New Noise-Tolerant Neural Algorithms for Future Dynamic Nonlinear Optimization With Estimation on Hessian Matrix Inversion

Lin Wei, Long Jin, *Member, IEEE*, Chenguang Yang, *Senior Member, IEEE*, Ke Chen, and Weibing Li

Abstract—Nonlinear optimization problems with dynamical parameters are widely arising in many practical scientific and engineering applications, and various computational models are presented for solving them under the hypothesis of short-time invariance. To eliminate the large lagging error in the solution of the inherently dynamic nonlinear optimization problem, the only way is to estimate the future unknown information by using the present and previous data during the solving process, which is termed the future dynamic nonlinear optimization (FDNO) problem. In this paper, to suppress noises and improve the accuracy in solving FDNO problems, a novel noise-tolerant neural (NTN) algorithm based on zeroing neural dynamics is proposed and investigated. In addition, for reducing algorithm complexity, the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is employed to eliminate the intensively computational burden for matrix inversion, termed NTN-BFGS algorithm. Moreover, theoretical analyses are conducted, which show that the proposed algorithms are able to globally converge to a tiny error bound with or without the pollution of noises. Finally, numerical experiments are conducted to validate the superiority of the proposed NTN and NTN-BFGS algorithms for the online solution of FDNO problems.

Index Terms—Noise-tolerant neural algorithm, neural dynamics, quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS), future dynamic nonlinear optimization (FDNO), robotics.

I. INTRODUCTION

TO date, due to the important role that the nonlinear optimization problem plays in various areas [1]–[11], many numerical methods and neural dynamics have been developed and extended to solve it, among which, gradient related methods and Newton-Raphson iteration (NRI) and their modifications are commonly used [2]. For example, a class of nonlinear conjugate gradient methods aiming at solving optimization problems are summarized in [12], which are of global convergence properties. More recently, a three-term conjugate gradient algorithm providing descent searching directions is investigated in [13]. It is worth pointing out that a large number of practical problems are dynamic in nature, of which the parameters involved are varying with time, thereby leading to a time-dependent theoretical solution. When solved by these traditional algorithms, a dynamic optimization problem is assumed to be time-invariant during the computational interval and thus the generated solution is directly employed to the problem at the next time instant. This is mainly due to the fact that, without leveraging the velocity compensation for the dynamic parameters, a traditional model is not able to track the time-dependent theoretical solutions in a predictive manner [14]. Therefore, for a time-dependent problem aided with a traditional model, large lagging error is unavoidable.

Neural networks and the related neural dynamics methods have shown superior properties in parallel distribution and high-speed computing with extensive applications in neurophysiology, chemical equilibrium and robotics [2], [15]–[26]. For instance, Liu and Tong present an adaptive neural network based on optimal control for a class of nonlinear discrete-time systems in [17], which achieves optimal control performance with system stability guaranteed. Continuous-time zeroing neural dynamics is reported to be able to track the time-dependent solution of dynamic problems in an error-free manner [20]. A discrete-time numerical algorithm based on zeroing neural dynamics is presented in [25], [27], which is able to solve time-varying nonlinear optimization (termed future dynamic nonlinear optimization (FDNO) problem) accurately without perturbed by noises. However, in spite of the fact that noises and perturbations are widely existing in the online solution process, existing methods for solving the FDNO problem in the presence of noises are considerably rare. Therefore, it is of crucial importance to find a new computational method to handle noises and perturbations with high accuracy achieved for the FDNO problem.

Considering that the continuous-time model can not be applied to digital hardware directly, computational method depicted in discrete form is desirable. To this end, based on zeroing neural dynamics, a discrete-time noise-tolerant neural (NTN) algorithm is constructed in this paper to solve the FDNO problem in the presence of noises and perturbations.

L. Wei and L. Jin are with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China, and also with Key Laboratory of Industrial Internet of Things & Networked Control, Ministry of Education, Chongqing 400000, China (e-mail: jinlongsysu@foxmail.com; longjinlongjin@ieee.org).

C. Yang is with Bristol Robotics Laboratory, University of the West of England, Bristol, BS16 1QY, UK (e-mail: cyang@ieee.org).

K. Chen is with the School of Electronics and Information, South China University of Technology, Guangzhou 510640, China (e-mail: chenk@scut.edu.cn).

W. Li is with the Chow Yuk Ho Technology Centre for Innovative Medicine, the Chinese University of Hong Kong, Hong Kong (email: weibingli@cuhk.edu.hk).

Given that the Hessian matrix inversion is involved in the NT-N algorithm, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [28], [29] is leveraged to approximate the inverse of Hessian matrix, which is especially helpful for the situation that direct computing of Hessian matrix inverse is expensive or difficult to conduct. The content of this paper is organized as follows. In Section II, the FDNO problem is formulated and the NTN and NTN-BFGS algorithms are proposed to handle such a future problem. Moreover, for comparison, the existing solutions are presented as well. Then, Section III provides theoretical analyses to illustrate the global convergence of the proposed NTN and NTN-BFGS algorithms with or without noises. Moreover, numerical experiments and applications to the robot manipulator are presented in Section IV to validate the superiority of the proposed NTN and NTN-BFGS algorithms, as compared with other existing models. Finally, conclusions are drawn in Section VI. In the end of this introductory section, main contributions of this paper are summarized as follows.

1) This is the first work for solving nonlinear optimizations with dynamic parameters and noise suppressed, of which an intrinsic requirement is that the solution should be calculated before its corresponding mathematical formulation appeared. In this sense, this is quite different from the conventionally investigated static optimization, and thus termed future dynamic nonlinear optimization (FDNO) problem.

2) Two neural algorithms, termed NTN and NTN-BFGS, are proposed to solve the FDNO problems in the presence of noises based on neural dynamics approach, of which the latter one eliminates the intensively computational burden for matrix inversion.

3) Control techniques are leveraged to conduct the theoretical analyses, which reveal that the residual errors of the two proposed neural algorithms are able to converge to a tiny value near zero globally with or without noises.

II. PROBLEM FORMULATION AND SOLUTIONS

This section presents the framework and formulations of the FDNO problem with two newly proposed neural algorithms. For comparison, existing solutions are provided as well.

A. Problem Formulation

It is required in the digital hardware implementation that a problem should be depicted in discrete form. Therefore, it is desirable to formulate a problem in discrete manner. Let t_s and t_f denote the start and the final time instant of the solving process, respectively. An FDNO problem, for which the calculation should be conducted during the time internal $[t_k, t_{k+1}) \in [t_s, t_f]$, is expressed as

$$\min_{\mathbf{y}(t_{k+1}) \in \mathbb{R}^m} \Phi\big(\mathbf{y}(t_{k+1}), t_{k+1}\big) \in \mathbb{R}, \qquad (1)$$

where $t = k\delta$ with updating index $k = 0, 1, 2, \cdots$, which is abbreviated as t_k; $\delta > 0$ represents the time sampling gap; $\Phi\big(\mathbf{y}(t_{k+1}), t_{k+1}\big)$ is discretized from the smoothly time-varying signal $\Phi\big(\mathbf{y}(t), t\big)$, for which the following assumptions are made: $\Phi(\cdot, \cdot)$ is a time-varying nonlinear function and twice differentiable and lower bounded.

This work is dedicated to finding the future solution $\mathbf{y}(t_{k+1}) \in \mathbb{R}^m$ during the computational interval $[t_k, t_{k+1})$ that makes function (1) achieve its minimum value at time instant t_{k+1}. Note that, during the present computational interval $[t_k, t_{k+1})$, $\Phi(\mathbf{y}(t_{k+1}), t_{k+1})$ and its derivatives are not available. In this sense, only the present and/or previous data (e.g., $\mathbf{y}(t_k)$) rather than the unknown data (e.g., $\mathbf{y}(t_{k+1})$) can be leveraged to compute $\mathbf{y}(t_{k+1})$.

B. Continuous-Time NTN Model

The continuous-time FDNO problem is defined as

$$\min_{\mathbf{y}(t) \in \mathbb{R}^m} \Phi\big(\mathbf{y}(t), t\big) \in \mathbb{R}, t \in [t_s, t_f], \qquad (2)$$

of which the gradient is

$$\mathbf{q}(\mathbf{y}(t), t) = \frac{\partial \Phi(\mathbf{y}(t), t)}{\partial \mathbf{y}(t)} \in \mathbb{R}^m. \qquad (3)$$

The 2-norm of $\mathbf{q}(\mathbf{y}(t), t)$ of an algorithm is a measure of the geometric distance between the current solution $\mathbf{y}(t)$ and the zero of $\mathbf{q}(\mathbf{y}(t), t)$. An intuitive approach to obtain the desired path $\mathbf{y}^*(t)$ on which $\mathbf{q}(\mathbf{y}(t), t) = 0$ is to exploit the derivative method. Therefore, to obtain the online solution of FDNO (2), the derivative of $\mathbf{q}(\mathbf{y}(t), t)$ with regard to time t should be 0 for each time instant $t \in [t_s, t_f]$, that is,

$$\begin{aligned} \frac{d\mathbf{q}(\mathbf{y}(t), t)}{dt} &= \frac{\partial \mathbf{q}(\mathbf{y}(t), t)}{\partial t} + \frac{\partial \mathbf{q}(\mathbf{y}(t), t)}{\partial \mathbf{y}(t)} \frac{d\mathbf{y}(t)}{dt} \\ &= \dot{\mathbf{q}}_t(\mathbf{y}(t), t) + H(\mathbf{y}(t), t) \frac{d\mathbf{y}(t)}{dt}, \qquad (4) \end{aligned}$$

where $H(\mathbf{y}(t), t) \in \mathbb{R}^{m \times m}$ represents Hessian matrix. In detail, $\dot{\mathbf{q}}_t(\mathbf{y}(t), t)$ is the derivative of $\mathbf{q}(\mathbf{y}(t), t)$ with respect to time t and can be defined as

$$\dot{\mathbf{q}}_t(\mathbf{y}(t), t) = \frac{\partial \mathbf{q}(\mathbf{y}(t), t)}{\partial t} = \frac{\partial^2 \Phi(\mathbf{y}(t), t)}{\partial \mathbf{y}(t) \partial t} \in \mathbb{R}^m.$$

For performance evaluations in this paper, how well each model solves the FDNO problem is observed through the following error-function

$$\boldsymbol{\xi}(t) = [\xi_1(t), \xi_2(t), \cdots, \xi_m(t)]^{\mathrm{T}} = \mathbf{q}(\mathbf{y}(t), t) \in \mathbb{R}^m, \quad (5)$$

where $\xi_h(t)$ is the hth element of $\boldsymbol{\xi}(t)$, $\forall h \in \{1, 2, \cdots, m\}$.

