Computation of Maximum loadability limit in MATPOWER

Camille Hamon

May 19, 2017

Contents

1 Examples

1.1 IEEE 9 bus system

1.1.1 With reactive power limits

The following code looks for the maximum loadability point limit in a the direction of load increase defined by `dir_mll`.

```
%   the IEEE 9

%   MATPOWER
%   Copyright (c) 2015-2016, Power Systems Engineering Research
   Center (PSERC)
%   by Camille Hamon
%
%   This file is part of MATPOWER.
%   Covered by the 3-clause BSD License (see LICENSE file for
   details).
%   See http://www.pserc.cornell.edu/matpower/ for more info.
```

The printed results are shown in Figure 1. The first block of information gives information about the stability margin to the maximum loadability limit (MLL) point and the type of bifurcation at the MLL point. In this example, the MLL is a limit-induced bifurcation that occurs when generator 2 reaches its reactive power limit. The second block of information gives the participation of all buses in the direction of load increase, the loads and the voltage magnitudes at all buses at the MLL. The third block of information shows the reactive power generation at all generators as compared to their limits, and the voltage magnitude at the generator buses as compared to their set points. A flag `REF` indicates which generator is the slack. In the case of a limit-induced bifurcation, a flag `LIB` indicates the generator which generator caused the LIB when reaching its reactive power limit.

1.1.2 Without reactive power limits enforced

In this example, the reactive power limits are not enforced in the search for the MLL point. This is done by setting the flag `use_qlim` to 0.

```

% Loading the system and defining parameters
mpc = loadcase('case9'); % load ieee9
```

The results are shown in Figure 2. Since the reactive power limits are not enforced, the maximum loadability limit point is now a saddle-node bifurcation point (SNB). It can be seen that the reactive power generation at the generator at bus 2 is 341.64 Mvar which is larger than its limit (300 Mvar).

```
=======================================
      Maximum loadability problem
=======================================
The stability margin is 339.53 MW
The type of bifurcation is limit-induced bifurcation (LIB).
Generator responsible for LIB: Gen 2 connected at bus 2

--------------------------------------------------
   Bus nb    Direction     Load at MLL    Voltages
   ------    ---------     -----------    --------
      1          0            0.00          1.000
      2          0            0.00          1.000
      3          0            0.00          1.000
      4          0            0.00          0.867
      5          0           90.00          0.806
      6          0            0.00          0.852
      7          1          439.53          0.663
      8          0            0.00          0.819
      9          0          125.00          0.784

=============================================================
Reactive power production and voltages at the generators
=============================================================
   Bus nb      Qgen       Qmin       Qmax        Vm      Vref
  --------    -------    ------     ------      -----    -----
      1        305.15    -300.00    9999.00    1.0000   1.0000    REF
      2        300.00    -300.00     300.00    1.0000   1.0000    LIB
      3        255.64    -300.00     300.00    1.0000   1.0000
```

Figure 1: Results of the search for the MLL in the IEEE 9 bus system with reactive power limit enforced.

1.1.3 Comparison with the MATPOWER CPF function

The continuation power flow is implemented in the MATPOWER function `runcpf`. It does not consider reactive power limits. The following code runs the continuation power flow in the same direction of load increase as in the previous section (load increase at bus 7).

```

%% With reactive power limits (default)
% Running the search for MLL with verbose set to 1 to print the
    results.
results_mll = maxloadlim(mpc,dir_mll,'verbose',1); %
```

```

%% Without enforcement of the reactive power limits.
% The reactive power limits are not enforced in the following
   example.
results_mll = maxloadlim(mpc,dir_mll,'verbose',1,'use_qlim',0);

% The same MLL point can be obtained by MATPOWER implementation of
    the CPF.
define_constants;
mpc_target = mpc;
nonzero_loads = mpc_target.bus(:,PD) ~= 0;
```

```
========================================
      Maximum loadability problem
========================================
The stability margin is 342.37 MW
The type of bifurcation is saddle-node bifurcation (SNB).

---------------------------------------------------
   Bus nb    Direction     Load at MLL    Voltages
   ------    ---------     -----------    --------
      1          0             0.00         1.000
      2          0             0.00         1.000
      3          0             0.00         1.000
      4          0             0.00         0.846
      5          0            90.00         0.776
      6          0             0.00         0.830
      7          1           442.37         0.620
      8          0             0.00         0.793
      9          0           125.00         0.753

==============================================================
Reactive power production and voltages at the generators
==============================================================
   Bus nb      Qgen       Qmin        Qmax        Vm       Vref
  --------    -------    ------      ------      -----     -----
      1        348.43    -300.00     9999.00    1.0000    1.0000    REF
      2        341.64    -300.00     9999.00    1.0000    1.0000
      3        293.07    -300.00     9999.00    1.0000    1.0000
```

