A Neural Network Based on the Metric Projector for Solving SOCCVI Problem

Juhe Sun, Weichen Fu, Jan Harold Alcantara, and Jein-Shan Chen

***Abstract*— We propose an efficient neural network for solving the second-order cone constrained variational inequality (SOCCVI). The network is constructed using the Karush–Kuhn–Tucker (KKT) conditions of the variational inequality (VI), which is used to recast the SOCCVI as a system of equations by using a smoothing function for the metric projection mapping to deal with the complementarity condition. Aside from standard stability results, we explore second-order sufficient conditions to obtain exponential stability. Especially, we prove the nonsingularity of the Jacobian of the KKT system based on the second-order sufficient condition and constraint nondegeneracy. Finally, we present some numerical experiments, illustrating the efficiency of the neural network in solving SOCCVI problems. Our numerical simulations reveal that, in general, the new neural network is more dominant than all other neural networks in the SOCCVI literature in terms of stability and convergence rates of trajectories to SOCCVI solution.**

***Index Terms*— Metric projector, neural network, second-order cone (SOC), second-order sufficient condition, stability, variational inequality (VI).**

I. Introduction

MANY problems in mathematical sciences, such as engineering, optimization, operations research, and economics, among others, can be cast as variational inequalities (VIs). For instance, complementarity problems and some fixed point problems correspond to specific instances of VIs. A detailed discussion of solution methods for VIs can be found in [16] and [19].

In this article, we solve the second-order cone constrained VI (SOCCVI) problem: given a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ and a subset $C \subseteq \mathbb{R}^n$ given as

$$C = \left\{x \in \mathbb{R}^n \mid h(x) = 0, -g(x) \in \mathcal{K}\right\}$$

where $h : \mathbb{R}^n \to \mathbb{R}^l$ ($l \geq 0$) and $g : \mathbb{R}^n \to \mathbb{R}^m$ ($m \geq 1$), the SOCCVI problem is to obtain a point $x \in C$ with the property that for all $y \in C$

$$\langle F(x), \text{y-x}\rangle \geq 0. \tag{1}$$

Manuscript received October 7, 2019; revised April 5, 2020; accepted July 6, 2020. Date of publication August 5, 2020; date of current version July 7, 2021. The work of Juhe Sun was supported by the National Natural Science Foundation of China under Grant 11301348. The work of Jein-Shan Chen was supported by the Ministry of Science and Technology, Taiwan. *(Corresponding author: Jein-Shan Chen.)*

Juhe Sun and Weichen Fu are with the School of Science, Shenyang Aerospace University, Shenyang 110136, China.

Jan Harold Alcantara and Jein-Shan Chen are with the Department of Mathematics, National Taiwan Normal University, Taipei 11677, Taiwan (e-mail: 80640005S@ntnu.edu.tw; jschen@math.ntnu.edu.tw).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3008661

Here, $\langle \cdot, \cdot \rangle$ is the usual inner product, and $\mathcal{K}$ is given by

$$\mathcal{K} = \mathcal{K}^{m_1} \times \cdots \times \mathcal{K}^{m_p} \tag{2}$$

where $m_i \geq 1$, $m_1 + \cdots + m_p = m$, and each $\mathcal{K}^{m_i}$ is a second-order cone (SOC)

$$\mathcal{K}^{m_i} := \{(x_{i1}, x_{i2}, \ldots, x_{im_i})^{\mathrm{T}} \in \mathbb{R}^{m_i} \mid \|(x_{i2}, \ldots, x_{im_i})\| \leq x_{i1}\}$$

where $\|\cdot\|$ is the usual Euclidean norm and $\mathcal{K}^1$ is defined to be the set of nonnegative real numbers. Note that a special case of (2) is when $p = n$ and $m_1 = \cdots = m_p = 1$, which corresponds to the nonnegative orthant $\mathcal{K} = \mathbb{R}^n_+$. Throughout this article, we assume continuous differentiability of F, and twice continuous differentiability of h and g. We also denote $g(x) = (g_{m_1}(x), \ldots, g_{m_p}(x))^{\mathrm{T}}$ and $g_{m_i} = (g^i_0, \bar{g}^i) : \mathbb{R}^n \to \mathbb{R}^{m_i}$ for $i \in \{1, \ldots, p\}$.

A convex SOC program (CSOCP), which is given by

$$\begin{aligned} &\min \ f(x) \\ &\text{s.t. } Ax = b - g(x) \in \mathcal{K} \end{aligned} \tag{3}$$

is a special case of the SOCCVI (1). In (3), we assume that $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable convex function, $g : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable, A is an $l \times n$ matrix with full row rank, and $b \in \mathbb{R}^l$. Indeed, by looking into the Karush–Kuhn–Tucker (KKT) conditions, the CSOCP (3) is equivalent to the SOCCVI problem (1) with $F(x) = \nabla f(x)$ and $h(x) = Ax - b$. This special case has wide applications in management science and engineering [1], [24], [27].

Because of various applications, there have been significant research efforts on computational approaches to VIs and complementarity problems (see [5], [7], [9], [12], [16], [19], [37] and references therein). One main issue, however, is that these methods usually do not provide real-time solutions, which is necessary especially in scientific and engineering applications. Fortunately, we can obtain real-time solutions by utilizing neural networks applied to optimization. This approach was first introduced by Hopfield and Tank [20], [36] in the field of optimization and, since then, has been applied to several optimization problems (see [4], [8], [13]–[15], [18], [21]–[23], [26], [38], [39], [42], [43] and references therein). In this approach, the key is to set up an energy function, which is then used to formulate a system of first-order differential equations, which is a representation of an artificial neural network. Under stability conditions, the neural network converges to a stationary solution of the differential equation, which, in turn, is a possible solution to the mathematical programming problem.

Neural networks have already been used to solve the CSOCP (3), which is a special case of (1), as mentioned earlier. In [24], two kinds of neural networks for CSOCP (3), where $g(x) = -x$ using the smoothed Fischer–Burmeister (FB) function and the projection mapping, were proposed. More general neural models to efficiently solve (3) were proposed in [29] and [30]. Meanwhile, there has also been a plethora of research works making use of neural models to solve more general VIs (see [21], [22], [38] and references therein). However, in the case of SOCCVI (1), only four neural networks exist in the literature. The first two of which were designed in [33]. One of them is constructed using the FB function to obtain a merit function for corresponding KKT conditions, while the other one is constructed by using a projection map to obtain a reformulation of the SOCCVI as a system of equations. In both models, the equilibrium solutions of the network are candidate solutions of (1). The other two neural networks that were used in [34] are inspired by the construction of the first neural network in [33]. Instead of the FB function, two newly discovered SOC-complementarity functions of discrete-type were used to construct the merit functions. Recently, a neural network that is supposed to solve SOCCVI (1) was proposed in [31]. However, we wish to point out that the presented model in [31] is in fact equipped to solve only the CSOCP (3).

In summary, the current literature on the SOCCVI problem is very limited, and the analysis of existing models that have been studied so far is based on the first-order necessary conditions. To the best of our knowledge, there is no existing literature on second-order sufficient conditions for the SOCCVI problem. Apart from limitations of theoretical analysis to first-order conditions, the above-mentioned neural networks considered in [34] and [35] have some disadvantages, such as sensitivity to initial conditions, oscillating solutions, and long convergence time. One other major shortcoming of these neural models is their complete failure to solve some SOCCVI. Motivated by these, we present another neural network for solving the SOCCVI problem based on the smoothing metric projector. One main theoretical contribution of this article arising from formulating this new neural network is the exploration of second-order conditions to achieve exponential stability, which has not been done in the past, as mentioned earlier. On the other hand, from a numerical point of view, the major merit of the proposed neural network is that it addresses the inadequacies and shortcomings of the current models in [34] and [35].

This article is organized as follows. In Section II, we present some mathematical preliminaries pertaining to the second-order cone. In Section III, we present our new neural network and provide conditions to achieve different kinds of stability. We shall note that the stability analysis of the network is analogous to the analyses presented in our earlier works [33], [34]. However, we also present in Section IV a rigorous analysis on how to achieve the conditions that are required to obtain a special type of stability, namely, exponential stability. In particular, it is well-known in the neural network literature that nonsingularity is significant to guarantee exponential stability. Hence, we look at the Jacobian of the KKT system corresponding to (1) and provide a sufficient requirement for its nonsingularity. Finally, in Section V, we provide numerical reports on the performance of the neural network in solving the SOCCVI.

II. Preliminaries

In this section, we review important concepts associated with SOCs (2). Most of these materials can be found in [3].

For any two vectors $x = (x_0, \bar{x})$ and $y = (y_0, \bar{y})$ in $\mathbb{R} \times \mathbb{R}^{m-1}$, the Jordan product of x and y is denoted by $x \circ y := (x^{\mathrm{T}}y, y_0\bar{x} + x_0\bar{y})$. With this Jordan product, the pair $(\mathbb{R} \times \mathbb{R}^{m-1}, \circ)$ is a Jordan algebra with $e = (1, 0, \ldots, 0)^{\mathrm{T}} \in \mathbb{R} \times \mathbb{R}^{m-1}$. We shall denote $x \circ x$ by x^2, which is known to belong to $\mathcal{K}^m$ for any $x \in \mathbb{R}^m$. The square root of a vector in $\mathcal{K}^m$ is also well-defined since there always exists a unique point in $\mathcal{K}^m$ (which we denote by $x^{1/2}$ or $\sqrt{x}$) such that $x = (x^{1/2})^{1/2}$. We also denote $|x| := (x^2)^{1/2}$.

Any $x = (x_0, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{m-1}$ has the following spectral decomposition:

$$x = \lambda_1(x)c_1(x) + \lambda_2(x)c_2(x) \tag{4}$$

where λ_1 and λ_2 are the spectral values of x with formulas

$$\lambda_i(x) = x_0 + (-1)^i\|\bar{x}\| \quad (i = 1, 2) \tag{5}$$

while c_1, c_2 are the spectral vectors associated with x given by

$$c_i(x) = \begin{cases} \dfrac{1}{2}\left(1, (-1)^i\dfrac{\bar{x}}{\|\bar{x}\|}\right) & \text{if} \quad \bar{x} \neq 0, \\ \dfrac{1}{2}(1, (-1)^i w), & \text{if} \quad \bar{x} = 0 \end{cases} \qquad (i = 1, 2) \tag{6}$$

where w is an arbitrary unit vector in $\mathbb{R}^{m-1}$.

Given the spectral decomposition of x as in (4), the projection $\Pi_{\mathcal{K}^m}(x)$ of x onto $\mathcal{K}^m$ is

$$\Pi_{\mathcal{K}^m}(x) = \max\{0, \lambda_1(x)\}\, c_1(x) + \max\{0, \lambda_2(x)\}\, c_2(x). \tag{7}$$

Indeed, plugging in $\lambda_i(x)$ and $c_i(x)$ given in (5) and (6), respectively, yields

$$\Pi_{\mathcal{K}^m}(x) = \begin{cases} \dfrac{1}{2}\left(1 + \dfrac{x_0}{\|\bar{x}\|}\right)(\|\bar{x}\|, \bar{x}), & \text{if} \quad |x_0| < \|\bar{x}\| \\ (x_0, \bar{x}), & \text{if} \quad \|\bar{x}\| \leqslant x_0 \\ 0, & \text{if} \quad \|\bar{x}\| \leqslant -x_0. \end{cases}$$

The following proposition gives a formula for the directional derivative of the mapping given by (7). In what follows, we denote by $\mathrm{int}K$, $\mathrm{bd}K$, and $\mathrm{cl}K$ as the interior, boundary, and closure of a set $K \subset \mathbb{R}^n$, respectively.

Lemma 1: [44, Lemma 2] $\Pi_{\mathcal{K}^m}(\cdot)$ is directionally differentiable at x for any $d \in \mathbb{R}^m$. Moreover, the directional derivative is described by

$$\Pi'_{\mathcal{K}^m}(x; d) = \begin{cases} J\Pi_{\mathcal{K}^m}(x)d, & \text{if } x \in \mathbb{R}^m \backslash (\mathcal{K}^m \cup -\mathcal{K}^m) \\ d, & \text{if } x \in \mathrm{int}\mathcal{K}^m \\ d - 2\left[c_1(x)^{\mathrm{T}}d\right]_- c_1(x), & \text{if } x \in \mathrm{bd}\mathcal{K}^m \backslash \{0\} \\ 0, & \text{if } x \in -\mathrm{int}\mathcal{K}^m \\ 2\left[c_2(x)^{\mathrm{T}}d\right]_+ c_2(x), & \text{if } x \in -\mathrm{bd}\mathcal{K}^m \backslash \{0 \\ \Pi_{\mathcal{K}^m}(d), & \text{if } x = 0 \end{cases}$$

where

$$J\Pi_{\mathcal{K}^m}(x) = \frac{1}{2}\begin{pmatrix} 1 & \dfrac{\bar{x}^{\mathrm{T}}}{\|\bar{x}\|} \\ \dfrac{\bar{x}}{\|\bar{x}\|} & I + \dfrac{x_0}{\|\bar{x}\|}I - \frac{x_0}{\|\bar{x}\|}\cdot\dfrac{\bar{x}\bar{x}^{\mathrm{T}}}{\|\bar{x}\|^2} \end{pmatrix}$$

$$\left[c_1(x)d^{\mathrm{T}}\right]_- := \min\left\{0, c_1(x)^{\mathrm{T}}d\right\}$$

$$\left[c_2(x)d^{\mathrm{T}}\right]_+ := \max\left\{0, c_2(x)^{\mathrm{T}}d\right\}.$$

For convenience in subsequent discussion, we state the definitions of the tangent, regular, and normal cones of a closed set at a point. These concepts can be found in [32]. For a closed set $K \subseteq \mathbb{R}^n$ and a point $\bar{x} \in K$, we define the following sets:

1) the tangent (Bouligand) cone

$$T_K(\bar{x}) := \limsup_{t\downarrow 0} \frac{K - \bar{x}}{t}$$

2) the regular (Fréchet) normal cone

$$\hat{N}_K(\bar{x}) := \{v \in \mathbb{R}^n \big| \langle v, y - \bar{x}\rangle \leqslant o(\|y - \bar{x}\|) \quad \forall y \in K\}$$

3) the limiting (in the sense of Mordukhovich) normal cone $N_K(\bar{x})$: and $T^2_{\mathcal{K}^m}(x, d)$, as shown at the bottom of the page.

