Power System Toolbox
Version 2.0

Load Flow Tutorial and Functions

Cherry Tree Scientific Software
RR#5 Colborne
Ontario K0K 1S0
Canada

phone & fax: (905)349-2485
email: cherry@eagle.ca

Table of Contents

Load Flow: A Tutorial

Introduction

In power systems, a load flow study is performed to obtain a set of feasible steady state system conditions which obey certain system constraints. It requires that the system structure is specified together with the generators' real powers and the system's active and reactive power loads. System bus voltage magnitudes and angles are then calculated by solving the nonlinear algebraic network equations so that the specified loads are supplied.

Although load flow studies are important in their own right, they are also required to act as starting points for dynamic simulation.

Data Requirements

The system structure is specified, in PST, by two matrices, **bus** and **line**. The format for these two specification matrices is given in **Function: loadflow**. The example given in that function description is used as a basis for this tutorial.

Load Flow Example Data

The bus and line data of a 4 generator, 2 area system [1] are

```
bus = […
1   1.03     18.5   7.00  1.61  0.00  0.00  0.00  0.00  1  99.0  -99.0  22.0
1.1  .9;
2   1.01     8.80   7.00  1.76  0.00  0.00  0.00  0.00  2  5.0   -2.0   22.0
1.1  .9;
3   0.9781  -6.1    0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   500.0
1.5  .5;
4   0.95    -10     0.00  0.00  9.76  1.00  0.00  0.00  3  0.0   0.0   115.0
1.05 .95;
10  1.0103  12.1    0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5;
11  1.03    -6.8    7.16  1.49  0.00  0.00  0.00  0.00  2  5.0  -2.0    22.0
1.1  .9;
12  1.01    -16.9   7.00  1.39  0.00  0.00  0.00  0.00  2  5.0  -2.0    22.0
1.1  .9;
13  0.9899  -31.8   0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   500.0
1.5  .5;
14  0.95    -38     0.00  0.00  17.67 1.00  0.00  0.00  3  0.0   0.0   115.0
1.05 .95;
20  0.9876   2.1    0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5;
101 1.05    -19.3   0.00  8.00  0.00  0.00  0.00  0.00  2  99.0  -99.0 500.0
1.5  .5;
110 1.0125  -13.4   0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5;
120 0.9938  -23.6   0.00  0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5 ];

line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005     0.00   1.0  0. 1.2 0.8 0.05;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
```

```
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13   14 0.0     0.005    0.00    1.0  0. 1.2 0.8 0.05;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.];
```

The single line diagram of the test system is shown in Fig. 1. The system consists of two identical areas interconnected by two long transmission lines. In each area, there are two generators, at buses 1 and 2 in area 1, and at buses 11 and 12 in area 2. The loads are at bus 4 in area 1, and at bus 14 in area 2. Bus 1 acts as the swing bus. Bus 101 is considered to be a generator in the load flow. It has zero real power generation and acts as a reactive power source to hold the voltage at the center of the interconnecting transmission lines. When we come to do dynamic simulations, this bus will be the site of a static VAR compensator, and the reactive generation will give the initial susceptance of the SVC.

There are step down under-load tap changing transformers between bus 3 and bus 4, and bus 13 and bus 14. The tap settings are changed during a load flow solution so that the load bus voltages are maintained between the limits set in columns 14 and 15 of the **bus** matrix .

The generators at buses 2, 11, and 12 have reactive power limits set to -2pu to 5pu. The swing bus generator and the reactive power source at bus 101 has limits -99pu to 99pu.

The rated voltage (kV) for each bus is specified in column 13 of **bus**. This is not used in an ac power flow, but we will see later, that in a dc power flow the information is necessary, since the dc system is modelled in natural units rather than in per unit.

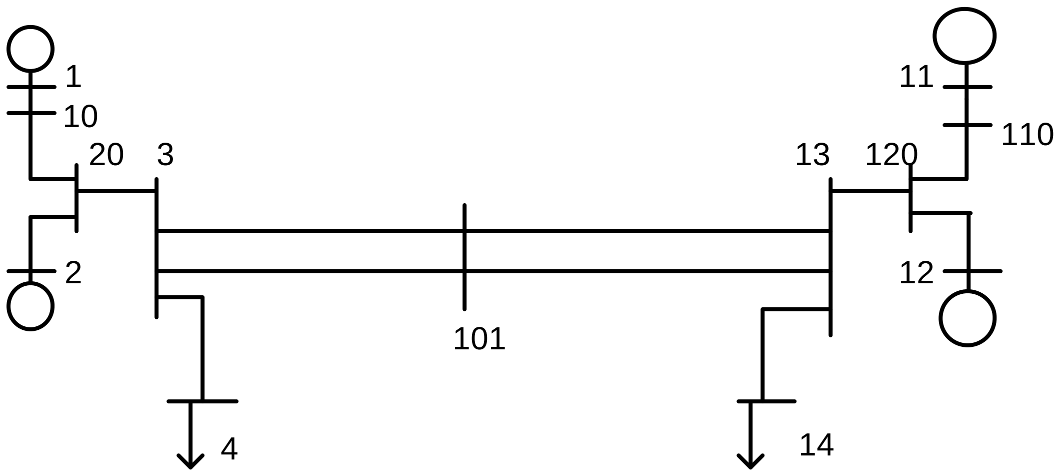


Figure 1 Single Line Diagram 2 Area System

Load Flow Demo

The script file **lfdemo** is an ac load flow driver. When this is typed at the MATLAB command , you are asked to choose a data file which contains the bus and line load flow specification files. In our example case, these are specified in **data2a.m**. If your choice of file contains valid load flow data, you will be asked whether you wish to have a load flow report. Entering **‘y’** opens a diary file in the current MATLAB directory with the name **lf_report.txt**. type **‘n’** or press **enter** if you do not want a report.

As the solution progresses, it is a Newton_Raphson algorithm performed by **loadflow**, the voltages at the load buses are found to be out-of-limits. The corresponding transformer taps are adjusted to bring the load voltage back in range.

