DETECTION OF FALSE DATA INJECTION ATTACKS IN SMART GRIDS

A PROJECT REPORT

SUBMITTED BY

Arushi Singh(2K19/IT/032)

Ananya Komal Singh(2K19/IT/017)

SUBMITTED TO

Mrs. Swati Sharda

Guest Lecturer

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our teacher Prof.Swati Sharda who gave us the golden opportunity to do this wonderful project on the topic:- Detection of False Data Injection in Smart Grids
This project helped us in understanding smart grids and IEEE bus systems better and we learnt about many new things. We would also like to thank our university, Delhi Technological University for giving us this opportunity to explore and research. We would also like to thank our peers and teacher for making this subject interesting and fun to learn.

Thank You

Arushi Singh(2K19/IT/032)
Ananya Komal Singh(2K19/IT/017)

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **"DETECTION OF FALSE DATA INJECTION ATTACKS IN SMART GRIDS"** which is submitted by Arushi Singh(2K19/IT/032) and Ananya Komal Singh(2K19/IT/017); INFORMATION TECHNOLOGY, Delhi Technological University, Delhi , is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: 1-12-2020

Swati Sharda
SUPERVISOR

ABSTRACT

The smart grid combines the classical power system with information technology, leading to a cyber-physical system. In such an environment the malicious injection of data has the potential to cause severe consequences. Classical residual based methods for bad data detection are unable to detect well designed false data injection (FDI) attacks. Moreover, most work on FDI attack detection is based on the linearized DC model of the power system and fails to detect attacks based on the AC model. The aim of this project is to address these problems by using the graph structure of the grid and the AC power flow model. We derive an attack injection method to inject attacks in our smart grids along with an attack detection method that is able to detect previously undetectable FDI attacks. By comparing the changes in the load flow analysis at two consecutive periods of time, we can detect whether our grid is attacked or not. Case studies on the different IEEE bus systems demonstrate that the proposed method is able to detect a wide range of previously undetectable attacks, on magnitudes of the voltages.

Table of Contents

CHAPTER 1
INTRODUCTION

INTRODUCTION TO SMART GRIDS

Maybe you have heard of the Smart Grid on the news or from your energy provider. But not everyone knows what the grid is, let alone the Smart Grid. "The grid," refers to the electric grid, a network of transmission lines, substations, transformers and more that deliver electricity from the power plant to your home or business. It's what you plug into when you flip on your light switch or power up your computer. Our current electric grid was built in the 1890s and improved upon as technology advanced through each decade. Today, it consists of more than 9,200 electric generating units with more than 1 million megawatts of generating capacity connected to more than 300,000 miles of transmission lines. Although the electric grid is considered an engineering marvel, we are stretching its patchwork nature to its capacity. To move forward, we need a new kind of electric grid, one that is built from the bottom up to handle the groundswell of digital and computerized equipment and technology dependent on it—and one that can automate and manage the increasing complexity and needs of electricity in the 21st Century.

What Makes a Grid "Smart?"

In short, the digital technology that allows for two-way communication between the utility and its customers, and the sensing along the transmission lines is what makes the grid smart. Like the Internet, the Smart Grid will consist of controls, computers, automation, and new technologies and equipment working together, but in this case, these technologies will work with the electrical grid to respond digitally to our quickly changing electric demand.

What does a Smart Grid do?

The Smart Grid represents an unprecedented opportunity to move the energy industry into a new era of reliability, availability, and efficiency that will contribute to our economic and environmental health. During the transition period, it will be critical to carry out testing, technology improvements, consumer education, development of standards and regulations, and information sharing between projects to ensure that the benefits we envision from the Smart Grid become a reality. The benefits associated with the Smart Grid include:

- More efficient transmission of electricity
- Quicker restoration of electricity after power disturbances
- Reduced operations and management costs for utilities, and ultimately lower power costs for consumers
- Reduced peak demand, which will also help lower electricity rates
- Increased integration of large-scale renewable energy systems
- Better integration of customer-owner power generation systems, including renewable energy systems
- Improved security

Today, an electricity disruption such as a blackout can have a domino effect—a series of failures that can affect banking, communications, traffic, and security. This is a particular threat in the winter, when homeowners can be left without heat. A smarter grid will add resiliency to our electric power System and make it better prepared to address emergencies such as severe storms, earthquakes, large solar flares, and terrorist attacks. Because of its two-way interactive capacity, the Smart Grid will allow for automatic rerouting when equipment fails or outages occur. This will minimize outages and minimize the effects when they do happen. When a power outage occurs, Smart Grid technologies will detect and isolate the outages, containing them before they become large-scale blackouts. The new technologies will also help ensure that electricity recovery resumes quickly and strategically after an emergency—routing electricity to emergency services first, for example. In addition, the Smart Grid will take greater advantage of customer-owned power generators to produce power when it is not available from utilities. By combining these "distributed generation" resources, a community could keep its health center, police department, traffic lights, phone System, and grocery store operating during emergencies. In addition, the Smart Grid is a way to address an aging energy infrastructure that needs to be upgraded or replaced. It’s a way to address energy efficiency, to bring increased awareness to consumers about the connection between electricity use and the environment. And it’s a way to bring increased national security to our energy System—drawing on greater amounts of home-grown electricity that is more resistant to natural disasters and attack.

Giving Consumers Control

The Smart Grid is not just about utilities and technologies; it is about giving you the information and tools you need to make choices about your energy use. If you already manage activities

such as personal banking from your home computer, imagine managing your electricity in a similar way. A smarter grid will enable an unprecedented level of consumer participation. For example, you will no longer have to wait for your monthly statement to know how much electricity you use. With a smarter grid, you can have a clear and timely picture of it. "Smart meters," and other mechanisms, will allow you to see how much electricity you use, when you use it, and its cost. Combined with real-time pricing, this will allow you to save money by using less power when electricity is most expensive. While the potential benefits of the Smart Grid are usually discussed in terms of economics, national security, and renewable energy goals, the Smart Grid has the potential to help you save money by helping you to manage your electricity use and choose the best times to purchase electricity. And you can save even more by generating your own power.

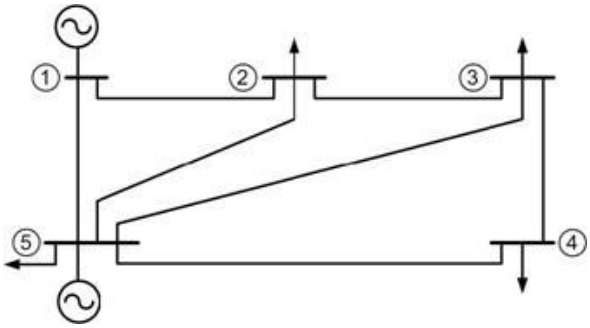
Building and Testing the Smart Grid

The Smart Grid will consist of millions of pieces and parts—controls, computers, power lines, and new technologies and equipment. It will take some time for all the technologies to be perfected, equipment installed, and systems tested before it comes fully online. And it won’t happen all at once—the Smart Grid is evolving, piece by piece, over the next decade or so. Once mature, the Smart Grid will likely bring the same kind of transformation that the Internet has already brought to the way we live, work, play, and learn.

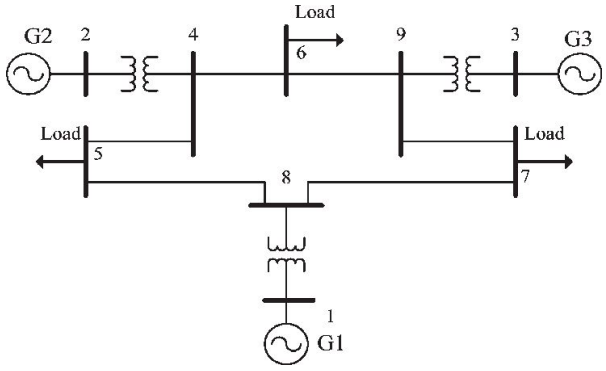
INTRODUCTION TO IEEE BUS SYSTEM

The IEEE standard bus systems represent real systems, and their data are neatly listed. In the distribution system, you can identify the buses based on load measurements at the primary level. Each load point or bus is then at the location of the primary side of the step down distribution transformer.

