XXVII Verano de la Ciencia

Universidad de Guanajuato

Manual de herramientas de co-simulación para sistemas eléctricos

UNIVERSIDAD DE GUANAJUATO

Nombre del proyecto: Implementación de micro redes eléctricas en sistemas de distribución mediante herramientas de co-simulación.

Autores: Israel Bautista Vázquez.

Juana Remedios Arvizu Lara.

Omar Lara Diosdado.

Tutor: Guillermo Tapia Tinoco.

Fecha: 22 de junio de 2022.

Índice.

Introducción.

El presente manual muestra una guía con los pasos a realizar para la implementación de redes eléctricas en sistemas de distribución mediante herramientas de co-simulación. Dichas herramientas permiten la interacción armónica de diferentes softwares de simulación y el intercambio de información entre ellos. En este trabajo se implementa la co-simulación entre el software OpenDSS y el software Matlab para analizar el impacto de la generación distribuida en sistemas de distribución de redes eléctricas. Explotando las ventajas de ambos softwares de simulación, se utiliza OpenDSS para implementar el modelo en estado estable de la red eléctrica y calcular su flujo de potencia. Por su parte en Matlab se implementa el modelo detallado de sistemas de generación solar fotovoltaica.
En este manual se revisan las partes de la co-simulación con sus respectivos códigos realizados en cada software para su funcionamiento, para ello el lector debe tener conocimientos básicos de programación y estar familiarizado con la interfaz de Matlab y OpenDSS.

CO-SIMULACIÓN, MATLAB Y OPENDSS.

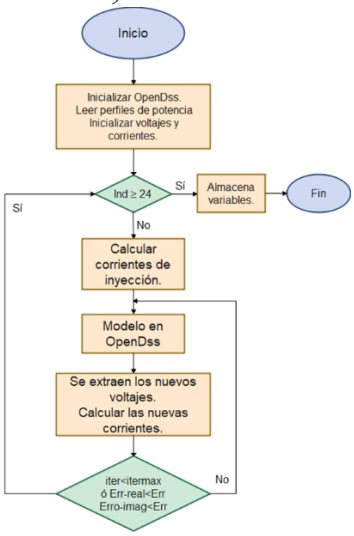

Figura 1 muestra el diagrama de flujo del proceso de co-simulación descrito en este manual. El algoritmo comienza con la etapa de inicialización donde se establece la conexión entre ambos softwares de simulación, se cargan perfiles de potencia y se inicializan las variables utilizadas.
Posteriormente, pasa a un ciclo "*For*" el cual permite simular escenarios de operación. En este caso toma el valor de 24 debido a que se analizará un escenario de un día con intervalos de tiempo de 1h. Antes de entrar al lazo que calcula el flujo de potencia se inicializan las corrientes de inyección utilizando para su cálculo la potencia activa inyectada y su respectivo voltaje en el bus de conexión. Posteriormente se compila y soluciona el modelo en OpenDSS, el cual arroja como salida los voltajes y corrientes de los elementos que conforman la red. A partir de estos resultados se verifica el error de convergencia y el número de iteraciones. Hay dos maneras que el lazo se detenga: La primera se obtiene cuando se alcanzó el máximo número de iteraciones. En esta condición se detiene el algoritmo, pero no se obtuvo una solución adecuada y esta es desechada. La otra opción es que el error de la parte real e imaginaria calculado con el voltaje de la iteración anterior y la actual ($Err\_real \leq Err$ y ($Err\_imag \leq Err$) sean menores al error de convergencia (Err). Si ambas condiciones se cumplen antes que se llegue al máximo número de iteraciones se detiene el algoritmo y se guardan los resultados. El mismo procedimiento se repite hasta que el escenario de 24 h haya sido simulado.

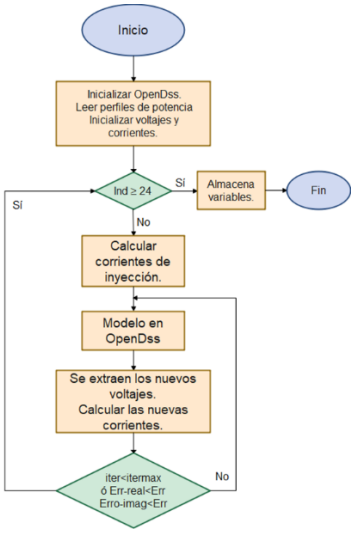

Figura 1. Diagrama de flujo de co-simulación entre OpenDSS y Matlab.

Planteamiento del problema.

En la Figura 2 se muestra la red eléctrica implementada en este caso de estudio. Dicha red corresponde a la red IEEE de 13 buses modificada. A la red se le agregan fuentes de corriente que permiten inyectar corriente en diferentes buses de la red. Estas fuentes de corriente son controladas desde Matlab donde se implementa el modelo de una micro red integrada por generación solar fotovoltaica. De esta manera es posible utilizar ambos softwares de simulación para analizar el impacto de la generación distribuida en la red de distribución.