At the end of the solution process, either the solution has converged, or the number of allowed iterations has been exceeded. In either case, the user is given a list of solution viewing options. For the example case the solution progress is as follows:

Do you need a load-flow solution report? [y/n] >>
voltage low changing tap on line 3
taps reset to
tap = 0.9500
voltage low changing tap on line 10
taps reset to
tap = 0.9500
voltage low changing tap on line 10
taps reset to
tap = 0.9000

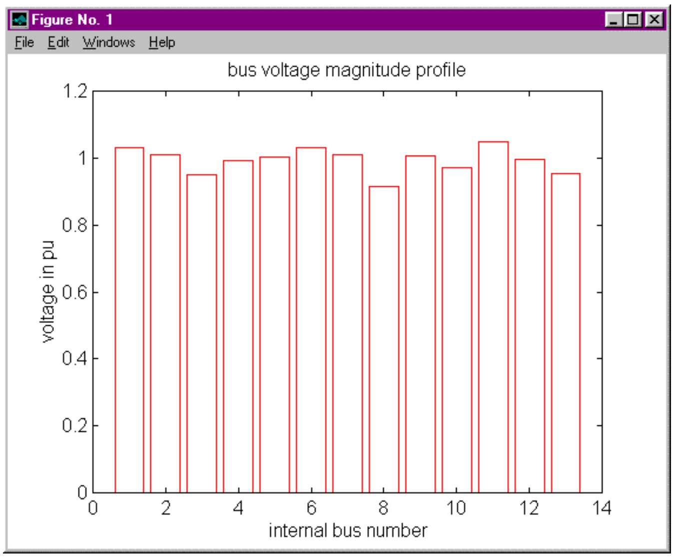
You can examine the system data
Type 1 to see initial bus data
    2 to see modified line data
    3 to see solved load flow bus solution
    4 to see line flow
    5 to see bus voltage magnitude profile
    6 to see bus voltage phase profile
    0 to quit

enter selection >> 3
Solved Bus Data

			GENERATION		LOAD	
BUS	VOLTS	ANGLE	REAL	REACTIVE	REAL	REACTIVE
1.0000	1.0300	18.5000	7.2138	2.0926	0	0
2.0000	1.0100	8.1584	7.0000	2.8023	0	0
3.0000	0.9485	-7.3757	0	0	0.0000	0.0000
4.0000	0.9921	-10.1997	0	0	9.7600	1.0000
10.0000	1.0029	11.8028	0	0	0.0000	0.0000
11.0000	1.0300	-6.9696	7.1600	2.5494	0	0
12.0000	1.0100	-17.3151	7.0000	3.9297	0	0
13.0000	0.9151	-33.3468	0	0	0.0000	0.0000
14.0000	1.0081	-38.2916	0	0	17.6700	1.0000
20.0000	0.9706	1.3096	0	0	0.0000	0.0000
101.0000	1.0500	-21.0220	0.0000	5.0029	0	0
110.0000	0.9955	-13.6667	0	0	0.0000	0.0000
120.0	0.9521	-24.2977	0	0	0.0000	0.0000

You can examine the system data
Type 1 to see initial bus data
    2 to see modified line data
    3 to see solved load flow bus solution
    4 to see line flow
    5 to see bus voltage magnitude profile
    6 to see bus voltage phase profile
    0 to quit

enter selection >> 5



Voltage Stability Demo

The script file **vsdemo** is a driver for steady state voltage stability analysis. The ac load flow program **loadflow** is used in this demo, so as it stands it cannot be used to examine voltage stability in systems having HVDC links.

The demonstration allows the total active and reactive power loads to be increased in steps by a ratio of the original bus loads. A load flow is performed at each step, and if required the inverse eigenvalues of the load flow Jacobian ($\partial Q / \partial V$)can be found. The maximum eigenvalue and the maximum element of the corresponding eigenvector are displayed. The critical eigenvalue may be plotted if desired.

Normally, the as the load increases, the load flow will take longer to converge. Close to voltage stability, it will likely fail to converge. Consequently, the user is given the option of starting the next load flow from the previous load flow solution, or from the original load flow data.

On output, a history of the loads is contained in **load_p** and **load_q**, and that of the system voltages in **v_mag**. These may be plotted to show the system **V/P** characteristics.

