CAPS, Study

By: Mohamed A. Numair

1- CONTENTS

2- INTRODUCTION

In the exam comments are more important than codes

In Oral Exam, you will be asked in Power System Analysis

1.1 Power System

Generation	Prime mover, source of energy, protection and control of generation
Transmission	Transformer., cables, Towers, insulators, protection and control
Distribution	Loads ch/c, demand power, voltage, Impedance (VI = const)

1.2 Computer Application in Power system aspects

We use computer in 2 things concerning power systems

1.2.1 Model

We create a model for an actual system that for the same input for both will have the same output from both

1.2.2 Analysis

Comparing the system performance to the ideal performance

We need analysis to be (Easy, accurate, repeatable, not time consuming)

Making analysis for economic revenue

And improving voltage regulation and power factor

1.2.3 Planning

Load Estimation – is the load far or close – Transmission Lines- Energy Source – Generator Power

1.2.4 Load Flow study (output v, δ, P, Q)

1.2.5 Testing, transient analysis, Economics, control stability

1.2.6 Control Panels

1.2.7 Modeling linear and non-linear circuits

2 YBUS

2.1 Types of buses

	Known	Unknown
Slack	v, δ	$P, Q\ assumed$
Load	P, Q	v, δ
Voltage control	v, P	Q, δ

2.2 Concept

$$Y_{BUS} = \begin{bmatrix} ON & off & off \\ off & ON & off \\ off & off & ON \end{bmatrix}$$

<u>On Diagonal</u> :are self admittance=earth admittance+mutual admittaqnce with other buses $i.e.\ \ y_{11} = y_{10} + y_{12} + y_{13} + \cdots + y_{1n}$

<u>off Diagonal</u> are the mutual admittance between two buses

$$Y_{ij} = Y_{ji} = -y_{ij} = -y_{ji}$$

2.3 Input-Process-Output

<u>Calculations consider (</u> System balanced – Steady State – PU)

Inputs	- System configuration (from and to) - Impedances of T.L (R, X_l, X_c) generator internal impedance - Grounding system - No. of. Buses - Load type (static, dynamic) - Base values (S, V)
Process	Calculate all elements of Ybus and allocate elements in the matrix
Output	Ybus of the power system

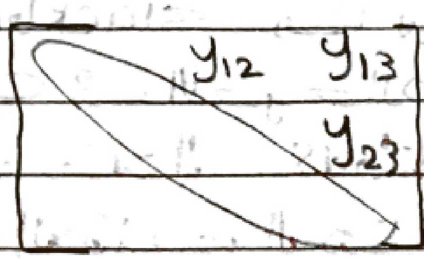
When calculating off diagonal we use fb(i) ~=0 && tb(i) ~= 0 <u>then do</u>
or we can say if fb(i)=0 || tb(i)=0 do nothing <u>else do</u>

we present Tr with impedance in P.U.

Q : Where do we use YBUS other than in Load Flow ?

2.4 CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program calculates Y-Bus matrix to a given electrical power system %
% This program requires input matrix which discribs the configuration of  %
%                               the system                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear all
  %(1) Getting Data from User or manually
%-------------------------- line data ------------------------------------
%           [ From |  To   |   R     |   X     |   yc/2  ]
%             Bus  | Bus   |   pu    |   pu    |   pu
linedata=input('enter the line data=')
% here the user will enter or I can add it manually



% (2) Defining The Column in the linedata
fb=linedata(:,1);                            %from busbar
tb=linedata(:,2) ;                           %to busbar
r=linedata(:,3) ;                           %line resistance
x=linedata(:,4) ;                            %line reactance
b=i.*linedata(:,5);

% (3) Calculating the number of branches (nbr) and buses number (nbus)
nbr=length(fb)                              %no of branches (TL)
nbus=max(max(fb),max(tb))                   %no of buses

% (4) Calculating the admittances in a matrix "y"
z=r+1i.*x ;
y=1./z;

% (5) Making an empty matrix "Ybus" to put data in
Ybus=zeros(nbus,nbus);



% (6) Calculating off diagonal elements
for i=1:nbr
    if (fb(i)~=0 && tb(i)~=0)   % make sure it's off diagonal
    Ybus(fb(i),tb(i))=Ybus(fb(i),tb(i))-y(i);
    %sum if there are more than one TL between 2 buses
    Ybus(tb(i),fb(i))=Ybus(fb(i),tb(i));
    % symmetricallty filling
    end
end



%(7) Calculating on diagonal elements
for i=1:nbus
    for k=1:nbr
        if (fb(k)==i||tb(k)==i)       %To check all lines connected to Bus i
        Ybus(i,i)=Ybus(i,i)+y(k)+b(k);
        end
    end
end

% (8) Outputting the Y matrix
Ybus
```



SubProg to Calc connected bus

يعني ناخد عدد الـ buses الموصلة بعضها ازاي "متوصل T.L" في الـ half offdiag.

For K=nbus:1

بيدخل الـ ondiagonal بالعكس "من تحت لفوق" "بالترتيب"

For K = 1 : nbus

هيدخل الـ ondiagonal بالترتيب من فوق لتحت

```
%QUIZ
```

```
%%
%Q1

% to calc connected buses


y = [ 1 9 9 ;
      9 2 0;
      9 0 7]
nb=3;
sum=0;
for n=1:nb
  for k=1:nb
      if n~=k
      if y(n,k) ~=0
   sum=sum+1 ;

      end
      end
  end
end
```

```
nconnect = sum/2


nbr = length ( fb) ;  % number of all TL

% no of TL

Sum =0 ;
For i = 1 :nbus
If ( fb(i) ~= 0 && tb(i) ~=0 )
  Sum = sum +1

End

Sum ;  % number of TL between buses
```

```
%%
%Q2

clc
clear all

nb= 50 ;
x=0 ;

for i= nb-2 : nb;

    %code to be repeated 3 times %
    x=x+1;

end
```

3 METHODS OF LOAD FLOW CALCULATIONS

1- Gauss Seidel
2- Newton Raphson (most accurate)
3- Fast Decoupled ($X_{TL} \gg R$)

Most accurate is NR , accurate and quick (double quick cuz half calculations) is FD

But, Gauss seidel can diverge instead of converging

<u>Why do we study load flow ?</u>

1- To know if the bus has reached it's rated limit
2- Did the conductor reach it's ($I_{net} = X\ I_{Max}$) if the loadings is not distributed 95% and 10% then we must redistribute
3- Load equals % of the generated
4- PF improvement losses
5- $\Delta V\ \%\ Regulation$ using condensers
6- To know if the bus needs any **compensation** or **load to be decreases**
7- If the generator is heavily loaded (e.g. 99%) or not
8- If there's a fault in one line can it's load be distributed to other working lines
9- Redistribute loads
10- Capability of the Power System and its components
11- Study Power Losses between buses
12- Stability Studies

In some methods we use $\alpha: acceleration\ factor$ ($1.2 \rightarrow 1.6$) to reduce number of iterations

Why is the error in NR is so small and there is no accumulating error ?

Because we deal with rate of change not the value itself

Load Flow calculation outputs

1- P,Q for Slack bus
2- δ ,Q for PV bus
3- V, δ for PQ bus
4- $T.L$ current ($(v_1 - v_2)y$
5- Which T.L. reached it's limits (redistribute)
6- Voltage Regulation (over – under)
7- Losses (S12 , S21) [need Qinj ?+ve or -ve]
8- Power Flow direction
9- δ ($\propto f$) as indication of stability
10- δ to $\delta_{critical}$
11- Need control or not ?

	G.S	N.R	F.D
no of iterations	. عدد iteration بين F.D و N.R	. اقل عدد iteration من FD ، GS	. اكبر عدد iteration من GS و NR .
Divergence.	تباعد من (error) اقل من NR واكبر من FD	تباعد كبير بين قيم error في كل ite	. اقل تباعد بين قيم error من كل ite .
accuracy	accuracy between N.R and FD	High accuracy	Low accuracy X>>R
using online Calculation	using off line Calculation	using online Calculation	Sometimes use online Calculation
	عدد معادلات اقل . واقل تعقيداً من NR	عدد معادلات اكثر واكثر تعقيداً	اقل من عدد المعادلات واقل تعقيداً

4 GAUSS SEIDEL GS

4.1 Remember

	input	calculate
Slack	v, δ	$P, Q\ assumed$
Load	P, Q	v, δ
Voltage control	v, P	Q, δ

PV buses is used to maintain voltage at certain bus constant thus maintain system voltage using Q but if we exceed limit we use max or min and make PV to PQ

Gauss Main Problem is that it depends largely on initial value

4.2 Input-Process-Output

<table>
<tr><td>Input</td><td>1- Bus type
Slack -> 0
PV Bus -> 1
PQ Bus -> 2
2- Bus Number
3- Initial Values (Limits, $Q_{inj}, |V|, \delta, P_G, Q_G, P_D, Q_D$) (voltages =1 , other = 0)
4- Assume actual (scheduled) S or P and Q</td></tr>
<tr><td>Process</td><td>

(1) $$P_i^{(k+1)} = Real\left[v_i^{*(k)} \left(v_i^{(k)} Y_{ii} + \sum_{\substack{j=1 \\ i \neq j}}^{n} y_{ij}\, V_j^{(k)} \right) \right]$$

(2) $$Q_i^{(k+1)} = -img\left[v_i^{*(k)} \left(v_i^{(k)} Y_{ii} + \sum_{\substack{j=1 \\ i \neq j}}^{n} y_{ij}\, V_j^{(k)} \right) \right]$$

$$v_i^{(k+1)} = \frac{1}{Y_{ii}} \left[\frac{P_i - jQ_i}{v_i^{*(k)}} - \sum y_{ij}\, v_j^{(k)} \right]$$

v_i *and* Q_i *are iteration equations*
NOTE as V is a complex number it's V = VM(cos(VA) + i sin(VA))
Calculate vi for all buses
Qi is calculated only for PV bus and we check limits
Calculate currents between 2 buses
Calculate losses and capability
Find solutions , like if bus voltage is 0.5 ,add bar or decrease loads
(3)
Stop after
1- number of iterations (experience – error neglected – no certain inform)(easy)
2- error is < tolerance (most accurate)
3- until the V equation is the same as actual or P</td></tr>
<tr><td>output</td><td>Slack bus P,Q once
PV Bus δ, Q iteration
PQ Bus $|v|, \delta$ iteration</td></tr>
</table>

4.3 CODE

```
    %%%%%%%%%%%%%%%%%%%% GAUSSE METHOD %%%%%%%%%%%%%%%%%%%%%%%
clear all
clc

% (1) Ybus=input('enter Ybus=')
 Ybus=[8.985-j*44.836  -3.816+j*19.078   -5.170+j*25.848 0
      -3.816+j*19.078   8.985-j*44.836   0  -5.170+j*25.848
      -5.170+j*25.848  0  8.193-j*40.864  -3.024+j*15.119
       0  -5.170+j*25.848  -3.024+j*15.119  8.193-j*40.864]
% (2) busdata=input('enter busdata=')
%-------------------------- Bus data ----------------------------------
%      [bus  |bus | vm |  va    | Pg   | Qg   |  Pd | Qd | Qmin | Qmax | Qinj]
%       no.  |type|  pu |degree|  pu  |  pu |  pu | pu   |pu   |pu

busdata=[1 0 1 0  0 0   .5 .3 0 0 0 ;
         2 2 1 0  0 0   1.7 1.0535 0 0 0 ;
         3 2 1 0  0 0   2 1.2394 0 0 0;
         4 1 1.02 0 3.18 0 .8 .4958 0 0 0];

 % (3) taking out data columns
Sb=1; %Sbase
nobus=busdata(:,1);
type=busdata(:,2);
vm=busdata(:,3);
va=busdata(:,4);
Pg=busdata(:,5)/Sb;         % if Pg=......MW else(pu) Sb=1;
Qg=busdata(:,6)/Sb;         % if Pg=......MVAR else (pu) Sb=1
Pd=busdata(:,7)/Sb;
Qd=busdata(:,8)/Sb;
Qmin=busdata(:,9)/Sb;
Qmax=busdata(:,10)/Sb;
Qinj=busdata(:,11)/Sb;
% (4) get number of buses
nbus=length(nobus);

% (5) make out matrices of V,Q,P,S
v=vm.*cosd(va)+1i*vm.*sind(va);
Q=Qg-Qd+Qinj;
P=Pg-Pd;
s=P+1i*Q;

%(6) Prepare to Begin of Gauss  (v2,tol,dv,itr)
v1=v;
tol=10^-6;
dv=1;
itr=0;

%(7) start loop condition
while (dv>tol)

% (8) calculate sum part of S,V for no slack
for i=1:nbus
    if type(i)~=0  % not slack
        sum=0;
    for j=1:nbus
        if j~=i
          sum=sum+Ybus(i,j)*v(j);
        end
    end
```

```
% (9) for PV calculate Q and limit it

    if type(i)==1 %pvbus
        Q(i)=-imag(conj(v(i))*(Ybus(i,i)*v(i)+sum));
% (9-1) check limit of generation
        Qg(i)=Q(i)+Qd(i)-Qinj(i);
    if Qmax(i) ~=0 % if Qmax is given
        if Qg(i)>Qmax(i)
                    Qg(i)=Qmax(i);
        elseif Qg(i)<Qmin(i)
                        Qg(i)=Qmin(i);
        end
    end

% (10) Output Q,S, Va of PV bus
        Q(i)=Qg(i)-Qd(i)+Qinj(i);
        s(i)=P(i)+1i*Q(i);
        v(i)=(1/Ybus(i,i))*((conj(s(i))/conj(v(i)))-sum);
        va(i)=angle(v(i))*180/pi;
        v(i)=vm(i)*cosd(va(i))+1i*vm(i)*sind(va(i));
% (11) Output V of PQ bus
    else%PQ bus if type(i)== 2
      v(i)=(1/Ybus(i,i))*((conj(s(i)))/conj(v(i))-sum);
    end
    end
% (12) Updating values (vm,va,mis)
    vm(i)=abs(v(i));
    va(i)=angle(v(i))*180/pi;
    mis(i)=abs(v(i)-v1(i)); % error or mismatch
end
% (13) Updating values (itr,dv,v1)
itr=itr+1;
dv=max(mis); % maximum value of error
v1=v;
end
% (14) Output values (v,vm,va,it,mis)
v
vm
va
itr
mis
```