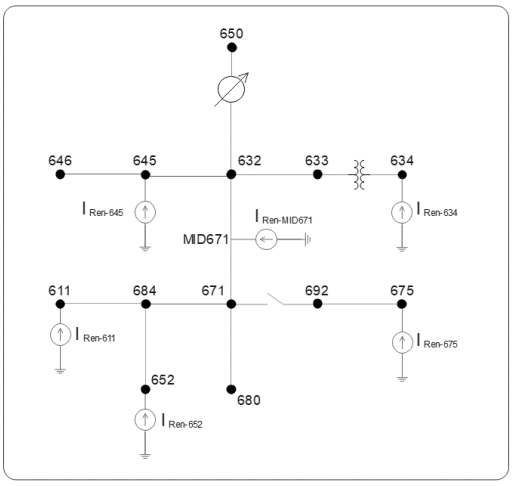

Figura 2. Red eléctrica IEEE 13 buses.

Código de Matlab.

Primeramente es desarrollada la programación correspondiente en el software Matlab que permititirá el intercambio de información con el software OpenDSS mediante una interfaz compatible. En la Figura 3 se muestra la inicialización del código en Matlab, en el cual, el comando “clear all” sirve para limpiar todas las variables utilzadas anterirormente y el comando “clc” limpia la pantalla prinicipal

“Comand Window”. Además, es necesario realizar la interfaz entre ambos softwares utilizando la función “Inic_opendss”, la cual es descrita a detalle en la Figura 4.

```
clear all;
clc;
Inic_opendss;
```

Figura 3. Códigos de inicialización en Matlab.

A continuación, en la Figura 4. Inic_OpenDSS. se muestra el código correspondiente para realizar la interfaz de los softwares OpenDSS y Matlab, en el cual el comando “actxserver (‘OpenDSSengine.DSS’)” es utilizado para realizar la búsqueda del software OpenDSS en la computadora y realizar la vinculación correspondiente. Seguidamente, el objeto definido como “DSSObj” se utiliza para ingresar a las propiedades de la interfaz DSS y para ejecutar el software se utiliza la función “DSSObj.Start(0)”. Si existe algún error de ejecución se mostrará en pantalla por el comando “disp”. Las últimas dos líneas de código son utilizadas para crear la interfaz de texto y la interfaz de circuito activo para poder editar las propiedades de los elementos del circuito.

```
DSSObj = actxserver('OpenDSSEngine.DSS')
if~DSSObj.Start(0),
    disp('Unable to start the OpenDSS Engine')
    return
end

DSSText = DSSObj.Text;
DSSCircuit = DSSObj.ActiveCircuit
```

Figura 4. Inic_OpenDSS.

En la Figura 5. Perfiles solares. se determinan las potencias que son inyectadas al sistema de distribución, mediante la variable “P_SOLAR” son almacenados los valores de las potencias por hora a partir de mediciones reales extraídas de una base de datos cuyo nombre es “perfiles_solar.csv”, que con el comando “csvread” se leen todos los datos. Posteriormente se eligieron arbitrariamente diferentes días del año para determinar la potencia producida por los perfiles solares. Se utilizan seis perfiles de potencia donde la numeración utilizada está relacionada con el bus de conexión de las fuentes de corriente en la Figura 2. Cada variable es normalizada a 1 pu al dividir entre el valor máximo en cada variable. De esta manera es posible establecer la potencia de operación de cada generador.

```
P_SOLAR=csvread('perfiles_solar.csv',1,1);
dia675=10;dia632_671=20;dia634=93;dia645=189;dia611=237;dia652=8;

Per675=P_SOLAR(288*dia675 + 3:12:288*(dia675 + 1));
Per632_671=P_SOLAR(288*dia632_671 + 3:12:288*(dia632_671 + 1));
Per634=P_SOLAR(288*dia634 + 3:12:288*(dia634 + 1));
Per645=P_SOLAR(288*dia645 + 3:12:288*(dia645 + 1));
Per611=P_SOLAR(288*dia611 + 3:12:288*(dia611 + 1));
Per652=P_SOLAR(288*dia652 + 3:12:288*(dia652 + 1));

Per675=(Per675/max(Per675));
Per632_671=(Per632_671/max(Per632_671));
Per634=(Per634/max(Per634));
Per645=(Per645/max(Per645));
Per611=(Per611/max(Per611));
Per652=(Per652/max(Per652));
```

Figura 5. Perfiles solares.

En la Figura 6. Inicialización de variables. son inicializados los voltajes de cada bus que tendrá inyección de potencia activa. Se utiliza un voltaje nominal de 4160 V y una secuancia ABC para los buses trifásicos a expcepción del bus 634 que opera a un voltaje de 480 V. Por su parte para los buses monofasicos se utiliza el voltaje de fase y su correspopondiente ángulo dependiendo de su fase de conexión.

```
for ind=1:1:length(Per675)

    iter=0;iterMax=10;
    Err=0;
    Vb1=4160/sqrt(3)*[cosd(0)+j*sind(0);cosd(-120)+j*sind(-120);cosd(120)+j*sind(120)];
    Vb2=480/sqrt(3)*[cosd(0)+j*sind(0);cosd(-120)+j*sind(-120);cosd(120)+j*sind(120)];
    V675=Vb1;
    V611c=480/sqrt(3)*(cosd(120)+j*sind(120));
    V652a=480/sqrt(3)*(cosd(0)+j*sind(0));;
    V632_671=Vb1;
    V634=Vb2;
    V645b=480/sqrt(3)*(cosd(-120)+j*sind(-120));
```

Figura 6. Inicialización de variables.

