DEMO Code of

Topology Identification and Line Parameter Estimation for non-PMU Distribution Network: A Numerical Method

Jiawei Zhang, Student Member, IEEE, Yi Wang, Member, IEEE, Yang Weng, Member, IEEE, and Ning Zhang, Senior Member, IEEE

clear;

fprintf('[Identification] Pragramme starts .........\n')

[Identification] Pragramme starts .........

# Data processing

## Import data

fprintf('[Identification] Data processing .........\n')

[Identification] Data processing .........

pf = loadcase('case33bw'); % test case

load = readLD; % power load origin file

freq = 1/10; % times per hour

err = 0.001; % additional error

tol = 10^-8; % tol for ac flow calculation

qlist = data\_processing(pf,load,freq,err,tol); % build datasets

[data processing] Build No. 50 dataset

[data processing] Build No. 100 dataset

[data processing] Build No. 150 dataset

[data processing] Build No. 200 dataset

[data processing] Build No. 240 dataset

# Basic identification

fprintf('[Identification] Basic identification .........\n')

[Identification] Basic identification .........

pect = 0.02; % gammar: remove branch thro

thro = 10^-8; % decide stop basic identification

[mpfn,flag] = basicidentify(qlist,pect,thro); % basic identification

[Basic Identification] No.1 ends with error 199.136773

Wrong branch +/-: 192,0

[Basic Identification] No.2 ends with error 85.058044

Wrong branch +/-: 43,0

[Basic Identification] No.3 ends with error 19.823915

Wrong branch +/-: 14,0

[Basic Identification] No.4 ends with error 8.336062

Wrong branch +/-: 10,0

[Basic Identification] No.5 ends with error 2.329934

Wrong branch +/-: 8,0

[Basic Identification] No.6 ends with error 2.361255

Wrong branch +/-: 6,0

[Basic Identification] No.7 ends with error 2.233993

Wrong branch +/-: 6,0

[Basic Identification] Converged after 8 iterations with error 0.000000e+00

Wrong branch +/-: 6,0

Time passed 0.064540 s。

% flag == 0: wrong topo

% detected

if flag == 0

fprintf('[Basic identification] Wrong topology detected. REDUCE PECT!\n We terminate the programme to save time\n')

quit();

end

tic;

# Fine identification

fprintf('[Identification] Fine identification .........\n')

[Identification] Fine identification .........

## Select last 30 data

mpfn.q = mpfn.q(:,end-29:end);

mpfn.smat = mpfn.smat(:,end-29:end);

mpfn.dsmat = zeros(size(mpfn.smat));

mpfn.vmat = mpfn.vmat(:,end-29:end);

mpfn.thetamat = mpfn.thetamat(:,end-29:end);

## Fine identification SETTINGS

MAXITER = 25; % maximum iteration time

varsigma = 0.01; % decide whether to remove branches

xi = 0.05; % threshold to remove branches

varphi = 1\*10^-10; % decide whether to end iterations

## Constants

[n,~] = size(mpfn.bus);

[m,~] = size(mpfn.branch);

[~,M] = size(mpfn.smat);

## Initialization

deltapqlist = deltaPQ(mpfn);

dftp = norm(deltapqlist,2); % throshold

## Error evaluation when starts

[gberror,wrong\_add,wrong\_miss] = errorevaluate(mpfn,qlist(1));

gberror = gberror\*100; % display in 100%

fprintf('[fine identification] Starts with error %e\n',dftp);

[fine identification] Starts with error 8.482319e-02

fprintf('MAPE: g: %f%%, b: %f%%; Wrong branch +/-: %d,%d\n',gberror(1),gberror(2),length(wrong\_add),length(wrong\_miss));

MAPE: g: 35.028203%, b: 41.714352%; Wrong branch +/-: 6,0

## Start iteration

for T = 1:MAXITER

### Pseudo-Power Flow Calculation

mpfn = pseudopf(mpfn);

### Jacobian matrix

| A B C |

| D E F |

Jmat = Jacobianmat(mpfn);

### Generalized reverse

invJmat = pinv(Jmat);

% Delta S = Delta P,Q

deltapqlist = deltaPQ(mpfn);

% delta g,b,theta

deltagbt = invJmat\*deltapqlist;

% renew g,b and theta

mpfn = renewgbt(mpfn,deltagbt);

### Calculate throshold

deltapqlist = deltaPQ(mpfn);

df = norm(deltapqlist,2); % throshold

### remove wrong branches

if abs(df - dftp)<varsigma

mpfn = removebran(mpfn,xi);

end

### calculate throshold

deltapqlist = deltaPQ(mpfn);

df = norm(deltapqlist,2); % throshold

### Error evaluation

[gberror,wrong\_add,wrong\_miss] = errorevaluate(mpfn,qlist(1));

gberror = gberror\*100; % display in 100%

### Decide whether to end iteration

if abs(df - dftp)<varphi && T~=1

fprintf('[fine identification] Ends with %d iterations, with error %e\n',T,df);

fprintf('MAPE: g: %f%%, b: %f%%; Wrong branch +/-: %d,%d\n',gberror(1),gberror(2),length(wrong\_add),length(wrong\_miss));

break;

else

fprintf('[fine identification] No.%d ends with error %e\n',T,df);

fprintf('MAPE: g: %f%%, b: %f%%; Wrong branch +/-: %d,%d\n',gberror(1),gberror(2),length(wrong\_add),length(wrong\_miss));

end

[fine identification] No.1 ends with error 1.846560e-01

MAPE: g: 20.058325%, b: 16.032614%; Wrong branch +/-: 6,0

[fine identification] No.2 ends with error 3.951914e-02

MAPE: g: 0.776318%, b: 2.068346%; Wrong branch +/-: 6,0

[fine identification] No.3 ends with error 8.341512e-03

MAPE: g: 0.220732%, b: 0.418977%; Wrong branch +/-: 6,0

[fine identification] No.4 ends with error 2.228952e-03

MAPE: g: 0.305660%, b: 0.240022%; Wrong branch +/-: 0,0

[fine identification] No.5 ends with error 4.800817e-05

MAPE: g: 0.113633%, b: 0.156629%; Wrong branch +/-: 0,0

[fine identification] No.6 ends with error 4.347213e-05

MAPE: g: 0.113615%, b: 0.156165%; Wrong branch +/-: 0,0

[fine identification] Ends with 7 iterations, with error 4.347213e-05

MAPE: g: 0.113614%, b: 0.156166%; Wrong branch +/-: 0,0

dftp = df;

end

toc;

fprintf('[Identification] Pragramme successfully ends .........\n')

Time passed 41.840540 s。

[Identification] Pragramme successfully ends .........