<table>
<thead>
<tr>
<th>Title</th>
<th>A smoothing SQP method for nonlinear programs with stability constraints arising from power systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Tong, X; Qi, L; Wu, SY; Wu, FF</td>
</tr>
<tr>
<td>Citation</td>
<td>Computational Optimization And Applications, 2012, v. 51 n. 1, p. 175-197</td>
</tr>
<tr>
<td>Issued Date</td>
<td>2012</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/144895">http://hdl.handle.net/10722/144895</a></td>
</tr>
<tr>
<td>Rights</td>
<td>The Author(s); This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.</td>
</tr>
</tbody>
</table>
Abstract This paper investigates a new class of optimization problems arising from power systems, known as nonlinear programs with stability constraints (NPSC), which is an extension of ordinary nonlinear programs. Since the stability constraint is described generally by eigenvalues or norm of Jacobian matrices of systems, this results in the semismooth property of NPSC problems. The optimal conditions of both NPSC and its smoothing problem are studied. A smoothing SQP algorithm is proposed for solving such optimization problem. The global convergence of algorithm is established. A numerical example from optimal power flow (OPF) is done. The computational results show efficiency of the new model and algorithm.
Keywords Power systems · Stability constraint · Nonlinear programs · Smoothing SQP method · Convergence

1 Introduction

In this paper, we consider a class of optimization problems

$$\min f(x, u)$$

s.t. $$h(x, u) = 0,$$

$$g(x, u) \leq 0,$$

$$\phi(x, u) \geq \delta > 0,$$

(1.1)

where $$\delta > 0$$ is a small constant vector; $$x \in \mathbb{R}^n$$ stands for the state variable; $$u \in \mathbb{R}^m$$ represents a control variable; $$f : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$$; $$h(x, u) = (h_1(x, u), h_2(x, u), \ldots, h_n(x, u))^T : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$$; $$g(x, u) = (g_1(x, u), g_2(x, u), \ldots, g_l(x, u))^T : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^l$$ are twice continuously differentiable; $$\phi(x, u) : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^s$$ is locally Lipschitz, possibly semismooth (see [17] for the semismooth concept). Here the last inequality constraint describes the stability property of dynamic systems. We call (1.1) Nonlinear Programs with Stability Constraints (NPSC) throughout this paper.

Problem (1.1) has a strong background and arises from power systems. The OPF is a fundamental tool of power systems for planning and operation. The classical OPF is an ordinary nonlinear programming problem, which has been studied widely, and many solution methods have been proposed and implemented over the years [6, 7, 19–21, 23]. On the other hand, it is well-known that power systems are described by differential and algebraic equations (DAEs), and they belong to dynamic systems. One of the concerned issues for such systems is the static stability at the equilibrium point (see [13]), which is the ability to recover the normal operation under some small disturbances. We note that classical OPF models consider little about system stability. This indicates that the solution of classical OPF cannot ensure the stable operation, especially for the current market mechanism where it requires the system operation under economic, stable and secure properties. Hence, it is necessary to extend the OPF models and algorithms with stability consideration. This is our motivation for this paper.

Both static and transient stability of power systems without considering optimal operation, have been studied extensively in the literature (see [2, 3, 10, 13, 14, 24]). In this paper, we are interested in static stability (also called small signal stability), which is not related to the dynamic operation of the system. Whereas it is associated with eigenvalues or singular values of Jacobian of function $$h$$ in (1.1) (see [13, 18]). Recently, the study of OPF with static stability receives many attention in power system analysis. For example, Canizares and his research group have made a series of investigations (see [1, 12] and references therein). Their main research topic focuses on calculating technology with respect to bifurcation analysis of power systems, such as the saddle-node bifurcation, the Hopf bifurcation, and the limit-induced bifurcation. We also note that in the existing study of OPF with stability constraints, the stability requirement implements are mainly from physical properties of systems, or
A smoothing SQP method for nonlinear programs

via multi-layer optimization models (see [25] and its comments), which may not ad-
dress the stability exactly or may result in some calculation difficulty. Moreover, the
research of general static stability, i.e., considering negative real part of eigenvalues
of Jacobian matrices, has little progress since the eigenvalue function is complicated.
According to the system requirement and the research progress, we will set up a new
mathematical model to describe the steady stability of power systems, as well as some
effective algorithms for solving such model.

Some stability models and solution methods of dynamic systems with respect to
nonlinear equations are studied in our previous work [11]. In this paper, the semi-
smooth methodology is used for solving stability equilibrium solutions in power sys-
tems. In this paper, along with the line of stability constraints, we develop the sta-
bility solution of nonlinear equations to optimization problem, which has the form
of (1.1). Due to the nonsmooth property of stability constraints, we then study (1.1)
on correlative contents, such as optimal conditions, nonsmooth property and smooth-
ing approach. Furthermore, we consider its solution method by using the smoothing
technology in [22] and a smoothing SQP algorithm is presented for solving (1.1).
The global convergence of the algorithm is investigated. An example arising from
the OPF with stability constraints is applied to test the model and the algorithm. The
numerical results show the efficiency of this study. The main contribution of this pa-
paper is twofold. First, it presents a uniform mathematical model to describe a class of
practical optimization problems. The model enjoys nice semismooth characteristic.
Second, related issues such as optimal conditions, computational method, and con-
vergence of the method, are investigated, which are necessary and usable from the
viewpoint of practical implementations.

The paper is organized as follow: In Sect. 2, we present a stability constraint model
associated with some typical stability problems in dynamic systems. Section 3 dis-
cusses the optimal conditions of the NPSC problem, as well as its smoothing one.
In Sect. 4, a smoothing SQP method is addressed. Section 5 investigates the conver-
gegence of the algorithm. Numerical example is done in Sect. 6. Section 7 gives some
final conclusions.

2 Typical stability constraints in dynamic systems

This section presents some typical models of stability constraints for dynamic sys-
tems. Here we just describe the models. The detailed contents are introduced in our
previous work [11].

2.1 Stability constraints

A nonlinear dynamic system can be described by

\[ \dot{x} = F(x, u), \]

where \( F(x, u) = (f_1(x, u), \ldots, f_n(x, u))^T : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \) is a twice continuously
derifferentiable function, and \( x \) and \( u \) are called state and control variables, respectively.
Such a system arises from various engineering problems. In this paper, we are not
interested in the dynamic operation of (2.1), but concerned with steady-state stability analysis. In other words, the system (2.1) provides just a background of stability analysis.

We call $x$ an equilibrium point of the system (2.1) if it satisfies

$$F(x, u) = 0$$

with given control variable $u$. In addition, combining with the stability requirement of dynamic systems, we define a so-called stability equilibrium operation of (2.1) in our research [11] as

$$\begin{align*}
F(x, u) &= 0, \\
\phi(x, u) &\geq \delta > 0,
\end{align*}$$

where $\delta > 0$ is a given constant vector of $\mathbb{R}^m_1$. Here $\phi(x, u) \in \mathbb{R}^m_1$ is called a stability function. Such function has a property that if there exists $j$ such that $\phi_j(x) = 0$, then the correspondent state of system is unstable.

Let $u$ be a variable. We then set up an optimal operation with some optimal objective. The optimal operation is called optimal power flow (OPF) in power systems. Furthermore, if we combine it with the stability requirement, this will formulate the optimal problem of stability constraints (1.1).

Problem (1.1) is an extension of classical OPF in power systems. The key issue is the way to set the stability constraints. Generally, there are three typical stability problems associated with dynamic system (2.1), which will result in voltage collapse and system oscillation in power systems (see [13]).