En la Figura 7. Inyección de potencia del caso base. se muestra la inicialización de las variables utilizadas para almacenar la potencia activa inyectada en cada bus. Se puede observar que en el caso de buses trifásicos se utiliza un vector de 3x1 y para los monofásicos un escalar.

```
    Pren_675=[0;0;0];
    Pren_652a=0;
    Pren_611c=0;
    Pren_632_671=[0;0;0];
    Pren_634=[0;0;0];
    Pren_645b=0;
```

Figura 7. Inyección de potencia del caso base.

En la Figura 8. Inyección de potencia con corrientes modificadas. se implementó la inyección de potencia activa mediante cada una de las variables en las cuales se almacenan los valores de la nueva potencia. Para este caso de estudio todas las variables se ajustaron al valor máximo de 1000 W y su variación en cada intervalo de tiempo esta dado por el perfil de potencia. La posición de este valor en el vector es controlada por la variable “ind”

```
    Pren_675=[1;1;1]*1E3*Per675(ind);
    Pren_652a=1*1E3*Per652(ind);
    Pren_611c=1*1*1E3*Per611(ind);
    Pren_632_671=[1;1;1]*1E3*Per632_671(ind);
    Pren_634=[1;1;1]*1E3*Per634(ind);
    Pren_645b=1*1E3*Per645(ind);
```

Figura 8. Inyección de potencia con corrientes modificadas.

Para continuar con la implementación del código en Matlab, se debe calcular la corriente inyectada en cada uno de los buses en los cuales están interconectados los sistemas fotovoltaicos. En la Figura 9. Cálculo de fuentes de corriente en cada bus. se determinan y almacenan los valores en cada una de las variables a partir de su potencia y voltaje.

```
    Iren_675=conj(Pren_675./V675);
    Iren_611c=conj(Pren_611c./V611c);
    Iren_652a=conj(Pren_652a./V652a);
    Iren_632_671=conj(Pren_632_671./Vb1);
    Iren_634=conj(Pren_634./Vb2);
    Iren_645b=conj(Pren_645b./V645b);
```

Figura 9. Cálculo de fuentes de corriente en cada bus.

El intercambio de información entre Matlab y OpenDSS se realiza utilizando el comando de interconexión de texto “DSSText.command” y en la sección de “compile” se agrega la dirección en donde se encuentra almacenado el archivo creado en software OpenDSS nombrado como “IEE13_mod.dss” como se muestra en la Figura 10. Interfaz entre ambos softwares.. Con el comando “DSSSolution=DSSCircuit.Solution” se permite la modificación de las propiedades de los elementos del circuito, en este caso la red IEEE 13 buses. Esta parte de código se encuentra dentro del ciclo “while” y se invocará en cada solución del circuito en OpenDSS.

```
while (iter<iterMax)

    iter=iter+1;

    DSSText.Command = 'Compile (C:\Users\arviz\OneDrive\Escritorio\OneDrive_2022-07-15\Verano UG 2022\IEE13_mod.dss)';
    Isource=DSSCircuit.Isources;
    DSSSolution=DSSCircuit.Solution;
    DSSText.Command='Set MaxControlIter=30';
```

Figura 10. Interfaz entre ambos softwares.

Después de realizar la interfaz entre ambos softwares, se carga el valor de la magnitud y ángulo de fase en cada una de las fuentes de corriente del modelo en OpenDSS. El comando "Isource.first" se utiliza para tener acceso a la primera fuente de corriente en OpenDSS. En este caso corresponde a la fuente de corriente conectada en la fase A del bus 675 (Iren_675). Con el comando "Isource.next" se tiene acceso a la siguiente fuente de corriente y como se puede observar se utiliza repetidamente hasta completar todas las fuentes de corriente. Para modificar la magnitud de la fuente de corriente se utiliza el comando "Isource.Amps" y para modificar su ángulo de fase se utiliza el comando "Isource.Angle". En la Figura 11. Interfaz de fuentes de corriente líneas trifásicas. se muestra el código correspondiente para determinar la magnitud y ángulo de las fuentes de corriente en el bus 675.

```
    Isource.first
    Isource.Amps=abs(Iren_675(1,1));
    Isource.Angle=angle(Iren_675(1,1))*180/pi;

    Isource.next
    Isource.Amps=abs(Iren_675(2,1));
    Isource.Angle=angle(Iren_675(2,1))*180/pi;

    Isource.next
    Isource.Amps=abs(Iren_675(3,1));
    Isource.Angle=angle(Iren_675(3,1))*180/pi;
```

Figura 11. Interfaz de fuentes de corriente líneas trifásicas.