Based on the design formula constructed in [30], [31]: $\dot{\boldsymbol{\xi}}(t) = -\gamma \boldsymbol{\xi}(t) - \lambda \int_0^t \boldsymbol{\xi}(\alpha) d\alpha$, a continuous-time NTN (CT-NTN) algorithm perturbed by noises is designed as

$$\begin{aligned} H(\mathbf{y}(t), t) \dot{\mathbf{y}}(t) = &-\gamma \mathbf{q}(t) - \dot{\mathbf{q}}_t(\mathbf{y}(t), t) - \\ &\lambda \int_0^t \mathbf{q}(\mathbf{y}(\alpha), \alpha) d\alpha + \epsilon(t), \qquad (6) \end{aligned}$$

where $\lambda > 0$ and $\gamma > 0$ with an invertible Hessian matrix; $\epsilon(t)$ represents additive noises in the system, which can be classified as constant noises, linear noises and random noises.
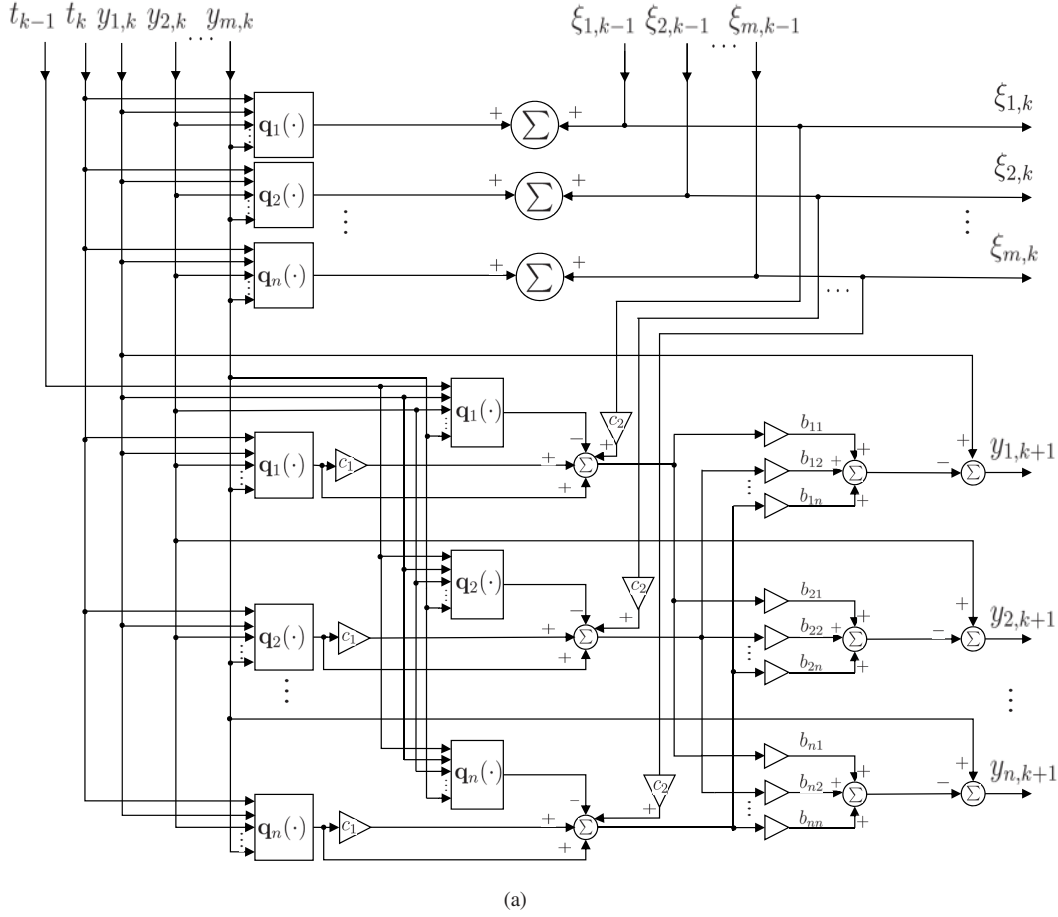
(a)

Fig. 1. Schematic circuit diagram realizing NTN algorithm (13) for solving FDNO problem (1), where (13) can be rewritten as following two equations: $\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - H^{-1}(\mathbf{y}(t_k), t_k)\big((c_1+1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2\boldsymbol{\xi}(t_k)\big)$ and $\boldsymbol{\xi}(t_k) = \xi(t_{k-1}) + \mathbf{q}(\mathbf{y}(t_k), t_k)$, with $B = H^{-1}(\mathbf{y}(t_k), t_k)$ and its $\sigma\varsigma$th entry denoted by $b_{\sigma\varsigma}$. The subscripts "m" and "n" represent the dimension of the input $(\mathbf{y}(t_k), t_k)$ and the gradient $\mathbf{q}(\mathbf{y}(t), t)$, respectively.

Then, the CT-NTN algorithm can be formulated as

$$\dot{\mathbf{y}}(t) = -H^{-1}(\mathbf{y}(t), t)\big(\gamma\mathbf{q}(\mathbf{y}(t), t) + \dot{\mathbf{q}}_t(\mathbf{y}(t), t) + \lambda \int_0^t \mathbf{q}(\mathbf{y}(\alpha), \alpha)\mathrm{d}\alpha + \epsilon(t)\big)$$
$$= -H^{-1}(\mathbf{y}(t), t)\big(\gamma\left(\frac{\partial\Phi(\mathbf{y}(t), t)}{\partial\mathbf{y}(t)}\right) + \frac{\partial^2\Phi(\mathbf{y}(t), t)}{\partial\mathbf{y}(t)\partial t} + \lambda \int_0^t \frac{\partial\Phi(\mathbf{y}(\alpha), \alpha)}{\partial\mathbf{y}(\alpha)}\mathrm{d}\alpha + \epsilon(t)\big). \tag{7}$$

It has been proven in [25] that the FDNO (2) achieves its minimal when the solution to equation (7) is obtained with a position definite $H(\mathbf{y}(t), t)$.

C. Existing Discrete-Time Solutions

Existing discrete-time solutions are presented here for comparison. The discrete-time zeroing dynamics (DTZD) model derived from [30], [31] is formulated as

$$\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - H^{-1}(\mathbf{y}(t_k), t_k)\big((c+1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1})\big), \tag{8}$$

where step-size $c = \delta\gamma > 0$. The three-step DTZD model obtained from [30]–[32] is

$$\mathbf{y}(t_{k+1}) = \frac{3}{2}\mathbf{y}(t_k) - \mathbf{y}(t_{k-1}) + \frac{1}{2}\mathbf{y}(t_{k-2}) - H^{-1}(\mathbf{y}(t_k), t_k)\big((c+1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1})\big), \tag{9}$$

and the four-step DTZD model [30]–[32] is

$$\mathbf{y}(t_{k+1}) = \frac{-1}{8}\mathbf{y}(t_k) + \frac{3}{4}\mathbf{y}(t_{k-1}) + \frac{5}{8}\mathbf{y}(t_{k-2}) - \frac{1}{4}\mathbf{y}(t_{k-3}) - \frac{9}{4}H^{-1}(\mathbf{y}(t_k), t_k)\big((c+1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1})\big). \tag{10}$$

In addition, the five-step DTZD model [33] is also presented here:

$$\mathbf{y}(t_{k+1}) = \frac{6}{13}\mathbf{y}(t_k) + \frac{2}{13}\mathbf{y}(t_{k-1}) + \frac{4}{13}\mathbf{y}(t_{k-2}) + \frac{3}{13}\mathbf{y}(t_{k-3}) - \frac{2}{13}\mathbf{y}(t_{k-4}) - \frac{24}{13}H^{-1}(\mathbf{y}(t_k), t_k)\big((c+1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1})\big). \tag{11}$$

Besides, the NRI model in [34] is provided as follows:

$$\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - H^{-1}(\mathbf{y}(t_k), t_k)\mathbf{q}(\mathbf{y}(t_k), t_k). \quad (12)$$

D. NTN and NTN-BFGS Neural Algorithms

Noise-interference is ever present during the solving process, e.g., the observational error, the truncation error, the quantization error and the sampling error. Therefore, a high-speed algorithm with noise-tolerant competence for solving the FDNO problem is in demand. In this section, discrete-time NTN and NTN-BFGS algorithms are derived to tolerate noises during the solution.

In order to simplify the structure of discrete-time NTN algorithm, the numerical differentiation formula with the least items is chosen. Thus, substituting Euler forward difference [27] to CT-NTN algorithm (7), we can obtain discrete-time NTN algorithm:

$$\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - H^{-1}(\mathbf{y}(t_k), t_k)\big((c_1 + 1)\mathbf{q}(\mathbf{y}(t_k), t_k) -$$
$$\mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j)\big), \quad (13)$$

where step-size $c_1 = \delta\gamma > 0$ and $c_2 = \delta\lambda > 0$; noise term $\epsilon(t_k)$ is eliminated to depict the NTN algorithm structure only. Sum term of $\mathbf{q}(\mathbf{y}(t_j), t_j)$, which is discretized from integral term of (7), plays a significant role in offsetting the impact brought by abrupt disturbance for (13). The circuit diagram showing general components of NTN algorithm (13) is illustrated in Fig. 1.

To solve $\mathbf{y}(t_{k+1})$ through NTN algorithm (13), the calculation of Hessian matrix inversion is unavoidable, which could be quite costly if Hessian matrix is complicated. Besides, it is not likely to make the calculation offline because $H^{-1}(\mathbf{y}(t_k), t_k)$ is required to be computed online. In order to overcome this drawback, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [35] is utilized in this section.

The highlight of BFGS algorithm [35] is its capability to explicitly escape the usage of inverted Hessian matrix, reducing the computation complexity. In BFGS, exact Hessian matrix is replaced by an approximation consisting of least-change updates generated from gradient at every iteration. It is known in Section II-A that $\Phi(\cdot, \cdot)$ is a convex function whose Hessian matrix is positive-definite. Thus we can conclude that the Hessian approximation matrix obtained by BFGS algorithm [35] converges to the Hessian matrix inversion.