```
        %%%%%%%%%%%%%%%%%% POWER FLOW %%%%%%%%%%%%%%%%%%%%
for n=1:nbus
    for k=1:nbus
        if n~=k
      s(n,k)=v(n)*conj(Ybus(n,k)*(v(k)-v(n)));
      s(k,n)=v(k)*conj(Ybus(n,k)*(v(n)-v(k)));
      sloss(n,k)=s(n,k)+s(k,n);
        end
    end
end
% p,Q forslack buses ;Q for pv buses
for n=1:nbus
    if type(n)~=2 %no pQ bus
        sum=0;
        for k=1:nbus
                sum=sum+v(k)*Ybus(n,k);
        end
        s(n)=conj(conj(v(n))*sum);
        Q(n)=imag(s(n));
        Qg(n)=Q(n)+Qd(n)-Qinj(n);
        if type(n)==0 %slack bus
        P(n)=real(s(n));
        Pg(n)=P(n)+Pd(n);
        end
    end
end
sloss
Q
Pg
Qg
ploss=real(sloss)
```

5 NEWTON RAPHSON

- Independent of initial condition because we work with rate of change
- Advantage: Most Accurate , Always Converges
- Disadvantage: it's complicated , uses matrices calculations

NOTE : we use partial $\frac{\partial P}{\partial V}$ not $\frac{dP}{dV}$ because P depends on many variables not only V

P depends slightly on V as we don't change it that much but depends heavily on δ

5.1 Remember Newton – Raphson

$$\overbrace{\begin{bmatrix}\Delta \boldsymbol{P}\\ \Delta \boldsymbol{Q}\end{bmatrix}}^{\text{mismatch matrix}} = \overbrace{\begin{bmatrix}\begin{bmatrix}\frac{\partial \mathbf{P}}{\partial \boldsymbol{\delta}}\end{bmatrix} & \begin{bmatrix}\frac{\partial \mathbf{P}}{\partial \mathbf{v}}\end{bmatrix}\\ \begin{bmatrix}\frac{\partial \mathbf{Q}}{\partial \boldsymbol{\delta}}\end{bmatrix} & \begin{bmatrix}\frac{\partial \mathbf{Q}}{\partial \mathbf{v}}\end{bmatrix}\end{bmatrix}}^{\boldsymbol{J}} \overbrace{\begin{bmatrix}\boldsymbol{\Delta\delta}\\ \boldsymbol{\Delta v}\end{bmatrix}}^{\boldsymbol{output\ missmatch}}$$

$$\overbrace{\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix}}^{\text{mismatch matrix}} = \overbrace{\begin{bmatrix}J1 & \mathrm{J2}\\ \mathrm{J3} & \mathrm{J4}\end{bmatrix}}^{J} \overbrace{\begin{bmatrix}\Delta\delta\\ \Delta \mathrm{v}\end{bmatrix}}^{output\ missmatch}$$

$$J = \begin{bmatrix}\begin{bmatrix}\frac{\partial P}{\partial \delta}\end{bmatrix} & \begin{bmatrix}\frac{\partial P}{\partial v}\end{bmatrix}\\ \begin{bmatrix}\frac{\partial Q}{\partial \delta}\end{bmatrix} & \begin{bmatrix}\frac{\partial Q}{\partial v}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}J_1 & J_2\\ J_3 & J_4\end{bmatrix}$$

$J_1 = \frac{\partial P}{\partial \delta} = \begin{bmatrix}\frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial_3}\\ \frac{\partial P_3}{\partial \delta_2} & \frac{\partial P_3}{\partial \delta_3}\end{bmatrix}$ (sin) Sis of J13	(n-1)*(n-1)
$J_2 = \frac{\partial P}{\partial V} = \begin{bmatrix}\frac{\partial P_2}{\partial V_2}\\ \frac{\partial P_3}{\partial V_2}\end{bmatrix}$ +Cos J24	(n-1)*(n-1-nPV)
$J_3 = \frac{\partial Q}{\partial \delta} = \begin{bmatrix}\frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3}\end{bmatrix}$ (cos) sis of J13	(n-1-nPV) *(n-1)
$J_4 = \frac{\partial Q}{\partial V} = \begin{bmatrix}\frac{\partial Q_2}{\partial V_2}\end{bmatrix}$	(n-1-nPV)*(n-1-nPV)

-Sin J24	

Note : we calculate the change $\Delta\delta$ *and* ΔV so we have to add to the ideal value

5.1.1 Mismatch matrix calculate as follow

$$P_i = \sum_{k=1}^{n} |V_i||V_k||Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_i = -\left[\sum_{k=1}^{n} |V_i||V_k||Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)\right]$$

$$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = \begin{bmatrix}P_{sch(net)} - P_{cal}\\ Q_{sch(net)} - Q_{cal}\end{bmatrix} = \begin{bmatrix}P_{gen} - P_{dem} - P_{cal}\\ Q_{gen} - Q_{dem} - Q_{cal}\end{bmatrix}$$

$P_{sch}\ (given\ ) if\ not = 0$
$P\ and\ Q\ calc\ we\ calculate\ it\ in\ the\ problem\ and\ compare$

For PV bus I have only ΔP because it doesn't have a Qsch

Mismatch dimension is a square 2n-m-2 (n number of bus , m number of PV bus -2 for slack)

We use 2 for loops with 2 ifs

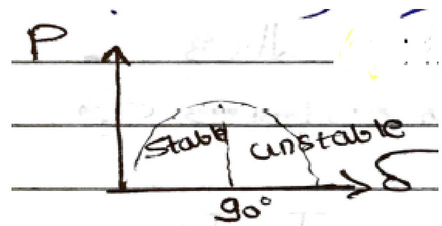
not Slack	Calculate $[\Delta P]$
If PQ bus	*calculate* $[Q]$

5.2 Input-Process-Output

Input	Bus Data Line Data
Process	1- Calc YBUS put it in $ybusm\angle ybusa$ 2- Input tolerance 3- Check for bus type 4- For all buses except slack (ΔP) 5- For all load buses (ΔQ) 6- Calculate first iteration based on initial values 7- Calculate Pi then ΔP then in mismatch 8- Repeat based on calculated values of ($\lvert V\rvert, \delta$) 9- We create the Jacobians with all buses but then we eliminate the slack bus so we need a counter for value and position as they are different HOW TO STOP 10- Number of iteration (quicker than gauss and accurate) 11- Error tolerance (not sure fo V or delta)(not a good solution) 12- Check tolerance < for ΔP_{Max} 13- $[\Delta P]^{n-1}$ we stop when for is near 0 Q: What do we eliminate in PV buses
Output	

Program output :

1- Slack bus P,Q
2- $PV\ bus\ \ \delta, Q$
3- $PQ\ bus\ V\ , \delta$
4- T.L current $\frac{v_1 - v_2}{y}$
5- Which T.L. reach limits
6- Voltage reg. at buses (over voltage or undervoltage)
7- Losses (transmission efficiency) (Qinj needed ?) needed at planning
8- Power Flow direction
9- $\delta\ value$ as indication stability , and frequency
10- How far is δ from $\delta_{critical}$
11- Is control needed or not .



$$\frac{\partial P_i}{\partial \delta_i} = \sum_{\substack{j=1 \\ j \neq i}}^{n} V_i V_j Y_{ij} \sin(\theta_{ij} + \delta_j - \delta_i), \quad \frac{\partial P_i}{\partial \delta_j} = -V_i V_j Y_{ij} \sin(\theta_{ij} + \delta_j - \delta_i)$$

$$\frac{\partial Q_i}{\partial V_i} = -2V_i Y_{ii} \sin\theta_{ii} - \sum_{\substack{j=1 \\ j \neq i}}^{n} V_j Y_{ij} \sin(\theta_{ij} + \delta_j - \delta_i), \quad \frac{\partial Q_i}{\partial V_j} = -V_i Y_{ij} \sin(\theta_{ij} + \delta_j - \delta_i)$$

$$P_i = V_i^2 Y_{ii} \cos\theta_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^{n} V_i V_j Y_{ij} \cos(\theta_{ij} + \delta_j - \delta_i), \quad i = 1,2,3,\ldots\ldots,n$$

$$Q_i = -V_i^2 Y_{ii} \sin\theta_{ii} - \sum_{\substack{j=1 \\ j \neq i}}^{n} V_i V_j Y_{ij} \sin(\theta_{ij} + \delta_j - \delta_i), \quad i = 1,2,3,\ldots\ldots,n$$

$$S_{ji} = V_j I_{ji}^* = V_j [Y_{ji} (V_i - V_j)]^*, \quad S_i^* = V_i^* (Y_{ii} V_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} Y_{ij} V_j)$$

5.3 Jacobian Matrix

5.3.1 For Jacobian 1! احذف صف وعمود سلاك باص في كلهم

$$\frac{\partial P_i}{\partial \delta_i} = \sum_{j \neq i}^{n} |V_i||V_j||Y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i)$$

Here on diagonal $\frac{\partial P_i}{\partial \delta_i}$

Off diagonal $\frac{\partial P_i}{\partial \delta_j}$

- Calculate for all elements including (slack bus)
- Eliminate for slack bus component

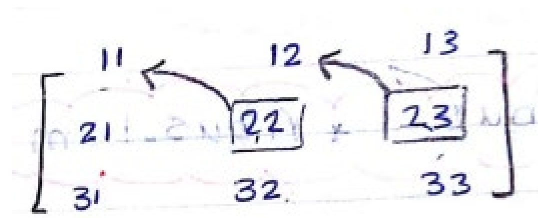
For n=3 , matrix 3*3

1- Slack

2- PV

3- PQ

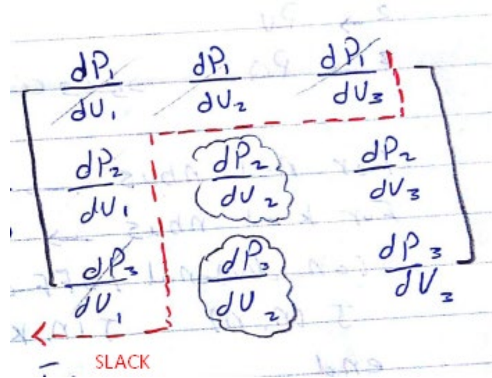
And slack eliminates its




```
For n = 1: nbus
For k = 1 :nbus
if k ~= n  % not slack
J1(n,n) = j1(n,n) + vm(n)*vm(k)*ybusm(n,k)*sin(ybusa(n,k) +va(k)-va(n))
```

For it nbus-1 * nbus-1

Elimination




```
J1E = zeros(nbus-1,nbus-1);
x=0;
for n=1:nbus
if type(n) ~= 0
x= x+1
z=0
for k=1 : nbus
if type (k) ~= 0

z=z+1
J1E(x,z)= j1(n,k)
end
end
end
end
% ( x-> row , z -> coloumn )
% matrix [ nbus-1 * nbus-1 ]
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%OR %%%%%%%%%%%%%%%%%%%%%
PV=find(type==1); %find position of pv bus
nv=length(PV)
sl=find(type==0);
J1(sl,:)=[];
J1(:,sl)=[]
```

5.3.2 For Jacobian 2 احذف عمود

Before elimination it was nbus*nbus

Elimination

$$nbus - 1 * nbus - 1 - m$$

m : no. of PV bus

if type(k) ~= 0 and type(k) ~= 1

NOTE : we can use one condition only

```
type(k)=2 ,%it's PQ
```

5.3.3 For Jacobian3 احذف صف

If type (n) ~= 0 && type (n) = 1

5.3.4 For Jacobian4 احذف صف وعمود

n-1-m * n-1-m

PV, slack

Q : does the program has to count the PV

```
%%%%%%%%%%%%%%%%%%%%%%%%%%% EngNUMAIR %%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear all
%---------------------------(1) YBUS ------------------------------------

Ybus=[8.985-j*44.836  -3.816+j*19.078   -5.170+j*25.848 0
      -3.816+j*19.078   8.985-j*44.836   0  -5.170+j*25.848
      -5.170+j*25.848  0  8.193-j*40.864  -3.024+j*15.119
       0  -5.170+j*25.848  -3.024+j*15.119  8.193-j*40.864];
%OR Ybus=input('enter Ybus');

%--------------------------- (2)Bus data ------------------------------------
%      [bus  |bus | vm |   va   | Pg  | Qg  |  Pd | Qd | Qmin | Qmax | Qinj]
%       no.  |type|  pu |degree|  pu  |  pu |  pu | pu   |pu  |pu

busdata=[1 0 1 0  0 0   .5 .3 0 0 0 ;
         2 2 1 0  0 0   1.7 1.0535 0 0 0 ;
         3 2 1 0  0 0   2 1.2394 0 0 0;
         4 1 1.02 0 3.18 0 .8 .4958 0 0 0];
% OR busdata=input('enter Bus data');
%--------------------------- (3)Column ------------------------------------
Sb=1;
nobus=busdata(:,1);
type=busdata(:,2);
vm=busdata(:,3);
va=busdata(:,4);
Pg=busdata(:,5)/Sb;        % if Pg=......MW ; sb=.... else(pu) Sb=1
Qg=busdata(:,6)/Sb;        % if Pg=......MVAR,sb=.... else (pu) Sb=1
Pd=busdata(:,7)/Sb;
Qd=busdata(:,8)/Sb;
Qmin=busdata(:,9)/Sb;
Qmax=busdata(:,10)/Sb;
Qinj=busdata(:,11)/Sb;
%--------------------------- (4)Nbus------------------------------------
nbus=length(nobus); % OR max(nobus)
%--------------------- (5)prepare initials Q,P,ym,ya, ------------------------
Q=Qg-Qd+Qinj;
P=Pg-Pd;
ym=abs(Ybus);
ya=angle(Ybus)*180/pi;
%--------------------- (6)prepare initials itr,error,tol------------------------
itr=0;
error=1;
tol=10^-12;
%--------------------------- (7)Start Loop ------------------------------------
 while error>tol

%--------------------------- (8)calculate jacobians------------------------------------
 for i=1:nbus
    sum2=0;
    sum4=0;
    J1(i,i)=0;
    J3(i,i)=0;
    for j=1:nbus
        if j~=i
%J1
            J1(i,i)=J1(i,i)+vm(i)*vm(j)*ym(i,j)*sind(ya(i,j)+va(j)-va(i));
            J1(i,j)=-1*vm(i)*vm(j)*ym(i,j)*sind(ya(i,j)+va(j)-va(i));
%J2
            sum2=sum2+(vm(j)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i)));
            J2(i,j)=vm(i)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i));
```

```
%J3
            J3(i,i)=J3(i,i)+(vm(i)*vm(j)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i)));
            J3(i,j)=-1*vm(i)*vm(j)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i));
%J4
            sum4=sum4+(vm(j)*ym(i,j)*sind(ya(i,j)+va(j)-va(i)));
            J4(i,j)=-1*vm(i)*ym(i,j)*sind(ya(i,j)+va(j)-va(i));
        end
    end
% finish J2 and J4
    J2(i,i)=2*vm(i)*ym(i,i)*cosd(ya(i,i))+sum2;
    J4(i,i)=-2*vm(i)*ym(i,i)*sind(ya(i,i))-sum4;
end

%-------------------------- (9) elimination of jacobian ---------------------------------
PV=find(type==1); %find position of pv bus
npv=length(PV)    % no of PV bus
sl=find(type==0);    % the find function is a MATLAB function
%J1                  % which returns the indices of elements
J1(sl,:)=[];         % that satisfy the condition (type == 1)
J1(:,sl)=[]
% J2
J2(sl ,:)=[];
J2(:,[sl PV])=[]
%J3
J3([sl PV] ,:)=[];
J3(:,sl )=[]
%J4
J4([sl PV] ,:)=[];
J4(:,[sl PV])=[];

