Interfaz gráfica para el cálculo de flujos de potencia en redes eléctricas

Jorge Arturo Pérez Venzor[1], Oscar Núñez Ortega[1], Néstor Abraham Corchado Nevarez[1], Raúl Ortiz Chavarría[1], Abel Eduardo Quezada Carreón[1], Lidia Hortencia Rascón Madrigal, Osiel Ramírez Sandoval[1]

[1]Instituto de Ingeniería y Tecnología, Universidad Autónoma de Ciudad Juárez.

Resumen

En el siguiente artículo se muestra el diseño de una interfaz gráfica para el cálculo de flujos de potencia en una red eléctrica, con el fin de tener una herramienta que facilite la obtención de los cálculos. Se presenta el desarrollo de las interfaces graficas (GUI'S), los códigos en MATLAB que se utilizan para crearlas así como el método matemático que desarrolla. Se explica cómo se desarrolla la interfaz gráfica basándose en códigos que utilizan método iterativo de Newton-Raphson.

Palabras clave: GUI, Interfaz gráfica, Newton-Raphson, flujos de potencia.

Introducción

Cuando se realiza un análisis de un sistema de potencia, normalmente, se estudia una red formada por un generador principal, un transformador elevador, una línea de transmisión, tomando en cuenta que hay otro transformador que reduce el voltaje y por ultimo una carga conectada. En cambio de los sistemas eléctricos de potencia que en el mundo son una compleja red eléctrica que cuenta con una gran cantidad de elementos interconectados (Glover, & Sarma, 2003).

Los estudios de flujos de potencia son de gran utilidad, para sincronizar en que cantidad entregan energía a las distintas plantas generadoras conectadas en una red eléctrica, es un factor que nos permita conocer la forma y la cantidad de energía que nos aportan las plantas generadoras a nuestro sistema, además de saber la magnitud y el ángulo de fase del voltaje así como determinar la potencia reactiva y la potencia real que fluye en cada línea del sistema, todos estos resultados se expresan en valores por unidad (P.U).

Con este programa computacional se pueden reducir los costos ya que los programas comerciales son de un costo demasiado elevado (Power World Corporation, 2015), que hace difícil su adquisición, es por esta razón que se está desarrollando una herramienta que permite resolver redes eléctricas medianas.

La interfaz gráfica se crea con un apartado de MATLAB el cual es capaz de crearlas, agregando los códigos que contienen el método matemático al código raíz.

Métodos

El procedimiento para calcular estos datos es un proceso algo tardado, incluso en un estudio de una red con cinco nodos, y debido al número de variables que son usadas para su cálculo, es posible cometer errores. Por esta razón se toma la decisión de crear un programa, con las características necesarias para ser una herramienta aceptable y funcional. Este se crea resolviendo las ecuaciones que se requieren para llegar al resultado, utilizando el método de Newton-Raphson. Esta herramienta se diseña con un código crea en MATLAB y con la ayuda de la librería Interfaces Graficas de Usuario o GUI, (*Graphical User Interfaces*) para crear una interfaz amigable y de fácil comprensión, para que, con solo capturar los datos de un problema planteado en algún libro de texto o verídico se obtengan los ángulos y magnitudes del voltaje.

El método Newton-Raphson es un método iterativo que nos permite aproximar la solución de una ecuación del tipo $f_{(x)=0}$. Partiendo de una estimación inicial de la solución x_0 y construyendo una aproximación o sucesión de forma continua mediante la fórmula, el método de Newton-Raphson es un método para resolver ecuaciones algebraicas no lineales (Guzmán, 2012).

Considérese un sistema de n ecuaciones algebraicas no lineales:

$$f(x_1, x_2, \dots, x_n) = 0 \,;\, i = 1,2, \dots, n \qquad (1)$$

Supongamos que los valores iniciales de las incógnitas son las siguientes:

$x^0_{1\,,} x^0_{1,}$,..., $x^0_{n.}$ sean $\Delta x^0_{1,} \Delta x^0_2, \dots, \Delta x^0_{n.}$

Da la solución siguiente:

$$fi(x^0_1 + \Delta x^0_1, x^0_2 + \Delta x^0_2, \dots, x^0_n + \Delta x^0_n) = 0; \quad i = 1,2, \dots, n \qquad (2)$$

Al desarrollar esta ecuación, se obtiene lo siguiente:

$$fi(\, x^0_1 \,, x^0_2 \,, \dots, x^0_n) + [(\frac{\partial fi}{\partial x1})^0 \,\Delta x^0_1 + (\frac{\partial fi}{\partial x2})^0 \,\Delta x^0_2 + \dots + (\frac{\partial fi}{\partial xn})^0 \,\Delta x^0_n]$$