**Definition 2.1** Consider the dynamic system (2.1) with given $u$. Let $\lambda(A)$ be the eigenvalue of matrix $A$, and $\nabla F(x, u)$ is the derivative of $F$ with respect to the variable $x$.

(i) If $x$ satisfies

$$F(x, u) = 0, \quad \text{Re}\left(\lambda(\nabla F(x, u))\right) < 0,$$

then we call $x$ a Static Stable Equilibrium Point of (2.1).

(ii) If $x \in C_S$, where

$$C_S = \{x \in \mathbb{R}^n \mid F(x, u) = 0, \; \det(\nabla F(x, u)) = 0\},$$

then we call $x$ a Saddle-Node Bifurcation Point of (2.1).

(iii) If $x \in C_H$ with

$$C_H = \{x \in \mathbb{R}^n \mid F(x, u) = 0, \; \text{Re}\left(\lambda(\nabla F(x, u))\right) = 0\},$$

then we call $x$ a Hopf Bifurcation Point of (2.1), i.e., $\nabla F(x, u)$ has a pair of conjugate eigenvalues passing the imaginary axis.

The stable equilibrium models of power systems are required to satisfy (2.3), or to avoid the point of (2.4) and (2.5). On the other hand, from the viewpoint of numerical calculations, the proposed models should be solved by some typical numerical
methods, such as Newton-type methods. To this end, we construct stability functions satisfying some semismooth property. For example, the Saddle-Node Bifurcation can be avoided by setting \( \det(\nabla F(x)) \neq 0 \). Then we set the stability function as

\[
\phi(x, u) \equiv \min_{1 \leq i \leq n} \lambda_i(\nabla F(x, u)^T \nabla F(x, u)) > 0.
\]

Based on the stable and semismooth requirements, by using mathematical analysis, three semismooth stability constraints with respect to Definition 2.1 and \( m_1 = 1 \) can be set up (see [11] for the detailed analysis). Here \( m_1 = 1 \) means that for each stable model, we just consider one stability requirement.

In the following analysis of this section, the control variable \( u \) is assumed to be constant.

- **(i) Static Stability Constraint:**
  \[
  \phi^I(x, u) = -\max_{1 \leq i \leq n} \lambda_i(\nabla F(x, u) + \nabla F(x, u)^T) \geq \delta,
  \]
  \[
  \phi^{II}(x, u) = 1 - \|C(\nabla F(x, u))\|_1 \geq \delta.
  \]

- **(ii) Avoiding Saddle-node Bifurcation:**
  \[
  \phi_2(x, u) = \min_{1 \leq i \leq n} \lambda_i(\nabla F(x, u)^T \nabla F(x, u))
  = -\max_{1 \leq i \leq n} \lambda_i(-\nabla F(x, u)^T \nabla F(x, u)) \geq \delta.
  \]

- **(iii) Avoiding Hope Bifurcation:**
  \[
  \phi_3(x, u) = 1 + \max\{-\|C(\nabla F(x, u))\|_1, -\|C^{-1}(\nabla F(x, u))\|_1\} \geq \delta.
  \]

Here \( C(A) = (A - \sigma I)^{-1}(A + \sigma I) \) is the Cayley transformation of matrix \( A \).

We use \( \phi(x, u) \) to express the functions \( \phi_i(x, u) \) \((i = 1, 2, 3)\). In addition, we introduce a slack variable \( y \in \mathbb{R} \) to transfer the inequality constraint to equality one. Then the stability inequality constraint \( \phi(x, u) \geq \delta \) can be transformed equivalently to an equality constraint:

\[
\Phi(x, u, y) = -\phi(x, u) + |y| + \delta = 0,
\]

where \( |y| \) expresses the absolute value of \( y \), and is semismooth [5].

### 2.2 Smoothing function of maximal eigenvalue function

We note that the functions defined in (2.9) are related to the so-called maximum function as follow:

\[
\hat{\phi}(x) \equiv \max\{x_1, \ldots, x_n\}.
\]

Since \( \hat{\phi}(x) \) is semismooth, we will consider its smoothing version as

\[
\hat{\phi}_s(\epsilon, x) = \begin{cases} 
\epsilon \ln(\sum_{i=1}^n e^{x_i/\epsilon}), & \text{if } \epsilon \neq 0, \\
\max_{1 \leq i \leq n}\{x_i\}, & \text{if } \epsilon = 0.
\end{cases}
\]
Here $\epsilon > 0$ is called a smoothing parameter. Such smoothing function is also called an exponential penalty function and has some interesting characteristics (see Lemma 2.1 in [4] and Lemma 7 in [16]).

**Lemma 2.1** $\hat{\phi}_s(\epsilon, x)$ has the following properties: for any $\epsilon > 0$.

(i) $\hat{\phi}_s(\epsilon, x)$ is increasing with respect to $\epsilon$, i.e., for any $\epsilon_1 > \epsilon_2 > 0$, it holds $\hat{\phi}_s(\epsilon_1, x) \geq \hat{\phi}_s(\epsilon_2, x)$. Further, we have

$$0 \leq \hat{\phi}_s(\epsilon, x) - \hat{\phi}(x) \leq \epsilon \ln n. \tag{2.10}$$

(ii) $\hat{\phi}_s(\epsilon, x)$ is continuously differentiable and

$$\nabla \hat{\phi}_s(\epsilon, x) = \sum_{i=1}^n a_i(\epsilon, x)e_i, \tag{2.11}$$

$$\left\{ \begin{align*}
\frac{d\hat{\phi}_s(\epsilon, x)}{d\epsilon} &= \ln \sum_{i=1}^n e^{x_i/\epsilon} - \frac{1}{\epsilon} \sum_{i=1}^m a_i(\epsilon, x)x_i, \\
\lim_{\epsilon \to 0^+} \frac{d\hat{\phi}_s(\epsilon, x)}{d\epsilon} &= \ln |I(x)|, \quad \ln |I(x)| \leq \frac{df(\epsilon, x)}{d\epsilon} \leq \ln n
\end{align*} \right. \tag{2.12}$$

with fixed $x$, where

$$a_i(\epsilon, x) = \frac{e^{x_i/\epsilon}}{\sum_{i=1}^m e^{x_i/\epsilon}}, \quad I(x) = \{ i \mid \hat{\phi}(x) = x_i \}, \tag{2.13}$$

and $e_i$ is the unit vector with its $i$-th component to be 1.

(iii) For any fixed $x \in \mathbb{R}^n$,

$$\text{dist}(\nabla_x \hat{\phi}_s(\epsilon, x), \partial \hat{\phi}(x)) = o(\epsilon). \tag{2.14}$$

On the other hand, the formulas (2.6)–(2.8) indicate that stability functions are related to the maximal eigenvalue of matrix. We then introduce the so-called spectral function (denoted by $(f \circ \lambda)$), which is a composition of symmetric function $f : \mathbb{R}^n \to \mathbb{R}$ and eigenvalue function $\lambda(\cdot)$ of symmetric matrices, see [4] for detailed discussion. Here the symmetric function $f$ means $f(x) = f(Px)$ with any permutation matrix $P$. The spectral function has the following properties [17].

**Lemma 2.2** A spectral function $(f \circ \lambda)$ satisfies the following properties:

(i) it is semismooth if and only if $f$ is semismooth. If $f$ is $\rho$-order semismooth ($0 < \rho \leq 1$), the spectral function $(f \circ \lambda)$ is $\min\{1, \rho\}$-order semismooth;

(ii) it is $\mathcal{LC}^1$ if and only if $f$ is $\mathcal{LC}^1$;

(iii) it is $\mathcal{SC}^1$ if and only if $f$ is $\mathcal{SC}^1$.