When K is a closed convex set, it is known that $T_K(\bar{x}) = \mathrm{cl}(K + \mathbb{R}\bar{x})$ and $\hat{N}_K(\bar{x}) = N_K(\bar{x}) = T_K(\bar{x})^\circ = \{v \in K^\circ | \langle v, x\rangle \leqslant 0\}$, where K° denotes the polar of K.

The tangent and second-order tangent cones are explicitly known as stated in the following result.

Lemma 2: [2, Lemma 2.5] The tangent and second-order tangent cones of $\mathcal{K}^m$ at $x \in \mathcal{K}^m$ are described, respectively, by

$$T_{\mathcal{K}^m}(x) = \begin{cases} \mathbb{R}^m, & \text{if } x \in \mathrm{int}\mathcal{K}^m \\ \mathcal{K}^m, & \text{if } x = 0 \\ \{d = (d_0, \bar{d}) \in \mathbb{R}\times\mathbb{R}^{m-1} & \\ |\langle \bar{d}, \bar{x}\rangle - x_0 d_0 \leqslant 0\}, & \text{if } x \in \mathrm{bd}\mathcal{K}^m \backslash \{0\} \end{cases}$$

and

We close this section by introducing some notations that will be used throughout this article. Given a sequence $\{t_n\} \in \mathbb{R}$, we write $t_n \downarrow 0$ to mean that $\{t_n\}$ is monotone decreasing and converges to zero. The distance from a point x to a set $K \subset \mathbb{R}^n$, denoted by $\mathrm{dist}(x, K)$, is given by

$$\mathrm{dist}(\bar{x}, K) := \inf\{\|\bar{x} - \bar{y}\| : \forall \bar{y} \in K\}.$$

By $\mathrm{lin}K$, we mean the linear subspace generated by K. Given $x, y \in \mathbb{R}^n$, we write $x \perp y$ if and only if $\langle x, y\rangle = 0$. For a function $f : \mathbb{R}^n \to \mathbb{R}$, we denote by $\nabla f(x)$ and $\nabla^2 f(x)$ the gradient and Hessian of f, respectively. Finally, given a function $F : \mathbb{R}^n \to \mathbb{R}^m$, we denote by $JF(x)$ the Jacobian of F, and we let $\nabla F(x) = JF(x)^{\mathrm{T}}$. To emphasize that the derivative is taken with respect to x, we write $J_x F(x)$ and $\nabla_x F(x)$, respectively.

III. Model and Stability Analysis

Similar to the neural networks in [34] and [35], we use the KKT conditions of the SOCCVI (1) to construct a neural network. Recall that the VI Lagrangian function is given by

$$L(x, \mu, \lambda) = F(x) + \nabla h(x)\mu + \nabla g(x)\lambda \tag{8}$$

with $\mu \in \mathbb{R}^l$ and $\lambda \in \mathbb{R}^m$. Then, the KKT system of (1) is described by

$$\begin{cases} L(x, \mu, \lambda) = 0 \\ \langle g(x), \lambda\rangle = 0, \ -g(x) \in \mathcal{K}, \ \lambda \in \mathcal{K} \\ h(x) = 0. \end{cases} \tag{9}$$

We formulate a neural network that can solve the system (9), which are the candidate solutions of the SOCCVI (1). First, to achieve the complementarity requirement in system (9), we may use an SOC-complementarity function $\phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$, i.e., a function such that $\phi(x, y) = 0$ if and only if $x \in \mathcal{K}^m$, $y \in \mathcal{K}^m$ and $\langle x, y\rangle = 0$. Two popular examples are the FB function

$$\phi_{\mathrm{FB}}(x, y) := (x^2 + y^2)^{1/2} - (x + y)$$

and the natural residual (NR) function [17]

$$\phi_{\mathrm{NR}}(x, y) := x - \Pi_{\mathcal{K}^m}(x - y) \tag{10}$$

where $\Pi_{\mathcal{K}^m}$ is the metric projector given by (7). Both these functions are nonsmooth. In [33], a smoothed FB function given by

$$\phi^{\varepsilon}_{\mathrm{FB}}(x, y) = (x^2 + y^2 + \varepsilon^2 e)^{1/2} - (x + y) \tag{11}$$

was employed to construct a merit function for (9), which was the basis to design the neural network involving a smoothing parameter ε. We do note that $\phi^{\varepsilon}_{\mathrm{FB}}$ is not an SOC-complementarity function.

On the other hand, "discrete" generalizations of the FB and NR function were used in [34] to design neural networks, which are given, respectively, by

$$\phi^p_{\mathrm{D-FB}}(x, y) = \left(\sqrt{x^2 + y^2}\right)^p - (x + y)^p \tag{12}$$

and

$$\phi^p_{\mathrm{NR}}(x, y) = x^p - [(x - y)_+]^p \tag{13}$$

where $p > 1$ is an odd integer in both cases. These discrete generalizations are continuously differentiable functions, which makes them suitable for neural network approaches.

In this article, we use a smoothed natural residual function to design a neural model. We begin with a smoothing metric projector function $\Phi : \mathbb{R}_+ \times \mathbb{R}^m \to \mathbb{R}^m$ given by

$$\Phi(\varepsilon, u) := \frac{1}{2}\left(u + \sqrt{\varepsilon^2 e + u^2}\right) \quad \forall (\varepsilon, u) \in \mathbb{R}_+ \times \mathbb{R}^m. \tag{14}$$

$$N_K(\bar{x}) := \limsup_{x \to \bar{x}^K} \hat{N}_K(x).$$

$$T^2_{\mathcal{K}^m}(x, d) = \begin{cases} \mathbb{R}^m, & \text{if } x \in \mathrm{int}T_{\mathcal{K}^m}(x) \\ T_{\mathcal{K}^m}(d), & \text{if } x = 0 \\ \{w = (w_0, \bar{w}) \in \mathbb{R}\times\mathbb{R}^{m-1} | \langle \bar{w}, \bar{s}\rangle - w_0 x_0 \leqslant d_0^2 - \|\bar{d}\|^2\}, & \text{otherwise.} \end{cases}$$

Observe that $\Phi(0,u)=\Pi_{\mathcal{K}^m}(u)$. Moreover, Φ is continuously differentiable on any neighborhood of $(\varepsilon,u)\in\mathbb{R}\times\mathbb{R}^m$ provided that $(\varepsilon^2e+u^2)_0\neq\left\|\varepsilon^2e+u^2\right\|$. From [22], it is known that Φ is globally Lipschitz continuous and is strongly semismooth for all $(0,u)\in\mathbb{R}\times\mathbb{R}^m$. Furthermore, applying the concept of SOC-functions in [6], [10], and [11], it can be verified that the function $\Phi(\varepsilon,u)$ given in (14) can alternatively be expressed as

$$\Phi(\varepsilon,u)=\phi(\varepsilon,\lambda_1)c_1+\phi(\varepsilon,\lambda_2)c_2 \tag{15}$$

where $\phi(\varepsilon,t):=(1/2)(t+(\varepsilon^2+t^2)^{1/2})$, where λ_i and c_i are given in (5) and (6), respectively. Hence, we can write out the function Φ as

$$\Phi(\varepsilon,u)=\begin{cases}\dfrac{1}{2}u+\dfrac{1}{4}\begin{pmatrix}\sqrt{\varepsilon^2+\lambda_1^2}+\sqrt{\varepsilon^2+\lambda_2^2}\\ \left(\sqrt{\varepsilon^2+\lambda_1^2}+\sqrt{\varepsilon^2-\lambda_2^2}\right)\frac{\bar{u}}{\|\bar{u}\|}\end{pmatrix}, & \text{if } \bar{u}\neq 0\\ \dfrac{1}{2}\begin{pmatrix}u_0+\sqrt{\varepsilon^2+u_0^2}\\ 0\end{pmatrix}, & \text{if } \bar{u}=0.\end{cases} \tag{16}$$

For $(\varepsilon^2e+u^2)_0\neq\left\|\varepsilon^2e+u^2\right\|$, we calculate the derivative of Φ with respect to ε as follows:

$$\begin{aligned}\nabla_\varepsilon\Phi(\varepsilon,u)&=\frac{1}{2}\left(\frac{\partial}{\partial\varepsilon}\phi(\varepsilon,\lambda_1)c_1^{\mathrm{T}}+\frac{\partial}{\partial\varepsilon}\phi(\varepsilon,\lambda_2)c_2^{\mathrm{T}}\right)\\&=\frac{1}{2}\left(\frac{\varepsilon c_1^{\mathrm{T}}}{\sqrt{\varepsilon^2+\lambda_1^2}}+\frac{\varepsilon c_2^{\mathrm{T}}}{\sqrt{\varepsilon^2+\lambda_2^2}}\right).\end{aligned}$$

As for the differential with respect to u, we have two cases.

1) For $u\neq 0$

$$\nabla_u\Phi(\varepsilon,u)=\frac{1}{2}\begin{pmatrix}1+\dfrac{1}{2}\left(\dfrac{\lambda_1}{\sqrt{\varepsilon^2+\lambda_1^2}}+\dfrac{\lambda_2}{\sqrt{\varepsilon^2+\lambda_2^2}}\right) & Y^{\mathrm{T}}\\ Y & Z\end{pmatrix} \tag{17}$$

where

$$Y=\frac{1}{2}\left[\frac{\lambda_2}{\sqrt{\varepsilon^2+\lambda_2^2}}-\frac{\lambda_1}{\sqrt{\varepsilon^2+\lambda_1^2}}\right]\frac{\bar{u}}{\|\bar{u}\|}$$

and

$$\begin{aligned}Z=&\left[1+\frac{\sqrt{\varepsilon^2+\lambda_2^2}-\sqrt{\varepsilon^2+\lambda_1^2}}{\lambda_2-\lambda_1}\right]I_{m-1}\\&+\left[\frac{1}{2}\left(\frac{\lambda_1}{\sqrt{\varepsilon^2+\lambda_1^2}}+\frac{\lambda_2}{\sqrt{\varepsilon^2+\lambda_2^2}}\right)\right.\\&\left.-\frac{\sqrt{\varepsilon^2+\lambda_2^2}-\sqrt{\varepsilon^2+\lambda_1^2}}{\lambda_2-\lambda_1}\right]\frac{\bar{u}\bar{u}^{\mathrm{T}}}{\|\bar{u}\|^2}.\end{aligned}$$

2) For $\bar{u}=0$

$$\nabla_u\Phi(\varepsilon,u)=\frac{1}{2}\left[1+\frac{u_0}{\sqrt{\varepsilon^2+u_0^2}}\right]I_m.$$

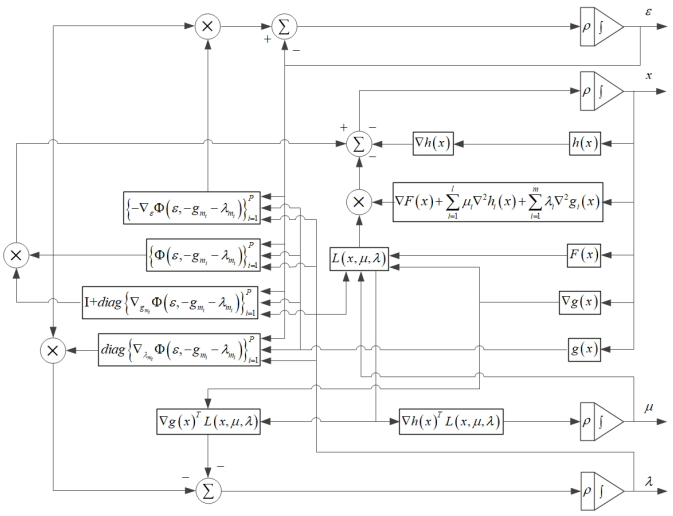

Fig. 1. Block diagram of the proposed neural network with $\phi_{\mathrm{NR}}^{\varepsilon}$.

For $(\varepsilon^2e+u^2)_0=\left\|\varepsilon^2e+u^2\right\|$, Φ is nonsmooth at (ε,u), but its B-subdifferential can nevertheless be computed.

According to the above-mentioned $\Phi(\varepsilon,u)$ given in (14), (15), or (16), we introduce the smoothing NR function given as

$$\phi_{\mathrm{NR}}^{\varepsilon}(x,y)=x-\Phi(\varepsilon,x-y) \tag{18}$$

which is the basis of our neural network. Now, define $S:\mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^m\to\mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^m$ by

$$S(z)=\begin{bmatrix}\varepsilon\\ L(x,\mu,\lambda)\\ h(x)\\ \phi_{\mathrm{NR}}^{\varepsilon}\left(-g_{m_1}(x),\lambda_{m_1}\right)\\ \vdots\\ \phi_{\mathrm{NR}}^{\varepsilon}\left(-g_{m_p}(x),\lambda_{m_p}\right)\end{bmatrix}$$

where $z=(\varepsilon,x,\mu,\lambda)\in\mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^m$

Then, it is clear to see that solving (9) is equivalent to solving the problem

$$\min\Psi(z):=\frac{1}{2}\|S(z)\|^2. \tag{19}$$

Hence, Ψ is a merit function for (9), and in turn, we consider the dynamical system given by

$$\begin{cases}\dfrac{dz(t)}{dt} &= -\rho\,\nabla\Psi(z(t))=-\rho\nabla S(z(t))S(z(t))\\ z(t_0) &= z_0\end{cases} \tag{20}$$

where $\rho>0$ is a scaling factor, for solving the SOCCVI. We refer to the above as "the smoothed NR neural network." The block diagram of the above-mentioned neural network is shown in Fig. 1. The circuit for (20) requires $n+l+m+1$ integrators, n processors for $F(x)$, m processors for $g(x)$, mn processors for $\nabla g(x)$, l processors for $h(x)$, ln processors for $\nabla h(x)$, $(1+m+l)n^2$ processors for $\nabla_xL(x,\mu,\lambda)$, $2m+2\sum_{i=1}^{p}m_i^2$ processors for Φ and its derivatives, and some analog multipliers and summers.

Let $u_{m_i} = -g_{m_i}(x) - \lambda_{m_i}$. For subsequent use in the numerical simulations, we shall note that, $\nabla S(z)$, as shown at the bottom of the page. It is clear that Ψ is a nonnegative function that attains the value 0 at $z = (\varepsilon, x, \mu, \lambda)$ if and only if (x, μ, λ) is a KKT point. Moreover, KKT points are equilibrium points of (20), and the converse holds if we have the nonsingularity of $\nabla S(z)$.