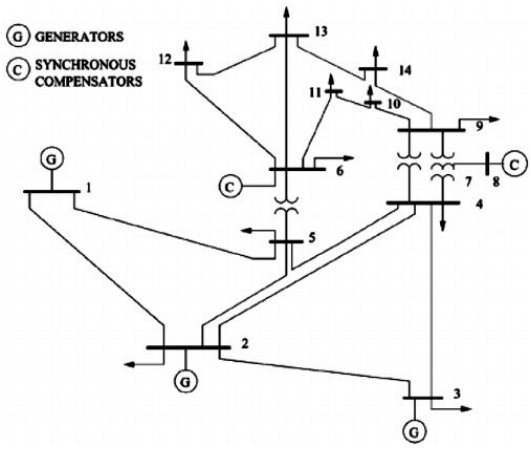
Types of IEEE Bus Systems



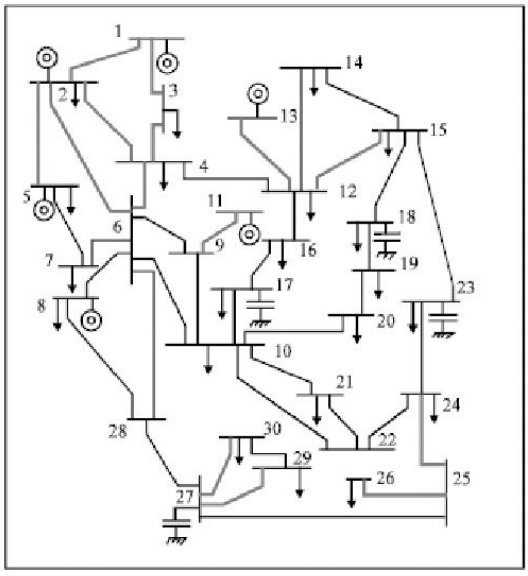
IEEE 5-BUS SYSTEM



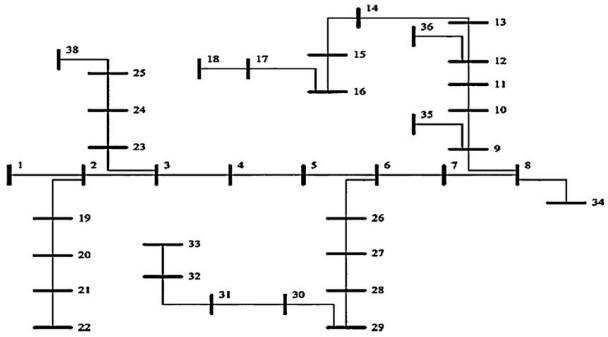
IEEE 9-BUS SYSTEM



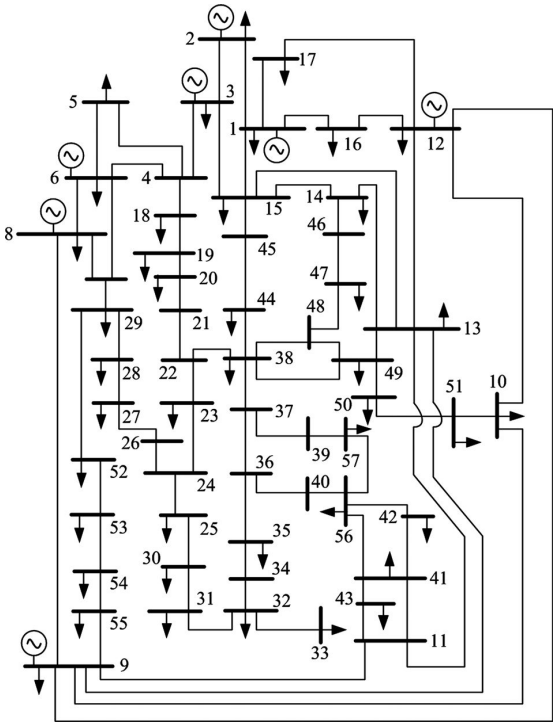
IEEE 14-BUS SYSTEM



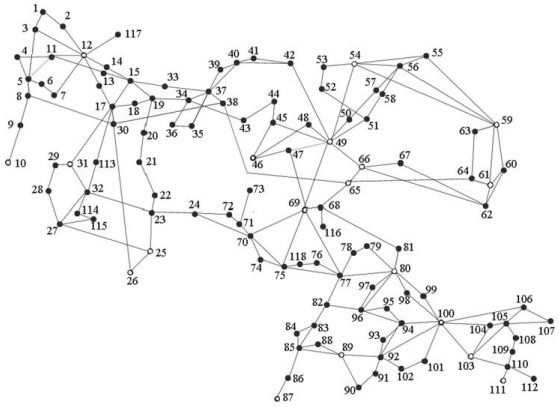
IEEE 30-BUS SYSTEM



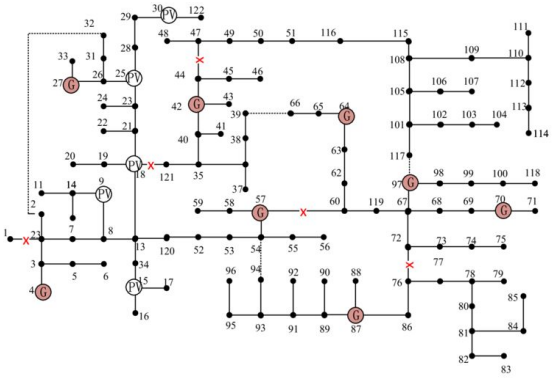
IEEE 37-BUS SYSTEM



IEEE 57-BUS SYSTEM



IEEE 118-BUS SYSTEM



IEEE 123-BUS SYSTEM

CHAPTER 2
PROJECT DESCRIPTION

The smart grid combines the classical power system with information technology, leading to a cyber-physical system. In such an environment the malicious injection of data has the potential to cause severe consequences. Classical residual based methods for bad data detection are unable to detect well designed false data injection (FDI) attacks. Moreover, most work on FDI attack detection is based on the linearized DC model of the power system and fails to detect attacks based on the AC model. The aim of this project is to address these problems by using the graph structure of the grid and the AC power flow model. Initially using a standard data set of different IEEE bus systems stored in our MATLAB code, we performed Newton Raphson Load Flow Analysis. We derive an attack injection method to inject attacks in our smart grids along with an attack detection method that is able to detect previously undetectable FDI attacks. By comparing the changes in the load flow analysis at two consecutive periods of time, we can detect whether our grid is attacked or not. Case studies on the different IEEE bus systems demonstrate that the proposed method is able to detect a wide range of previously undetectable attacks, on magnitudes of the voltages.

Newton Raphson Load Flow Analysis

The Newton-Raphson method (also known as Newton's method) is a way to quickly find a good approximation for the root of a real-valued function *f*(*x*)=0. It uses the idea that a continuous and differentiable function can be approximated by a straight line tangent to it.

How it works?

Suppose you need to find the root of a continuous, differentiable function *f*(*x*), and you know the root you are looking for $x = x_0.$ is near the point . Then Newton's method tells us that a better approximation for the root is

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

This process may be repeated as many times as necessary to get the desired accuracy. In general, for any x-value x_n, the next value is given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Note: the term "near" is used loosely because it does not need a precise definition in this context. However, should be closer to the root you need than to any other root (if the function has multiple roots).

Steps of Execution

- Conduct Load flow analysis on the chosen Power System by the user on the Predefined datasets and output the line flow and losses using Newton- Raphson method of analysis. Consequently form the Graph Topology for the particular Bus system and store the obtained data on the respective nodes and edges.
- Perform Attack on any of the Nodes/Buses and do a comparative study for the change in respective voltages on the adjacent nodes by again performing the load flow analysis after the attack.
- Devise an algorithm to detect the node in which the false data has been injected.

Future Aspects

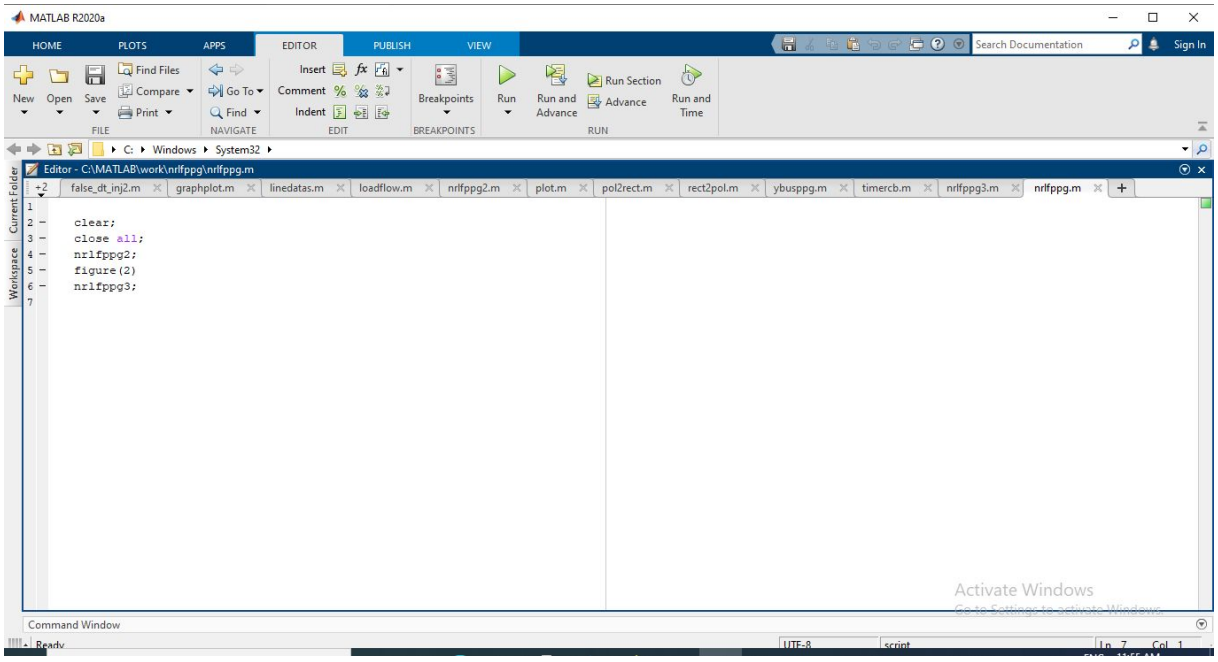
The normal operation of a power system relies on accurate state estimation that faithfully reflects the physical aspects of the electrical power grids. However, recent research shows that carefully synthesized false-data injection attacks can bypass the security system and introduce arbitrary errors to state estimates. In the future, we are planning to use graphical methods to study defending mechanisms against false-data injection attacks on power system state estimation. By securing carefully selected meter measurements, no false data injection attack can be launched to compromise any set of state variables. We characterize the optimal protection problem, which protects the state variables with minimum number of measurements, as a variant Steiner tree problem in a graph. Based on the graphical characterization, we propose both exact and reduced-complexity approximation algorithms. In particular, we show that the proposed tree-pruning based approximation algorithm significantly reduces computational complexity, while yielding negligible performance degradation compared with the optimal algorithms. The advantageous performance of the proposed defending mechanisms is verified in IEEE standard power system test cases.