Figure 2: Results of the search for the MLL in the IEEE 9 bus system without enforcement of the reactive power limits.

```
Q_P = mpc_target.bus(nonzero_loads,QD)./mpc_target.bus(
    nonzero_loads,PD);
mpc_target.bus(:,PD) = mpc_target.bus(:,PD)+2*dir_mll*mpc_target.
    baseMVA;
mpc_target.bus(nonzero_loads,QD) = Q_P.*mpc_target.bus(
    nonzero_loads,PD);
% Run the CPF with matpower
[results,~] = runcpf(mpc,mpc_target);
```

The results are shown in Figure 3. Comparing with the results in Figure 2, it can be seen that both algorithms lead to very similar results; the small differences can be explained by the step length of the CPF algorithm.

```
================================================================================
|     Bus Data                                                                 |
================================================================================
 Bus      Voltage          Generation             Load
  #   Mag(pu) Ang(deg)   P (MW)   Q (MVAr)   P (MW)   Q (MVAr)
----- ------- --------  --------  --------  --------  --------
    1  1.000    0.000*   482.24    362.02       -         -
    2  1.000  -50.453    163.00    355.52       -         -
    3  1.000  -55.236     85.00    305.48       -         -
    4  0.839  -19.338       -         -         -         -
    5  0.766  -37.631       -         -       90.00     30.00
    6  0.822  -58.708       -         -         -         -
    7  0.605  -79.871       -         -      442.07    154.72
    8  0.784  -57.915       -         -         -         -
    9  0.743  -38.424       -         -      125.00     50.00
                        --------  --------  --------  --------
               Total:    730.24   1023.02    657.07    234.72
```

Figure 3: Results of the MATPOWER implementation of continuation power flow.

2 Notations

If not indicated otherwise, the notations are the same as in MATPOWER's manual.

3 The maximum loadability limit problem

Finding the maximum loadability limit from base-case loading $P_d^0 \in \mathbb{R}^{n_b}$ in a load increase direction $d \in \mathbb{R}^{n_b}$ can be done by solving the following optimisation problem

$$
\begin{aligned}
\max_{z} \quad & \alpha && \text{(1a)}\\
\text{subject to} \quad & P_d = P_d^0 + \alpha d && \text{(1b)}\\
& Q_d = \operatorname{diag}(\tan\phi^0) P_d && \text{(1c)}\\
& g(z) = 0 && \text{(1d)}\\
& z_{\min} \le z \le z_{\max} && \text{(1e)}
\end{aligned}
$$

where

$$
z = [x^T \ \alpha]^T = \begin{bmatrix} \Theta \\ V_m \\ P_g \\ Q_g \\ \alpha \end{bmatrix}, \tag{2}
$$

where x is defined in Equation (6.5) in MATPOWER's manual and the function g is defined is Equations (4.2) and (4.3) in MATPOWER's manual. Equation (1c) ensures that the power factor of the loads is kept constant to $\cos(\phi_i^0)$, the value defined in the base case. Note that $\tan\phi^0 \in \mathbb{R}^{n_b}$. The limits (1e) are as follows:

$$
\begin{aligned}
\theta_i &= \theta_i^{\text{ref}}, && i \in \mathcal{I}_{\text{ref}} && (3)\\
v_m^i &= v_m^{i,\text{ref}}, && i \in \mathcal{I}_{\text{ref}} && (4)\\
0 \le & v_m^i \le v_m^{i,\text{ref}}, && i \in \{1,\ldots,n_g\}\backslash\mathcal{I}_{\text{ref}} && (5)\\
q_g^{i,\min} \le & q_g^i \le q_g^{i,\max}, && i \in \{1,\ldots,n_g\}\backslash\mathcal{I}_{\text{ref}} && (6)\\
p_g^i &= p_g^{i,0}, && i \in \{1,\ldots,n_g\}\backslash\mathcal{I}_{\text{ref}} && (7)\\
\alpha &\in [0,+\infty]. && && (8)
\end{aligned}
$$

where $p_g^{i,0}$ is the active power generation at bus i in the base case. Equations (3) and (4) lock the voltage angle and magnitude of the reference buses to their set points. Equations (5) and (6) limit the terminal voltages and reactive power generation at the PV buses. Note that in theory, complementarity constraints

$$
(v_m^i - v_m^{i,\text{ref}}) \cdot (q_g^i - q_g^{i,\max}), \quad i \in \{1,\ldots,n_g\}\backslash\mathcal{I}_{\text{ref}} \tag{9}
$$

should be included. However, in practice, solvers will set $v_m^i = v_m^{i,\text{ref}}$ if q_g^i is smaller than $q_g^{i,\max}$ and let v_m^i be free if q_g^i has reached its limit, since doing so maximizes loadability.