Más términos de mayor orden igual a 0

Donde $\frac{\partial fi}{\partial x_1})^0$, $(\frac{\partial fi}{\partial x_2})^0$, ... , $(\frac{\partial fi}{\partial x_n})^0$ son las derivaciones de f_i con respecto a $x_1, x_2, \dots, x_n$ evaluadas en ($x^0_{1\,,} x^0_{1,}$,..., $x^0_{n.}$) Si los términos de orden superior se desprecian, puede escribirse en forma matricial:

$$\begin{bmatrix} f_1{}^0 \\ f_2{}^0 \\ \vdots \\ f_n{}^0 \end{bmatrix} + \begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)^0 & \left(\frac{\partial f_1}{\partial x_2}\right)^0 & \cdots & \left(\frac{\partial f_1}{\partial x_n}\right)^0 \\ \left(\frac{\partial f_2}{\partial x_1}\right)^0 & \left(\frac{\partial f_2}{\partial x_2}\right)^0 & \cdots & \left(\frac{\partial f_2}{\partial x_n}\right)^0 \\ \vdots & \vdots & & \vdots \\ \left(\frac{\partial f_n}{\partial x_1}\right)^0 & \left(\frac{\partial f_n}{\partial x_2}\right)^0 & \cdots & \left(\frac{\partial f_n}{\partial x_n}\right)^0 \end{bmatrix} \begin{bmatrix} \Delta \mathrm{x}_1^0 \\ \Delta \mathrm{x}_2^0 \\ \vdots \\ \Delta \mathrm{x}_n^0 \end{bmatrix} \cong \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

O en forma de matriz vectorial:

$$f^0 + j^0\, \Delta x^0 \cong 0 \qquad (3)$$

Se conoce como matriz jacobiana a j^0 (que se obtiene al diferenciar el vector función f^0 con respecto a x y se evalúa en (x^0). La ecuación anterior se puede escribir como:

$$f^0 \cong [-j^0]\, \Delta x^0 \qquad (4)$$

Se puede obtener valores aproximados de corrección Δx^0 Como estos constituyen un sistema de ecuaciones algebraicas lineales se pueden resolver de manera eficiente mediante triangulación y sustitución.

Los valores de x son entonces: $x^1 = x^0 + \Delta x^0$

O en general para la iteración $(r+1) - estimacion.$

$$x^{(r+1)} = x^{(r)} + \Delta x^{(r)} \qquad (5)$$

Las iteraciones se continúan hasta que la ecuación (1) se satisfaga para cualquier exactitud deseada, es decir:

$$\left|fi(x^{(r)})\right| < \varepsilon \text{ (Un valor especificado);} \quad i = 1,2 \ldots n \qquad (6)$$

El método requiere de un numero iteraciones bastante amplio, las cuales requieren de tiempo, por lo tanto este diseño de ecuaciones son escritas con un código, que facilita su manejo y disminuye la posibilidad de que cometer errores (Guzmán, 2012).

Con este método se estructuran las funciones creadas en MATLAB que siguen una secuencia para llegar a la solución. Al iniciar se corre la función YBus (Acha, Fuerte-Esquivel, Ambriz-Perez, & Angeles-Camacho, 2004) que se encarga de crear las matrices de admitancias. La figura 1 muestra cómo se obtiene dicha matriz.

Después de que la matriz de admitancias es calculada se inicia con el cálculo con el método Newton-Raphson (Acha, Fuerte-Esquivel, Ambriz-Perez, & Angeles-Camacho, 2004), que es el cálculo final para obtener los valores para el ángulo y magnitud del voltaje por nodo.

