salud y fuerza para realizar este proyecto.

A mi esposa Fátima y mi hijo Ángel Obed por acompañarme y apoyarme en todas las etapas de este proyecto, por ser comprensivos y prestarme tiempo que era para ellos.

A la empresa Schweitzer Engineering Laboratories por proporcionarme los recursos, el tiempo y la confianza para desarrollar este trabajo de investigación.

A mis padres Manuel y Margarita por enseñarme el valor del trabajo duro y constante.

A mis hermanos Rebeca, Héctor y Arturo por alegrarse de cada uno de mis logros.

A mi asesor el Dr. Juan Segundo Ramírez por compartir sus conocimiento y experiencia en el área de Sistemas de Potencia y siempre estar en la mejor disponibilidad para apoyarme.

A mi Coasesor el Dr. Emilio Barocio Espejo por poner a disposición su experiencia y conocimiento en el área de oscilaciones electromecánicas.

Al Dr. Ciro Alberto Núñez Gutiérrez y la Dra. Nancy Visairo Cruz por todo su apoyo a lo largo de la maestría.

Índice

Introducción	1
I. Planteamiento del Problema.....	1
II. Estado del Arte.....	1
III. Objetivo General.....	2
IV. Objetivos Particulares.....	2
V. Descripción Del Trabajo de Tesis.....	2
1. Capítulo I. Formulación Matemática Del Método De Prony.....	4
1.1. Revisión Del Problema De Oscilaciones Electromecánicas.....	4
1.2. Tipos De Datos Muestreados Usando PMU.....	5
1.3. Detección De Datos Tipo “Ringdown”.....	6
1.4. Método De Prony.....	6
1.4.1. Método Básico De Prony.....	7
1.4.2. Método De Prony Por Ventanas Deslizantes.....	9
1.4.2.1. Efecto De La Frecuencia De Muestreo Y Del Orden En El Desempeño Del Método De Prony.....	10
1.4.2.2. Discriminación Entre Modos Dominantes Y Triviales.....	12
1.5. Algoritmos Para La Implementación Del Algoritmo De Prony.....	13
1.5.1. Algoritmo Para Solución De Polinomios.....	13
1.5.2. Algoritmo Para Evaluación de Polinomios.....	14
1.5.3. Algoritmo Para Degradación De Polinomios.....	15
1.5.4. Solución De Sistemas Sobredeterminados.....	16
1.6. Programación En MATLAB®.....	16
1.7. Ejemplo 1. Aplicación Del Método De Prony.....	17
2. Capítulo II. Plataforma De Pruebas Y Programación.....	21
2.1. Plataforma De Prueba: RTDS + RTAC + PMU.....	21
2.1.1. Ajustes Del Relavador SEL-351S Como PMU.....	24
2.1.2. Configuración De Equipo SEL-3530 Como Controlador En Tiempo Real.....	26
2.2. Estándar IEC-61131-3.....	28
2.3. Bibliotecas Propietarias SEL-RTAC.....	30
2.4. Bloques De Funciones Programadas Por El Usuario.....	32
2.4.1. Bloque De Función Laguerre Para Cálculo de Raíces De Polinomio.....	33
2.4.2. Bloque De Función Evaluación De Polinomio.....	34

2.4.3.	Bloque De Función Degradación De Polinomio.	35
2.5.	Diagrama De Flujo Del Algoritmo Básico De Prony.....	36
3.	Capitulo III. Resultados Experimentales.....	39
3.1.	Prueba Del Algoritmo.....	39
3.2.	Sistema De Prueba RSCAD: Fuente De Tensión Con Modos De Oscilación Conocidos.	41
3.3.	Sistema De Prueba RSCAD: Sistema De Dos Fuentes, Dos Generadores Y Dos Líneas De Transmisión.	43
3.4.	Biblioteca Propietaria SEL-RTAC: SVP Plus.....	47
4.	Conclusiones Y Trabajo Futuro.....	51
	Bibliografía.....	52

Índice De Figuras

Figura 1. Oscilaciones electromecánicas de baja frecuencia.	5
Figura 2. Frecuencias de muestreo disponibles en la PMU.	10
Figura 3. Muestreo de la señal $y(t)$ para valores de (a) $d = 1$, (b) $d=2$	12
Figura 4. Gráfica en dominio del tiempo de señal del Ejemplo 1.	18
Figura 5. Muestreo de señal del Ejemplo 1.....	18
Figura 6. Plataforma de Prueba: RTDS + RTAC + PMU [14]	21
Figura 7. Procesador PB5 [14].....	22
Figura 8. Tarjeta de salidas analógicas GTA0 [14].....	22
Figura 9. Interfaz de Software RSCAD.....	23
Figura 10. SEL-3530 RTAC.....	23
Figura 11. Vista frontal del relevador SEL-351S.....	24
Figura 12. Vista posterior del relevador SEL-351S	24
Figura 13. Configuración de dirección IP de la PMU.	25
Figura 14. Configuración del mensaje PMU.	25
Figura 15. Configuración del puerto 5	26
Figura 16. Integración del equipo SEL-351S al SEL-3530 mediante protocolo C37.118.	27
Figura 17. Configuración del equipo SEL-3530 para recibir mediciones sincrofasoriales desde el equipo SEL-351S.....	27
Figura 18. Configuración de la velocidad de recepción de las mediciones de la PMU.....	28
Figura 19. Declaraciones en lenguaje de texto estructurado.....	29
Figura 20. Bibliotecas dentro de AcSelerator RTAC.	31
Figura 21. Funciones de biblioteca MathMatrix.....	31
Figura 22. Funciones de biblioteca MathComplex	32
Figura 23. Muestreo fuera de línea de función (3.1).....	40
Figura 24. Estimación de ecuación (3.1) con el algoritmo de Prony.	40
Figura 25. Sistema de prueba.....	41
Figura 26. Magnitud y frecuencias de oscilación.....	42
Figura 27. (a) Tensión fase a, (b) fasor de tensión fase a.	42
Figura 28. Estimación de modos de oscilación para la tensión de la Figura 27.....	43
Figura 29. Sistema de prueba en RTDS.....	44
Figura 30. Parámetros de las líneas de transmisión utilizadas.	45
Figura 31. Oscilaciones por variación de carga (a) Potencia Mecánica (b) Fasor de tensión fase a (V) (c) Tensión fase a (kV).....	45
Figura 32. Estimación de modos de oscilación por variaciones de carga utilizando el algoritmo de Prony.	46
Figura 33. Oscilaciones debido a falla de fase a tierra. (a) Potencia (b) Fasor de tensión fase a (c) Tensión fase a.....	46
Figura 34. Estimación de modos de oscilación ante una falla de fase a tierra utilizando el algoritmo de Prony.	47
Figura 35. Estimación de modos de oscilación ante variaciones de carga.....	48

Figura 36. Estimación de modos de oscilación ante variaciones de carga utilizando el algoritmo de Prony.....	49
Figura 37. Estimación de modos de oscilación ante variaciones de carga utilizando la biblioteca SVP Plus.....	49
Figura 38. Estimación de modos de oscilación ante una falla de fase a tierra utilizando el algoritmo de Prony.....	50

Índice De Tablas

Tabla 1. Direcciones IP.....	24
Tabla 2. Configuración PMU.	25
Tabla 3. Factores utilizados en RTDS y PMU.....	42
Tabla 4. Parámetros del modelo de la Figura 29.	44
Tabla 5. Parámetros de entrada para el análisis modal.	47

Introducción

I. Planteamiento del Problema.

Las propiedades dinámicas de los sistemas de potencia se pueden describir a través de sistemas lineales, esto se logra mediante la linealización de las ecuaciones del sistema alrededor de un punto de operación. De manera tradicional se han utilizado estos modelos lineales para realizar estudios de estabilidad basados en análisis de los valores propios, lo cual es válido para pequeñas perturbaciones, sin embargo, cuando se presentan perturbaciones mayores en el sistema, es muy probable que los estados del sistema se alejen mucho del punto de operación alrededor del cual se linealiza, lo cual implica que nuestro modelo lineal pierde validez.

Actualmente existen métodos basados en medición para poder estimar los modos oscilatorios de un sistema de potencia utilizando procesamiento digital de señales. Cada uno de estos modos es descrito por sus propiedades de frecuencia, magnitud, fase y coeficiente de amortiguamiento.

Estos métodos basados en medición tienen la ventaja de que no necesitan conocer el modelo exacto del sistema, sino que utilizan un modelo paramétrico el cual es ajustado mediante técnicas de procesamiento digital de señales para ajustarlo a las mediciones previamente tomadas y poder predecir el comportamiento futuro.

II. Estado del Arte.

En la literatura se han reportado diversos métodos para la identificación modal basados en medición, entre los que destacan el método de Prony, el algoritmo ERA (eigenvalue realization algorithm, por sus siglas en inglés), el método de la matriz Pencil, el método de los mínimos cuadrados de Hankel, entre otros [1].

El método de Prony es uno de los métodos más reportados en la literatura, el cual fue desarrollado por Gaspard Riche de Prony en 1795, propuesto para estudiar la expansión de gases; sin embargo, este método fue reportado por primera vez como una herramienta para análisis modal en los sistemas eléctricos de potencia por Hauerm Demeure y Scharf en 1990 [2].

Diversas variantes del método de Prony han sido reportadas en la literatura, entre ellas está el método de Prony básico, el método de Prony por ventanas deslizantes [3], el método de Prony recursivo [4], el método de Prony modificado [5], el método de Prony de filtrado por energía [6], entre otros.

III. Objetivo General.

Implementación en tiempo real de un algoritmo de Prony con mediciones sincrofasoriales.

IV. Objetivos Particulares.

Dentro del alcance del trabajo de tesis se encuentran los siguientes objetivos:

- Estudio del modelo matemático de Prony.
- Programación en MATLAB® del algoritmo de Prony por ventanas deslizantes.
- Programación en lenguaje IEC-61131-3 del algoritmo de Prony por ventanas deslizantes.
- Validación del algoritmo de Prony fuera de línea en la plataforma RTDS.
- Pruebas HIL (hardware-in-the-loop, por sus siglas en inglés) del algoritmo de Prony utilizando la plataforma RTDS para una fuente de tensión variable de parámetros conocidos.
- Pruebas HIL para un sistema de dos generadores, dos transformadores y dos líneas de transmisión ante variaciones de carga y ante fallas.
- Pruebas del algoritmo de Prony utilizando la biblioteca SVP Plus (Synchrophasor Vector Processor, por sus siglas en inglés) Plus del controlador SEL-3530 RTAC.

V. Descripción Del Trabajo de Tesis.

Este trabajo de tesis está dividido de la siguiente manera:

El Capítulo I presenta el desarrollo matemático del método de Prony y las herramientas necesarias para la ejecución del mismo (funciones para la solución de polinomios, métodos de minimización de error en sistemas sobredeterminados, etc), las cuales deberán ser programadas en los lenguajes MATLAB® e IEC-61131-3. Además, se presenta la simulación en MATLAB® de una función senoidal amortiguada para ejemplificar la aplicación del algoritmo.

El capítulo II describe la parte de hardware y software utilizado en la implementación del método de Prony, la cual se presenta en tres partes, la primera es la plataforma de simulación RTDS, la segunda es el controlador SEL-3530 RTAC y la tercera es el equipo utilizado como PMU.

El Capítulo III presenta la validación del algoritmo de Prony programado, para lo cual se utilizan pruebas fuera de línea en donde se ingresaron muestras de una función conocida de manera manual, además se realizan pruebas de HIL con una fuente de tensión de magnitud constante, a la cual se le sumaron modos de oscilación de magnitud y frecuencia conocidas. Con las dos pruebas anteriormente mencionadas, fue posible validar la operación del algoritmo ya que los parámetros que buscamos obtener son previamente conocidos. Una vez validado el algoritmo se realizaron pruebas HIL con un sistema de potencia de dos generadores, dos transformadores y dos líneas de

transmisión para observar las frecuencias de oscilación que se generan ante escenarios de variación de carga y falla.

Por último, se presentan las conclusiones obtenidas, así como el trabajo futuro propuesto.

1. Capítulo I. Formulación Matemática Del Método De Prony.

1.1. Revisión Del Problema De Oscilaciones Electromecánicas.

Las oscilaciones interárea son aquellas que se presentan cuando un área del sistema oscila contra otra, este tipo de oscilaciones presenta frecuencias características en el rango de 0.1 a 1 Hz y pueden originarse por varias razones, entre ellas el incremento súbito de carga, la pérdida de unidades de generación, el recierre exitoso después de una falla transitoria, etcétera [6].

El problema con este tipo de oscilaciones se presenta cuando el sistema oscila con un amortiguamiento bajo o positivo (oscilaciones crecientes o sostenidas), ya que esto compromete la estabilidad del sistema.

El análisis modal se ha empleado de manera tradicional para el estudio de las oscilaciones electromecánicas en sistemas de potencia, sin embargo, para utilizarlo es necesario tener un conocimiento de los parámetros del sistema con el fin de hacerlo lineal cerca de algún punto de operación, lo cual en la práctica es complicado ya que no siempre se tiene esta información actualizada [7].

Otra aproximación utilizada es la basada en mediciones de variables eléctricas del sistema tales como corrientes y voltajes, en donde éstas son utilizadas para ajustar los parámetros de algún modelo propuesto. Este método tiene la ventaja de no necesitar un conocimiento exacto del sistema como en el análisis modal; sin embargo, es necesario definir ciertos parámetros de ajuste como, por ejemplo, el orden de la aproximación, la frecuencia de muestreo, en número de muestras, etc.

Los métodos basados en mediciones se han utilizado para análisis fuera de línea, no obstante, el auge de las unidades de medición sincrofasorial (PMU, Phasor Measurement Units por sus siglas en inglés) ha permitido utilizarlos para la identificación de modos de oscilación en tiempo real de tal forma que se pueda tomar decisiones con la información obtenida por medición.

Dentro de estos métodos basados en mediciones se encuentran la transformada discreta de Fourier (DFT) [8], los filtros de Kalman [9], la transformada Wavelet [10], el método de Prony [2], entre otros.

En este trabajo de tesis se utiliza el método de Prony para la identificación de oscilaciones electromecánicas en sistemas de potencia utilizando mediciones sincrofasoriales.

El método de Prony se basa en ajustar un conjunto de mediciones a la solución de un modelo lineal que consta de una sumatoria de exponenciales amortiguadas.

1.2. Tipos De Datos Muestreados Usando PMU.

Al utilizar un método como Prony, el cual como se mencionó es basado en mediciones, se debe estar consciente de los diferentes tipos de datos que se obtienen usando PMU's, los cuales de acuerdo a [4] se clasifican en los siguientes:

- Datos Ambientales (*Ambient Data*): Datos del sistema en estado estacionario o cuasiestacionario en donde las mayores perturbaciones se deben a pequeñas variaciones aleatorias en la carga.
- Datos tipo "Ringdown": Este tipo de datos se presentan ante perturbaciones mayores tales como fallas, pérdida de unidades de generación o variaciones considerables de la carga.

La Figura 1 muestra oscilaciones típicas presentes en un sistema de potencia, es necesario identificar cada tipo de datos ya que cada uno de ellos posee diferente información. Debido a que los datos tipo "ringdown" se relacionan con las oscilaciones que son de nuestro interés, debemos ser capaces de identificar cuando estos estén presentes en la señal medida.

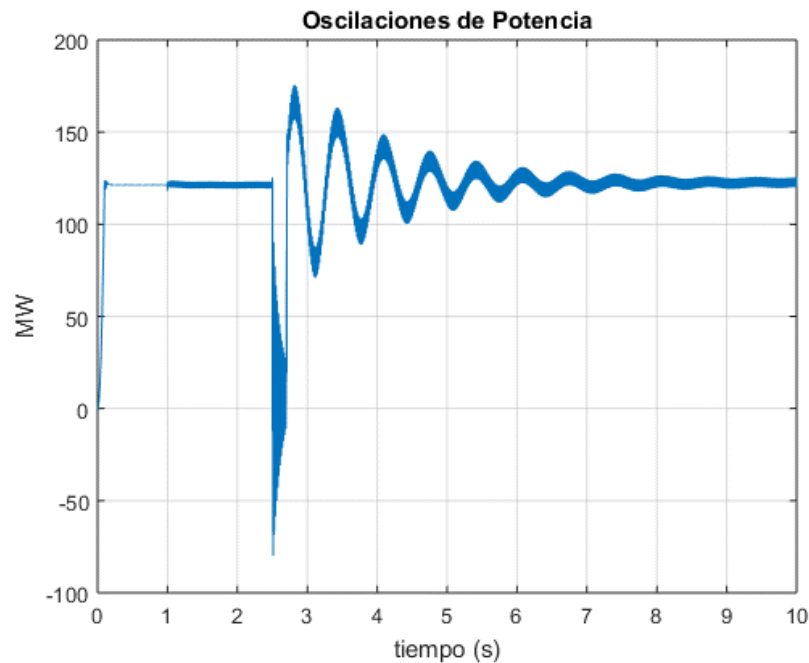


Figura 1. Oscilaciones electromecánicas de baja frecuencia.

La siguiente sección describe dos índices que son utilizados para determinar la presencia de los datos tipo "ringdown", los cuales serán utilizados para dar validez a los resultados entregados por el método de Prony y por ende a la presencia de datos tipo "ringdown".

1.3. Detección De Datos Tipo “Ringdown”.

Como se mencionó en la sección anterior, el algoritmo de Prony está pensado para utilizarse con datos tipo “ringdown”, debido a que estos poseen información sobre las oscilaciones de baja frecuencia presentes en el sistema.

Por lo anterior, es necesario identificar este tipo de datos, para lo cual utilizaremos el nivel de ruido relativo, el cual se define como el porcentaje del ruido con respecto a la energía total [4]:

$$Ruido\ Relativo[k] = \frac{\sqrt{\sum_{j=k}^{k+N-1} |e[j]|^2}}{\sqrt{\sum_{j=k}^{k+N-1} |\hat{y}[j]|^2}} \quad (1.1)$$

En donde $\hat{y}[j]$ es la señal medida y $e[j]$ es la diferencia entre la señal medida y la señal estimada.

Cuando el ruido relativo tiene un valor bajo nos indica que la señal estimada es una buena aproximación de la señal medida y que los parámetros estimados son confiables.