The following NTN-BFGS algorithm is given for FDNO (1) in case of computing the Hessian matrix inversion being expensive.

$$\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - \widetilde{D}_k(\mathbf{y}(t_k), t_k)\big((c_1 + 1)\mathbf{q}(\mathbf{y}(t_k), t_k) -$$
$$\mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j)\big), \quad (14)$$

where $\widetilde{D}_k(\mathbf{y}(t_k), t_k)$ is the approximation of $H^{-1}(\mathbf{y}(t_k), t_k)$ supported by BFGS iterative formula:

$$\widetilde{D}_{k+1} = \widetilde{D}_k + \Delta\widetilde{D}_k, \quad (15)$$

where

$$\Delta\widetilde{D}_k = \frac{(s_k^{\mathrm{T}}z_k + z_k^{\mathrm{T}}\widetilde{D}_k z_k)(s_k s_k^{\mathrm{T}})}{(s_k^{\mathrm{T}}z_k)^2} - \frac{(\widetilde{D}_k z_k s_k^{\mathrm{T}} + s_k z_k^{\mathrm{T}}\widetilde{D}_k)}{(s_k^{\mathrm{T}}z_k)}; \quad (16)$$

$\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) + s_k$; $\widetilde{D}_k(\mathbf{y}(t_{k+1}) - \mathbf{y}(t_k)) = z_k$. Computing (16) has no temporary matrices. Besides, scalar $s_k^{\mathrm{T}}z_k$, $z_k^{\mathrm{T}}\widetilde{D}_k z_k$ and symmetric $\widetilde{D}_k$ accelerate the computation. The initial iterative value $\widetilde{D}_0$ should be positive-definite to achieve rapid convergence, wherein $\widetilde{D}_0 = I$ is a typical choice.

Remark 1: The BFGS iterative formula (15) uses the hypothesis of short-time invariance, which pays the price to avoid expensive computation on the inversion of $H(\mathbf{y}(t_k), t_k)$. Even so, the major scheme of NTN-BFGS algorithm (14) still breaks the hypothesis of short-time invariance, which is supported by the following theoretical analyses.

For one thing, NTN-BFGS algorithm (14) exploits the time difference [i.e., term $\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1})$] during the real-time solution process and therefore adapts to the change of coefficients in a predictive manner, making itself suitable for solving FDNO (1), whereas many conventional algorithms do not. For another, NTN-BFGS algorithm (14) exploits the error-feedback information [i.e., term $c_1\mathbf{q}(\mathbf{y}(t_k), t_k)$] as the input to handle the occurrence of computational errors. Additionally, sum term of $\mathbf{q}(\mathbf{y}(t_j), t_j)$ plays a significant role in offsetting the impact brought by abrupt disturbance.

III. THEORETICAL ANALYSES AND RESULTS

NTN algorithm (13) polluted by noises is written as

$$\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - H^{-1}(\mathbf{y}(t_k), t_k)\big((c_1 + 1)\mathbf{q}(\mathbf{y}(t_k), t_k) -$$
$$\mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j) + \epsilon(t_k)\big). \quad (17)$$

Theorem 1: There is an equivalence between NTN algorithm (13) and the following equation

$$\boldsymbol{\xi}(t_{k+1}) + (c_1 - 1)\boldsymbol{\xi}(t_k) + c_2 \sum_{j=0}^{k} \boldsymbol{\xi}(t_j) + \mathbf{O}(\delta^2) = 0, \quad (18)$$

where $\mathbf{O}(\delta^2)$ denotes the vector of truncation errors with each entry being $O(\delta^2)$.

Proof: (13) can be rewritten as

$$H(\mathbf{y}(t_k), t_k)(\mathbf{y}(t_{k+1}) - \mathbf{y}(t_k)) = -(c_1 + 1)\mathbf{q}(\mathbf{y}(t_k), t_k) +$$
$$\mathbf{q}(\mathbf{y}(t_k), t_{k-1}) - c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j). \quad (19)$$

Then, we can obtain

$$- c_1\mathbf{q}(\mathbf{y}(t_k), t_k) - c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j)$$
$$= H(\mathbf{y}(t_k), t_k)\big(\mathbf{y}(t_{k+1}) - \mathbf{y}(t_k)\big) + \big(\mathbf{q}(\mathbf{y}(t_k), t_k) -$$
$$\mathbf{q}(\mathbf{y}(t_k), t_{k-1})\big). \quad (20)$$

The Euler forward difference [27] is utilized for

$$\mathbf{y}(t_{k+1}) - \mathbf{y}(t_k) = \delta\dot{\mathbf{y}}(t_k) + \mathbf{O}(\delta^2),$$

and

$$\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_{k-1}), t_{k-1}) = \delta \dot{\mathbf{q}}(\mathbf{y}(t_k), t_k) + \mathbf{O}(\delta^2).$$

Substituting the above two formulas into (20) can directly generate

$$H(\mathbf{y}(t_k), t_k)(\delta \dot{\mathbf{y}}(t_k) + \mathbf{O}(\delta^2)) + \delta \dot{\mathbf{q}}(\mathbf{y}(t_k), t_k) + \mathbf{O}(\delta^2) =$$
$$- c_1 \mathbf{q}(\mathbf{y}(t_k), t_k) - c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j). \quad (21)$$

The discrete version of equation (4) is

$$\dot{\boldsymbol{\xi}}(t_k) = \dot{\mathbf{q}}_t(\mathbf{y}(t_k), t_k) + H(\mathbf{y}(t_k), t_k)\dot{\mathbf{y}}(t_k). \quad (22)$$

Then, substituting (22) into (21), a simple form of (21) with respect to error function is shown as

$$\delta \dot{\boldsymbol{\xi}}(t_k) + \mathbf{O}(\delta^2) = -c_1 \boldsymbol{\xi}(t_k) - c_2 \sum_{j=0}^{k} \boldsymbol{\xi}(t_j). \quad (23)$$

In addition, operating Euler forward difference [27] on $\boldsymbol{\xi}(\mathbf{y}(t_k), t_k)$ leads to

$$\delta \frac{\boldsymbol{\xi}(t_{k+1}) - \boldsymbol{\xi}(t_k)}{\delta} + \mathbf{O}(\delta^2) = -c_1 \boldsymbol{\xi}(t_k) - c_2 \sum_{j=0}^{k} \boldsymbol{\xi}(t_j), \quad (24)$$

which is simplified as

$$\boldsymbol{\xi}(t_{k+1}) + (c_1 - 1)\boldsymbol{\xi}(t_k) + c_2 \sum_{j=0}^{k} \boldsymbol{\xi}(t_j) + \mathbf{O}(\delta^2) = 0. \quad (25)$$

The proof is completed. $\qquad\square$

Remark 2: To prove the linear property of formula (18), we rewrite (18) as a function $\boldsymbol{\xi}(t_{k+1}) = f(\boldsymbol{\xi}(t_k))$.

Setting a as a random coefficient, it is evident that $f(a\boldsymbol{\xi}(t_k)) = a\boldsymbol{\xi}(t_{k+1}) = af(\boldsymbol{\xi}(t_k))$, which proves the homogeneity of formula (18). Similarly, by substituting $\boldsymbol{\xi}(t_k) + \boldsymbol{\xi}(t_{k+1})$ into $f(\cdot)$, we can get $f(\boldsymbol{\xi}(t_k) + \boldsymbol{\xi}(t_{k+1})) = f(\boldsymbol{\xi}(t_k)) + f(\boldsymbol{\xi}(t_{k+1}))$, which proves the additivity of formula (18). In summary, the linear property of formula (18) is proved from aspects of homogeneity and additivity.

Regarding the noise suppressing property of NTN algorithm (13), we offer the following theoretical analyses.

Theorem 2: Consider FDNO (1). The residual error $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ of NTN algorithm (13) for solving future dynamic nonlinear optimization problem is $O(\delta^2)$, wherein $\|\cdot\|_2$ denotes 2-norm.

Proof: Using $\boldsymbol{\xi}^i(t_k)$ to denote the ith subsystem of $\boldsymbol{\xi}(t_k)$ generates

$$\boldsymbol{\xi}^i(t_k) + (c_1 - 1)\boldsymbol{\xi}^i(t_{k-1}) + c_2 \sum_{j=0}^{k-1} \boldsymbol{\xi}^i(t_j) + O(\delta^2) = 0. \quad (26)$$

Subtracting (26) from the ith subsystem of (18), we can get:

$$\boldsymbol{\xi}^i(t_{k+1}) = (2 - c_1 - c_2)\boldsymbol{\xi}^i(t_k) + (c_1 - 1)\boldsymbol{\xi}^i(t_{k-1}) + O(\delta^2). \quad (27)$$
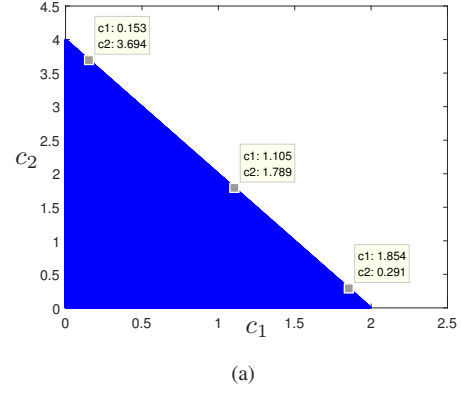

(a)

Fig. 2. Parameter region of convergence (PROC) on NTN algorithm (13). PROC upper border lies in function $c_2 = -2c_1 + 4$ which is approximated by least square method. c1 and c2 in the picture correspond to step-size c_1, c_2 respectively.

Letting $\boldsymbol{\Upsilon}^i(t_{k+1}) = [\boldsymbol{\xi}^i(t_{k+1}), \boldsymbol{\xi}^i(t_k)]^{\mathrm{T}}$, we can transform equation (27) into the pattern of state-space matrix:

$$\boldsymbol{\Upsilon}^i(t_{k+1}) = U\boldsymbol{\Upsilon}^i(t_k) + \mathbf{O}(\delta^2), \quad (28)$$

where

$$U = [(2 - c_1 - c_2), (c_1 - 1); 1, 0]. \quad (29)$$

Taking (28) into Minkowski's inequality [36] for the 2-norm, we have

$$\begin{aligned}
\|\boldsymbol{\Upsilon}^i(t_{k+1})\|_2 &\leq \|U\boldsymbol{\Upsilon}^i(t_k)\|_2 + \|\mathbf{O}(\delta^2)\|_2 \\
&= \|U\boldsymbol{\Upsilon}^i(t_k)\|_2 + O(\delta^2) \\
&\leq \|U\|_2 \|\boldsymbol{\Upsilon}^i(t_k)\|_2 + O(\delta^2) \\
&\leq \|U\|_2 \|U\boldsymbol{\Upsilon}^i(t_{k-1})\|_2 + \|U\|_2 O(\delta^2) + O(\delta^2) \\
&= \|U\|_2 \|U\boldsymbol{\Upsilon}^i(t_{k-1})\|_2 + O(\delta^2) \\
&\vdots \\
&\leq \|U\|_2^k \|\boldsymbol{\Upsilon}^i(t_1)\|_2 + O(\delta^2). \quad (30)
\end{aligned}$$

Matrix U (29) has two different eigenvalues which are $\mu_1 = 1 - (c_1 + c_2 + \sqrt{(c_1 + c_2)^2 - 4c_2})/2$ and $\mu_2 = 1 - (c_1 + c_2 - \sqrt{(c_1 + c_2)^2 - 4c_2})/2$. The region shown in Fig. 2 describes the value of c_1, c_2 that make the real part of μ_1, μ_2 ranging from -1 to 1. Thereby, we can obtain $\lim_{k\to\infty} \|U\|_2^k = 0$ which further assists that

$$\lim_{k\to\infty} \|\boldsymbol{\Upsilon}^i(t_{k+1})\|_2 \leq \lim_{k\to\infty} \|U\|_2^k \|\boldsymbol{\Upsilon}^i(t_1)\|_2 + O(\delta^2) = O(\delta^2). \quad (31)$$

As long as c_1, c_2 which we choose for (13) belonging to that parameter region of convergence (PROC) depicted in Fig. 2, we have $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ of NTN algorithm (13) convergent to $O(\delta^2)$. The proof is completed. $\qquad\square$

Theorem 3: Consider FDNO (1). Polluted by arbitrary linear noise $\epsilon(t_k) = \eta\delta k + \Omega$, the residual error $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ of NTN algorithm (13) for solving future dynamic nonlinear optimization problem is $\|\eta\delta/c_2\|_2 + O(\delta^2)$; for constant noise $\epsilon(t_k) = \Omega$ degraded from linear noises with $\eta = 0$, the residual error of NTN algorithm (13) is $O(\delta^2)$, which has nothing to do with the value of constant noise Ω.