The stability analysis of the above-mentioned system (20) is fairly standard and is analogous to the analysis of the smoothed FB neural network in [33]. However, we point out that our main contributions are as follows: 1) in Section IV, we look into second-order sufficient conditions for nonsingularity and 2) in Section V, we demonstrate that our neural model has better numerical properties among all neural networks for SOCCVI problems. For the sake of completeness, we present here a fundamental stability result, whose proof is similar to earlier works (for instance, [33]) and is, therefore, omitted.

Theorem 1: Isolated equilibrium points of (20) are asymptotically stable. Moreover, we obtain exponentially stability if $\nabla S(z)$ is nonsingular.

From Theorem 1, we see the importance of the nonsingularity of the transposed Jacobian of S, namely, $\nabla S(z)$. We explore sufficient conditions to guarantee this property in Section IV.

IV. Second-Order Sufficient Condition and Nonsingularity Theorem

This section is devoted to deriving the second-order sufficient condition for (1) and building up some conditions to achieve the nonsingularity of $\nabla S(0, x^*, \mu^*, \lambda^*)$. To this end, we write out the first-order optimality conditions for the SOCCVI problem (1). Let $L(x, \mu, \lambda)$ be given by (8) and let $(\mu, \lambda) = (\mu, \lambda_{m_1}, \ldots, \lambda_{m_p}) \in \mathbb{R}^l \times \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_p} = \mathbb{R}^l \times \mathbb{R}^m$. Suppose that x^* is a solution of (1), and Robinson's constraint qualification

$$\begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} \\ -\nabla g(x^*)^{\mathrm{T}} \end{pmatrix} \mathbb{R}^n + T_{\{0_l\}\times\mathcal{K}}(h(x^*), -g(x^*)) = \mathbb{R}^l \times \mathbb{R}^m$$

holds at x^*. The first-order optimality condition is

$$\langle F(x^*), d\rangle \geq 0 \forall d \in T_C(x^*) \tag{21}$$

where

$$T_C(x^*) = \{d \mid \nabla h(x^*)^{\mathrm{T}} d = 0 - \nabla g(x^*)^{\mathrm{T}} d \in T_{\mathcal{K}}(-g(x^*))\}.$$

It is known that $T_C(x^*)$ is convex and

$$N_C(x^*) = \nabla h(x^*)\mathbb{R}^l + \{\nabla g(x^*)\lambda \mid -\lambda \in N_{\mathcal{K}}(-g(x^*))\}$$

where $N_{\mathcal{K}}(y) := N_{\mathcal{K}^{m_1}}(y_{m_1}) \times N_{\mathcal{K}^{m_2}}(y_{m_2}) \times \cdots \times N_{\mathcal{K}^{m_p}}(y_{m_p})$ for $y = (y_{m_1}, \ldots, y_{m_p}) \in \mathbb{R}^m$, and

$$N_{\mathcal{K}^{m_i}}(y_{m_i}) := \{u_{m_i} \in \mathbb{R}^{m_i} \mid \langle u_{m_i}, v - y_{m_i}\rangle \leq 0 \forall v \in \mathcal{K}^{m_i}$$

is the normal cone of $\mathcal{K}_{m_i}$ at y_{m_i}. Note that (21) holds if and only if $0 \in F(x^*) + N_C(x^*)$, which is equivalent to: $\exists\, \mu \in \mathbb{R}^l$, $\lambda \in \mathbb{R}^m$ such that

$$L(x^*, \mu, \lambda) = 0, \quad -\lambda \in N_{\mathcal{K}}(-g(x^*))$$

and the set of multipliers (μ, λ) denoted by $\Lambda(x^*)$ is nonempty compact. Therefore, x^* satisfies the following KKT condition:

$$\begin{cases} L(x^*, \mu, \lambda) = 0 \\ h(x^*) = 0 \\ -\lambda \in N_{\mathcal{K}}(-g(x^*)). \end{cases}$$

Using the metric projector and the definition of the normal cone, the KKT condition can be expressed as

$$S(x, \mu, \lambda) = \begin{pmatrix} L(x, \mu, \lambda) \\ h(x) \\ -g(x) - \Pi_{\mathcal{K}}(-g(x) - \lambda) \end{pmatrix} = 0$$

where

$$\Pi_{\mathcal{K}}(-g(x) - \lambda) := \left(\Pi_{\mathcal{K}^{m_1}}(-g_{m_1}(x) - \lambda_{m_1})^{\mathrm{T}}, \ldots, \Pi_{\mathcal{K}^{m_p}}(-g_{m_p}(x) - \lambda_{m_p})^{\mathrm{T}}\right)^{\mathrm{T}}.$$

It is particularly emphasized that

$$\Pi'_{\mathcal{K}}(-g(x) - \lambda; d) := \mathrm{diag}\{\Pi'_{\mathcal{K}^{m_1}}(-g_{m_i}(x) - \lambda_{m_i}; d_{m_i})\}_{i=1}^p$$

for $d \in \mathbb{R}^m$.

Before presenting our main results, we recall the following concept needed in the proof.

Definition 1: [2] The critical cone at x^* is defined by

$$\mathcal{C}(x^*) = \{d \mid d \in T_C(x^*), d \perp F(x^*)\}.$$

Theorem 2: Suppose that x^* is a feasible point of the SOCCVI (1) such that $\Lambda(x^*) = \{(\mu, \lambda)\}$ is nonempty and compact. If $JF(x^*)$ is positive semidefinite and Robinson's CQ holds at x^*, then

$$\sup_{(\mu,\lambda)\in\Lambda(x^*)} \{\langle J_x L(x^*, \mu, \lambda)d, d\rangle - \delta^*(\lambda \mid T^2_{\mathcal{K}}(-g(x^*) - \nabla g(x^*)^{\mathrm{T}} d))\} > 0 \ \forall d \in \mathcal{C}(x^*)\backslash\{0\} \tag{22}$$

is the second-order sufficient condition of (1), where

$$\delta^*(\lambda \mid T^2_{\mathcal{K}}(-g(x^*), -\nabla g(x^*)^{\mathrm{T}} d)) = \begin{cases} 0, & \text{if } \lambda \in N_{\mathcal{K}}(-g(x^*)) \text{ and } \langle \lambda, -\nabla g(x^*)^{\mathrm{T}} d\rangle = 0 \\ +\infty, & \text{otherwise.} \end{cases}$$

$$\nabla S(z) = \begin{pmatrix} 1 & 0 & 0 & \{-\nabla_\varepsilon \Phi(\varepsilon, u_{m_i})\}_{i=1}^p \\ 0 & \nabla_x L(x, \mu, \lambda)^{\mathrm{T}} & \nabla h(x) & -\nabla g(x)\left(I - \mathrm{diag}\{\nabla_{u_{m_i}} \Phi(\varepsilon, u_{m_i})\}_{i=1}^p\right) \\ 0 & \nabla h(x)^{\mathrm{T}} & 0 & 0 \\ 0 & \nabla g(x)^{\mathrm{T}} & 0 & \mathrm{diag}\{\nabla_{u_{m_i}} \Phi(\varepsilon, u_{m_i})\}_{i=1}^p \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \{-\nabla_\varepsilon \Phi(\varepsilon, -g_{m_i}(x) - \lambda_{m_i})\}_{i=1}^p \\ 0 & \nabla_x L(x, \mu, \lambda)^{\mathrm{T}} & \nabla h(x) & -\nabla g(x)\left(I + \mathrm{diag}\{\nabla_{g_{m_i}} \Phi(\varepsilon, -g_{m_i}(x) - \lambda_{m_i})\}_{i=1}^p\right) \\ 0 & \nabla h(x)^{\mathrm{T}} & 0 & 0 \\ 0 & \nabla g(x)^{\mathrm{T}} & 0 & -\mathrm{diag}\{\nabla_{\lambda_{m_i}} \Phi(\varepsilon, -g_{m_i}(x) - \lambda_{m_i})\}_{i=1}^p \end{pmatrix}.$$

Proof: Let x^* be a solution of (1). Since $JF(x^*)$ is positive semidefinite, we see that for some small $\varepsilon > 0$

$$\langle F(x^*), x - x^* \rangle \geqslant 0 \ \forall x \in \mathbb{B}_\varepsilon(x^*) \cap C$$

where $\mathbb{B}_\varepsilon(x^*)$ denotes the ε-neighborhood of x^*. Equivalently

$$x^* \in \arg\min\{\langle F(x^*), x - x^* \rangle \,|\, x \in \mathbb{B}_\varepsilon(x^*) \cap C\}. \quad (23)$$

Again, due to $JF(x^*)$ being positive semidefinite, it is clear that (23) holds if and only if

$$x^* \in \arg\min\big\{\langle F(x^*), x - x^* \rangle + \langle JF(x^*)(x - x^*), x - x^* \rangle | \ x \in \mathbb{B}_\varepsilon(x^*) \cap C\big\}. \quad (24)$$

Therefore, we turn to deduce the second-order sufficient condition of (24). To this end, we consider the optimization problem

$$\begin{aligned} &\min \ \langle F(x^*), x - x^* \rangle + \frac{1}{2}\langle JF(x^*)(x - x^*), x - x^* \rangle \\ &\text{s.t. } x \in \mathbb{B}_\varepsilon(x^*) \cap C. \end{aligned} \quad (25)$$

First, it is known that x^* is the stationary point of problem (25) if and only if

$$0 \in F(x^*) + JF(x^*)(x - x^*) + N_{\mathbb{B}_\varepsilon(x^*) \cap C}(x^*) \quad (26)$$

where

$$N_{\mathbb{B}_\varepsilon(x^*) \cap C}(x^*) = N_{\mathbb{B}_\varepsilon(x^*)}(x^*) + N_C(x^*) = N_C(x^*). \quad (27)$$

On the other hand, (26) and (27) imply that $0 \in F(x^*) + N_C(x^*)$. Hence, if x^* is a solution of (1), we conclude that x^* is the stationary point of problem (25).

Now, we prove that the critical cones $\mathcal{C}_p(x^*)$ and $\mathcal{C}(x^*)$ of (25) and (1), respectively, are equal. Indeed

$$\begin{aligned} \mathcal{C}_p(x^*) = \Bigg\{ d \in \mathbb{R}^n \Bigg| & \begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} d \\ -\nabla g(x^*)^{\mathrm{T}} d \\ d \end{pmatrix} \\ & \in T_{\{0\} \times \mathcal{K} \times \mathbb{B}_\varepsilon(x^*)}(h(x^*), -g(x^*), x^*), \text{ and} \\ & \langle d, F(x^*) + JF(x^*)(x - x^*) \rangle = 0 \Bigg\}. \end{aligned}$$

Notice that

$$\begin{aligned} & T_{\{0\} \times \mathcal{K} \times \mathbb{B}_\varepsilon(x^*)}(h(x^*), -g(x^*), x^*) \\ &\quad = T_{\{0\} \times \mathcal{K}}(h(x^*), -g(x^*)) \times T_{\mathrm{B}_\varepsilon(x^*)}(x^*) \\ &\quad = T_{\{0\} \times \mathcal{K}}(h(x^*), -g(x^*)) \times \mathbb{R}^n. \end{aligned}$$

This yields that

$$\begin{aligned} \mathcal{C}_p(x^*) = \Bigg\{ d \in \mathbb{R}^n \Bigg| & \begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} d \\ -\nabla g(x^*)^{\mathrm{T}} d \end{pmatrix} \in T_{\{0\} \times \mathcal{K}}(h(x^*) \\ & -g(x^*)), \langle d, F(x^*) \rangle = 0 \Bigg\} \\ = \mathcal{C}(x^*). \end{aligned}$$

Next, the Lagrange function of problem (25) is

$$\begin{aligned} \mathcal{L}(x^*, \lambda, \mu, \nu) = & \langle F(x^*), (x - x^*) \rangle \\ & + \frac{1}{2}\langle JF(x^*)(x - x^*), x - x^* \rangle \\ & + \langle h(x), \mu \rangle + \langle g(x), \lambda \rangle + \langle x, \nu \rangle \end{aligned}$$

which gives

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, \lambda, \mu, \nu) = & F(x^*) + JF(x^*)(x - x^*) + \nabla h(x)\mu + \nu \\ & + \nabla g(x)\lambda \\ \nabla^2_{xx} \mathcal{L}(x^*, \lambda, \mu, \nu) = & JF(x^*) + \Sigma_{i=1}^{l} \mu_i \nabla^2 \ h_i(x^*) \\ & + \Sigma_{i=1}^{m} \lambda_i \nabla^2 \ g_i(x^*). \end{aligned}$$

Here, we note that $\nabla^2_{xx} \mathcal{L}(x^*, \lambda, \mu, \nu) = J_x L(x^*, \lambda, \mu)$.