En la Figura 12 se muestran los comandos utilizados para modificar la magnitud y ángulo de fase de la fuente de corriente conectada al bus monofásico 611.

```
    Isource.next
    Isource.Amps=abs(Iren_611c(1,1));
    Isource.Angle=angle(Iren_611c(1,1))*180/pi;
```

Figura 12. Interfaz de fuentes de corriente líneas monofásicas.

En la Figura 13. Exportación de datos a archivos Excel. se muestran las líneas de código donde se realiza la exportación de las nuevas corrientes y voltajes solucionados en el software de OpenDSS, mediante el comando "solve mode=snap" se soluciona el modelo y se calcula el flujo de potencia. Con el comando "Export voltajes" se exportan los voltajes de los buses y con el comando "Export currentes" se exportan las corrientes de todos los elmentos que conforman la red. Esta información es almacenda en dos archivos de excel con la extensión "csv". Para una mejor manipulación de estos resultados la información de los archivos en excel es almacenado en las variables "V_opendss" y "I_opendss" respectivamente.

```
    DSSText.Command = 'solve mode =snap';
    DSSText.Command = 'Export Voltages';
    DSSText.Command = 'Export Currents';

    V_opendss=csvread('IEEE13_mod_EXP_VOLTAGES.csv',1,1);
    I_opendss=csvread('IEEE13_mod_EXP_CURRENTS.csv',1,1);
```

Figura 13. Exportación de datos a archivos Excel.

Posteriormente se necesita un procesamiento de los resultados y es necesario conocer las posiciones de los voltajes. Con las líneas de código de la Figura 14. Determinación de los voltajes en cada uno de los buses., se extraen únicamente los voltajes de los buses donde se tienen ubicadas las fuentes de corriente.

```
V675_n=[V_opendss(9,3)*(cosd(V_opendss(9,4))+j*sind(V_opendss(9,4)));V_opendss(9,7)*(cosd(V_opendss(9,8))+j*sind(V_opendss(9,8)));
    V_opendss(9,11)*(cosd(V_opendss(9,12))+j*sind(V_opendss(9,12)))];
V611c_n=[V_opendss(10,3)*(cosd(V_opendss(10,4))+j*sind(V_opendss(10,4)))];
V652a_n=[V_opendss(11,3)*(cosd(V_opendss(11,4))+j*sind(V_opendss(11,4)))];
V632_671_n=[V_opendss(12,3)*(cosd(V_opendss(12,4))+j*sind(V_opendss(12,4)));V_opendss(12,7)*(cosd(V_opendss(12,8))+j*sind(V_opendss(12,8)));
    V_opendss(12,11)*(cosd(V_opendss(12,12))+j*sind(V_opendss(12,12)))];
V634_n=[V_opendss(4,3)*(cosd(V_opendss(4,4))+j*sind(V_opendss(4,4)));V_opendss(4,7)*(cosd(V_opendss(4,8))+j*sind(V_opendss(4,8)));
    V_opendss(4,11)*(cosd(V_opendss(4,12))+j*sind(V_opendss(4,12)))];
V645b_n=[V_opendss(6,3)*(cosd(V_opendss(6,4))+j*sind(V_opendss(6,4)))];
```

Figura 14. Determinación de los voltajes en cada uno de los buses.

En la Figura 15. Cálculo de las nuevas corrientes. se muestra el código para calcular el valor de cada una de las nuevas corrientes a partir de las potencias calculadas en la Figura 8. Inyección de potencia con corrientes modificadas. y los voltajes determinados en la Figura 14. Determinación de los voltajes en cada uno de los buses. que serán inyectadas nuevamente en cada uno de los buses con fuentes de corriente y se repiten los pasos anteriores hasta lograr una convergencia.

```
Iren_675=conj(Pren_675./V675_n);
Iren_611c=conj(Pren_611c./V611c_n);
Iren_652a=conj(Pren_652a./V652a_n);
Iren_632_671=conj(Pren_632_671./V632_671_n);
Iren_634=conj(Pren_634./V634_n);
Iren_645b=conj(Pren_645b./V645b_n);
```

Figura 15. Cálculo de las nuevas corrientes.

En este paso se determinan los errores de cada una de las iteraciones realizadas por el modelo como se muestra en la Figura 16. Determinación de las variables de control., pero al tratarse de números complejos es importante calcular el máximo error en su parte real como imaginaria, donde se hace la diferencia entre el valor del voltaje de la iteración actual y el valor del voltaje de la iteración anterior. Estos errores se almacenar en el vector “Err” y solamente se toma el error máximo de la parte real e imaginaria para checar la convergencia.

```
Err=[V675-V675_n;V611c-V611c_n;V652a-V652a_n;V632_671-V632_671_n;V634-V634_n;V645b-V645b_n];
Err_real=max(real(Err));
Err_imag=max(imag(Err));
```

Figura 16. Determinación de las variables de control.