%-------------------------- (10) mismatch calcu. dP & dQ  -------------------------------
PQ=find(type==2);
pq_pv=find(type~=0);

for i=1:nbus
        Pcal(i)=0;
        Qcal(i)=0;
        for j=1:nbus
            Pcal(i)=Pcal(i)+vm(i)*vm(j)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i));
            Qcal(i)=Qcal(i)-vm(i)*vm(j)*ym(i,j)*sind(ya(i,j)+va(j)-va(i));
        end
end
        dP=P(pq_pv)-Pcal(pq_pv)'; %(') for transpose matrix
        dQ=Q(PQ)-Qcal(PQ)';
%-------------------------- (11) output calcu  ---------------------------------
J=[J1 J2;J3 J4];
dPQ=[dP;dQ];

mismatch=inv(J)*dPQ;          %mismatch=[dva;dvm]%

pq_pv=find(type~=0);
va(pq_pv)=va(pq_pv)+mismatch(1:nbus-1)*180/pi;

PQ=find(type==2);
vm(PQ)=vm(PQ)+mismatch(nbus:2*nbus-2-npv)
%-------------------------- (12) update solver  ---------------------------------
itr=itr+1;
error=max(abs(dPQ));
end
```

```
%------------------------- (13) output  --------------------------------
vm
va
itr
J
v=vm.*cosd(va)+1i*vm.*sind(va);
```

6 POWER FLOW

$$S_i^* = V_i^*(Y_{ii}V_i + \sum_{\substack{j=1\\ j\neq i}}^{n} Y_{ij}V_j), \quad i = 1,2,3,\ldots\ldots,n$$

$$P_i = \text{real}\,(S_i^*)$$
$$Q_i = \text{imag}\,(S_i^*)$$

$$P_{gi} = P_i + P_{di}$$
$$Q_{gi} = Q_i + Q_{di}$$

$$S_{ij} = V_i\,[Y_{ij}(V_j - V_i)]^*$$

$$S_{ji} = V_j\,[Y_{ji}\big(V_i - V_j\big)]^*$$

$$S_{jiLoss} = S_{ij} + S_{ji}$$

```
 % (1) calculate stl

for i=1:nbus
    for j=1:nbus
        if i~=j
      s(i,j)=v(i)*conj(Ybus(i,j)*(v(j)-v(i)));
      s(j,i)=v(j)*conj(Ybus(j,i)*(v(i)-v(j)));
        end
    end
end
stl=sparse(s); %eliminates zeros
disp(stl) ;

% (2) calculate sloss

for i=1:nbus
    for j=1:nbus
        if i~=j

     sloss(i,j)=s(i,j)+s(j,i);

       end
    end
end

sloss =sparse(sloss); %eliminates zeros
disp(sloss) ;


% (3) calculate p,Q for pv ,slack buses
for i=1:nbus
    if type(i)~=2 %no pQ bus
        sum=0;
        for j=1:nbus
            if j~=i
                sum=sum+v(j)*Ybus(i,j);
        end
      end

       s(n)=conj(conj(v(i))*(v(i)*Ybus(i,i))+sum);

        P(i)=real(s(i));
        Q(i)=imag(s(i));

        if type(i)==0 %slack bus
        Pg(i)=P(i)+Pd(i);
        Qg(i)=Q(i)+Qd(i)-Qinj(i);  %Qinj= 0
      end

      if type(n)==1 %PV bus
        Qg(i)=Q(i)+Qd(i)-Qinj(i);
      end

    end
end
sloss
Q
Pg
Qg
ploss=real(sloss)
```

7 FAST DECOUPLED

When TL is $X \gg R \;\; or \; \frac{X}{R} \gg 1 \quad long\ TL$

It can be quick & accurate in the same time by decreasing tolerance get higher accuracy

NR is used with protection because it's accurate
using high speed processors

Reactive Power is a function of **Voltage**

$$\Delta Q \Rightarrow f(\Delta|V|)$$

We improve PF by using capacitor to inject Q , which improve V

EX:When we have voltage drop in end we use capacitors (Q)

Active Power is a function of **Power Angle**

$$\Delta P \Rightarrow f(\Delta\delta)$$

EX:When loading increases (P) then the angle δ also increase

What About Cables ? Does it have the same relation ?(X>>R)

In fast decoupled we use only J1 and J4

$$J = \begin{bmatrix} \frac{\partial P}{\partial \delta} & 0 \\ 0 & \frac{\partial Q}{\partial V} \end{bmatrix} \; as \;\; X \gg R$$

Quicker but (slightly) less accuracy (half calculation are done)

For same system – to get to same accuracy – tol

$$Iterations\ of\ FD\ > Iterations\ of\ NR$$

- For large system performance defers from small systems

7.1 Applications

7.1.1 Power Flow analysis

Quick ,Some Accurate method

7.1.2 Control

Because it's fast , and control systems needs quick action
to increase accuracy – decrease tolerance

7.2 Mathematical Approach.

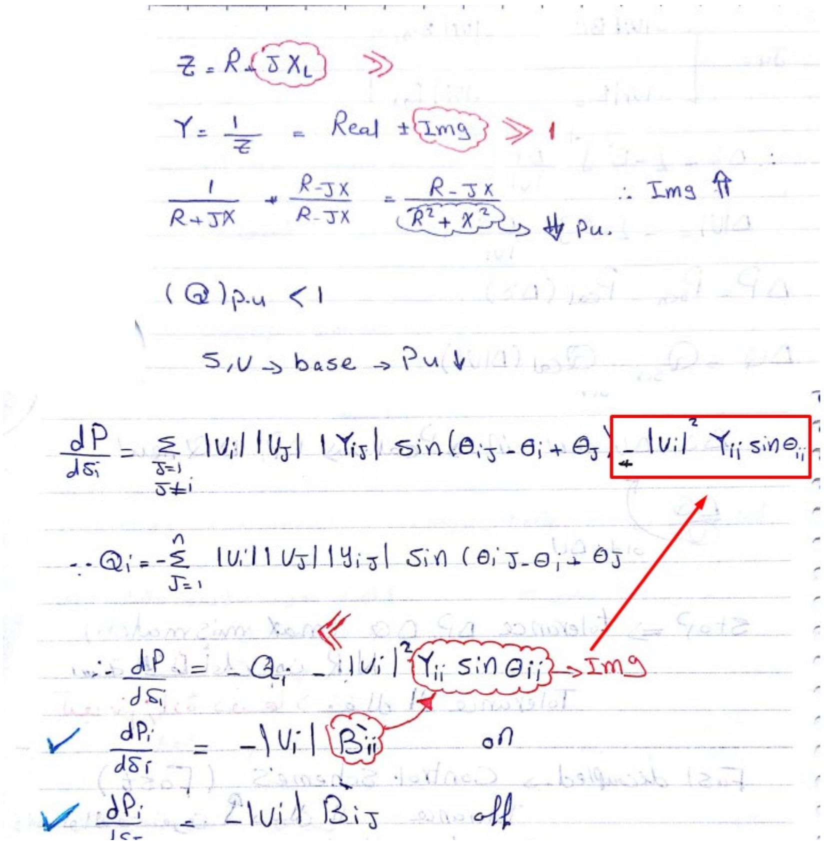
$$Y_{bus} \rightarrow Real + Img \quad (img \gg)$$

Proof, then we find

$$B_{ii} = img\ part\ of\ Y_{bus}\ (on - diag)$$
$$B_{ij} = img\ part\ of\ Y_{bus}\ (off - diag)$$

$$\frac{\partial P_i}{\partial \delta_i} = -|v_i|B_{ii} \qquad \frac{\partial P_i}{\partial \delta_j} = -|v_i|B_{ij}$$

$$\frac{\partial Q_i}{\partial \delta_i} = -|v_i|B_{ii} \qquad \frac{\partial Q_i}{\partial \delta_j} = -|v_i|B_{ij}$$



$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial P}{\partial \delta} & 0 \\ 0 & \frac{\partial Q}{\partial V} \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}$$

$$\Delta P = \frac{\partial P}{\partial \delta} * \Delta \delta$$

$$\Delta Q = \frac{\partial Q}{\partial V} * \Delta |V|$$

NOTE :

$$B_{II}, B_I\ are\ const$$

$$V_i : changes\ from\ Bus\ to\ Bus\ , and\ from\ iteration\ to\ iteration$$

to get B_{II}

1- شيل صف وعمود بتوع الslack

2- المصفوفه المتبقيه خد سالب التخيلي بتاعه

3- مبروك كده انت جبت B_{II}

$$[\Delta\delta] = [\ B_{II}]^{-1} \left[\frac{\Delta P}{V_i}\right]$$

$$[\Delta V] = [B_I]^{-1} \left[\frac{\Delta Q}{V_i}\right]$$

$$B_I = -img\ [Y_{BUS}]\ \ after\ removing\ column\ and\ row\ of\ slack\ and\ PV\ buses$$

In program miss match matrix is divided by Vi $\left[\frac{\Delta P}{V_i}\right]$, $\left[\frac{\Delta Q}{V_i}\right]$

7.3 STOP

- Using tolerance ΔP , ΔQ max mismatch
- Error will be larger than NR but decreasing Tol may increase accuracy

7.4 CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%% Eng_Mohamed Numair %%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear all

%% (1) GET YBUS
Ybus=[8.985-j*44.836  -3.816+j*19.078   -5.170+1i*25.848 0
      -3.816+j*19.078   8.985-j*44.836   0  -5.170+j*25.848
      -5.170+j*25.848  0  8.193-j*40.864  -3.024+j*15.119
       0  -5.170+j*25.848  -3.024+j*15.119  8.193-j*40.864];
%OR Ybus=input('enter Ybus');

 %% (2) GET Bus DATA
%-------------------------- Bus data ------------------------------------
%      [bus   |bus | vm |    va    | Pg   | Qg   |   Pd | Qd | Qmin | Qmax | Qinj]
%       no.   |type|  pu |degree|  pu   |   pu |   pu | pu    |pu   |pu

busdata=[1  0 1 0   0 0    .5 .3 0 0 0 ;
         2  2 1 0   0 0    1.7 1.0535 0 0 0 ;
         3  2 1 0   0 0    2 1.2394 0 0 0;
         4  1 1.02 0 3.18 0 .8 .4958 0 0 0];
% OR busdata=input('enter Bus data');
 %% (3) Columning
Sb=1;
nobus=busdata(:,1);
type=busdata(:,2);
vm=busdata(:,3);
va=busdata(:,4);
Pg=busdata(:,5)/Sb;         % if Pg=......MW else(pu) Sb=1;
Qg=busdata(:,6)/Sb;         % if Pg=......MVAR else (pu) Sb=1
Pd=busdata(:,7)/Sb;
Qd=busdata(:,8)/Sb;
Qmin=busdata(:,9)/Sb;
Qmax=busdata(:,10)/Sb;
Qinj=busdata(:,11)/Sb;
nbus=length(nobus); % OR max(nobus)
 %% (4) Getting Q,P,B
Q=Qg-Qd+Qinj;
P=Pg-Pd;
ybusm=abs(Ybus);
ybusa=angle(Ybus)*180/pi;
B=-imag(Ybus);
 %% (5) calculate no.of PV bus
PV=find(type==1); %find position of pv bus
nv=length(PV);
 %% (6) Elimination in B
%____________________________________elimination
________________________________________%
sl=find(type==0);
B1=B;
B4=B;
% ______________________________________elimination of
B1_____________________________________________%
B1(sl,:)=[]; % this is BII
B1(:,sl)=[]
% ______________________________________elimination of
B4_____________________________________________%
B4(:,[sl PV])=[]; % this is BI
B4([sl PV] ,:)=[]
```

```
%%%%%%%%%%%%%%%%%%%%%%%%FDPF%%%%%%%%%%%%%%%%%%%%%%
 %% (7) Start Loop(itr,error,tol)
itr=0;
error=1;
tol=10^-12;

while error>tol

%%%%%%%%%%%%%%%% dp&dQ calculation %%%%%%%%%%%%%%%%

%% (8) dP (dP/Vm) calculation
r=0;
for i=1:nbus
    if type(i)~=0 % no slack
        r=r+1;
        Pcal=0;
        for j=1:nbus
            Pcal=Pcal+vm(i)*vm(j)*ybusm(i,j)*cosd(ybusa(i,j)+va(i)-va(j));
        end
        dP(r)=(P(i)-Pcal)/vm(i);
    end
end

%% (9) dQ (dQ/Vm) calculation
        w=0;
        for i=1:nbus
            if type(i)==2 % PQ bus
                w=w+1;
                Qcal=0;
               for j=1:nbus
                  Qcal=Qcal-vm(i)*vm(j)*ybusm(i,j)*sind(ybusa(i,j)+va(j)-va(i));
               end
                dQ(w)=(Q(i)-Qcal)/vm(i);
            end
        end



        %%%%%%%%% dva & dvm calculation %%%%%%%%%%%%

        %% (10) dva & dvm calculation
  dvm=inv(B4)*dQ';
  dva=inv(B1)*dP';              %(') for transpose matrix

        %% (11) update angle
 pv_pq=find(type~=0) ; % no slack
 va(pv_pq)=va(pv_pq)+dva*180/pi;
        %% (12) update magnitude
 pq=find(type==2)  % PQ bus
vm(pq)=vm(pq)+dvm;
        %% (13) update solver
itr=itr+1;
error=max(max(abs(dP)),max(abs(dQ)));

end
        %% (14) outputting
vm
va
itr
v=vm.*cosd(va)+1i*vm.*sind(va);

%%
```

```
%%%%%%%%%%%%%%%%%% POWER FLOW %%%%%%%%%%%%%%%%%%%%
for n=1:nbus
    for k=1:nbus
        if n~=k
      s(n,k)=v(n)*conj(Ybus(n,k)*(v(k)-v(n)));
      s(k,n)=v(k)*conj(Ybus(n,k)*(v(n)-v(k)));
      sloss(n,k)=s(n,k)+s(k,n);
        end
    end
end
% p,Q for pv ,slack buses
for n=1:nbus
    if type(n)~=2 %no pQ bus
        sum=0;
        for k=1:nbus
                sum=sum+v(k)*Ybus(n,k);
        end
       s(n)=conj(conj(v(n))*sum);
        Q(n)=imag(s(n));
        Qg(n)=Q(n)+Qd(n)-Qinj(n);
        if type(n)==0 %slack bus
        P(n)=real(s(n));
        Pg(n)=P(n)+Pd(n);
        end
    end
end
sloss
Q
Pg
Qg
ploss=real(sloss)
```

8 LOAD FLOW ANALYSIS QUICK

8.1 YBUS

$$\boldsymbol{where} \quad \overbrace{Z_{bus}}^{\text{Impedance Matix}} = \boldsymbol{inverse\ of} \quad \overbrace{Y_{bus}}^{\text{Admittance Matix}}$$

$$S = V_i^* \sum_{k=1}^{n} Y_{ik} V_k$$

$$P_i = |V_i| \sum |V_k||Y_{ik}| \cos(\theta_{ik} + \delta_k - \delta_i)$$

$$Q_i = -|V_i| \sum |V_k||Y_{ik}| \sin(\theta_{ik} + \delta_k - \delta_i)$$

8.1.1 Notes of Ybus and Zbus

- Ybus can have many zeros in off-diagonal , system is large
- Ybus on diagonal can't be zero
- Zbus is the inverse of Ybus , but if Ybus 10*10 , can u get it ?!
- In symmetrical system we study only one case and can be applied to others
- Zbus – Zth so is it only used for symmetrical (3ph faults) ??

