Unbalanced Distribution Load Flow using OpenDSS Matlab COM Interface

OpenDSS is an open distribution system simulation tool developed by the Electric Power Research Institute (EPRI). A user can simulate any distribution system using OpenDSS while utilizing COM interfacing (see OpenDSS manual for details).

Here, OpenDSS utilized for a distribution system load flow using Matlab COM Interfacing. The following steps are from installation to distribution load flow. Intermediate steps are about various definitions of components of the distribution system.

Steps from OpenDSS download to Daily Load flow:

(1) A very first step is to download OpenDSS from https://sourceforge.net/projects/electricdss/ and then install it properly on your computer (Remember to run your system as an "Administrator").

(2) After its successful installation, it's time to define Loads, Lines, Transformer, Voltage Regulators, Capacitors, loadshapes, generator (if any), etc. either in ".txt" or in ".dss" file.

(3) For example, in this case we are defining all above definitions for IEEE-13 Bus radial distribution feeder. All data and line configurations can be found at https://ewh.ieee.org/soc/pes/dsacom/testfeeders/. Complete data with all definition is also available in OpenDSS at C:\opendss\IEEETestCases\13Bus (where OpenDSS installed) with "IEEE13Nodeckt.dss" name.

(i) LOAD DEFINITION: Open a .txt or .dss file and enter the definition in following format.

New Load.671 Bus1=671.1.2.3 Phases=3 Conn=Delta Model=1
kV=4.16 ~kW=1155 kvar=660

Whenever defining a load, a user has to start with "New Load".

In this case "671" is load name, "671.1.2.3" is bus with all three nodes, "3" is no. of phases for load "671", "Delta" is the type of connection defined for load (given in data). Model "1" means constant PQ type of load. "4.16". Here Load kW and kVAr are "1155" and "660" respectively.

The symbol ~ is used for continuation. In all definitions (not limited to load definitions), whenever continuation is required, then just "Enter" and insert the symbol ~ and start writing other properties.

After defining all loads save this file as "Loads.txt". Fig.1 shows its snapshot.

Loads - Notepad

File Edit Format View Help

```
!LOAD DEFINITIONS FOR IEEE-13 BUS RADIAL DISTRIBUTION SYSTEM
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=670.1     Phases=1 Conn=Wye   Model=1 kV=2.4    kW=17    kvar=10
New Load.670b Bus1=670.2     Phases=1 Conn=Wye   Model=1 kV=2.4    kW=66    kvar=38
New Load.670c Bus1=670.3     Phases=1 Conn=Wye   Model=1 kV=2.4    kW=117   kvar=68
```

Fig.1: Loads.txt for IEEE-13 Bus Radial Distribution System

(ii) DEFINING LINES AND LINE CODE:

- Writing Line Code

New linecode.mtx601 nphases=3 BaseFreq=60

~ rmatrix = (0.3465 | 0.1560 0.3375 | 0.1580 0.1535 0.3414)

~ xmatrix = (1.0179 | 0.5017 1.0478 | 0.4236 0.3849 1.0348)

~ units=mi

Line code definitions start with New linecode, thereafter its name (here "mtx601" or it can be number/alphabete/alphanumeric as per user interest). Here rmatrix and xmatrix also defined along with its unit (here in "ohm/mile"). If we look at the data, then we saw that these two matrixes are symmetrical. Therefore, lower/upper half triangular matrix will represent all resistances (R) and reactances (X). In this definition, the lower half triangular matrix is given for "R" as well as for "X". Line code can also be generated using "Show Line constants" command if we define a line geometry. (For Line geometry refer C:\opendss\IEEETestCases\LineCodes\IEEELineGeometry.dss).

- <u>Create Line (Can be one, two, or three-phase lines)</u>

New Line.650632 Phases=3 Bus1=RG60.1.2.3 Bus2=632.1.2.3
~ LineCode=mtx601 Length=2000 units=ft

The line definition starts with New Line then write the line name (here line name is "650632", it can be any numeric, alphabetic, or numeric-alphabetic etc. as per user interest). "RG60.1.2.3" is ***From Bus*** and "632.1.2.3" is ***To Bus***. "mtx601" is line code which has already been defined in Line Code.