(iv) $(f \circ \lambda)$ is differentiable at a matrix $X \in \mathcal{S}_n$ if and only if $f$ is differentiable at point $\lambda(X)$. The gradient of $(f \circ \lambda)$ at $X$ is given by

$$\nabla(f \circ \lambda)(X) = U^T(\text{diag}(\nabla f(\lambda(X))))U, \quad \forall U \in O(X). \tag{2.15}$$
More generally, the gradient of \((f \circ \lambda)\) has the following formula
\[
\nabla (f \circ \lambda)(X) = V^T (\text{diag} (\nabla f(\mu))) V,
\]
(2.16)
for any orthogonal matrix \(V \in O_n\) and \(\mu \in \mathbb{R}^n\) satisfying \(X = V^T (\text{diag} \mu) V\). Here \(S_n\) and \(O_n\) denote the real \(n \times n\) symmetric matrix set and orthogonal matrix set respectively, and \(O(X) = \{P \in O_n \mid PXPT = \text{diag}(\lambda(X))\}\).

As an example, we choose the following special spectral function:
\[
(\hat{\phi} \circ \lambda)(X) = \max_{1 \leq i \leq n} \lambda_i(X), \quad R^{n \times n} \rightarrow \mathbb{R}.
\]
(2.17)
Here \(X\) belongs to the set of symmetric matrices. Then from Lemma 2.2 we have that \((\hat{\phi} \circ \lambda)(X)\) is semismooth, so does the correspondent functions of stability constraints proposed in (2.6)–(2.8).

In order to design solution algorithm for NPSC problem, we use the smoothing way of maximum function and denote the smoothing function of \((\hat{\phi} \circ \lambda)(X)\) as \((\hat{\phi}_s \circ \lambda)(\epsilon, X)\), which has the following version for \(\epsilon > 0\)
\[
(\hat{\phi}_s \circ \lambda)(\epsilon, X) = \epsilon \ln \left( \sum_{i=1}^{n} e^{\lambda_i(X)/\epsilon} \right).
\]
(2.18)
Combining with Lemma 2.1, we have the following conclusions with respect to the spectral function and its smoothing one (see Proposition 2.2 in [4]).

**Lemma 2.3** Let \(X\) be symmetric matrix. For any \(\epsilon > 0\), it holds
\[
0 \leq (\hat{\phi}_s \circ \lambda)(\epsilon, X) - (\hat{\phi} \circ \lambda)(X) \leq \epsilon \ln n,
\]
(2.19)
\[
\lim_{\epsilon \to 0} \nabla (\hat{\phi}_s \circ \lambda)(\epsilon, X) \in \partial (\hat{\phi} \circ \lambda)(X).
\]
(2.20)

### 2.3 Smoothing technology for stability constraints

In the reminder part of this section, we focus on the smoothing version of three stability functions \(\phi^I_1(x), \phi^{II}_4(x), \phi_2(x), \phi_3(x)\).

Denote the smoothing functions of stability functions in a uniform form as \(\phi_s(\epsilon, x)\). Then we obtain a smoothing system of (2.9) as
\[
\Phi_s(\epsilon, x, u, y) = -\phi_s(\epsilon, x, u) + \chi(\epsilon, y) + \delta = 0,
\]
(2.21)
where \(\chi(\epsilon, y)\) is the smoothing function of \(|y|\) via the smoothing version of maximum function:
\[
\chi(\epsilon, y) = \epsilon \ln(e^{y/\epsilon} + e^{-y/\epsilon}).
\]
From above Lemmas 2.1–2.3, we can obtain the uniform approximate property of \(\Phi_s(\epsilon, x, u, y)\) and \(\Phi(x, u, y)\) as follows.
Proposition 2.1 Let $\Phi_s(\epsilon, x, u, y)$ be the smoothing function of $\Phi(x, u, y)$ with respect to three stability functions. Then the following conclusions hold for the smoothing parameter $\epsilon > 0$.

(i) Approximation property:

$$|\Phi_s(\epsilon, x, u, y) - \Phi(x, u, y)| = O(\epsilon).$$  \hspace{0.5cm} (2.22)

(ii) The derivative of $\Phi_s(\epsilon, x, u, y)$ with respective to $(x, y)$:

$$\nabla \Phi_s(\epsilon, x, u, y) = -\left(\nabla_x \phi_s(\epsilon, x, u) + \begin{pmatrix} 0 \\ \frac{e^{y/\epsilon} - e^{-y/\epsilon}}{e^{y/\epsilon} + e^{-y/\epsilon}} \end{pmatrix}\right).$$ \hspace{0.5cm} (2.23)

(iii) Property of Jacobian:

$$\lim_{\epsilon \to 0} \text{dist}\left(\nabla \Phi_s(\epsilon, x, u, y), \partial \Phi(x, u, y)\right) = 0.$$ \hspace{0.5cm} (2.24)

Here $\partial \Phi(x, u, y)$ expresses the generalized Jacobian of $\Phi(x, u, y)$ in the sense of Clarke derivative \[5\]. The symbol dist indicates the distance of a point to the set.

Proof As an example, we prove the proposition by the static stability constraint function $\phi^I_1(x, u)$. For such case, we have the stability function and their smoothing one as:

$$\begin{cases}
\phi^I_1(x, u) = -\max_{1 \leq i \leq n} \lambda_i (\nabla F(x, u) + \nabla F(x, u)^T), \\
\phi_s(\epsilon, x, u) = -\epsilon \ln \left(\sum_{i=1}^{n} e^{\lambda_i (\nabla F(x, u) + \nabla F(x, u)^T)/\epsilon}\right).
\end{cases}$$

(i) From Lemma 2.1 and Lemma 2.3 we have

$$\begin{cases}
-\epsilon \ln n \leq \phi_s(\epsilon, x, u) - \phi^I_1(x, u) \leq 0, \\
0 \leq \chi(\epsilon, y) - |y| \leq \epsilon \ln 2.
\end{cases}$$

Then it follows

$$0 \leq \Phi_s(\epsilon, x, u, y) - \Phi(x, u, y) \leq \epsilon (\ln n + \ln 2),$$

which shows the result of (2.22).

(ii) This is a direct derivative calculation of function.

(iii) We obtain the result from (2.20) in Lemma 2.3 and (2.14) in Lemma 2.1 for functions $\phi^I_1(x, u)$ and $|y|$, respectively.

For other stability constraint functions, the proposition can be proved by the same method. The difference with 1-norm $\| \cdot \|_1$ in the stability constraint is where two-times smoothing process of maximum function is needed.

We complete the proof. \[\square\]
3 Optimal conditions of NPSC problems

3.1 Optimal conditions for semismooth NPSC problems

We analyze the optimal conditions of (1.1) with $\phi$ defined in (2.6)–(2.8) and $n_s = 1$ in this section. It is obvious that (1.1) is equivalent to the following problem by adding a slack variable $y \in \mathbb{R}$:

$$
\begin{align*}
\min_{(x,u,y) \in \mathbb{R}^{n+1}} & \quad f(x,u) \\
\text{s.t.} & \quad h(x,u) = 0, \\
& \quad g(x,u) \leq 0, \\
& \quad \Phi(x,u,y) \equiv -\phi(x,u) + |y| + \delta = 0.
\end{align*}
$$

(3.1)

Here the typical stability problems are discussed in Sect. 2, i.e., the stability functions $\phi(x,u)$ are defined in (2.6)–(2.8) as $\phi^I_1(x,u)$, $\phi^{II}_1(x,u)$, $\phi_2(x,u)$, $\phi_3(x,u)$.

According to the optimal condition of nonsmooth optimization [5], we introduce the concept of generalized stationary points of (3.1).