  NO, IT CAN BE MODIFIED to be used in unsymmetrical faults
- We use Zbus to calulate SS fault current

8.2 GAUS SEIDEL STEPS

(1) Calculate Bus admittance matrix

(2) Assume initial values of voltages and angles for unknown buses (V = 1 , $\delta = 0$)

(3) Calculate initial value of P and Q for unknown buses (convert in pu)

(4) Do the first iteration and calculate voltages and angles at all buses

(5) $V_i = \frac{1}{Y_{ii}} \left[\frac{P_i - jQ_i}{V_i^*} - \sum_{\left[\substack{k=1 \\ k \neq i}\right]}^{n} Y_{ik} V_k \right]$

(6) Deduce the value of P and Q that pass through T.L

(7) Computation of slack bus power and losses

(8) Computation of line flows

$$\Rightarrow \boldsymbol{S_{12}^{*} = V_1^{*}\,(V_1 - V_2)(-Y_{12})}\ \ \textbf{then get}\ \boldsymbol{S_{12}}$$

$$\Rightarrow \boldsymbol{S_{21}^{*} = V_2^{*}\,(V_2 - V_1)(-Y_{12})}\ \ \textbf{then get}\ \boldsymbol{S_{21}}$$

$$\boldsymbol{S_{loss} = S_{ik} + S_{ki}}$$

اذا وجد PV أي voltage controlled bus نحلها الأول ثم بعد كده نحل الPQ بالطريقة المشروحة أعلاه

لايجاد ال Q

$$\boldsymbol{Q = -im\left\{V_i^{*}\sum Y_{ik}V_k\right\}}$$

ولايجاد ال δ

$$\boldsymbol{\delta_i = Angle\ of}\left[\frac{1}{Y_{ii}}\left[\frac{P_i - jQ_i}{(V_i^{r})^{*}} - \sum_{\begin{bmatrix}k=1\\k\neq i\end{bmatrix}}^{n} Y_{ik}V_k^{r+1}\right]\right]$$

8.3 Newton Raphson

Advantages :

- Faster -Sure to converge

Disadvantages

- Need for large computer memory

(1) Convert all to per unit calculations <u>on same base</u>

(2) Obtain the Y_{BUS}

(3) Assume the initial values as

- $V = 1\angle 0 \quad P.Q\ bus$
- $\delta = 0 \qquad P,V\ bus$

(4) Miss match matrix

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} P_{sch(net)} - P_{cal} \\ Q_{sch(net)} - Q_{cal} \end{bmatrix} = \begin{bmatrix} P_{gen} - P_{dem} - P_{cal} \\ Q_{gen} - Q_{dem} - Q_{cal} \end{bmatrix}$$

$P_{sch}\ (given\ ) if\ not = 0$
$P\ and\ Q\ calc\ we\ calculate\ it\ in\ the\ problem\ and\ compare$

(5) (Δ calculated then added to initial values)

$$\begin{bmatrix} \Delta\delta \\ \Delta v \end{bmatrix} = \begin{bmatrix} \left[\frac{\partial P}{\partial \delta}\right] & \left[\frac{\partial P}{\partial v}\right] \\ \left[\frac{\partial Q}{\partial \delta}\right] & \left[\frac{\partial Q}{\partial v}\right] \end{bmatrix}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$$

8.4 Fast Decoupled

<u>**to get B_{II}**</u>

4- شيل صف وعمود بتوع الslack

5- المصفوفه المتبقيه خد سالب التخيلي بتاعه

6- مبروك كده انت جبت B_{II}

$$[\Delta\boldsymbol{\delta}] = [\ \boldsymbol{B_{II}}]^{-1} \left[\frac{\Delta \boldsymbol{P}}{\boldsymbol{V_i}}\right]$$

$$[\Delta\boldsymbol{V}] = [\boldsymbol{B_I}]^{-1} \left[\frac{\Delta \boldsymbol{Q}}{\boldsymbol{v_i}}\right]$$

$\boldsymbol{B_I = -img\ [Y_{BUS}]\ \ after\ removing\ column\ and\ row\ of\ slack\ and\ PV\ buses}$

9 FAULT ANALYSIS

Balanced: 3ϕ or $3\phi - G$

$$I_f = \frac{E_{th}}{Z_{th} + Z_f}$$

Unbalanced: $L - G\ (common)$ or $L - L$ or $L - L - G$

L-G (i.e. a-g)	L-L (i.e. b-c)	L-L-G
$I_b = I_c = 0$ $I_a = 3I_{a_1}$ $V_{a1} + V_{a2} + V_{a0} = 3\,Z_f I_{a1}$ $I_f = 3I_{a1} = \frac{3E_a}{Z_1 + Z_2 + Z_3 + 3Z_f}$	$I_a = 0$ $I_b = -I_c$ $3(V_{a1} - V_{a2}) = j\sqrt{3}\,Z_f(I_b)$ $= j\sqrt{3}\,Z_f\left(-j\sqrt{3}\,I_{a1}\right)$ $I_b = -I_c = \frac{E_a}{Z_1 + Z_2 + Z_f}$	$i_f = I_b + I_c = 3I_{a_0}$ $I_{a1} = \frac{E_a}{Z_1 + \left(Z_2 \parallel \left(Z_0 + 3Z_f\right)\right)}$ $I_{a_0} = -I_{a1} * \frac{Z_2}{Z_1 + Z_2 + 3Z_f}$ $I_f = 3I_{ao}$

9.1.1 What are fault causes ?

1- Due to lightning

2- Tree limbs falling on line

3- Wind damage

4- Vandalism

5- Insulation deterioration

9.1.2 What Happens during Fault ?

(1) Flow of excessive current as $I = \frac{V}{Z}$ and during fault Z decreases so current increases excessively

(2) Abnormal voltage

(3) Voltage elevation of system neutral

(4) Induced overvoltages on neighbouring equipments

(5) Hazards to human, equipment and animals

9.1.3 Why do we do fault analysis ?

1- The information obtained from fault studies is used for (rating of protection devices)

a. Selecting the size of C.B

b. Fuse characteristic

c. Setting of relay

2- Setting for protection devices (Time , Value) [time constant of devices]

3- Determining the capability of system to withstand the fault

4- Avreage and RMS values of quantities (V,I) during dault and it's waveform

9.1.4 What does fault current value depend on ?

On impedance

9.1.5 What does impedance depend on ?

(1) Impedance of network

(2) Internal impedance of generator

(3) Resistance of fault (arc resistance)

Network impedance is governed by :

1- Transmission line impedance

2- Transformer connection and impedances

3- Grounding connections and resistance

9.1.6 C.B. vs Disconnector

Circuit Breaker	**Disconnector**
Has a box to damp the arcing as (SF6)	Doesn't have a box

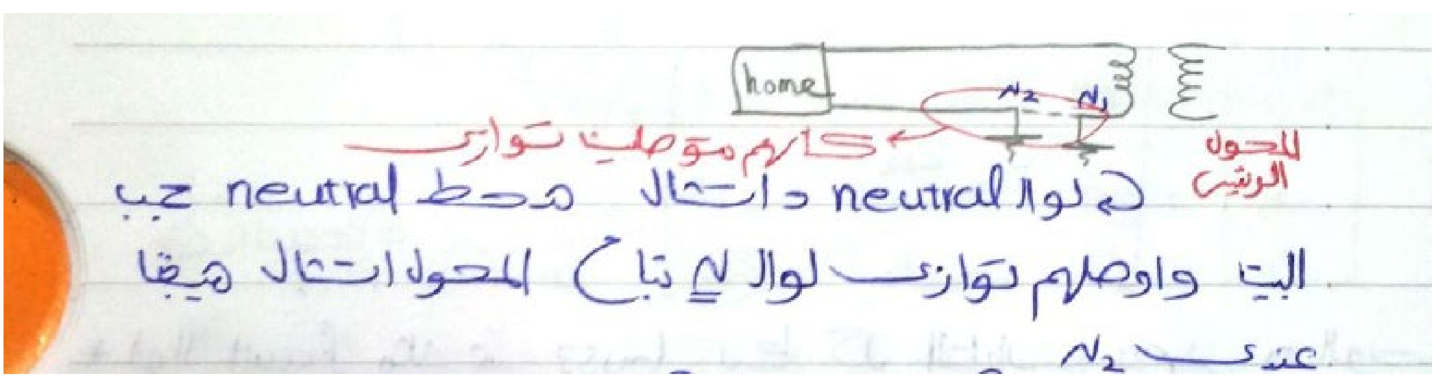
9.1.7 What do we do in Fault Analysis ?

1- Study the sys (Before- During – After)
2- Determine the location of fault
3- Determine the values of fault currents

9.1.8 If normal voltage 220 and it was seen as 280 V? what's the cause?

$L-G$ or $L-L-Gfault$ causes this problem as the neutral point is removed

And voltage has become <u>floating</u> and has no reference



9.1.9 What's the voltage profile of unfaulted and faulted phases in case of faults ?

For L-G fault (a-g)

Then $V_a = 0\ , but\ V_b and\ V_c\ will\ increase\ as\ neutral\ is\ moved$

But current of unfaulted phases = 0

9.1.10 How to tell if it is L-L-G or a L-G

10 ZBUS

$$where \quad \overbrace{Z_{bus}}^{\text{Impedance Matix}} \quad = inverse\ of \quad \overbrace{Y_{bus}}^{\text{Admittance Matix}}$$

Symmetric Ybus lead to **symmetric** Zbus

The **on-diagonal** elements of Zbus are called **driving point impedances** of the nodes,(**are Thevenin Imp.)**

Self impdances of this bus and other connected bus + earth impedance

$$Z_{ii} = Z_{io} + \sum_{i \neq j} Z_{ij}$$

the **off-diagonal** elements are called **transfer impedances** of nodes. Doesn't represent actual impedance in the actual system, only used in problems

Zero elements of Ybus become **non-zero** in the corresponding Zbus elements.

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}$$

NOTE: FAULT MUST OCCUR ON ONE OF THE BUSES

When fault occurs we put it's value with $I_f = I_3$ and rest are neglected =0

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta V_3 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ I_3 = -I_f \end{bmatrix}$$

Pre-fault

[V] are the prefault and calculated from LOAD FLOW

$$I_f = \frac{V}{Z_{11} + Z_f}$$

During fault

$[\Delta V]$ calculated to determine increase in voltage then to be added to prefault condition to get after fault

10.1.1 To calculate current between bus 1 and 2 (TL)

$$I_{12} = \frac{V_1 - V_2}{Z_{12}}$$

NOTE

In large system Ybus has many zero elements which are not zero in Zbus

We can get Zbus from inverse Ybus or from building algorith

10.2 Solution Steps

1- Convert all to PU

2- Get Y_{Bus}

3- $get\ \ Z_{bus} = \ [Y_{Bus}]^{-1}$ (always +ve) ($[J]^{-1} \ = \ -J$)

4- Note : Fault must happen at one of the buses (if not add bus at point of fault)

5- $i_{f_i} = \frac{E_{th}}{Z_{th_{ii}}}$

6- $V_{afterfault} = V_{pre} + \Delta V$

7- $[\Delta V] = [Z_{Bus}]\left[-i_f\right]$

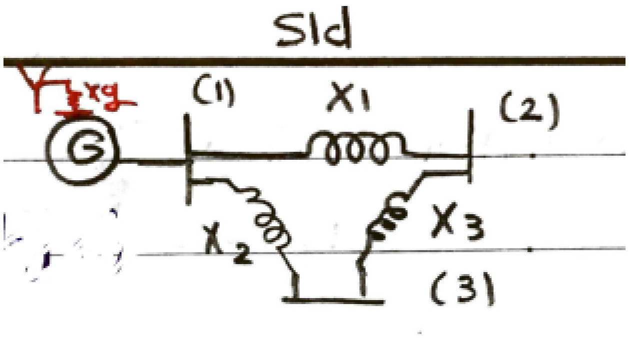
8- $NOTE$: put i_f -ve at the bus of fault

9- $[V] = \left[V_{pre}\right] + \ [Z_{Bus}]\left[-i_f\right]$

NOTE when calculate by hand j*j = -1

10.3 Z_{bus} Building

For the following Network



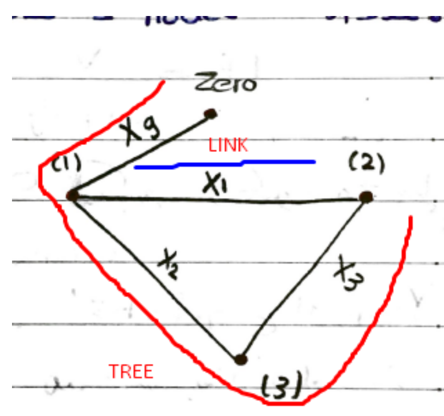
Each bus will be replaced by a <u>node</u>

Taking (G , Tr, TL) Impedances

10.4 Definitions

- Node
- Branch
- Link : a TL not included in Tree
- Tree : Open Loop contains all nodes only once (0132 or 0123 or 2310)
- Co-Tree

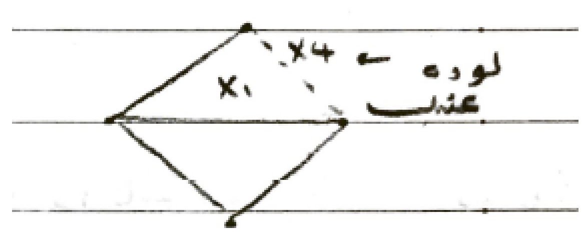
10.4.1 TREE



Take Tree (0132) we see we didn't pass by X1

So we call X1 a <u>Link</u>

10.4.2 CO-TREE



Contains all <u>links !!! (links here X1,X4)</u>

Like if we have a 4 bus system with tree (0132)

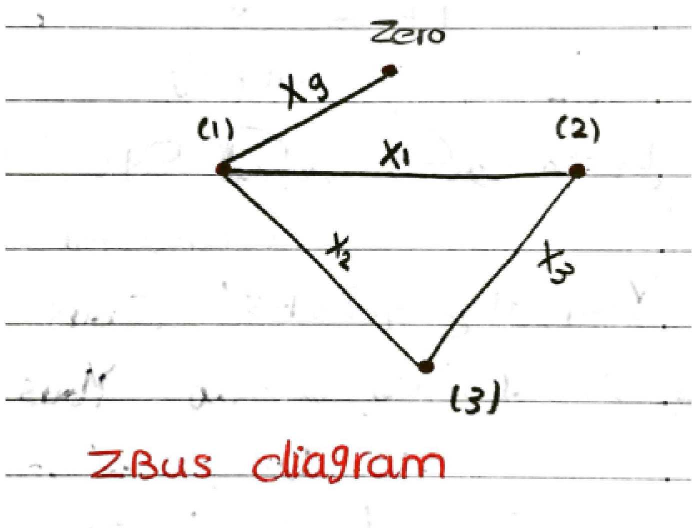
A co tree is x1,x4

10.5 Steps

Step 1 : Nodes

We add node (Zero)(ground) (reference)

$$N_{nodes} = N_{buses} + 1$$



STEP 2

All TL are replaced by Lines (in tree)

$$N_{lines} = N_{impedances}$$

Step 3 : Detemine Tree

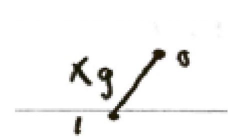
Tree (0123)

So → $Branches\ Number = 3\ :(0-1)X_g\ \ \&\ \ (1-2)X_2\ \ \&\ (2-3)X_3$

and→ $Links\ Number = 1 : X_2$

STEP 4 : Add Node (1) to Reference (0)

$$Z_{BUS} = \begin{bmatrix} X_g & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



STEP 5 : Add Node (2) to Node (1)

$$\boldsymbol{Z_{bus}} = \begin{bmatrix} X_g & X_g & 0 \\ X_g & X_g + X_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

STEP 6 : add Node (3) to Node (2)

$$Z_{bus} = \begin{bmatrix} X_g & X_g & X_g \\ X_g & X_g + X_1 & X_g + X_1 \\ X_g & X_g + X_1 & X_g + X_1 + X_3 \end{bmatrix}$$

NO NO NO ! You did not finish yet

Link X2 is crying you left it out !!!