Otro indicador de la medida en que el modelo representa a las mediciones y por lo tanto de la presencia de datos tipo “ringdown” es la relación de señal a ruido (*SNR*, signal-to-noise ratio por sus siglas en inglés), la cual se define de la siguiente manera:

$$SNR = 10 \log_{10} \left(\frac{\sum_{j=k}^{k+N-1} |\hat{y}[j]|^2}{\sum_{j=k}^{k+N-1} |e[j]|^2} \right) \quad (1.2)$$

A diferencia del *Ruido Relativo*, un valor alto de *SNR* indica una buena aproximación del modelo a las mediciones.

Es posible utilizar cualquiera de estos dos índices como indicadores de la validez de los resultados obtenidos con el algoritmo de Prony así como de la presencia de datos tipo “ringdown”.

1.4. Método De Prony.

El método de Prony nos permite ajustar a un modelo paramétrico un conjunto de mediciones de una señal [6]. Este método se basa en caracterizar las respuestas del sistema a una perturbación, la cual puede deberse entre otras causas a la operación exitosa de un recierre posterior a una falla, al ingreso de una carga de potencia elevada, a la pérdida de generación, etcétera.

El método de Prony a diferencia del análisis de Fourier, nos permite obtener además de las frecuencias y magnitudes de las componentes presentes en una señal, sus fases y sus coeficientes de amortiguamiento.

1.4.1. Método Básico De Prony.

Suponga un sistema lineal e invariante en el tiempo de la forma siguiente, el cual es sujeto a una perturbación:

$$\dot{x}(t) = Ax(t) \quad (1.3)$$

Si consideramos que la perturbación es eliminada en tiempo t_0 , entonces el sistema evolucionara de la siguiente manera [2]:

$$y(t) = \sum_{i=1}^n C_i e^{\lambda_i t} \quad (1.4)$$

En donde n es el orden del sistema, λ_i son los valores característicos del sistema y C_i son las magnitudes o residuos de cada una de las respuestas.

El método de Prony utiliza (1.4) para aproximar un conjunto de N mediciones tomadas en intervalos de Δt segundos. De manera tradicional, el método de Prony ha sido utilizado para procesar información fuera de línea debido a que el algoritmo implica operaciones con matrices y solución de polinomios los cuales representan una carga computacional considerable [3].

Con el surgimiento de las unidades de medición sincrofasorial, así como de sistemas de cómputo más potentes, la extracción de información modal en tiempo real se ha vuelto posible [11].

Suponga que se toman muestras de (1.4) en intervalos constantes de Δt segundos, obteniendo la siguiente representación:

$$y[k] = \sum_{i=1}^n C_i e^{(\lambda_i \Delta t)k} \quad (1.5)$$

Considerando que el sistema (1.5) es causal, es decir que los valores de $y[k]$ dependen únicamente de valores previos, obtenemos la siguiente expresión.

$$y[k] + \sum_{i=1}^n b_i y[k - i] = 0 \quad (1.6)$$

Tomando la transformada z de (1.6) y despejando $Y(z)$ se obtiene el polinomio característico, cuyas raíces nos dan la información de los polos del sistema:

$$H(z) = 1 + \sum_{i=1}^n b_i z^{-i} \quad (1.7)$$

Para poder resolver el polinomio característico (1.7) primero es necesario obtener los valores de los coeficientes b_i , para lo cual se hace uso de la ecuación (1.6) evaluada para diferentes valores de k .

Suponiendo que se tienen N muestras de $y[k]$, evaluamos (1.6) para los valores $n, n + 1, \dots, N - 1$; obteniendo el siguiente sistema de ecuaciones:

$$\begin{bmatrix} y[n] \\ y[n+1] \\ y[n+2] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} y[n-1] & y[n-2] & \dots & y[0] \\ y[n] & y[n-1] & \dots & y[1] \\ y[n+1] & y[n] & \dots & y[2] \\ \vdots & \vdots & \ddots & \vdots \\ y[N-2] & y[N-3] & \dots & y[N-n-1] \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (1.8)$$

Se busca que la expresión (1.8) sea un sistema con más ecuaciones que incógnitas (sobredeterminado), por lo que para solucionarlo debemos seleccionar un método de optimización, así como una norma a minimizar, la cual normalmente es una función de tipo energía.

Representaremos (1.8) con la siguiente notación con el fin de presentar una solución general al sistema:

$$y = Ab \quad (1.9)$$

La solución de este tipo de sistemas se revisará en una de las secciones siguientes, por lo cual aquí presentaremos únicamente el resultado.

$$b = (A^T A)^{-1} A^T y \quad (1.10)$$

Utilizando (1.10) encontramos los coeficientes del polinomio característico dado por (1.7), el cual debemos resolver para obtener los polos complejos del sistema.

Una vez encontradas las raíces de (1.7), los coeficientes de amortiguamiento y las frecuencias de oscilación se obtienen de manera directa como se muestra a continuación:

$$\sigma_i = \frac{1}{\Delta t} \text{real}(\ln z_i) \quad (1.11)$$

$$f_i = \frac{1}{2\pi\Delta t} \text{imag}(\ln z_i) \quad (1.12)$$

De manera adicional, es necesario encontrar los valores de C_i en (1.5) empleando el siguiente sistema de ecuaciones:

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_p \\ \vdots & \vdots & \dots & \vdots \\ z_1^{N-1} & z_2^{N-1} & \dots & z_p^{N-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_p \end{bmatrix} \quad (1.13)$$

De manera similar a (1.8), (1.13) es un sistema sobredeterminado que se debe resolver por un método de optimización.

Utilizando la misma norma que en el sistema dado por (1.8), obtenemos lo siguiente:

$$C = [Z^T Z]^{-1} Z^T y \quad (1.14)$$

Con lo anterior se obtiene la parametrización del modelo dado por (1.5), la cual representa nuestra aproximación al conjunto de mediciones.

1.4.2. Método De Prony Por Ventanas Deslizantes.

El método de Prony como se ha mencionado, utiliza un conjunto de N mediciones como información de entrada, las cuales pueden ingresar al algoritmo de distintas formas. Al vector de N mediciones se le conoce como ventana de datos.

Existen dos tipos de ventanas de datos, las ventanas crecientes en las cuales los datos nuevos ingresan a la ventana sin que los datos más antiguos sean eliminados, por otro lado, están las ventanas deslizantes en las cuales los datos del muestreo más reciente son introducidos a la ventana, mientras que los datos más antiguos son eliminados de la misma [3].

El algoritmo presentado en este trabajo de tesis utiliza ventanas de datos del tipo deslizantes debido a que se emplea un número de muestras N , el cual es constante, mismo que nos servirá para proponer los demás parámetros del algoritmo.

El seleccionar un valor de N constante nos permitirá definir criterios para la correcta selección de la frecuencia de muestreo y el orden de aproximación, mismos que deben satisfacer ciertas restricciones con el fin de poder ser empleadas para realizar una correcta estimación.

1.4.2.1. Efecto De La Frecuencia De Muestreo Y Del Orden En El Desempeño Del Método De Prony.

El método de Prony requiere que ciertos parámetros sean definidos cuidadosamente con la finalidad de obtener resultados correctos. Uno de estos parámetros es la frecuencia de muestreo, que en nuestro caso está definida por el valor de los mensajes por segundo de la unidad de medición sincrofásorial PMU.

Una correcta selección de la frecuencia de muestreo debe considerar que el algoritmo de Prony únicamente puede estimar señales cuya frecuencia no sea mayor a la frecuencia de Nyquist [12].

Es decir:

$$f_{muestreo} > 2 \times \text{Ancho de banda} \quad (1.15)$$

Considerando ancho de banda de 1 Hz obtenemos la siguiente desigualdad:

$$f_{muestreo} > 2 \text{ Hz} \quad (1.16)$$

El relevador SEL-351S utilizado en este trabajo de tesis como PMU cuenta con los siguientes valores de muestras por segundo como se muestra en la Figura 2: 1, 2, 4, 5, 10, 12, 15, 20, 30 y 60.

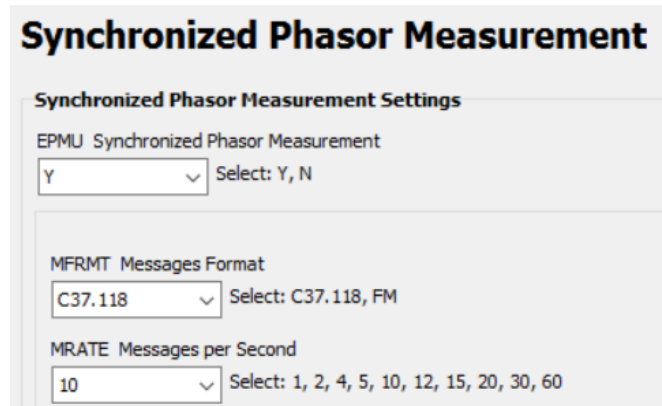


Figura 2. Frecuencias de muestreo disponibles en la PMU.

Se selecciona una frecuencia de muestreo de 10 Hz, la cual es una de las opciones disponibles en el equipo PMU que satisfacen la desigualdad (1.16).

Ahora investigaremos las restricciones impuestas por el algoritmo de Prony a la frecuencia de muestreo y analizaremos qué tanto puede ser disminuida sin afectar el desempeño correcto del algoritmo.

Partiendo de la Ecuación (1.6) podemos introducir un factor de decimación d para disminuir la frecuencia de muestreo y determinar un valor apropiado dentro de las opciones disponibles en la PMU.

$$y[nd + k] + \sum_{i=1}^n b_i y[(n - i)d + k] = 0 \quad (1.17)$$

Armando la matriz (1.8) con base a la Ecuación (1.17) y considerando que el sistema debe ser sobredeterminado para disminuir el efecto adverso del ruido contenido en las mediciones, obtenemos el sistema (1.18) y la desigualdad (1.19), la cual nos indica que el número de filas debe ser mayor que el número de columnas.

$$\begin{bmatrix} y[nd] \\ y[nd + 1] \\ y[nd + 2] \\ \vdots \\ y[N - 1] \end{bmatrix} = \begin{bmatrix} y[(n - 1)d] & y[(n - 2)d] & \dots & y[0] \\ y[(n - 1)d + 1] & y[(n - 2)d + 1] & \dots & y[1] \\ y[(n - 1)d + 2] & y[(n - 2)d + 2] & \dots & y[2] \\ \vdots & \vdots & \vdots & \vdots \\ y[(N - 1) - d] & y[(N - 1) - 2d] & \dots & y[(N - 1 - nd)] \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (1.18)$$

Para satisfacer que el sistema sea sobredeterminado, se debe cumplir la siguiente desigualdad:

$$N - nd > n \quad (1.19)$$

$$d < \frac{N - n}{n} \quad (1.20)$$

Seleccionando los valores $N = 50$, $n = 10$ obtenemos lo siguiente:

$$d < \frac{50 - 10}{10} = 4 \quad (1.21)$$

Considerando la ecuación anterior junto con la Ecuación (1.16) obtenemos las siguientes restricciones:

$$\begin{aligned} d &< 4 \\ f_{\text{muestreo}} &> 2 \text{ Hz} \end{aligned} \quad (1.22)$$

Tomando como base la frecuencia seleccionada de 10 Hz, y considerando que se deben satisfacer las restricciones (1.22) utilizando los valores discretos disponibles de frecuencia de muestreo de la PMU (1, 2, 4, 5, 10, 12,15, 20, 30 y 60), se puede verificar que los únicos valores de d que satisfacen (1.21) son los siguientes:

$$d = \begin{cases} 1 \\ 2 \end{cases} \quad (1.23)$$

Con los cuales obtenemos los valores de frecuencia de muestreo que satisfacen las restricciones impuestas por el algoritmo de Prony:

$$f = \begin{cases} 10 \text{ Hz} \\ 5 \text{ Hz} \end{cases} \quad (1.24)$$

La Figura 3(a) muestra una señal muestreada a 10 Hz y un factor $d = 1$, mientras que la Figura 3(b) muestra la misma señal utilizando un factor $d = 2$.

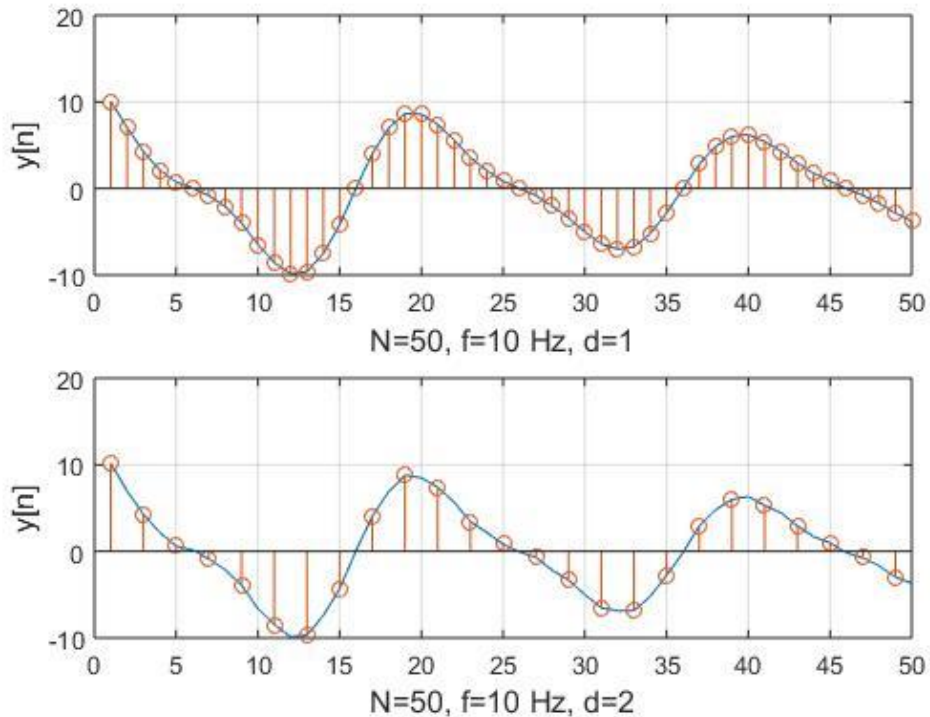


Figura 3. Muestreo de la señal $y(t)$ para valores de (a) $d = 1$, (b) $d=2$.

1.4.2.2. Discriminación Entre Modos Dominantes Y Triviales.

Al utilizar el método de Prony debemos seleccionar un número de polos para aproximar las mediciones al modelo, sin embargo, esta selección generalmente difiere del número de polos realmente presentes en la señal.

El aproximar el modelo con un número arbitrario de polos implica que podríamos identificar polos que no son representativos de las oscilaciones presentes en el sistema, que en caso de presentar un factor de amortiguamiento bajo o positivo (oscilaciones crecientes) nos pueden indicar de manera errónea que el sistema tiende a la inestabilidad.

Debido a lo anterior, es necesario discriminar cuándo un modo es representativo de la oscilación y cuándo se trata de un modo trivial.

Lan Ding presenta tres índices que nos permiten discriminar entre modos dominantes y modos triviales [11].

En este trabajo de tesis utilizaremos un índice basado en la energía de cada modo evaluado en las N muestras del vector de entrada.

$$E_k = \sum_{n=0}^{N-1} |C_k z_k^n|^2 \quad (1.25)$$

El valor de energía obtenido de cada modo nos permitirá discriminar los modos dominantes en la oscilación y descartar los modos triviales que pueden deberse a ruido en la medición.

1.5. Algoritmos Para La Implementación Del Algoritmo De Prony.

1.5.1. Algoritmo Para Solución De Polinomios.

Uno de los pasos del algoritmo de Prony implica la solución de un polinomio grado n , en donde n corresponde al número de polos con los cuales estamos realizando la aproximación.

En este trabajo de tesis se seleccionó el método de Laguerre para la solución de las raíces del polinomio característico [13].

Suponga un polinomio grado n de la forma siguiente:

$$P_n(z) = b_1 z^n + b_2 z^{n-1} + \dots + b_n z + b_{n+1} \quad (1.26)$$

Ahora considere un caso particular en el que el polinomio tiene una raíz en $z = r$, y $n-1$ raíces en $z=q$, entonces el polinomio podría factorizarse de la siguiente manera:

$$P_n(z) = (z - r)(z - q)^{n-1} \quad (1.27)$$

Este caso particular se puede utilizar para plantear un método iterativo para encontrar una raíz a la vez [13].

Diferenciando (1.26) respecto a z obtenemos las siguientes expresiones:

$$P'_n(z) = (z - q)^{n-1} + (n - 1)(z - q)^{n-2}(z - r) \quad (1.28)$$

$$P'_n(z) = P_n(z) \left(\frac{1}{z - r} + \frac{n - 1}{z - q} \right) \quad (1.29)$$

$$\frac{P'_n(z)}{P_n(z)} = \frac{1}{z - r} + \frac{n - 1}{z - q} \quad (1.30)$$

Diferenciando con respecto de z nuevamente y ordenando se obtiene la siguiente expresión:

$$\frac{P_n''(z)}{P_n(z)} - \left[\frac{P_n'(z)}{P_n(z)} \right]^2 = -\frac{1}{(z-r)^2} - \frac{n-1}{(z-q)^2} \quad (1.31)$$

Haciendo las siguientes definiciones:

$$G(z) = \frac{1}{z-r} + \frac{n-1}{z-q} \quad (1.32)$$

$$H(z) = \frac{1}{(z-r)^2} + \frac{n-1}{(z-q)^2} \quad (1.33)$$

Resolviendo (1.32) para $z-q$ y sustituyendo en (1.33) se obtiene una ecuación para $z-r$ cuya solución es la fórmula de Laguerre [13].

$$z-r = \frac{n}{G(z) \pm \sqrt{(n-1)[nH(z) - G^2(z)]}} \quad (1.34)$$

El algoritmo para encontrar las raíces se describe a continuación:

1. Asignar un valor inicial a la raíz buscada.
2. Evaluar $P_n(z)$, $P_n'(z)$ y $P_n''(z)$
3. Calcular $G(z)$ y $H(z)$.
4. Determinar la nueva aproximación de la raíz.
5. Escoger una tolerancia para detener las iteraciones cuando la raíz se aproxime lo suficiente a la raíz real. $|P(z)| < \varepsilon$.
6. Con la raíz obtenida se degrada el polinomio original $P(z)$ de grado n para obtener un nuevo polinomio grado $n-1$.
7. El algoritmo se repite hasta que se obtienen todas las raíces del polinomio.