On the other hand, in light of [3, Proposition 3.269], we can check that $\{0\} \times \mathcal{K}$ is second-order regular at $(h(x^*), -g(x^*))$ along the direction $(\nabla h(x^*)^{\mathrm{T}} d, -\nabla g(x^*)^{\mathrm{T}} d)$ with respect to the mapping $\begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} \\ -\nabla g(x^*)^{\mathrm{T}} \end{pmatrix}$ for all $d \in \mathcal{C}(x^*)$. Then, using the definition of the second-order regularity (see [3, Definition 3.85]) yields

$$\begin{aligned} y_n = \begin{pmatrix} h(x^*) \\ -g(x^*) \end{pmatrix} + t_n \begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} d \\ -\nabla g(x^*)^{\mathrm{T}} d \end{pmatrix} \\ + \frac{1}{2} {t_n}^2 r_n \quad \forall y_n \in \{0\} \times \mathcal{K} \end{aligned}$$

where $t_n \downarrow 0$, $r_n = \begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} w_n \\ -\nabla g(x^*)^{\mathrm{T}} w_n \end{pmatrix} + a_n$ with a_n being a convergent sequence and $t_n w_n \to 0$, $(n \to +\infty)$ such that

$$\begin{aligned} \lim_{n \to \infty} \mathrm{dist}(r_n, {T^2}_{\{0\} \times \mathcal{K}}((h(x^*), -g(x^*)), (\nabla h(x^*)^{\mathrm{T}} d \\ - \nabla g(x^*)^{\mathrm{T}} d))) = 0. \end{aligned}$$

According to the above-mentioned result, for all $P_n \in \{0\} \times \mathcal{K} \times \mathbb{B}_\varepsilon(x^*)$, we have

$$P_n = \begin{pmatrix} h(x^*) \\ -g(x^*) \\ x^* \end{pmatrix} + t_n \begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} d \\ -\nabla g(x^*)^{\mathrm{T}} d \\ d \end{pmatrix} + \frac{1}{2} {t_n}^2 \begin{pmatrix} r_n \\ q_n \end{pmatrix}$$

where $t_n \downarrow 0$

$$\begin{pmatrix} r_n \\ q_n \end{pmatrix} = \begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} w_n \\ -\nabla g(x^*)^{\mathrm{T}} w_n \\ w_n \end{pmatrix} + \begin{pmatrix} a_n \\ b_n \end{pmatrix}$$

with $\begin{pmatrix} a_n \\ b_n \end{pmatrix}$ being a convergent sequence and $t_n w_n \to 0$, $(n \to +\infty)$. Therefore, we obtain

$$\begin{aligned} \lim_{n \to \infty} \mathrm{dist}(r_n, {T^2}_{\{0\} \times \mathcal{K}}((h(x^*), -g(x^*)), (\nabla h(x^*)^{\mathrm{T}} d \\ - \nabla g(x^*)^{\mathrm{T}} d))) = 0 \end{aligned}$$

and

$$\begin{aligned} & \lim_{n \to \infty} \mathrm{dist}\Bigg(\begin{pmatrix} r_n \\ q_n \end{pmatrix}, {T^2}_{\{0\} \times \mathcal{K} \times \mathbb{B}_\varepsilon(x^*)} \\ & \qquad \times ((h(x^*), -g(x^*), x^*)(\nabla h(x^*)^{\mathrm{T}} d, -\nabla g(x^*)^{\mathrm{T}} d, d))\Bigg) \\ & = \lim_{n \to \infty} \mathrm{dist}\Bigg(\begin{pmatrix} r_n \\ q_n \end{pmatrix}, {T^2}_{\{0\} \times \mathcal{K}}((h(x^*), -g(x^*)), (\nabla h(x^*)^{\mathrm{T}} \\ & \qquad - \nabla g(x^*)^{\mathrm{T}} d)) \times {T^2}_{\mathbb{B}_\varepsilon(x^*)}(x^*, d)\Bigg) \\ & = \lim_{n \to \infty} \mathrm{dist}(r_n, {T^2}_{\{0\} \times \mathcal{K}}((h(x^*), -g(x^*)), (\nabla h(x^*)^{\mathrm{T}} d \\ & \qquad - \nabla g(x^*)^{\mathrm{T}} d))) = 0 \end{aligned}$$

and, thus, $\{0\} \times \mathcal{K} \times \mathbb{B}_\varepsilon(x^*)$ is second-order regular at the point $(h(x^*), -g(x^*), x^*)$ along $(\nabla h(x^*)^{\mathrm{T}} d, -\nabla g(x^*)^{\mathrm{T}} d, d)$

with respect to the mapping $\begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} \\ -\nabla g(x^*)^{\mathrm{T}} \\ I \end{pmatrix}$ for all $d \in \mathcal{C}(x^*)$, with I as the identity map.

This together with [3, Th. 3.86] indicates that for (25), the second-order sufficient condition is

$$\sup_{(\lambda,\mu,\nu)\in\bar{\Lambda}(x^*)} \times \Big\{\nabla^2{}_{xx}\mathcal{L}(x^*,\lambda,\mu,\nu) - \delta^* \times ((\mu,\lambda,\nu), T^2{}_{\{0\}\times\mathcal{K}\times\mathbb{B}_\varepsilon(x^*)} \times ((h(x^*), -g(x^*), x^*), (\nabla h(x^*)^{\mathrm{T}}d - \nabla g(x^*)^{\mathrm{T}}d, d)))\Big\} > 0 \quad \forall d \in \mathcal{C}_p(x^*)\backslash\{0\}.$$

We can further simplify it as

$$\begin{aligned}
&\sup_{(\lambda,\mu,\nu)\in\bar{\Lambda}(x^*)} \Big\{\nabla^2{}_{xx}\mathcal{L}(x^*,\lambda,\mu,\nu)(d,d) - \delta^* \times ((\mu,\lambda,\nu), T^2{}_{\{0\}\times\mathcal{K}\times\mathbb{B}_\varepsilon(x^*)} \times ((h(x^*), -g(x^*), x^*) (\nabla h(x^*)^{\mathrm{T}}, -\nabla g(x^*)^{\mathrm{T}}d, d)))\Big\} \\
&= \sup_{(\lambda,\mu,\nu)\in\bar{\Lambda}(x^*)} \times \Big\{\nabla^2{}_{xx}\mathcal{L}(x^*,\lambda,\mu,\nu)(d,d) - \delta^* \times ((\mu,\lambda,\nu), T^2{}_{\{0\}}(h(x^*), \nabla h(x^*)^{\mathrm{T}}d) \times T^2{}_{\mathcal{K}}(-g(x^*), -\nabla g(x^*)^{\mathrm{T}}d) \times T^2{}_{\mathbb{B}_\varepsilon(x^*)}(x^*, d))\Big\} \\
&= \sup_{(\lambda,\mu,\nu)\in\bar{\Lambda}(x^*)} \times \Big\{\nabla^2{}_{xx}\mathcal{L}(x^*,\lambda,\mu,\nu)(d,d) - \delta^* \times ((\mu,\lambda,\nu), \{0\} \times T^2{}_{\mathcal{K}}(-g(x^*), -\nabla g(x^*)^{\mathrm{T}}d \times \mathbb{R}^n)\Big\} \\
&= \sup_{(\mu,\lambda)\in\Lambda(x^*)} \Big\{J_xL(x^*,\lambda,\mu)(d,d) - \delta^* (\lambda | T^2{}_{\mathcal{K}}(-g(x^*) - \nabla g(x^*)^{\mathrm{T}}d))\Big\}.
\end{aligned}$$

To sum up, the second-order sufficient condition of the SOCCVI (1) is described by

$$\sup_{(\mu,\lambda)\in\Lambda(x^*)} \Big\{\big\langle J_xL(x^*,\lambda,\mu)d, d\big\rangle - \delta^* \times (\lambda | T^2{}_{\mathcal{K}}(-g(x^*), -\nabla g(x^*)^{\mathrm{T}}d))\Big\} > 0 \quad \forall d \in \mathcal{C}(x^*)\backslash\{0\}$$

as desired. ∎

As we saw in Theorem 1, $\nabla S(0, x^*, \mu^*, \lambda^*)$ being nonsingular is crucial to guarantee that the equilibrium point of our network becomes a solution of the SOCCVI (1) and that it is exponential stable. Now, we present some conditions to achieve the nonsingularity of $\nabla S(0, x^*, \mu^*, \lambda^*)$.

Theorem 3: Suppose that (x^*, μ^*, λ^*) is a KKT point of (1). Then, $\nabla S(0, x^*, \mu^*, \lambda^*)$ is nonsingular if the following holds.

1) $\Lambda(x^*) \neq \emptyset$.
2) The second-order sufficient condition (22) holds.
3) $-\lambda^* \in \mathrm{int}N_{\mathcal{K}}(-g(x^*))$ holds.
4) The following constraint nondegeneracy holds

$$\begin{pmatrix} \nabla h(x^*)^{\mathrm{T}} \\ -\nabla g(x^*)^{\mathrm{T}} \end{pmatrix}\mathbb{R}^n + \mathrm{lin}T_{\{0_l\}\times\mathcal{K}}(h(x^*), -g(x^*)) = \mathbb{R}^l \times \mathbb{R}^m.$$

Proof: It is enough to verify that M given as shown at the bottom of the page is nonsingular, where $u^*_{m_i} = -g_{m_i}(x^*) - \lambda^*_{m_i}$. From Lemma 1 and (17), we can deduce that

$$\lim_{\varepsilon\to 0}[\nabla_u\Phi(\varepsilon, u)]^{\mathrm{T}}d = \Pi'_{\mathcal{K}^m}(u; d)$$

for $d \in \mathbb{R}^m$ and $u \in \mathbb{R}^m\backslash(\mathcal{K}^m \cup -\mathcal{K}^m)$ or $u \in \mathrm{int}\mathcal{K}^m$. Then, for $\varepsilon \to 0$ and $(dx, d\mu, d\lambda) \in \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^m$, we have

$$M\begin{pmatrix} dx \\ d\mu \\ d\lambda \end{pmatrix} = \begin{pmatrix} J_xL(x^*,\mu^*,\lambda^*)dx + \nabla h(x^*)d\mu + \nabla g(x^*)d\lambda \\ \nabla h(x^*)^{\mathrm{T}}dx \\ H \end{pmatrix}$$

where

$$\begin{aligned}
H &= \left(I - \mathrm{diag}\left\{\lim_{\varepsilon\to 0}\nabla_{u^*_{m_i}}\Phi\big(\varepsilon, u^*_{m_i}\big)\right\}_{i=1}^{p}\right)^{\mathrm{T}}(-\nabla g(x^*))^{\mathrm{T}}dx \\
&\quad + \left(\mathrm{diag}\left\{\lim_{\varepsilon\to 0}\nabla_{u^*_{m_i}}\Phi\big(\varepsilon, u^*_{m_i}\big)\right\}_{i=1}^{p}\right)^{\mathrm{T}}d\lambda \\
&= -\nabla g(x^*)^{\mathrm{T}}dx - \left(\mathrm{diag}\left\{\lim_{\varepsilon\to 0}\nabla_{u^*_{m_i}}\Phi\big(\varepsilon, u^*_{m_i}\big)\right\}_{i=1}^{p}\right)^{\mathrm{T}} \times [-\nabla g(x^*)^{\mathrm{T}}dx - d\lambda] \\
&= -\nabla g(x^*)^{\mathrm{T}}dx - \Pi'_{\mathcal{K}}(-g(x^*) - \lambda^*; -\nabla g(x^*)^{\mathrm{T}}dx - d\lambda).
\end{aligned}$$

Therefore, we have

$$M\begin{pmatrix} dx \\ d\mu \\ d\lambda \end{pmatrix} = \begin{pmatrix} J_xL(x^*,\mu^*,\lambda^*)dx + \nabla h(x^*)d\mu + \nabla g(x^*)d\lambda \\ \nabla h(x^*)^{\mathrm{T}}dx \\ -\nabla g(x^*)^{\mathrm{T}}dx - \Pi'_{\mathcal{K}}\big(-g(x^*) - \lambda^*; -\nabla g(x^*)^{\mathrm{T}}dx - d\lambda\big) \end{pmatrix}. \tag{28}$$

Suppose that $M\begin{pmatrix} dx \\ d\mu \\ d\lambda \end{pmatrix} = 0$. We need to show that $dx = 0$, $d\mu = 0$, and $d\lambda = 0$. First, from the second and thirs expressions of (28), we obtain

$$\begin{cases} \nabla h(x^*)^{\mathrm{T}}dx = 0 \\ -\nabla g(x^*)^{\mathrm{T}}dx = \Pi'_{\mathcal{K}}\big(-g(x^*) - \lambda^*; -\nabla g(x^*)^{\mathrm{T}}dx - d\lambda\big) \end{cases} \tag{29}$$

which implies that $dx \in \mathcal{C}(x^*)$. In addition, from the first expression of (28), we obtain

$$\big\langle J_xL(x^*,\mu^*,\lambda^*)dx, dx\big\rangle + \big\langle \nabla g(x^*)^{\mathrm{T}}dx, d\lambda\big\rangle = 0. \tag{30}$$

$$M = \begin{pmatrix} \nabla_x L(x^*,\mu^*,\lambda^*)^{\mathrm{T}} & \nabla h(x^*) & -\nabla g(x^*)\left(I - \mathrm{diag}\left\{\lim_{\varepsilon\to 0}\nabla_{u^*_{m_i}}\Phi\big(\varepsilon, u^*_{m_i}\big)\right\}_{i=1}^{p}\right) \\ \nabla h(x^*)^{\mathrm{T}} & 0 & 0 \\ \nabla g(x^*)^{\mathrm{T}} & 0 & \mathrm{diag}\left\{\lim_{\varepsilon\to 0}\nabla_{u^*_{m_i}}\Phi\big(\varepsilon, u^*_{m_i}\big)\right\}_{i=1}^{p} \end{pmatrix}^{\mathrm{T}}$$

To proceed, we consider the following sets:

$$I^* = \{i \mid -g_{m_i}(x^*) \in \text{int}\mathcal{K}^{m_i}, i = 1, \ldots, p\}$$
$$B^* = \{i \mid -g_{m_i}(x^*) \in \text{bd}\mathcal{K}^{m_i}, g_{m_i}(x^*) \neq 0\}$$
$$Z^* = \{i \mid g_{m_i}(x^*) = 0\}.$$

Note that

$$\mathcal{C}_{\mathcal{K}}(-g(x^*)) = \{d \in \mathbb{R}^n \mid -\nabla g(x^*)^{\mathrm{T}} d \in T_{\mathcal{K}}(-g(x^*))\}$$

and

$$T_{\mathcal{K}}(-g(x^*)) = \left\{ d \;\middle|\; \begin{array}{c} -\nabla g_0^i(x^*)^{\mathrm{T}} d - \dfrac{\nabla \bar{g}^i(x^*)^{\mathrm{T}} d}{g_0^i(x^*)} \geqslant 0, i \in B^* \\ -\nabla g_0^i(x^*) d + \nabla \bar{g}^i(x^*)^{\mathrm{T}} d \geqslant 0, i \in Z^* \end{array} \right\}.$$

Since $-\lambda \perp -g(x)$

$$\lambda = \left\{ \begin{array}{c} \lambda \big| \lambda_{m_i} = 0, i \in I^* \\ \lambda_{m_i} = \sigma(-g_0^i(x^*), \bar{g}^i(x^*)), \sigma > 0, i \in B^* \\ \lambda_{m_i} \in \text{int}\mathcal{K}^{m_i}, i \in Z^* \end{array} \right\}$$

which further yields that

$$\left[-g(x^*) - \lambda^*\right]_{m_i} = \begin{cases} -g_{m_i}(x^*) \in \text{int}\mathcal{K}^{m_i}, & i \in I^*. \\ ((1-\sigma)(-g_0^i(x^*)), (1+\sigma)(-\bar{g}^i(x^*))), & i \in B^*. \\ \lambda_{m_i} \in \text{int}\,\mathcal{K}^{m_i}, & i \in Z^*. \end{cases}$$

On the other hand, Condition III implies that $\mathcal{C}(x^*)$ is as shown at the bottom of the page, and $\mathcal{C}(x^*)$ is a linear space. Therefore, we have

$$\delta^*\left(\lambda \middle| T_{\mathcal{K}}^2(-g(x^*), -\nabla g(x^*)^{\mathrm{T}} d)\right) = \sum_{i \in B^*} \frac{\lambda_0^i}{-g_0^i(x^*)} \left[\left\| \nabla g_0^i(x^*)^{\mathrm{T}} dx \right\|^2 - \left\| \nabla \bar{g}^i(x^*)^{\mathrm{T}} dx \right\|^2 \right]$$

with $\lambda_{m_i} = (\lambda_0^i, \bar{\lambda}^i)$.