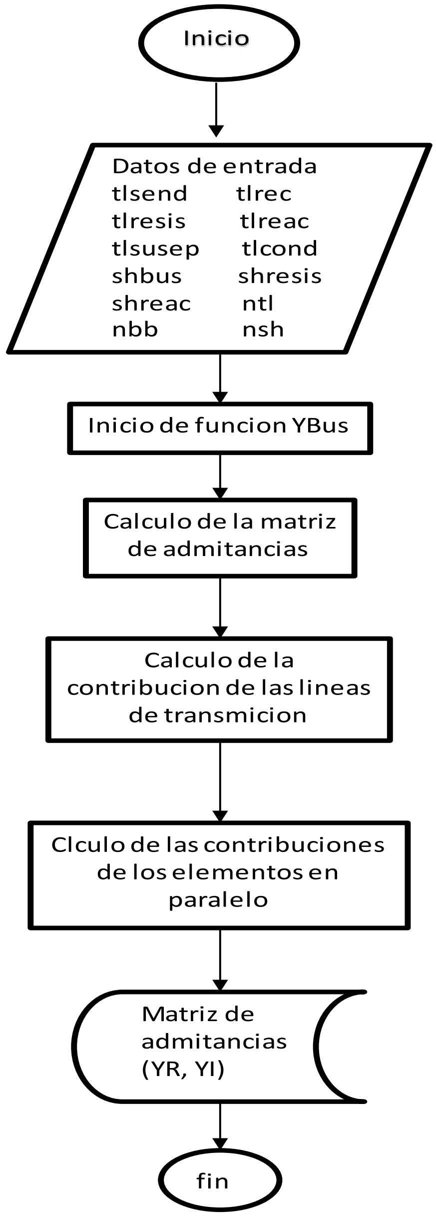

Figura 1: Diagrama de flujo para YBus. (Granada Echeverri, 2001)

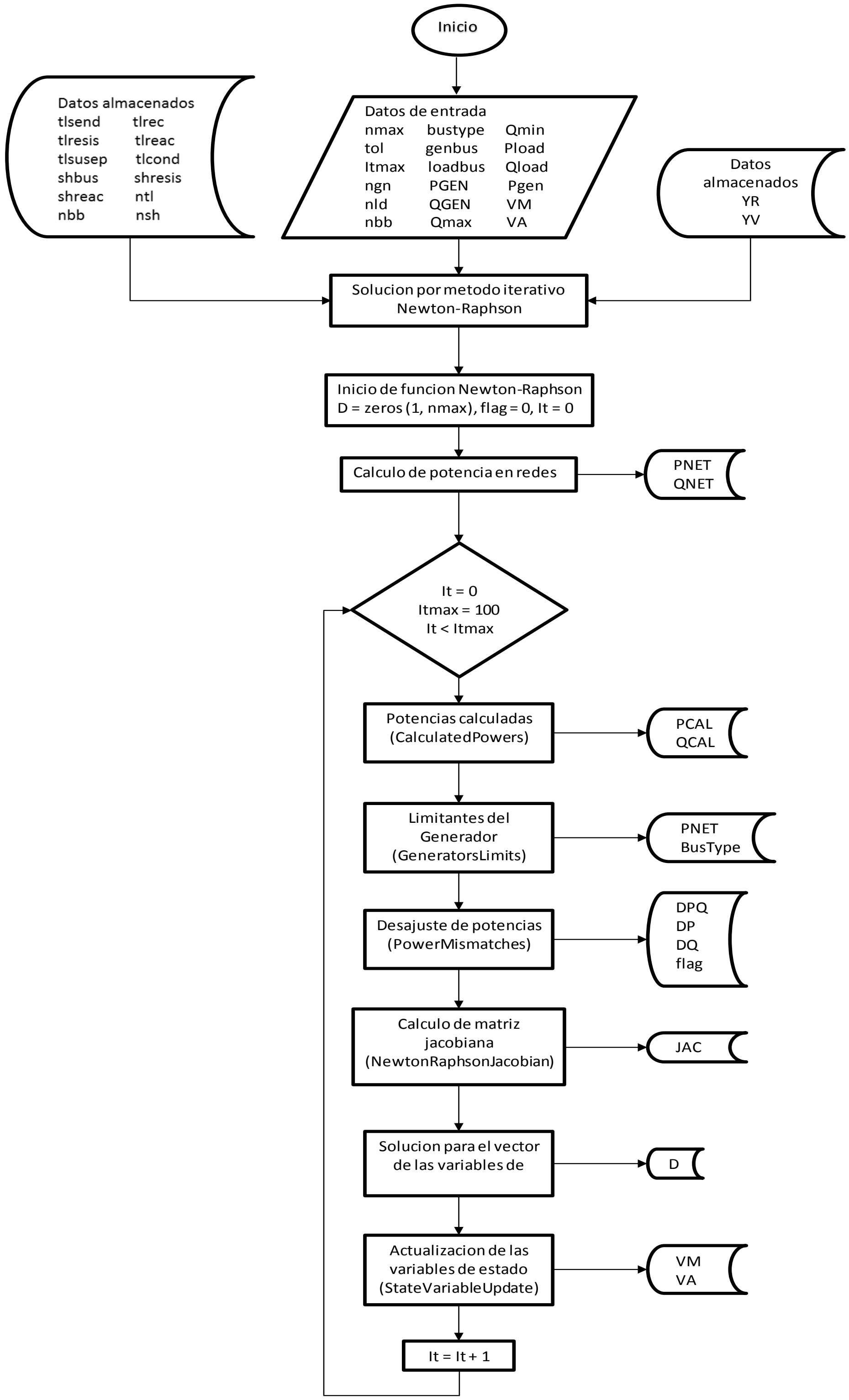

Figura 2: Diagrama de flujo para Newton-Raphson (Granada Echeverri, 2001)