1.5.2. Algoritmo Para Evaluación de Polinomios.

El método de Laguerre presentado en la sección anterior requiere evaluar el polinomio, así como su primera y segunda derivadas, para lo cual se aprovecha el hecho de que contamos con vector que almacena los coeficientes del polinomio (1.26) para implementar un método iterativo para este fin.

En [13] se muestra el análisis para llegar al siguiente código que evalúa un polinomio y sus primeras dos derivadas en el valor previamente calculado por el método de Laguerre, en donde Pol , $DPol$ y $DDPol$ representan el polinomio, su primera derivada y su segunda derivada evaluados en la raíz z .

```

Pol = b[1]
DPol = 0
DDPol = 0
FOR j := 1 TO n DO
    DDPol := DDPol · z + 2 · DPol
    DPol := DPol · z + Pol
    Pol := Pol · z + b(j + 1)
END

```

El código anterior es programado en un bloque de función que nos sirve para la ejecución del paso (2) del algoritmo de Laguerre.

1.5.3. Algoritmo Para Degradación De Polinomios.

El paso (6) del algoritmo de Laguerre implica que con la raíz obtenida sea degradado el polinomio original para poder proceder con el cálculo de la siguiente raíz.

En [13] se muestra un procedimiento en donde se propone un segundo polinomio grado $n - 1$ el cual al ser multiplicado por $(z - r)$ (en donde r representa la raíz obtenida) es equivalente al polinomio original.

El producto de $(z - r)$ por el nuevo polinomio es igualado al polinomio original para obtener una expresión que relaciona los coeficientes del polinomio nuevo (degradado) con los coeficientes del polinomio original y la raíz z obtenida previamente.

A continuación, se presenta el código utilizado para la degradación del polinomio, en donde b representa el vector de coeficientes del nuevo polinomio y a representa el vector de coeficientes del polinomio original.

```

b[1] := a[1]
FOR i := 2 TO n DO
    b[i] := a[i] + b[i - 1] · z
END

```

El polinomio degradado es utilizado como nueva entrada al algoritmo de Laguerre para obtener la siguiente raíz.

1.5.4. Solución De Sistemas Sobredeterminados.

El algoritmo de Prony trabaja con un vector de muestras de las PMU que generalmente son mayores que el número de polos, un valor típico de N satisface $N > 2n$. Lo anterior implica que para determinar el valor de los coeficientes del polinomio característico debemos resolver un sistema sobredeterminado de $(N - n) \times n$.

$$\begin{bmatrix} y[n] \\ y[n+1] \\ y[n+2] \\ \vdots \\ y[N-1] \end{bmatrix} = \begin{bmatrix} y[n-1] & y[n-2] & \dots & y[0] \\ y[n] & y[n-1] & \dots & y[1] \\ y[n+1] & y[n] & \dots & y[2] \\ \vdots & \vdots & \vdots & \vdots \\ y[N-2] & y[N-3] & \dots & y[N-n-1] \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (1.35)$$

Un sistema sobredeterminado no tiene una solución exacta, en lugar de eso lo que se busca es un vector b que haga que el vector de error $y - Ab$ sea minimizado utilizando alguna norma para evaluar este error.

La ecuación (1.36) muestra la norma utilizada para minimizar el vector de error:

$$J(b) = \|y - Ab\|_2^2 \quad (1.36)$$

Desarrollando la ecuación anterior:

$$J(b) = (y - Ab)^T (y - Ab) \quad (1.37)$$

$$J(b) = y^T y - 2y^T A b + b^T A^T A b \quad (1.38)$$

Derivando la ecuación anterior obtenemos:

$$\frac{\partial}{\partial b} J(b) = -2A^T y + 2A^T A b \quad (1.39)$$

Igualando la derivada a cero para obtener el valor de b que minimice la función de energía:

$$\frac{\partial}{\partial b} J(b) = 0 \Rightarrow A^T A b = A^T y \quad (1.40)$$

$$b = (A^T A)^{-1} A^T y \quad (1.41)$$

1.6. Programación En MATLAB®.

Se utilizó el software MATLAB® para la programación del método de Prony, mismo que se describe a continuación:

- Se utiliza una función de prueba para evaluar el algoritmo de Prony, por ejemplo: $y(t) = 10e^{-0.15t} \cos 4\pi t + 4e^{-0.25t} \cos(12\pi t + \pi/2)$.
- Se selecciona un valor de Δt constante y se evalúa la función de prueba en múltiplos enteros de este valor.
- Se ordena el conjunto de N valores $y(k\Delta t)$ en un vector datos de entrada.
- Se selecciona el orden de la estimación n .
- Se arma la matriz A y el vector y de acuerdo a la ecuación (1.8).
- Se utiliza la pseudo inversa de Moore-Penrose para solucionar el sistema sobredeterminado (1.8).
- Se obtiene el polinomio característico con el resultado del paso anterior.
- Se calculan las raíces del polinomio característico utilizando la función *roots*, la cual realiza los cálculos mediante la obtención de los valores característicos de la matriz acompañante (companion matrix) del polinomio característico.
- Los valores de los coeficientes de amortiguamiento y las frecuencias de oscilación se calculan de manera directa mediante (1.11) y (1.12).
- Con las raíces del polinomio característico se arma el sistema (1.13), mismo que se resuelve utilizando la pseudo inversa de Moore-Penrose.
- La solución del sistema del paso anterior nos da como resultado la magnitud y fase de los modos de oscilación estimados.

A continuación, se presenta un ejemplo de aplicación del método programado en el software MATLAB®.

1.7. Ejemplo 1. Aplicación Del Método De Prony.

Considere la siguiente función para fines de ejemplo:

$$y(t) = 10e^{-0.15t} \cos 4\pi t + 4e^{-0.25t} \cos(12\pi t + \pi/2) [1] \quad (1.42)$$

Como se puede observar en la ecuación (1.42), el sistema tiene dos modos de oscilación a las frecuencias 4π y 12π rad/s, por lo tanto, necesitamos al menos dos pares de polos complejos conjugados en el modelo de Prony, es decir $n = 4$.

El número de muestras se escoge de tal manera que cumplan el criterio $N > 2n$, en nuestro ejemplo consideramos $N = 50$.

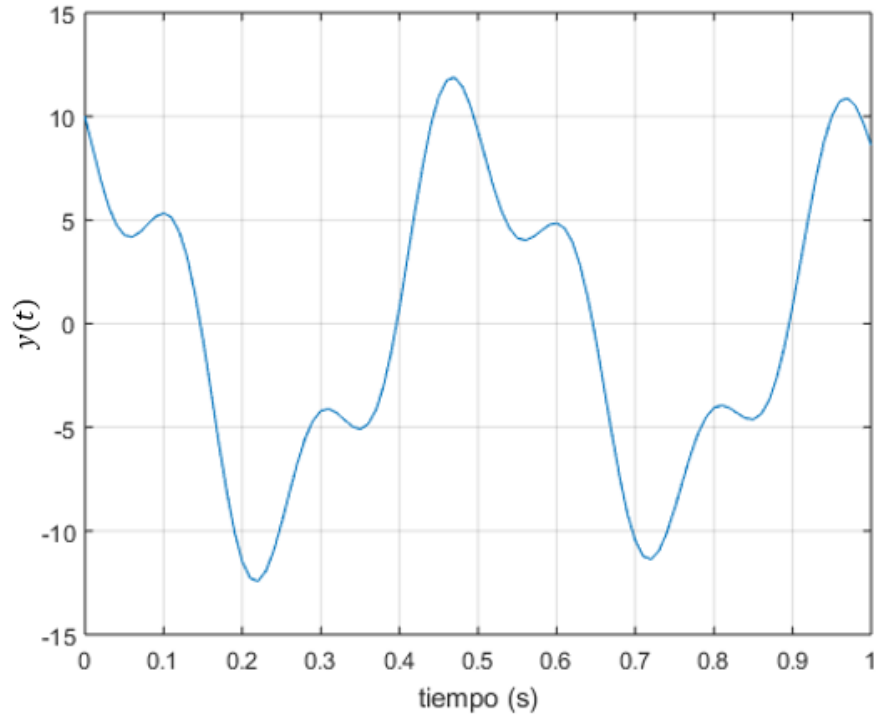


Figura 4. Gráfica en dominio del tiempo de señal del Ejemplo 1.

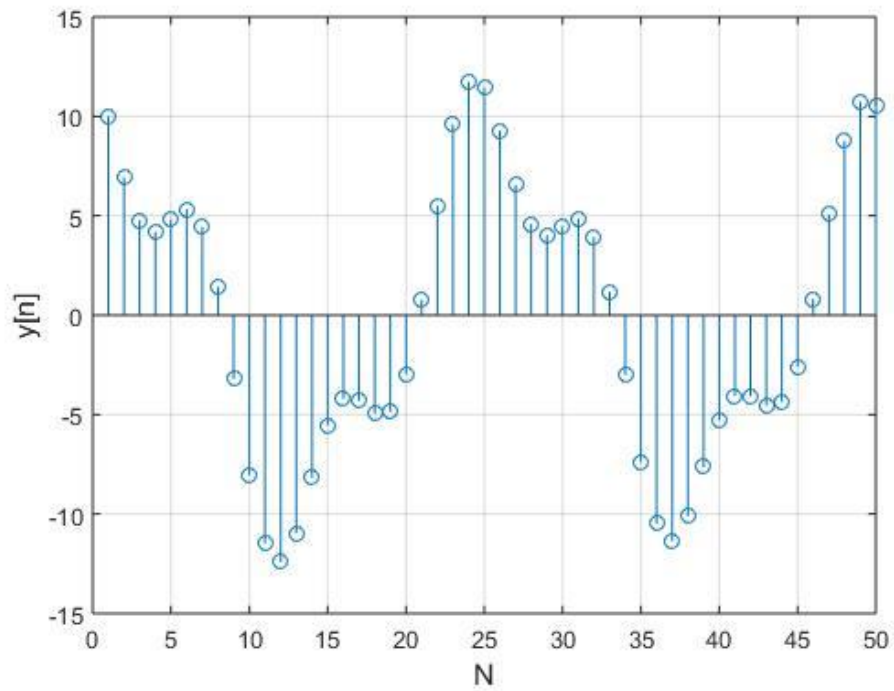


Figura 5. Muestreo de señal del Ejemplo 1.

El siguiente vector corresponde a la evaluación de la función del ejemplo 1 en intervalos de tiempo igual a 0.02 s.

$$y[k] = \begin{bmatrix} 10 \\ 6.9323 \\ 4.7583 \\ \vdots \\ 10.7281 \\ 10.5049 \end{bmatrix} \quad (1.43)$$

Utilizando (1.8) se obtiene la siguiente matriz y vector:

$$A = \begin{bmatrix} 4.1882 & 4.7583 & 6.9323 & 10 \\ 4.8029 & 4.1882 & 4.7583 & 6.9323 \\ 5.3373 & 4.8029 & 4.1882 & 4.7583 \\ \vdots & \vdots & \vdots & \vdots \\ 8.7676 & 5.0659 & 0.8225 & -2.6030 \\ 10.7281 & 8.7676 & 5.0659 & 0.8225 \end{bmatrix} \quad (1.44)$$

$$y = \begin{bmatrix} 4.8029 \\ 5.3373 \\ 4.4297 \\ \vdots \\ 10.7281 \\ 10.5049 \end{bmatrix} \quad (1.45)$$

Resolviendo el sistema $y = Ab$, obtenemos los coeficientes del polinomio característico:

$$b = [3.382 \quad -4.7858 \quad 3.3541 \quad -0.9841]^T \quad (1.46)$$

$$z^4 - 3.382z^3 + 4.7858z^2 - 3.3541z + 0.9841 = 0 \quad (1.47)$$

Resolviendo el polinomio característico se obtienen los siguientes polos en el dominio z :

$$z = [0.7253 \pm 0.6811i \quad 0.9657 \pm 0.2479i]^T \quad (1.48)$$

Finalmente, los valores de α y ω son obtenidos como sigue:

$$\lambda = \frac{\ln(z)}{\Delta t} = \begin{bmatrix} -0.25 + 37.6991i \\ -0.25 - 37.6991i \\ -0.15 + 12.5664i \\ -0.15 + 12.5664i \end{bmatrix} \quad (1.49)$$

Como puede observarse, el algoritmo de Prony estima de manera correcta los modos presentes en la señal del ejemplo 1.

En este ejemplo, la función a aproximar es conocida, así como el número de modos presentes; sin embargo, como ya fue mencionado esta información normalmente no es conocida y tiene que ser propuesta por el usuario.

Ahora realizaremos el mismo ejemplo, pero utilizando un número mayor de polos para verificar que implicaciones tiene esta selección.

Considerando ahora $n = 6$ y agregando una distribución normal de ruido de magnitud máxima igual a 0.1 obtenemos los siguientes polos:

$$z = \begin{bmatrix} -0.99 + 0.1411i \\ -0.99 - 0.1411i \\ 0.7253 + 0.6811i \\ 0.7253 - 0.6811i \\ 0.9657 + 0.2479i \\ 0.9657 - 0.2479i \end{bmatrix} \quad (1.50)$$

$$\lambda = \begin{bmatrix} 3.33 \times 10^{-13} + 1.5 \times 10^2 i \\ 3.33 \times 10^{-13} - 1.5 \times 10^2 i \\ -0.25 + 37.6991i \\ -0.25 - 37.6991i \\ -0.15 + 12.5664i \\ -0.15 + 12.5664i \end{bmatrix} \quad (1.51)$$

Como se puede observar, ahora aparecen un par de polos complejos adicionales (un modo adicional), los cuales tienen un factor de amortiguamiento cercano a cero, pero positivo, es decir inestable.

Ahora utilizaremos el cálculo de energía para determinar cuáles de los 6 modos son dominantes y cuales son modos triviales originados por el ruido presente en la señal.

$$E = \begin{bmatrix} 0.0235 \\ 0.0235 \\ 156.6021 \\ 156.6021 \\ 1076.7 \\ 1076.7 \end{bmatrix} \quad (1.52)$$

Como se puede verificar, los modos 3, 4, 5 y 6 son los modos dominantes debido a que presentan una cantidad de energía que es varios ordenes de magnitud mayor que los modos 1 y 2. Se puede observar también que la energía asociada a los polos “inestables” es la menor de todos. Esto se debe a que los coeficientes C_k son pequeños para esos polos, lo que indica también que no contribuyen significativamente en la señal sintetizada.

Este ejemplo muestra la utilidad de las funciones de energía para poder discriminar entre modos dominantes y triviales.

El procesador PB5 mostrado en la Figura 7 es utilizado para solucionar el modelo de red. Actualmente RTDS cuenta con una nueva generación de procesadores llamados Novacore®.



Figura 7. Procesador PB5 [14].

En las pruebas de HIL (Hardware-in-the-loop, por sus siglas en inglés) es necesario enviar señales analógicas hacia los equipos de prueba, para lo cual se utiliza la tarjeta GTA0 (Giga-Transceiver Analogue Output Card, por sus siglas en inglés), la cual permite aislar ópticamente las señales del simulador hacia los equipos en prueba.

La tarjeta PB5 es la encargada de enviar las señales hacia las tarjetas de salidas GTA0 mediante fibra óptica.



Figura 8. Tarjeta de salidas analógicas GTA0 [14].

El simulador RTDS utiliza el software RSCAD®, en el cual es modelado el sistema de prueba y son configuradas las salidas y entradas al sistema.

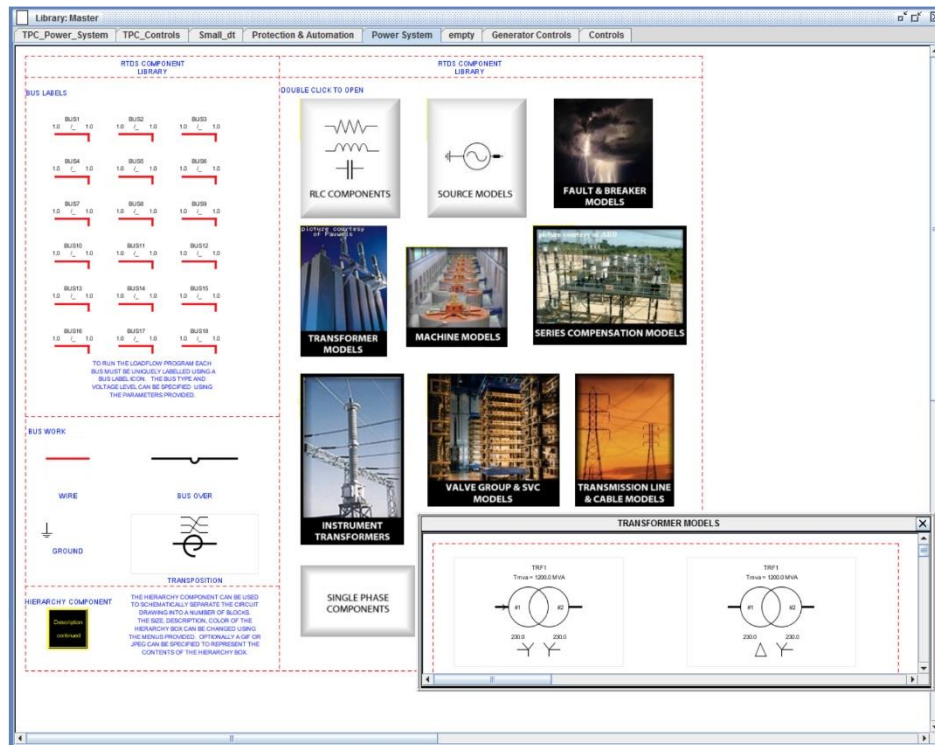


Figura 9. Interfaz de Software RSCAD.

De manera adicional al sistema RTDS, se utilizó un equipo para la programación del algoritmo de Prony. El equipo seleccionado fue el controlador en tiempo real SEL-3530 RTAC, el cual cuenta con el protocolo IEEE C37.118 para integrar mediciones sincrofatorias del equipo PMU.

Además, el equipo SEL-3530 RTAC cuenta con un motor lógico embebido IEC 61131-3 que permite realizar la programación del algoritmo de Prony basado en este estándar [16].