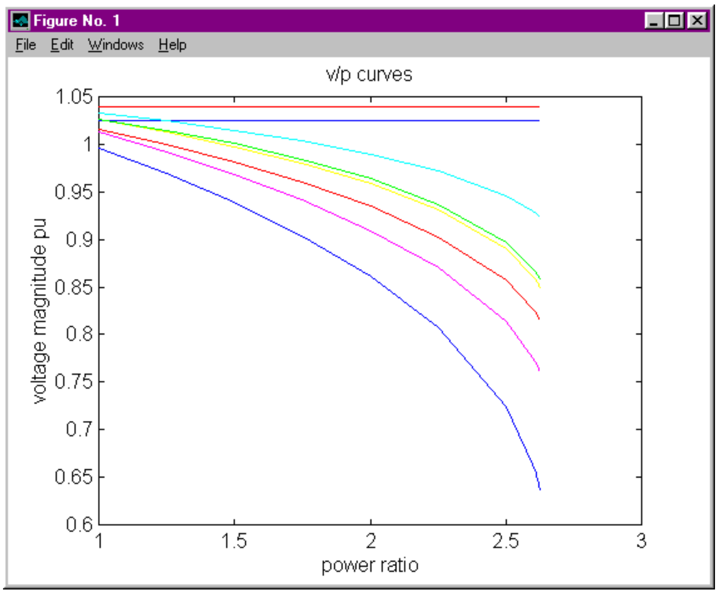
Voltage Stability Example

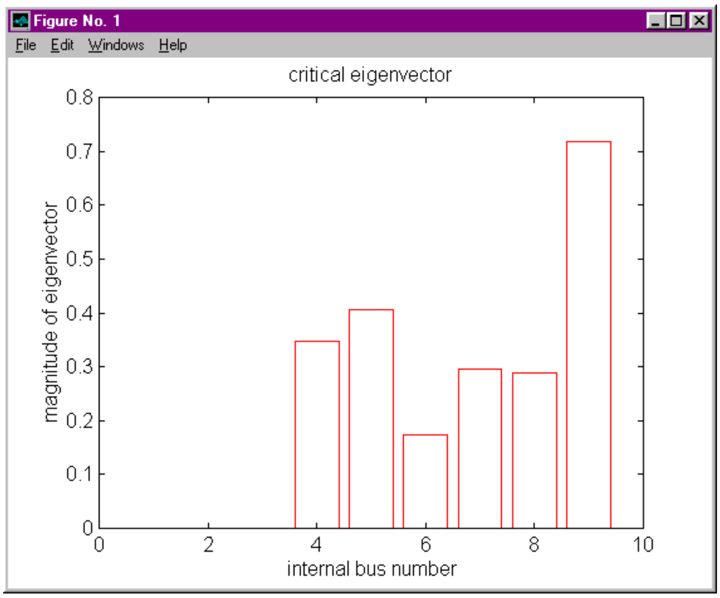
The following data represents a 3 generator, 9 bus system

```
bus = [ 1 1.04    0.00   0.00   0.00  0.00  0.00  0.00  0.00 1;
        2 1.02533 0.00   1.63   0.00  0.00  0.00  0.00  0.00 2;
        3 1.02536 0.00   0.85   0.00  0.00  0.00  0.00  0.00 2;
        4 1.00    0.00   0.00   0.00  0.00  0.00  0.00  0.00 3;
        5 1.00    0.00   0.00   0.00  0.90  0.30  0.00  0.00 3;
        6 1.00    0.00   0.00   0.00  0.00  0.00  0.00  0.00 3;
        7 1.00    0.00   0.00   0.00  1.00  0.35  0.00  0.00 3;
        8 1.00    0.00   0.00   0.00  0.00  0.00  0.00  0.00 3;
        9 1.00    0.00   0.00   0.00  1.25  0.50  0.00  0.00 3];
```

```
line = [1 4 0.0    0.0576 0.     1. 0. ;
        4 5 0.017  0.092  0.158  1. 0. ;
        5 6 0.039  0.17   0.358  1. 0. ;
        3 6 0.0    0.0586 0.     1. 0. ;
        6 7 0.0119 0.1008 0.209  1. 0. ;
        7 8 0.0085 0.072  0.149  1. 0. ;
        8 2 0.0    0.0625 0.     1. 0. ;
        8 9 0.032  0.161  0.306  1. 0. ;
        9 4 0.01   0.085  0.176  1. 0. ];
```

The following results are obtained using vsdemo





The critical eigenvalue at a power ratio of 2.625 is 1.052. The maximum eigenvector is 0.7174 at bus number 9.

HVDC Load Flow

The script file **lfdc** is a load flow driver for systems having HVDC links. In addition to ac load flow data, dc data must be supplied in the form of the dc converter specification matrix (**dcsp_con**) and the dc line specification matrix (**dcl_con**).

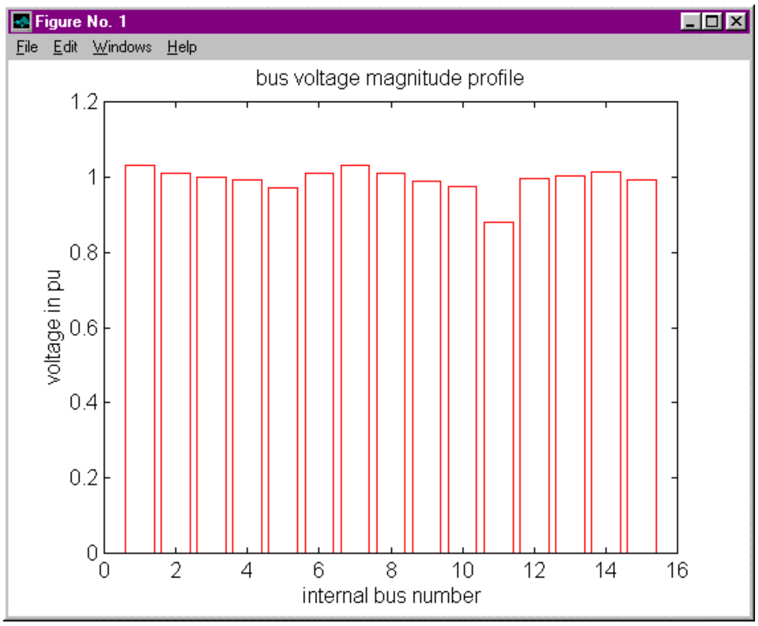
A complete set of data (**d_testdc.m)** for the two area system having an HVDC link between ac bus 5 and ac bus 15 is

```
bus = [ …
1   1.03 18.5  7.00 1.61   0.00  0.00  0.00  0.00  1  99.0 -99.0   22.0  1.1
.9;
2   1.01 8.80  7.00 5.76   0.00  0.00  0.00  0.00  2  99.0 -99.0   22.0  1.1
.9;
3   1.0  -6.1  0.00 0.00   0.00  0.00  0.00  6.00  2   0.0   0.0  500.0  1.5
.5;
4   0.95 -10   0.00 0.00   9.76  1.00  0.00  0.00  3   0.0   0.0  115.0  1.05
.95;
5   1.0  -10   0.00 0.00   10.7  2.8   0.00  0.00  3  99.0 -99.0  115.0  1.2
.8;
10  1.01 12.1  0.00 0.00   0.00  0.00  0.00  0.00  3   0.0   0.0  230.0  1.5
.5;
11  1.03 -6.8  7.16 1.49   0.00  0.00  0.00  0.00  2  99.0 -99.0   22.0  1.1
.9;
12  1.01 -16.9 7.00 1.39   0.00  0.00  0.00  0.00  2  99.0 -99.0   22.0  1.1
.9;
13  0.99 -31.8 0.00 0.00   0.00  0.00  0.00  0.00  2   0.0   0.0  500.0  1.5
.5;
14  0.95 -38   0.00 0.00   17.7  1.00  0.00  0.00  3   0.0   0.0  115.0  1.05
.95;
15  1.0  -14   0.00 0.00  -10.4  2.7   0.00  6.00  3  99.0  -99.0 115.0  1.2
.8;
20  0.99  2.1  0.00 0.00   0.00  0.00  0.00  0.00  3   0.0   0.0  230.0  1.5
.5;
101 1.05 -19.3 0.00 2.00   0.00  0.00  0.00  0.00  3  99.0  -99.0  500.0  1.5
.5;
110 1.01 -13.4 0.00 0.00   0.00  0.00  0.00  0.00  3   0.0    0.0  230.0  1.5
.5;
120 0.99 -23.6 0.00 0.00   0.00  0.00  0.00  0.00  3   0.0    0.0  230.0  1.5
.5 ];

line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005    0.00    1.0  0. 1.5 0.5 0.05;
3    5  0.0     0.007    0.00    1.0 0. 2.0 0.5 0.005;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13   14 0.0     0.007    0.00    1.0  0. 1.5 0.5 0.05;
13   15 0.0     0.01     0.00    1.0 0. 2.0 0.5 0.005;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.];

dcsp_con = [...
1  5  1  500  6    4   5   30;
2  15 2  500  6    4  18   25];
```

```
dcl_con = [...
1  2  20  0  0  0  0  1000  15];
```

The ac buses 5 and 15 are the LT buses of the HVDC converter transformers. The corresponding HT buses are buses 3 and 13 respectively.