La interfaz gráfica cuenta con diversas herramientas las cuales ayudaran a llegar a la solución del sistema, como lo representa los diagramas de flujos.

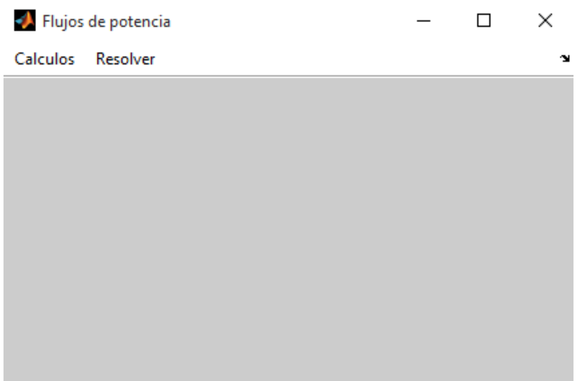

Figura 3: Pantalla principal (Fernández de Córdoba, 2007)

La interfaz gráfica cuenta con tres pantallas, la primera es la figura 3, en la cual muestra un menú donde seleccionas las funciones que se necesitan para resolver el problema de flujos de potencia. En la primera opción es “Cálculos” donde se encuentra la función YBus, que esta referenciada a la figura 1, que nos muestra el curso del sistema, La segunda es la encargada de desplegar la función Newton-Raphson, la cual esta referenciada en la figura 2.

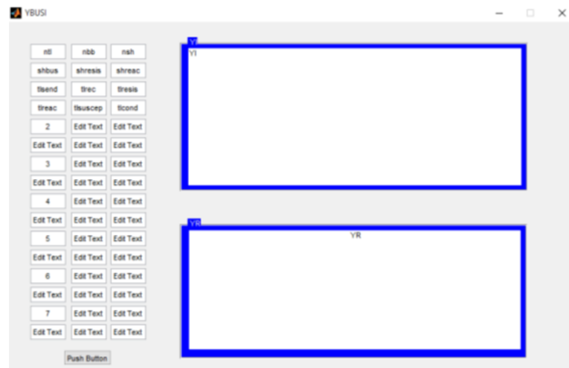

Figura 2: Pantalla YBus (Fernández de Córdoba, 2007)

Una de las pantalla que se encarga de resolver el problema es YBus se encarga de adquirir los valores de admitancias que están referenciados como datos de entrada en la figura 1.

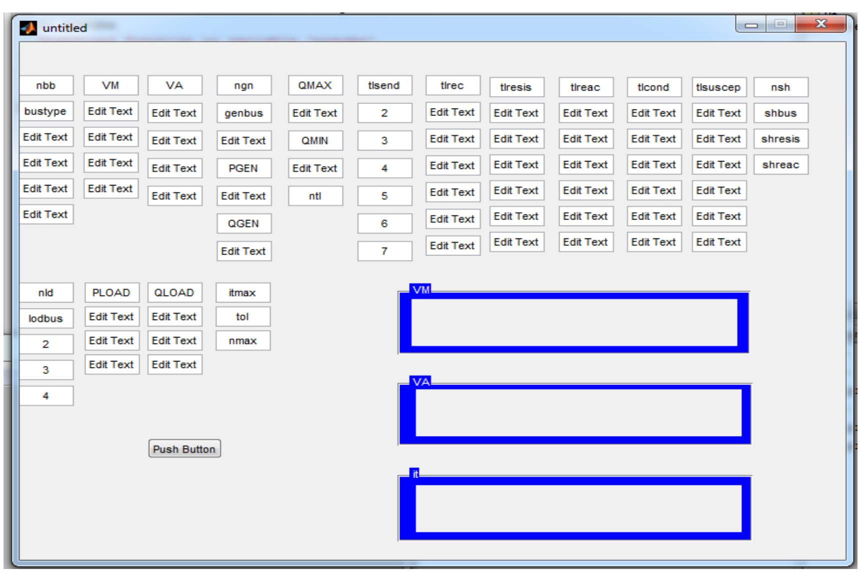

Figura 3: Newton Raphson (Fernández de Córdoba, 2007)

Resultados

Para probar el funcionamiento de la interfaz gráfica se toma un ejemplo, el cual consta de una red de 5 nodos, 2 generadores y 7 redes de transmisión (Acha, Fuerte-Esquivel, Ambriz-Perez, & Angeles-Camacho, 2004). En este mismo ejemplo se cuenta con los datos necesarios para realizar el cálculo. Obtienen los siguientes resultados, los cuales coinciden con los del ejemplo.