4 Implementation in MATPOWER

4.1 Inputs

We assume that the following inputs are available:

1. A Matpower base case `mpc`.

2. A load increase direction $d \in R^{n_b}$, with the assumption that only nonzero loads in `mpc` can be increased.

4.2 Defining dispatchable loads

A new Matpower case `mpc_vl` is built where all nonzero loads in `mpc` are transformed to dispatchable loads:

```
mpc_vl = load2disp(mpc)
```

Dispatchable loads are described in Section 6.4.2 of Matpower's package. Note that, conceptually, this means that the active parts of the nonzero loads in P_d enter P_g in (2) and their reactive parts enter Q_g.

4.3 Adjusting the Matpower case

4.3.1 Power constraints of dispatchable loads

In Matpower, dispatchable loads are assumed to maintain a constant power factor. This power factor is determined by the values in the columns `PMIN` and `QMIN` corresponding to the dispatchable loads in the field `mpc_vl.gen`. When using the function `load2disp`, these fields are set to the opposite of the active and reactive power parts of the nonzero loads in the base case `mpc`. When solving the optimisation problem (1) with Matpower, the columns `PMIN` and `QMIN` also constrain the maximum values of dispatchable loads. Since we are looking for maximum loadability limits, upper values on loads must not be binding. Therefore, the values `PMIN` of rows in `mpc_vl.gen` corresponding to dispatchable loads is set to a very large negative values and the values `QMIN` of these dispatchable loads are changed accordingly to keep the constant power factor defined in the base case.

4.3.2 Power constraints on generators

The maximum loadability limit problem differs from a standard AC OPF as defined in Equations (6.1) to (6.4) in Matpower's manual in the following way

1. Active power production at PV buses is locked to the base case values. Therefore, the columns `PMIN` and `PMAX` of these generators are set to their active power production in the base case.

2. Generators at reference buses pick up the loads during the load increase process. The assumption here is that the reference buses correspond to strong generators that do not reach their power limits. Therefore, active and reactive power limits of the generators at reference buses are set to large values.

4.3.3 Voltage constraints

Similarly to power constraints discussed above, voltage constraints in the maximum loadability problem differ from a standard AC OPF as follows

1. Voltage magnitudes at the reference buses are locked at their set point values, see Equation (4).
2. Voltage magnitudes at all PV and PQ buses are not constrained from below. The reason for this is that the maximum loadability limit point typically corresponds to an operating point that is outside the normal allowed band on bus voltage magnitude. Therefore, to find this point, lower limits on bus voltage magnitudes of PV and PQ buses must be relaxed.
3. For the same reason, upper bounds on voltage magnitudes of PQ buses are also relaxed.
4. Upper bounds on voltage magnitudes of PV buses are set to the voltage set point of the corresponding generators.

4.3.4 Branch flows

For the same reason as for bus voltages, the branch flow constraints must be relaxed so that they do not become binding during the optimisation process.

4.3.5 Decision variable α and additional constraints

Compared to the standard AC OPF formulation in Matpower, the formulation in (1) contains an additional decision variable α, see (2), and the additional constraints (1b). Note that constraints (1c) does not exist in the standard AC OPF formulation either, but Matpower assumes a constant power factor model for the dispatchable loads, see also Section 4.3.1. Therefore, there is no need to define the additional constraint (1c). The decision variable α and the constraint (1b) are defined using a callback function for the `formulation` stage.

Following Section 7.2 in Matpower's manual, we create a callback function called `userfcn_direction_mll_formulation(om,args)` in which we define the new variable and constraints. To define the new constraint, recall that when transforming the loads to dispatchable loads, the nonzero loads of the base case enter the set of generators, see Section 4.2. Matpower has a feature to only define the additional constraints on the relevant components of the decision variable z in (2). For Equation (1b), the relevant component of z are P_g (since the dispatchable loads enter P_g) and α.

4.3.6 Costs

When solving an OPF with dispatchable loads in Matpower, the costs of dispatchable loads is added to that of the generator production costs. Since the objective function of the maximum loadability limit is just α (see (1a)), we are not interested in keeping track of the cost of dispatchable loads and generators. These costs are therefore set to zero.

The cost for α is defined in the callback function introduced in Section 4.3.5. It is set to -1 since the objective in (1a) is to maximize α, whereas MATPOWER performs a minimisation.

4.3.7 Functions

The functions in Table 1 are used to set up and solving the maximum loadability limit problem with MATPOWER.

Name	Description
`maxloadlim`	Main function for solving the maximum loadability limit problem.
`postproc_maxloadlim`	Post-processing function that transforms the dispatchable loads back to normal loads.
`prepare_maxloadlim`	Prepares the MATPOWER case before the AC OPF is run, see Section 4.3.
`print_maxloadlim`	Pretty prints the results from the maximum loadability problem.
`userfcn_direction_mll_ext2int`	Converts the vector of generator production changes to internal indexing.
`userfcn_direction_mll_formulation`	Callback function to define new variables, constraints and costs, see Section 4.3.5.

Table 1: Functions used for the maximum loadability limit problem