STEP 7 : adding the link (3-1)

$$[\boldsymbol{Z_{new}}] = [\boldsymbol{Z_{old}}] - \frac{\mathbf{1}}{\boldsymbol{Z_{LL}}}[\boldsymbol{Z_{J-K}}][\boldsymbol{Z_{J-K}}]^{\boldsymbol{T}}$$

$$\boldsymbol{as}\ [\boldsymbol{Z_{j-K}}] = [\boldsymbol{Z_{3-1}}] = \begin{bmatrix} X_g \\ X_g + X_1 \\ X_g + X_1 + X_3 \end{bmatrix} - \begin{bmatrix} X_g \\ X_g \\ X_g \end{bmatrix} = \begin{bmatrix} X_g - X_g \\ X_g + X_1 - X_g \\ X_g + X_1 + X_3 - X_g \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{X_1} \\ \boldsymbol{X_1} + \boldsymbol{X_3} \end{bmatrix}$$

$$[\boldsymbol{Z_{j-K}}]^{\boldsymbol{T}} = [\mathbf{0} \quad \boldsymbol{X_1} \quad \boldsymbol{X_1} + \boldsymbol{X_3}]$$

$$\boldsymbol{Z_{LL}} = \boldsymbol{Z_{KK}} + \boldsymbol{Z_{JJ}} - \mathbf{2}\boldsymbol{Z_{JK}} + \boldsymbol{Z_b}$$

$$\boldsymbol{Z_{LL}} = X_g + X_g + X_1 + X_3 - \mathbf{2}X_g + \boldsymbol{X_2} = \boldsymbol{X_1} + \boldsymbol{X_2} + \boldsymbol{X_3}$$

$$[\boldsymbol{Z_{new}}] = \begin{bmatrix} X_g & X_g & X_g \\ X_g & X_g + X_1 & X_g + X_1 \\ X_g & X_g + X_1 & X_g + X_1 + X_3 \end{bmatrix} - \frac{\mathbf{1}}{\boldsymbol{X_1} + \boldsymbol{X_2} + \boldsymbol{X_3}} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{X_1} \\ \boldsymbol{X_1} + \boldsymbol{X_3} \end{bmatrix} [\mathbf{0} \quad \boldsymbol{X_1} \quad \boldsymbol{X_1} + \boldsymbol{X_3}]$$

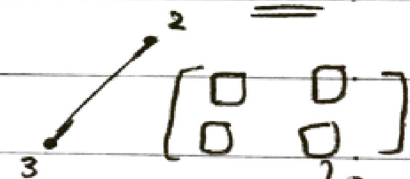
NOTES :

1- According to your path the Z before link is different but the final is same

2-We start and refer to node (0)

3-

note



دخلنا من الحسابات دي node(3) سبب فيه mismatch بينا وبينا

المكان ال هتحط فيه

Zthevenin at bus 2

⇐ إزاي نعمل Fault analysis بال Zbus

11 GUI PROGRAM
"GRAPHICAL USER INTERFACE"

11.1 Programs used as

1- Simulation with blocks

2- Matlab package

3- C

4- PSCAD

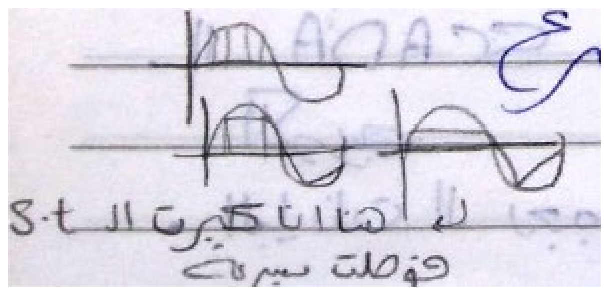
5- ETAP

6- EMPT

11.2 Program Choice upon

1- Easy Interface

2- Block library contains any choices , has components I need



3- Transient Calculations are the most complex

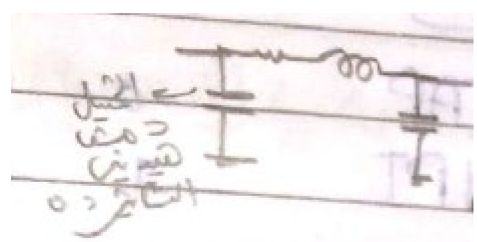
4- Compare between programs in S.S. output

5- Step time of process

6- Solving method , not all programs allow you to use the solving method you want

7- Components model and parameter's are available

8- Inductance between phases can be calculated by some programs if you inter **diameter**



9- Conditions that the program apply simulation in (time , temperature , etc)

10- Can be interfaced with a circuit (HMI)

11- Updated control library

11.3 GUI Advantages

1- Easy to deal with as it uses graphical interface

2- Easy to model any system even if it is complicated using blocks

3- Easy Error detection and any abnormal operation condition

4- Has a library containing blocks, components and tasks

5- The user doesn't have to use a programming language

11.4 SCADA "Supervision Control and Data Acquisition"

It's the most important GUI example in control systems we use SCADA system to monitor and control

We collect data in Power Systems to SCADA using **RTU** "Remote Terminal Unit"

11.5 EMS "Energy Management System"

11.5.1 Used for

1- High Efficiency

2- Minimum Cost

11.5.2 Used when

We have <u>many multiple</u> energy sources, and a load , use which first to get the highest efficiency in the source side , and least cost

11.5.3 Works as follows

It studies nature of sources as life time, starting time , cost of operation and starting then manageing them together to achieve

(1) Longer Life time (i.e. for Batteries)

(2) Least Maintainance Cost

(3) Least degradation and depreciation for components

11.5.4 Applications of EMS

1- Electric Vehicle

2- Micro-grids

3- Large Buildings

11.6 GUI vs C++ ,Fortran

GUI	C++, Fortran
آسهل وأبسط فى الإستخدام . easy . simple For user	—
—	. can modified.
. Library component	. equivalent circuit model
الدقة تقريباً مثل بعض ولكن تختلف فى طريقة الحل لا يعرف طريقة الحل وبالتالى الدقة تعتمد على خصائص البرنامج Prop. reties	طريقة الحل معروفة
تقدر تمثل خصائص مجهولة characteristic (المعادلة)	
. لها مشكلة فى معرفة ال error	
sample Time :- يشوف ال system كل اد ايه يفضل يكون مقبول ولكن run time يزيد	
إمكانية التحكم فى sample time	—

أسطرع د- PI لازم يكونو قريبين من بعض ضيفه أن يكون مش شايف عاصم system سريع أوي PI control

ماهو دور الحاسب الآلى فى 12 MG

• Energy management system.

برنامج لإدارة الطاقة.

تحدد حالة charge/discharge rate

فى حالة تعدد المصادر (PV/wind/--) السيستم بيتحكم

- التعريفة افضل أستخدم. امته نبيع للشبكة.

- تنسيق الادوار بين Sources / Battery

- التشغيل عند أقصى بأقل سعر لكل ال system.

دوره اقوى فى حالة grid connected لانه بينسق مع الشبكة

الامكانات متواجد للأفضل على حسب المنظومة.

4. Transient --

[2] smart grid

* مثل ال micro ولكن فى Communication بين المصادر system

المختلفة حيث تكون كل component ترى الاخرى

سواء wire أو wireless

11.7 Advanced Application of computer in PS

ADVANCED Application of Computer in PS.

- Micro grid (M.G)
- smart grid (SG)
- smart micr grid (SMG)
- Nano grid
- IoT (Internet of Thing)

١- لا تختلف عن الشبكة إلا إنها حجم أصغر، في المناطق النائية Isolated
تتكون من مصادر طاقة متجددة. إما أن تكون
او متصلة بالشبكة. لمصدر واحد او أكثر.
- لابد من وجود storage component

1- Renewable Energy

2- Power Electronic
- DC to AC
- level

3. Storage System
- Battery
- Capacitive.

Isolated — Grid-Connected.

Isolated

* Storage 100%.
↳ supply Energy To Load.

$P_{G(RE)} + P_{(storage)} = P_{load}$

at night $P_{storg} = P_{load}$

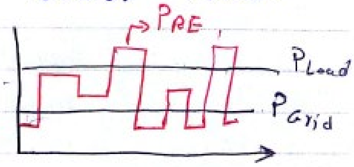
• يحتاج إلى بطاريات كبيرة جداً (تكلفة)

- صعب تشغل بطاريات قديمة مع جديدة

Grid-Connected

* storage

$P_{G(RE)} + P_{G(storage)} + P_{G(grid)} = P_{Load}$.



- البطارية مصدر أساسي للطاقة

• احيانا المجموع أقل أو أكثر من الحمل عن طريق إستخدام بطارية تشحن في وقت الزيادة وتفرغ في وقت النقصان

- البطارية مصدر لعمل إخماد للـ oscillation

• Flucted Power Source

11.8 IOT

Using a mobile application we can connec multiple devices

- Wide area control
- Optimization
- Artificial Intelligence
- SCADA

11.9 KEYWORDS

- Compare between Wide Area Control & SCADA
- Microcontroller – Advantages – Disadvantages
- Control type choosing criteria
- Online – off Line
- Design – Operation
- SCADA data is synced or not

11.10 WIDE AREA CONTROL Vs SCADA

- مكونات wide areal control
- مقارنة بين wide و scada

local areaControl vs wide area control

مساحات صغيرة | دخل مدى
limitted space | عند الوقت instantaneous
| قيمة معينة rms
| للتحكم في نقطة معينة

t = ... أماكن مختلفة
v = instantens
v = value
v = rms

satellite ← لعمل control من مسافات بعيدة عن طريق خطوط نقل

← في مصر scada (مركز التحكم القومي بالقاهرة).

(I)(V) ← يتم قياس
P , Q من الحاجات

Pmu خاص جداً أين نضع Pmu أفضل مكان نضع فيه
بناء على دراسة وهي نحقق أكبر قدر من المعلومات

من خلال الحسابات نقدر نجيب النقط اللي نشغلها

اهم حاجة فيه الحصول على إشارات عند نفس اللحظة.

- عيق ايه تيلس؟!
بنبدأ بالإحساس بالمشكلة
الوصول لحل المشكلة

. ليه Fuzzy أسرع من PI , PID ؟

Parameter K_P , K_I , K_I ثابتة
من حسب ذا عايز ايه وبتدور في one point.
لو مفتاح الكنترول وعملت perturbations
مختلفة مش هتعرف تعدل K

K_I K_P حسابات

Protection

مشكلتي

↓

Modeling?

Control

Adaptive PI controller
متغير K_I , K_P

Adaptive ⟸ Fuzzy

ANN

Neural برنامج (نفس البرنامج ممكن يشتغل في مجالات مختلفة.

hidden
input layer ⟹ عددهم؟! (i/p, o/p) Function
e° , cos , sin , exp hiders

multi input ← multi output
data دقيقة

كل ما data زادت
كل ما نتوقع ظروف
أكبر كل ما كان
operating condition ع مشاكل
مواضيع أكثر

→ Training

→

input	output
	=
معلوم	

نبدأ نضبط الـ Weights
عشان نقلل الـ error

B. E

11.11 OPTIMIZATION EXAMPLES

11.11.1 Optimization Techique

optimization technique.

الفرق بينهم.

. UPS
. Battery

مش كل بطارية ups ولا العكس

ups ← جهاز طوارئ ثم لحظة الفصل يغذي (الطوارئ)

↲ sizing للـ الطوارئ لوقت قدايش (أطول وقت متوقع).

مبدأ الطوارئ (الديزل تقدر تقول عليه ups / ملفات / ملحقات)

11.11.2 Cross section area of a conductor

• objective :- optimall cross section Area of Conductor with minimum cost.

• Cost → Fixed "C_f"
 → Runing "C_r"

$C_f = K_1 \, a$

$C_r = K_2 / a$

$C_T = D \, K_1 \, a + \frac{K_2}{a}$

D → فائدة

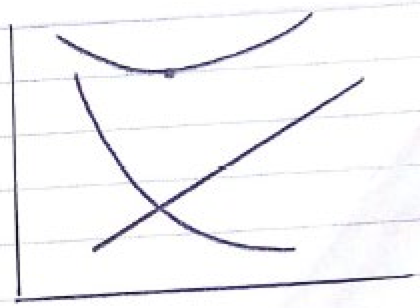
In Put: K , a, D — Process: C_f , C_r , C_T, Draw — out Put: optimall a.

min → Index

Index يعمل مقارنة لـ

ليعطي c , a

Find (   )



11.11.3 Economical Dispatch

Economic dispatch

- operating cost $C = \alpha + \beta P + \gamma P^2$

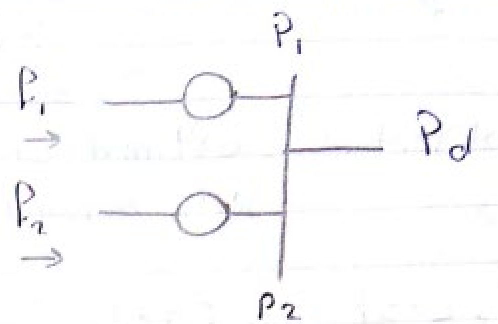
- objective: out put generated power of each Gen. That achieve minimum cost.

Date

$F_T = F_1 + F_2$

$P_d = P_1 + P_2$

$$\frac{dF_1}{dP_1} = \frac{dF_2}{dP_2} = \lambda$$



$$P_i = \frac{\lambda - \beta_i}{2\gamma_i}$$

$$\lambda = \left[P_d + \Sigma \frac{\beta_i}{2\gamma_i}\right] / \Sigma \frac{1}{2\gamma_i}$$

التكرار على حسب عدد ال Generator

لو G مختلفة limit هتختلف رقم ال Gen. هو رقم (β_i, γ_i, P_i)

Flow chart in PdF.