Mas adelante, se requieren actualizar los nuevos voltajes para determinar los valores que serán necesarios para los nuevos cálculos de las próximas iteraciones como se muestra en la Figura 17. Actualización de los nuevos voltajes., donde los voltajes con la letra “n”, antepuesta por un guion bajo indican el valor del voltaje de la iteración actual.

```
V675=V675_n;
V611c=V611c_n;
V652a=V652a_n;
V632_671=V632_671_n;
V634=V634_n
V645b=V645b_n;
```

Figura 17. Actualización de los nuevos voltajes.

Para finalizar el código es necesario que la parte real e imaginaria del error calculado en la Figura 16 sea menor al parámetro establecido por el desarrollador, que en este caso es el 1% o 0.01 como se muestra en la Figura 18.

```
    if (Err_real<0.01 && Err_imag<0.01)
        break
    end

end
```

Figura 18. Finalización del código.

El último paso es almacenar los resultados de la potencia activa inyectada por las fuentes de corriente y los voltajes en los buses. La variable “ind” controla la posición en el vector y está asociada a cada intervalo de tiempo del escenario implementado.

```
    P675_res(:,ind)=V675.*conj(Iren_675);
    P611c_res(:,ind)=V611c.*conj(Iren_611c);
    P652a_res(:,ind)=V652a.*conj(Iren_652a);
    P632_671_res(:,ind)=V632_671.*conj(Iren_632_671);
    P634_res(:,ind)=V634.*conj(Iren_634);
    P645b_res(:,ind)=V645b.*conj(Iren_645b);

    V632_671_res(:,ind)=V632_671;
    V675_res(:,ind)=V675;
    V634_res(:,ind)=V634;
    V611_res(:,ind)=V611c;
    V645_res(:,ind)=V645b;
    V652_res(:,ind)=V652a;

end
```

Figura 19. Extracción de los voltajes y potencias de salida.

Código de OpenDSS.

Los códigos realizados en el software OpenDSS inician con el comando “clear”, su función es evitar fallos y errores al compilar, eliminando de la memoria todas las configuraciones, simulaciones y archivos temporales que existan de análisis anteriores.

```
Clear
```

Figura 20. Comando Clear OpenDSS.

Para agregar un nuevo elemento o iniciar con un nuevo circuito se utiliza el comando “New Object”, usualmente se omite “object” por el nombre del elemento, como se observa en la Figura 21 lleva por nombre el elemento que es un circuito y posteriormente el nombre que recibe que en este caso es “circuit.IEEE13_mod”. En la siguiente línea de código se especifica el voltaje base de operación en kilo-volts “basekV=4.16”, se asigna el valor de 1 pu, número de fases “phases=3” y se asigna el bus de conexión “650”. Adicionalmente se asigna un ángulo de fase “Angle=0”. Los valores de MVAsc3=100000 y MVASC1=10000 corresponde a la potencia de corto circuito y estos valores dan como resultado una impedancia muy pequeña y por lo tanto obteniendo una fuente ideal.

```
new circuit.IEEE13_mod
~ basekv=4.16 pu=1.0 phases=3 bus1=650
~ Angle=0
~ MVAsc3=100000 MVASC1=100000    ! stiffen the source to approximate inf source
```

Figura 21. Fuente de alimentación en bus principal.

En la Figura 22 se muestra el código para implementar 3 reguladores idénticos, que son modelados como un transformador monofásico los cuales se conectan entre los buses 650 y RG650. Al ser un nuevo elemento se agrega el comando “New Transformer” para cada uno de los trasformadores, nombrados como Reg1, Reg2 y Reg3. Para indicar que se trata de un transformador monofásico se utiliza “phases=1”. El parámetro XHL=0.01 determina la reactancia entre el devanado de alta y baja tensión. La potencia de los devanados del transformador se asigna en KVAs y esta tiene un valor de “KVAs= [1666 1666]”.

En la siguiente línea se asignan los buses de conexión del devanado primario y secundario respectivamente "[650.1 R660.1]". Para indicar la fase al número de los buses se le agrega "." seguido de los números 1, 2 o 3, donde 1 es para la fase A, 2 es para la fase B y 3 es para la fase C. Los voltajes de operación de ambos devanados se dan en kilo-volts "kVs= [2.4 2.4].

```
New Transformer.Reg1 phases=1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4  2.4] %LoadLoss=0.01
new regcontrol.Reg1  transformer=Reg1 winding=2  vreg=122  band=2  ptratio=20 ctprim=700  R=3   X=9

New Transformer.Reg2 phases=1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.2 RG60.2] kVs=[2.4  2.4] %LoadLoss=0.01
new regcontrol.Reg2  transformer=Reg2 winding=2  vreg=122  band=2  ptratio=20 ctprim=700  R=3   X=9

New Transformer.Reg3 phases=1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.3 RG60.3] kVs=[2.4  2.4] %LoadLoss=0.01
new regcontrol.Reg3  transformer=Reg3 winding=2  vreg=122  band=2  ptratio=20 ctprim=700  R=3   X=9
```

Figura 22. Implementación de 3 reguladores.