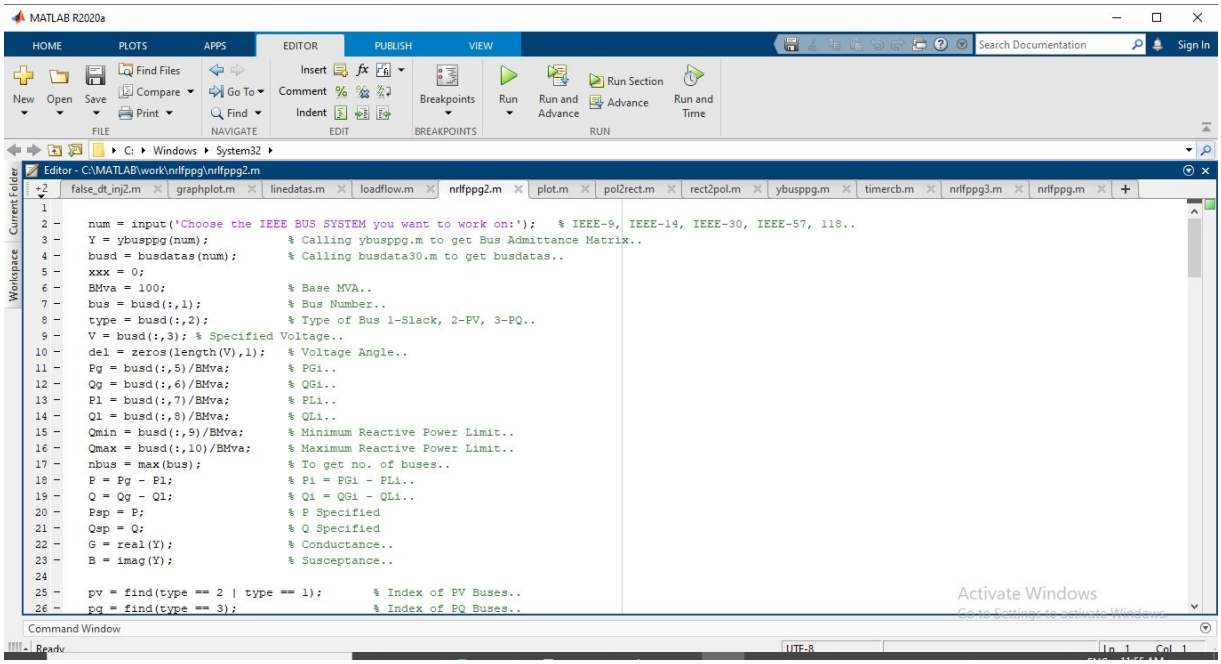
CHAPTER 3
CODE

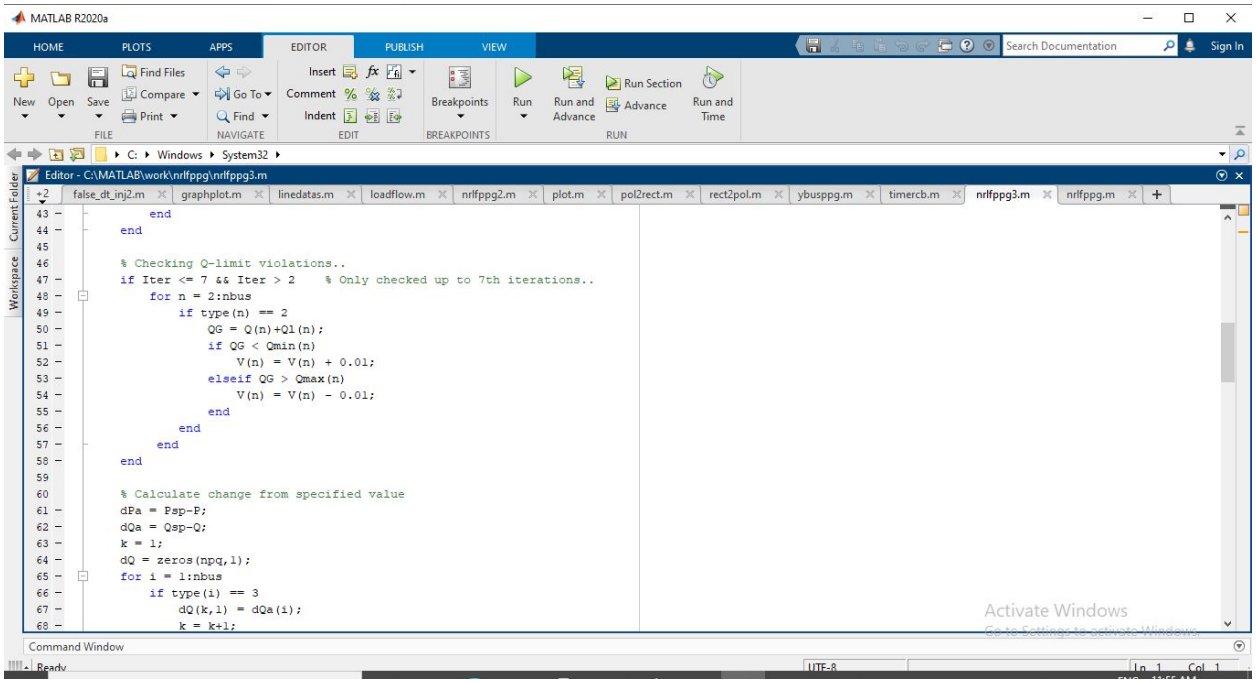
GITHUB LINK

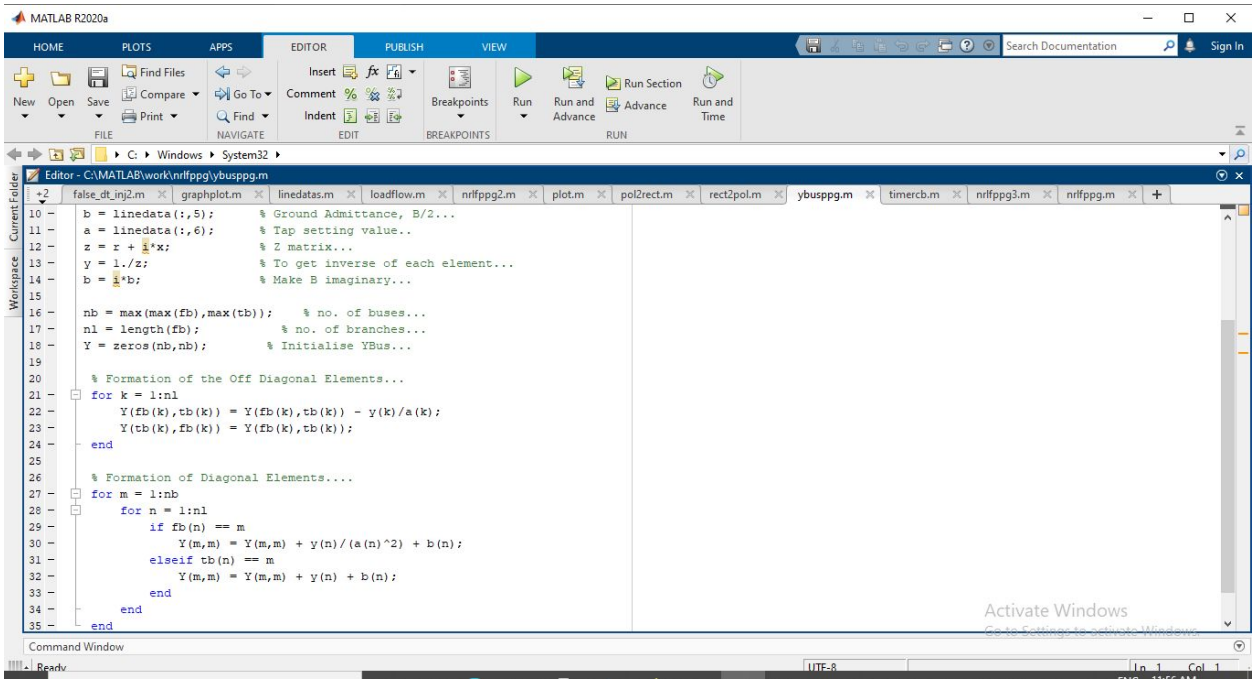
[arushisingh10/FDI (github.com)]