**Definition 3.1** A point $(x^*, u^*) \in \mathbb{R}^{n+1}$ is said to be a generalized stationary point of (3.1) if there exists a KKT multiplier vector $(\lambda_h, \lambda_g, \lambda_\phi) \in \mathbb{R}^{n+l+1}$ such that the following generalized Karush-Kuhn-Tucker (GKKT) conditions hold:

$$
\begin{align*}
0 \in & \left( \nabla f(x^*, u^*) \right) + \left( h'(x^*, u^*)^T \right) \lambda_h + \left( g'(x^*, u^*)^T \right) \lambda_g + \left( \partial \phi(x^*, u^*)^T \alpha \right) \lambda_\phi, \\
h(x^*, u^*) = & \ 0, \\
0 \leq & -g(x^*, u^*) \perp \lambda_g \geq 0, \\
\Phi(x^*, u^*, y^*) \equiv & -\phi(x^*, u^*) + |y^*| + \delta = 0,
\end{align*}
$$

(3.2)

where $\alpha \in [-1, 1]$.

For convenience, we may assume that in the inequality constraints of (3.1), the first $l_1$ ($l_1 \leq l$) inequality constraints are active and the rest are inactive at $(x^*, u^*)$. Denote

$$
G(x,u) \equiv (h_1(x,u), \ldots, h_n(x,u), g_1(x,u), \ldots, g_{l_1}(x,u))^T, \\
\bar{G}(x,u,y) \equiv \left( \begin{array}{c} G(x,u) \\ \Phi(x,u,y) \end{array} \right).
$$

If each element of the generalized Jacobian of $\bar{G}(x,u,y)$ has full row rank, then we say that the **Generalized Linear Independent Constraint Qualification** (GLICQ) condition of (3.1) is satisfied. For the addressed problem, the following conclusion gives a sufficient condition of constrained qualification.
Proposition 3.1 Suppose that $G'(x,u)$ has full row rank at $(x^*,u^*)$ and $y^* \neq 0$. Then the GLICQ holds for NPSC problem (3.1).

Proof The active constraints of (3.1) can be written as

\[ \tilde{G}(x,u,y) = \begin{pmatrix} G(x,u) \\ -\phi(x,u) + |y| + \delta \end{pmatrix}. \]

It is obvious that for each element of generalized Jacobian $V \in \partial \tilde{G}(x,u,y)$ with $V_\phi \in \partial \phi(x,u)$ it has the following form

\[ V = \begin{pmatrix} G'(x,u) \\ -V_\phi \alpha \end{pmatrix}. \] (3.3)

On the other hand, the generalized derivative of $|y|$ is

\[ \partial |y| = \begin{cases} 1, & \text{if } y > 0, \\ [-1,1], & \text{if } y = 0, \\ -1, & \text{if } y < 0. \end{cases} \]

By the condition of proposition $y^* \neq 0$, it holds

\[ \alpha = \begin{cases} 1, & \text{if } y^* > 0, \\ -1, & \text{if } y^* < 0, \end{cases} \]

which combines with the expression of $V$ to obtain that $V$ is full row rank. We complete the proof. \qed

Above proposition relies on the assumption $y^* \neq 0$. We note that on the one hand, the GLICQ condition is a sufficient condition of constraint qualification of nonlinear programming. This means that for case $y^* = 0$, the constrained qualification condition of NPSC may hold. On the other hand, for given $\delta > 0$, if it happens $y^* = 0$ at an optimal solution (i.e., the stability constraint is active), we can choose $\delta_1 < \delta$ in (3.1). Then we may obtain an optimal solution with $y^* \neq 0$, which satisfies the assumption of the proposition. The constrained qualification condition of NPSC is a valuable problem.

3.2 Optimal conditions for smoothing problems

According to the smoothing functions of the stability functions, we have nonlinear program with the smoothed stability equilibrium operation in the constraint set of (3.1) as:

\[
\begin{align*}
\min_{(x,u,y) \in \mathbb{R}^{n+m+1}} & \quad f(x,u) \\
\text{s.t.} & \quad h(x,u) = 0, \\
& \quad g(x,u) \leq 0, \\
& \quad \Phi_s(\epsilon,x,u,y) \equiv -\phi_s(\epsilon,x,u) + \chi(\epsilon,y) + \delta = 0.
\end{align*}
\] (3.4)
Obviously, if \((x, u, y)\) is a feasible point (local solution, global solution) of (3.1), it is a feasible point (local solution, global solution) of (3.4) with \(\epsilon = 0\).

We now write the KKT condition of the smoothing nonlinear program (3.4). For \(\epsilon \neq 0\), it holds

\[
0 = \begin{pmatrix}
\nabla f(x, u) \\
0
\end{pmatrix} + \begin{pmatrix}
 h'(x, u)^T \\
0
\end{pmatrix} \lambda_h + \begin{pmatrix}
 g'(x, u)^T \\
0
\end{pmatrix} \lambda_g + \begin{pmatrix}
 \phi_s'(\epsilon, x, u)^T \\
\chi'(\epsilon, y)
\end{pmatrix} \lambda_{\Phi_1},
\]

\[
h(x, u) = 0,
\]

\[
0 \leq -g(x, u) \perp \lambda_g \geq 0,
\]

\[
\Phi_s(\epsilon, x, u, y) = 0.
\]

(3.5)

Here Jacobian of \(\phi_s'(\epsilon, x, u)\) with respect to \((x, u)\) can be obtained from the smoothing function of \(\phi_s\). The derivative of \(\chi(\epsilon, y)\) with respect to \(y\) is

\[
\chi'(\epsilon, y) = \frac{e^{y/\epsilon} - e^{-y/\epsilon}}{e^{y/\epsilon} + e^{-y/\epsilon}}.
\]

(3.6)

4 A smoothing SQP algorithm

We now present a smoothing SQP method for solving problem (3.4). For more discussion on smoothing SQP methods, see [9, 15].

Let \(d = (dx, du, dy)\). For given \((x, u, y) \in \mathbb{R}^{n+m+1}\) and \(r > 0, \epsilon > 0\). Similar to reference [8], we define a modified quadratic program \(QP(x, u, y, \epsilon, r)\) subproblem of (3.4) by

\[
\text{min}_{d \in \mathbb{R}^{n+m+1}, \xi \in \mathbb{R}^l} \nabla f(x, u)^T \begin{pmatrix} dx \\ du \end{pmatrix} + \frac{1}{2} d^T W d + r \sum_{i=1}^l \xi_i \\
\text{s.t. } h'(x, u) \begin{pmatrix} dx \\ du \end{pmatrix} + h(x, u) = 0, \\
g'(x, u) \begin{pmatrix} dx \\ du \end{pmatrix} + g(x, u) \leq \xi, \\
\Phi_s'(\epsilon, x, u, y) d + \Phi_s(\epsilon, x, u, y) = 0, \\
\xi \geq 0,
\]

(4.1)

where \(W \in \mathbb{R}^{(n+m+1) \times (n+m+1)}\) is a symmetric positive definite matrix, and \(r > 0\) is a penalty parameter. Here we add one variable \(\xi\) to ensure the feasibility of the inequality constraint. In addition, we can see that the feasibility of equality constraints in (4.1) is satisfied if the matrix

\[
V_1 = \begin{pmatrix}
 h'(x, u) & 0 \\
-\phi_s'(x, u) & \chi'(\epsilon, y)
\end{pmatrix}
\]

has full row rank.
It is not difficult to write the KKT system of (4.1) as

\[
\begin{align*}
\left( \nabla f(x, u) \right) + Wd + \left( h'(x, u)^T \right) \lambda_h + \left( g'(x, u)^T \right) \lambda_g + \left( \phi'(\epsilon, x, u)^T \right) \lambda_{\Phi_1} &= 0, \\
r &= \lambda_g + \lambda_{\xi}, \\
h'(x, u) \left( \frac{dx}{du} \right) + h(x, u) &= 0, \\
0 \leq -\left[ g'(x, u) \left( \frac{dx}{du} \right) + g(x, u) - \xi \right] \perp \lambda_g \geq 0, \\
-\phi'(\epsilon, x, u) \left( \frac{dx}{du} \right) + \chi'(\epsilon, y)dy + \Phi_1(\epsilon, x, u, y) &= 0, \\
0 \leq \xi \perp \lambda_{\xi} \geq 0.
\end{align*}
\]

(4.2)

Define a merit function by a penalty function as:

\[
\Theta_{r_g, r_h, r_{\Phi}, \epsilon}(x, u, y) = f(x, u) + r_g \sum_{i=1}^{l} \max \{ g_i(x, u), 0 \} + r_h \sum_{i=1}^{n} |h_i(x, u)| + r_{\Phi} |\Phi_1(\epsilon, x, u, y)|,
\]

(4.3)

where \( r_g, r_h \) and \( r_{\Phi} \) are positive numbers.