Proof: The factors that determine the residual error of noise-polluted NTN algorithm (17) can be classified as $\mathbf{O}(\delta^2)$ and injecting constant noise $\epsilon(t_k)$. The linear property of (18) which is proved in *Remark 2* allows us to independently investigate the two factors.

Firstly, we simply change the form of (18) as

$$\boldsymbol{\xi}(t_{k+1}) = \left((1-c_1)\boldsymbol{\xi}(t_k) - c_2 \sum_{j=0}^{k} \boldsymbol{\xi}(t_j)\right) + \mathbf{O}(\delta^2). \quad (32)$$

It has been proven in Theorem 2 that the residual error of (32) is $O(\delta^2)$.

Next, consider how constant noises influence the convergent performance of

$$\boldsymbol{\xi}(t_{k+1}) + (c_1 - 1)\boldsymbol{\xi}(t_k) + c_2 \sum_{j=0}^{k} \boldsymbol{\xi}(t_j) + \epsilon(t_k) = 0. \quad (33)$$

In a more general sense, constant noise $\epsilon(t_k) = \Omega$ is a subcase of linear time-variant noise $\epsilon(t_k) = \eta \delta k + \Omega$, $k = 0, 1, 2, \cdots$. Thus, the subsystem of (33) can be rewritten as

$$\boldsymbol{\xi}^i(t_{k+1}) = \left((1-c_1)\boldsymbol{\xi}(t_k) - c_2 \sum_{j=0}^{k} \boldsymbol{\xi}(t_j)\right) + \eta^i k \delta + \Omega^i. \quad (34)$$

The Z-transformation of (34) is

$$\boldsymbol{\xi}^i(z) = \frac{-\Omega^i(z-1)^2 - \eta^i \delta z + z(z-1)^2 \xi^i(0)}{z(z-1)^2 + (c_1-1)(z-1)^2 + c_2 z(z-1)}, \quad (35)$$

where $\xi^i(0)$ is the initiation of $\xi^i(z)$ and its poles are $z_1 = 1 - \left(c_1 + c_2 + \sqrt{(c_1+c_2)^2 - 4c_2}\right)/2$ and $z_2 = 1 - \left(c_1 + c_2 - \sqrt{(c_1+c_2)^2 - 4c_2}\right)/2$. It is the same as the range of c_1, c_2 in Fig. 2. Thus, utilizing Z-transformation final theorem of (35), the limit formula of $\xi^i(t_k)$ is

$$\lim_{k \to \infty} \boldsymbol{\xi}^i(t_k) = \lim_{z \to 1} (z-1)\boldsymbol{\xi}^i(z)$$
$$= \lim_{z \to 1} \frac{-\Omega^i(z-1)^2 - \eta^i \delta z + z(z-1)^2 \xi^i(0)}{z(z-1) + (c_1-1)(z-1) + c_2 z} = \frac{-\eta^i \delta}{c_2}. \quad (36)$$

Summing up, taking linear noise $\epsilon(t_k) = \eta \delta k + \Omega$ and the residual error of (32) into account, the residual error $\lim_{k \to \infty} \|\boldsymbol{\xi}(t_k)\|_2$ of NTN algorithm (17) is $\|\eta \delta / c_2\|_2 + O(\delta^2)$; Setting $\eta = 0$, we can know that when it comes to constant noise $\epsilon(t_k) = \Omega$, it is for certain that the residual error $\lim_{k \to \infty} \|\boldsymbol{\xi}(t_k)\|_2$ of NTN algorithm (17) is $O(\delta^2)$, which has nothing to do with the value of constant noises. The proof is completed. $\square$

For further investigation, the ensuring theorem reveals how NTN algorithm (13) handles bounded random noises.

Theorem 4: Consider FDNO (1). For bounded random noise $\epsilon(t) = \rho$, NTN algorithm (13) keeps a bounded residual error $\lim_{k \to \infty} \|\boldsymbol{\xi}(t_k)\|_2$, which is $2m \sup_{1 \leq \iota \leq k, \ 1 \leq i \leq m} |\rho_\iota^i|/(1 - \|U\|_2) + O(\delta^2)$.

Proof: In accordance with superposition principle exploited in

Theorem 3, the difference equation is generated:

$$\boldsymbol{\xi}^i(t_{k+1}) = (2 - c_1 - c_2)\boldsymbol{\xi}^i(t_k) + (c_1 - 1)\boldsymbol{\xi}^i(t_{k-1}) + \rho^i(t_k) - \rho^i(t_{k-1}). \quad (37)$$

Let $\varpi^i(t_k) = [\rho^i(t_k) - \rho^i(t_{k-1}), 0]^{\mathrm{T}}$, the aforesaid (37) in terms of $\Upsilon^i(t_k)$ is $\Upsilon^i(t_{k+1}) = U\Upsilon^i(t_k) + \varpi^i(t_k)$ and $U = [(2 - c_1 - c_2), (c_1 - 1); 1, 0]$. Then, it is of foundation to obtain

$$\|\Upsilon^i(t_{k+1})\|_2$$
$$\leq \|U\Upsilon^i(t_k)\|_2 + \|\varpi^i(t_k)\|_2$$
$$\leq \|U\|_2 \|\Upsilon^i(t_k)\|_2 + \|\varpi^i(t_k)\|_2$$
$$\leq \|U\|_2 \|U\Upsilon^i(t_{k-1})\|_2 + \|U\|_2 \|\varpi^i(t_{k-1})\|_2 + \|\varpi^i(t_k)\|_2$$
$$\vdots$$
$$\leq \|U\|_2^k \|\Upsilon^i(t_1)\|_2 + \|U\|_2^{k-1} \|\varpi^i(t_1)\|_2 + \ldots + \|\varpi^i(t_k)\|_2$$
$$< \|U\|_2^k \|\Upsilon^i(t_1)\|_2 + \max_{1 \leq \iota \leq k} \|\varpi_\iota^i\|_2/(1 - \|U\|_2)$$
$$< \|U\|_2^k \|\Upsilon^i(t_1)\|_2 + 2 \max_{1 \leq \iota \leq k} |\rho_\iota^i|/(1 - \|U\|_2). \quad (38)$$

Moreover, with $\lim_{k \to \infty} \|U\|_2^k = 0$, we can get

$$\lim_{k \to \infty} \|\Upsilon_{k+1}^i\|_2 < 2 \max_{1 \leq \iota \leq k} |\rho_\iota^i|/(1 - \|U\|_2).$$

Finally, we come to the conclusion that

$$\lim_{k \to \infty} \|\boldsymbol{\xi}(t_k)\|_2 < 2m \sup_{1 \leq \iota \leq k, \ 1 \leq i \leq m} |\rho_\iota^i|/(1 - \|U\|_2) + O(\delta^2).$$

The proof is completed. $\square$

BFGS algorithm has been found dramatically helpful for easing the expensive computational burden for NTN algorithm (13) while the Hessian matrix inversion is diffcult to get. Several theorems regarding to NTN-BFGS algorithm (14) are provided as follows. In addition, noises denote by $\epsilon(t)$ polluting NTN-BFGS algorithm (14)

$$\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - \widetilde{D}_k(\mathbf{y}(t_k), t_k)\big((c_1 + 1)\mathbf{q}(\mathbf{y}(t_k), t_k)$$
$$- \mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j) + \epsilon(t_k)\big). \quad (39)$$

Theorem 5: Consider FDNO (1). Given that $\|\widetilde{D}_k - H_k^{-1}\|_2 = O(\delta^2)$, the theoretical solution generated by NTN-BFGS algorithm (14) converges to that solutioned by NTN algorithm (13) with the residual error being $O(\delta^2)$.

Proof: To begin this proof, $\widetilde{\mathbf{y}}(t_k)$ and $\mathbf{y}(t_k)$ are used to denote the solutions of NTN-BFGS algorithm (14) and NTN algorithm (13), respectively. We further substitute $\|\widetilde{D}_k - H_k^{-1}\|_2 =$

TABLE I
THE NOISE LEVEL THAT NTN ALGORITHM (13) CAN HANDLE WITH THE DIFFERENT PRECISION χ OF THE RESIDUAL ERROR $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ FOR SOLVING FDNO (41) WITH $\delta = 0.0001$ s, $c_1 = 0.05$, $c_2 = 0.6$

$\chi <$	1.0×10^{-3}	1.0×10^{-2}	1.0×10^{-1}
Linear noises: $\eta <$	3.0	3×10^1	3×10^2
random noises: $\rho^+ - \rho^- <$	2.0×10^{-3}	2.0×10^{-2}	2.0×10^{-1}

TABLE II
THE NOISE LEVEL THAT NTN-BFGS ALGORITHM (14) CAN HANDLE WITH THE DIFFERENT PRECISION χ OF THE RESIDUAL ERROR $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ FOR SOLVING FDNO (41) WITH $\delta = 0.0001$ s, $c_1 = 0.05$, $c_2 = 0.6$

$\chi <$	1.0×10^{-3}	1.0×10^{-2}	1.0×10^{-1}
Linear noises: $\eta <$	3.0	3×10^1	3×10^2
random noises: $\rho^+ - \rho^- <$	1.0×10^{-3}	1.0×10^{-2}	1.0×10^{-1}

$O(\delta^2)$ into (14) and then obtain

$$
\begin{aligned}
\widetilde{\mathbf{y}}(t_{k+1}) &= \mathbf{y}(t_k) - (H_k^{-1} + O(\delta^2))\big((c_1+1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \\
&\quad \mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j)\big) \\
&= \mathbf{y}(t_k) - (H_k^{-1}\big((c_1+1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \\
&\quad \mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j)\big) + \\
&\quad O(\delta^2) \\
&= \mathbf{y}(t_{k+1}) + O(\delta^2). \quad\quad (40)
\end{aligned}
$$

It is evident that $\widetilde{\mathbf{y}}(t_{k+1}) = \mathbf{y}(t_{k+1}) + O(\delta^2)$. Therefore, we can get that the residual error relationship between NTN-BFGS algorithm (14) and NTN algorithm (13): $\|\widetilde{\mathbf{y}}(t_k) - \mathbf{y}(t_k)\|_2 = O(\delta^2)$. The proof is completed. $\square$

Theorem 5 verifies the feasibility of using NTN-BFGS algorithm (14) to replace inverse calculation from the perspective of theoretical derivation. In addition, the following theorem is provided to reveal the effectiveness of the NTN-BFGS algorithm (14).