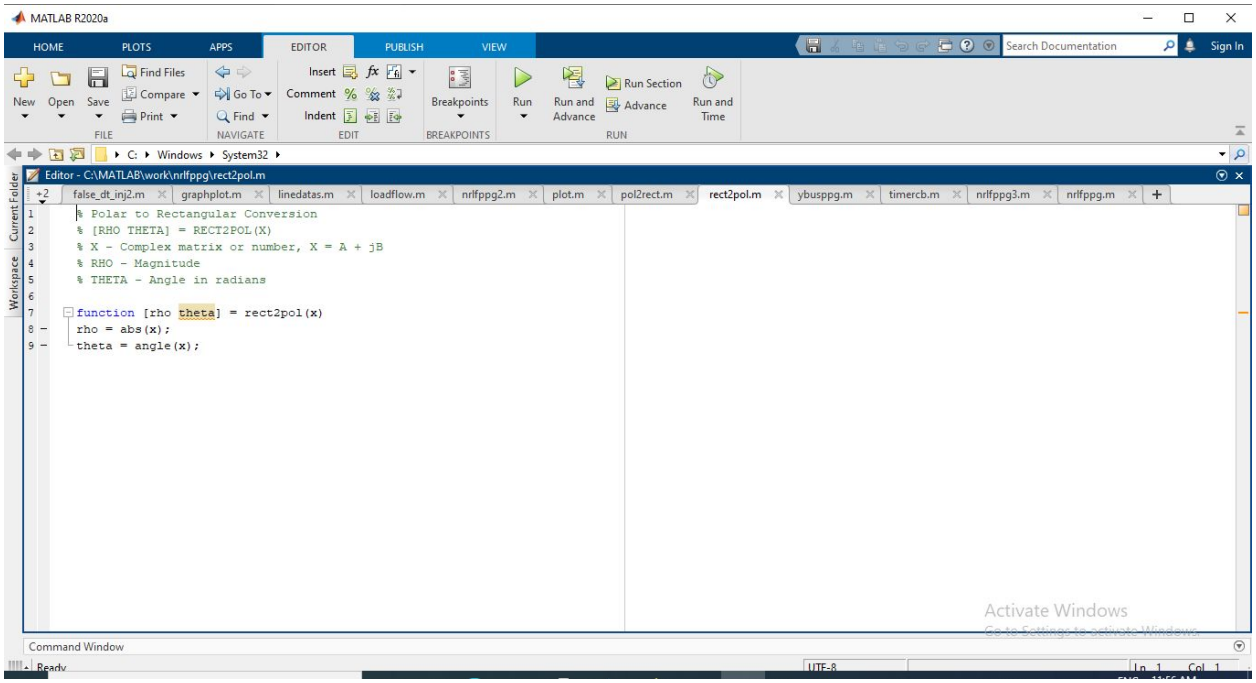
CHAPTER 4
SCREENSHOTS

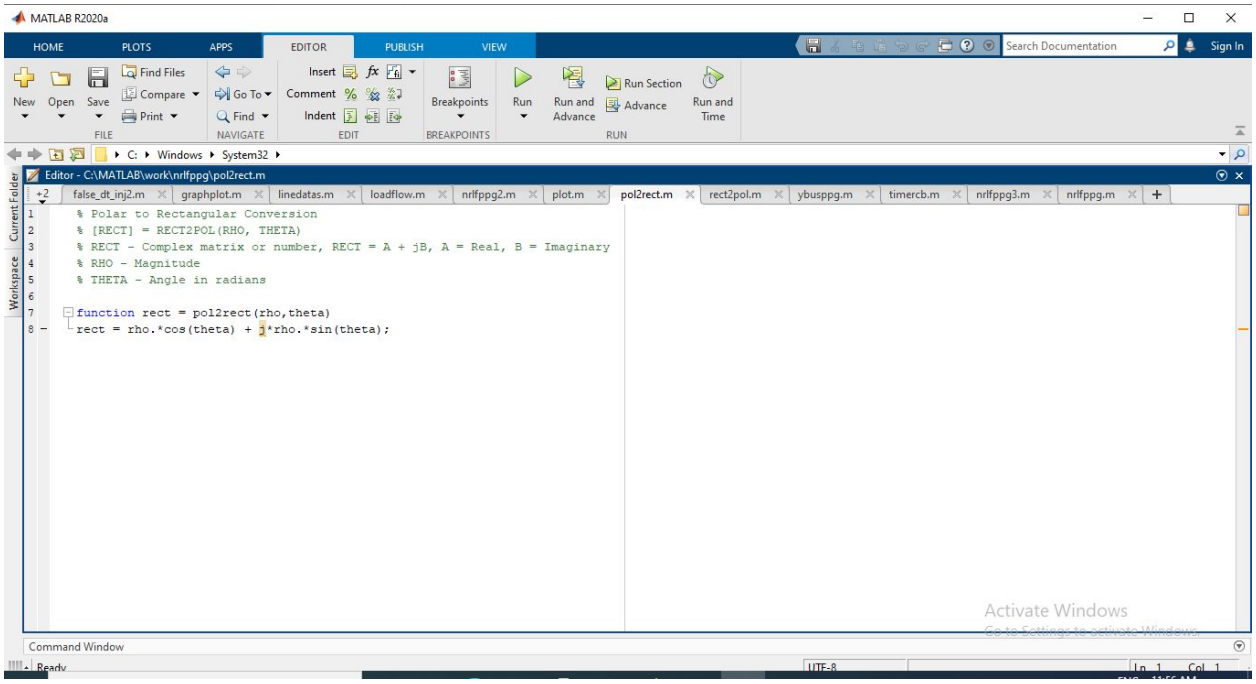


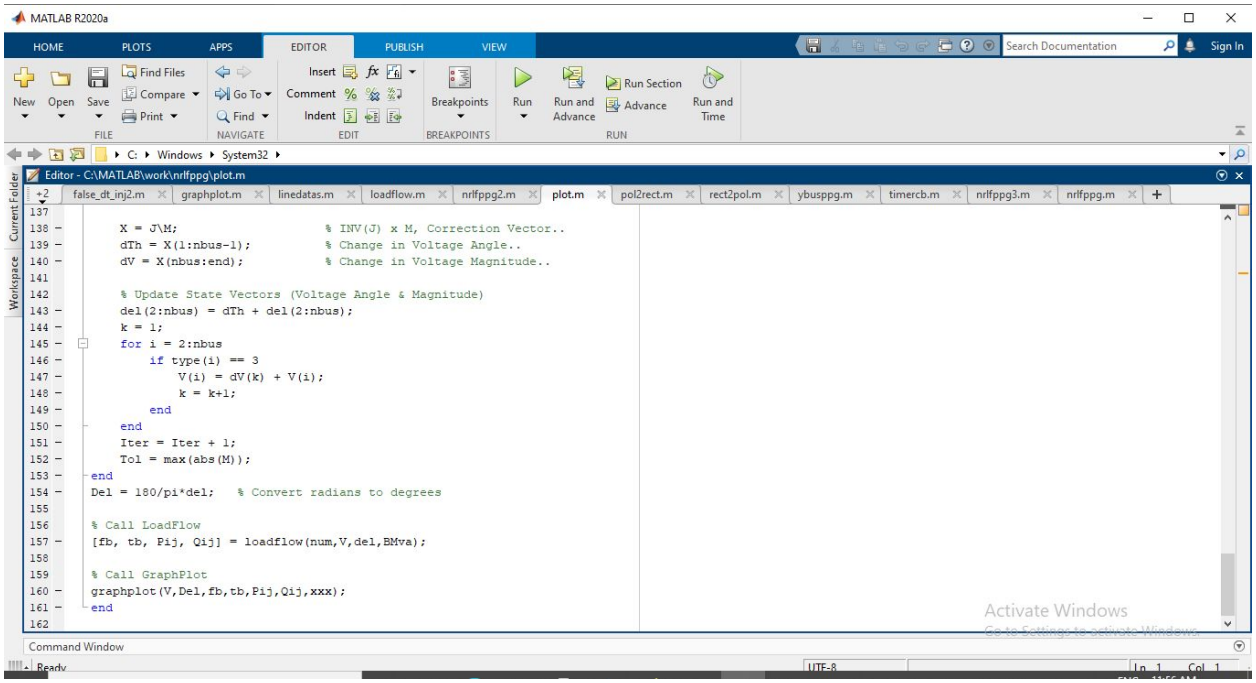


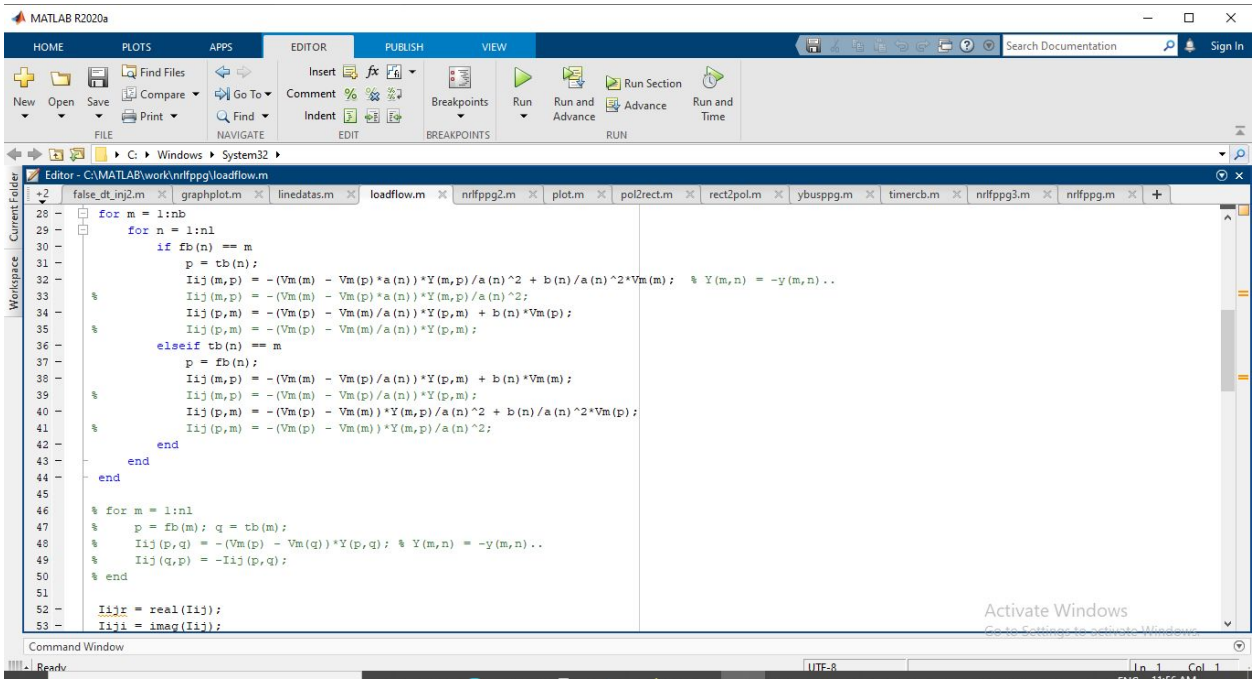

MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\nrlfppg3.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
Current Folder
Workspace
end
end
% Checking Q-limit violations..
if Iter <= 7 && Iter > 2 % Only checked up to 7th iterations..
for n = 2:nbus
if type(n) == 2
QG = Q(n)+Ql(n);
if QG < Qmin(n)
V(n) = V(n) + 0.01;
elseif QG > Qmax(n)
V(n) = V(n) - 0.01;
end
end
end
end
% Calculate change from specified value
dPa = Psp-P;
dQa = Qsp-Q;
k = 1;
dQ = zeros(npq,1);
for i = 1:nbus
if type(i) == 3
dQ(k,1) = dQa(i);
k = k+1;
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1

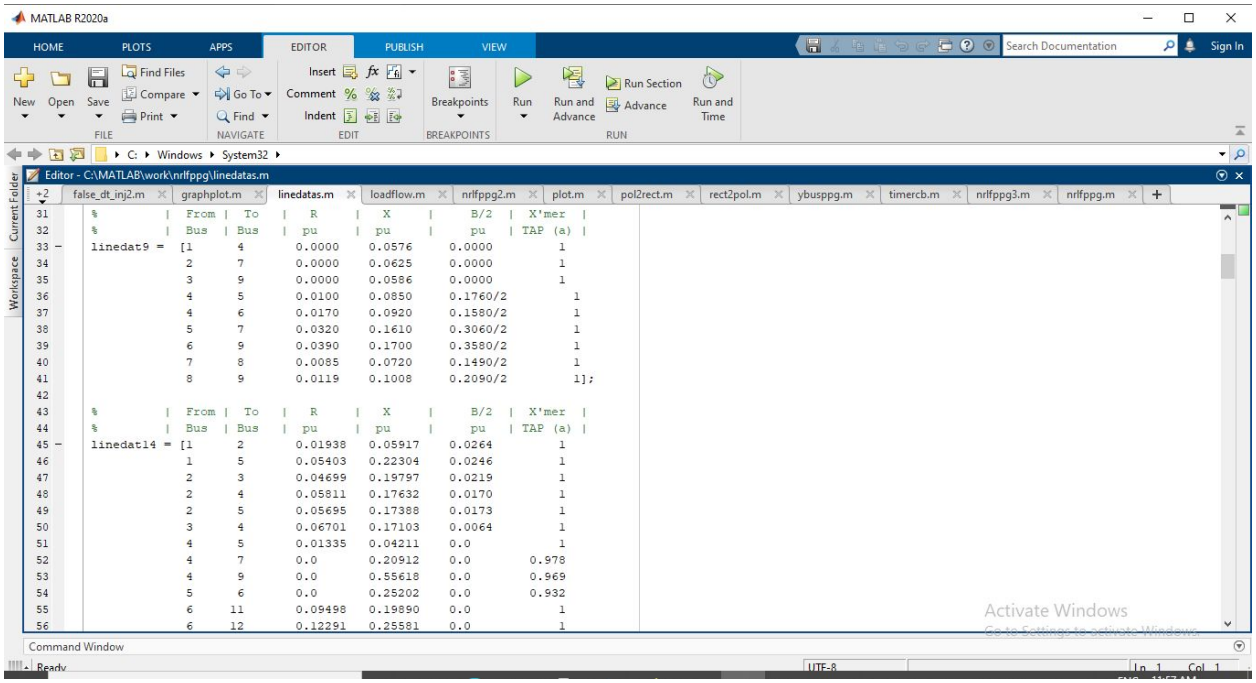

MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\ybusppg.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
Current Folder
Workspace
b = linedata(:,5); % Ground Admittance, B/2...
a = linedata(:,6); % Tap setting value..
z = r + i*x; % Z matrix...
y = 1./z; % To get inverse of each element...
b = i*b; % Make B imaginary...
nb = max(max(fb),max(tb)); % no. of buses...
nl = length(fb); % no. of branches...
Y = zeros(nb,nb); % Initialise YBus...
% Formation of the Off Diagonal Elements...
for k = 1:nl
Y(fb(k),tb(k)) = Y(fb(k),tb(k)) - y(k)/a(k);
Y(tb(k),fb(k)) = Y(fb(k),tb(k));
end
% Formation of Diagonal Elements....
for m = 1:nb
for n = 1:nl
if fb(n) == m
Y(m,m) = Y(m,m) + y(n)/(a(n)^2) + b(n);
elseif tb(n) == m
Y(m,m) = Y(m,m) + y(n) + b(n);
end
end
end
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1