Now we state the SQP method for solving (1.1) by a smoothing method. We handle \( \epsilon \) as the smoothing parameter and adjust it in every iteration.

**Algorithm 4.1** Smoothing SQP Algorithm.

**Step 0.** (Initialization)
Let \( r_{k-1} > 0, \delta_1 > 0, \delta_2 > 0, \beta_\epsilon \in (0, 1), \sigma \in (0, 1), \tau \in (0, 1) \); choose \( w^0 = (x^0, u^0, y^0) \in R^{n+m+1} \), and \( \epsilon_0 > 0 \); and a symmetric positive definite matrix \( W_0 \in R^{(n+m+1) \times (n+m+1)} \). Set \( k := 0 \).

**Step 1.** (Search direction)
Solve the QP subproblem (4.1) with \( w^k = (x^k, u^k, y^k), \epsilon = \epsilon_k, W = W_k, r = r_{k-1} \). Let \( (d^k, \xi^k) \) be a solution of the QP, and \( \lambda^k = (\lambda_h, \lambda_g, \lambda_{\Phi}, \lambda_{\xi}) \) be its corresponding KKT multiplier.

**Step 2.** (Termination check)
If a stopping rule is satisfied, terminate. Otherwise, go to Step 3.

**Step 3.** (Penalty update)
Let

\[
\tilde{r}_k = \begin{cases} 
    r_{k-1}, & \text{if } r_{k-1} \geq \max_{1 \leq i \leq m+n+1} \{ |\lambda^k_i| \}, \\
    \delta_1 + \max_{1 \leq i \leq n+l+2} \{ |\lambda^k_i| \}, & \text{otherwise}.
\end{cases}
\]

(4.4)
Define $r^g_k = r_{k-1}$, and $r^h_k = r^\Phi_k = \tilde{r}_k$ and

$$r_k = \begin{cases} \tilde{r}_k, & \text{if } \sum_{i=1}^l \xi^k_i = 0, \\ \tilde{r}_k + \delta_2, & \text{otherwise}. \end{cases} \tag{4.5}$$

Step 4. (Line Search)

Let $t_k = (\tau)^i_k$, where $i_k$ is the smallest nonnegative integer such that $i = i_k$ satisfies

$$\Theta(r^g_k, r^h_k, r^\Phi_k, \epsilon_k)(w^k + (\tau)^i_k d_k) - \Theta(r^g_k, r^h_k, r^\Phi_k, \epsilon_k)(w^k) \leq -\sigma(\tau)^i_k (d^k)^T W_k d_k. \tag{4.6}$$

Step 5. (Update Smoothing Parameter)

Let

$$w^{k+1} = w^k + t_k d^k, \tag{4.7}$$

$$\epsilon_{k+1} = \begin{cases} \beta \epsilon \epsilon_k, & \text{if } \|d^k\| \leq \epsilon_k, \\ \epsilon_k, & \text{otherwise}. \end{cases} \tag{4.8}$$

Choose a symmetric positive definite matrix $W_{k+1} \in R^{(m+n+1) \times (m+n+1)}$. Set $k := k + 1$ and go to Step 1.

Remark 4.1 At Step 2 in Algorithm 4.1, we do not specify a stopping rule. Generally, we stop the iteration when $d^k$ and $\epsilon_k$ are sufficiently small. For example, the stopping rule is chosen

$$\|d^k\| + \epsilon_k \leq \hat{\tau},$$

where $\hat{\tau}$ is a given small constant.

The search direction obtained from the subproblem (4.1) in Step 1 of Algorithm 4.1 has the following descent property.

**Proposition 4.1** For any $\epsilon > 0$, we have

(i) $\Theta_{(r^g, r^h, r^\Phi, \epsilon)}(x, u, y)$ is directionally differentiable at $w = (x, u, y)$. Furthermore, if $(d, \xi)$ is a solution of QP subproblem (4.1), $r^g = r$ and $r^h = r^\Phi \geq \max_{1 \leq i \leq n+1+2} |\lambda_i|$ with $\lambda = (\lambda_h, \lambda_g, \lambda_\Phi, \lambda_\xi)$ to be the corresponding multiplier of (4.1), then

$$\Theta'_{(r^g, r^h, r^\Phi, \epsilon)}(x, u, y; d) \leq \nabla f(x, u)^T \tilde{d} + \lambda_g^T g'(x, u) \tilde{d} + \lambda_h^T h'(x, u) \tilde{d} + \lambda_\Phi^T \Phi'(x, u, y) d,$$  \tag{4.9}

where $\tilde{d} = (\frac{dx}{du})$. Moreover, we have

$$\Theta'_{(r^g, r^h, r^\Phi, \epsilon)}(x, u, y; d) \leq -d^T W d. \tag{4.10}$$
Suppose that $W \in \mathbb{R}^{(n+m+1) \times (n+m+1)}$ is symmetric positive definite. If $(d, \xi)$ is a solution of (4.1) with $d \neq 0$, then $d$ is a descent direction of the penalty function $\Theta_{(r^g, r^h, r^\Phi, \epsilon)}(w)$ for $r^g = r$, and $r^h = r^\Phi$ satisfying the condition in (i).

Proof (i) The directionally differentiable conclusion is obtained from the property of the maximal function and the absolute function. In addition, we calculate the directional derivative of $\Theta$ as

$$
\Theta_{(r^g, r^h, r^\Phi, \epsilon)}(x, u, y; d)
= \nabla f(x, u)^T \bar{d} + r^g \left[ \sum_{i: g_i < 0} 0 + \sum_{i: g_i = 0} \max\{g_i'(x, u)\bar{d}, 0\} + \sum_{i: g_i > 0} g_i'(x, u)\bar{d} \right] + r^h \left[ \sum_{j: h_j > 0} h_j'(x, u)\bar{d} + \sum_{j: h_j = 0} |h_j'(x, u)\bar{d}| + \sum_{j: h_j < 0} -h_j'(x, u)\bar{d} \right] + r^\Phi \kappa_\Phi,
$$

(4.11)

where

$$
\kappa_\Phi = \begin{cases} 
\Phi'_s(\epsilon, w)d, & \text{if } \Phi_s(\epsilon, w) > 0, \\
|\Phi'_s(\epsilon, w)d|, & \text{if } \Phi_s(\epsilon, w) = 0, \\
-\Phi'_s(\epsilon, w)d, & \text{if } \Phi_s(\epsilon, w) < 0.
\end{cases}
$$

In order to prove the inequality (4.9), the KKT system (4.2) of the subproblem is used repeatedly.