Theorem 6: Consider FDNO (1).

1) When it comes to NTN-BFGS algorithm (14) for solving future dynamic nonlinear optimization problem, the residual error $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ is $O(\delta^2)$.
2) With respect to noise-polluted NTN-BFGS algorithm (39) for solving future dynamic nonlinear optimization problem, the residual error $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ is $O(\delta^2)$ which has nothing to do with the value of constant noise $\epsilon(t)$.
3) For a bounded random noise $\epsilon(t) = \rho$, the residual error of NTN-BFGS algorithm (39) is bounded, with its upper bound $\lim_{k\to\infty} \|\boldsymbol{\xi}(t_k)\|_2$ being $2m \sup_{1\le\iota\le k,\ 1\le i\le m} |\rho_\iota^i|/(1 - \|U\|_2) + O(\delta^2)$.

Proof: It can be generalized from proofs in Theorem 2 though Theorem 4, and thus omitted. The proof is completed. $\square$

Remark 3: To illustrate how noise level influences the performance of NTN algorithm (13), upper bounds of every

kind of noises are given with the precision of the residual error being χ according to Theorem 2-5.

For constant noise $\epsilon(t_k) = \Omega$, there is no upper bound for the residual error of NTN algorithm (14) invariably being $O(\delta^2)$. For linear noise $\epsilon(t_k) = \eta\delta k + \Omega$, there only exists an upper bound of the rate of change η, which is $c_2\chi/\delta + O(\delta^2)$. For random noises, the upper bound lies in $\sup_{1\le\iota\le k,\ 1\le i\le m} |\rho_\iota^i| \le (1 - \|U\|_2)\chi + O(\delta^2)/2m$.

In addition, based on Theorem 6, the above conclusion can be applied to NTN-BFGS algorithm (14).

Via the theorem proposed in [25], the residual error of traditional methods which are intrinsically constructed to solve static optimization problems for future dynamic nonlinear optimization is $O(\delta)$, within computing interval $[0, \delta]$. Besides, a step-by-step methodology to carry out noise-polluted NTN algorithm (17) and NTN-BFGS algorithm (39) is presented in Algorithm 1.

IV. NUMERICAL EXPERIMENTS

In this section, the effectiveness of the proposed NTN algorithm (13) and NTN-BFGS algorithm (14) with rapid calculating ability for FDNO (1) is substantiated through numerical experiments in the presence of noises. Meanwhile, several representative existing models, which are DTZD model (8), three-step DTZD model (9), four-step DTZD model (10), five-step DTZD model (11) and NRI (12) model, are also used to solve the same FDNO problem in the presence of noises for comparison.

A. Example 1

In this subsection, one FDNO benchmark problem originated from [25] is addressed by above mentioned discrete models:

$$
\begin{aligned}
&\min_{\mathbf{y}(t_{k+1})\in\mathbb{R}^4} \Phi(\mathbf{y}(t_k), t_k) \\
&= (y_1(t_k) + t_k)^2 + (y_2(t_k) + t_k)^2 + (y_3(t_k) - \\
&\quad \exp(-t_k))^2 + 0.1(t_k - 1)y_3(t_k)y_4(t_k) - (y_1(t_k) + \\
&\quad \ln(0.1t_k + 1))(y_2(t_k) + \sin(t_k)) + (y_1(t_k) + \\
&\quad \sin(t_k))y_3(t_k) + (y_4(t_k) + \exp(-t_k))^2, \quad\quad (41)
\end{aligned}
$$

where $[k\delta, (k+1)\delta] \in [0, 10]$ denotes the kth time computing interval, $\forall k \in \{0, 1, 2, \cdots\}$, and $\mathbf{y}(t_k) = [y_1(t_k), y_2(t_k), y_3(t_k), y_4(t_k)]$. In order to evaluate different performances generated by models just mentioned, we choose the L_2 norm of residual error which is $\|\boldsymbol{\xi}(t_k)\|_2 = \|\partial\Phi(\mathbf{y}(t), t)/\partial\mathbf{y}(t)\|_2$ as the monitoring variable.

To begin with, Fig. 3 shows comparative performances of NRI model (12) and NTN algorithm (13) in the presence of random noises. In addition, the starting states $\mathbf{y}(t_0)$ used in computing process is randomly generated. Specifically speaking, as demonstrated in Fig. 3(a), NTN algorithm (13) successfully obtains the minimum value of $\Phi(\mathbf{y}(t_k), t_k)$ at each time instant, while NRI model (12) fails to cope with the interference caused by random noises. Furthermore, Fig. 3(b) shows that the residual error of NTN algorithm (13) converges to a very small value which is basically 10^3 times less than that of NRI model (12). In addition, the comparison of each
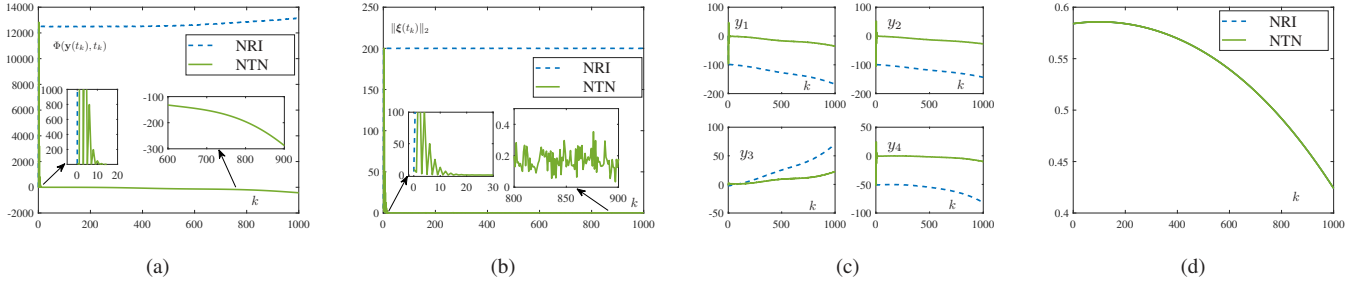
Fig. 3. Comparisons of NRI model (12) and NTN algorithm (13) for solving FDNO (41) with $\delta = 0.01$ s and random noise $\rho \in [99.9, 100.1]$. (a) Comparison of $\Phi(\mathbf{y}(t_k), t_k)$. (b) $\|\boldsymbol{\xi}(t_k)\|_2$ of NRI model (12) and NTN algorithm (13). (c) Comparison of $\mathbf{y}(t_k)$. (d) Comparison of minimal eigenvalue of Hessian matrix.
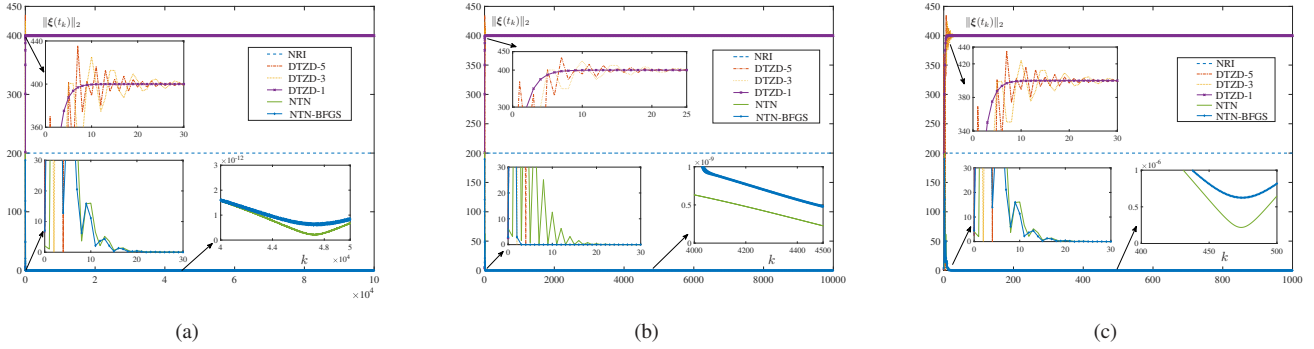

Fig. 4. Convergence performance of the proposed NTN algorithm (13), NTN-BFGS algorithm (14), DTZD model (8), three-step DTZD model (9), five-step DTZD model (11) and NRI model (12) and for solving FDNO (41) with constant noise $\epsilon(t) = 100$. (a) $\delta = 0.0001$ s. (b) $\delta = 0.001$ s. (c) $\delta = 0.01$ s.
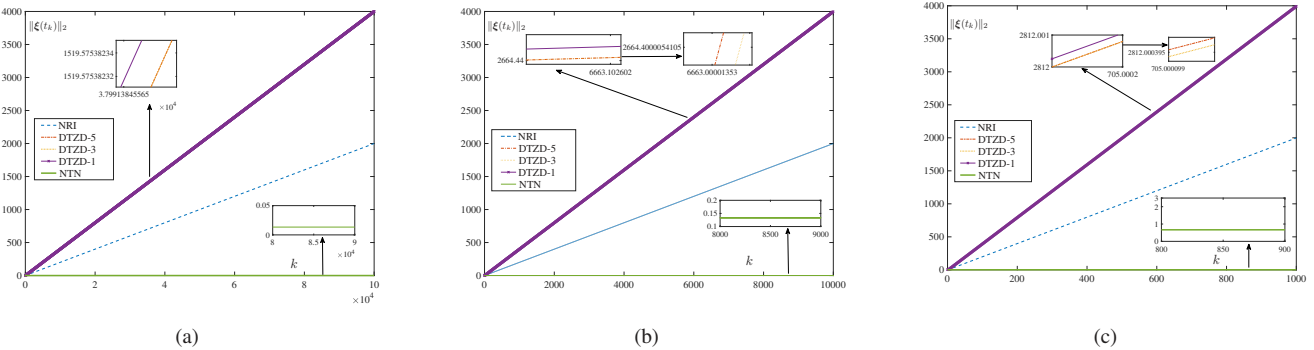

Fig. 5. Convergence performance of the proposed NTN algorithm (13), DTZD model (8), three-step DTZD model (9), five-step DTZD model (11) and NRI model (12) and for solving FDNO (41) with linear noise $\epsilon(t) = 100(t - 1)$. (a) $\delta = 0.0001$ s. (b) $\delta = 0.001$ s. (c) $\delta = 0.01$ s.

element trajectory of $\mathbf{y}(t_k)$ is plotted in Fig. 3(c). As shown in Fig. 3(d), Hessian matrix minimal eigenvalues of FDNO benchmark problem (41) obtained from two models coincide with each other, which are both larger than zero during the computational time. That is the prerequisites for utilizing NRI model (12) and NTN algorithm (13) on FDNO benchmark problem (41).