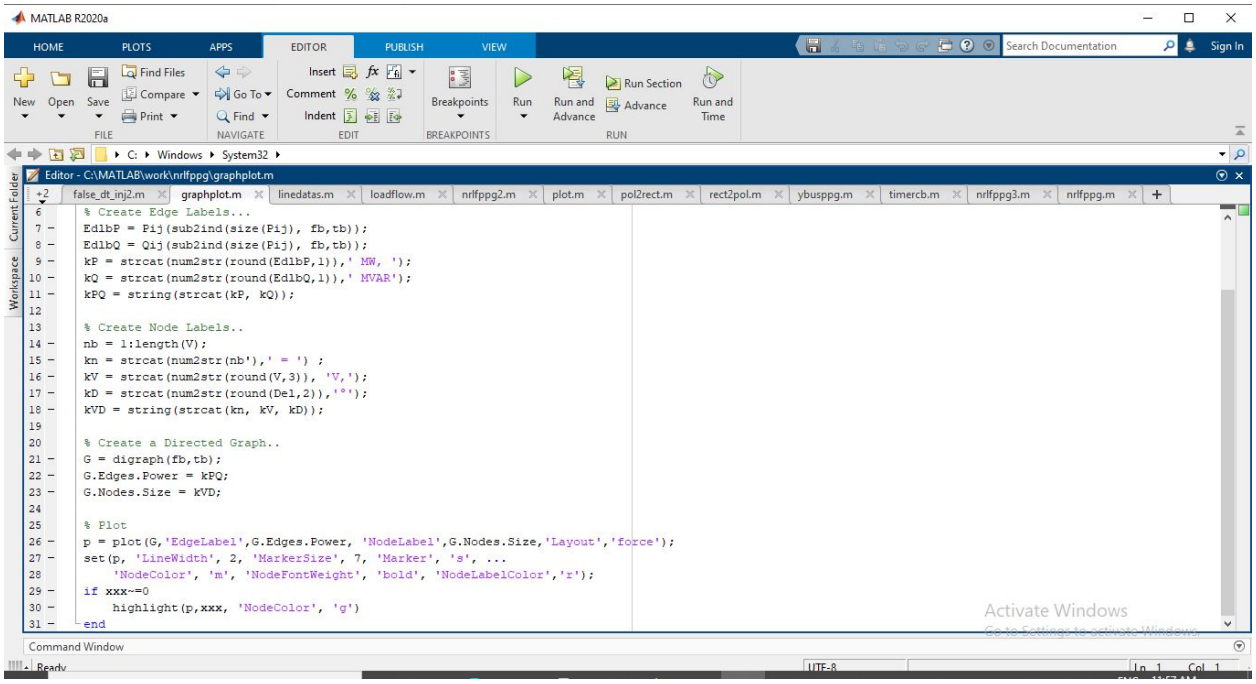

MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\rect2pol.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
% Polar to Rectangular Conversion
% [RHO THETA] = RECT2POL(X)
% X - Complex matrix or number, X = A + jB
% RHO - Magnitude
% THETA - Angle in radians
function [rho theta] = rect2pol(x)
rho = abs(x);
theta = angle(x);
Activate Windows
Command Window
Ready
UTF-8

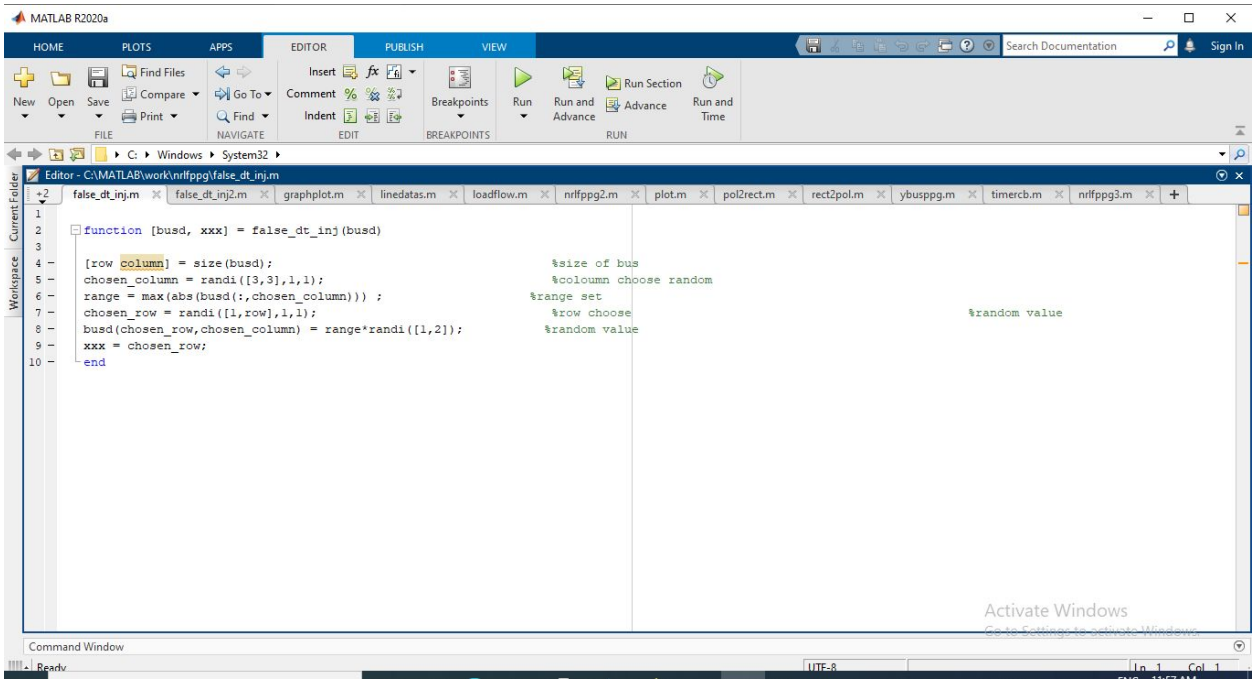

MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\pol2rect.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
% Polar to Rectangular Conversion
% [RECT] = RECT2POL(RHO, THETA)
% RECT - Complex matrix or number, RECT = A + jB, A = Real, B = Imaginary
% RHO - Magnitude
% THETA - Angle in radians
function rect = pol2rect(rho,theta)
rect = rho.*cos(theta) + j*rho.*sin(theta);
Activate Windows
Command Window
Ready
UTF-8

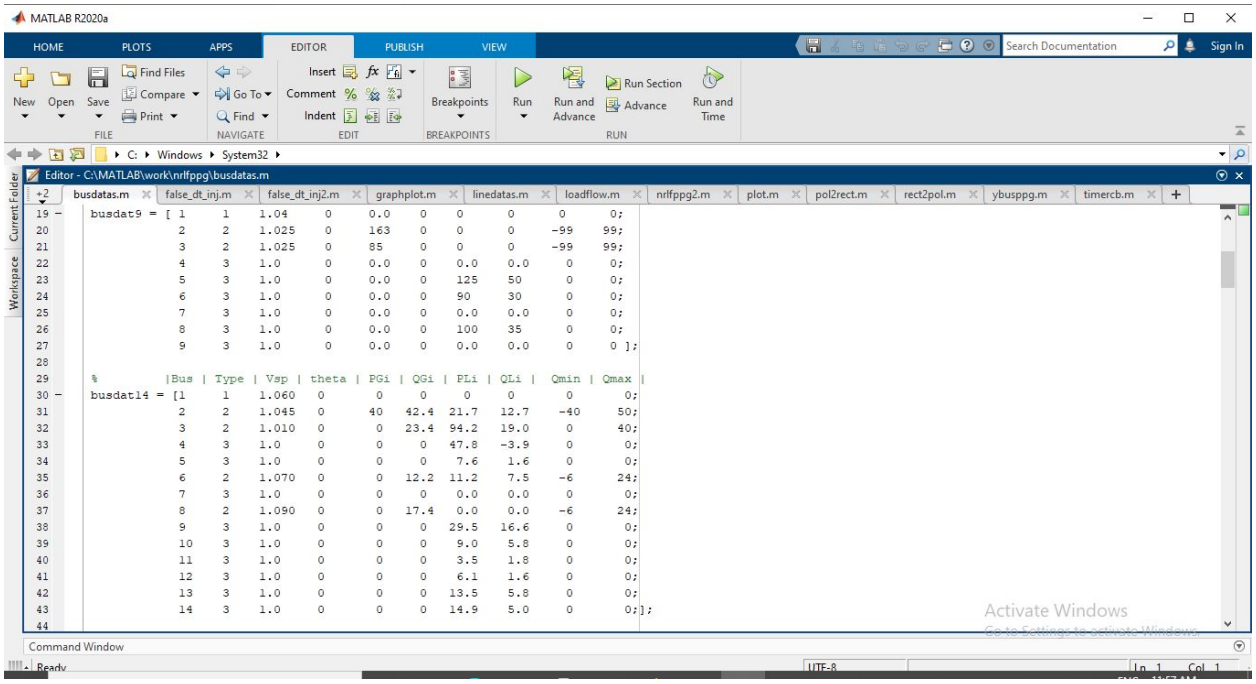

MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\plot.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
X = J\M; % INV(J) x M, Correction Vector..
dTh = X(1:nbus-1); % Change in Voltage Angle..
dV = X(nbus:end); % Change in Voltage Magnitude..
% Update State Vectors (Voltage Angle & Magnitude)
del(2:nbus) = dTh + del(2:nbus);
k = 1;
for i = 2:nbus
if type(i) == 3
V(i) = dV(k) + V(i);
k = k+1;
end
end
Iter = Iter + 1;
Tol = max(abs(M));
end
Del = 180/pi*del; % Convert radians to degrees
% Call LoadFlow
[fb, tb, Pij, Qij] = loadflow(num,V,del,BMva);
% Call GraphPlot
graphplot(V,Del,fb,tb,Pij,Qij,xxx);
end
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1


MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\loadflow.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
for m = 1:nb
for n = 1:nl
if fb(n) == m
p = tb(n);
Iij(m,p) = -(Vm(m) - Vm(p)*a(n))*Y(m,p)/a(n)^2 + b(n)/a(n)^2*Vm(m); % Y(m,n) = -y(m,n)..
% Iij(m,p) = -(Vm(m) - Vm(p)*a(n))*Y(m,p)/a(n)^2;
Iij(p,m) = -(Vm(p) - Vm(m)/a(n))*Y(p,m) + b(n)*Vm(p);
% Iij(p,m) = -(Vm(p) - Vm(m)/a(n))*Y(p,m);
elseif tb(n) == m
p = fb(n);
Iij(m,p) = -(Vm(m) - Vm(p)/a(n))*Y(p,m) + b(n)*Vm(m);
% Iij(m,p) = -(Vm(m) - Vm(p)/a(n))*Y(p,m);
Iij(p,m) = -(Vm(p) - Vm(m))*Y(m,p)/a(n)^2 + b(n)/a(n)^2*Vm(p);
% Iij(p,m) = -(Vm(p) - Vm(m))*Y(m,p)/a(n)^2;
end
end
end
% for m = 1:nl
% p = fb(m); q = tb(m);
% Iij(p,q) = -(Vm(p) - Vm(q))*Y(p,q); % Y(m,n) = -y(m,n)..
% Iij(q,p) = -Iij(p,q);
% end
Iijr = real(Iij);
Iiji = imag(Iij);
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1


MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\linedatas.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
% | From | To | R | X | B/2 | X'mer |
% | Bus | Bus | pu | pu | pu | TAP (a) |
linedat9 = [1 4 0.0000 0.0576 0.0000 1
2 7 0.0000 0.0625 0.0000 1
3 9 0.0000 0.0586 0.0000 1
4 5 0.0100 0.0850 0.1760/2 1
4 6 0.0170 0.0920 0.1580/2 1
5 7 0.0320 0.1610 0.3060/2 1
6 9 0.0390 0.1700 0.3580/2 1
7 8 0.0085 0.0720 0.1490/2 1
8 9 0.0119 0.1008 0.2090/2 1];
% | From | To | R | X | B/2 | X'mer |
% | Bus | Bus | pu | pu | pu | TAP (a) |
linedat14 = [1 2 0.01938 0.05917 0.0264 1
1 5 0.05403 0.22304 0.0246 1
2 3 0.04699 0.19797 0.0219 1
2 4 0.05811 0.17632 0.0170 1
2 5 0.05695 0.17388 0.0173 1
3 4 0.06701 0.17103 0.0064 1
4 5 0.01335 0.04211 0.0 1
4 7 0.0 0.20912 0.0 0.978
4 9 0.0 0.55618 0.0 0.969
5 6 0.0 0.25202 0.0 0.932
6 11 0.09498 0.19890 0.0 1
6 12 0.12291 0.25581 0.0 1
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1

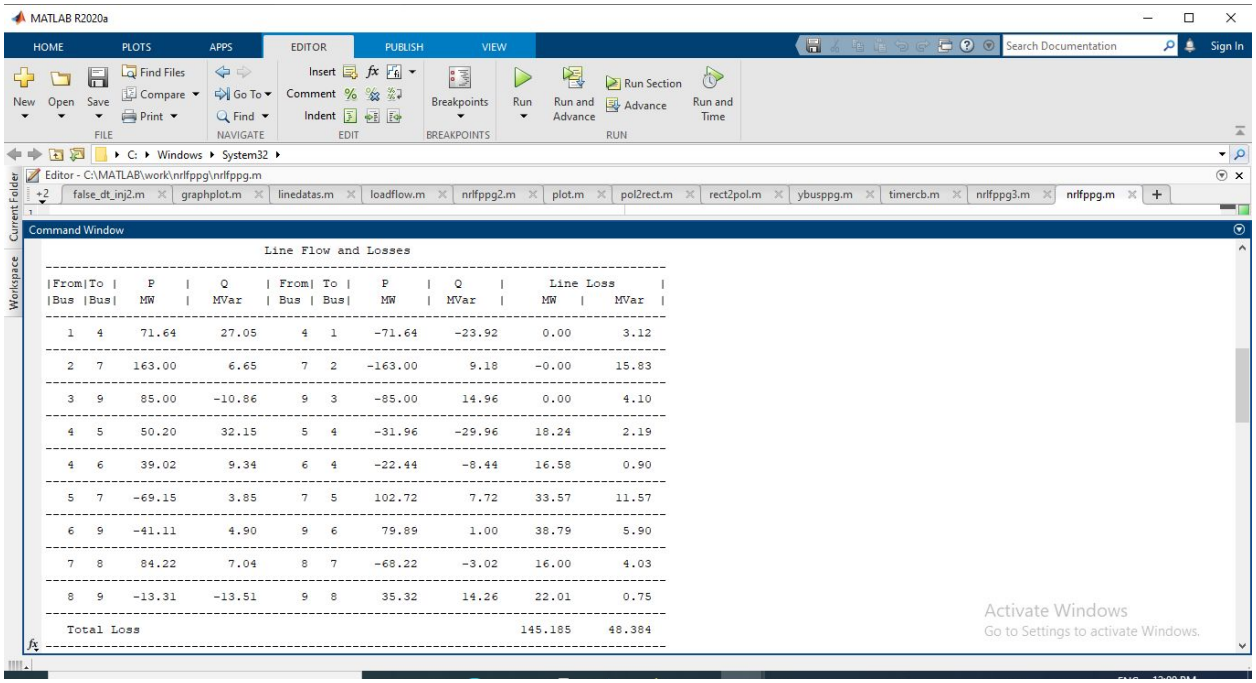

MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\graphplot.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
nrlfppg.m
% Create Edge Labels...
EdlbP = Pij(sub2ind(size(Pij), fb,tb));
EdlbQ = Qij(sub2ind(size(Pij), fb,tb));
kP = strcat(num2str(round(EdlbP,1)),' MW, ');
kQ = strcat(num2str(round(EdlbQ,1)),' MVAR');
kPQ = string(strcat(kP, kQ));
% Create Node Labels..
nb = 1:length(V);
kn = strcat(num2str(nb'),' = ') ;
kV = strcat(num2str(round(V,3)), 'V,');
kD = strcat(num2str(round(Del,2)),'°');
kVD = string(strcat(kn, kV, kD));
% Create a Directed Graph..
G = digraph(fb,tb);
G.Edges.Power = kPQ;
G.Nodes.Size = kVD;
% Plot
p = plot(G,'EdgeLabel',G.Edges.Power, 'NodeLabel',G.Nodes.Size,'Layout','force');
set(p, 'LineWidth', 2, 'MarkerSize', 7, 'Marker', 's', ...
'NodeColor', 'm', 'NodeFontWeight', 'bold', 'NodeLabelColor','r');
if xxx~=0
highlight(p,xxx, 'NodeColor', 'g')
end
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1

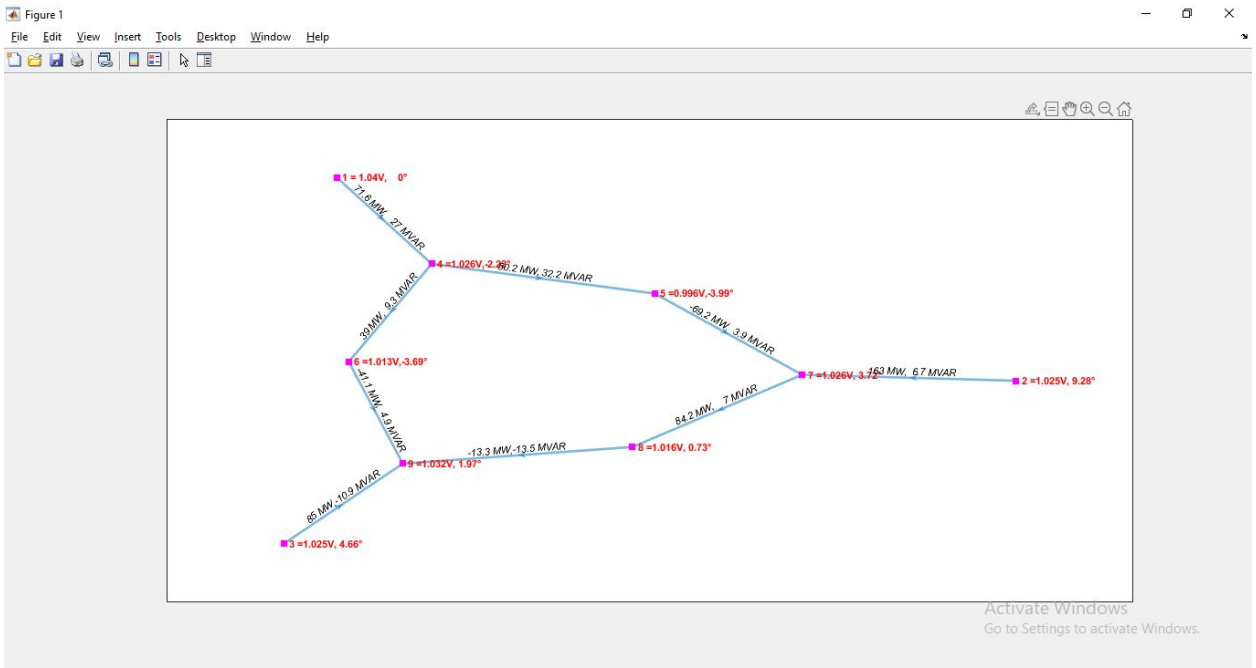

MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\false_dt_inj.m
false_dt_inj.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
nrlfppg3.m
function [busd, xxx] = false_dt_inj(busd)
[row column] = size(busd); %size of bus
chosen_column = randi([3,3],1,1); %coloumn choose random
range = max(abs(busd(:,chosen_column))) ; %range set
chosen_row = randi([1,row],1,1); %row choose %random value
busd(chosen_row,chosen_column) = range*randi([1,2]); %random value
xxx = chosen_row;
end
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1


MATLAB R2020a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Windows ▸ System32 ▸
Editor - C:\MATLAB\work\nrlfppg\busdatas.m
busdatas.m
false_dt_inj.m
false_dt_inj2.m
graphplot.m
linedatas.m
loadflow.m
nrlfppg2.m
plot.m
pol2rect.m
rect2pol.m
ybusppg.m
timercb.m
busdat9 = [1 1 1.04 0 0.0 0 0 0 0 0;
2 2 1.025 0 163 0 0 0 -99 99;
3 2 1.025 0 85 0 0 0 -99 99;
4 3 1.0 0 0.0 0 0.0 0.0 0 0;
5 3 1.0 0 0.0 0 125 50 0 0;
6 3 1.0 0 0.0 0 90 30 0 0;
7 3 1.0 0 0.0 0 0.0 0.0 0 0;
8 3 1.0 0 0.0 0 100 35 0 0;
9 3 1.0 0 0.0 0 0.0 0.0 0 0];
% |Bus | Type | Vsp | theta | PGi | QGi | PLi | QLi | Qmin | Qmax |
busdat14 = [1 1 1.060 0 0 0 0 0 0 0;
2 2 1.045 0 40 42.4 21.7 12.7 -40 50;
3 2 1.010 0 0 23.4 94.2 19.0 0 40;
4 3 1.0 0 0 0 47.8 -3.9 0 0;
5 3 1.0 0 0 0 7.6 1.6 0 0;
6 2 1.070 0 0 12.2 11.2 7.5 -6 24;
7 3 1.0 0 0 0 0.0 0.0 0 0;
8 2 1.090 0 0 17.4 0.0 0.0 -6 24;
9 3 1.0 0 0 0 29.5 16.6 0 0;
10 3 1.0 0 0 0 9.0 5.8 0 0;
11 3 1.0 0 0 0 3.5 1.8 0 0;
12 3 1.0 0 0 0 6.1 1.6 0 0;
13 3 1.0 0 0 0 13.5 5.8 0 0;
14 3 1.0 0 0 0 14.9 5.0 0 0;];
Activate Windows
Command Window
Ready
UTF-8
Ln 1 Col 1