Case I: If $i \in B^*$, we have $\lambda^*{}_{m_i} = (-\sigma g_0^i(x^*), \sigma \bar{g}^i(x^*))$. Then, by Lemma 1 and (29), we obtain

$$\begin{aligned} &\Pi'_{\mathcal{K}^{m_i}}(-g_{m_i}(x^*) - \lambda^*_{m_i}; -J g_{m_i}(x^*) dx - d\lambda_{m_i}) \\ &= \frac{1}{2} \begin{pmatrix} 1 & w_i^{\mathrm{T}} \\ w_i & \dfrac{2}{1+\sigma} I - \dfrac{1-\sigma}{1+\sigma} w_i w_i^{\mathrm{T}} \end{pmatrix} \\ &\quad \times \left(-\nabla g_{m_i}(x^*)^{\mathrm{T}} dx - d\lambda_{m_i}\right) \\ &= A_i(-\nabla g_{m_i}(x^*)^{\mathrm{T}} dx - d\lambda_{m_i}) = -\nabla g_{m_i}(x^*)^{\mathrm{T}} dx \end{aligned} \quad (31)$$

where

$$A_i = \frac{1}{2} \begin{pmatrix} 1 & w_i^{\mathrm{T}} \\ w_i & \dfrac{2}{1+\sigma} I - \dfrac{1-\sigma}{1+\sigma} w_i w_i^{\mathrm{T}} \end{pmatrix}$$

and $w_i = (-\bar{g}^i(x^*))/(\|\bar{g}^i(x^*)\|)$. Now, we need to prove that $dx \in T_C(x^*)$ and

$$-\nabla g_0^i(x^*)^{\mathrm{T}} dx \geqslant \frac{\bar{g}^i(x^*)^{\mathrm{T}} \nabla \bar{g}^i(x^*)^{\mathrm{T}} dx}{\|\bar{g}^i(x^*)\|}. \quad (32)$$

From $-g_0^i(x^*) = \|\bar{g}^i(x^*)\|$, we know

$$\lambda^*_{m_i} = \begin{pmatrix} -\sigma g_0^i(x^*) \\ +\sigma \bar{g}^i(x^*) \end{pmatrix} = -\sigma g_0^i(x^*) \begin{pmatrix} 1 \\ -w_i \end{pmatrix}$$

where $\|w_i\| = 1$ and $w_i = (\bar{g}^i(x^*))/(g_0^i(x^*)) = (-\bar{g}^i(x^*))/(\|\bar{g}^i(x^*)\|)$ for $i \in B^*$. Hence, we achieve

$$\begin{aligned} \lambda^*_{m_i} A_i &= \left(1 - \|w_i\|^2, w_i{}^{\mathrm{T}} - \frac{2}{1+\sigma} w_i{}^{\mathrm{T}} + \frac{1-\sigma}{1+\sigma} w_i{}^{\mathrm{T}} \|w_i\|^2\right) \\ &= (0, 0). \end{aligned} \quad (33)$$

Combining (31) and (33) yields

$$\left\langle \lambda^*_{m_i}, -\nabla g_{m_i}(x^*)^{\mathrm{T}} dx \right\rangle = 0$$

which means that $dx \in \mathcal{C}(x^*)$. Then, it follows from (31) that:

$$\begin{aligned} &A_i(-\nabla g_{m_i}(x^*)^{\mathrm{T}} dx - d\lambda_{m_i}) \\ &= -\nabla g_{m_i}(x^*)^{\mathrm{T}} dx \\ &\Leftrightarrow (A_i - I)(-\nabla g_{m_i}(x^*)^{\mathrm{T}}) dx = A_i d\lambda_{m_i} \\ &\Leftrightarrow (1, -w_i{}^{\mathrm{T}}) \begin{pmatrix} \dfrac{1}{2} & -\dfrac{1}{2}\dfrac{1}{2} w_i{}^{\mathrm{T}} \\ \dfrac{1}{2} w_i & \dfrac{-\sigma}{1+\sigma} I - \dfrac{1}{2}\dfrac{1-\sigma}{1+\sigma} w_i w_i{}^{\mathrm{T}} \end{pmatrix} \\ &\quad \times \begin{pmatrix} -\nabla g_0^i(x^*)^{\mathrm{T}} dx \\ -\nabla \bar{g}^i(x^*) dx \end{pmatrix} \\ &= (1, -w_i{}^{\mathrm{T}}) \begin{pmatrix} \dfrac{1}{2} & \dfrac{1}{2} w_i{}^{\mathrm{T}} \\ \dfrac{1}{2} w_i & \dfrac{1}{1+\sigma} I - \dfrac{1}{2}\dfrac{1-\sigma}{1+\sigma} w_i w_i{}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} d\lambda_0^i \\ d\bar{\lambda}^i \end{pmatrix}. \end{aligned} \quad (34)$$

In summary, we deduce that

$$\left(-1, \frac{1}{2} w_i{}^{\mathrm{T}} + \frac{\sigma}{1+\sigma} w_i{}^{\mathrm{T}} + \frac{1}{2}\frac{1-\sigma}{1+\sigma} w_i{}^{\mathrm{T}}\right) \times \begin{pmatrix} -\nabla g_0^i(x^*)^{\mathrm{T}} dx \\ -\nabla \bar{g}^i(x^*)^{\mathrm{T}} dx \end{pmatrix} = 0$$

which is equivalent to

$$(-1, w_i{}^{\mathrm{T}}) \begin{pmatrix} -\nabla g_0^i(x^*)^{\mathrm{T}} dx \\ -\nabla \bar{g}^i(x^*)^{\mathrm{T}} dx \end{pmatrix} = 0.$$

This indicates that

$$-\nabla g_0^i(x^*)^{\mathrm{T}} dx = \frac{\bar{g}^i(x^*)^{\mathrm{T}} \nabla \bar{g}^i(x^*)^{\mathrm{T}} dx}{\|\bar{g}^i(x^*)\|} \quad (35)$$

and hence, (32) holds.

Case II: Let $i \in Z^*$. From the second equation of (29), we have

$$\Pi'_{\mathcal{K}_{m_i}}(0 - \lambda_{m_i}; -\nabla g_{m_i}(x^*)^{\mathrm{T}} dx - d\lambda_{m_i}) = -\nabla g_{m_i}(x^*)^{\mathrm{T}} dx.$$

Hence, $-\nabla g_{m_i}(x^*)^{\mathrm{T}} dx = 0$.

$$\mathcal{C}(x^*) = \left\{ \begin{array}{c} d \big| \nabla h(x^*)^{\mathrm{T}} d = 0, -\nabla g_{m_i}(x^*)^{\mathrm{T}} d = 0, i \in Z^* \\ -\nabla g_{m_i}(x^*)^{\mathrm{T}} d \in T_{\mathcal{K}}(-g_{m_i}(x^*)), \left\langle \lambda_{m_i}, -\nabla g_{m_i}(x^*)^{\mathrm{T}} d \right\rangle = 0, i \in B^* \end{array} \right\}$$

Case III: Let $i \in I^*$. Again, from the second equation of (29), we have

$$\begin{aligned}&\Pi'_{\mathcal{K}_{m_i}}(-g_{m_i}(x^*), -\nabla g_{m_i}(x^*)^{\mathrm{T}}dx - d\lambda_{m_i})\\&\quad= -\nabla g_{m_i}(x^*)^{\mathrm{T}}dx - d\lambda_{m_i}\\&\quad= -\nabla g_{m_i}(x^*)^{\mathrm{T}}dx\end{aligned}$$

which says $d\lambda_{m_i} = 0$.

From all the above, we conclude that $dx \in \mathcal{C}(x^*)$ implies

$$\begin{cases}\nabla g_{m_i}(x^*)^{\mathrm{T}}dx = 0, i \in Z^*\\ g_0^i(x^*)\nabla g_0^i(x^*)^{\mathrm{T}}dx = \bar{g}^i(x^*)^{\mathrm{T}}\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx, i \in B^*.\end{cases}$$

Applying (28) and (29), we have that

$$J_xL(x^*, \mu^*, \lambda^*)dx + \nabla h(x^*)d\mu + \nabla g(x^*)d\lambda = 0 \tag{36}$$

$$\nabla h(x^*)^{\mathrm{T}}dx = 0 \tag{37}$$

$$-\nabla g(x^*)^{\mathrm{T}}dx - \Pi'_{\mathcal{K}}(-g(x^*) - \lambda^*; -\nabla g(x^*)^{\mathrm{T}}dx - d\lambda) = 0. \tag{38}$$

Using (36) and (37) gives

$$\begin{aligned}0 &= \langle dx, J_xL(x^*, \mu^*, \lambda^*)dx + \nabla h(x^*)d\mu + \nabla g(x^*)d\lambda\rangle\\&= \langle dx, J_xL(x^*, \mu^*, \lambda^*)dx\rangle - \sum_{i\in B^*}\langle -\nabla g_{m_i}(x^*)dx, d\lambda_{m_i}\rangle.\end{aligned}$$

Thus, for $i \in B^*$

$$\begin{aligned}&\left\langle -\nabla g_{m_i}(x^*)^{\mathrm{T}}dx, d\lambda_{m_i}\right\rangle\\&\quad= -\nabla g_0^i(x^*)dxd\lambda_0^i + \langle -\nabla\bar{g}^i(x^*)dx, d\bar{\lambda}^i\rangle\\&\quad= \nabla g_0^i(x^*)^{\mathrm{T}}dx \cdot \frac{\bar{g}^i(x^*)}{\|\bar{g}^i(x^*)\|}d\bar{\lambda}^i - \langle\nabla\bar{g}^i(x^*)dx, d\bar{\lambda}^i\rangle\\&\quad= \frac{\bar{g}^i(x^*)^{\mathrm{T}}\nabla\bar{g}^i(x^*)dx}{\|\bar{g}^i(x^*)\|^2}\bar{g}^i(x^*)^{\mathrm{T}}d\bar{\lambda}^i - \langle\nabla\bar{g}^i(x^*)dx, d\bar{\lambda}^i\rangle\\&\quad= \left\langle(-\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx)^{\mathrm{T}}\left[I - \frac{\bar{g}^i(x^*)\bar{g}^i(x^*)^{\mathrm{T}}}{\|\bar{g}^i(x^*)\|^2}\right], d\bar{\lambda}^i\right\rangle.\end{aligned} \tag{39}$$

On the other hand, from (34), we have (40), as shown at the bottom of the page. From (35), we can deduce that

$$\begin{aligned}&\frac{1}{2}w_i\left(-\nabla g_0^i(x^*)^{\mathrm{T}}dx - w_i{}^{\mathrm{T}}\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx \cdot \frac{1-\sigma}{1+\sigma}\right)\\&\qquad+\frac{\sigma}{1+\sigma}\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx\\&\quad= \frac{1}{2}w_i(-\nabla g_0^i(x^*)^{\mathrm{T}}dx - \frac{1-\sigma}{1+\sigma}w_i{}^{\mathrm{T}}\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx)\\&\qquad+\frac{\sigma}{1+\sigma}\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx\\&\quad= \frac{1}{2}w_i(-\nabla g_0^i(x^*)^{\mathrm{T}}dx + \frac{1-\sigma}{1+\sigma}\nabla g_0^i(x^*)^{\mathrm{T}}dx)\\&\qquad+\frac{\sigma}{1+\sigma}\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx\\&\quad= \frac{\sigma}{1+\sigma}\left(w_i(-\nabla g_0^i(x^*)^{\mathrm{T}})dx + \nabla\bar{g}^i(x^*)^{\mathrm{T}}dx\right)\end{aligned} \tag{41}$$

and

$$\begin{aligned}&\frac{1}{2}w_i\left(d\lambda_0^i - \frac{1-\sigma}{1+\sigma}w_i{}^{\mathrm{T}}d\bar{\lambda}^i\right) + \frac{1}{1+\sigma}d\bar{\lambda}^i\\&\quad= \frac{1}{2}w_i\left(d\lambda_0^i + \frac{1-\sigma}{1+\sigma}d\lambda_0^i\right) + \frac{1}{1+\sigma}d\bar{\lambda}^i\\&\quad= \frac{1}{1+\sigma}w_id\lambda_0^i + \frac{1}{1+\sigma}d\bar{\lambda}^i\\&\quad= \frac{1}{1+\sigma}(w_id\lambda_0^i + d\bar{\lambda}^i).\end{aligned} \tag{42}$$

Therefore, applying (40)–(42) implies that

$$\begin{aligned}&\frac{1}{1+\sigma}(w_id\lambda_0^i + d\bar{\lambda}^i)\\&\qquad= \frac{\sigma}{1+\sigma}(w_i(-\nabla g_0^i(x^*)^{\mathrm{T}})dx + \nabla\bar{g}^i(x^*)^{\mathrm{T}}dx)\end{aligned}$$

which means that

$$w_id\lambda_0^i + d\bar{\lambda}^i = -\sigma(w_i\nabla g_0^i(x^*)^{\mathrm{T}}dx - \nabla\bar{g}^i(x^*)^{\mathrm{T}}dx). \tag{43}$$

Note that

$$w_id\lambda_0^i + d\bar{\lambda}^i = (I - w_iw_i{}^{\mathrm{T}})d\bar{\lambda}^i = \left(I - \frac{\bar{g}^i(x^*)\bar{g}^i(x^*)^{\mathrm{T}}}{\|\bar{g}^i(x^*)\|^2}\right)d\bar{\lambda}^i. \tag{44}$$