First, we consider the equality parts $h(x, u)$ and $\Phi_s(\epsilon, w)$. By direct calculation and the KKT condition (4.2), it holds from (4.2) that

$$
r^h \left[ \sum_{j: h_j > 0} h_j'(x, u)\bar{d} + \sum_{j: h_j = 0} |h_j'(x, u)\bar{d}| + \sum_{j: h_j < 0} -h_j'(x, u)\bar{d} \right] = r^h \left[ \sum_{j: h_j > 0} (-h_j(x, u)) + \sum_{j: h_j = 0} |h_j(x, u)| + \sum_{j: h_j < 0} h_j(x, u) \right] = -r^h \|h(x, u)\|_1.
$$

(4.12)

On the other hand, from the KKT system (4.2) again and the condition of $r^h$ we have

$$(\lambda_h)^T h'(x, u)\bar{d} = - (\lambda_h)^T h(x, u) \geq - \sum_{1 \leq j \leq n} |(\lambda_h)_j||h_j(x, u)| \geq -r^h \|h(x, u)\|_1,$$

which combines with (4.12) to yield

$$
r^h \left[ \sum_{j: h_j > 0} h_j'(x, u)\bar{d} + \sum_{j: h_j = 0} |h_j'(x, u)\bar{d}| + \sum_{j: h_j < 0} -h_j'(x, u)\bar{d} \right] \leq \lambda_h^T h'(x, u)\bar{d}.
$$

(4.13)
By using a similar way we can prove that
\[ r^\Phi k_\Phi \leq \lambda_\Phi \Phi'_\Sigma(\omega, w)d. \] (4.14)

Second, we consider the inequality part. From (4.11), to prove the conclusion (4.9) suffices to prove the follows:
\[ r^g(\max\{g_i(x, u), 0\})'(x, u; \bar{d}) \leq (\lambda_g)_i g'_i(x, u)\bar{d} \quad (i = 1, 2, \ldots, l). \] (4.15)

We consider two cases for proving (4.15). Note that from the second relationship in the KKT condition (4.2) it implies
\[ r^g \geq (\lambda_g)_i \quad (i = 1, \ldots, l). \] (4.16)

**Case I:** \( \xi_i = 0. \)

If \( g_i(x, u) > 0, \) then from the KKT condition (4.2) we derive that
\[ (\max\{g_i(x, u), 0\})'(x, u; \bar{d}) = g'_i(x, u)\bar{d} \leq - g_i(x, u) + \xi_i < 0. \]

This implies (4.15) from (4.16).

If \( g_i(x, u) = 0, \) then it holds
\[ (\max\{g_i(x, u), 0\})'(x, u; \bar{d}) = \max\{g'_i(x, u)\bar{d}, 0\} = 0. \]

From the complementarity condition in (4.2) we have
\[ (\lambda_g)_i g'_i(x, u)\bar{d} = (\lambda_g)_i (g_i(x, u) + \xi_i) = 0. \]

So (4.15) holds.

If \( g_i(x, u) < 0, \) on the one hand, the complementarity condition shows that
\[ (\lambda_g)_i g'_i(x, u)\bar{d} = -(\lambda_g)_i (g_i(x, u) - \xi_i) \geq 0. \]

On the other hand, it has
\[ r^g(\max\{g_i(x, u), 0\})'(x, u; \bar{d}) = 0. \]

This also implies (4.15).

**Case II:** \( \xi_i > 0. \)

For this case, we have \( r^g = (\lambda_g)_i \) since \( \lambda_\xi = 0 \) and the second expression in (4.2).
If \( g_i(x, u) > 0, \) the conclusion (4.15) is straightforward.
If \( g_i(x, u) = 0, \) the complementarity condition in (4.2) implies the follows
\[ r^g(\max\{g_i(x, u), 0\})'(x, u; \bar{d}) = r^g \max\{g'_i(x, u)\bar{d}, 0\} \]
\[ = \max\{(\lambda_g)_i g'_i(x, u)\bar{d}, 0\} = \max\{(\lambda_g)_i \xi_i, 0\} \]
\[ = (\lambda_g)_i \xi_i = (\lambda_g)_i g'_i(x, u)\bar{d}. \]

This also proves (4.15).
If \( g_l(x, u) < 0 \), the complementarity condition implies that
\[
 r^g(\max\{g_l(x, u), 0\})'(x, u; \bar{d}) = \frac{d}{\lambda^g}(\lambda^g_l g'(x, u) + \xi) = \frac{d}{\lambda^g} g'(x, u) \bar{d}.
\]
This proves (4.15).

Hence, for all cases, the result (4.15) holds. Combining with (4.11), (4.13)–(4.14) and (4.15) the inequality (4.9) is proved.

The conclusion (4.10) follows from (4.9) and the first equality of (4.2).

(ii) The conclusion is obtained from the conclusion (i), the positive definite matrix \( W \) and \( d \neq 0 \).

We complete the proof of the proposition. \( \square \)

Remark 4.2 (i) Proposition 4.1 shows that line search in Step 4 of Algorithm 4.1 is well-defined for \( d \neq 0 \), so does the smoothing SQP method when \( \epsilon_k > 0 \) and \( W_k \) to be a symmetric positive definite matrix at each iteration. On the other hand, from Remark 4.1 the algorithm will stop when \( ||d|| \) is sufficiently small.

(ii) Assume that \( r_k = r_* \) for large \( k \) with constant \( r_* \). Then Algorithm 4.1 implies \( \sum_{i=1}^{l} \xi_i = 0 \). From the KKT system (4.2) of subproblem (4.1), we can derive that \( d = 0 \) to be a KKT point of (3.4).

5 Global convergence

We make the following standard assumptions in order to prove the global convergence of Algorithm 4.1.

(A1) There exist two positive scalars \( m \) and \( M \) with \( m \leq M \) such that the symmetric matrix \( W_k \) in Algorithm 4.1 satisfies the following condition
\[
 m||w||^2 \leq w^T W_k w \leq M||w||^2 \tag{5.1}
\]
with any vector \( w = (x, u, y) \in \mathbb{R}^{n+m+1} \).

(A2) For all large \( k \), it holds \( r_k = r_* > 0 \), \( r^g_k = r_k \), \( r^h_k = r^\Phi_k = r^h_k \).

We have the following global convergence theorem.

Theorem 5.1 Assume that (A1) and (A2) hold. Let \( \{w^k\} \) and \( \{\epsilon_k\} \) be the sequences generated by the smoothing SQP algorithm.

(i) Assume that \( \{w_k\} \) has a limiting point. Then
\[
 \lim_{k \to \infty} \epsilon_k = 0. \tag{5.2}
\]

(ii) Let
\[
 K = \{k \mid ||d^k|| \leq \epsilon_k \}. \tag{5.3}
\]
If \( \{w^k\}_{k \in K} \) has an accumulation point \( w^* = (x^*, u^*, y^*) \), then \( w^* \) is a generalized stationary point of (3.1).
Proof (i) It is obvious that \( \{\epsilon_k\} \) is bounded. Suppose that \( \epsilon_* \) is an accumulation point of \( \{\epsilon_k\} \). If \( \epsilon_* > 0 \), then \( \|d^k\| \leq \epsilon_k \) occurs only finitely many times. This means that after a finite number of iterations, \( \epsilon_k \) remains unchanged, i.e., for some \( k_0 \) and all \( k \geq k_0 \), \( \epsilon_k = \epsilon_{k_0} \). In this case, our smoothing method reduces to the smooth modified SQP method proposed in [9] for a smooth nonlinear optimization, see Appendix in [9]. Then Theorem A.1 in [9] indicates that \( d_k \rightarrow 0 \), which implies that \( \|d^k\| \leq \epsilon_{k_0} \) for sufficiently large \( k \). This contradicts with the above conclusion. Therefore, \( \lim_{k \rightarrow \infty} \epsilon_k = 0 \).