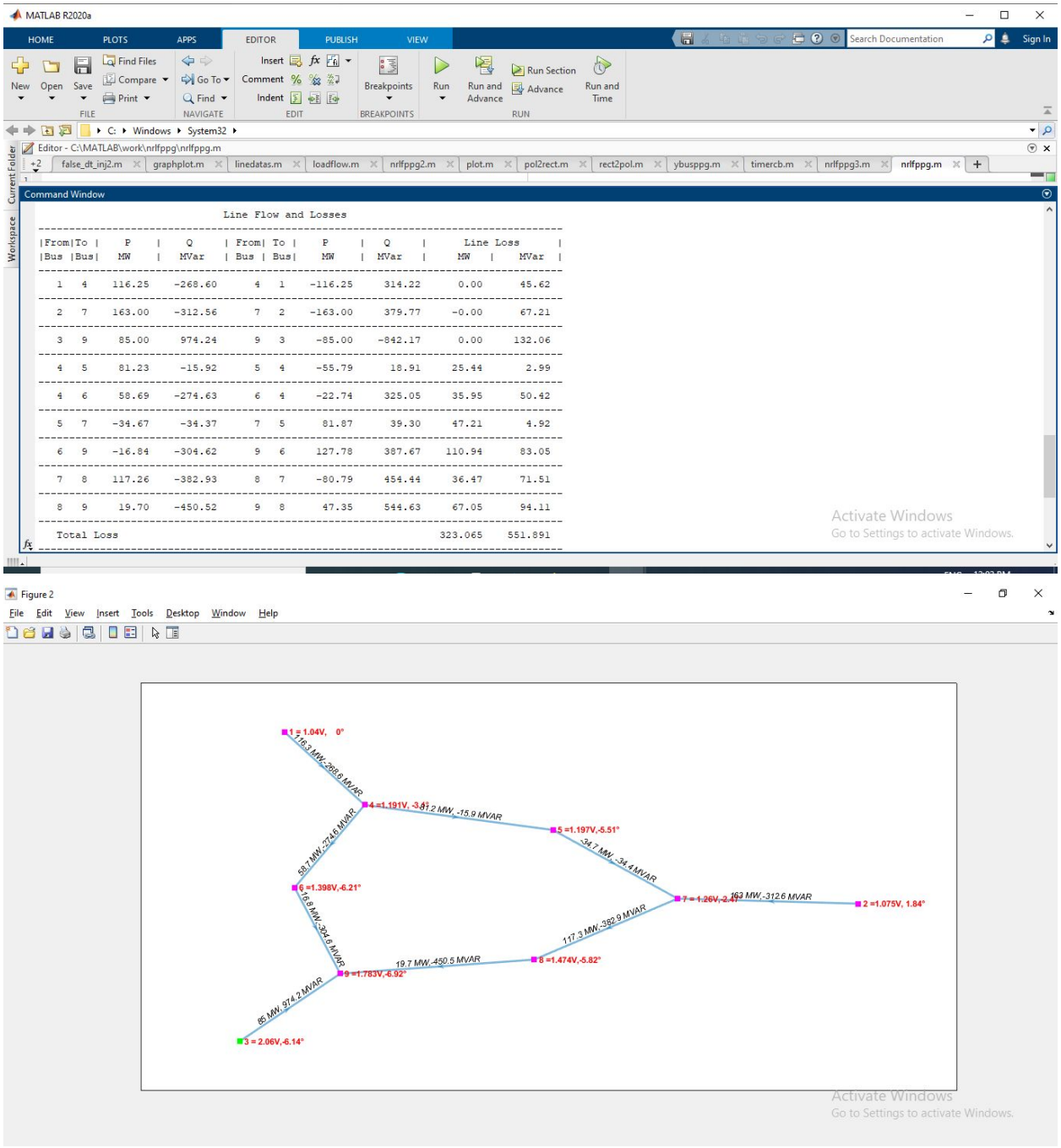
OUTPUT

IEEE - 9 BUS SYSTEM

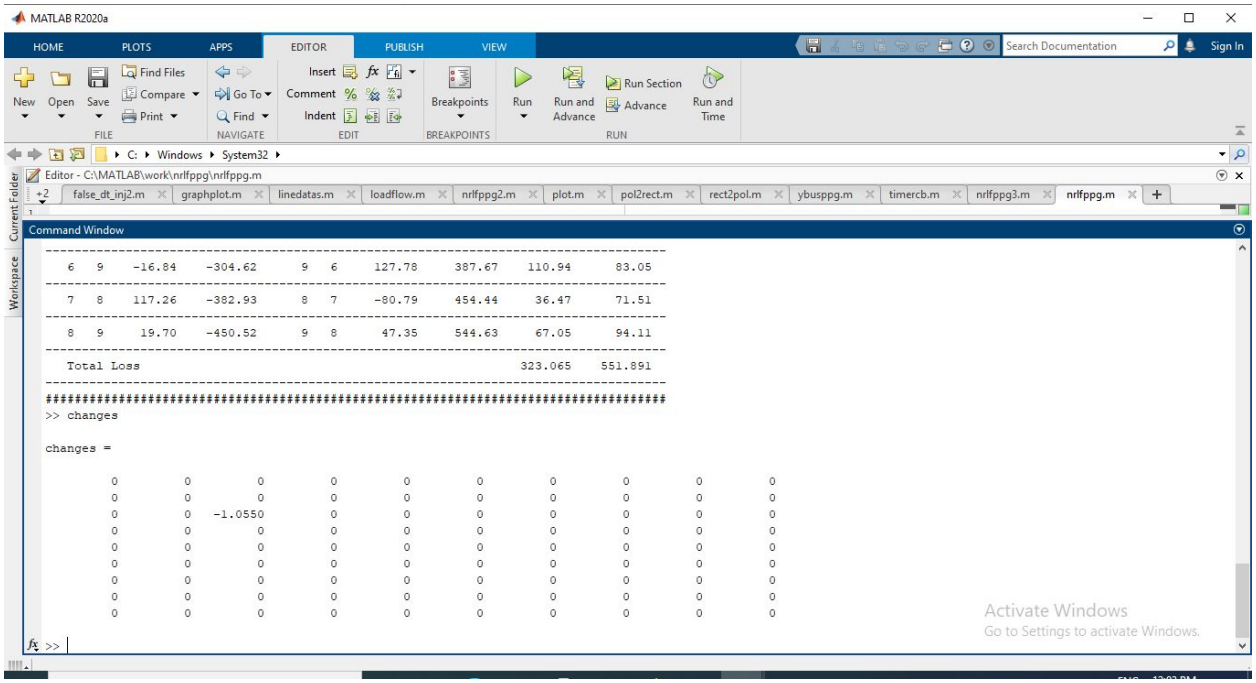




Before FDI attack

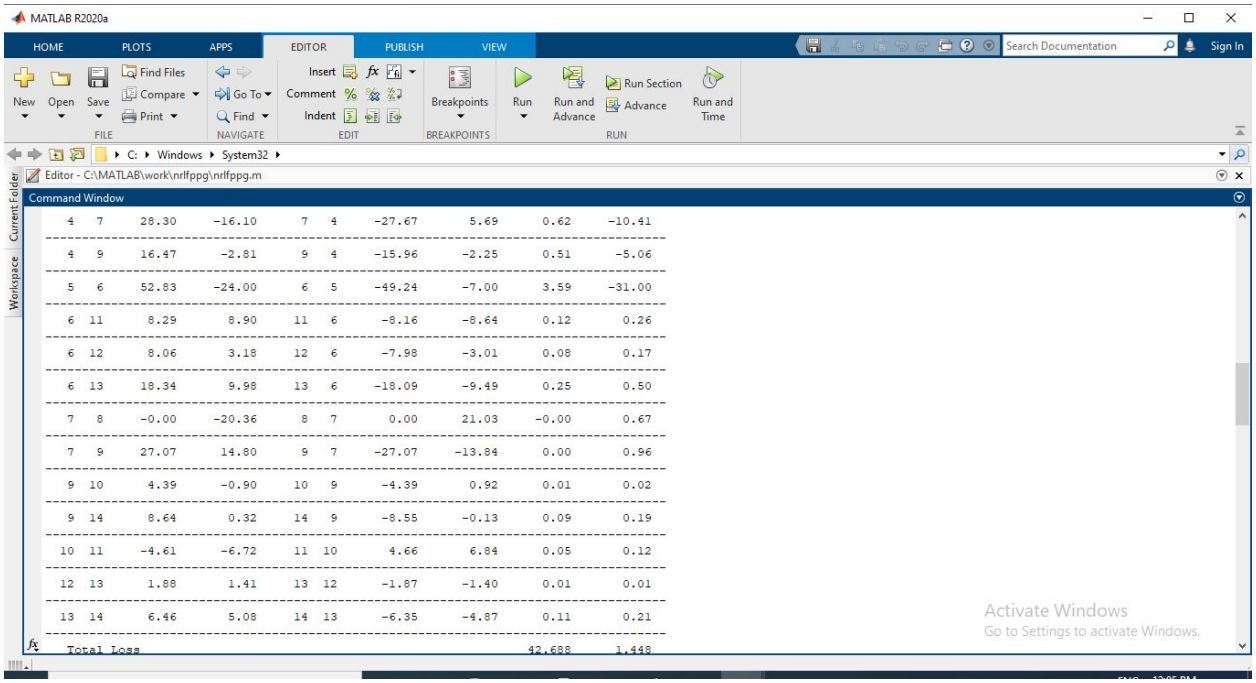


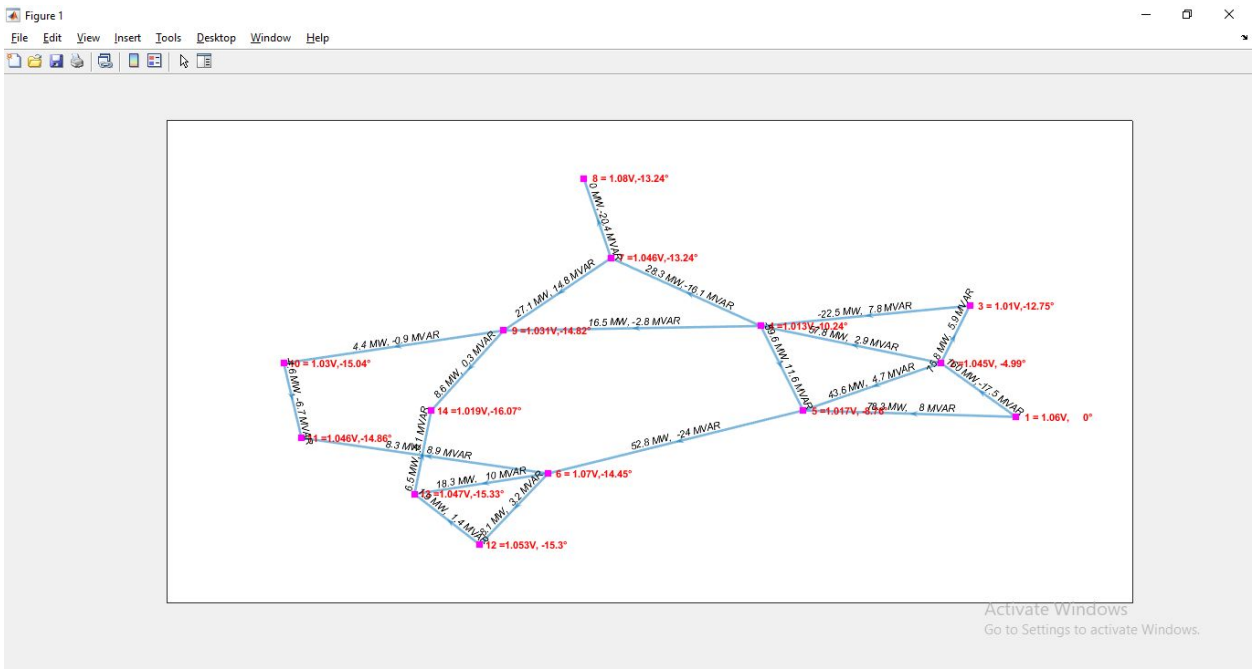
After FDI attack



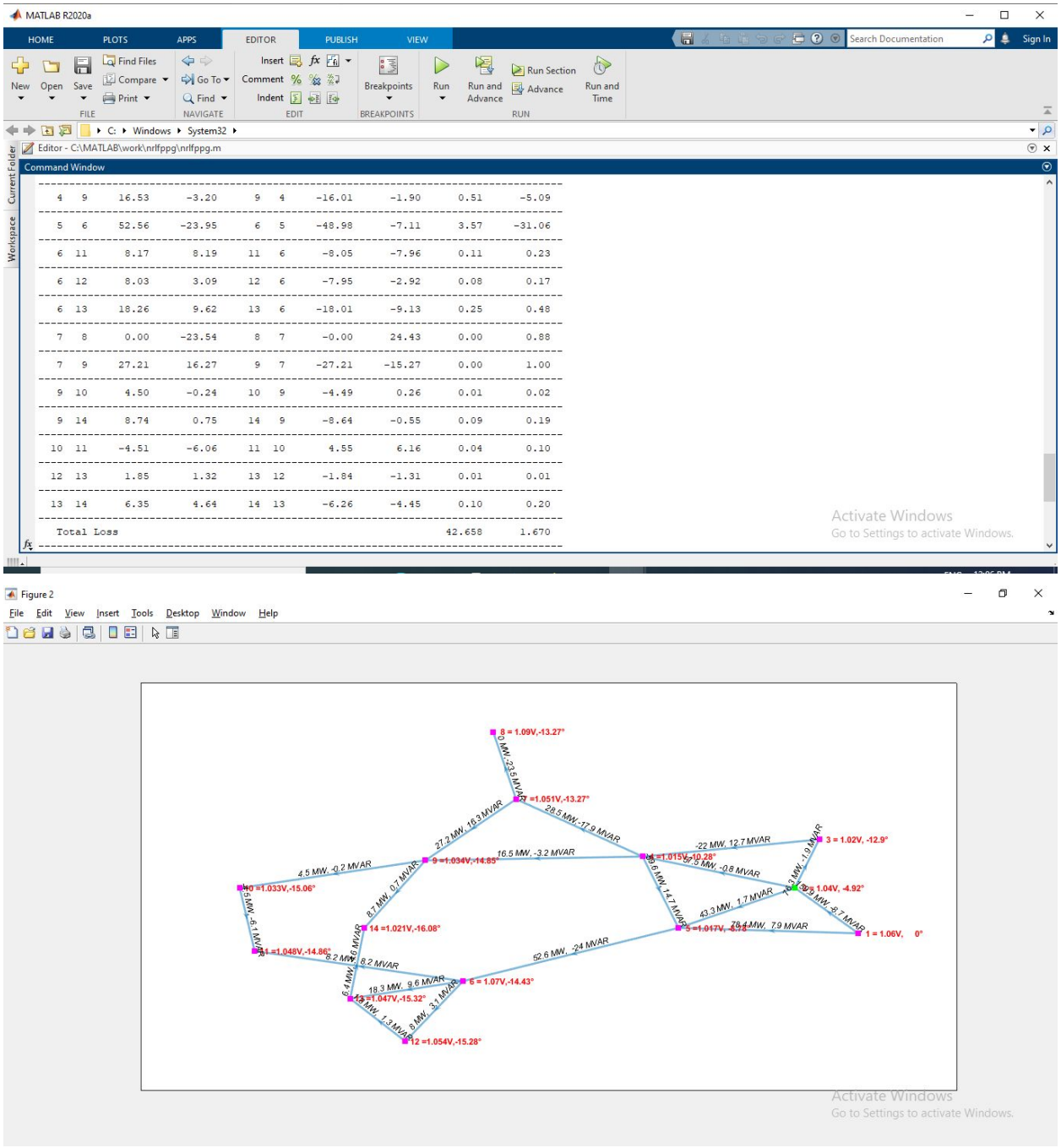
Node 3 is attacked

IEEE - 14 BUS SYSTEM

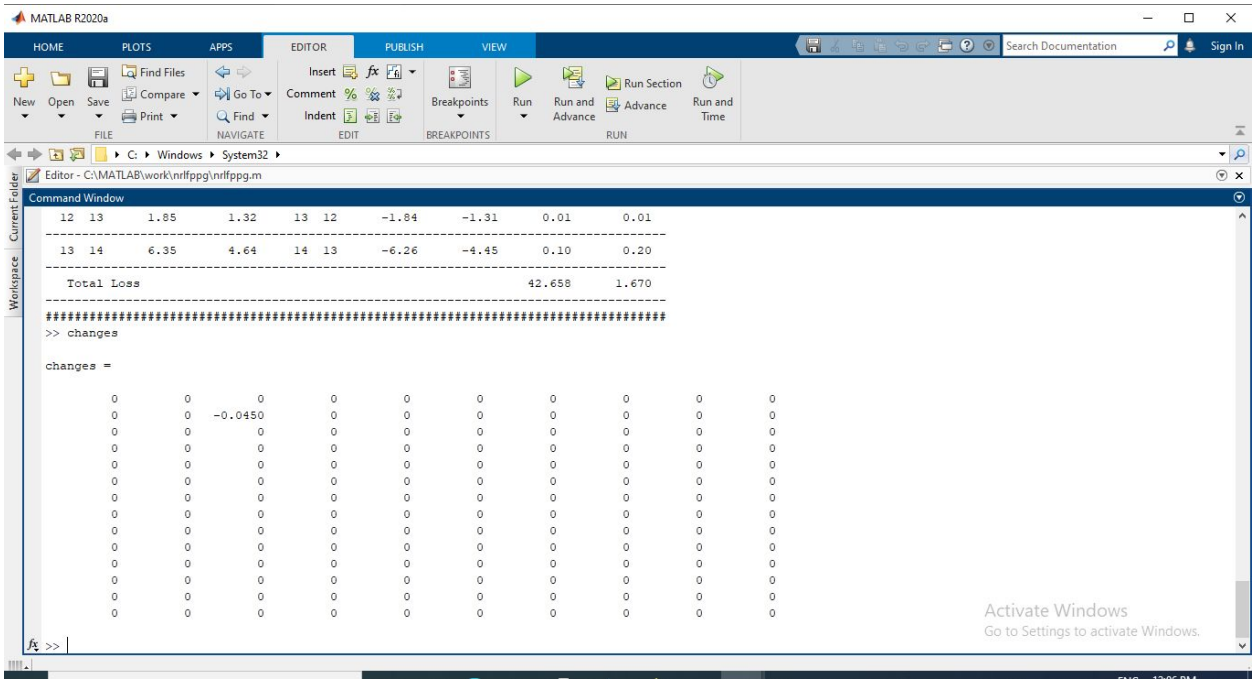




Before FDI attack



After FDI attack



Node 2 has been attacked

CHAPTER 5
RESULT AND CONCLUSION

In this project we proposed a novel method to inject false data in our smart grids along with a method to detect an undetectable attack. Our outputs are in the form of graphs and data is represented in the Jacobian matrix. The attacked node is highlighted with a different colour in the graph.

In the end we also gave a matrix showcasing the amount of voltage change happened at the attacked node which affected the entire system.

CHAPTER 6
REFERENCES

- A. Niglia, *Critical Infrastructure Protection Against Hybrid Warfare Security Related Challenges.*, ser. NATO Science for Peace and Security Series, D, Information and Communication Security. IOS Press, 2016, vol. 46.
- T. G. Lewis, *Critical infrastructure protection in homeland security: defending a networked nation*. Hoboken, N.J.: Wiley, 2006.
- S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, “Catastrophic cascade of failures in interdependent networks,” *Nature*, vol. 464, no. 7291, p. 10251028, 2010.