The dc load flow is performed by a sequence of ac load flows, followed by dc load flows which reset the dc controls and the loads at the LT buses. When both the ac and dc load flows have converged, the overall HVDC load flow is taken to be converged.



The converter parameters are:
Rectifier

alpha = 22.73 deg
Vdc = 541.1 kV
Pdc = 1082 Mw

Inverter

gamma = 18deg
Vdc = 501.1 kV
Pdc = 1002 Mw

Line current = 2 kA

It should be noted that the LT bus voltage magnitudes are low. This is the result of the dc load flow solution adjusting the taps to get the required dc voltage at the inverter with the minimum extinction angle of 18^{o}.

calc

Purpose:

Calculate the bus power mismatch and check convergence.

Synopsis:

[delP,delQ,P,Q,conv_flag] = calc(nbus,V,ang,Y,Pg,Qg,Pl,Ql,sw_bno,g_bno,tol)

Description:

[delP,delQ,P,Q,conv_flag] = calc(nbus,V,ang,Y,Pg,Qg,Pl,Ql,sw_bno,g_bno,tol) returns the active and reactive power mismatches at the buses and the estimated active and reactive powers **P** and **Q**. The flag **conv_flag** is set to 1 if the mismatch is less than the specified tolerance **tol**, and is set to 0 otherwise.

Inputs:

nbus - total number of buses
V - row vector of bus voltage magnitudes
ang - row vector of bus voltage angles
Y - admittance matrix in sparse matrix form
Pg - row vector of active power generation
Qg - row vector of reactive power generation
Pl - row vector of active power of load
Ql - row vector of reactive power of load
sw_bno - a vector , formed in **loadflow**, of length nbus. All entries are 1, except for that corresponding to the swing buses which are 0
g_bno - a vector , formed in **loadflow**, of length nbus. All entries are 1, except for that corresponding to the generator buses which are 0
tol - desired accuracy of the solution

Outputs:

delP - row vector of changes in bus active power injection
delQ - row vector of changes in bus reactive power injection
P - estimated active power injection
Q - estimated reactive power injection
conv_flag - 1 if mismatch less than **tol** , 0 if mismatch greater than **tol**

Algorithm:

The current injected into each bus is obtained by pre multiplying the complex bus voltage vector by the Y matrix. From the current and the voltage, the active(**P**) and reactive(**Q**) powers are calculated. The resultant power mismatches are

$$delP = Pg - Pl - P$$
$$delQ = Qg - Ql - Q$$

This algorithm is implemented in the M-file **calc.m** in the POWER SYSTEM TOOLBOX.

See also: **loadflow**

chq_lim

Purpose:

To detect generator reactive power outside limit

Syntax:

f = chq_lim(qg_max,qg_min)

Global variables

Qg - bus generator reactive power
bus_type - vector of bus types:1 for swing bus, 2 for generator, 3 for load
g_bno - vector indicating generator buses, formed in load flow
PQV_no - index of generator buses
PQ_no - index of load buses
ang_red - matrix to eliminate swing bus voltage angles
volt_red - matrix to eliminate swing and generator bus voltage magnitudes
Q - estimated reactive power injection
Ql - reactive power load

Description:

If the generator reactive power is outside the limits specified by **qg_max** and **q_gmin,**
f = chq_lim(qg_max,qg_min) sets

- the generator reactive power to zero
- the bus reactive load to negative of the corresponding limit
- bus_type to 3

It recalculates
- the **ang_red** and **volt_red** which are used to eliminate the swing bus and generator buses from the load flow calculation
- recalculates the generator index

Inputs:

qg_max and **qg_min** are columns 11 and 12 of the bus matrix.

Output:

f is set to zero if no limit reached, or to 1 if a limit is reached.

Algorithm:

The generator reactive powers are compared with the limits following convergence of the Newton-Raphson algorithm in **loadflow**. The function **chk_lim** is then used to determine whether the reactive power limits have been exceeded and to change the load flow configuration if they have.

This algorithm is implemented in the M-file **chk_lim** in the POWER SYSTEM TOOLBOX.

dc_indx

Purpose:

Forms indexes for the rectifier and inverter in the dc load flow
and indicates the ac buses connected to the converters

Syntax:

f = dc_indx(dcsp_con,dcl_con,bus,line)

Global Variables

System:
bus_int - internal bus number index[1]

HVDC:
r_idx i_idx - rectifier and inverter indexes
n_dcl - number of dc lines
n_conv - number of dc converters
ac_bus - internal ac bus numbers associated with converter ac_bus specification
ac_line - index of ac lines connecting HT and LT converter buses
rec_ac_bus - index of rectifier ac buses
inv_ac_bus - index of inverter ac buses
inv_ac_line - index of ac lines connecting HT and LT inverter buses
rec_ac_line - index of ac lines connecting HT and LT rectifier buses
dcli_idx - index of dc lines connected to inverters

Inputs:

dcsp_con dc converter specification matrix[2]
dcl_con dc line specification matrix[2]
bus ac bus specification matrix[3]
line ac line specification matrix[3]

Output:

f is a dummy variable set to zero.

Algorithm:

Identifies the converter LT buses in the ac and dc data files and checks for consistency.

[1] see form_jac
[2] for data format see dc_init
[3] for data format see loadflow

dc_lf

Purpose:

Performs dc load flow

Syntax:

[rec_par,inv_par,line_par,tap,Sr,Si] = dc_lf(dcsp_con,dcl_con,bus,line)

Global variables

System Variables
bus_int - internal bus number index[4]
basmva - system base MVA

DC Variables
r_idx - rectifier index
i_idx - inverter index
n_dcl - number of dc lines
n_conv - number of converters
con_ac_bus - converter ac bus index
rec_ac_bus - rectifier ac bus index
inv_ac_bus - inverter ac bus index
inv_ac_line - inverter ac line index
rec_ac_line - rectifier ac line index
ac_line - converter ac_line index
dcli_idx - dc line index
tap - tap settings
tapr - rectifier transformer tap settings
tapi - inverter transformer tap settings
tmax - maximum tap vector
tmin - minimum tap vector
tstep - tap step vector
tmaxr - rectifier maximum tap
tmaxi - inverter maximum tap
tminr - rectifier minimum tap
tmini - inverter minimum tap
tsepr - rectifier tap step
tsepi - inverter tap step
Vdc - dc voltage vector

[4] See **form_jac**

Description:

[rec_par,inv_par,line_par,tap,Sr,Si] = dc_lf(dcsp_con,dcl_con,bus,line) obtains ac and dc parameters and determines