(ii) By the assumption (A2), it implies that the KKT multiplier sequence \( \{\lambda_k\} \) is bounded and \( \xi_k = 0 \) for all large enough \( k \). We also obtain that \( \lim_{k \rightarrow \infty, k \in K} d_k = 0 \).

Without loss of generality, we assume that \( \lim_{k \rightarrow \infty, k \in K} w_k = w^* \).

It follows from the penalty update rule in Step 3 of the algorithm and the assumption (A2) that \( \xi^* = 0 \) and \( r_k = r_* \) for all sufficiently large \( k \). Moreover, from the update rule in Step 3 of Algorithm 4.1, it implies that the multiplier \( \{\lambda_k\} = \{(\lambda_g, \lambda_h, \lambda_\Phi, \lambda_\xi)\} \) and \( \{d^k\}_{k \in K} \) to be bounded. Passing to the subsequence, we may assume that

\[ \lim_{k \rightarrow \infty, k \in K} d_k = d^*, \quad \lim_{k \rightarrow \infty, k \in K} W_k = W^*, \quad \lim_{k \rightarrow \infty, k \in K} \lambda_k = \lambda^* = (\lambda_g^*, \lambda_h^*, \lambda_\Phi^*). \]

Then from \( \epsilon^* = 0 \) in (i) and the algorithm, it holds \( d^* = 0 \), \( d^* = 0 \) and \( \xi^* = 0 \). Finally, we have, from Proposition 2.1 and (4.2), that

\[ 0 \in \left( \frac{\nabla f(x^*, u^*)}{0} \right) + \left( \begin{array}{c} \lambda^*_h \left( g'(x^*, u^*) \right)^T \\ 0 \\ 0 \end{array} \right) + \left( \begin{array}{c} \lambda^*_g \\ 0 \\ 0 \end{array} \right) + \left( \begin{array}{c} \partial \Phi(x^*, u^*) \\ \partial |y^*| \end{array} \right) \lambda^*_\Phi, \]

\[ h(x^*, u^*) = 0, \quad 0 \leq -g(x^*, u^*) \perp \lambda^*_g \geq 0, \]

\[ \Phi(x^*, u^*, y^*) = -\phi(x^*, u^*) + |y^*| + \delta = 0. \]

Above expression means that \( w^* = (x^*, u^*, y^*) \) is the generalized stationary point of (3.1). The theorem is proved.

In the remainder of this section, we consider that which conditions can satisfy the assumption (A2). Denote

\[ H(\epsilon, w) = \begin{pmatrix} h(x, u) \\ \Phi_\delta(\epsilon, x, u, y) \end{pmatrix}. \]

By using a similar proof we can obtain the following conclusion.

**Proposition 5.1** Let \( \{(\epsilon_k, w^k)\} \) be generated by Algorithm 4.1 and \( (\epsilon_*, w^*) \) be an any accumulation point of \( \{(\epsilon_k, w^k)\} \). Suppose that

(i) (A1) holds;

(ii) \( \{w^k\} \) is bounded;

\( \Box \) Springer
(iii) The generalized Jacobian $\partial_w H(\epsilon_*, w^*)$ has full row rank with $\epsilon_* \geq 0$;
(iv) For any $V \in \partial_w H(\epsilon_*, w^*)$, there exists $d = (\bar{d}_{dy})$ with $\bar{d} = (dx, du)$ such that

$$g'(x, u)\bar{d} + g(x^*, u^*) < 0, \quad Vd + H(\epsilon_*, w^*) = 0.$$ 

Then the condition (A2) holds.

**Proof** The proof process is similar to the one of Theorem A.2 in [9], just to replace the Jacobian matrix of smooth function by the generalized Jacobian $V$. Here we omit the detailed process of the proof. □

6 Numerical example

In this section, we present an example for testing the NPSC model and Algorithm 4.1 proposed in Sect. 4. The WSCC 3-generators with 9-buses is chosen from PST (Power System Toolbox), see Fig. 1 for the system configuration.

The tested system has three generators, three loads and nine transmission electricity lines. Three loads (Load A-Load C) adopt dynamic model where the active power and reactive power are set

$$-P^d_i = -P^d_{0i} + D_i \hat{\delta}_i, \quad -Q^d_i(V_i) = -Q^d_{0i} + \alpha_2 V_i^2 + \alpha_1 V_i$$

with $D_i = 2.5$ and Table 1, where $\hat{\delta}_i$ means the differential of $\delta_i$ to time $t$.

**Fig. 1** WSCC 3-generators with 9-buses system

**Table 1** Load constants

<table>
<thead>
<tr>
<th></th>
<th>Load A</th>
<th>Load B</th>
<th>Load C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P^d_{0i}$</td>
<td>1.18</td>
<td>0.84</td>
<td>1.05</td>
</tr>
<tr>
<td>$Q^d_{0i}$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
In the tested system, the objective is to minimize the total energy wastage of generating fuel:

\[
W(P) = \sum_{i=1}^{n_g} W_i(P Gi) = \sum_{i=1}^{n_g} (a_i P_{Gi}^2 + b_i P_{Gi} + c_i) = \sum_{i=1}^{n_g} (a_i P_{Gi}^2 + b_i P_{Gi} + c_i)
\]  

(6.1)

with wastage coefficients \((a_i, b_i, c_i)\) \((i = 1, \ldots, n_g)\) given by Table 2.

Let the system consist of \(n_g\) generators and \((n_l - n_g)\) load buses. Denote \(Q = \{1, \ldots, n_l - n_g\}, C = \{n_l - n_g + 1, \ldots, n_l\}, N = \{n_l + 1, \ldots, n\}\) to be the index sets of load buses, generator terminal buses, and generator-inner buses, respectively. For the tested system of the 3-generators with 9-buses, we have \(n_l = 9, n_g = 3, n - n_l = 3\).

The dynamic description of power systems is DAEs as

\[
\begin{cases}
\dot{\delta} = T_2 \omega_g - T_1 D_l^{-1} T_1^T [f(\delta, V) - P], \\
\dot{\omega} = -M_g^{-1} D_g \omega_g - M_g^{-1} T_2^T [f(\delta, V) - P], \\
g(\delta, V) = 0,
\end{cases}
\]  

(6.2)

where the \(n\)-th bus is chosen as the reference bus; \(\delta = (\delta_1, \ldots, \delta_{n_l-1})^T\) and \(V = (V_1, \ldots, V_n)^T\) are bus angle and voltage magnitude; \(\omega_g = (\omega_{n_l+1}, \ldots, \omega_n)^T\) is the angle speed of generators; \(D_l, D_g\) and \(M_g\) are matrices of load-frequency, generator damp coefficient, and generator inertial coefficient, respectively. Denote \(T_1 = \begin{pmatrix} I_{n_l} \\ 0 \end{pmatrix} \in R^{(n_l-1) \times n_l}, T_2 = \begin{pmatrix} 0 & -1_{n_l-n_g} \\ 1_{n_g-1} & -1_{n_g-1} \end{pmatrix} \in R^{(n_l-1) \times n_g}\) with \(I_k\) to be \(k \times k\) unit matrix and \(1_k = (1, \ldots, 1)^T \in R^k\); \(P_i = P_{Gi} - P_{li}\) is the injective active power of buses. The functions in (6.2) are defined by

\[
\begin{cases}
f(\delta, V) = (f_1(\delta, V), \ldots, f_{n_l-1}(\delta, V))^T, \\
f_i(\delta, V) = V_i \sum_{j=1}^{n_l} V_j (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}).
\end{cases}
\]  