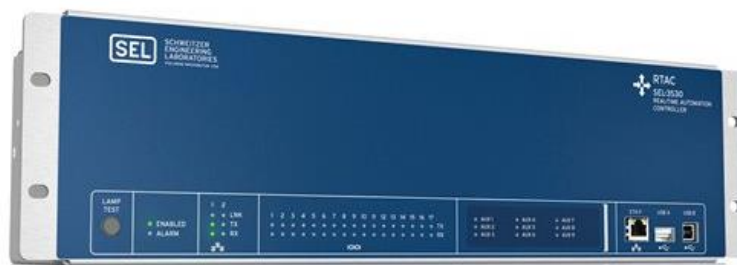


Figura 10. SEL-3530 RTAC

De manera adicional fue necesario utilizar un equipo que reciba las mediciones analógicas de tensión y la señal de sincronización para generar las mediciones sincrofásiales requeridas por el algoritmo de Prony para esta aplicación en tiempo real. Para este fin utilizamos el relevador SEL-351S mostrado en la Figura 11.

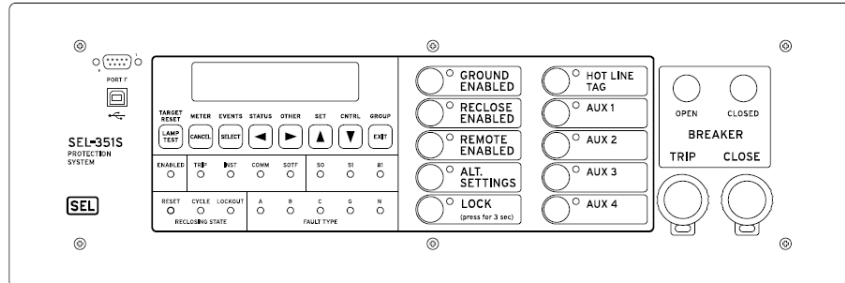


Figura 11. Vista frontal del relevador SEL-351S

El relevador SEL-351S cuenta con un puerto 10/100 Base-T, un puerto EIA-485, dos puertos seriales EIA-232 y un puerto IRIG-B [17]. El firmware del relevador SEL-351 incluye los protocolos IEEE C37.118 para sincrofasores, DNP3, Modbus TCP/RTU, Telnet, SNTP.



Figura 12. Vista posterior del relevador SEL-351S

2.1.1. Ajustes Del Relevador SEL-351S Como PMU.

Para realizar la configuración del relevador SEL-351S, utilizamos el software AcSelerator® QuickSet [18], el cual además de permitirnos configurar los ajustes de protección y las lógicas de control, nos permite configurarlo para ser utilizado como unidad de medición sincrofásial.

Previo a configurar el relevador SEL-351S, debemos asignar direcciones IP en el mismo segmento de red a los equipos que pertenecerán a la misma.

Tabla 1. Direcciones IP

Equipo	Dirección IP	Mascara de subred
PMU (SEL-351S)	148.224.57.64	255.255.255.0
RTAC (SEL-3530)	148.224.57.120	255.255.255.0
COMPUTADORA	148.224.57.100	255.255.255.0

El puerto 5 es configurado con la dirección IP asignada para el equipo PMU mostrada en la Tabla 1.

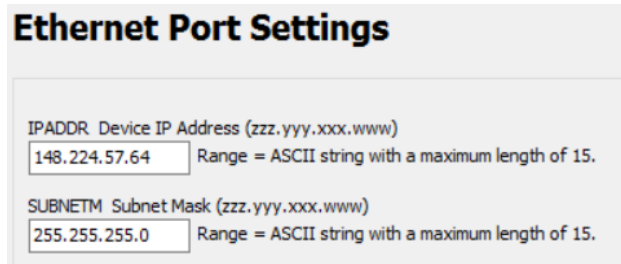


Figura 13. Configuración de dirección IP de la PMU.

Ahora es necesario configurar los parámetros mostrados en la Tabla 2, en los cuales se especifica el formato del mensaje, la velocidad de transmisión, el número de identificación del equipo PMU y el número del puerto físico de salida.

Tabla 2. Configuración PMU.

Equipo	Formato del Mensaje	Mensajes/Segundo (MRTE)	Número de Identificación (PMUID)	Puerto de Salida
PMU (SEL-351S)	C37.18	10	8	4782

La Figura 14 muestra cómo configurar el equipo PMU utilizando el software AcSelerator QuickSet, esta misma configuración debe ingresarse en el controlador SEL-3530 RTAC para que los equipos puedan comunicarse.

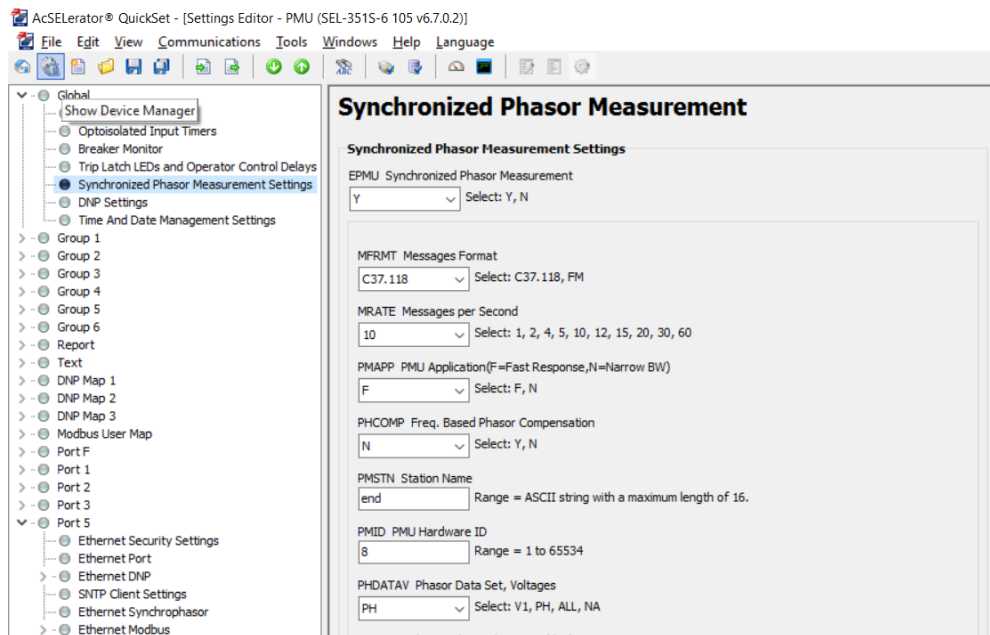


Figura 14. Configuración del mensaje PMU.

La Figura 15 muestra la configuración del puerto 5, en donde se debe ingresar la dirección IP del equipo que recibirá las mediciones de la PMU, así como el número de puerto de salida. Esta información debe corresponder con la configuración de la PMU cuando se integre al equipo SEL-3530 RTAC, como se mostrará en la siguiente sección.

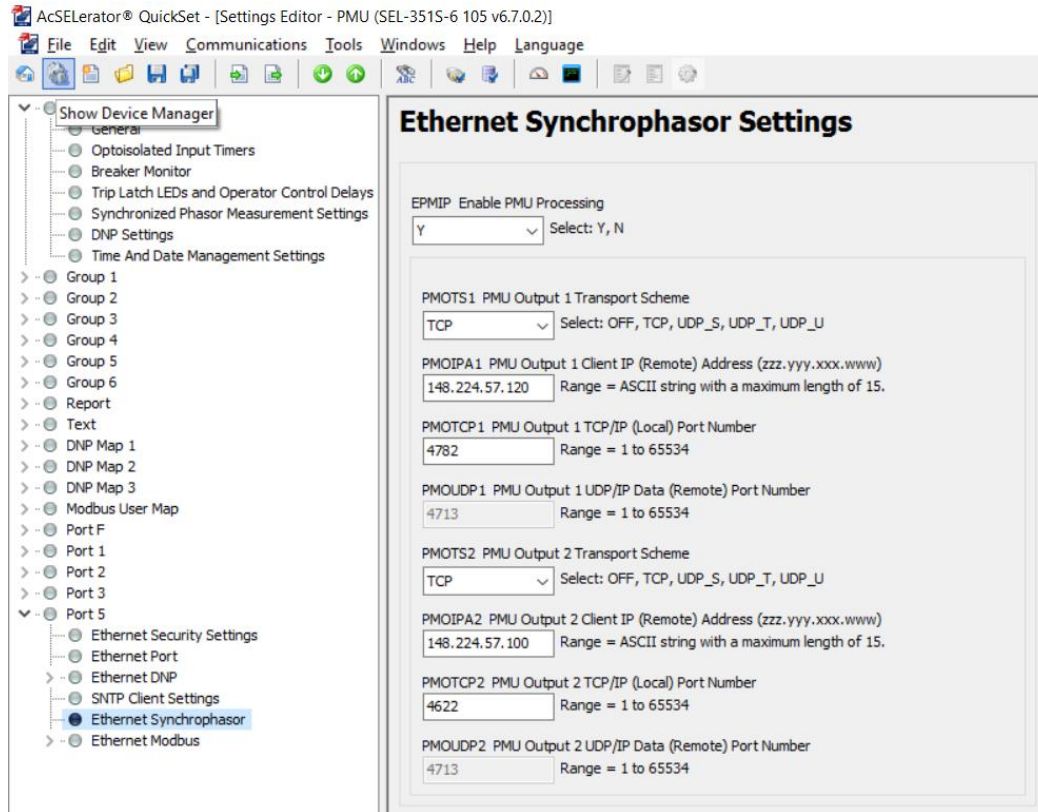


Figura 15. Configuración del puerto 5

2.1.2. Configuración De Equipo SEL-3530 Como Controlador En Tiempo Real.

Para la configuración del equipo SEL-3530 RTAC, utilizaremos el software AcSelerator RTAC®. El primer paso es integrar el equipo SEL-351S con el protocolo C37.118 como se muestra en la Figura 16, para que pueda recibir las mediciones sincrofásicas generadas por este.

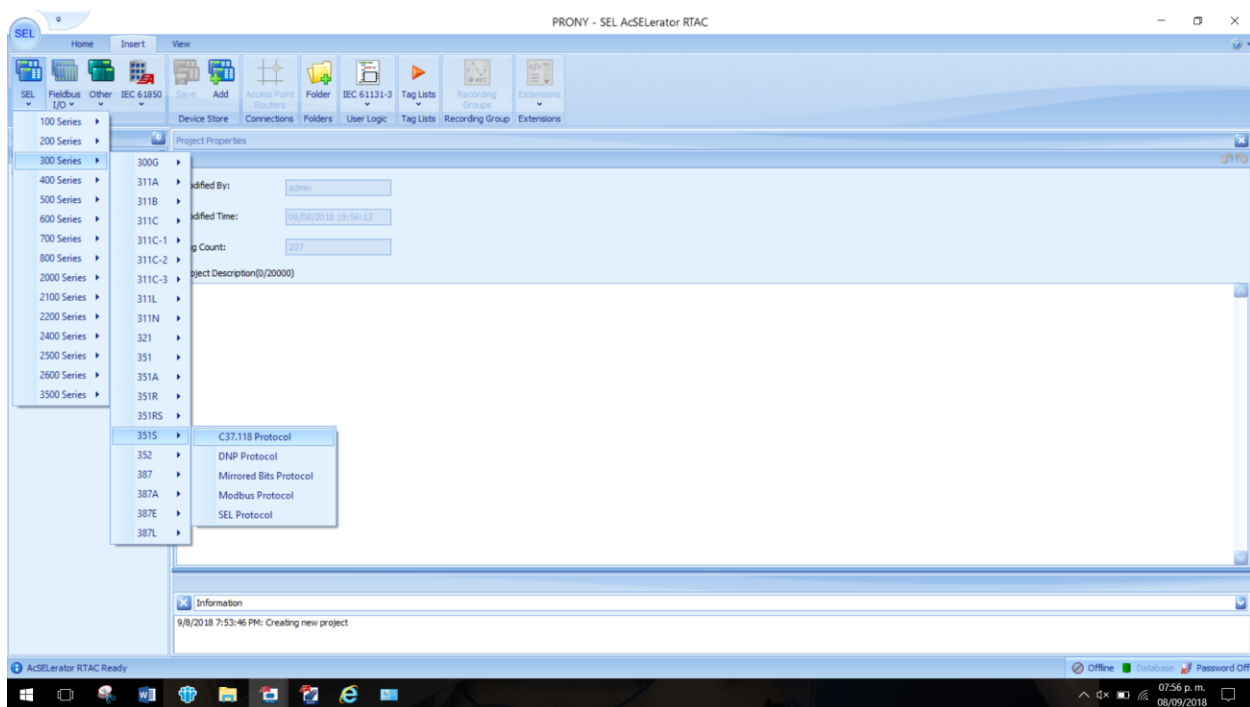


Figura 16. Integración del equipo SEL-351S al SEL-3530 mediante protocolo C37.118.

Los parámetros de la Tabla 2 son ingresados en los campos mostrados en las Figuras 17 y 18.

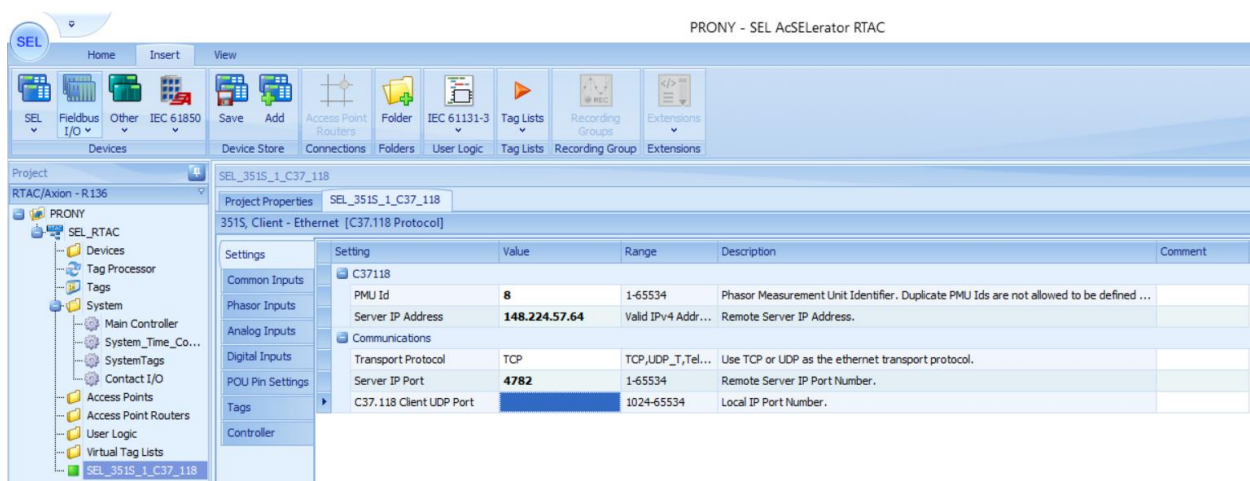


Figura 17. Configuración del equipo SEL-3530 para recibir mediciones sincrofásorales desde el equipo SEL-351S.

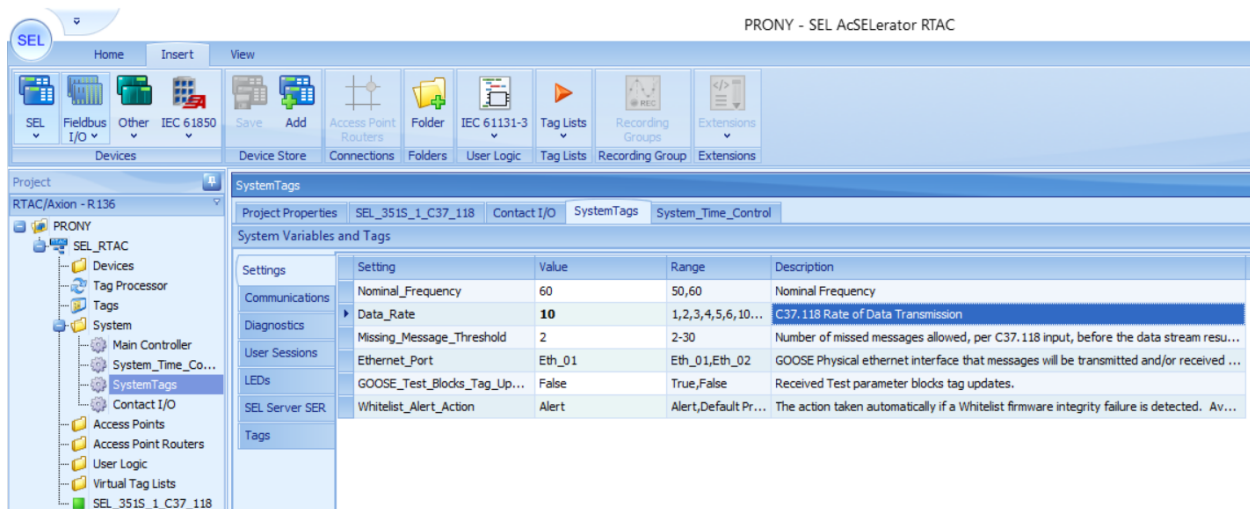


Figura 18. Configuración de la velocidad de recepción de las mediciones de la PMU.

2.2. Estándar IEC-61131-3.

El estándar IEC-61311-3 representa una combinación de diferentes estándares (IEC 50, IEC 559, IEC 617-12, IEC 617-13, IEC 848, ISO/AFNOR, ISO/IEC 646, ISO 8601, ISO 7185, ISO 7498) [19].

El estándar se compone de las siguientes partes.

1. Información General.
2. Requerimientos del equipo y pruebas.
3. Lenguajes de programación.
4. Guías de usuario.
5. Comunicaciones.
6. Seguridad Funcional.
7. Programación de lógica difusa.
8. Guías para la aplicación e implementación de lenguajes de programación.

Nos enfocaremos en la parte 3 del estándar, la cual describe los diferentes lenguajes de programación que posee.

Estos lenguajes de programación pueden ser textuales o gráficos, como se muestra a continuación [19]:

Lenguajes textuales:

1. Listado de instrucciones IL.
2. Texto estructurado ST.

Lenguajes gráficos:

1. Diagrama de escalera LD.
2. Bloques de funciones FBD.

En este trabajo de tesis se utilizó el texto estructurado ST como lenguaje para la programación del algoritmo de Prony.