- the rectifier converter parameters
  - firing angle
  - dc voltage
  - dc power
- the inverter converter parameters
  - extinction angle
  - dc voltage
  - dc power
- the dc line current
- the rectifier and inverter transformer tap settings
- the rectifier and inverter loads at the LT transformer bus

Inputs:

dcsp_con - converter specification matrix (see Table 1)

Table 1 Converter Specification Data

Column Number	Variable	Units
1	HVDC converter Number	
2	LT bus number from bus data	
3	Converter Type 1 - rectifier 2 - inverter	
4	Rated dc voltage	kV
5	Commutating Reactance (Xc)	Ohms per bridge
6	Number of bridges in series	
7	Rectifier - α_{min} Inverter - γ_{min}	degrees
8	Rectifier - α_{max} Inverter - γ_{max}	degrees

dcl_con - dc line specification matrix (see Table 2)

Table 2 DC Line Specification Data

Column Number	Variable	Units
1	Rectifier Number	
2	Inverter Number	
3	dc line resistance	ohms
4	dc line inductance	mH
5	dc line capacitance	μF
6	rectifier smoothing inductance	mH
7	inverter smoothing inductance	mH
8	dc line power rating	MW
9	current margin for inverter current control	%

bus - ac bus specification matrix[5]
line - ac line specification matrix[5]

Outputs:

rec_par - rectifier parameters: firing angle, dc voltage, dc power
inv_par - inverter parameters: extinction angle, dc voltage, dc power
line_par - dc line current
tap - converter transformer tap settings
Sr - complex load at rectifier LT bus
Si - complex power at inverter LT bus

Calls:

inv_lf, rec_lf

Called by:

lf_dc

Algorithm:

The function extracts the ac LT bus data from **bus.** Using this it calculates

- the inverter dc voltage assuming that the inverter extinction angle is set at its minimum value
- the inverter transformer tap ratio required to maintain the dc voltage is within 1% of the specified value
- the dc current from the rated dc line power and the inverter voltage
- the rectifier dc voltage
- the rectifier firing angle
- the rectifier transformer tap ratio required to maintain the firing angle within limits

[5] See **loadflow** for format

If the firing angle cannot be assigned a value greater than its minimum limit, the control mode is changed, and the inverter is set to control current at a value (100- current margin)% of the specified current, with the rectifier firing angle set to its minimum value.

This algorithm is implemented in the M-file **dc_lf.m** in the POWER SYSTEM TOOLBOX.

form_jac

Purpose:

Form the jacobian matrix for loadflow calculation

Synopsis:

Jac = form_jac(V,ang,Y,ang_red,volt_red)

[Jac11,Jac12,Jac21,Jac22] = form_jac(V,ang,Y,ang_red,volt_red)

Description:

form_jac forms the jacobian for the polar form Newton-Raphson method of solving the loadflow problem. All outputs are in sparse matrix form.

Jac = form_jac(V,ang,Y,ang_red,volt_red) returns the sparse jacobian matrix using the admittance matrix Y in sparse matrix form. The jacobian consists of the partial derivatives of active and reactive powers P and Q with respect to the bus voltage magnitudes (p.u.) and angles (rad)

Inputs:

V -	vector of bus voltage magnitudes in p.u.
ang -	vector of bus voltage angles in radians
ang_red -	a transformation matrix, generated in **loadflow**, which eliminates the swing bus voltage magnitude and angle from the Jacobian
volt_red -	a transformation matrix, generated in **loadflow**, which eliminates the generator bus voltage magnitudes from the Jacobian

Outputs:

Jac11 - $\frac{\partial P}{\partial \delta}$, swing bus angle eliminated

Jac12 - $\frac{\partial P}{\partial V}$, generator and swing bus voltages eliminated

Jac21 - $\frac{\partial Q}{\partial \delta}$, swing bus angle eliminated

Jac22 - $\frac{\partial Q}{\partial V}$, generator and swing bus voltages eliminated

The dimensions of the individual Jacobian matrices are n by n, n by m, m by n and m by m respectively. Where n is the number of non-swing buses, and m is the number of buses which are neither swing nor generator buses.

Algorithm:

The elements of the jacobian are obtained by taking partial derivatives of active and reactive powers of all buses in polar coordinates. Row and column reductions are then applied to eliminate those entries corresponding to specified components of bus angle and voltage.

This algorithm is implemented in the M-file **form_jac** in the POWER SYSTEM TOOLBOX.

See also: loadflow, y_sparse

Example:

Consider a 2-machine 3-bus system having bus 1 as the swing bus and bus 2 as the generator bus. In this case, V(1), V(2) and ang(1) are fixed and are eliminated from the Jacobian using

$$Jac11 = ang\_red * \partial P / \partial \delta * ang\_red'$$
$$Jac12 = ang\_red * \partial P / \partial V * volt\_red'$$
$$Jac21 = volt\_red * \partial Q / \partial \delta * ang\_red'$$
$$Jac22 = volt\_red * \partial Q / \partial V * volt\_red'$$
$$ang\_red = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$volt\_red = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

To obtain the Jacobian, call

Jac = form_jac(V,ang,Y,ang_red,v_red)

$$Y = \begin{bmatrix} 0.3+i & 0 & -0.3-i \\ 0 & i & -i \\ -0.3-i & -i & 0.3+2i \end{bmatrix}$$

$$V = \begin{bmatrix} 1. & 1.04 & 1.02 \end{bmatrix}$$

$$ang = \begin{bmatrix} 0 & 0.3 & -0.1 \end{bmatrix}$$

$$Jac = \begin{bmatrix} -0.9771 & 0.9771 & -0.4050 \\ 0.9771 & -2.0225 & 0.8183 \\ -0.4131 & 0.2105 & -2.0971 \end{bmatrix}$$

$$Jac11 = \begin{bmatrix} -0.9771 & 0.9771 \\ 0.9771 & -2.0225 \end{bmatrix}$$

$$Jac12 = \begin{bmatrix} -0.4050 \\ 0.8183 \end{bmatrix}$$

$$Jac21 = \begin{bmatrix} -0.4131 & 0.2105 \end{bmatrix}$$

$$Jac22 = -2.0971$$

inv_lf

Purpose:

Performs calculations to determine the dc load flow conditions at the inverter

Syntax:

[gamma] = inv_lf(mode,idc,Vdc_max,Vdc_min,ga_min,ga_max,Vdo,Rc)

Global variables

i_idx - inverter index
tapi - inverter transformer tap ratio
tmaxi - inverter transformer maximum tap ratio
tmini - inverter transformer minimum tap ratio
tstepi - inverter transformer tap step
Vdc - dc voltage vector

Description:

[gamma] = inv_lf(mode,idc,Vdc_max,Vdc_min,ga_min,ga_max,Vdo,Rc) determines the inverter extinction angle angle from bus details at the equivalent HT bus (calculated from the LT bus conditions in the ac load flow). Normally (mode 1) the extinction angle is set to γ_{min} and the transformer tap setting altered to ensure that the inverter dc voltage is within 1% of its rated value. If the rectifier is unable to control current, the inverter takes over this responsibility, and the extinction angle is set to control current.