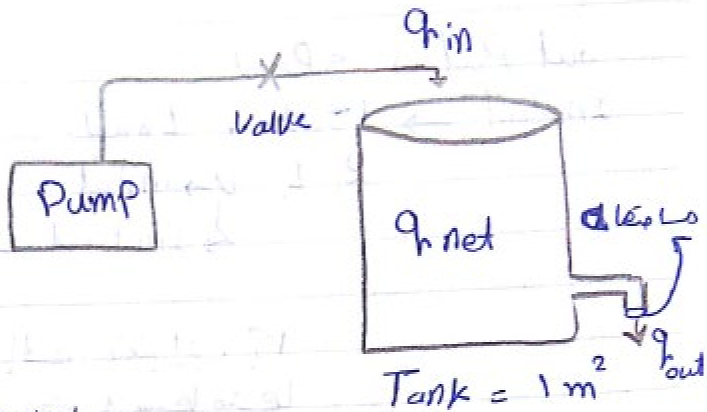
E-mail: dmyehia @ ....

11.12 Fuzzy and Micro Controller

Control by fuzzy.

$q_{net} = q_{in} - q_{out}$.

$h(t) \rightarrow$ variable.



In Put: a , $q_{initial}$

out Put: q_{out}.

$q_{out} = a\sqrt{2\, g\, h(t)}$

Fuzzy $\Rightarrow$ In Put بيشوف آلترمن Addactive
PI $\rightarrow$ k.p لو نفس Addactive بيفي أفضل وذلك
مش ثوابت

control signal $\Longrightarrow$ Selonid "interface $\rightarrow$ Valve.

Report (4)

micro controller in Impeded system.
ل الأنظمة الروبوتية (نضع micro (أو أجهزة كبيرة
للتحكم فيها

11.13 DC Motor Speed Control

Report ②

speed control of Dc motor

out Put → speed.
Input → Load , Load.
R, L, v → input
↳ Initial condition قيم نتغير

كيف يتم التغير من ثوابت إلى متغيرات ؟!
عن طريق معادلة أو بلاقي نسبها منها
دا ى تغير فى الجهد يقابله تغير فى السرعة).
V_s هو الوسيلة اللى بتحكم بيها فى السرعة.

← خطوة دراسة الـ system دون تحكم عشان نعرف هو محتاج تحكم أصلا ولا لأ.

open loop جملة مثل closed loop شكل الـ Input فى حالة الـ open يكون شكل السرعة.

عمل أية لتحقيق الهدف؟ وقيمة أداى؟

Input. single → single out Put
multi input → single out Put

12 THIS YEARS PRESENTATIONS

12.1 PF improvement

We mean improve power factor seen by generation point

So we will improve total PF for (Load + TL) , so we measure PF at the Generation point and determine if it's suitable, or not. Usually it won't so we need to improve it by

12.1.1 Methods of improving PF

1- Adding Capacitors
2- Capacitor and Inductor
3- Constant Capacitance value
4- Variable ~ ~
5- Series and Parallel

12.1.2 What are parameters on which we decide which method we will use

1- The value of Q required (from Power Triangle)
2- The Budget and Available component
3- Installation in system, and how will it affect system operation?

12.2 Automation in Power System

12.2.1 Definitions

Power System:	it's sums of equipment that generate, transmit and distribute electrical power.
Automation :	it's measuring, processing and controlling and doing action automatically without the interference of human
Automation System :	sums of equipment that measure and acquire data, then Process it and take a decision then apply that decision to the system, without human interference. is the act of automatically controlling the power system via instrumentation and control devices.
IED	intelligent electronic devices
RTU	Remote Terminal Unit
I &C	Insutruments and control
PLC	Programmable Logic Control
HMI	Human Machine Interface The front panel display and push buttons or a personal computer act as interfaces to system data and controls for personnel in the substation.
SCADA	supervisory control and data acquisition
Communications Processor	A communications processor is a substation controller that incorporates the functions of many other I&C devices into one IED .

12.2.2 Tasks of Automation System

Power-system automation is composed of several tasks:

Data acquisition	acquiring, or collecting data. This data is collected in the form of measured analog current or voltage values or the open or closed status of contact points from : **- Instrument Transformers with protective relays (CT , VT)** **- Auxiliary contacts of switch gears** . Acquired data can be used locally within the device collecting it, sent to another device in a substation or to database
Supervision:	**Computer processes** and **personnel supervise**, or monitor, the conditions and status of the power system using this acquired data. Operators and engineers monitor the information **remotely** on **computer** displays and graphical **wall displays** **or locally,** at the device, on **front-panel** displays and laptop computers.

Control	Control refers to sending **command messages** to a device to **operate the I&C** and **power-system devices**. Traditionally, Traditional *supervisory control and data acquisition* (SCADA) systems rely on operators to supervise the system and initiate commands from an operator console on the master computer. Field personnel can also control devices using front-panel push buttons or a laptop computer

12.2.2.1 Instruments and Control Devices

They are IED

RTU	intelligent RTU can **transfer** collected data to other devices and **receive** data and control commands from other devices. User programmable RTUs are referred to as "smart RTUs." It's from IEDs 
PLC	A Programmable Logic Controller can be **programmed to perform logical control**. *As with the RTU, a dedicated pair of copper conductors for each contact and transducer value is terminated on panels within the PLC.It is like a* *work-horse which work upon the command given by their master.*
Digital fault Recorder (DFR)	IED that records information <u>about power-system disturbances</u>. It is capable of storing data in a *digital format* when **triggered by conditions** detected on the power system. **Harmonics, frequency, and voltag**e are examples of data captured by DFRs. 
Protective Relay	A protective relay is an IED designed to **sense power- system disturbances** and automatically **perform control actions** on the I&C system and the power system to protect personnel and equipment. **The relay has local termination** so that the copper conductors for each contact do not have to be routed to a central termination panel associated with RTU.
Meter	A meter is an IED that is used to create accurate measurements of power-system **current, voltage, and power values.** Metering values such as demand and peak are saved within the meter to create **historical information about the activity of the power system.** 

12.2.2.2 Controlling

Load Tap Changer (LTC)	Changes taps to change output voltage
Recloser Control	Open and recloses on fault to determine if it's a transient o permanent fault

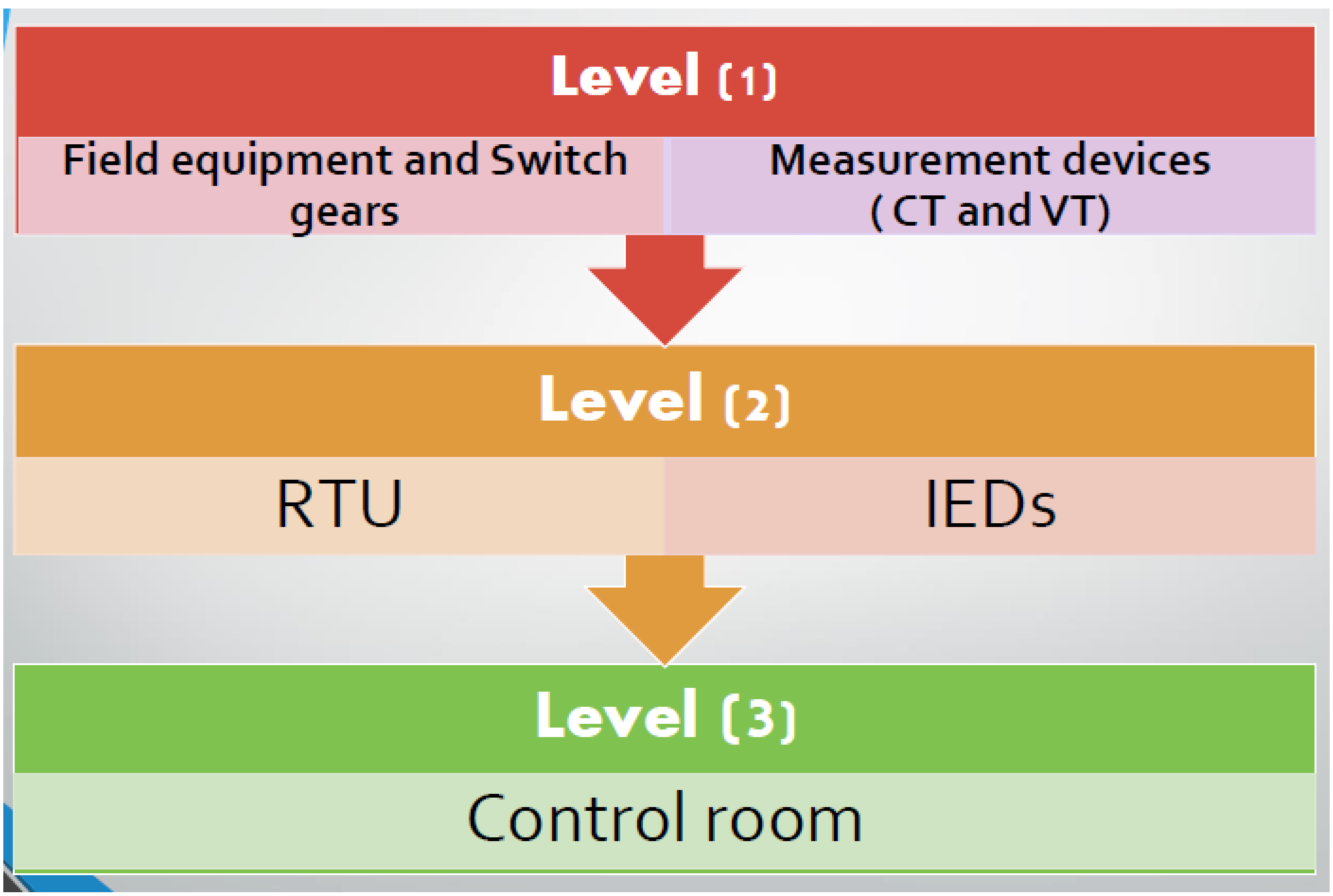
12.2.3 Benefits of Power System Automation

1. Improved **quality** of service and **reduced manpower** requirements.
2. Improved **reliability** with reduced system implementation costs.
3. **Maintenance**/expansion of customer base and **Reduced operating costs**.
4. High value service provider and reduced maintenance costs.
5. **Reduced Fault duration**
6. Flexible Billing Options.

12.2.4 Classification

Substation	Substations have been equipped to perform automatic **re closing, bus sectionalizing, load transfers, capacitor switching, etc.** Advances in communication : (LAN) is used to transmit data instead of costly cables
Distribution	**Electric utility** to **monitor**, **coordinate** and **operate** system components in a real time mode from remote locations the distribution automation is modular and may be implemented in phases to include remote monitoring and control of **substation**, **feeder** and **consumer** devices and loads.

12.2.5 Conclusion



12.3 Elimination in Newton Raphson Method

<u>NR used in</u>

1- Solution of non-linear equations
2- Load flow analysis
3- NR is more accurate and converges, better than gauss method
4- Doesn't depend on initial condition, better than gauss method

Disadvantage : Needs more storage space, more iterations and code

كلام فاضي ! في الكود ! وداله find كله مشروح في الكود فوق

هي الفكره بس عشان تعرف هتحذف ايه من الجاكوبيان

البسط : موجود عند مين ؟؟ (PV , PQ , Both)

المقام : عاوز اجيبه لمين ؟؟ (PV , PQ , Both)

بالتالي من اجابه السؤالين دول هعد عدد الباصات اللي تحقق السؤال الاول n و التاني m

J – nxm

Before elimination (4bus system) Jacobian is 8x8 After 5x5

12.4 Comparison of power flow analysis methods

Power flow study is essential to obtain critical information for determinin

- if system **voltages** remain within specified **limits** under certain conditions,
- equipment such as **transformers** , **generators**, motors and conductors are **overloaded**

AND

- planning, **economic** scheduling
- **control** of an existing system
- planning its **future expansion**.

Information (V,δ, Qg and Pg , flowing P,Q and line losses S12)

BUT it's COMPLEX It takes hours of work and huge number of complex equations in order to obtain the correct data of a specific system.

SO, computational methods were created to obtain these essential data through different algorithms done flawlessly fast on a computer saving both precious time and strenuous efforts.

Gauss-Seidel	Newton Raphson	Fast Decoupled
$P_i^{(k+1)} = R\left\{V_i^{*(k)}\left[V_i^{(k)}\sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij}V_j^{(k)}\right]\right\} \quad j \neq i$ $Q_i^{(k+1)} = -Im\left\{V_i^{*(k)}\left[V_i^{(k)}\sum_{j=0}^{n} y_{ij} - \sum_{j=1}^{n} y_{ij}V_j^{(k)}\right]\right\} \quad j \neq i$ $V_i^{(k+1)} = \frac{\frac{P_i - jQ_i}{V_i^{*(k)}} - \sum_{j\neq i} Y_{ij}V_j^{(k)}}{Y_{ii}}$	$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = \begin{bmatrix}J_1 & J_2\\ J_3 & J_4\end{bmatrix}\begin{bmatrix}\Delta\delta\\ \Delta\lvert V\rvert\end{bmatrix}$ $P_i = \sum_{j=1}^{n}\lvert V_i\rvert\lvert V_j\rvert\lvert Y_{ij}\rvert\cos(\theta_{ij}-\delta_i+\delta_j)$ $Q_i = -\sum_{j=1}^{n}\lvert V_i\rvert\lvert V_j\rvert\lvert Y_{ij}\rvert\sin(\theta_{ij}-\delta_i+\delta_j)$ $\Delta P_i^{(k)} = P_i^{sch} - P_i^{(k)}, \quad \Delta Q_i^{(k)} = Q_i^{sch} - Q_i^{(k)}$ $J_1^k = \left.\frac{\partial P_i}{\partial \delta_i}\right\rvert_{\delta_i^k, V_i^k} \quad J_2^k = \left.\frac{\partial P_i}{\partial V_i}\right\rvert_{\delta_i^k, V_i^k} \quad J_3^k = \left.\frac{\partial Q_i}{\partial \delta_i}\right\rvert_{\delta_i^k, V_i^k} \quad J_4^k = \left.\frac{\partial Q_i}{\partial V_i}\right\rvert_{\delta_i^k, V_i^k}$	$\frac{\partial P_i}{\partial \delta_i} = \sum_{j=1}^{n}\lvert V_i\rvert\lvert V_j\rvert\lvert Y_{ij}\rvert\sin(\theta_{ij}-\delta_i+\delta_j) - \lvert V_i\rvert^2\lvert Y_{ii}\rvert\sin\theta_{ii}$ $\frac{\partial P_i}{\partial \delta_i} = -Q_i - \lvert V_i\rvert^2\lvert Y_{ii}\rvert\sin\theta_{ii} = -Q_i - \lvert V_i\rvert^2 B_{ii}$ $\frac{\partial P_i}{\partial \delta_j} = -\lvert V_i\rvert\lvert V_j\rvert B_{ij}$ Where: $B_{ii} \gg Q_i$, $\lvert V_i\rvert^2 \approx \lvert V_i\rvert$, $\lvert V_j\rvert \approx 1$ $\frac{\partial P_i}{\partial \delta_i} = -\lvert V_i\rvert B_{ii}$ $\frac{\partial P_i}{\partial \delta_j} = -\lvert V_i\rvert B_{ij}$ $\frac{\partial Q_i}{\partial \lvert V_i\rvert} = -\lvert V_i\rvert\lvert Y_{ii}\rvert\sin\theta_{ii} - \sum_{j=1}^{n}\lvert V_i\rvert\lvert V_j\rvert\lvert Y_{ij}\rvert\sin(\theta_{ij}-\delta_i+\delta_j)$ $\frac{\partial Q_i}{\partial \lvert V_i\rvert} = -\lvert V_i\rvert\lvert Y_{ii}\rvert\sin\theta_{ii} + Q_i$ $B_{ii} = Y_{ii}\sin\theta_{ii} \gg Q_i, Q_i$ $\frac{\partial Q_i}{\partial \lvert V_i\rvert} = -\lvert V_i\rvert B_{ii}$ assuming $\theta_{ij} - \delta_i + \delta_j \approx \theta_{ij}$ $\frac{\partial Q_i}{\partial \lvert V_j\rvert} = -\lvert V_i\rvert B_{ij}$ $\frac{\Delta P}{\lvert V_i\rvert} = -B'\,\Delta\delta$ $\frac{\Delta Q}{\lvert V_i\rvert} = -B''\,\Delta\lvert V\rvert$ $\Delta\delta = -[B']^{-1}\frac{\Delta P}{\lvert V\rvert}$ $\Delta\lvert V\rvert = -[B'']^{-1}\frac{\Delta Q}{\lvert V\rvert}$ $\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix} = \begin{bmatrix}J_1 & 0\\ 0 & J_4\end{bmatrix}\begin{bmatrix}\Delta\delta\\ \Delta\lvert V\rvert\end{bmatrix}$