En texto estructurado como en diversos lenguajes de programación, el flujo del programa es controlado por una combinación de palabra clave y expresiones.

Las palabras clave son palabras reservadas que tienen un propósito definido en el estándar, ejemplo de ellas son las utilizadas para declaración de variables (VAR_INPUT, VAR_OUTPUT), las utilizadas para designar el tipo de datos (BOOL, REAL, BYTE, etc), las utilizadas para controlar el flujo del programa (FOR, WHILE, IF, etc), entre otras.

Las expresiones consisten en operandos y operadores. Los operandos pueden ser variables, llamadas de funciones, etc. mientras que los operadores son aquellos que realizan algún cambio o acción sobre los operandos, ejemplo de ellos son operadores lógicos (AND, OR, NOT, etc), aritméticos (+, -, *, /) y de comparación (=, >, <, <>).

La Figura 19 muestra un fragmento de código escrito en el estándar IEC-61131-3 utilizando texto estructurado, en donde se pueden apreciar las expresiones y palabras clave.

```
IF (SEL_351S_1_C37_118_Controller.SEL_351S_1_C37_118_POU.Message_Received_Count) > MSG_READ THEN
  FOR i:= 0 TO N-2 DO
    X[i]:= X[i+1];
  END_FOR
  X[N-1].Re:= SEL_351S_1_C37_118.VAFM.cVal.mag/1000;
  MSG_READ:=SEL_351S_1_C37_118_Controller.SEL_351S_1_C37_118_POU.Message_Received_Count;
```

Figura 19. Declaraciones en lenguaje de texto estructurado.

El estándar IEC-61131-3 llama Unidades de Organización de Programa (POU, program organization units, por sus siglas en inglés) a los programas, funciones y bloques de funciones.

El programa es el bloque lógico más básico sin el cual no es posible insertar funciones o bloques de funciones.

Las funciones son llamadas desde el programa principal o desde los bloques de funciones y se utilizan para resolver problemas repetitivos. Una de las características de las funciones es que estas cuando son llamadas con los mismos valores de entrada producen siempre el mismo valor de salida ya que no tienen elementos de memoria que afecten el resultado.

Un bloque de función es una rutina para la cual se pueden tener muchas instancias de esta, lo cual puede entenderse como “copias” del mismo código que poseen los mismos atributos que el bloque original.

Los bloques de funciones se componen de variables de entrada, variables de salida, variables internas y lógica interna. Estos bloques son llamados dentro del programa utilizando instancias de estos como se menciona en el párrafo anterior.

Una de las principales características de un bloque de función es que posee memoria, lo que permite a las instancias de este retener valores. Esta característica es una de las principales diferencias con las funciones.

El estándar IEC-61131-3 permite al usuario programar funciones y bloques de funciones de acuerdo con sus necesidades, permitiéndole utilizarlas posteriormente dentro del programa principal.

En este trabajo se programarán bloques de funciones para la ejecución de tareas específicas dentro del algoritmo de Prony y que no estén contenidas en las bibliotecas propietarias del controlador SEL-3530 RTAC.

2.3. Bibliotecas Propietarias SEL-RTAC.

Las bibliotecas son un conjunto de funciones y bloques de funciones creadas para ejecutar tareas específicas simplificando la escritura del programa.

El software AcSelerator RTAC contiene una serie de bibliotecas específicamente diseñadas para aplicaciones en los sistemas eléctricos de potencia, entre las cuales se encuentran AnalogCond para procesamiento de señales analógicas y filtrado, PowerMetering para el cálculo de energía y demanda, PowerSystemProtection para identificar condiciones de falla, entre otras [16].

Además de las aplicaciones mencionadas, el software AcSelerator RTAC tiene aplicaciones para realizar operaciones con matrices (MathMatrix) y números complejos (MathComplex), las cuales nos ayudaran para simplificar la escritura del código de Prony [16].

La Figura 20 muestra la manera de insertar una biblioteca en el ambiente del software AcSelerator RTAC para utilizarla dentro del código del programa.

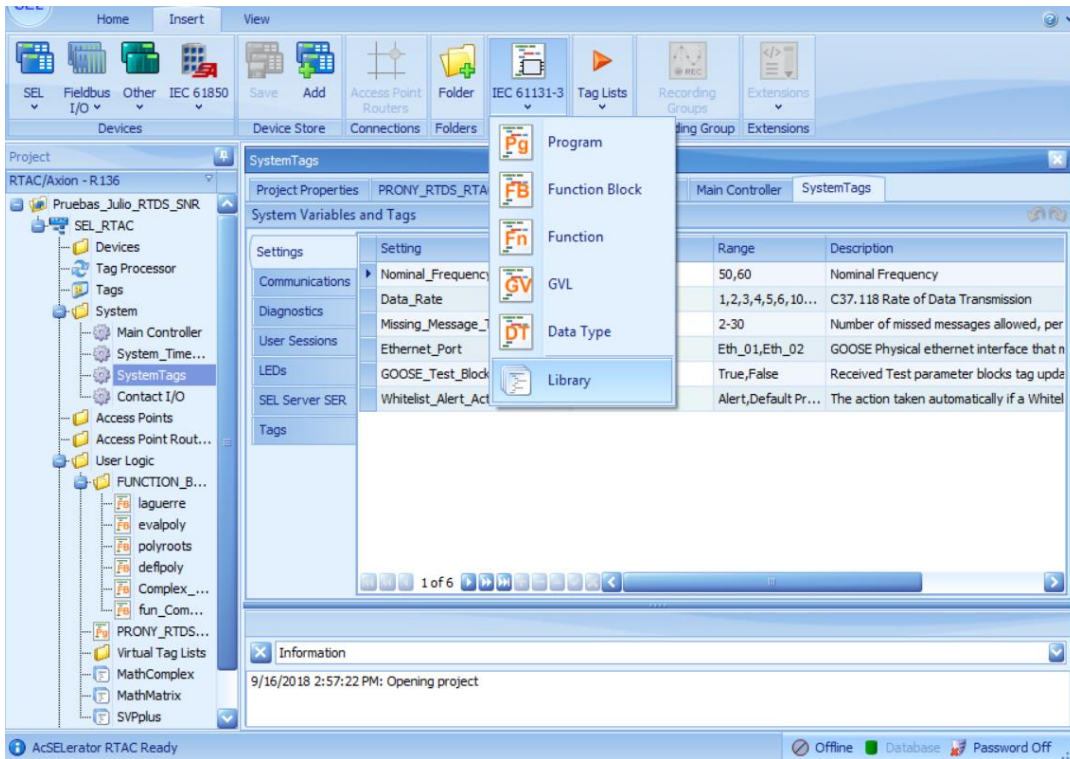


Figura 20. Bibliotecas dentro de AcSelerator RTAC.

En la escritura del código de Prony utilizaremos dos bibliotecas propietarias, la primera de ellas es la biblioteca MathMatrix, la cual se empleará para solucionar el sistema que determina los coeficientes del polinomio característico.

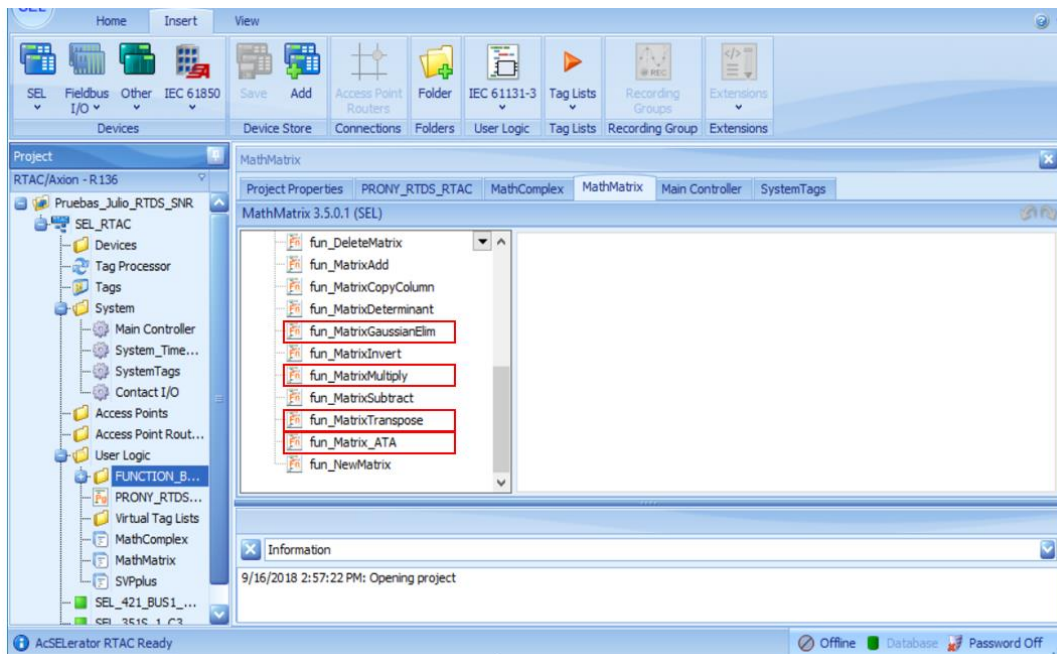


Figura 21. Funciones de biblioteca MathMatrix.

La segunda biblioteca que utilizaremos en el desarrollo del código es la biblioteca MathComplex, la cual como se aprecia en las funciones de la Figura 22, nos permite manipular números complejos.

Esta biblioteca será de particular ayuda al desarrollar el bloque de función para el cálculo de las raíces de un polinomio, el cual es parte esencial para el cálculo de la frecuencia de oscilación y el coeficiente de amortiguamiento.

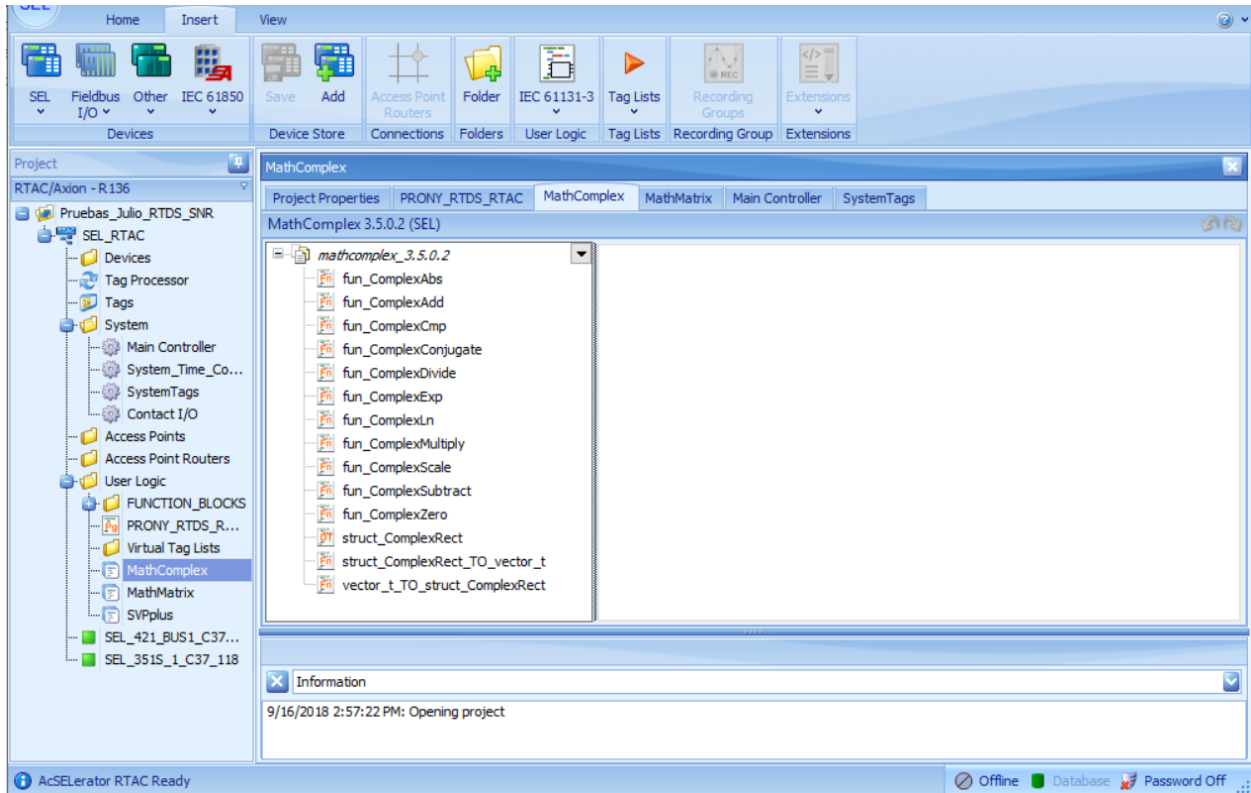


Figura 22. Funciones de biblioteca MathComplex

2.4. Bloques De Funciones Programadas Por El Usuario.

Si bien las bibliotecas revisadas en la sección anterior nos facilitan la programación del código, en ocasiones para resolver un problema específico es necesario programar funciones adicionales.

Con la finalidad de simplificar el código utilizaremos bloques de funciones para cada uno de los siguientes casos:

1. Cálculo de una raíz de un polinomio.
2. Evaluación de un polinomio.
3. Degradación de un polinomio.
4. Evaluación de un número complejo elevado a la n .

2.4.1. Bloque De Función Laguerre Para Cálculo de Raíces De Polinomio.

Para el cálculo de las raíces del polinomio característico utilizaremos el algoritmo de Laguerre presentado en el capítulo anterior:

FUNCTION_BLOCK Laguerre

VAR

```
i: INT;  
my_n_real: LREAL;  
my_Complex_Sqrt: Complex_Sqrt;  
my_evalpoly: evalpoly;  
g: struct_ComplexRect;  
h: struct_ComplexRect;  
f: struct_ComplexRect;  
dx: struct_ComplexRect;  
my_pol: struct_ComplexRect;  
my_dpol: struct_ComplexRect;  
my_ddpol: struct_ComplexRect;
```

END_VAR

VAR_INPUT

```
a: ARRAY[1..10] OF struct_ComplexRect; // COEFICIENTES DEL POLINOMIO  
n: INT; // GRADO DEL POLINOMIO
```

END_VAR

VAR_OUTPUT

```
x: struct_ComplexRect; // RAIZ DEL POLINOMIO
```

END_VAR

VAR

```
// VARIABLES NECESARIAS EN EL CODIGO
```

END_VAR

```
// CODIGO DEL BLOQUE DE FUNCIÓN
```

```
x_la.RE:= 1;  
x_la.Im:= 0.1;
```

```
my_n_real := INT_TO_LREAL(n_la);
```

FOR i:= 1 **TO** 100 **BY** 1 **DO**

```
my_evalpoly(a:=a_la, x_eval:= x_la, n:=n_la);  
my_pol:=my_evalpoly.pol;  
my_dpol := my_evalpoly.dpol;  
my_ddpol:= my_evalpoly.ddpol;  
IF (fun_ComplexAbs(my_pol))< tol THEN  
    RETURN;  
ELSE  
    g:= fun_ComplexDivide(my_dpol,my_pol);
```



```

        h:= fun_ComplexSubtract(fun_ComplexMultiply(g,g),fun_ComplexDivide(my_ddpol,my_pol));
        f:=
fun_ComplexScale(fun_ComplexSubtract(fun_ComplexScale(h,my_n_real),fun_ComplexMultiply(g,g)),(m
y_n_real-1));
        my_Complex_Sqrt(num:=f);
        f:= my_Complex_Sqrt.raiz;
            IF fun_ComplexAbs(fun_ComplexAdd(g,f)) >=
fun_ComplexAbs(fun_ComplexSubtract(g,f)) THEN

                dx:=fun_ComplexScale((fun_ComplexDivide(Unos,fun_ComplexAdd(g,f))),my_n_real);
                ELSE

                dx:=fun_ComplexScale((fun_ComplexDivide(Unos,fun_ComplexSubtract(g,f))),my_n_real);

                END_IF
            x_la:= fun_ComplexSubtract(x_la,dx);
            IF ABS(x_la.lm) < 0.00001 THEN
                x_la.lm:= 0;
            END_IF
            IF fun_ComplexAbs(dx)<tol THEN
                RETURN;
            END_IF
        END_IF
    END_FOR

```

2.4.2. Bloque De Función Evaluación De Polinomio.

El bloque de función evalpoly es utilizado por el bloque Laguerre para evaluar el polinomio característico en cada iteración usando la raíz estimada en ese momento y con base en la tolerancia establecida, determinar si ese valor corresponde a una raíz del polinomio o bien si son necesarias más iteraciones.

```

FUNCTION_BLOCK evalpoly

```

```

VAR

```

```

    j, k: INT;

```

```

END_VAR

```

```

VAR_INPUT

```

```

    a: ARRAY[1..15] OF struct_ComplexRect;

```

```

    // ARREGLO DE COEFICIENTES DEL POLINOMIO

```

```

    x_eval: struct_ComplexRect;

```

```

    // VALOR DE "x" A EVALUAR EN EL //POLINOMIO

```

```

        n: INT;
END_VAR

VAR_OUTPUT
    pol: struct_ComplexRect;
// VALOR QUE RESULTA DE EVALUAR EL //POLINOMIO EN x = x_eval
    dpol: struct_ComplexRect;
// VALOR QUE RESULTA DE EVALUAR LA PRIMERA //DERIVADA DEL POLINOMIO EN x = x_eval
    ddpol: struct_ComplexRect;
// VALOR QUE RESULTA DE EVALUAR LA //SEGUNDA DERIVADA DEL POLINOMIO EN x = x_eval
END_VAR

// CODIGO DEL BLOQUE DE FUNCIÓN
pol := a[1];
dpol.Re := 0.0;
dpol.Im := 0.0;
ddpol.Re := 0.0;
ddpol.Im := 0.0;
FOR j:= 1 TO n BY 1 DO
    ddpol := fun_Complexadd(fun_ComplexMultiply(ddpol,x_eval),fun_ComplexScale(dpol,2));
    dpol := fun_Complexadd(fun_ComplexMultiply(dpol,x_eval),pol);
    pol := fun_Complexadd(fun_ComplexMultiply(pol,x_eval), a[j+1]);
END_FOR