Inputs:

mode - 1 for rectifier current control, $\gamma = \gamma_{min}$, 2 for inverter current control
idc - desired dc current
Vdc_max - maximum allowed dc voltage in mode 1
Vdc_min - minimum allowed dc voltage in mode 1
ga_min - minimum extinction angle
ga_max - maximum extinction angle
Vdo - Equivalent HT voltage vector (commutating voltage)
Rc - commutating resistance

Outputs:

gamma - inverter extinction angle

Algorithm:

For mode = 1

The dc voltage is calculated from

$$V_{dc} = V_{do}\cos(g_{\min}) - R_c i_{dc}$$

idc is the desired dc current set by the rated power/Vdc

If V_{dc} is within the limits set by **Vdc_max** and **Vdc_min,** γ is set to γ_{min}.
If V_{dc} is outside the limits, the inverter transformer taps are reset this alters the value of **Vdo**
If this is impossible within the specified tap range, an error is flagged

For mode = 2

idc is the desired current reduced by the current margin
Vdc is set by the rectifier at minimum firing angle

The inverter extinction angle is calculated from

$$\cos(g) = (V_{dc} + R_c i_{dc}) / V_{do}$$

If γ is out of limits, the inverter transformer taps are reset to bring it within range
If this is not achieved within the tap range an error is flagged

lf_tap

Purpose:

Sets transformer taps to put low side voltage within limits

Syntax:

lf_tap

Description:

lf_tap is a MATLAB script file used within **loadflow**. It calculates the tap setting required to keep the LT (to) bus voltage within the voltage limits. It assumes that the HT (from) bus voltage remains unchanged.

Algorithm:

The LT bus voltage magnitude is checked. If it is out of limits, the error between the voltage and the limit is calculated and used to alter the tap setting so as to reduce the error. If the tap limits are reached an error is flagged.

This algorithm is implemented in the M-file **lf_tap.m** in the POWER SYSTEM TOOLBOX.

lfdc

Purpose:

Driver script for combined ac and dc load flow

Syntax:

lfdc

Global variables

System Variables
bus_int - internal bus number index
basmva - system base MVA

DC Variables
r_idx - rectifier index
i_idx - inverter index
n_dcl - number of dc lines
n_conv - number of converters
con_ac_bus - converter ac bus index
rec_ac_bus - rectifier ac bus index
inv_ac_bus - inverter ac bus index
inv_ac_line - inverter ac line index
rec_ac_line - rectifier ac line index
ac_line - converter ac_line index
dcli_idx - dc line index
tap - tap settings
tapr - rectifier transformer tap settings
tapi - inverter transformer tap settings
tmax - maximum tap vector
tmin - minimum tap vector
tstep - tap step vector
tmaxr - rectifier maximum tap
tmaxi - inverter maximum tap
tminr - rectifier minimum tap
tmini - inverter minimum tap
tsepr - rectifier tap step
tsepi - inverter tap step
Vdc - dc voltage vector

Description:

Inputs data files and performs ac loadflow and dc load flow in turn until both are converged

Calls

loadflow - for ac load flow solution
dc_lf - for dc load flow solution

loadflow

Purpose:

Solve loadflow equations of a power system using the Newton-Raphson method in polar coordinates

Synopsis:

[bus_sol,line_sol,line_flw] = loadflow(bus,line,tol,iter_max,acc,display,flag)

Description:

[bus_sol,line_sol,line_flw] = loadflow(bus,line,tol,iter_max,acc,display,flag)
returns a load flow solution of a power system using the bus data **bus** and line data **line**. The Newton-Raphson algorithm terminates if the largest bus power mismatch is less than the tolerance **tol** or the number of iterations exceeds **iter_max**.

Global Variables

bus_int - internal bus number index
Qg - generator reactive power vector
bus_type - 1 for swing bus; 2 for generator bus; 3 for load bus
g_bno - vector to identify generator buses
PQV_no - generator bus index
PQ_no - load bus index
ang_red - matrix to eliminate swing bus angles
volt_red - matrix to eliminate swing and generator bus voltages
Q - reactive power at bus
Ql - load reactive power at bus

Inputs:

bus - ac bus specification matrix
line - ac line specification matrix
tol - the load flow tolerance
iter_max - the maximum number of load flow iterations
acc - the acceleration (**acc** >1) or deceleration (**acc** <1) factor for improving the convergence of the Newton-Raphson iteration.
display - **'y'** creates a load flow report, **'n'** suppresses a load flow report
flag - 1, a new Jacobian is calculated at each iteration

Outputs:

bus_sol - contains the solved power flow bus data
line_sol - contains the solved power flow line data
line_flw- contains the solved line power flows

Table 1 Bus Specification Matrix Format

column	variable	unit
1	bus number	
2	voltage	pu
3	angle	pu
4	P generation	pu on system base
5	Q generation	pu on system base
6	P load	pu on system base
7	Q load	pu on system base
8	G shunt	pu on system base
9	B shunt	pu on system base
10	bus type	1, swing bus 2, generator (PV) bus 3, load (PQ) bus
11	Q gen max	pu on system base
12	G gen min	pu on system base
13	rated bus voltage	kV
14	Maximum bus voltage	pu
15	Minimum bus voltage	pu

Table 2 Line Specification Matrix Format

column	variable	unit
1	from bus	
2	to bus	
3	resistance	pu
4	reactance	pu
5	line charging	pu
6	tap ratio	0 - no tap changing
7	phase shifter angle	degrees
8	maximum tap ratio	
9	minimum tap ratio	
10	tap step	

It is assumed that the tap and the phase shifter are located at the **from** bus. There is no need to input tap and phase shifter data if there are none.