(6.3)

\[
\begin{cases}
g(\delta, V) = (g_1(\delta, V), \ldots, g_{n_l-n_g}(\delta, V), g_{n_l+1}(\delta, V), \ldots, g_n(\delta, V))^T, \\
g_i(V, V) = V_i \sum_{j=1}^{n_l} V_j (G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) - Q_i^d(V_i).
\end{cases}
\]  

(6.4)
The corresponding function of dynamic system (6.2) is denoted by

\[ F(\delta, \omega_g, V, P) = \begin{pmatrix} T_2\omega_g - T_1 D_1^{-1} T_1^T [f(\delta, V) - P] \\ -M_g^{-1} D_g \omega_g - M_g^{-1} T_2^T [f(\delta, V) - P] \end{pmatrix}. \] (6.5)

Then from \( g(\delta, V) = 0 \), the derivative of \( F(\cdot) \) with respect to \((\delta, \omega_g)\) is

\[ -T_1 D_1^{-1} J \tilde{F}(\delta, V) T_2 \]

\[ -M_g^{-1} T_2^T J \tilde{F}(\delta, V) - M_g^{-1} D_g \]

\[ \in R^{(2n - n_l - 1) \times (2n - n_l - 1)}, \] (6.6)

which is the function of \((\delta, V, P_G)\), and we denote it as \( \nabla F(\delta, V, P_G) \) simply. Here \( J \tilde{F}(\delta, V) \) represents the reduce Jacobian of \( f(\cdot) \) to variable \( \delta \)

\[ J \tilde{F}(\delta, V) \equiv \frac{\partial f}{\partial \delta} - \frac{\partial f}{\partial V} \left( \frac{\partial g}{\partial V} \right)^{-1} \frac{\partial g}{\partial \delta}. \]

Consider the steady stability, i.e., all the eigenvalues of \( \nabla F \) are on the left half plane. By the Cayley transformation, i.e., \( C(A) = (A - \sigma I)^{-1}(A + \sigma I) \), the stable constraint is formulated to the follows (see [11])

\[ \phi(\delta, V) = 1 - \|C(\nabla F(\delta, V, P_G))\|_1 \geq \gamma \] (6.7)

with given constant \( \gamma > 0 \).

Finally, the NPSC model with respect to OPF problems is reformulated as

\[
\begin{align*}
\min_{(\delta, V, P_G)} & \quad W(\delta, V, P_G) \\
\text{s.t.} & \quad G(\delta, V, P_G) = \begin{pmatrix} f(\delta, V) - (P_G - P_l) \\ g(\delta, V) \end{pmatrix} = 0, \\
& \quad H_{\min} \leq H(\delta, V, P_G) \leq H_{\max}, \\
& \quad \phi(\delta, V) = 1 - \|C(\nabla F(\delta, V, P_G))\|_1 \geq \gamma.
\end{align*}
\] (6.8)

Here \( G(\delta, V, P_G) \) is called power flow equations with respect to active power and reactive power, and \( \delta_i = 0 \) for steady state.

We choose \( \sigma = 9 \) in the Cayley transformation and \( \gamma = 10^{-4} \) in the stability constraint. The smoothing technology proposed in [11] is used to obtain the smoothing function of \( \|C(\nabla F(\delta, V))\|_1 \) and to form the smoothing optimization problem.

In the numerical example, inequality constraints in (6.8) are set limits of line transfer power \( P_{\text{line max}} = 2.8 \). The inequality limits are with respect to the active and reactive power of generators, and the voltages and angles of buses. The upper and lower boundaries of them are set as:

\[
\begin{align*}
P_{g_{\min}} &= (0.25, 0.25, 0.25)^T, & P_{g_{\max}} &= (2.5, 2.5, 2.5)^T, \\
Q_{g_{\min}} &= (-0.8, -0.8, -0.8)^T, & Q_{g_{\max}} &= (2.5, 2.5, 2.5)^T, \\
V_{\min} &= 0.9 \times (1, 1, 1, 1, 1, 1)^T, & V_{\max} &= 1.1 \times (1, 1, 1, 1, 1, 1)^T,
\end{align*}
\]
Table 3  Numerical results for OPF problem

<table>
<thead>
<tr>
<th></th>
<th>$W_{min}$</th>
<th>CEV</th>
<th>$P^*_G1$</th>
<th>$P^*_G2$</th>
<th>$P^*_G3$</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without-SC</td>
<td>138.4477</td>
<td>0.1134 ± 4.7099i</td>
<td>0.9221</td>
<td>1.8852</td>
<td>0.3335</td>
<td>40</td>
</tr>
<tr>
<td>$\gamma = 0.003$</td>
<td>138.9961</td>
<td>-0.0599 ± 4.9673i</td>
<td>0.4334</td>
<td>0.2500</td>
<td>2.4837</td>
<td>25</td>
</tr>
<tr>
<td>$\gamma = 0.0003$</td>
<td>138.6684</td>
<td>-0.0528 ± 4.9052i</td>
<td>0.4269</td>
<td>0.5194</td>
<td>2.1973</td>
<td>30</td>
</tr>
<tr>
<td>$\gamma = 0.00003$</td>
<td>138.6159</td>
<td>-0.0452 ± 5.0449i</td>
<td>0.3941</td>
<td>0.5495</td>
<td>2.1976</td>
<td>28</td>
</tr>
</tbody>
</table>

$\delta_{min} = -45 \ast (1, 1, 1, 1, 1, 1, 1)^T$,  \quad \delta_{max} = 45 \ast (1, 1, 1, 1, 1, 1, 1)^T$.

The parameters of Algorithm 4.1 are chosen as follows:

$r_{-1} = 10^3$, \quad $\delta_1 = \delta_2 = 20$, \quad $\beta_\epsilon = 0.1$, \quad $\sigma = 0.5$,

$\tau = 0.8$, \quad $\epsilon_0 = 10^{-4}$.

We use the usual stopping rule in SQP algorithm, i.e., $\|d^k\| \leq 10^{-4}$, which is also the adopted way in reference [9, 15]. It needs to explain that since the initial smoothing parameter is chosen $\epsilon_0 = 10^{-4}$, from the algorithm it holds $\epsilon_k \leq 10^{-4}$. So we omit the stopping rule of smoothing parameter $\epsilon$.

In order to compare the effect of stability constraint, we also consider the ordinary OPF problem in the example, i.e., without including stability constraint. The computing results are reported in Table 3, where $W_{min}$ means the optimal value of the objective; CEV expresses the Critical Eigenvalue, which is the maximal value of the real part of eigenvalues; $P^*_G_i$ is the optimal solution with respect to active power at the $i$-th generator; Iter indicates iterative number by using Algorithm 4.1.

From Table 3 we have the following conclusions:

- Without stability constraint in the OPF problem (denoted as Without-SC in Table 3), we obtain an unsteady optimal solution. On the other hand, we find a stable solution by solving NPSC with differently stable constant $\gamma$;
- The larger the stability constant $\gamma$, the better the resulting stability of the system becomes. Here the stability performance is described via the magnitude of negative real part of the CEV (the Critical Eigenvalue of $\nabla F$).

7 Conclusions

This paper studies a class of nonlinear programs with stability constraints (NPSC). The problem has a strong background and is an extension of ordinary nonlinear programming problems. The models of stability constraints are discussed in detail in our research for stability solution of nonlinear equations. In this paper, we combine the stability constraint into optimization problems, which results in semismooth nonlinear programming problems. Then the optimal conditions and smoothing algorithm are investigated in this paper. The global convergence is also established. A numerical example arising from power systems is tested. The computational result shows
the efficiency of the NPSC model and the algorithm. The further research is to study large-scale NPSC problems.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