- A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, “Secure control systems: A quantitative risk management approach,” *IEEE Control Systems*, vol. 35, no. 1, p. 2445, Feb. 2015.
- E. Smith, S. Corzine, D. Racey, P. Dunne, C. Hassett, and J. Weiss, “Going beyond cybersecurity compliance: What power and utility companies really need to consider,” *IEEE Power Energy Mag.*, vol. 14, no. 5,p. 4856, Sep. 2016.
- S. Soltan, M. Yannakakis, and G. Zussman, “Power grid state estimation following a joint cyber and physical attack,” *IEEE Trans. Control of Network Syst.*, vol. 5, no. 1, p. 499512, Mar. 2018.
- E. Drayer, N. Kechagia, J. Hegemann, M. Braun, M. Gabel, and R. Caire, “Distributed self-healing for distribution grids with evolving search space,” *IEEE Trans. Power Del.*, vol. 33, no. 4, p. 17551764, Aug. 2018.
- T. Routtenberg and L. Tong, “Joint frequency and phasor estimation under the KCL constraint,” *IEEE Signal Process. Lett.*, vol. 20, no. 6, p. 575578, June. 2013.
- T. Routtenberg, R. Concepcion, and L. Tong, “PMU-based detection of voltage imbalances with tolerance constraints,” *IEEE Trans. Power Del.*,vol. 32, no. 1, p. 484494, Feb. 2017.
- T. Routtenberg and Y. C. Eldar, “Centralized identification of imbalances in power networks with synchrophasor data,” *IEEE Trans. Power Syst.*, vol. 33, no. 2, p. 19811992, Mar. 2018.