Calls:
y_sparse, calc, form_jac, lf_tap

Example:

The bus and line data of a 4 generator, 2 area system [1] are

```
bus = […
1  1.03     18.5  7.00   1.61  0.00  0.00  0.00  0.00  1  99.0  -99.0  22.0
1.1  .9;
2  1.01     8.80  7.00   1.76  0.00  0.00  0.00  0.00  2  5.0   -2.0   22.0
1.1  .9;
3  0.9781  -6.1   0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   500.0
1.5  .5;
4  0.95    -10    0.00   0.00  9.76  1.00  0.00  0.00  3  0.0   0.0   115.0
1.05 .95;
10  1.0103  12.1  0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5;
11  1.03   -6.8   7.16   1.49  0.00  0.00  0.00  0.00  2  5.0  -2.0    22.0
1.1  .9;
12  1.01   -16.9  7.00   1.39  0.00  0.00  0.00  0.00  2  5.0  -2.0    22.0
1.1  .9;
13  0.9899 -31.8  0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   500.0
1.5  .5;
14  0.95   -38    0.00   0.00  17.67 1.00  0.00  0.00  3  0.0   0.0   115.0
1.05 .95;
20  0.9876  2.1   0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5;
101 1.05   -19.3  0.00   8.00  0.00  0.00  0.00  0.00  2  99.0  -99.0 500.0
1.5  .5;
110 1.0125 -13.4  0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5;
120 0.9938 -23.6  0.00   0.00  0.00  0.00  0.00  0.00  3  0.0   0.0   230.0
1.5  .5];

line = [...
1   10  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
2   20  0.0     0.0167   0.00    1.0  0. 0.  0.  0.;
3    4  0.0     0.005     0.00   1.0  0. 1.2 0.8 0.05;
3   20  0.001   0.0100   0.0175  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
3   101 0.011   0.110    0.1925  1.0  0. 0.  0.  0.;
10  20  0.0025  0.025    0.0437  1.0  0. 0.  0.  0.;
11  110 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
12  120 0.0     0.0167   0.0     1.0  0. 0.  0.  0.;
13   14 0.0     0.005    0.00    1.0  0. 1.2 0.8 0.05;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  101 0.011   0.11     0.1925  1.0  0. 0.  0.  0.;
13  120 0.001   0.01     0.0175  1.0  0. 0.  0.  0.;
110 120 0.0025  0.025    0.0437  1.0  0. 0.  0.  0.];
```

The report generated by

[bus_sol,line_sol,line_flw] = loadflow(bus,line,1e-6,30,1,'y',1)

is given on the following page. Note that the timing of the computation is based on a 100 MHz Pentium PC.

Algorithm:

In polar coordinates, the unknown variables are the bus voltage magnitudes and angles of the load buses. A jacobian in terms of these variables is generated at each iteration and used to update the magnitude and angle variables. Sparse matrices are used to set up the jacobian to reduce storage and to speed up the solution algorithm.

This algorithm is implemented in the M-file **loadflow** in the POWER SYSTEM TOOLBOX.

See also: ybus, calc, form_jac, pst_var

Load Flow Report

lfdemo run with data2a.m

Solution Progress

```
voltage low changing tap on line
     3

taps reset to

tap =

    0.9500

voltage low changing tap on line
    10

taps reset to

tap =

    0.9500

voltage low changing tap on line
    10

taps reset to

tap =

    0.9000
```

Solution Summary

```
                             LOAD-FLOW STUDY
                    REPORT OF POWER FLOW CALCULATIONS

3-Dec-96
SWING BUS                  : BUS 1
NUMBER OF ITERATIONS       : 6
SOLUTION TIME              : 0.22 sec.
TOTAL TIME                 : 2.92 sec.
TOTAL REAL POWER LOSSES    : 0.943795.
TOTAL REACTIVE POWER LOSSES: 14.377.
```

			GENERATION		LOAD	
BUS	VOLTS	ANGLE	REAL	REACTIVE	REAL	REACTIVE
1.0000	1.0300	18.5000	7.2138	2.0926	0	0
2.0000	1.0100	8.1584	7.0000	2.8023	0	0
3.0000	0.9485	-7.3757	0	0	0.0000	0.0000
4.0000	0.9921	-10.1997	0	0	9.7600	1.0000
10.0000	1.0029	11.8028	0	0	0.0000	0.0000
11.0000	1.0300	-6.9696	7.1600	2.5494	0	0
12.0000	1.0100	-17.3151	7.0000	3.9297	0	0
13.0000	0.9151	-33.3468	0	0	0.0000	0.0000
14.0000	1.0081	-38.2916	0	0	17.6700	1.0000
20.0000	0.9706	1.3096	0	0	0.0000	0.0000
101.0000	1.0500	-21.0220	0.0000	5.0029	0	0
110.0000	0.9955	-13.6667	0	0	0.0000	0.0000
120.0000	0.9521	-24.2977	0	0	0.0000	0.0000

LINE FLOWS

LINE	FROM BUS	TO BUS	REAL	REACTIVE
1.0000	1.0000	10.0000	7.2138	2.0926
2.0000	2.0000	20.0000	7.0000	2.8023
3.0000	3.0000	4.0000	9.7600	1.4890
4.0000	3.0000	20.0000	-13.8668	0.3349
5.0000	3.0000	101.0000	2.0534	-0.9119
6.0000	3.0000	101.0000	2.0534	-0.9119
7.0000	10.0000	20.0000	7.2138	1.2045
8.0000	11.0000	110.0000	7.1600	2.5494
9.0000	12.0000	120.0000	7.0000	3.9297
10.0000	13.0000	14.0000	17.6700	2.5412
11.0000	13.0000	101.0000	-1.9373	-0.8076
12.0000	13.0000	101.0000	-1.9373	-0.8076
13.0000	13.0000	120.0000	-13.7954	-0.9259
14.0000	110.0000	120.0000	7.1600	1.6401
1.0000	10.0000	1.0000	-7.2138	-1.2045
2.0000	20.0000	2.0000	-7.0000	-1.8716
3.0000	4.0000	3.0000	-9.7600	-1.0000
4.0000	20.0000	3.0000	14.0807	1.7879
5.0000	101.0000	3.0000	-1.9935	1.3181
6.0000	101.0000	3.0000	-1.9935	1.3181
7.0000	20.0000	10.0000	-7.0807	0.0837
8.0000	110.0000	11.0000	-7.1600	-1.6401
9.0000	120.0000	12.0000	-7.0000	-2.8748
10.0000	14.0000	13.0000	-17.6700	-1.0000
11.0000	101.0000	13.0000	1.9935	1.1833
12.0000	101.0000	13.0000	1.9935	1.1833
13.0000	120.0000	13.0000	14.0237	3.1933
14.0000	120.0000	110.0000	-7.0237	-0.3185