Then, it follows from (39), (43), and (44) that:

$$\begin{aligned}&\langle -\nabla g_{m_i}(x^*)dx, d\lambda_{m_i}\rangle\\&= \left\langle -\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx, \left(I - \frac{\bar{g}^i(x^*)\bar{g}^i(x^*)^{\mathrm{T}}}{\|\bar{g}^i(x^*)\|^2}\right)d\bar{\lambda}^i\right\rangle\\&= \sigma\left(\left\langle -\nabla g^i(x^*)^{\mathrm{T}}dx, w_i(-\nabla g_0^i(x^*)^{\mathrm{T}})dx\right\rangle - \|\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx\|^2\right)\\&= \sum_{i\in B^*}\frac{\lambda_0^i}{-g_0^i(x^*)}\left(\left\|\nabla g_0^i(x^*)^{\mathrm{T}}dx\right\|^2 - \|\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx\|^2\right)\\&= \delta^*\left(\lambda\middle|T_{\mathcal{K}}{}^2(-g(x^*); -\nabla g(x^*)^{\mathrm{T}}dx)\right).\end{aligned}$$

$$\begin{pmatrix}\frac{1}{2}\nabla g_0^i(x^*)^{\mathrm{T}}dx + \frac{1}{2}\frac{\bar{g}^i(x^*)^{\mathrm{T}}}{\|\bar{g}^i(x^*)\|}J\bar{g}^i(x^*)dx\\ \frac{1}{2}w_i\left(-\nabla g_0^i(x^*)^{\mathrm{T}}dx - w_i{}^{\mathrm{T}}\nabla\bar{g}^i(x^*)^{\mathrm{T}}dx \cdot \frac{1-\sigma}{1+\sigma}\right) - \frac{\sigma}{1+\sigma}(-\nabla\bar{g}^i(x^*)^{\mathrm{T}})dx\end{pmatrix} = \begin{pmatrix}\frac{1}{2}d\lambda_0^i + \frac{1}{2}w_i{}^{\mathrm{T}}d\bar{\lambda}^i\\ \frac{1}{2}w_i(d\lambda_0^i - \frac{1-\sigma}{1+\sigma}w_i{}^{\mathrm{T}}d\bar{\lambda}^i) + \frac{1}{1+\sigma}d\bar{\lambda}^i\end{pmatrix} \tag{40}$$

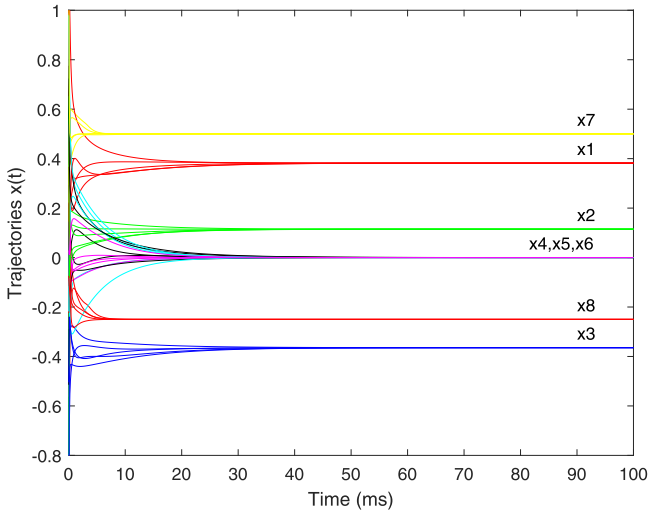

Fig. 2. Convergence of $x(t)$ to the SOCCVI solution in Example 1 using five random initial points, where $\rho = 10^3$.

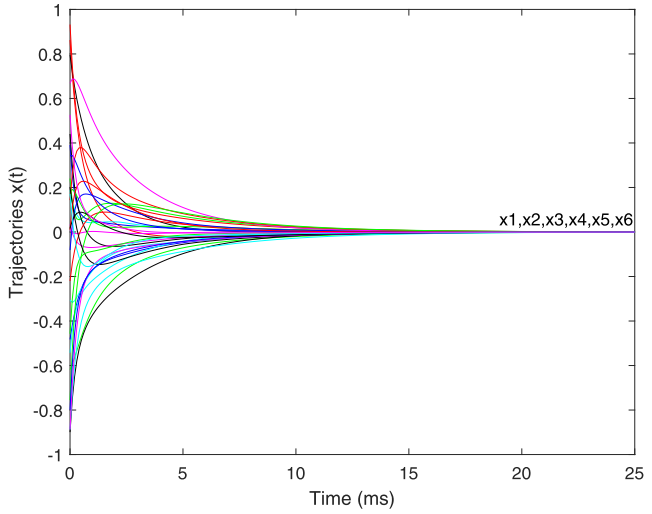

Fig. 3. Convergence of $x(t)$ to the SOCCVI solution in Example 2 using five random initial points, where $\rho = 10^3$.

This together with (30) yields

$$\left\langle J_x L(x^*, \mu^*, \lambda^*)dx, dx \right\rangle - \delta^* \times \left(\lambda \left| T_{\mathcal{K}}{}^2(-g(x^*); -\nabla g(x^*)^{\mathrm{T}} dx)\right.\right) = 0.$$

Now, using the second-order sufficient condition (Condition II), we reach $dx = 0$. Plugging this into (36) leads to

$$\nabla h(x^*)d\mu + \nabla g(x^*)d\lambda = 0.$$

Applying (38) and Condition IV guarantees $d\mu = 0$ and $d\lambda = 0$. Thus, M is nonsingular, and the proof is complete. ■

V. Numerical Experiments

We illustrate the efficiency of the smoothed NR neural network to solve some illustrative SOCCVI problems. We also present a thorough numerical comparison of (20) with other neural networks in the SOCCVI literature.

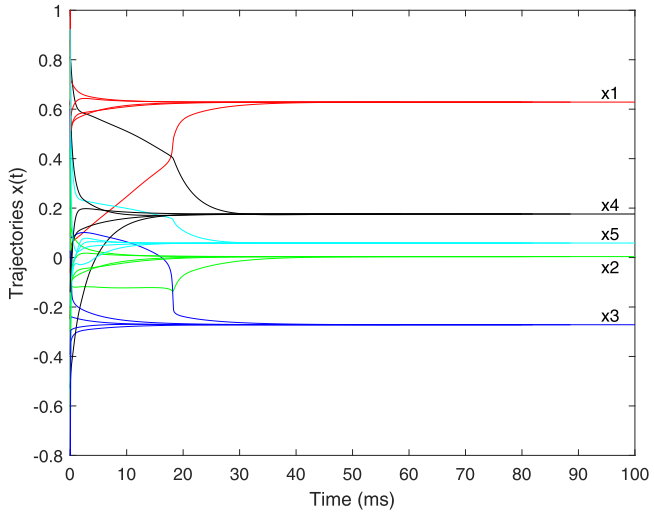

Fig. 4. Convergence of $x(t)$ to the SOCCVI solution in Example 3 using five random initial points, where $\rho = 10^3$.

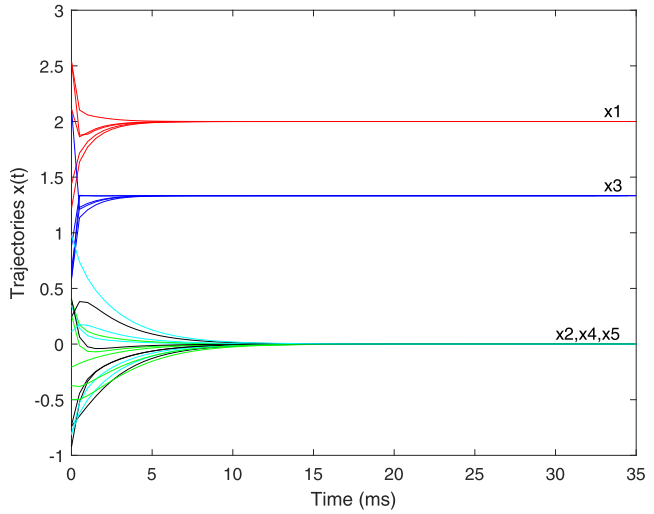

Fig. 5. Convergence of $x(t)$ to the SOCCVI solution in Example 4 using five random initial points, where $\rho = 10^3$.

A. Test Problems and Simulations

We consider six standard test problems for SOCCVI (1). In our simulations, the solver adopted is ode23 s, and the stopping criterion is set at $\|\nabla\Psi(z(t)\| \leq 1\times10^{-6}$. We simulate the neural network (20) for each of the following examples using five random initial points z_0. The trajectories of (20) for the above-mentioned problems are shown in Figs. 2–7. Observe that all the trajectories were successfully able to converge to the SOCCVI solution, which we shall denote by x^*.

Example 1: [33, Example 4.1] Let $F(x)$ be as shown at the bottom of the next page. Here, $x^* = (0.3820, 0.1148, -0.3644, 0.0000, 0.0000, 0.0000, 0.5000, -0.2500)$ is the SOCCVI solution.

Example 2: [35, Example 5.1] Let

$$\left\langle \frac{1}{2}Dx, \text{y-x} \right\rangle \geq 0 \quad \forall y \in C$$

where

$$C = \{x \in \mathbb{R}^n \mid Ax - a = 0, \ \text{Bx-b} \preceq 0\}.$$

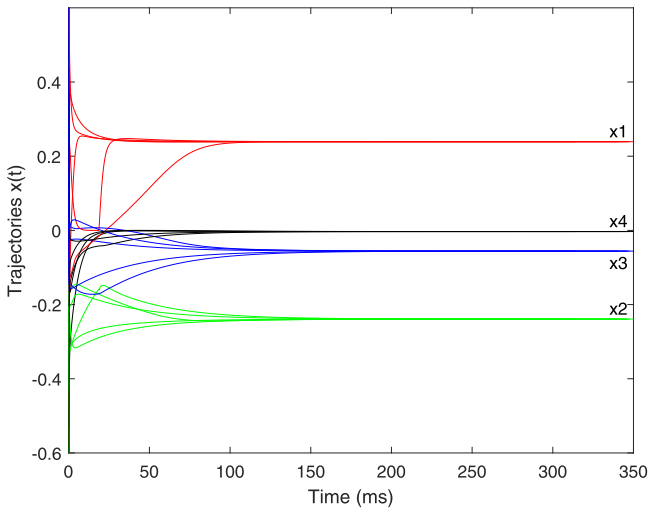

Fig. 6. Convergence of $x(t)$ to the SOCCVI solution in Example 5 using five random initial points, where $\rho = 10^3$.

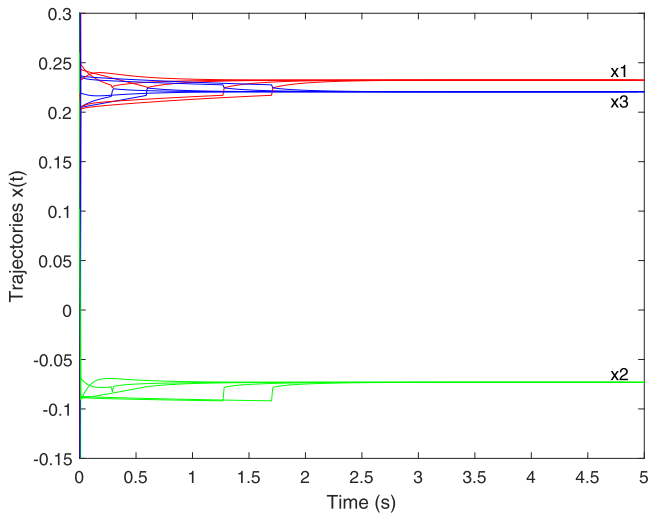

Fig. 7. Convergence of $x(t)$ to the SOCCVI solution in Example 6 using five random initial points, where $\rho = 10^3$.

$A \in \mathbb{R}^{l\times n}$, $B \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{n\times n}$ is symmetric, $d \in \mathbb{R}^n$, and $a \in \mathbb{R}^l$ and $b \in \mathbb{R}^m$, with $l + m \le n$.

As in [35, Example 5.1], we let

$$D = (D_{ij})_{n\times n}, \text{ where } D_{ij} = \begin{cases} 2, & i = j \\ 1, & |i-j| = 1 \\ 0, & \text{otherwise.} \end{cases}$$

$A = \begin{bmatrix} I_{l\times l} & 0_{l\times(n-l)} \end{bmatrix}_{l\times n}$, $B = \begin{bmatrix} 0_{m\times(n-m)} & I_{m\times m} \end{bmatrix}_{m\times n}$, $a = 0_{l\times 1}$, and $b = (e_{m_1}, \ldots, e_{m_p})$, where $e_{m_i} = (1, 0, \ldots, 0)^{\mathrm{T}} \in \mathbb{R}^{m_i}$. We set $l = m = 3$ and $n = 6$ for the simulations, and the SOCCVI has $x^* = (0, 0, 0, 0, 0, 0)$ as its solution.

Example 3: [33, Example 5.3] Let

$$F(x) = \begin{pmatrix} x_3\exp(x_1x_3) + 6(x_1 + x_2) \\ 6(x_1 + x_2) + \dfrac{2(2x_2 - x_3)}{\sqrt{1 + (2x_2 - x_3)^2}} \\ x_1\exp(x_1x_3) - \dfrac{2x_2 - x_3}{\sqrt{1 + (2x_2 - x_3)^2}} \\ x_4 \\ x_5 \end{pmatrix}$$

and

$$C = \{x \in \mathbb{R}^5 \,|\, h(x) = 0, -g(x) \in \mathcal{K}^3 \times \mathcal{K}^2\}$$

with

$$h(x) = -62x_1^3 + 58x_2 + 167x_3^3 - 29x_3 - x_4 - 3x_5 + 11$$

$$g(x) = \begin{pmatrix} -3x_1^3 - 2x_2 + x_3 - 5x_3^3 \\ 5x_1^3 - 4x_2 + 2x_3 - 10x_3^3 \\ -x_3 \\ -x_4 \\ -3x_5 \end{pmatrix}.$$

Here, $x^* = (0.6287, 0.0039, -0.2717, 0.1761, 0.0587)$.