Las siguientes líneas de código que se ilustran en la Figura 23, representan el transformador que se encuentra entre los buses 633 y 634 denominado XFM1. Como se trata de un transformador trifásico este se indica como "Phases=3". Después se indica el número de devanados que lo conforman con el comando "windings=2". En seguida se modela cada uno de los devanados que se denotan con el comando "wdg", se especifica la conexión que se tiene para este transformador es conexión estrella. Además, se agregan los parámetros de operación como voltaje "kv", potencia "kva", porcentaje de resistencias "r%" y dependiendo del devanado el porcentaje de reactancia "XHT" o "XLT".

```
New Transformer.XFM1  Phases=3   Windings=2  XHL=2
~ wdg=1 bus=633       conn=Wye kv=4.16    kva=500    %r=.55     XHT=1
~ wdg=2 bus=634       conn=Wye kv=0.480    kva=500    %r=.55   XLT=1
```

Figura 23. Modelado del transformador buses 633 a 632.

En la Figura 24 se puede visualizar el comando "redirect" que tiene como función redireccionar al archivo IEELineCodes.dss, las cuales se pueden observan en la Figura 25, que corresponden a los parámetros de las líneas de distribución. Estos parámetros se muestran como matrices triangulares inferiores en Ω/kft donde se indican la resistencia "rmatrix", reactancia "xmatrix" y capacitancia de la línea de distribución "cmatrix".

```
redirect IEEELineCodes.dss
```

Figura 24. Comando redirect.

Como se puede observar en la figura, se sigue el mismo procedimiento que para crear otro tipo de elemento: el comando "New", se le asigna un nombre "601", el número de fase y la frecuencia base

```
New linecode.601 nphases=3 BaseFreq=60
~ rmatrix = [0.065625    | 0.029545455  0.063920455  | 0.029924242  0.02907197  0.064659091]
~ xmatrix = [0.192784091 | 0.095018939  0.19844697   | 0.080227273  0.072897727  0.195984848]
~ cmatrix = [3.164838036 | -1.002632425  2.993981593 | -0.632736516  -0.372608713  2.832670203]
New linecode.602 nphases=3 BaseFreq=60
~ rmatrix = [0.142537879 | 0.029924242  0.14157197   | 0.029545455  0.02907197  0.140833333]
~ xmatrix = [0.22375     | 0.080227273  0.226950758  | 0.095018939  0.072897727  0.229393939]
~ cmatrix = [2.863013423 | -0.543414918  2.602031589 | -0.8492585  -0.330962141  2.725162768]
New linecode.603 nphases=2 BaseFreq=60
~ rmatrix = [0.251780303 | 0.039128788  0.250719697]
~ xmatrix = [0.255132576 | 0.086950758  0.256988636]
~ cmatrix = [2.366017603 | -0.452083836  2.343963508]
New linecode.604 nphases=2 BaseFreq=60
~ rmatrix = [0.250719697 | 0.039128788   0.251780303]
~ xmatrix = [0.256988636  | 0.086950758  0.255132576]
~ cmatrix = [2.343963508 | -0.452083836 2.366017603]
New linecode.605 nphases=1 BaseFreq=60
~ rmatrix = [0.251742424]
~ xmatrix = [0.255208333]
~ cmatrix = [2.270366128]
New linecode.606 nphases=3 BaseFreq=60
~ rmatrix = [0.151174242 | 0.060454545  0.149450758 | 0.053958333  0.060454545  0.151174242]
~ xmatrix = [0.084526515 | 0.006212121  0.076534091 | -0.002708333  0.006212121  0.084526515]
~ cmatrix = [48.67459408 | 0  48.67459408 | 0  0  48.67459408]
New linecode.607 nphases=1 BaseFreq=60
~ rmatrix = [0.254261364]
~ xmatrix = [0.097045455]
~ cmatrix = [44.70661522]
```

de “60” Hertz. En seguida se agregan las líneas de código de cada una de las matrices triangulares inferiores, tal que: $Z = R + jX$, donde R corresponde a la resistencia y X a la reactancia medida en Ohms/kft. Finalmente se crea la matriz de capacitancia en nano faradios/kft. Siguiendo este procedimiento para cada una de las líneas de la red eléctrica que se modele.

Figura 25. Código IEELineCodes.dss.