```

2.4.3. Bloque De Función Degradación De Polinomio.

Debido a que el algoritmo de Laguerre obtiene una raíz del polinomio a la vez, es necesario degradar el polinomio original con la raíz calculada y de esta manera obtener un nuevo polinomio (un grado menor que el polinomio original) el cual será la nueva entrada para el bloque de función de Laguerre para el cálculo de la siguiente raíz.

```

FUNCTION_BLOCK deflpoly
VAR
    i:INT; // VARIABLE AUXILIAR PARA CICLOS
END_VAR

VAR_INPUT
    a: ARRAY[1..15] OF struct_ComplexRect;
// COEFICIENTES DEL POLINOMIO ORIGINAL
    root: struct_ComplexRect;

```

```

// RAIZ DEL POLINOMIO ORIGINAL
n: INT; // GRADO DEL POLINOMIO ORIGINAL
END_VAR

VAR_OUTPUT
b: ARRAY[1..15] OF struct_ComplexRect; // POLINOMIO DEGRADADO
END_VAR

// CODIGO DE LA FUNCIÓN

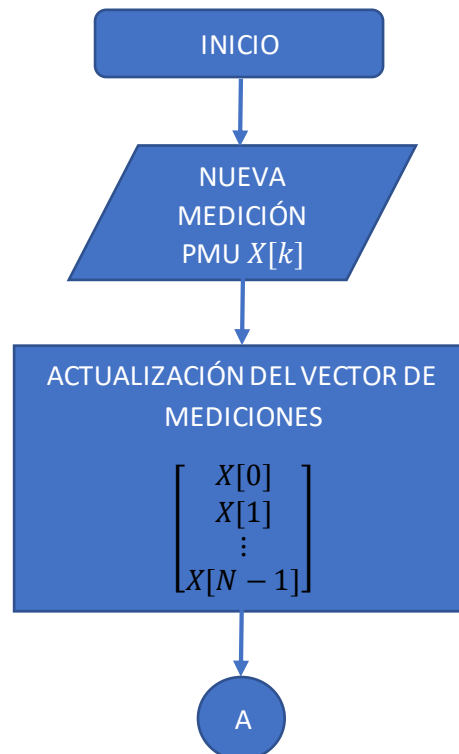
FOR i:= 1 TO 15 BY 1 DO
    fun_ComplexZero(b[i]); // INICIALIZA LOS COEFICIENTES EN "0"
END_FOR

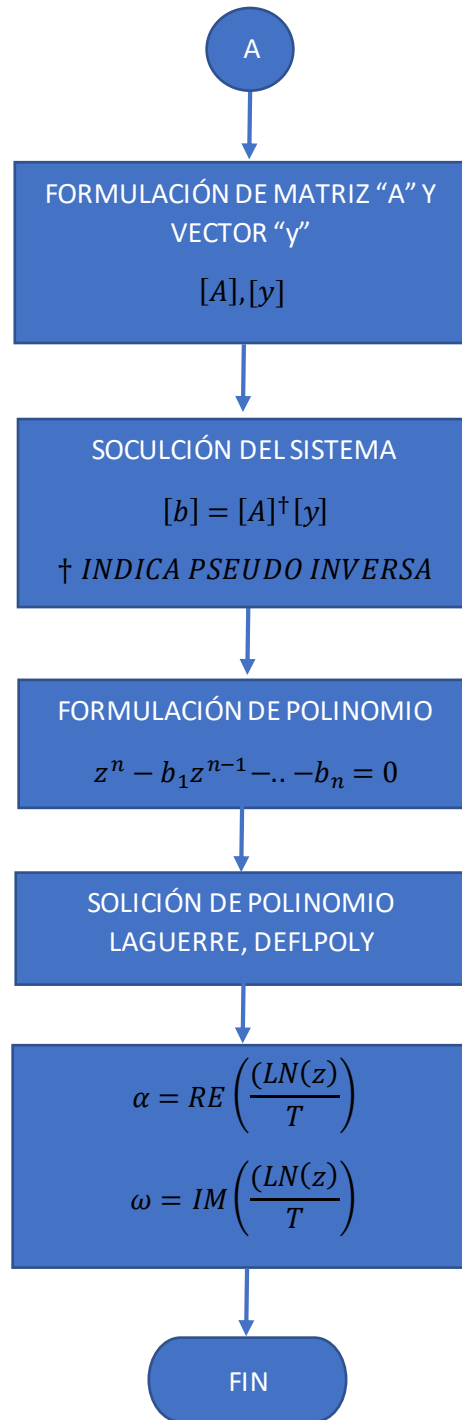
b[1]:= a[1];
// IGUALA EL PRIMER COEFICIENTE DEL NUEVO POLINOMIO CON EL DEL ORIGINAL
FOR i:= 2 TO n BY 1 DO
    b[i]:= fun_ComplexAdd(a[i], fun_ComplexMultiply(root,b[i-1]));
END_FOR

```

2.5. Diagrama De Flujo Del Algoritmo Básico De Prony.

A continuación, se muestra de manera simplificada el diagrama de flujo del algoritmo de Prony Básico.





Existen muchas maneras de mejorar el algoritmo de Prony básico, muchas de ellas se basan en optimizar la forma de obtener la solución al sistema sobredeterminado, otras de ellas buscan la relación óptima de frecuencia de muestreo y longitud del vector de muestras.

Este trabajo de tesis se determinará la frecuencia de muestreo necesaria para la ejecución del algoritmo dentro de las opciones disponibles en los equipos utilizados con base en un tamaño de vector de muestras previamente seleccionado.

Además, se realizará un análisis de los resultados obtenidos utilizando discriminación por energías para determinar la validez de los mismos y se evaluarán los índices presentados en el Capítulo I para conocer qué tanto se aproxima la estimación a las mediciones.

3. Capitulo III. Resultados Experimentales.

En este capítulo se presentan los resultados experimentales de pruebas realizadas al algoritmo de Prony programado utilizando el simulador RTDS.

Las pruebas son divididas en tres tipos, la primera es realizada para validar el algoritmo operando fuera de línea utilizando muestras de una función conocida, las cuales son ingresadas manualmente al código del programa.

La segunda prueba es realizada utilizando un modelo de fuente de tensión variable sobre la cual se pueden montar modos de oscilación de frecuencias conocidas y de esta manera verificar si el algoritmo estima de manera adecuada los parámetros de las mismas.

Una vez validado el algoritmo con la primera y segunda prueba, como tercera prueba se utilizará un modelo de dos generadores, dos transformadores y dos líneas de transmisión para estimar las oscilaciones presentes ante casos de variación de carga y ante fallas.

De manera adicional se realizará una prueba utilizando la biblioteca SVP Plus del controlador RTAC, en donde se utilizará el sistema de dos generadores, dos transformadores y dos líneas de transmisión para probar su desempeño.

3.1. Prueba Del Algoritmo.

Como primera prueba del algoritmo utilizaremos la función (3.1) para generar los datos de entrada, mismos que serán ingresados de manera manual al código de programa para verificar su desempeño.

$$Y = e^{-0.1t} \cos(3\pi t) + 0.25e^{-0.25t} \cos(14\pi t + \frac{\pi}{8}) [1] \quad (3.1)$$

La Figura 23 muestra el vector $X[n]$, el cual corresponde a la función (3.1) muestreada a una frecuencia de 50 Hz para fines de ejemplificar la funcionalidad del algoritmo.

```

PRONY_RTDS_RTAC
Project Properties  fun_ComplexPot  PRONY_RTDS_RTAC
Program
6  VAR
7      N: UINT:= 50;           // NUMERO DE MUESTRAS DEL SISTEMA
8      M: UINT:= 10;          // ORDEN DE LA ESTIMACION (NUMERO DE INGEVALORES)
9      X: ARRAY[0..49] OF struct_ComplexRect; // VENTANA DE MUESTRAS
10     X_Est: ARRAY[0..49] OF REAL; // VECTOR DE VALORES ESTIMADOS CON EL ALGORITMO
11     X_Est2: ARRAY[1..50] OF REAL; // VECTOR DE VALORES ESTIMADOS CON EL ALGORITMO
12     f_muestreo :REAL:= 50; // FRECUENCIA DE MUESTREO DE PMU's
13  END_VAR

10
11  // Y = exp(-0.1*t)*cos(3*pi*t)+0.25*exp(-0.25*t)*cos(14*pi*t+pi/8)
12  X[0].Re := 1.230969795 ;
13  X[1].Re := 1.053467298 ;
14  X[2].Re := 0.790173457 ;
15  X[3].Re := 0.594485528 ;
16  X[4].Re := 0.547183895 ;
17  X[5].Re := 0.601068133 ;
18  X[6].Re := 0.619192394 ;
19  X[7].Re := 0.478074031 ;
20  X[8].Re := 0.160631261 ;
21  X[9].Re := -0.228250639;
22  X[10].Re := -0.53414007 ;
23  X[11].Re := -0.660494052;
24  X[12].Re := -0.633396686;
25  X[13].Re := -0.577466229;
26  X[14].Re := -0.621341303;

```

Figura 23. Muestreo fuera de línea de función (3.1)

La Figura 24 muestra los resultados obtenidos con el algoritmo de Prony. Los recuadros en rojo contienen los valores estimados de los coeficientes de amortiguamiento, las frecuencias de oscilación, las magnitudes de los modos y los ángulos de fase.

Expression	Type	Value	Prepared value	Address
N	UINT	50		
M	UINT	4		
***** CALCULO DE SNR *****				
***** RESULTADOS *****				
ENERGIJA [1]	ALPHA [1]	Freq [1]	MAG [1]	ANG [1]
11.4	-0.1	1.5	0.5	2.41E-06
ENERGIJA [2]	ALPHA [2]	Freq [2]	MAG [2]	ANG [2]
11.4	-0.1	-1.5	0.5	-2.41E-06
ENERGIJA [3]	ALPHA [3]	Freq [3]	MAG [3]	ANG [3]
0.618	-0.25	7	0.125	0.393
ENERGIJA [4]	ALPHA [4]	Freq [4]	MAG [4]	ANG [4]
0.618	-0.25	-7	0.125	-0.393

Figura 24. Estimación de ecuación (3.1) con el algoritmo de Prony.

Como se puede observar, los parámetros que definen la función (3.1) son estimados de manera correcta, lo cual valida la funcionalidad del algoritmo corriendo fuera de línea.

3.2. Sistema De Prueba RSCAD: Fuente De Tensión Con Modos De Oscilación Conocidos.

Con la finalidad de realizar pruebas HIL (hardware-in-the-loop, por sus siglas en inglés), utilizaremos el siguiente sistema de prueba, el cual consiste en una fuente de tensión trifásica cuya magnitud se conforma de una componente constante más dos componentes oscilatorias de frecuencia y magnitud variable.

La tensión es enviada utilizando los canales 7, 8 y 9 de la tarjeta GTA0 al equipo PMU para realizar el cálculo del fasor, posteriormente el fasor es enviado hacia el controlador SEL-3550 RTAC para la ejecución del algoritmo.

De manera adicional, el fasor es calculado internamente dentro del sistema RTDS para fines de visualizarlo en la IHM (Interfaz Humano Máquina) del software RSCAD.

Se utiliza un valor constante de 150 kV L-L para las tensiones de las fases A, B y C. Los controles Mag_Mode_1, Frec_Mode_1, Mag_Mode_2, Frec_Mode_2 son utilizados en la interfaz humano máquina para sumarle oscilaciones a la tensión constante.

La plataforma RTDS escalará las señales de tensión utilizando el módulo GTA0 antes de ser sacadas a través de este y ser ingresadas al relevador PMU.

La Tabla 3 muestra los factores utilizados tanto en RTDS como en la PMU, así como la tensión que será medida por el equipo una vez que los factores hayan sido aplicados.

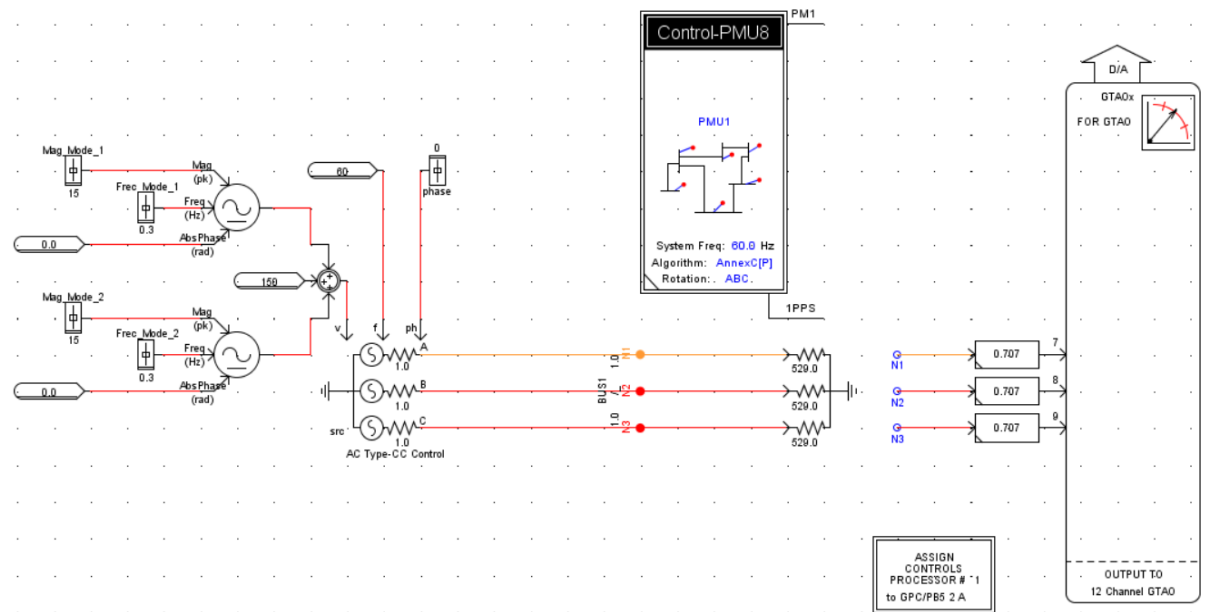


Figura 25. Sistema de prueba.

Tabla 3. Factores utilizados en RTDS y PMU.

Tensión (L-L)	Tensión (L-L)	Tensión (L-N)	Factor G _{TAO}	Factor PTR (PMU)	Tensión Medida PMU
Constante	150 kV	86.6 kV	0.001332	7500	865.200 kV
Modo 1 (1.31416 Hz)	4 kV	86.6 kV	0.001332	7500	23.074 kV
Modo 2 (1.62832 Hz)	4 kV	86.6 kV	0.001332	7500	23.074 kV

Como primera prueba, se adicionan a la magnitud constante de tensión oscilaciones a las frecuencias de 1.31416 Hz y 1.62832 Hz. La Figura 26 muestra los controles utilizados para variar la magnitud y frecuencia de los modos de oscilación inyectados.

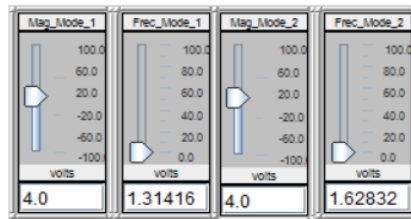
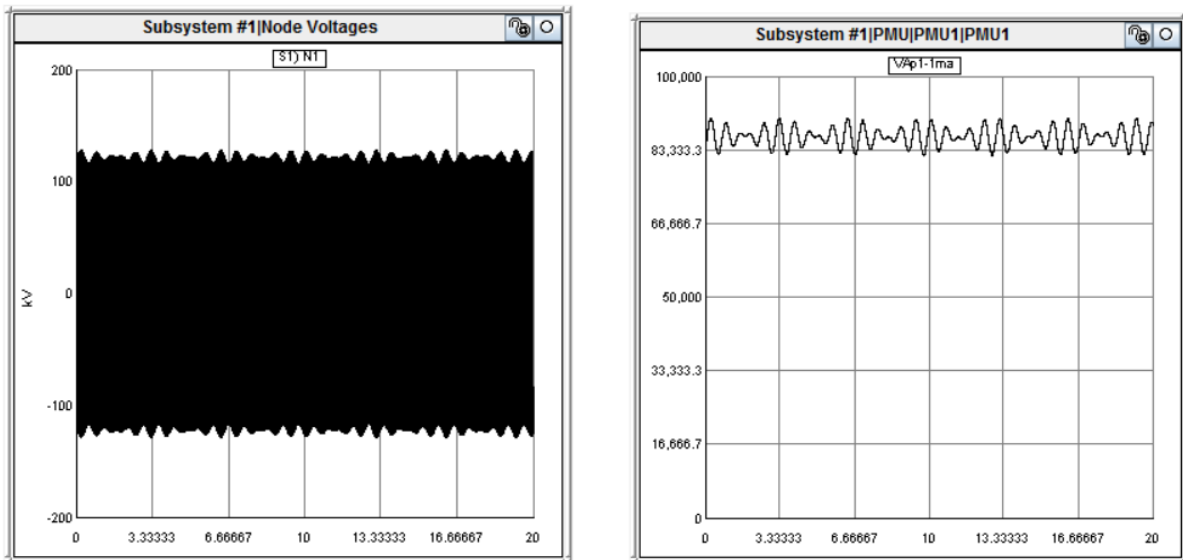


Figura 26. Magnitud y frecuencias de oscilación.

La Figura 27 muestra la forma de onda de la tensión y de su fasor. Como se puede observar, las oscilaciones en el dominio del tiempo son mapeadas al fasor, lo que permite que sea utilizado como dato de entrada para el algoritmo de Prony.



(a) Tensión fase a, (b) fasor de tensión fase a.