Next, no matter what kind of noises they are, we can use combinations of constant noises, linear noises and random noises to define it mathematically. Thereby, it is allowed to explore effects which noises exert on different models in a categorical way. Experimental results of FDNO benchmark problem (41) among all the aforementioned models in the presence of different kinds of additive noises are plotted Fig. 4 through Fig. 7.

1) *Constant Noises:* As shown in Fig. 4, DTZD model (8), three-step DTZD model (9), five-step DTZD model (11) and NRI model (12) basically fail to handle constant noises. The residual error generated by NRI model (12) is smaller than these three DTZD models. Fig. 4(a) through (c) indicate that constant noises have negligible influence on existing models with residual error being about 2×10^2 or 4×10^2. In contrast, since associating the error-integration term with the proposed novel NT-N algorithm (13) and the NTN-BFGS algorithm (14), residual error values of those two are expedited from order 10^{-7} to 10^{-12}, as time sampling interval being 0.01 s, 0.001 s and 0.0001 s.

2) *Linear Time-Varying Noises:* As observed from Fig. 5, when linear time-varying noise $\epsilon(t) = 100(t - 1)$, the
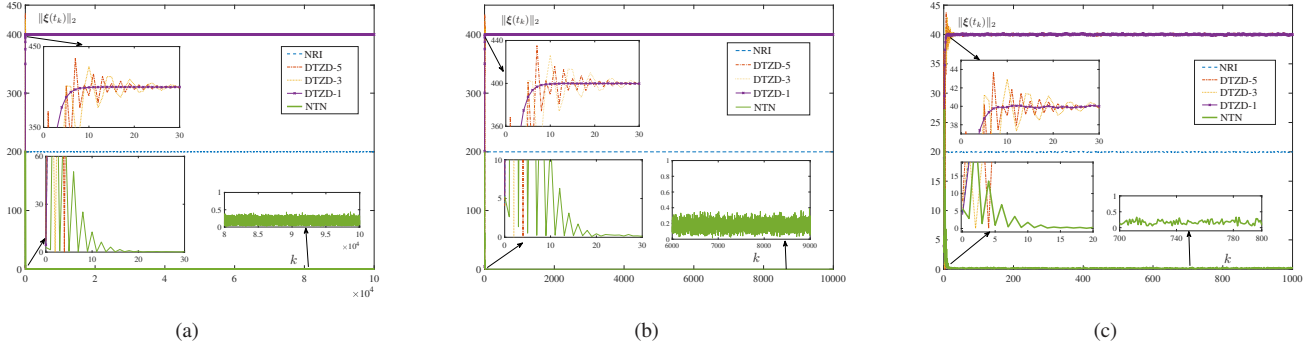
Fig. 6.   Convergence performance of the proposed NTN algorithm (13), DTZD model (8), three-step DTZD model (9), five-step DTZD model (11) and NRI model (12) and for solving FDNO (41) with random noise $\epsilon(t) \in [99.9, 100.1]$. (a) $\delta = 0.0001$ s. (b) $\delta = 0.001$ s. (c) $\delta = 0.01$ s.
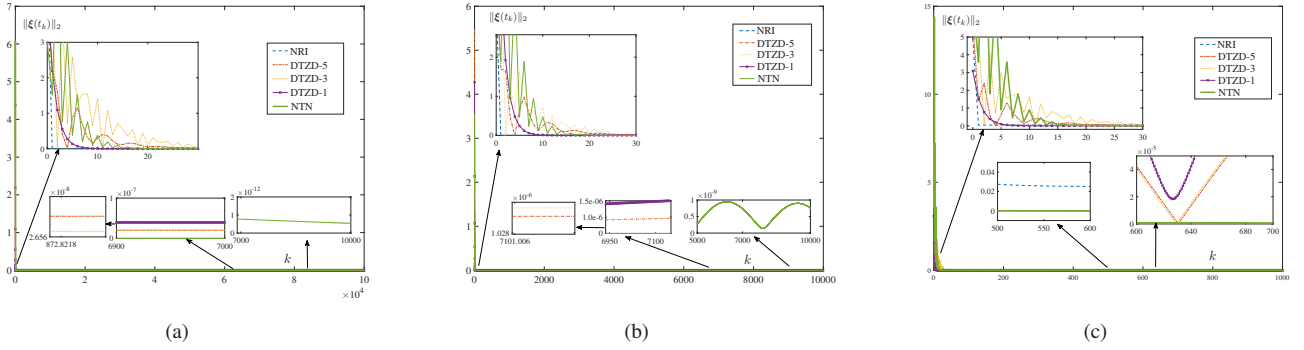

Fig. 7.   Convergence performance of the proposed NTN algorithm (13), DTZD model (8), three-step DTZD model (9), five-step DTZD model (11) and NRI model (12) and for solving FDNO (41) without noises. (a) $\delta = 0.0001$ s. (b) $\delta = 0.001$ s. (c) $\delta = 0.01$ s.

TABLE III
COMPARISON BETWEEN METHODS FOR SOLVING FUTURE DYNAMIC NONLINEAR OPTIMIZATION PROBLEM (41)

Method	Residual error in terms of different kinds of noises				Property in noisy environment		Essentially designed to solve time-variant problems
	Constant noises	Linear noises	Random noises	Zero noises	Convergent	Noise-tolerant	
NTN (13)	2.9×10^{-13}	1.0×10^{-2}	2.5×10^{-1}	7.6×10^{-13}	Yes	Yes	Yes
NTN-BFGS (14)	7.2×10^{-13}	3.3×10^{-2}	1.8×10^{-1}	1.4×10^{-12}	Yes	Yes	Yes
one-step DTZD (8)	8.0×10^{2}	infinite	4.0×10^{2}	1.6×10^{-8}	No†	No	Yes
three-step DTZD (9)	8.0×10^{2}	infinite	4.0×10^{2}	2.7×10^{-8}	No†	No	Yes
four-step DTZD (10)	8.0×10^{2}	infinite	4.0×10^{2}	7.7×10^{-8}	No†	No	Yes
five-step DTZD (11)	8.0×10^{2}	infinite	4.0×10^{2}	2.7×10^{-8}	No†	No	Yes
NRI model (12)	4.0×10^{2}	infinite	2.0×10^{2}	4.5×10^{-4}	No†	No	No

Parameters involved are set as follows: $\delta = 0.0001$ s, $c_2 = 0.6$, constant noise $\epsilon(t) = 200$, linear noise $\epsilon(t) = 100(t-1)$, random noise $\epsilon(t) \in [99.9, 100.1]$;

For c_1, four-step DTZD (10) is 0.05; the other aforementioned algorithms is 0.5.

No† means that influences exerted by noises is non-negligible

residual error of NTN algorithm (13) holds at the order of 10^0, 10^{-1} and 10^{-2}, respectively. The above result of convergence rate verifies that NTN algorithm (13) follows $\mathbf{O}(\delta)$ convergent speed proved in Theorem 3. DTZD model (8), three-step DTZD model (9), five-step DTZD model (11) and NRI model (12), of which residual errors linearly change with time as rather large slope, are not capable of solving FDNO problems with linear noises.

3) *Random Noises:* From Fig. 6, it is evident that the maximum steady residual error synthesized by NTN algorithm (13) is restricted to 0.2, which is 1×10^2 times

less than mean value of input random noises. In contrast, other existing models have pretty high residual errors.

4) *No Noises:* Observing contrasting principle, Fig. 7 displays results of benchmark problem (41) as to no noises. All the discrete-time models perform accurately under the condition that there is no perturbation. Besides, NTN algorithm (13) synthesizes the smallest residual error compared with other discrete-time existing models. From the perspective of convergence rate, NRI model (12) has the best performance. Meanwhile, subgraphs of Fig. 7 indicate that NTN algorithm (13) converges a little bit slower than DTZD model (8) and NRI model
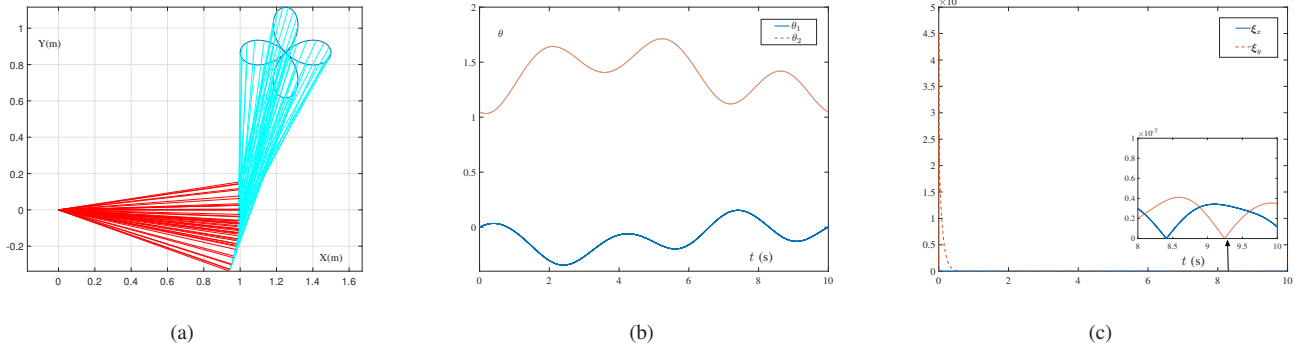
Fig. 8. End-effector movement of a two-link robot following a four-leaf-clover trajectory synthesized by NTN algorithm (13) with $\delta = 0.001$ s. (a) Motion trajectories without perturbation. (b) Profiles of joint angle. (c) Profiles of position error in Cartesian coordinate system.
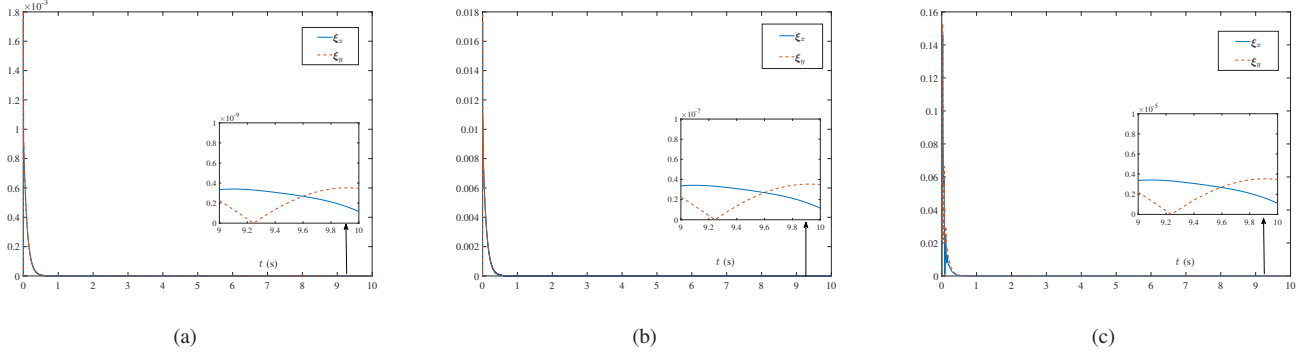