Lines - Notepad

File Edit Format View Help

```
New linecode.mtx601 nphases=3 BaseFreq=60
~ rmatrix = (0.3465 | 0.1560 0.3375 | 0.1580 0.1535 0.3414 )
~ xmatrix = (1.0179 | 0.5017 1.0478 | 0.4236 0.3849 1.0348 )
~ units=mi
New linecode.mtx602 nphases=3 BaseFreq=60
~ rmatrix = (0.7526 | 0.1580 0.7475 | 0.1560 0.1535 0.7436 )
~ xmatrix = (1.1814 | 0.4236 1.1983 | 0.5017 0.3849 1.2112 )
~ units=mi
New linecode.mtx603 nphases=2 BaseFreq=60
~ rmatrix = (1.3238 | 0.2066 1.3294 )
~ xmatrix = (1.3569 | 0.4591 1.3471 )
~ units=mi
New linecode.mtx604 nphases=2 BaseFreq=60
~ rmatrix = (1.3238 | 0.2066 1.3294 )
~ xmatrix = (1.3569 | 0.4591 1.3471 )
~ units=mi
New linecode.mtx605 nphases=1 BaseFreq=60
~ rmatrix = (1.3292 )
~ xmatrix = (1.3475 )
~ units=mi
New linecode.mtx606 nphases=3 BaseFreq=60
~ rmatrix = (0.7982 | 0.3192 0.7891 | 0.2849 0.3192 0.7982 )
~ xmatrix = (0.4463 | 0.0328 0.4041 | -0.0143 0.0328 0.4463 )
~ Cmatrix = [257 | 0 257 | 0 0 257]
~ units=mi
New linecode.mtx607 nphases=1 BaseFreq=60
~ rmatrix = (1.3425 )
~ xmatrix = (0.5124 )
~ cmatrix = [236]
~ units=mi

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

!LINE DEFINITIONS
New Line.650632    Phases=3 Bus1=RG60.1.2.3   Bus2=632.1.2.3  LineCode=mtx601 Length=2000 units=ft
New Line.632670    Phases=3 Bus1=632.1.2.3    Bus2=670.1.2.3  LineCode=mtx601 Length=667  units=ft
New Line.670671    Phases=3 Bus1=670.1.2.3    Bus2=671.1.2.3  LineCode=mtx601 Length=1333 units=ft
New Line.671680    Phases=3 Bus1=671.1.2.3    Bus2=680.1.2.3  LineCode=mtx601 Length=1000 units=ft
New Line.632633    Phases=3 Bus1=632.1.2.3    Bus2=633.1.2.3  LineCode=mtx602 Length=500  units=ft
New Line.632645    Phases=2 Bus1=632.3.2      Bus2=645.3.2    LineCode=mtx603 Length=500  units=ft
New Line.645646    Phases=2 Bus1=645.3.2      Bus2=646.3.2    LineCode=mtx603 Length=300  units=ft
New Line.692675    Phases=3 Bus1=692.1.2.3    Bus2=675.1.2.3  LineCode=mtx606 Length=500  units=ft
New Line.671684    Phases=2 Bus1=671.1.3      Bus2=684.1.3    LineCode=mtx604 Length=300  units=ft
New Line.684611    Phases=1 Bus1=684.3        Bus2=611.3      LineCode=mtx605 Length=300  units=ft
New Line.684652    Phases=1 Bus1=684.1        Bus2=652.1      LineCode=mtx607 Length=800  units=ft


!SWITCH DEFINITIONS
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
```

Fig. 2: Lines.txt for IEEE-13 Bus System

(iii) CAPACITOR DEFINITION

This definition is almost similar to "Load Definition". For example:

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

In IEEE-13 Bus data, two capacitors are defined at Bus 675 and Bus 611. On Bus 675 each phase having 200 kVAr capacitor. So we can combine them and define for 3-phase. The user may adopt a similar definition as defined for "Cap2" if separate phase definition is required. But remember whenever Capacitor or Load is defined at "single phase" then kV is phase voltage not Line voltage. Fig. 3 shows a snapshot of capacitor definitions.

```
!CAPACITOR DEFINITIONS
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
```

Fig. 3: Capacitors.txt file

This is well represented in "Cap2" definitions, where the capacitor is only connected at phase "c"(means Bus1 shall be defined as "611.3", it means "Cap2" is connected at node-3 of 611). It means single phase, then kV is 2.4 (=4.16/√3).

(iii) TRANSFORMER AND VOLTAGE REGULATOR DEFINITION

Definition of transformer and/or voltage regulator starts with New Transformer.

```
New Transformer.Sub Phases=3 Windings=2 XHL=(8 1000 /) wdg=1
~ buses=(SourceBus,650) conns=(delta,wye) kvs=(115,4.16)
 ~ kvas=(5000,5000)   %r=(.0005,0.0005)  XHT=4    XLT=4
```