SEL_RTAC.Application.PRONY_RTDS_RTAC		
Expression	Type	Value
N	UINT	50
M	UINT	10
X	ARRAY [0..49] OF st...	
X_Est	ARRAY [0..49] OF R...	
X_Est2	ARRAY [1..50] OF R...	
f_muestreo	REAL	10


```

128 AUX_SNR_DEN [ 4.9E+03 ] := SNR_DEN [ 4.9E+03 ];
129 END_FOR
130 SNR [ 38.8 ] := 10 * LOG ( SNR_NUM [ 3.68E+07 ] / SNR_DEN [ 4.9E+03 ] );
131 (*****
132
133 (***** RESULTADOS *****)
134 ENERGIA [1] [ 3.75E+07 ] ; ALPHA [1] [ 0.000657 ] ; Freq [1] [ 0 ] ; MAG [1] [ 864 ] ; ANG [1] [ -1.03E-18 ] ;
135 ENERGIA [2] [ 3.65E+03 ] ; ALPHA [2] [ -0.152 ] ; Freq [2] [ 1.26 ] ; MAG [2] [ 11.8 ] ; ANG [2] [ -1.01 ] ;
136 ENERGIA [3] [ 3.65E+03 ] ; ALPHA [3] [ -0.152 ] ; Freq [3] [ -1.26 ] ; MAG [3] [ 11.8 ] ; ANG [3] [ 1.01 ] ;
137 ENERGIA [4] [ 2.95E+03 ] ; ALPHA [4] [ -0.169 ] ; Freq [4] [ 1.62 ] ; MAG [4] [ 11 ] ; ANG [4] [ -0.71 ] ;
138 ENERGIA [5] [ 2.95E+03 ] ; ALPHA [5] [ -0.169 ] ; Freq [5] [ -1.62 ] ; MAG [5] [ 11 ] ; ANG [5] [ 0.71 ] ;
139 ENERGIA [6] [ 0 ] ; ALPHA [6] [ 0 ] ; Freq [6] [ 0 ] ; MAG [6] [ 0 ] ; ANG [6] [ 0 ] ;
140 ENERGIA [7] [ 0 ] ; ALPHA [7] [ 0 ] ; Freq [7] [ 0 ] ; MAG [7] [ 0 ] ; ANG [7] [ 0 ] ;
141 ENERGIA [8] [ 543 ] ; ALPHA [8] [ -10.9 ] ; Freq [8] [ 5 ] ; MAG [8] [ 21.9 ] ; ANG [8] [ -7.13E-15 ] ;
142 ENERGIA [9] [ 0 ] ; ALPHA [9] [ 0 ] ; Freq [9] [ 0 ] ; MAG [9] [ 0 ] ; ANG [9] [ 0 ] ;
143 ENERGIA [10] [ 0 ] ; ALPHA [10] [ 0 ] ; Freq [10] [ 0 ] ; MAG [10] [ 0 ] ; ANG [10] [ 0 ] ;
144
145 END_IF
146 RETURN

```

Figura 28. Estimación de modos de oscilación para la tensión de la Figura 27.

Como se puede observar en la Figura 28, el algoritmo estima una componente a frecuencia fundamental de 864 kV contra 865.2 kV que son inyectados.

Además, estima un modo de oscilación de 1.26 Hz con una magnitud de 23.6 kV (11.8 + 11.8), el cual corresponde al modo inyectado de 1.31416 Hz y 23.074 kV (Ver tabla 1).

De manera similar, el algoritmo estima un modo de 1.62 Hz con una magnitud de 22 kV (11 + 11), el cual corresponde al modo inyectado de 1.62832 Hz y 23.074 kV.

Ahora se utilizará un modelo de un sistema de potencia ante variaciones de carga y bajo falla con la finalidad de estimar los parámetros de las oscilaciones presentes en ambos casos.

3.3. Sistema De Prueba RSCAD: Sistema De Dos Fuentes, Dos Generadores Y Dos Líneas De Transmisión.

La Figura 29 muestra el sistema de dos generadores, dos transformadores y dos líneas de transmisión que será utilizado con la finalidad de determinar las oscilaciones presentes ante condiciones de variaciones de carga y falla.

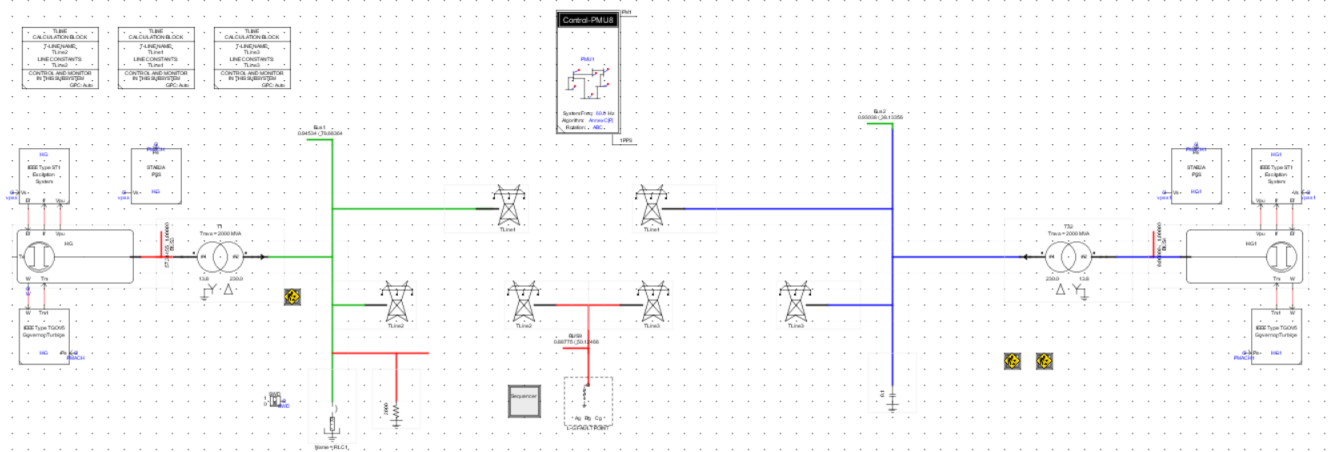


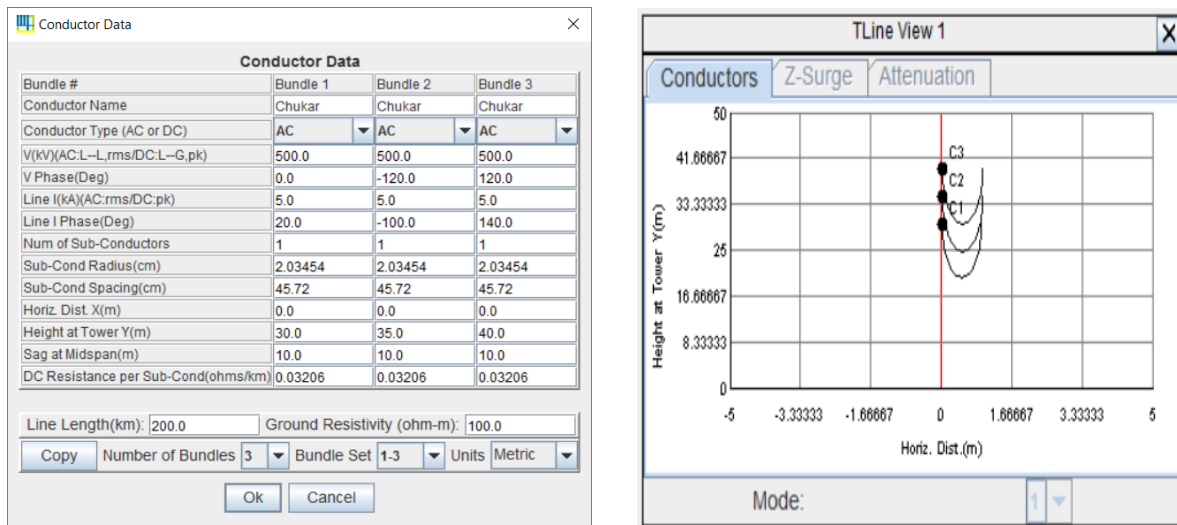
Figura 29. Sistema de prueba en RTDS.

Los parámetros del sistema de la Figura 29 así como el modelo de línea de transmisión utilizada son mostrados en la Tabla 4.

Tabla 4. Parámetros del modelo de la Figura 29.

GENERADORES	
Reactancia Transitoria	0.314 p.u.
Constante de inercia (H)	4 seg
EXITADOR IEEE TIPO ST1	
Constante de Tiempo T_r	0.0 seg
Límite de error máximo V_{imx}	10.0 p.u.
Límite de error mínimo V_{imn}	-10.0 p.u.
Constante de tiempo de adelanto T_c	1.0 seg
Constante de tiempo de atraso T_b	12.0 seg
Ganancia de regulador de voltaje K_a	200
Constante de tiempo de regulador de voltaje T_a	0.02 seg
Salida máxima del controlador V_{rmx}	5.64 p.u.
Salida mínima del controlador V_{rmn}	-4.53 p.u.
Factor de regulación K_c	0.0
Ganancia de retroalimentación K_f	0.03
Constante de tiempo de retroalimentación T_f	1.0 seg
TRANSFORMADORES	
Potencia Aparente	2000 MVA
Reactancia	0.152 p.u.
LÍNEAS DE TRANSMISIÓN	
Modelo	Bergeron
Parámetros	Ver Figura 30

Los parámetros, así como la disposición física de las líneas de transmisión utilizadas se muestran en la Figura 30.



(a) (b)
FIGURA 30. PARÁMETROS DE LAS LÍNEAS DE TRANSMISIÓN UTILIZADAS.

Utilizando el sistema de la Figura 29 realizaremos varias pruebas para provocar oscilaciones de baja frecuencia en el sistema y estimarlas utilizando el algoritmo de Prony.

La primera prueba consiste en una disminución de la carga conectada al bus 1 en 25 MW, lo que da como resultado las oscilaciones mostradas en la Figura 31.

Como se puede observar en la Figura 32, la magnitud de la componente de CD del fasor (1,280 kV) es la observada en la Figura 31 (b), considerando que los resultados mostrados en la Figura 32 están afectados por los factores de la Tabla 3 , además el algoritmo estima un modo de oscilación decreciente, el cual oscila a 0.247 Hz.

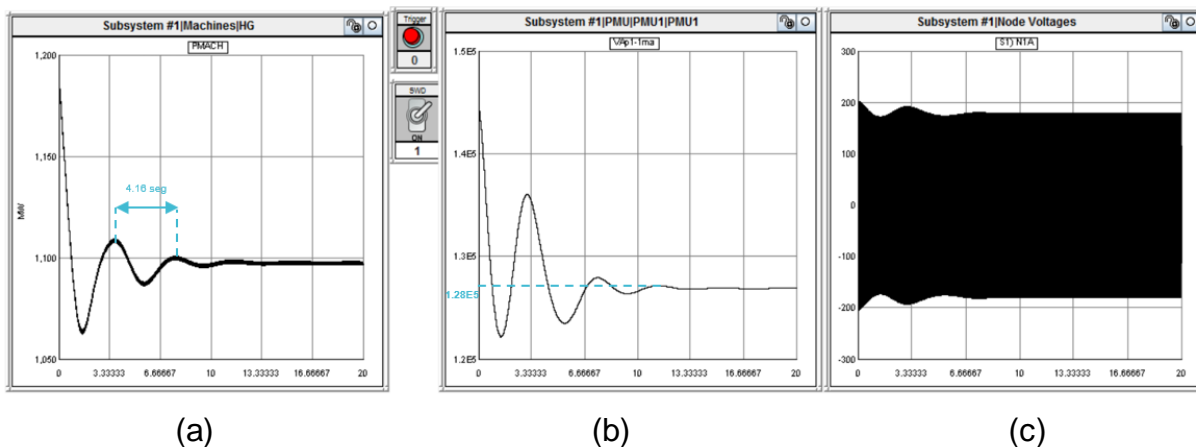


Figura 31. Oscilaciones por variación de carga (a) Potencia Mecánica (b) Fasor de tensión fase a (V) (c) Tensión fase a (kV).

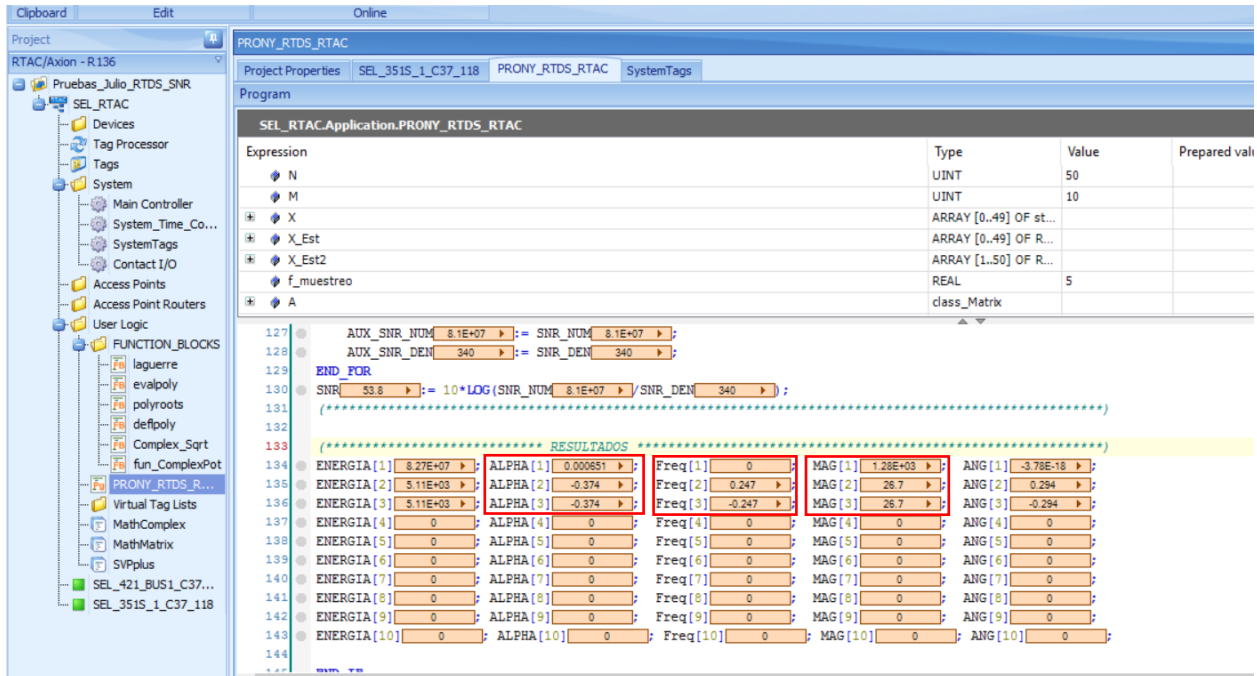
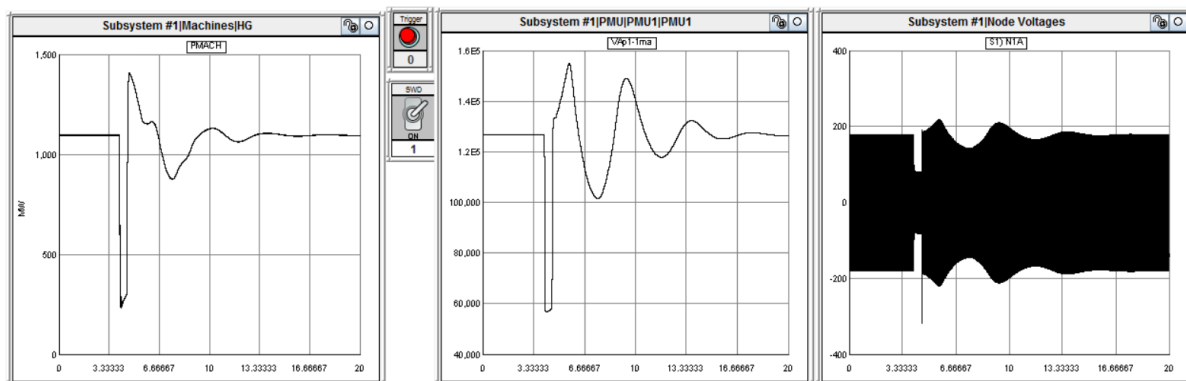


FIGURA 32. ESTIMACIÓN DE MODOS DE OSCILACIÓN POR VARIACIONES DE CARGA UTILIZANDO EL ALGORITMO DE PRONY.

La segunda prueba realizada es la de una falla temporal de fase a tierra con una resistencia de falla de 0.01Ω .

La Figura 33 muestra las oscilaciones en la potencia del generador G1 y del fasor de tensión de la fase a.

Como se puede observar en la Figura 34, la magnitud de la componente de CD del fasor (1,290 kV) se aproxima a la señal observada en la Figura 33 (b), además el algoritmo estima un modo de oscilación decreciente, el cual oscila a 0.26 Hz.



(a)

(b)

(c)

Figura 33. Oscilaciones debido a falla de fase a tierra. (a) Potencia (b) Fasor de tensión fase a (c) Tensión fase a.

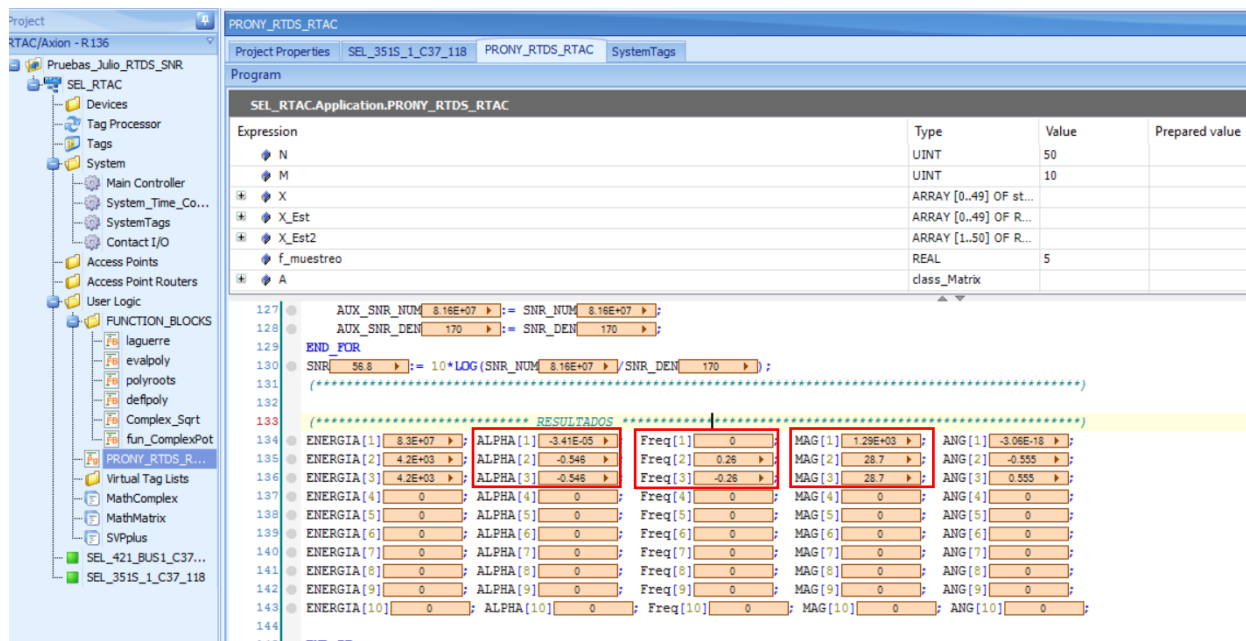


Figura 34. Estimación de modos de oscilación ante una falla de fase a tierra utilizando el algoritmo de Prony.