En las siguientes tablas se muestran las matrices de admitancias, estas matrices se calculan con la función representada en la figura 1.

Tabla 1. YI

-18.695	15	3.75	0	0
15	-32.415	5	5	7.5
3.75	5	-38.695	30	5
0	5	30	-38.695	3.75
0	7.5	0	3.75	-11.21

Tabla 2. YR

6.25	-5	-1.25	0	0
-5	10.833	-1.666	-1.666	-2.5
-1.25	-1.666	12.916	-10	0
0	-1.666	-10	12.916	-1.25
0	2.5	0	-1.25	3.75

Las siguientes tablas muestran los valores que se obtienen a partir de la figura 2. La tabla 3 muestra los valores del voltaje en cada uno de los buses. Estos valores se encuentran en un rango de 0.9 a 1.1 P.U's, lo que indica que se encuentran en un rango aceptable. Las tablas 4 y 5 muestran el valor angular de cada uno de los voltajes, se muestran en grados y radianes respectivamente y la tabla numero 6 nos indica en número de iteraciones totales para llegar a estos resultados.

Tabla 3. VM

1	2	3	4	5
1.06	1.00	0.987	0.984	0.917

Resultados representados en Angulo.

Tabla 4. VA

1	2	3	4	5
0.00	-2.06	-4.64	-4.96	-5.77

Resultados representados en radianes.

Tabla 5. VA

1	2	3	4	5
0.00	-0.035	-0.080	-0.086	-0.100

Iteraciones.

100

Conclusiones

Al realizar la interfaz gráfica y probarla se observa un buen desempeño y la obtención de los datos que se esperaban, además que se hace de una forma más fácil. Trabajar en el diseño de GUI's en MATLAB es una forma amigable y de un proceso sencillo. También que es posible aplicar los conocimientos de eléctrica a sistemas computacionales, creando herramientas que facilitan procesos para el análisis de redes eléctricas.

Se observa que esta GUI se puede mejorar para obtener un programa que sea capaz de resolver una red de un número mayor de nodos, puede hacerse más robusta y con más elementos.

Referencias

Acha, E., Fuerte-Esquivel, C. R., Ambriz-Perez, H., & Angeles-Camacho, C. (2004). FACTS: modelling and simulation in power networks. John Wiley & Sons.

Fernández, J.C. & Fuentes López, E. E. (2011). Modelo de Flujos Óptimos de Potencia Utilizando Técnicas de Optimización, Antiguo Cuscatlán, EL SALVADOR, C.A.

Fernández de Córdoba, G. (2007). Creación de Interfaces Gráficas de Usuario (GUI) con MatLab, Salamanca.

Franco, J. I. (2002). Estudio de Flujos de Potencia y Analisis de Fallas en Sistemas Electricos de Distribucion Radial. San Nicolas de los Garza. Universidad Autonoma de Nuevo Leon.

Glover, J. D., & Sarma, M. S. (2003). Sistemas de potencia: análisis y diseño. Cengage Learning Editores.

Granada Echeverri, M (2001). Flujos de potencia óptimos para sistemas de distribución usando los métodos de la cadena y del gradiente, Tesis. Universidad Tecnológica de Pereira, Colombia

Guzmán, M. (2012) Flujos de potencia con Matlab, Facultad de ingeniería mecánica eléctrica cuidad Mendoza Veracruz, junio del 2012, 84 hojas.

Izquierdo J. (2002). Estudio de potencia y análisis de fallas en sistemas eléctricos de distribución radial, Santa Nicolás de las garzas, N. L.

Oñate, P. E. (2008). Solución del problema de flujos de potencia óptimo con restricciones de seguridad por un optimizador de partículas modificado, Guadalajara.

Power World Corporation. (2015, 31 de Enero). PowerWorld Corporation. Obtenido de http://www.powerworld.com/products/simulator/overview.