Fig. 9. Position errors in Cartesian coordinate system of a two-link robot tracking an ideal four-leaf-clover path obtained from NTN algorithm (13) with constant noise $\epsilon(t) = 10$. (a) $\delta = 0.0001$ s. (b) $\delta = 0.001$ s. (c) $\delta = 0.01$ s.
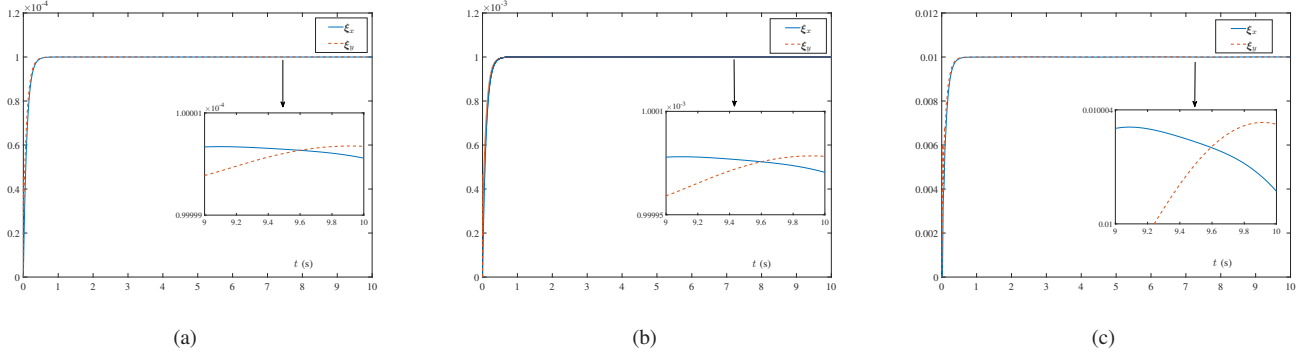

Fig. 10. Position errors in Cartesian coordinate system of a two-link robot tracking an ideal four-leaf-clover path obtained from NTN algorithm (13) with linearly time-dependent noise $\epsilon(t) = 10(t-1)$. (a) $\delta = 0.0001$ s. (b) $\delta = 0.001$ s. (c) $\delta = 0.01$ s.

(12), and the approximating integration term inevitably causes extra updates.

Note that, the noise level that NTN algorithm (13) and NTN-BFGS algorithm (14) can handle for solving FDNO (41) is shown in TABLE I and TABLE II, respectively. Besides, there is no upper bound of constant noises according to *Remark 3*. Table III is further given, which compares models of NTN algorithm (13), NTN-BFGS algorithm (14) and existing discrete models for solving FDNO problem (41) in the presence of noises. Summing up, we show how the theoretical analyses of NTN algorithm (13)and NTN-BFGS algorithm (14) are substantiated by experimental results.

Overall, FDNO benchmark problem (41) verifies that the

proposed NTN algorithm (13) and NTN-BFGS algorithm (14) possesses extraordinary ability in suppressing different kinds of noises, even their combinations, without any prior information of noises.

V. APPLICATION TO MOTION GENERATION

Robotic systems have evolved rapidly in engineering fields [37]–[40]. In this section, we apply NTN algorithm (13), NTN-BFGS algorithm (14) and NRI model (12) to solve the inverse kinematics of a two-link planar robot. Afterwards, let $\mathbf{a}(t) \in \mathbb{R}^2$ denotes the vector of practical end effector position in Cartesian coordinate system, meanwhile $\mathbf{a}_d(t)$ is the corresponding desired one; $\boldsymbol{\theta}(t) = [\theta_1(t), \theta_2(t)]^T$ symbolizes
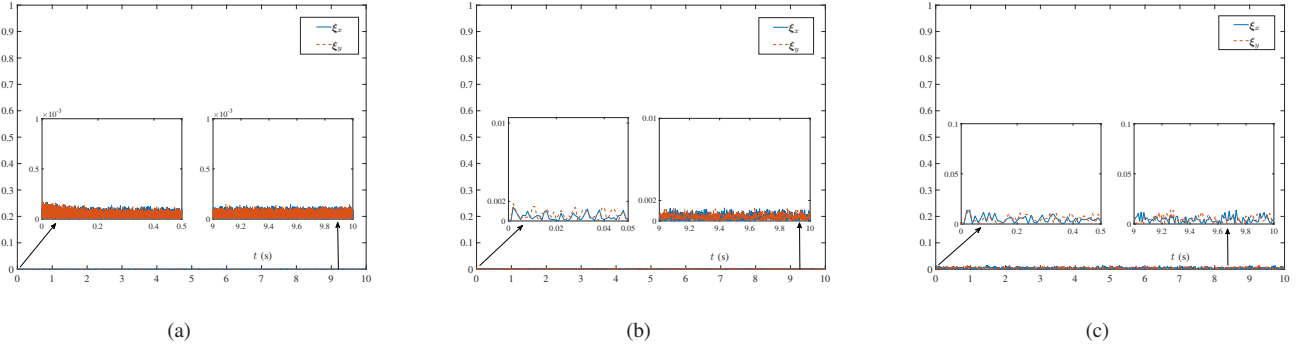
Fig. 11. Position errors in Cartesian coordinate system of a two-link robot tracking an ideal four-leaf-clover path obtained from NTN algorithm (13) with random noise $\epsilon(t) \in [9, 11]$. (a) $\delta = 0.0001$ s. (b) $\delta = 0.001$ s. (c) $\delta = 0.01$ s.

Algorithm 1 Solution to FDNO problem (1) generated by noise-polluted NTN algorithm (17) and NTN-BFGS algorithm (39)

---

1. initial set $m, maxstep, c_1, c_2, noise = \epsilon(t_k), \Phi(\cdot, \cdot)$, calculate random $\mathbf{y}(t_0)$ and $\mathbf{q}(\mathbf{y}(t_0), t_0)$

2.

for $(k = 1; k \leq maxstep; k++)$ **do**

$error = 0;$

calculate $\mathbf{q}(\mathbf{y}(t_k), t_k) = \partial \Phi(\mathbf{y}(t_k), t_k)/\partial \mathbf{y}(t_k);$

calculate $\sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j) = \mathbf{q}(\mathbf{y}(t_0), t_0) + \cdots + \mathbf{q}(\mathbf{y}(t_k), t_k);$

calculate $H(\mathbf{y}(t_k), t_k) = \partial \mathbf{q}(\mathbf{y}(t_k), t_k)/\partial \mathbf{y}(t_k);$

if NTN algorithm **then**

calculate $\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - H^{-1}(\mathbf{y}(t_k), t_k)((c_1 + 1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j) + noise);$

end if

if NTN-BFGS algorithm **then**

calculate $\mathbf{y}(t_{k+1})$ using $\widetilde{D}_{k+1} = \widetilde{D}_k + \Delta \widetilde{D}_k$, and $\mathbf{y}(t_{k+1}) = \mathbf{y}(t_k) - \widetilde{D}_k(\mathbf{y}(t_k), t_k)((c_1 + 1)\mathbf{q}(\mathbf{y}(t_k), t_k) - \mathbf{q}(\mathbf{y}(t_k), t_{k-1}) + c_2 \sum_{j=0}^{k} \mathbf{q}(\mathbf{y}(t_j), t_j + noise));$

end if

calculate $error$ equals to the Euclidean norm of $\mathbf{q}(\mathbf{y}(t_k), t_k);$

$ERROR(k) = \|\mathbf{q}(\mathbf{y}(t_k), t_k)\|_2;$

end for

3. plot the residual error

---

the joint-angle vector; $f(\cdot)$ represents the forward-kinematics mapping relation between the end effector position and the joint-angle [39], which means $f(\boldsymbol{\theta}(t)) = \mathbf{a}(t)$. Considering that T indicates the end point of solving process, every computational time interval can be expressed by $[k\delta, (k+1)\delta) \in [0, T]$, where $k = 0, 1, 2, \cdots$ and δ is sampling gap. Thereby, the cost function of the aforementioned robot motion could be defined as

$$\Phi(\boldsymbol{\theta}_k, t_k) = f^{\mathrm{T}}(\boldsymbol{\theta}_k)f(\boldsymbol{\theta}_k) - 2\mathbf{a}_{\mathrm{d}}^{\mathrm{T}}(t_k)f(\boldsymbol{\theta}_k). \tag{42}$$

Evidently, the solution of $\mathbf{a}(t)$ tracking $\mathbf{a}_{\mathrm{d}}(t)$ can be classified as the following FDNO problem:

$$\min_{\boldsymbol{\theta}_{k+1} \in \mathbb{R}^2} \Phi(\boldsymbol{\theta}_{k+1}, t_{k+1}). \tag{43}$$

For simple illustration, each link length is set as 1 m. To be specific, our aim is to use the two-link planar robot manipulator with NTN algorithm (13) and NRI model (12) embed to draw a four-leaf-clover.

The ensuring figures show comparative experimental results conducted under various experimental environments which differs in kinds of additive noises. For comparison, Fig. 8 plots profiles of the whole tracking motion trajectories, joint angle and the position error without the presence of noises. Through Fig. 9 to Fig. 11, it is shown that the given motion is completed well by NTN algorithm (13) where the position error is slight, despite additive noises. Specifically, adding constant noise $\epsilon(t) = 10$ to joint velocities when being solved by NTN algorithm (13), the position error of the manipulator's end-effector in Fig. 9(a) is less than 10^{-9} m. Thus, the proposed NTN algorithm (13) is feasible to industrial applications.