12.4.1 4-bus system (small)

Results overview

Difference	**Gauss-Seidel**	**Newton-Raphson**	**Fast Decoupled**
Error	$0.9561*10^{-4}$	$0.3205*10^{-10}$	$6.4717*10^{-4}$
	#2	#1	#3
Time	0.037015	0.061917	0.054505
	#1	#3	#2
Number of Iterations	17	4	7
	#3	#1	#2

12.4.2 30-Bus System

Results overview

Difference	**Gauss-Seidel**	**Newton-Raphson**	**Fast Decoupled**
Error	$9.51884*10^{-4}$	$7.54898*10^{-7}$	$9.19582*10^{-4}$
	#3	#1	#2
Time	0.06915	0.370300	0.088921
	#1	#3	#2
Number of Iterations	34	4	15
	#3	#1	

Conclusion

- Gauss- Seidel method needs more iterations as compared to other methodologies for the same values of |V|, angle, active and reactive power.
- Newton Raphson method gives better results than GS method for a fewer no. of iterations.
- Fast Decoupled method gives the approximately same results as obtained by NR method with less time

- NR gives accurate results for 4 iteration , but larger time !
- G has less time than FD as initial conditions are available (online)
- FD is accurate if X>>R
- **To get better results use 1 iteration NR then use these results as initial in gauss**
- **Accuracy and Speed are really required (no initial ,offline) FD is ur hero**
-

12.4.3 In control system

Small number of buses : NR

Large number of buses : FD

Don't us gauss, control needs less time and more accurate

Point of comparison	Gauess-Seidal	Newton-Raphson	Fast decoupled
Time elapsed	Lowest time	Highest time	Higher than GS and lower than NR
Number of iterations	Highest number of Iterations	Lowest number of iterations	Higher than NR and lower than GS
Dependence on initial values	Fully dependence on initial values	Slightly dependency on initial values **DOESN'T**	Slightly dependent on initial values **DOESN'T**
Accuracy	Low	High	Lowest (ok)
Number of Equations	Small number of equations (simple)	Highest number of equations (complex)	Half that of NR
Efficiency for large systems	Low efficiency and not practical	High efficiency and more practical	More efficient than GS and less than NR
Memory required	Least memory required	High amount required	More than GS and less than NR
Using online Calculations	Using offline calculations	Using online calculations	Sometimes use online calculations
Divergence	Diverges for wrong initails	Sure to converge	Sure to converge

F.O.C.	Gauss-Seidel	Newton-Raphson
No. of eqns.	Use small no. of equations for solving.	Require large and complex no. of equations.
Divergence	Small divergence between the error in every iteration	High divergence between the error in every iteration
Accuracy	low	high
Efficiency for large power system	Low and not practical for large power systems	More practical and high efficiency

12.5 Artificial Intelligence

12.5.1 History

1940-1950 (early days)

- 1943:mcCulloch&pitts:boolean Circuit Model Of Brain
- 1950:Turing' S "Computing Machinery And Intelligence"

1950-1970(excitement)

- 1950:Early Al Programs , including samul's Checkers Program ,(Newell & simon)
- 1956:dertmoouth meeting: " artificial intelligence"
- 1965:robinson's complete algorithm for logical reasoning

1970-1990 (knowledage - based approaches)

- 1980-1988:export systems industry booms

1990-------(statistical Approaches)

- Resurgence Of Probability , foucs On Uncertainty
- Learning Systems

12.5.2 Definition

AI : It is a branch of computer science that interested in creating programs to simulate human intelligence

12.5.3 Artificial Intelligence Techniques

- **Natural language processing (NLP)**
- **Image and pattern processing**
- **Robotics**
- **Expert System Techniques (XPSs)**
- **Artificial Neural Networks (ANNs)**
- **Fuzzy Logic systems (FL)**

12.5.4 Applications in power system

- **Stability** analysis and Load Flow
- Power system **control**
- **Fault** diagnosis
- **Security** assessment
- Load **forecasting**
- **Reactive** power planning and its control
- **State estimation**

Example

Electric Vehicle (AI or automation)

Experts, have previous knowledge and data about how should the output become based on the input.

AI input (actual data or Optimized)

i.e. load flow the AI takes input and built functions

AI has multiple output and begins to compare

Optimization chooses optimum solution (AI is used in optimization)

12.5.5 For a system we don't know it's model we use AI to model it as follows

1- Training (so the AI becomes more expertise)[on given i/p and o/p data]
2- Test (for input I know its output from actual system)
3- Model (when it passes the test)

12.6 Application of Artificial Intelligence in power system

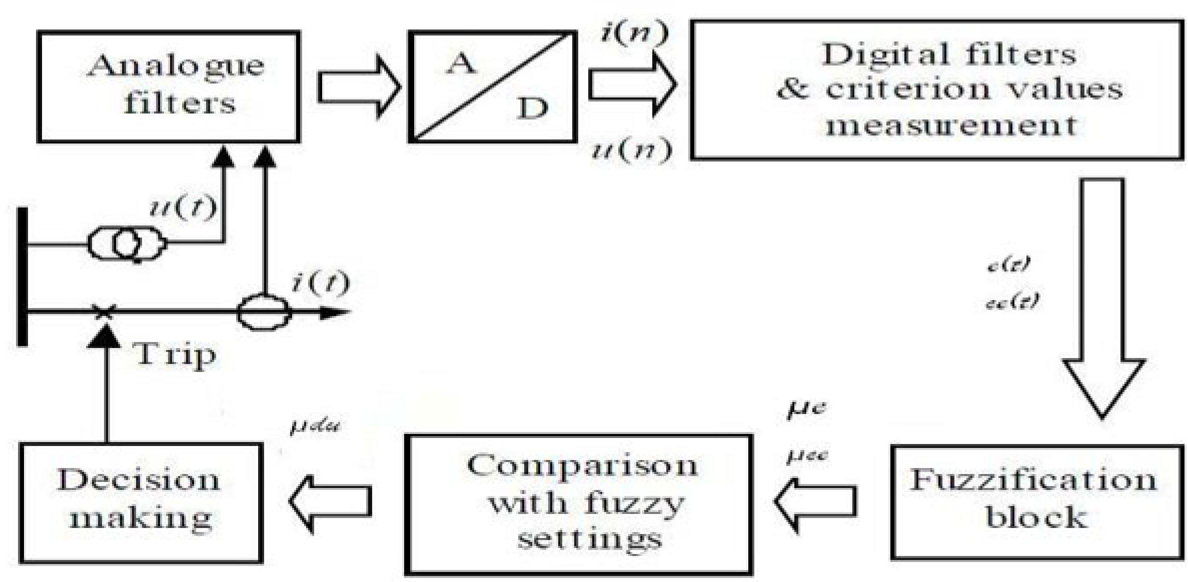
A **fast response is the most important** characteristic in protection.

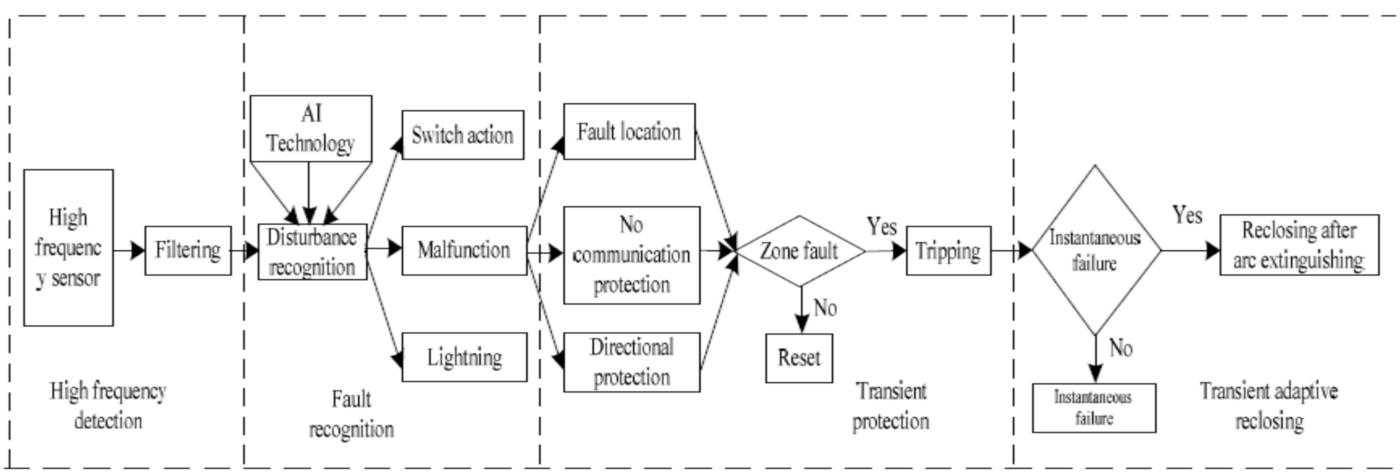
To accomplish this, modern artificial intelligence technologies are used

12.6.1 Need for AI in protection

- Improves the **efficiency** of dealing with incidents.
- Solves **coordination** problems of both overcurrent and directional overcurrent relays.
- Assists operational personnel to check and judge which greatly reduces the workload of operational staff.

12.6.2 Config





: High Frequency Signal Detection	*Fault Recognition*	*Transient Protection*	*Transient Adaptive Reclosing*

12.6.3 Fuzzy

AI depends on Fuzzy which depends on amount of change

Fault (change in current up)(change in voltage down)

Uses estimation of samples value (voltage)

$$X_{estimated} = A_1 * sin\left[2 * sin^{-1}\left(\frac{x_{sampled}(t+1)}{A}\right) - sin^{-1}\left(\frac{x_{sampled}(t)}{A}\right)\right]$$

$$e\,(t+2) = X_{sampled}(t+2) - X_{Estimated}(t+2)$$

$$ce(t+2) = e(t+2) - e(t+1)$$

e : error and ce : change of error

AI isolates fault faster than Relay

Fuzzy Rules

	Error	CE	Output	Trip
1	NB	NB	NB	√
2	NB	CZ	NB	√
3	NB	PB	CZ	×
4	CZ	NB	NB	√
5	CZ	CZ	CZ	×
6	CZ	PB	PB	√
7	PB	NB	CZ	×
8	PB	CZ	PB	√
9	PB	PB	PB	√

12.7 Optimization

<u>Optimize :</u> Choosing most effective solution within constains

<u>Optimization Aim and Focus</u> : Min Cost at Max Consumer welfare

- In power system, where there are many places where optimization is used a lot these days.

12.7.1 Why ?

In our distribution system there are a lot of things to improve like

- Reduce the losses.
- Improve voltage.
- Improve the stability limits etc….

12.7.2 Where use optimization

- Architecture
- Electrical Network
- Economics
- Material Design
- Image Processing
- Transportation

12.7.3 Optimizaation issues

1- Economic Dispatch :
For different types of generation , provide Power at min cost (fuel consu,ption)

2- Unit Commitment
Least No. of generators to provide power

3- Distributed generation
Best bus to apply the generation unit for min cost , best performance (min V drop), min losses

4- Phasor measurement unit placement
To use least number of PMU to monitor the whole system

5- OPTIMAL POWER FLOW (OPF)
The goal of OPF is to find the optimal settings of a given power system network that optimize the system objective functions.

12.7.4 Techniques

It has 3 main

Objective Function	Model or Function that defines the system
Designed Variables	Variable needed to be optimized
Constrains and limits (equation and inequalities)	

12.7.5 EXAMPLE For Power system (OPTIMAL POWER FLOW (OPF))

Objective Function contains

- Total generation cost.
- System loss.
- Bus voltage deviation.
- Emission of generating units.
- Load shedding while satisfying its power flow equations.
- System security.
- Equipment operating limits.
- Number of control actions

CONTROL ACTIONS

• Generators' real power outputs and voltages.

• Transformer tap changing settings.

• Phase shifters.

12.7.6 Techniques

Mathematical (objective and constrains are..)	Linear and Quadratic Programming Nonlinear Programming Integer and Mixed-Integer Programming (cont. and discrete) Dynamic Programming
Artificial Intelligence	Expert System Artificial Neural Network Fuzzy logic Evolutionary Computation Ant Colony Search Taboo Search
Hybrid AI	

In optimization we need to be near ideal

Optimize what ? (1- technical 2- times 3- cost 4- all ?)