Para definir las cargas se agrega el comando “New load” y se le asigna el nombre con respecto al bus al que este interconectada, también se define la fase. Posteriormente se define el bus y fase, esto se puede realizar en el mismo comando “Bus1” si los parámetros son iguales. Si cambian es necesario definir cada una de las fases por separado, como se puede visualizar en la Figura 26 en la cual se puede visualizar ambos casos. También es necesario indicar como está conectada la carga con el comando “conn” y en seguida agregar si están en delta o estrella. Hay 8 modelos de elementos de carga que actualmente se han implementado, para este caso se utilizaron los siguientes: modelo 1 que se refiere a P y Q constantes, modelo 2 impedancia (Z) constante y modelo 5 corriente (I) constante. Se indican los parámetros de operación kV, kW y kvar.

```
New Load.671 Bus1=671.1.2.3   Phases=3 Conn=Delta Model=1 kV=4.16    kW=1155  kvar=660
New Load.634a Bus1=634.1      Phases=1 Conn=Wye   Model=1 kV=0.277   kW=160   kvar=110
New Load.634b Bus1=634.2      Phases=1 Conn=Wye   Model=1 kV=0.277   kW=120   kvar=90
New Load.634c Bus1=634.3      Phases=1 Conn=Wye   Model=1 kV=0.277   kW=120   kvar=90
New Load.645 Bus1=645.2       Phases=1 Conn=Wye   Model=1 kV=2.4     kW=170   kvar=125
New Load.646 Bus1=646.2.3     Phases=1 Conn=Delta Model=2 kV=4.16    kW=230   kvar=132
New Load.692 Bus1=692.3.1     Phases=1 Conn=Delta Model=5 kV=4.16    kW=170   kvar=151
New Load.675a Bus1=675.1      Phases=1 Conn=Wye   Model=1 kV=2.4     kW=485   kvar=190
New Load.675b Bus1=675.2      Phases=1 Conn=Wye   Model=1 kV=2.4     kW=68    kvar=60
New Load.675c Bus1=675.3      Phases=1 Conn=Wye   Model=1 kV=2.4     kW=290   kvar=212
New Load.611 Bus1=611.3       Phases=1 Conn=Wye   Model=5 kV=2.4     kW=170   kvar=80
New Load.652 Bus1=652.1       Phases=1 Conn=Wye   Model=2 kV=2.4     kW=128   kvar=86
New Load.670a Bus1=MID671.1   Phases=1 Conn=Wye   Model=1 kV=2.4     kW=17    kvar=10
New Load.670b Bus1=MID671.2   Phases=1 Conn=Wye   Model=1 kV=2.4     kW=66    kvar=38
New Load.670c Bus1=MID671.3   Phases=1 Conn=Wye   Model=1 kV=2.4     kW=117   kvar=68
```

Figura 26. Cargas distribuidas en la red.

Los capacitores se definen como “New Capacitor” seguido de este comando se le asigna un nombre a cada uno. Se especifica en que bus está conectado y si es monofásico o bifásico se especifica la fase correspondiente con los numero 1, 2 y 3. Por otro lado, si es trifásico solo se agrega el número de “Phase=3”. Esto se puede visualizar claramente en la Figura 27, en la cual se tiene dos capacitores, uno trifásico y uno monofásico interconectado a la fase C. A demás, se complementa con los parámetros kVAR y kV.

```
New Capacitor.Cap1 Bus1=675 phases=3 kVAR=600 kV=4.16
New Capacitor.Cap2 Bus1=611.3 phases=1 kVAR=100 kV=2.4

!Bus 670 is the concentrated point load of the distributed load on line 632 to 671 located at 1/3 the distance from node 632
```

Figura 27. Modelación de capacitores.

Para agregar las líneas de la red se utiliza el comando “New” definiendo que se trata de una línea “Line” y posteriormente asignándole un nombre “por ejemplo 650632”, se designan los buses entre los que está, se añade el LineCode (ver Figura 28) y finalmente se define la longitud de la línea.

```
New Line.650632    Phases=3 Bus1=RG60.1.2.3      Bus2=632.1.2.3       LineCode=601 Length=2    units=kft
New Line.632670    Phases=3 Bus1=632.1.2.3       Bus2=MID671.1.2.3    LineCode=601 Length=1    units=kft
New Line.670671    Phases=3 Bus1=MID671.1.2.3    Bus2=671.1.2.3       LineCode=601 Length=1    units=kft
New Line.671680    Phases=3 Bus1=671.1.2.3       Bus2=680.1.2.3       LineCode=601 Length=1    units=kft
New Line.632633    Phases=3 Bus1=632.1.2.3       Bus2=633.1.2.3       LineCode=602 Length=0.5  units=kft
New Line.632645    Phases=2 Bus1=632.2.3         Bus2=645.2.3         LineCode=603 Length=0.5  units=kft
New Line.645646    Phases=2 Bus1=645.2.3         Bus2=646.2.3         LineCode=603 Length=0.3  units=kft
New Line.692675    Phases=3 Bus1=692.1.2.3       Bus2=675.1.2.3       LineCode=606 Length=0.5  units=kft
New Line.671684    Phases=2 Bus1=671.1.3         Bus2=684.1.3         LineCode=604 Length=0.3  units=kft
New Line.684611    Phases=1 Bus1=684.3           Bus2=611.3           LineCode=605 Length=0.3  units=kft
New Line.684652    Phases=1 Bus1=684.1           Bus2=652.1           LineCode=607 Length=0.8  units=kft
```

Figura 28. Modelado de líneas que integran la red.