Example 4: Let

$$F(x) = \left[2x_1 - 4, e^{x_2} - 1, 3x_3 - 4, -\sin(x_4), x_5\right]^{\mathrm{T}}$$

and

$$C = \left\{x \in \mathbb{R}^5 \,|\, -g(x) = x \in \mathcal{K}^5\right\}.$$

Here, $x^* = \left(2, 0, 1.3333, 0, 0\right)$.

Example 5: [33, Example 5.5] Consider the VI

$$F(x) = \begin{pmatrix} 4x_1 - \sin x_1 \cos x_2 + 1 \\ -\cos x_1 \sin x_2 + 6x_2 + \frac{9}{5}x_3 + 2 \\ \frac{9}{5}x_2 + 8x_3 + 3 \\ 2x_4 + 1 \end{pmatrix}$$

and

$$C = \left\{x \in \mathbb{R}^4 \,|\, h(x) = 0, -g(x) \in \mathcal{K}^2\right\}$$

$$F(x) = \begin{pmatrix} 2x_1 + x_2 + 1 \\ x_1 + 6x_2 - x_3 - 2 \\ -x_2 + 3x_3 - \dfrac{6}{5}x_4 + 3 \\ -\dfrac{6}{5}x_3 + 2x_4 + \dfrac{1}{2}\sin x_4 \cos x_5 \sin x_6 + 6 \\ \dfrac{1}{2}\cos x_4 \sin x_5 \sin x_6 + 2x_5 - \dfrac{5}{2} \\ -\dfrac{1}{2}\cos x_4 \cos x_5 \cos x_6 + 2x_6 + \dfrac{1}{4}\cos x_6 \sin x_7 \cos x_8 + 1 \\ \dfrac{1}{4}\sin x_6 \cos x_7 \cos x_8 + 4x_7 - 2 \\ -\dfrac{1}{4}\sin x_6 \sin x_7 \sin x_8 + 2x_8 + \dfrac{1}{2} \end{pmatrix}$$

$$C = \{x \in \mathbb{R}^4 \,|\, -g(x) = x \in \mathcal{K}^3 \times \mathcal{K}^3 \times \mathcal{K}^2\}.$$

with

$$h(x)=\begin{pmatrix} x_1^2-\dfrac{1}{10}x_2x_3+x_3 \\ x_3^2+x_4 \end{pmatrix} \text{ and } g(x)=\begin{pmatrix} -x_1 \\ -x_2 \end{pmatrix}.$$

Here, $x^*=(0.2391,-0.2391,-0.0558,-0.0031)$.

Example 6: Consider the CSOCP [25]

$$\begin{aligned} \min\ & \exp(x_1-x_3)+3(2x_1-x_2)^4+\sqrt{1+(3x_2+5x_3)^2} \\ \text{s.t.}\ & -g(x)=\begin{pmatrix} 4x_1+6x_2+3x_3-1 \\ -x_1+7x_2-5x_3+2 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}\in \mathcal{K}^2\times\mathcal{K}^3. \end{aligned}$$

For this CSOCP, $x^*=(0.2324,-0.07309,0.2206)$ is the approximate solution. This problem can be recast as an SOCCVI problem, as discussed in Section I.

Next, we consider a practical example.

Example 7: We consider the grasping-force optimization problem for multifingered robotic hands [27], [41], which involves determining the minimum force that each finger must exert on an object so as to maintain the finger's grasp. In particular, we consider the problem in [41] involving a three-fingered robotic hand with fingers positioned at $(0,1,0)$, $(1,0.5,0)$, and $(0,-1,0)$. The optimization problem is given by

$$\begin{aligned} \min\ & \frac{1}{2}y^{\mathrm{T}}y \\ \text{s.t.}\ & Gy=-f_{\text{ext}}\quad \sqrt{y_{i1}^2+y_{i2}^2}\le \mu_i y_{i3},\quad (i=1,\ldots,m) \end{aligned}$$

where $y=(y_{11},y_{12},y_{13},\ldots,y_{m3})\in\mathbb{R}^{3m}$, $G\in\mathbb{R}^{6\times 3m}$ is the grasping transformation matrix, f_{ext} is the (time-varying) external wrench, and μ_i is the friction coefficient at finger i. As in [41], we let $\mu_i=\mu=0.6$ for all i

$$G=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & -0.5 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0.5 & 1 & 0 & 0 \end{pmatrix}$$

and $f_{\text{ext}}=(0,f_c\sin\theta(t),-Mg+f_c\cos\theta(t),0,0,0)^{\mathrm{T}}$, where $g=9.8$ m/s^2, M is the mass of the object (assumed to be 0.1 kg), $f_c=Mv^2/r$, and $\theta(t)=vt/r$. The hand moves along a circular path of radius $r=0.5$ m and constant velocity $v=0.4\pi$ m/s.

In order to use our neural network, we recast the above problem as an SOCCVI. First, we let $(x_{i1},x_{i2},x_{i3})=(\mu f_{i3},f_{i1},f_{i2})$. By this transformation, it can be shown that the problem corresponds to the SOCCVI with F, g, and h given as follows:

$$\begin{aligned} F(x)&=\operatorname{diag}(1/\mu^2,1,1,1/\mu^2,1,1,1/\mu^2,1,1)\,x \\ -g(x)&=x\in\mathcal{K}^3\times\mathcal{K}^3\times\mathcal{K}^3 \\ h(x)&=\bar{G}x+f_{ext} \end{aligned}$$

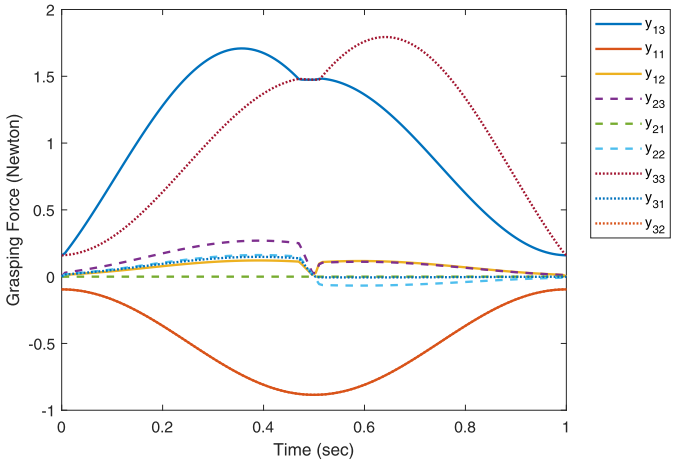

Fig. 8. Time-varying optimal grasping force for the three-fingered robotic hand.

where

$$\bar{G}=\begin{pmatrix} 0 & 0 & 1 & -1/\mu & 0 & 0 & 0 & 1 & 0 \\ -1/\mu & 0 & 0 & 0 & 0 & -1 & 1/\mu & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & -0.5 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0.5/\mu & 0 & -1 & 0 & 1 & 0 \end{pmatrix}.$$

We note that external wrench f_{ext} applied varies over time. In Fig. 8, we show the optimal force required as time varies from 0 to 1 s.

B. Comparisons With Some Neural Networks

We now make some comparisons between the four existing neural networks in the SOCCVI literature and our neural network given by (20). Two neural networks for SOCCVI were proposed in [33]. The first of which used the smoothed FB function $\phi_{\text{FB}}^{\varepsilon}$ given by (11), where ε is a smoothing parameter that ultimately decreases to zero. The other neural network in [33] is based on a projection formulation of the SOCCVI after an equivalent transformation of the KKT conditions. Recently, two more neural networks are proposed in [34] based on the discrete generalizations of the FB and NR functions given by (12) and (13), respectively. These two neural networks will be denoted by "DFB-NN" and "DNR-NN."

First, we summarize here our findings according to several experiments on the performance of the four aforementioned neural networks that were already studied in [33] and [34]. Among all these four neural networks, we found from our experiments that the smoothed FB-neural network may be considered as the one with the best numerical performance in the sense that it is more capable of solving SOCCVI problems compared with the other three. We have also verified through our numerical experiments that this neural network is also less sensitive to changes in the initial condition. Meanwhile, the projection-based neural network also has good numerical properties. Whenever both smoothed FB and projection-based neural networks converge to the SOCCVI solution, our numerical simulations reveal that a faster convergence time is usually obtained for the latter neural network. However, the projection-based neural network more often cannot solve some problems

TABLE I

SUMMARY OF SUCCESSFUL AND UNSUCCESSFUL SIMULATION RESULTS FOR THE FIVE NEURAL NETWORKS

	Reference	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6
Smoothed NR	Proposed NN	✓	✓	✓	✓	✓	✓
Smoothed FB	[34]	✓	✓	✓	✓	✓	✓
Projection-Based	[34]	✓	✓	✗	✗	✗	✓
DFB-NN	[35]	✗	✓	✗	✗	✓	✗
DNR-NN	[35]	✗	✓	✗	✗	✗	✗

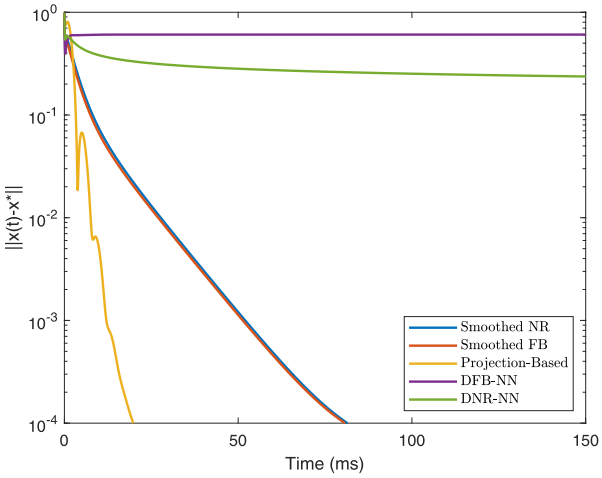

Fig. 9. Comparison of decay rates of $\|x(t)-x^*\|$ for the five neural networks for Example 1.

compared with the smoothed FB-neural network. Finally, the other two neural networks DFB-NN and DNR-NN are very sensitive to initial conditions.

Now, we compare our neural network with the four neural networks mentioned earlier. We have found that, in general, our proposed smoothed NR neural network (20) has better stability and convergence behavior. We shall illustrate this using the test problems given in Section V-A. In Table I, we summarize the results of simulating our neural network and the aforementioned four neural networks in the literature. The mark "✓" indicates that the neural network can solve the SOCCVI, while "✗" indicates otherwise. To compare the rates of convergence of the neural networks, we simulate the trajectories $z(t) = (x(t), \mu(t), \lambda(t))$ and evaluate the error term $\|x(t) - x^*\|$, where x^* is the SOCCVI solution (see Figs. 9–14). We summarize our findings as follows.

1) From Table I, observe that only our proposed neural network and the smoothed FB neural network were able to solve all the SOCCVI problems. The projection-based neural network was able to solve half of the problems, while the other two discrete-based neural networks have the worst success rate.
2) Projection-based neural network has a very fast convergence rate whenever it approaches the SOCCVI solution, as can be seen in Example 1 and Example 6 (see Figs. 9 and 14, respectively). However, its trajectories oscillate for some examples (namely, Examples 3 and 5) and, therefore, fails to converge to the SOCCVI solution.
3) Despite the fast convergence rate of the projection-based neural network, our smoothed NR neural network can still outperform this projection network, as shown in Fig. 10. Note as well that while DFB-NN and DNR-NN can both solve Example 2, the convergence is extremely slow.
4) The error plots shown in Figs. 9–14 reveal that the smoothed NR and smoothed FB neural networks have almost the same convergence rates. Both these neural networks are insensitive to changes in the initial condition. Hence, this suggests that these two neural networks are more applicable to be used in designing neural networks for SOCCVI.
5) Our numerical experiments suggest that the smoothed NR neural network that we proposed is less sensitive to initial conditions compared with smoothed FB network. For instance, from Example 1, it can be easily verified that the latter neural network cannot converge when the initial condition is $z_0 = (0, \ldots, 0)^{\mathrm{T}}$ or $z_0 = (1, \ldots, 1)^{\mathrm{T}}$.

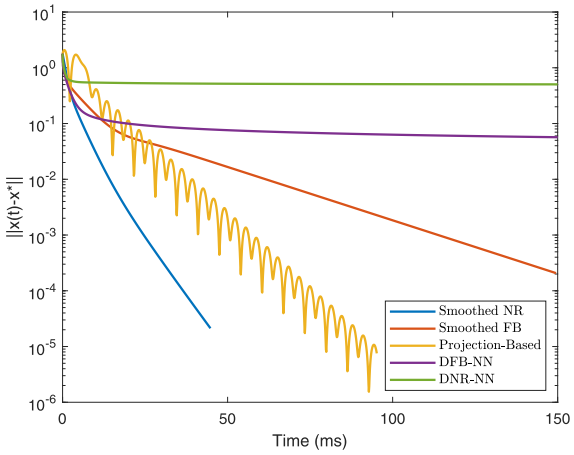

Fig. 10. Comparison of decay rates of $\|x(t)-x^*\|$ for the five neural networks for Example 2.

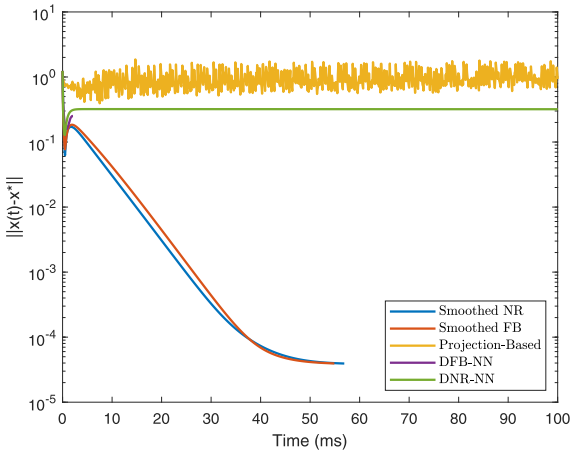

Fig. 11. Comparison of decay rates of $\|x(t)-x^*\|$ for the five neural networks for Example 3.