VI. CONCLUSIONS

In this paper, NTN algorithm (13) has been addressed for future dynamic nonlinear optimization problems in the presence of a class of noises affecting the system. Integration control from control-technique has been defined to minimize the cost function rapidly. The use of quasi-Newton BFGS method has been proposed through NTN-BFGS algorithm (14), which can not only deal with noises for the FDNO problem, but eliminate the expensive calculation of inversion matrices. Results obtained by numerical experiments are reported in comparison with DTZD model (8) , three-step DTZD model (9), five-step DTZD model (11) and NRI model (12), thereby highlighting the superiority of the proposed algorithms in robustness, efficacy and computational complexity when solving the FDNO problem with noises. Besides, a possible future research direction is the proposing of an algorithm independent of the short-time invariance hypothesis with the explicit matrix-inversion operation eliminated for solving FD-NO problems.

References

[1] Y. Zhang and J. Wang, "A dual neural network for constrained joint torque optimization of kinematically redundant manipulators," *IEEE Trans. Syst., Man, Cybern., B.*, vol. 32, no. 5, pp. 654–662, Oct. 2002.

[2] G. Grosan and A. Abraham, "A new approach for solving nonlinear equations systems," *IEEE Trans. Syst., Man, Cybern., A.*, vol. 38, no. 3, pp. 698–714, May 2008.

[3] C. Yang, K. Huang, H. Cheng, Y. Li, and C. Y. Su, "Haptic identification by ELM-controlled uncertain manipulator," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2398–2409, Aug. 2017.

[4] Y. Liu, J. Li, S. Tong, and C. L. P. Chen, "Neural approximation-based adaptive control for a class of nonlinear nonstrict feedback discrete-time systems," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 28, no. 7, pp. 1531–1541, Jul. 2017.

[5] Y. Liu, S. Tong, C. L. P. Chen, and D. Li, "Adaptive NN control using integral Barrier Lyapunov functionals for uncertain nonlinear block-triangular constraint systems," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3747–3757, Nov. 2017.

[6] D. Guo, Z. Nie, and L. Yan, "Novel discrete-time Zhang neural network for time-varying matrix inversion," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2301–2310, Aug. 2017.

[7] Y. Liu, S. Lu, S. Tong, X. Chen, C. L. P. Chen, and D. Li, "Adaptive control-based Barrier Lyapunov functions for a class of stochastic nonlinear systems with full state constraints," *Automatica.*, vol. 87, pp. 83–93, 2018.

[8] Y. Liu, Q. Zeng, L. Liu, and S. Tong, "An adaptive neural network controller for active suspension systems with hydraulic actuator," *IEEE Trans. Syst., Man, Cybern., Syst.*, In press with DOI 10.1109/TSMC.2018.2875187.

[9] Y. Zhang and S. Li, "Time-scale expansion-based approximated optimal control for underactuated systems using projection neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 11, pp. 1957–1967, 2018.

[10] Y. Zhang and S. Li, "Predictive suboptimal consensus of multiagent systems with nonlinear dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1701–1711, 2017.

[11] Z. Xie, L. Jin, X. Du, X. Xiao, H. Li, and S. Li, "On generalized RMP scheme for redundant robot manipulators aided with dynamic neural networks and nonconvex bound constraints," *IEEE Trans. Ind. Informat.*, In Press with DOI 10.1109/TII.2019.2899909.

[12] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific J. Optim.*, vol. 2, no. 1, pp. 35–58, 2006.

[13] N. Andrei, "An accelerated subspace minimization three-term conjugate gradient algorithm for unconstrained optimization," *Numer. Algor.*, vol. 65, no. 4, pp. 859–874, 2014.

[14] L. Jin and S. Li, "Distributed task allocation of multiple robots: a control perspective," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 693–701, May. 2018.

[15] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 8112–8123, Oct. 2018.

[16] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 6, pp. 1019–1029, Jun. 2017.

[17] Y. Liu and S. Tong, "Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2670–2680, Nov. 2016.

[18] Y. Liu, J. Li, S. Tong, and C. L. P. Chen, "Neural network control based adaptive learning design for nonlinear systems with full state constraints," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 27, no. 7, pp. 1562–1571, Jul. 2016.

[19] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 334–344, Mar. 2016.

[20] H. Lu, L. Jin, X. Luo, B. Liao, D. Guo, and L. Xiao, "RNN for solving perturbed time-varying underdetermined linear system with double bound limits on residual errors and state variables," *IEEE Trans. Ind. Informat.*, In Press with DOI 10.1109/TII.2019.2909142.

[21] L. Jin, S. Li, L. Xiao, R. Lu, and B. Liao, "Cooperative motion generation in a distributed network of redundant robot manipulators with noises," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 10, pp. 1715–1724, Oct. 2018.

[22] L. Xiao, S. Li, K. Li, L. Jin, and B. Liao, "Co-design of finite-time convergence and noise suppression: a unified neural model for time varying linear equations with robotic applications" *IEEE Trans. Syst., Man, Cybern., Syst.*, In press with DOI: 10.1109/TSMC.2018.2870489.

[23] Y. Zhang, Y. Zhang, D. Chen, Z. Xiao, and X. Yan, "From Davidenko method to Zhang dynamics for nonlinear equation systems solving," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 11, pp. 2817–2830, Nov. 2017.

[24] L. Jin, S. Li, B. Hu, M. Liu, and J. Yu, "A Noise-suppressing neural algorithm for solving time-varying system of linear equations: a control-based approach," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 236–246, Jan. 2019.

[25] L. Jin and Y. Zhang, "Continuous and discrete Zhang dynamics for real-time varying nonlinear optimization," *Numer. Algor.*, vol. 73, no. 1, pp. 115–140, 2016.

[26] Y. Liu, S. Tong, C. L. P. Chen, and D. Li, "Neural controller design-based adaptive control for nonlinear MIMO systems with unknown hysteresis inputs," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 9–19, Apr. 2015.

[27] L. Jin, S. Li, B. Liao, and Z. Zhang, "Zeroing neural networks: A survey," *Neurocomputing.*, vol. 267, no. 6, pp. 597–604, 2017.

[28] S. E. Selvan, U. Amato, K. A. Gallivan, C. Qi, M. F. Carfora, and M. Larobina, "Descent algorithms on oblique manifold for source-adaptive ICA contrast," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1930–1947, Dec. 2012.

[29] D. Li and M. Fukushima, "A modified BFGS method and its global convergence in nonconvex minimization," *J. Comput. Appl. Math.*, vol. 129, no. 1, pp. 15–35, Apr. 2001.

[30] L. Jin and Y. Zhang, "Discrete-time Zhang neural network for online time-varying nonlinear optimization with application to manipulator motion generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1525–1531, Jul. 2015.

[31] L. Jin, S. Li, X. Luo, Y. Li, and B. Qin, "Neural Dynamics for Cooperative Control of Redundant Robot Manipulators," *IIEEE Trans. Ind. Informat.*, vol. 14, no.9, pp. 3812–3821, 2018.

[32] Y. Zhang, L. He, C. Hu, J. Guo, J. Li, and Y. Shi, "General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization," *J. Comput. Appl. Math.*, vol. 347, pp. 314–329, Feb. 2019.

[33] B. Qiu, Y. Zhang, J. Guo, Z. Yang, and X. Li, "New five-step DTZD algorithm for future nonlinear minimization with quartic steady-state error pattern," *Numer. Algor.*, In press with DOI: 10.1007/s11075–018–0581–4.

[34] D. Liu, H. Li, and D. Wang, "Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1015–1027, Aug. 2014.

[35] J. Hu, Z. Wu, H. Mccann, L. E. Davis, and C. Xie, "BFGS quasi-Newton method for solving electromagnetic inverse problems," *IEEE Proc. Microw. Antennas Propag.*, vol. 153, no. 2, pp. 199–204, Apr. 2006.

[36] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 3936–3964, Sep. 2006.

[37] C. Yang, J. Luo, Y. Pan, Z. Liu, and C. Y. Su, "Personalized variable gain control with tremor attenuation for robot teleoperation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 10, pp. 1759–1770, Oct. 2018.

[38] C. Yang, X. Wang, Z. Li, Y. Li, and C. Y. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2125–2136, Aug. 2017.

[39] D. Guo and Y. Zhang, "Zhang neural network, Getz-Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots' kinematic control," *Neurocomputing*, vol. 97, pp. 22–32, Nov. 2012.

[40] L. Jin, S. Li, H. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Informat.*, vol. 64, no. 6, pp. 4710–4720, 2017.

Lin Wei received the B.E. degree from the Beijing Institute of Technology, Beijing, China, in 2018. She is currently pursuing her M.E. degree in communication and information systems with School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Her research interests include neural networks and robotics.

Long Jin (M'17) received the B.E. degree in automation and the PhD degree in information and communication engineering from Sun Yat-sen University, Guangzhou, China, in 2011 and 2016, respectively. He is currently a full professor with the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. Before joining Lanzhou University in 2017, he was a Postdoctoral Fellow with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. His main research interests include neural networks, robotics, and intelligent information processing. He is a member of the IEEE.

Chenguang Yang (M'10-SM'16) is a Professor of Robotics. He received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xian, China, in 2005, and the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010. He received postdoctoral training at Imperial College London, UK, from 2009 to 2010. He was awarded Marie Curie International Incoming Fellowship, EPSRC Innovation Fellowship, and the Best Paper Award of the IEEE Transactions on Robotics as well as over ten conference Best Paper Awards.His research interest lies in human robot interaction and intelligent system design.

Ke Chen is currently an associate professor in the South China University of Technology (SCUT), Guangzhou, China. He received his B.E. and M.E. degrees at the Sun Yat-sen University in 2007 and 2009 respectively, and Ph.D degree at the Queen Mary, University of London in 2013. Before he joined in SCUT, he worked at the Tampere University of Technology, Finland for five years. He has published more than 70 papers including top-tier CVPR, ICCV, IJCAI in the field. His research interests include computer vision, pattern recognition, neural dynamics and robotics.

Weibing Li received the B.S. degree in communication engineering from Changchun University, Changchun, China, in 2011, the M.S. degree in detection technology and automatic equipment from the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, in 2014, and the Ph.D. degree in mechanical engineering from the School of Mechanical Engineering, University of Leeds, Leeds, U.K., in 2018. He is currently a Post-Doctoral Researcher with the Chow Yuk Ho Technology Centre for Innovative Medicine, Chinese University of Hong Kong, Hong Kong. His current research interests include industrial robotics, modular robotics, medical robotics, and neural networks.