La red que se está analizando contiene un interruptor entre los buses 671 y 692, que porta la corriente que fluye en esa línea de la red. Se define el nuevo elemento como “New Line.671692”, se debe especificar de cuantas fases es “Phase=3”, los buses entre los que se encuentra “Bus1=671” y “Bus=692”. A demás se tiene el comando “Switch=y” que señala si el interruptor está cerrado o

abierto, en este caso se encuentra cerrado. Tiene similitud con la programación empleada para las líneas eléctricas, solo que para este elemento se trabaja en condiciones ideales con resistencias y reactancias pequeñas, y prácticamente sin capacitancias, esto se visualiza en las últimas líneas de código como r1, x1, c1 y c0.

```
New Line.671692    Phases=3 Bus1=671    Bus2=692  Switch=y  r1=1e-4 r0=1e-4 x1=0.000 x0=0.000 c1=0.000 c0=0.000
```

Figura 29. Interruptor entre buses 671 y 692.

La Figura 30 muestra el conjunto de fuentes de corrientes asociados con la generación distribuida que se integran a la red eléctrica. Para el modelado de estas fuentes de corriente es necesario especificar en qué fase están conectadas, si el bus de interconexión es trifásico se agregan tres fuentes de corrientes, si es bifásico se agregan dos especificando en que fases esta interconectado A(1), B(2) o C(3) dependiendo de las fases del bus, y para los monofásicos aplica el mismo criterio. Inicializándolas en cero su magnitud y ángulo, representando el caso base precargado en el software.

```
New Isource.IES_675a        Phases=1 Bus1=675.1        Amps=0 Angle=0
New Isource.IES_675b        Phases=1 Bus1=675.2        Amps=0 Angle=0
New Isource.IES_675c        Phases=1 Bus1=675.3        Amps=0 Angle=0
New Isource.IES_611c        Phases=1 Bus1=611.3        Amps=0 Angle=0
New Isource.IES_652a        Phases=1 Bus1=652.1        Amps=0 Angle=0
New Isource.IES_632_671a    Phases=1 Bus1=MID671.1     Amps=0 Angle=0
New Isource.IES_632_671b    Phases=1 Bus1=MID671.2     Amps=0 Angle=0
New Isource.IES_632_671c    Phases=1 Bus1=MID671.3     Amps=0 Angle=0
New Isource.IES_634a        Phases=1 Bus1=634.1        Amps=0 Angle=0
New Isource.IES_634b        Phases=1 Bus1=634.2        Amps=0 Angle=0
New Isource.IES_634c        Phases=1 Bus1=634.3        Amps=0 Angle=0
New Isource.IES_645b        Phases=1 Bus1=645.2        Amps=0 Angle=0
```

Figura 30. Integración e inicialización de corrientes.

Para que las cargas funcionen adecuadamente de acuerdo con su modelo especificado se agregan las líneas de código de la Figura 31. De esta manera las cargas soportan variaciones de voltaje de 0.7 a 1.15 pu sin modificaciones en su modelo. Si no se realiza este procedimiento las cargas operan en un rango de voltaje entre 0.95 y 1.05 pu y al violarse este rango se comportan como cargas de impedancia constante.

```
Load.671.vminpu=0.7  vmaxpu=1.15 daily=default
Load.634a.vminpu=0.7 vmaxpu=1.15 daily=default
Load.634b.vminpu=0.7 vmaxpu=1.15 daily=default
Load.634c.vminpu=0.7 vmaxpu=1.15 daily=default
Load.645.vminpu=0.7  vmaxpu=1.15 daily=default
Load.646.vminpu=0.7  vmaxpu=1.15 daily=default
Load.692.vminpu=0.7  vmaxpu=1.15 daily=default
Load.675a.vminpu=0.7 vmaxpu=1.15 daily=default
Load.675b.vminpu=0.7 vmaxpu=1.15 daily=default
Load.675c.vminpu=0.7 vmaxpu=1.15 daily=default
Load.611.vminpu=0.7  vmaxpu=1.15 daily=default
Load.652.vminpu=0.7  vmaxpu=1.15 daily=default
Load.670a.vminpu=0.7 vmaxpu=1.15 daily=default
Load.670b.vminpu=0.7 vmaxpu=1.15 daily=default
Load.670c.vminpu=0.7 vmaxpu=1.15 daily=default
```

Figura 31. Código de funcionamiento del Solve.

Para finalizar el código en OpenDss se tiene la serie de líneas de código que se muestran en la Figura 32. El comando “Set Voltagebases” sirve para definir los voltajes base en el circuito. En las líneas de código “Transformer” se ajustan los taps a los valores 10, 8 y 11. La función de estas líneas de código es que los taps de los reguladores queden fijos a los valores establecidos. Con el comando “Set Controlmode=OFF” se le indica a OpenDSS que no operara el control de voltaje y los taps deben mantenerse constantes.

```
Set Voltagebases=[115, 4.16, .48]
calcv

Transformer.Reg1.wdg=2 Tap=(0.00625    10 * 1 +)   ! Tap 10
Transformer.Reg2.wdg=2 Tap=(0.00625    8 * 1 +)    ! Tap 8
Transformer.Reg3.wdg=2 Tap=(0.00625    11 * 1 +)   ! Tap 11
Set Controlmode=OFF
```

Figura 32. Finalización del código OpenDSS

Referencias

Dugan, R. C., & Montenegro, D. (2022). *The Open Distribution System Simulator™ (OpenDSS)*. Electric Power Research Institute.