From the above-mentioned observations, we conclude that our proposed neural network and the smoothed FB neural network have the best performance in solving the SOCCVI problems. However, as noted earlier, the smoothed FB neural network is more sensitive to initial conditions, that is, we may obtain divergence for some initial conditions for smoothed FB network more often than we may encounter in our smoothed NR neural network. In addition, we note that another advantage of our neural network over the smoothed

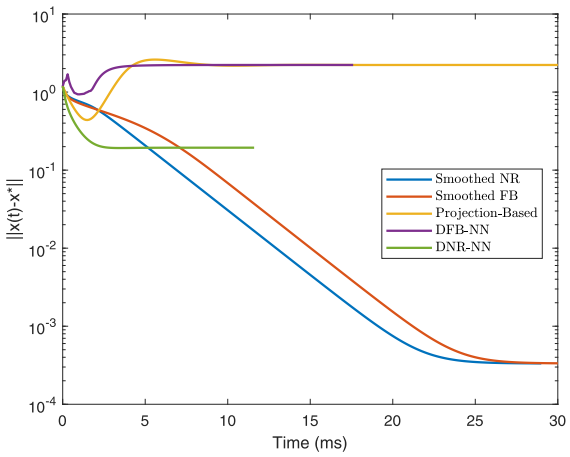

Fig. 12. Comparison of decay rates of $\|x(t)-x^*\|$ for the five neural networks for Example 4.

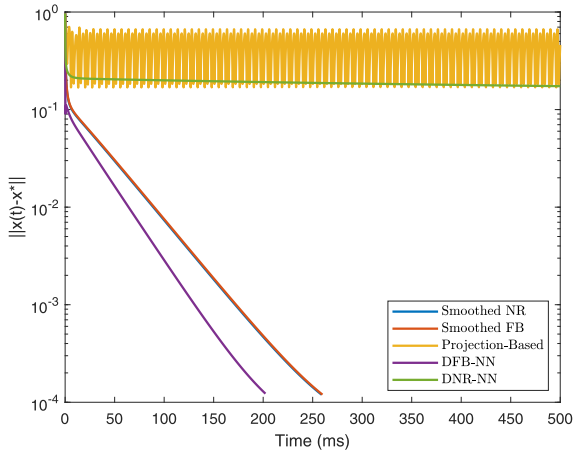

Fig. 13. Comparison of decay rates of $\|x(t)-x^*\|$ for the five neural networks for Example 5.

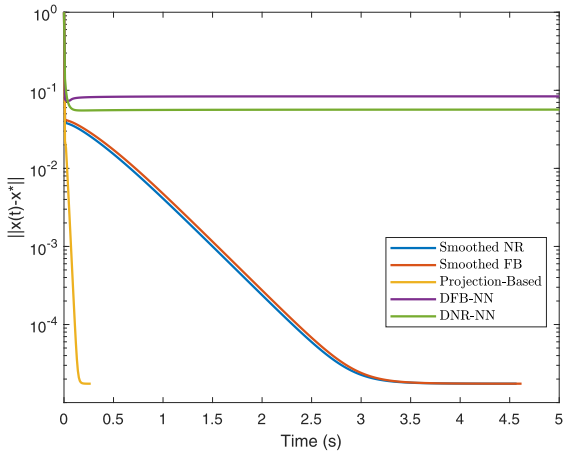

Fig. 14. Comparison of decay rates of $\|x(t)-x^*\|$ for the five neural networks for Example 6.

FB neural network is that the latter costs more expensive numerical computations. This is primarily due to the expensive calculations involve in computing derivatives for the smoothed FB function given by (11). To see this, recall from [33, Lemma 3.1] that for any $\varepsilon \neq 0$

$$\nabla_\varepsilon \phi^\varepsilon_{\mathrm{FB}}(a,b) = e^{\mathrm{T}} L_z^{-1} L_{\varepsilon e}, \quad \nabla_a \phi^\varepsilon_{\mathrm{FB}}(a,b) = L_z^{-1} L_a - I$$
$$\nabla_b \phi^\varepsilon_{\mathrm{FB}}(a,b) = L_z^{-1} L_b - I$$

where $z = (a^2 + b^2 + \varepsilon^2\ e)^{1/2}$ and $L_a = \begin{bmatrix} a_1 & a_2^{\mathrm{T}} \\ a_2 & a_1 I_{n-1} \end{bmatrix}$ for $a = (a_1, a_2)^{\mathrm{T}} \in \mathbb{R} \times \mathbb{R}^{n-1}$. Notice that all the above-mentioned formulas require the calculation of inverse matrices. It can also be verified that there are more calculations required when implementing the smoothed-FB neural network. Finally, we say a few words regarding the complexities of the above-mentioned five neural networks. It is easy to see that the architecture of the smoothed-FB neural network is similar to our neural network (see [33]), but our neural network has better convergence properties. On the other hand, DFB-NN and DNR-NN have slightly lower complexity (see [34]), while the projection-based network has the least complexity, but, as mentioned earlier, these neural models have several shortcomings.

VI. Conclusion

In this article, we target the SOCCVI by using the neural network based on the smoothing metric projector. Unlike previous studies on neural networks for SOCCVI, we herein provide second-order sufficient conditions for the invertibility of the Jacobian of the smoothed KKT system constructed via the smoothed NR function under constraint degeneracy. Some numerical simulations are conducted, which show the efficiency of the smoothed NR neural network. We also discovered that our neural network is more preferable to be employed in solving SOCCVI problems compared with other neural models in the literature. In particular, our neural network has an edge in terms of success rate, average convergence time, and sensitivity to initial conditions.

Acknowledgment

The authors would like to thank three anonymous reviewers for their valuable suggestions and comments that have significantly improved the quality of this article.

References

[1] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Math. Program.*, vol. 95, no. 1, pp. 3–51, Jan. 2003.
[2] J. F. Bonnans and H. Ramírez C., "Perturbation analysis of second-order cone programming problems," *Math. Program.*, vol. 104, nos. 2–3, pp. 205–227, Nov. 2005.
[3] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*. New York, NY, USA: Spring-Verlag, 2000.
[4] Y.-L. Chang, J.-S. Chen, and C.-Y. Yang, "Symmetrization of generalized natural residual function for NCP," *Oper. Res. Lett.*, vol. 43, no. 4, pp. 354–358, Jul. 2015.
[5] J.-S. Chen, "The semismooth-related properties of a merit function and a descent method for the nonlinear complementarity problem," *J. Global Optim.*, vol. 36, no. 4, pp. 565–580, Oct. 2006.
[6] J.-S. Chen, *SOC Functions and Their Applications*. Singapore: Springer, 2019.
[7] J.-S. Chen, H.-T. Gao, and S.-H. Pan, "An *R*-linearly convergent derivative-free algorithm for nonlinear complementarity problems based on the generalized Fischer–Burmeister merit function," *J. Comput. Appl. Math.*, vol. 232, no. 2, pp. 455–471, 2009.
[8] J.-S. Chen, C.-H. Ko, and S. Pan, "A neural network based on the generalized Fischer–Burmeister function for nonlinear complementarity problems," *Inf. Sci.*, vol. 180, no. 5, pp. 697–711, Mar. 2010.
[9] J.-S. Chen and S. Pan, "A family of NCP functions and a descent method for the nonlinear complementarity problem," *Comput. Optim. Appl.*, vol. 40, no. 3, pp. 389–404, Jul. 2008.

[10] J.-S. Chen and S.-H. Pan, "A survey on SOC complementarity functions and solution methods for SOCPs and SOCCPs," *Pacific J. Optim.*, vol. 8, no. 1, pp. 33–74, 2012.
[11] J.-S. Chen and P. Tseng, "An unconstrained smooth minimization reformulation of the second-order cone complementarity problem," *Math. Program.*, vol. 104, nos. 2–3, pp. 293–327, Nov. 2005.
[12] X. Chen, L. Qi, and D. Sun, "Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities," *Math. Comput.*, vol. 67, no. 222, pp. 519–540, 1998.
[13] C. Dang, Y. Leung, X.-B. Gao, and K.-Z. Chen, "Neural networks for nonlinear and mixed complementarity problems and their applications," *Neural Netw.*, vol. 17, no. 2, pp. 271–283, Mar. 2004.
[14] S. Effati, A. Ghomashi, and A. R. Nazemi, "Application of projection neural network in solving convex programming problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1103–1114, 2007.
[15] S. Effati and A. R. Nazemi, "Neural network models and its application for solving linear and quadratic programming problems," *Appl. Math. Comput.*, vol. 172, no. 1, pp. 305–331, Jan. 2006.
[16] F. Facchinei and J. Pang, *Finite-dimensional Variational Inequalities and Complementarity Problems*. New York, NY, USA: Springer, 2003.
[17] M. Fukushima, Z.-Q. Luo, and P. Tseng, "Smoothing functions for second-order-cone complementarity problems," *SIAM J. Optim.*, vol. 12, no. 2, pp. 436–460, Jan. 2002.
[18] Q. Han, L.-Z. Liao, H. Qi, and L. Qi, "Stability analysis of gradient-based neural networks for optimization problems," *J. Global Optim.*, vol. 19, pp. 363–381, Apr. 2001.
[19] P. T. Harker and J.-S. Pang, "Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications," *Math. Program.*, vol. 48, nos. 1–3, pp. 161–220, Mar. 1990.
[20] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, Jul. 1985.
[21] X. Hu and J. Wang, "Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1487–1499, Nov. 2006.
[22] X. Hu and J. Wang, "A recurrent neural network for solving a class of general variational inequalities," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 3, pp. 528–539, Jun. 2007.
[23] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. 35, no. 5, pp. 554–562, May 1988.
[24] C.-H. Ko, J.-S. Chen, and C.-Y. Yang, "Recurrent neural networks for solving second-order cone programs," *Neurocomputing*, vol. 74, no. 17, pp. 3646–3653, Oct. 2011.
[25] C. Kanzow, I. Ferenczi, and M. Fukushima, "On the local convergence of semismooth Newton methods for linear and nonlinear second-order cone programs without strict complementarity," *SIAM J. Optim.*, vol. 20, no. 1, pp. 297–320, Jan. 2009.
[26] L.-Z. Liao, H. Qi, and L. Qi, "Solving nonlinear complementarity problems with neural networks: A reformulation method approach," *J. Comput. Appl. Math.*, vol. 131, nos. 1–2, pp. 343–359, Jun. 2001.
[27] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, nos. 1–3, pp. 193–228, Nov. 1998.
[28] X. Miao, J.-S. Chen, and C.-H. Ko, "A smoothed NR neural network for solving nonlinear convex programs with second-order cone constraints," *Inf. Sci.*, vol. 268, pp. 255–270, Jun. 2014.
[29] X. Miao, J.-S. Chen, and C.-H. Ko, "A neural network based on the generalized FB function for nonlinear convex programs with second-order cone constraints," *Neurocomputing*, vol. 203, pp. 62–72, Aug. 2016.
[30] R. K. Miller and A. N. Michel, *Ordinary Differential Equations*. New York, NY, USA: Academic, 1982.
[31] A. Nazemi and A. Sabeghi, "A novel gradient-based neural network for solving convex second-order cone constrained variational inequality problems," *J. Comput. Appl. Math.*, vol. 347, pp. 343–356, Feb. 2019.
[32] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. New York, NY, USA: Springer-Verlag, 1998.
[33] J. Sun, J.-S. Chen, and C.-H. Ko, "Neural networks for solving second-order cone constrained variational inequality problem," *Comput. Optim. Appl.*, vol. 51, no. 2, pp. 623–648, Mar. 2012.
[34] J. Sun, X.-R. Wu, B. Saheya, J.-S. Chen, and C.-H. Ko, "Neural network for solving SOCQP and SOCCVI based on two discrete-type classes of SOC complementarity functions," *Math. Problems Eng.*, vol. 2019, pp. 1–18, Feb. 2019, Art. no. 4545064.
[35] J. Sun and L. Zhang, "A globally convergent method based on Fischer–Burmeister operators for solving second-order cone constrained variational inequality problems," *Comput. Math. Appl.*, vol. 58, no. 10, pp. 1936–1946, Nov. 2009.
[36] D. Tank and J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. 33, no. 5, pp. 533–541, May 1986.
[37] S. J. Wright, "An infeasible-interior-point algorithm for linear complementarity problems," *Math. Program.*, vol. 67, nos. 1–3, pp. 29–51, Oct. 1994.
[38] Y. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 4, pp. 447–458, Apr. 2002.
[39] Y. Xia and J. Wang, "A general projection neural network for solving monotone variational inequalities and related optimization problems," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 318–328, Mar. 2004.
[40] Y. Xia and J. Wang, "A recurrent neural network for solving nonlinear convex programs subject to linear constraints," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 379–386, Mar. 2005.
[41] Y. Xia, J. Wang, and L.-M. Fok, "Grasping-force optimization for multifingered robotic hands using a recurrent neural network," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 549–554, Jun. 2004.
[42] M. Yashtini and A. Malek, "Solving complementarity and variational inequalities problems using neural networks," *Appl. Math. Comput.*, vol. 190, no. 1, pp. 216–230, Jul. 2007.
[43] S. H. Zak, V. Upatising, and S. Hui, "Solving linear programming problems with neural networks: A comparative study," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 94–104, Jan. 1995.
[44] J. V. Outrata and D. F. Sun, "On the coderivative of the projection operator onto the second order cone," *Set-Valued Anal.*, vol. 16, pp. 999–1014, Jun. 2008.

Juhe Sun received the Ph.D. degree in mathematics from the Dalian University of Technology, Dalian, China, in 2009.

She is currently an Associate Professor with the School of Science, Shenyang Aerospace University, Shenyang, China. She has published more than ten journal articles and has been approved by two National Natural Science Foundations. Her research area is mainly in optimization theory and algorithms.

Weichen Fu received the master's degree in mathematics from Shenyang Aerospace University, Shenyang, China, in 2020. She is currently pursuing the Ph.D. degree with the Harbin Institute of Technology, Harbin, China.

Her research area is mainly in applied power systems.

Jan Harold Alcantara received the Ph.D. degree in mathematics from National Taiwan Normal University (NTNU), Taipei, Taiwan, in 2020.

He is also an incoming Post-Doctoral Researcher with NTNU starting from August 2020. His primary research is on solving complementarity problems in optimization using dynamical systems approaches via complementarity functions and smoothing techniques; other research interests include sparse optimization and second-order cone programming.

Jein-Shan Chen is currently a Professor with the Mathematics Department, National Taiwan Normal University, Taipei, Taiwan. His publications include around 125 journal articles and one monograph titled SOC Functions and Their Applications, Springer Optimization and Its Applications 143 (Springer, 2019). His research area is mainly in continuous optimization with side interests in nonsmooth analysis and operations research.