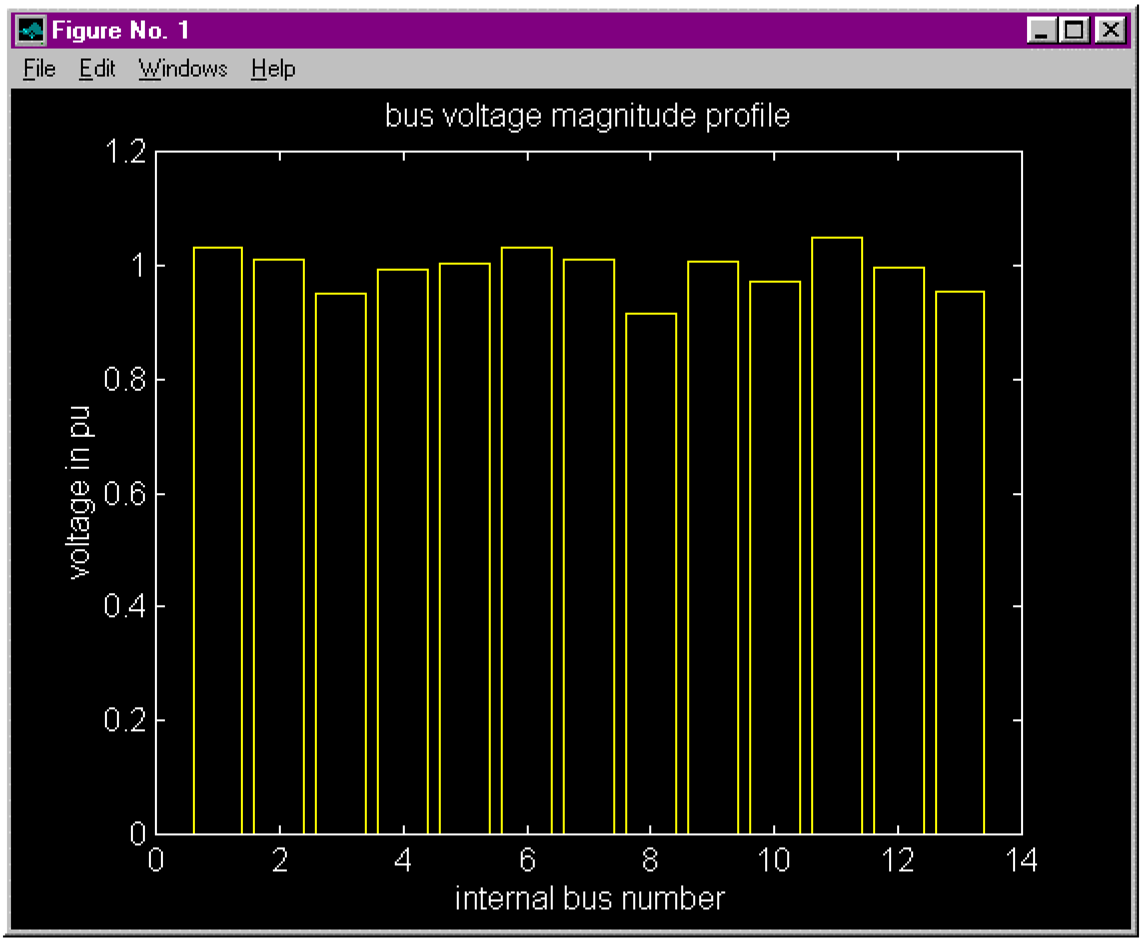


Figure No. 1
File Edit Windows Help
bus voltage magnitude profile
voltage in pu
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14
internal bus number

rec_lf

Purpose:

Performs calculation to determine the dc load flow conditions at the recifier

Syntax:

[alpha,mode,idc_new] = rec_lf(idc,al_min,al_max,Vdo,Rc,cm)

Global variables

r_idx - rectifier index
i_idx - inverter index
n_dcl - number of dc lines
tapr - rectifier transformer tap settings
tmaxr - maximum rectifier transformer tap ratio
tminr - maximum rectifier tap ratio
tstepr - rectifier transformer tap step
Vdc - dc voltage vector

Description:

[alpha,mode,idc_new] = rec_lf(idc,al_min,al_max,Vdo,Rc,cm) calculates the firing angle (α) necessary to maintain the desired dc current (**idc**). If the calculated value of α is out of range, the rectifier transformer taps are reset to bring it within range. If this is not possible and α is less than α_{min} the mode is changed to 2, idc_new is set to idc less the current margin, and α is set to α_{min}. If changing the taps will not make $\alpha<\alpha_{max}$, an error is flagged.

Inputs:

idc- desired dc current
al_min- minimum firing angle
al_max- maximum firing angle
Vdo- Equivalent HT voltage (commutating voltage)
Rc- Commutating resistance
cm- current margin %

Outputs:

alpha - rectifier firing angle
mode - 1 for recifier controlling current, 2 for inverter controlling current
idc_new- desired dc current (**idc***(1-**cm**/100))

Algorithm:

Mode is assumed to be 1 initially

the firing angle is calculated from

$$\cos(a) = (V_{dc} + R_c i_{dc}) / V_{do}$$

If α is outside limits the transformer taps are changed to bring it within range
If this is not possible, and $\alpha<\alpha_{min}$, **mode** =2, **idc_new** is set to **idc** less the current margin, $\alpha=\alpha_{min}$, and the dc voltage is calculated as

$$V_{dc} = V_{do} \cos(a_{\min}) - R_c i_{dc\_new}$$

This algorithm is implemented in the M-file **rec_lf.m** in the POWER SYSTEM TOOLBOX.

y_sparse

Purpose:

Construct a sparse admittance matrix for load flow solution using matrix operations.

Synopsis:

[Y,nSW,nPV,nPQ,SB] = y_sparse(bus,line)

Description:

[Y,nSW,nPV,nPQ,SB] = y_sparse(bus,line) uses the bus data and the line data in matrices **bus** and **line** to return the admittance matrix **Y** in sparse matrix form. Forms the global index matrix **bus_int.**

Inputs:

bus - ac bus specification matrix
line - ac line specification matrix

Outputs:

Y - sparse Y matrix for power system network
nSW - number of swing buses
nPV - number of generator buses
nPQ - number of load buses
SB - bus number of the swing bus

Global Variable

bus_int - a matrix index of internal buses

Algorithm:

The index matrix **bus_int** is a vector of length equal to the highest bus number specified in the first column of **bus**. It is zero everywhere except for the locations corresponding to the specified bus numbers. At these locations, the entry is the corresponding row number of **bus**. Thus **bus_int(bus(:,1))** gives the vector **[1 2 3 4 …..number of buses]'**.

This algorithm is implemented in the M-file **y_sparse** in the POWER SYSTEM TOOLBOX.