"Sub" is transformer name. As this is the substation transformer definition, which is connected between "SourceBus" and "650". This is defined under buses property. Here connections of primary and secondary sides are defined under conns property. For this particular transformer, connections are "delta" at 115 kV (primary) side, and "wye" at 4.16 kV (secondary) side. "kVs" and "kVAs" of primary and secondary can be defined using kvs and kvas properties, respectively. % winding resistance and reactances are defined under %r, XHT, XLT properties.

```
!SUB TRANSFORMER DEFINITION
! Although this data was given, it does not appear to be used in the test case results
! The published test case starts at 1.0 per unit at Bus 650. To make this happen, we will change the impedance
! on the transformer to something tiny by dividing by 1000 using the DSS in-line RPN math

New Transformer.Sub Phases=3 Windings=2   XHL=8
~ buses=(SourceBus,650)   conns=(delta, wye)  kvs=(115,4.16)  kvas=(5000,5000)   r = (0.0005, 0.0005)  XHT=4 XLT=4


! FEEDER 1-PHASE VOLTAGE REGULATORS
! Define low-impedance 2-wdg transformer

New Transformer.Reg1 phases=1 XHL=0.01 kVAs=[1666 1666]
~ Buses=[650.1 RG60.1] kVs=[2.4  2.4] %LoadLoss=0.001
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=0.001
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=0.001
new regcontrol.Reg3  transformer=Reg3 winding=2  vreg=122  band=2  ptratio=20 ctprim=700  R=3    X=9


!TRANSFORMER DEFINITION
New Transformer.XFM1  Phases=3   Windings=2  XHL=2 buses=(633, 634) conns=(wye,wye) kvs=(4.16, 0.480) kvas=(500,500)
```

Fig. 4: Transformers.txt where transformers and voltage regulator definitions are stored

(4) After all definitions (as given in (3)), master file has to be defined where all associated files are redirected using command "Redirect". A snapshot of Master file is given in Fig.5.

```
clear
new circuit.MasterIEEE13 ! Name of file MasterIEEE13
~ basekv=115 pu=1.0 phases=3 bus1=SourceBus
~ Angle=30  MVAsc3=2000000 MVASC1=2100000 ! to make an infinite source
Redirect loadshapes.DSS
Redirect Transformers.txt
Redirect Lines.txt
Redirect Loads.txt
Redirect Capacitors.txt

Set Voltagebases=[115, 4.16, .48] ! Three Voltage levels

solve
Calcvoltagebases
```

Fig.5: MasterIEEE33.DSS snapshot where all definitions are redirected

(5) After defining loads, lines, capacitors, transformer, and master file, thereafter write some code in Matlab to perform load flow. Run loadflow for snap mode (Just Run SnapLoadFlow.m)

Matlab code for snap mode load flow.

```
clear all;
clc;
DSSObj = actxserver('OpenDSSEngine.DSS');
if ~DSSObj.Start(0),
disp('Unable to start the OpenDSS Engine');
return
end
DSSText = DSSObj.Text; % Used for all text interfacing from matlab to opendss
DSSCircuit = DSSObj.ActiveCircuit; % active circuit
% Write Path where Master and its associated files are stored and compile as
per following command
DSSText.Command='Compile
(C:\Users\Admin\Desktop\SaturdayExample\ToUpload\MasterIEEE13.dss)';
DSSText.Command='batchedit load..* Vmin=0.8'; % Set Vmin lower so that load
model property will remain same
DSSTransformers=DSSCircuit.Transformers;
 %% SnapShot Mode (To Run for Peak Load Only)
DSSText.Command='set mode=snap';
DSSText.Command='solve';

DSSText.Command='Show current';
DSSText.Command='Show losses';
DSSText.Command='Show Taps';
DSSText.Command='Show Voltage';
SystemLosses=(DSSCircuit.Losses)/1000; % Will Give you Distribution System
Losses in kWs and kVArs
%% Line Losses
LineLosses=DSSCircuit.Linelosses;
%% Transformer Losses

TranLosses=SystemLosses-LineLosses; %Will give us losses of all transformer
in distribution network (if no storage)


%% Voltage Magnitude for each phase in p.u. can be obtained in this way
V1pu=DSSCircuit.AllNodeVmagPUByPhase(1); % For A-phase
V2pu=DSSCircuit.AllNodeVmagPUByPhase(2); % For B-phase
V3pu=DSSCircuit.AllNodeVmagPUByPhase(3); % For C-phase
```

(6) Run Load flow in daily mode (Just run file DailyLoadFlow.m), where loads are changing in 24-hour as per loadshape.