12.8 Applications of Optimization Techniques

12.8.1 Optimal PMU displacement (Phase Measurement unit)

PMU : measures V and I as magnitude and angle
Monitors system conditions (normal or up normal)
measure power flow at each bus

So we determine min number of PMU to see the whole system

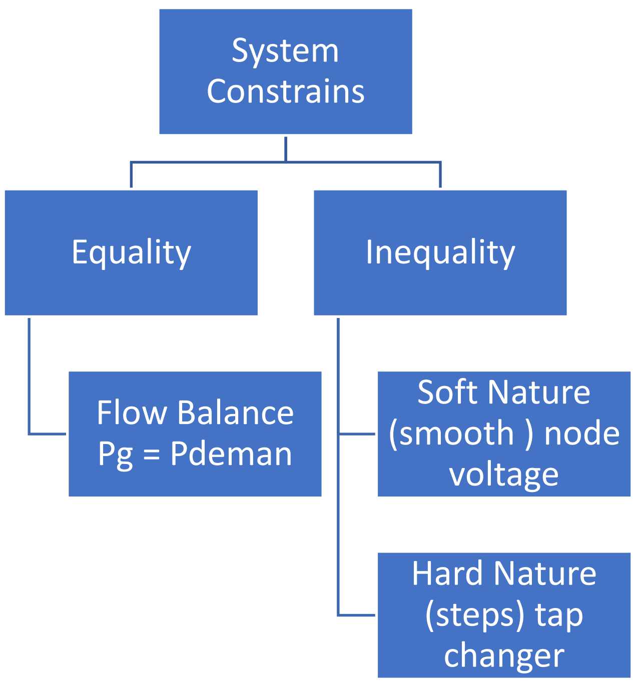
12.8.2 Economic Dispatch

Distribute load between generating units to get to the least cost of fuel

<u>USING COMPUTER</u>

To take into account a lot of variables such as (T.L. capability, Voltage Drop) and put all constrains into account

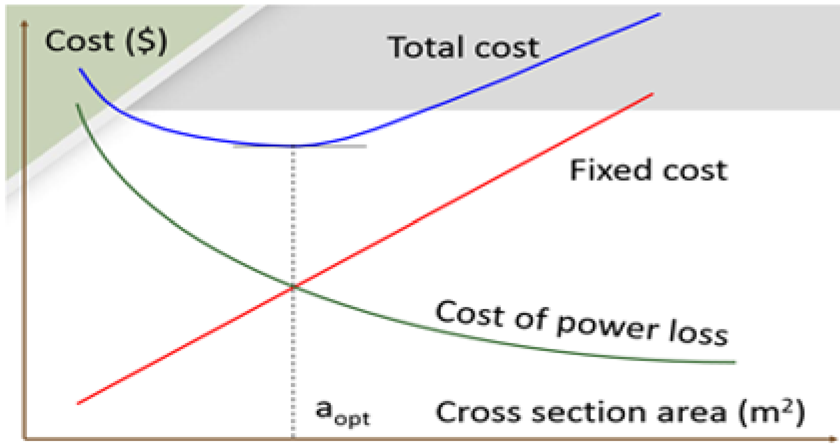
12.8.3 Constrains



12.8.4 Optimization of Power Transformer Design

Losses, Voltage Regulation, Tests

12.8.5 Optimization of cross section area of conductors



Factors and constrains

1- Thermal capability
2- Cost
3- Power loss
4- Voltage drop
5- Mechanical constrains
6- Cooling methods

NOTE

The more the constrains the complex the solution to optimize

12.8.6 Optimum Capacitor bank to improve PF

13 STEPS OF CODES

13.1 YBUS :

1- Get line data (f t r x y)
2- Column line data
3- Get nbr and nbus
4- Get z and y=1/z
5- Ybus = zeros(nbus,nbus)
6- Off diagonal
7- On diagonal
8- Output Ybus

13.2 Gauss-Seidel

1- Ybus
2- Bus data
3- Columining (busno,type,vm,va,Pg.Qg.,Pd,Qd,Qmin,Qmax,Qinj)
4- Nbus
5- Matrices (V, P,Q net , S net)
6- Begin solver (v1=v , tol, dv, itr)
7- While (dv > tol)
8- Get sum term
9- Get Q and limits Qg for PV
10- Get Q,S,Va
11- Get V for PQ
12- Update values (va,vm,mis)
13- Update solver (itr, dv, v1)
14- Output (v,vm,va,itr)

13.3 Newton-Raphson

1- YBUS
2- Busdata
3- Columining
4- Nbus
5- Initials (P,Q net , ya, ym)
6- Solver (itr , error , tol)
7- While (error > tol)
8- Get jacobians (and finish them)
9- Eliminate Jacobians
10- Get dP(pq_pv)' and dQ(pq)'
11- Output and add mismatch (va(pq_pv) (1:nb-1) vm(pq) (nb:2nb-2-npv))
12- Update solver (itr+1 ,error = max(dPQ))
13- Output (Vm,Va,V,J,itr)

13.4 Fast Decoupled

1- Ybus
2- Busdata
3- Columning
4- Nbus
5- Initial P,Q net ya ym B
6- B1=B B4=B then eliminate
7- Initiate solver tol itr error
8- While error > tol
9- dP/vm
10- dQ/vm
11- dva , dvm
12- update solver (itr+1) max(maxdP,maxdQ))
13- output vm va v itr

13.5 Power Flow

1- Stl (sij , sji)
2- Sloss = sij + sji
3- Pg , Qg for slack and PV (calc+dem-inj)

```
clc
clear all
 %%
%(1) ybus

linedata= [1 2 5 6 1;
           2 3 6  5 0;
           3 1 4  3  0;]

fb= linedata(:,1);
tb= linedata(:,2);
r=  linedata(:,3);
x=  linedata(:,4);
b=  1i.*linedata(:,4);
nbr= length(fb) ;
nbus = max(max(fb),max(tb)) ;
z= r+ 1i.*x ;
y = 1./z    ;
Ybus= zeros(nbus,nbus) ;
 for i= 1: nbr

     if( fb(i)~= 0 && tb(i) ~= 0 )
                 Ybus(fb(i),tb(i)) = Ybus(fb(i),tb(i))  - y(i) ;
         Ybus(tb(i),fb(i)) = Ybus(fb(i),tb(i)) ;
     end

 end
 for i= 1:nbus
     for j= 1:nbr
         if( fb(j) == i || tb(j) == i)
             Ybus(i,i)= Ybus(i,i) + y(j) + b(j);
          end
     end
 end
 Ybus
  --------------------------------------------------

  %% (2) Gauss-Seidel
 %Ybus
   busdata=[1 0 1 0  0 0   .5 .3 0 0 0 ;
         2 2 1 0  0 0   1.7 1.0535 0 0 0 ;
         3 2 1 0  0 0   2 1.2394 0 0 0;
         4 1 1.02 0 3.18 0 .8 .4958 0 0 0];
    busno = busdata(:,1) ;
    type = busdata(:,2) ;
    vm = busdata(:,3) ;
    va = busdata(:,4) ;
    Pg = busdata(:,5) ;
    Qg = busdata(:,6) ;
    Pd = busdata(:,7) ;
    Qd = busdata(:,8) ;
    Qmin = busdata(:,9) ;
    Qmax = busdata(:,10) ;
    Qinj = busdata(:,11) ;
    nbus= length(busno);
    v= vm.*cosd(va) + 1i.*va.*sind(va) ;
    P= Pg-Pd ;
    Q= Qg-Qd+Qinj ;
    S= P+ 1i.*Q ;
    itr=0 ;
    tol = 10^-6;
```

```
    dv=1 ;
    v1=v ;
while( dv>tol )
    for i= 1:nbus
        if type(i) ~= 0
        sum=0 ;
        for j=1:nbus
            if j~=i
                sum=sum+Ybus(i,j)*v(j);
            end
        end
            if type(i) == 1
                Q(i) = -imag(conj(v(i))*(v(i)*Ybus(i,i) + sum)) ;
                Qg(i) = Q(i) +Qd(i) -Qinj(i)                     ;
                if Qmax(i) ~= 0
                    if Qg(i) > Qmax(i)
                        Qg(i)= Qmax(i) ;
                    elseif Qg(i) < Qmin(i)
                            Qg(i) = Qmin(i) ;
                    end
                end
                Q(i)= Qg(i)  -Qd(i)   +Qinj(i) ;
                S(i)= P(i) + 1i*Q(i) ;
                v(i) = (1/Ybus(i,i))*((conj(S(i))/conj(v(i))) - sum );
                va(i) = angle(v(i))*180/pi ;
                v(i)= vm(i)*cosd(va(i)) + 1i.*vm(i)*sind(va(i)) ;
            end
            if type(i) ==2
            v(i) = (1/Ybus(i,i))* ((conj(S(i))/conj(v(i))) - sum );
            end
            vm(i)= abs(v(i) ) ;
            va(i) = angle (v(i))*180/pi ;
            mis(i) = abs(v(i)-v1(i)) ;
             end
     end
            dv= max(mis);
            itr= itr+1;
            v1= v ;
end
v
itr
vm
va
mis
```

```
 % (3) Newton-Raphson
 %Ybus
   busdata=[1 0 1 0  0 0   .5 .3 0 0 0 ;
         2 2 1 0  0 0   1.7 1.0535 0 0 0 ;
         3 2 1 0  0 0   2 1.2394 0 0 0;
         4 1 1.02 0 3.18 0 .8 .4958 0 0 0];
    busno = busdata(:,1) ;
    type = busdata(:,2) ;
    vm = busdata(:,3) ;
    va = busdata(:,4) ;
    Pg = busdata(:,5) ;
    Qg = busdata(:,6) ;
    Pd = busdata(:,7) ;
    Qd = busdata(:,8) ;
    Qmin = busdata(:,9) ;
    Qmax = busdata(:,10) ;
    Qinj = busdata(:,11) ;
```

```
nbus= length(busno);
P = Pg-Pd;
Q = Qg-Qd+Qinj ;
ym= abs(Ybus);
ya=angle(Ybus)*180/pi;
itr=0;
error= 1;
tol = 10^-12 ;
while( error > tol )
    for i = 1:nbus
        sum2=0;
        sum4=0;
        j1  =0;
        j3  =0;
        for j= 1: nbus
            if j~=i
                j1(i,i)=j1(i,i)+vm(i)*vm(j)*ym(i,j)*sind(ya(i,j)+va(j)-va(i));
                j1(i,j)=-1*vm(i)*vm(j)*ym(i,j)*sind(ya(i,j)+va(j)-va(i));

                sum2=  sum2+(vm(j)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i)));
                j2(i,j)=vm(i)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i));

                j3(i,i)=j3(i,i)+(vm(i)*vm(j)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i)));
                j3(i,j)=1*vm(i)*vm(j)*ym(i,j)*cosd(ya(i,j)+va(j)-va(i));

                sum4= sum4+(vm(j)*ym(i,j)*sind(ya(i,j)+va(j)-va(i)));
                j4(i,j)= -1*vm(i)*ym(i,j)*sind(ya(i,j)+va(j)-va(i));
                  end
        end
                j2(i,i)=2*vm(i)*ym(i,i)*cosd(ya(i,i))-sum2
                j4(i,i)= -2*vm(i)*ym(i,i)*sind(ya(i,i))+sum4
    end
    sl= find(type==0);
    PV= find(type==1);
    pv_pq= find(type~=0);
    PQ= find(type==2);
    npv= length(PV);
    j1(sl,:)= [];
    j1(:,sl)= [];
    j2(sl,:)= [];
    j2(:,[sl PV])= [] ;
    j3([sl PV],:)=[] ;
    j3(:, sl)= [];
    j4([sl PV],:)=[] ;
    j4(:,[sl PV])= [] ;
for i= 1:nbus
    Pcalc(i)=0 ;
    Qcalc(i)=0 ;
     for j=1:nbus
Pcalc(i)=  Pcalc(i)+ vm(i)*vm(j)*ym(i,j)*cosd(+va(j)+ya(i,j)-va(i)) ;
Qcalc(i)=  Qcalc(i)-  vm(i)*vm(j)*ym(i,j)*sind(+va(j)+ya(i,j)-va(i)) ;
     end
end
dP = P(pv_pq)-Pcalc(pv_pq)';
dQ = Q(PQ) - Qcalc(PQ)' ;
J= [ J1 J2; J3 J4];
dPQ = [ dP ; dQ] ;
mismatch = inv(J)*dPQ ;
va(pv_pq) = va(pv_pq) + mismatch(1:nbus-1) ;
vm(PQ) = vm(PQ) + mismatch(nbus:2*nbus-2-npv)*180/pi;
error =max(abs(dPQ)) ;
itr=itr+1
```

```
 end
 va
 vm
 itr
 J
 v=vm.*cosd(va)+1i*vm.*sind(va);
  --------------------------------------------------

 %%  (4) Fast Decoupled

  %Ybus
busdata=[1 0 1 0  0 0   .5 .3 0 0 0 ;
      2 2 1 0  0 0   1.7 1.0535 0 0 0 ;
      3 2 1 0  0 0   2 1.2394 0 0 0;
      4 1 1.02 0 3.18 0 .8 .4958 0 0 0];
 busno = busdata(:,1) ;
 type = busdata(:,2) ;
 vm = busdata(:,3) ;
 va = busdata(:,4) ;
 Pg = busdata(:,5) ;
 Qg = busdata(:,6) ;
 Pd = busdata(:,7) ;
 Qd = busdata(:,8) ;
 Qmin = busdata(:,9) ;
 Qmax = busdata(:,10) ;
 Qinj = busdata(:,11) ;
 nbus= length(busno);
 P = Pg-Pd;
 Q = Qg-Qd+Qinj ;
 ym= abs(Ybus);
 ya=angle(Ybus)*180/pi;
 B= -imag(Ybus) ;
 sl= find(type==0);
 pv= find(type==1);
 pq= find(type==2);
 pv_pq=find(type~=0);
 npv= length(pv);
 B1=B;
 B4=B;
 B1(:,sl)=[];
 B1(sl,:)=[];
 B4([sl pv],:)=[];
 B4(:,[sl pv])=[];
 itr=0;
 error=1;
 tol= 10^-12;
 while(error>tol)
     r=0;
     for i=1:nbus
         if(type(i)~=0)
             r=r+1;
             Pcalc=0;
             for j=1:nbus
             Pcalc= Pcalc+vm(i)*vm(j)*ybusm(i,j)*cosd(ybusa(i,j)+va(i)-va(j));
             end
           dP(r)=(P(i)-Pcalc)/vm(i) ;
         end
     end

w=o
for i=1:nbus
    if(type(i)==2)
```

```
            w=w+1;
            Qcalc=0
            for j=1:nbus
                Qcalc=Qcalc-vm(n)*vm(k)*ybusm(n,k)*sind(ybusa(n,k)+va(k)-va(n));
            dQ(w)= (Q(i)-Qcalc)/vm(i) ;
        end
    end
    dva= inv(B1)*dP' ;
    dvm= inv(B4)*dQ' ;
    va(pv_pq)= va(pv_pq) + dva*180/pi;
    vm(pq)= vm(pq) + dvm;
    itr=itr+1
    error = max(max(abs(dP)),max(abs(dQ))) ;
     end
     va
     vm
     v= vm*cosd(va)+ 1i.*vm.*sind(va);
     itr
```

```
     %% (5) sloss

     for i=1:nbus
         for j=1:nbus
             if j~=i
             s(i,j)= v(i)*conj(Ybus(i,j)*(v(j)-v(i))) ;
             s(j,i)= v(j)*conj(Ybus(j,i)*(v(i)-v(j))) ;
             end
         end
     end
stl= sparse(s);
display(stl);
for i=1:nbus
    for j=1:nbus
        if j~= i
                        sloss= s(i,j)+s(j,i);
        end
    end
end
sloss= sparse(sloss);
display(sloss)
for i=1:nbus
    if (type(i)~=2)
    for j=1:nbus
st(i)=conj(conj(v(i))*(v(j)*Ybus(i,j))
        P(i)=real(st(i))
        Q(i)= imag(st(i))
    end
    if (type(i)==0)
Pg(i) = P(i)+Pd(i)
Qg(i) = Q(i)+Qd(i)-Qinj(i);
elseif (type(i)==1)
        Qg(i) = Q(i)+Qd(i)-Qinj(i);
end
    end
end
 sloss
Q
Pg
Qg
ploss=real(sloss)
```