3.4. Biblioteca Proprietaria SEL-RTAC: SVP Plus.

La biblioteca SVP Plus (Synchrophasor Vector Processor Plus, por sus siglas en inglés) permite el análisis de Prony de señales de corriente y tensión, referido como análisis modal [20].

El bloque de función de análisis modal de esta biblioteca se encarga de dos tareas principales, la primera de ellas es recolectar y almacenar la información que le es enviada, y la segunda es analizar los datos almacenados y presentar los modos obtenidos como resultado [20].

La aproximación se evalúa utilizando el índice SNR revisado en el capítulo I.

Realizaremos una prueba de disminución de carga utilizando el esquema de la Figura 29 y compararemos los resultados obtenidos de la biblioteca SVP Plus con los entregados por el algoritmo de Prony programado en este trabajo de tesis.

Los parámetros utilizados tanto en la biblioteca SVP Plus como en el algoritmo de Prony programado se muestran en la tabla 5.

Tabla 5. Parámetros de entrada para el análisis modal.

Equipo	Dirección IP	Mascara de subred
Numero de Muestras	numSamples	50
Velocidad de Muestreo	sampleRate	5
Numero de Modos	g_c_NumModes	5

La Figura 35 muestra los resultados de la biblioteca SVP Plus para una variación de carga de 25 MW, mientras que la Figura 36 muestra los resultados obtenidos por el algoritmo del Prony programado.

Se puede observar en la Figura 35, que la biblioteca SVP Plus estima 4 modos de oscilación; sin embargo, el más significativo es aquel encerrado en el segundo recuadro rojo, ya que presenta la mayor magnitud, menor frecuencia y un bajo amortiguamiento.

Este modo de oscilación de la Figura 35, el cual oscila a 0.2659 Hz, se asemeja al estimado por el algoritmo de Prony programado mostrado en la Figura 36, el cual oscila a 0.248 Hz. Las magnitudes presentan variación, sin embargo, esto es normal ya que el algoritmo corre en línea y cada vez que este se ejecuta completamente, las magnitudes son actualizadas y si el coeficiente de amortiguamiento es negativo, las magnitudes estimadas decrecerán cada vez, por lo cual no es posible establecer una comparativa con base en las magnitudes para modos de oscilación de frecuencia diferente a 0 Hz.

Expression	Type	Value	Prepared value
g_SampleTimeValid	BOOL	TRUE	
g_SampleValid	BOOL	TRUE	
g_SNR	REAL	46.1978378	
g_ModeResults	ARRAY [1..g_c_Num...		
g_ModeResults[1]	struct_Mode		
Amplitude	REAL	1181839.38	
Frequency	REAL	0	
Phase	REAL	-2.385151E-15	
Damping	REAL	0.009538428	
DampingRatio	REAL	-1	
g_ModeResults[2]	struct_Mode		
Amplitude	REAL	22125.9121	
Frequency	REAL	0.265924871	
Phase	REAL	-1.11099648	
Damping	REAL	-0.859093666	
DampingRatio	REAL	0.457262427	
g_ModeResults[3]	struct_Mode		
Amplitude	REAL	7712.13574	
Frequency	REAL	2.087898	
Phase	REAL	-0.420806617	
Damping	REAL	-2.614221	
DampingRatio	REAL	0.195432574	
g_ModeResults[4]	struct_Mode		
Amplitude	REAL	2898.06714	
Frequency	REAL	1.77294993	
Phase	REAL	-2.27494717	
Damping	REAL	-0.9071313	
DampingRatio	REAL	0.0811631	
g_ModeResults[5]	struct_Mode		
Amplitude	REAL	2432.33276	
Frequency	REAL	0.851280451	
Phase	REAL	-0.9351794	
Damping	REAL	-0.459875464	
DampingRatio	REAL	0.08566204	
g_DampingAlarm	ARRAY [1..g_c_Num...		

Figura 35. Estimación de modos de oscilación ante variaciones de carga.

```

126 SNR_DEN := AUX_SNR_DEN + (X[1].Re - X_Est[1]) * (X[1].Re - X_
127 AUX_SNR_NUM 8.2E+07 := SNR_NUM 8.2E+07 ;
128 AUX_SNR_DEN 248 := SNR_DEN 248 ;
129 END_FOR
130 SNR[55.2] := 10*LOG(SNR_NUM 8.2E+07 / SNR_DEN 248) ;
131 (*****
132 *****
133 ***** RESULTADOS *****
134 ENERGIA [1] 8.31E+07 ; ALPHA [1] -0.000126 ; Freq[1] 0 ; MAG [1] 1.29E+03 ; ANG [1] 8.22E-18 ;
135 ENERGIA [2] 7.12E+03 ; ALPHA [2] -0.513 ; Freq[2] 0.248 ; MAG [2] 36.4 ; ANG [2] -1.19 ;
136 ENERGIA [3] 7.12E+03 ; ALPHA [3] -0.513 ; Freq[3] -0.248 ; MAG [3] 36.4 ; ANG [3] 1.19 ;
137 ENERGIA [4] 0 ; ALPHA [4] 0 ; Freq[4] 0 ; MAG [4] 0 ; ANG [4] 0 ;
138 ENERGIA [5] 0 ; ALPHA [5] 0 ; Freq[5] 0 ; MAG [5] 0 ; ANG [5] 0 ;
139 ENERGIA [6] 0 ; ALPHA [6] 0 ; Freq[6] 0 ; MAG [6] 0 ; ANG [6] 0 ;
140 ENERGIA [7] 0 ; ALPHA [7] 0 ; Freq[7] 0 ; MAG [7] 0 ; ANG [7] 0 ;
141 ENERGIA [8] 0 ; ALPHA [8] 0 ; Freq[8] 0 ; MAG [8] 0 ; ANG [8] 0 ;
142 ENERGIA [9] 0 ; ALPHA [9] 0 ; Freq[9] 0 ; MAG [9] 0 ; ANG [9] 0 ;
143 ENERGIA [10] 0 ; ALPHA [10] 0 ; Freq[10] 0 ; MAG [10] 0 ; ANG [10] 0 ;
144
145 END_IF
146 RETURN

```

Figura 36. Estimación de modos de oscilación ante variaciones de carga utilizando el algoritmo de Prony.

Las Figuras 37 y 38 muestran el caso para una falla de fase a tierra, el segundo recuadro en rojo de la Figura 37 muestra un modo de oscilación a 0.25 Hz, el cual corresponde con el modo detectado por el algoritmo de Prony programado de 0.247 Hz.

Expression	Type	Value	Prepared value	Address
g_SampleTimeValid	BOOL	TRUE		
g_SampleValid	BOOL	TRUE		
g_SNR	REAL	42.0929947		
g_ModeResults	ARRAY [1..g_c_Num...			
g_ModeResults[1]	struct_Mode			
Amplitude	REAL	1421636.25		
Frequency	REAL	0		
Phase	REAL	-2.771246E-15		
Damping	REAL	-0.01735138		
DampingRatio	REAL	0.99999994		
g_ModeResults[2]	struct_Mode			
Amplitude	REAL	44810.27		
Frequency	REAL	0.2504012		
Phase	REAL	-1.27139187		
Damping	REAL	-0.676400959		
DampingRatio	REAL	0.394965947		
g_ModeResults[3]	struct_Mode			
Amplitude	REAL	16108.1406		
Frequency	REAL	2.185401		
Phase	REAL	-0.6694605		
Damping	REAL	-2.76667356		
DampingRatio	REAL	0.197517529		
g_ModeResults[4]	struct_Mode			
Amplitude	REAL	9982.081		
Frequency	REAL	0.8677139		
Phase	REAL	-1.36399508		
Damping	REAL	-0.6288174		
DampingRatio	REAL	0.114577286		
g_ModeResults[5]	struct_Mode			
Amplitude	REAL	5048.04346		
Frequency	REAL	1.77346349		
Phase	REAL	-2.01198316		
Damping	REAL	-0.7005291		
DampingRatio	REAL	0.06274332		
g_DampingAlarm	ARRAY [1..g_c_Num...			

Figura 37. Estimación de modos de oscilación ante variaciones de carga utilizando la biblioteca SVP Plus.

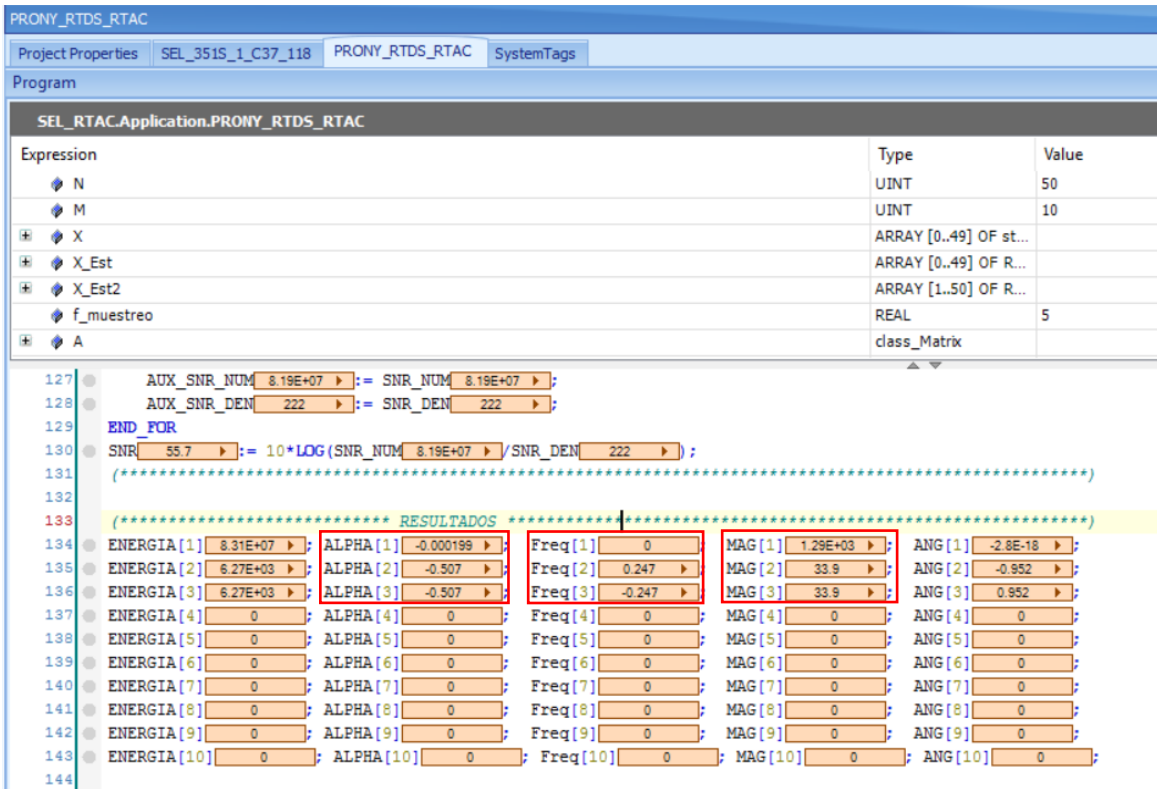


Figura 38. Estimación de modos de oscilación ante una falla de fase a tierra utilizando el algoritmo de Prony.

El utilizar señales con modos de oscilación cuyos parámetros son conocidos permite validar el funcionamiento del algoritmo. Esto se puede verificar por observación en las primeras dos pruebas del mismo, lo que permitió posteriormente utilizar el mismo código para identificar oscilaciones desconocidas en sistemas más complejos y ante situaciones que pueden presentarse en la operación de un sistema eléctrico.

Como se puede observar en las Figuras 36 a la 38, la biblioteca SVP Plus y el algoritmo de Prony programado en este trabajo de tesis detectan modos de oscilación cuyas frecuencias y coeficientes de amortiguamiento tienen valores muy cercanos, lo cual nos indica que detectan los mismos modos de oscilación en las señales obtenidas de la PMU. Cabe señalar que la magnitud de los modos de oscilación de frecuencias diferentes a 0 Hz no es un parámetro con el que podamos hacer una comparación entre los algoritmos ya que varían constantemente debido al factor de amortiguamiento, por lo que para lograr una comparación válida, ambos algoritmos tendrían que correr en paralelo y mostrar sus resultados en el mismo instante de tiempo.

4. Conclusiones Y Trabajo Futuro.

Esta tesis presenta una contribución en la implementación del algoritmo de Prony por ventanas deslizantes y discriminación por energía utilizando la plataforma SEL-3530 RTAC, la cual se encuentra instalada como equipo de comunicaciones y control en muchas subestaciones del país y en el extranjero.

Para llevar a cabo lo anterior, las bibliotecas del equipo SEL-3530 RTAC fueron complementadas con bloques de funciones programadas por el usuario para que en conjunto fueran utilizadas en la programación del algoritmo de Prony.

Con los resultados obtenidos se puede concluir lo siguiente:

1. El método de Prony debe ser ajustado en sus parámetros de número de muestras y frecuencia de muestreo de manera adecuada, obedeciendo a las restricciones impuestas por el modelo y hardware utilizado con la finalidad de obtener resultados válidos.
2. La cantidad de polos utilizados en el modelo de la aproximación puede ser un factor de error al momento de interpretar los resultados.
3. El método de discriminación por energías nos permite determinar cuáles son los modos dominantes en los resultados obtenidos por el algoritmo.
4. Se debe contar con parámetros que evalúen el desempeño del algoritmo para proporcionar información al usuario sobre el grado en que el modelo utilizado se aproxima a las mediciones.
5. Es posible utilizar la tecnología actualmente disponible en las subestaciones de potencia para implementar algoritmos que van más allá de los esquemas de protección tradicional y que proporcionan información adicional para la toma de decisiones.

Como sugerencias para trabajos futuros se propone lo siguiente:

1. Implementar un método de sintonización de los parámetros del algoritmo en línea, el cual se ajuste de manera automática utilizando señales de prueba.
2. Emplear ventanas de longitud de muestras variable que permita optimizar los recursos computacionales.
3. Utilizar el algoritmo de Prony programado para la implementación de un sistema de acción remedial para resolver algún problema de tiro de carga o de generación.

Bibliografía

- [1] M. L. Crow, E. Barocio, C. Canizares, J. H. Chow, L. Dosiek, H. Ghasemi, M. Gibbard and L. Haarla, "Identification of Electromechanical Modes in Power Systems," IEEE PES, 2012.
- [2] J. F. Hauer, C. J. Demeure and L. L. Scharf, "Initial Results In Prony Analysis of Power System Response Signals," *IEEE Transactions on Power Systems*, vol. 5, 1990.
- [3] W. Liang, H. Kang and L. Yao, "Detection of Power System Oscillation Using Moving Window Prony Method," in *International Conference on Power System Technology*, 2010.
- [4] N. Zhou, Z. Huang, F. Tuffner, J. Pierre and S. Jin, "Automatic Implementation of Prony Analysis for Electromechanical Mode Identification from Phasor Measurements".
- [5] Y. Gong and A. Guzmán, "Synchrophasor-Based Online Modal Analysis to Mitigate Power System Interarea Oscillation," *SEL Journal of Reliable Power*, vol. 2, 2011.
- [6] V. S. Patel, S. K. Bhil, F. S. Kazi and S. R. Wagh, "Energy-Sorted Prony Analysis for Identification of Dominant Low Frequency Oscillations," in *Australian Control Conference*, 2013.
- [7] P. Kundur, *Power System Stability and Control*, 1994: McGraw-Hill.
- [8] K. P. Poon and K. C. Lee, "Analysis of Transient Stability Swings in Large Interconnected Power Systems by Fourier Transformation," *IEEE Transactions on Power Systems*, vol. 3, pp. 1573-1581, 1988.
- [9] P. Korba, M. Larsson and C. Rehtanz, "Detection of Oscillations in Power Systems Using Kalman Filtering Techniques," *IEEE Conf. Control Applications*, vol. 1, pp. 183-188, 2003.
- [10] M. Meunier and F. Brouaye, "Fourier Transform, Wavelets, Prony Analysis: Tools for Harmonics And Quality Of Power," in *Proc. 8th Int. Conf. Harmonics and Quality of Power*, 1998.
- [11] L. Ding, A. Xue, F. Han, J. Li, M. Wang, T. Bi and J. Wang, "Dominant Mode Identification for Low Frequency Oscillations of Power Systems based on Prony Algorithm," in *International Conference on Critical Infrastructure*, 2010.
- [12] N. Zhou, J. Pierre and D. Trudnowski, "Some Considerations in Using Prony Analysis to Estimate Electromechanical Modes," in *IEEE Power & Energy Society General Meeting*, 2013.
- [13] J. Kiusalaas, *Numerical Methods en Engineering with MATLAB*.
- [14] M. Hydro, "<https://www.rtds.com/>," Manitoba Hydro. [Online].
- [15] N. Watson and J. Arrillaga, *Power Systems Electromagnetic Transients Simulation*, IET, 2003.

- [16] *AcSelerator RTACSEL-5033 Software, Instruction Manual*, Schweitzer Engineering Laboratories, 2017.
- [17] *SEL-351S Protection System, Instruction Manual*, Schweitzer Engineering Laboratories, 2017.
- [18] *AcSelerator Quickset, SEL-5030, Instruction Manual*, Schweitzer Engineering Laboratories.
- [19] K.-H. John and M. Tiegelkamp, IEC-61131-3: Programming Industrial Automation Systems, Springer, 2001.
- [20] SVP Plus IEC 61131 Library for AcSELerator RTAC Projects, Pullman, WA: Schweitzer Engineering Laboratories, 2014.
- [21] D. Maryam, B. Shayanfard and A. R. Khayatian, "PMU Ranking Based on Singular Value Decomposition of Dynamic Stability Matrix," *IEEE Transactions on Power Systems*, vol. 28, 2013.