This code is given below.

```
clear all;
clc;
DSSObj = actxserver('OpenDSSEngine.DSS');
if ~DSSObj.Start(0),
disp('Unable to start the OpenDSS Engine');
return
end
DSSText = DSSObj.Text; % Used for all text interfacing from matlab to opendss
DSSCircuit = DSSObj.ActiveCircuit; % active circuit
DSSText.Command='Compile
(C:\Users\Admin\Desktop\SaturdayExample\ToUpload\MasterIEEE13.dss)';% Path
where Master and its associated files are stored.
DSSText.Command='batchedit load..* Vmin=0.8'; % Set Vmin so that load model
property will remain same
DSSTransformers=DSSCircuit.Transformers;
%DSSText.Command='Batchedit regcontrol..* Enabled=no'; % uncomment for tap
change as per user's choice
DSSText.Command='batchedit load..* daily=PQmult'; % Loadshape
DSSText.Command='New EnergyMeter.Main Line.650632 1';% Energy meter
nt=24*1;
TimeArray=1:nt;
%% Uncomment following for Tap change as per user choice (manually)
% Xtap=[15  7   6   6   6   7   8   9   11  12  13  14  14  14  14  14  14
14  14  14  14  14  13  12
% 10    5   4   4   4   4   5   6   7   8   8   9   9   9   9   9   9   9   9
9   9   9   9   9
% 15    6   5   5   5   6   7   9   10  12  13  13  14  14  14  14  14  14
14  14  14  14  13  12 ];
% Reg1Tap=Xtap(1,:)-5;
% Reg2Tap=Xtap(2,:)-5;
% Reg3Tap=Xtap(3,:)-5;
% Vreg1=1+0.00625*Reg1Tap;
% Vreg2=1+0.00625*Reg2Tap;
% Vreg3=1+0.00625*Reg3Tap;
DSSText.Command='set mode=daily stepsize=1h number=1';
DSSText.Command='set hour=0'; % Start at second 0 of hour 5
for i=1:nt
    DSSText.Command='get hour';
    hour=DSSText.Result;
    %% Uncomment following for change in Tap Positions of Regulator as per
user choice
% DSSText.command = ['Transformer.Reg1.Tap=',num2str(Vreg1(i))];
% DSSText.command = ['Transformer.Reg2.Tap=',num2str(Vreg2(i))];
% DSSText.command = ['Transformer.Reg3.Tap=',num2str(Vreg3(i))];
DSSText.Command='Solve';
SystemLosses(i,:)=(DSSCircuit.Losses)/1000; % Will Give you Distribution
System Losses in kWs and kVArs
%% Line Losses
LineLosses(i,:)=DSSCircuit.Linelosses;
%% Transformer Losses

TranLosses(i,:)=SystemLosses(i,:)-LineLosses(i,:);

%% Voltage Magnitude in p.u. for 24-hours can be obtained in this way
V1pu(i,:)=DSSCircuit.AllNodeVmagPUByPhase(1);
```

```
V2pu(i,:)=DSSCircuit.AllNodeVmagPUByPhase(2);
V3pu(i,:)=DSSCircuit.AllNodeVmagPUByPhase(3);
DSSText.Command = '? Transformer.Reg1.Taps';
Reg1=str2num(DSSText.Result);
Vreg1S(i,:)=Reg1(2);% Secondary winding voltage of Reg1 in 24-hr
DSSText.Command = '? Transformer.Reg2.Taps';
Reg2=str2num(DSSText.Result);
Vreg2S(i,:)=Reg2(2);% Secondary winding voltage of Reg2 in 24-hr
DSSText.Command = '? Transformer.Reg3.Taps';
Reg3=str2num(DSSText.Result);
Vreg3S(i,:)=Reg3(2);% Secondary winding voltage of Reg3 in 24-hr
DSSText.Command='Export Meter'; % A MasterIEEE13_EXP_METERS.CSV file will be
saved in same path
end
EM=csvread('MasterIEEE13_EXP_METERS.CSV',1,4);
SubkWh=EM(:,1);
SubkVArh=EM(:,2);
SubkW24=[SubkWh(TimeArray(1)); SubkWh(TimeArray(2):TimeArray(end))-
SubkWh(TimeArray(1):TimeArray(nt-1))];
SubkVAr24=[SubkVArh(TimeArray(1)); SubkVArh(TimeArray(2):TimeArray(end))-
SubkVArh(TimeArray(1):TimeArray(nt-1))];
SubkVA24=abs(SubkW24+sqrt(-1)*SubkVAr24);
SubstationkWkVArandkVA24=[SubkW24 SubkVAr24 SubkVA24];
delete('C:\Users\Admin\Desktop\SaturdayExample\ToUpload\MasterIEEE13_EXP_METE
RS.CSV')
% Here SubkW24 SubkVAr24 and SubkVA24 are substation kWs, kVArs, and kVAs for
24 hours period
clearvars -except SystemLosses LineLosses SubkW24 SubkVAr24 SubkVA24 V1pu
V2pu V3pu Vreg1S Vreg2S Vreg3S
```

Note: Save Lines.txt, Loads.txt,Transformers.txt, Master.txt in same folder. Otherwise, give path in master file while redirecting Lines.txt, Loads.txt, etc. In Matlab file, e.g. in SnapLoadFlow.m or in DailyLoadFlow.m, set the correct path e.g. in my case the path is C:\Users\Admin\Desktop\SaturdayExample\ToUpload\MasterIEEE13.dss