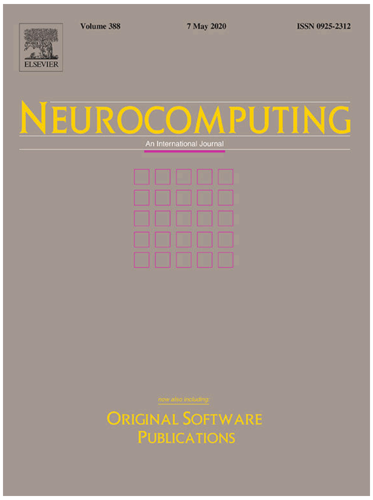


A multi-constrained zeroing neural network for time-dependent nonlinear optimization with application to mobile robot tracking control

Dechao Chen, Xinwei Cao, Shuai Li

PII: S0925-2312(21)01025-0
DOI: https://doi.org/10.1016/j.neucom.2021.06.089
Reference: NEUCOM 24042

To appear in: *Neurocomputing*

Received Date: 13 April 2021
Accepted Date: 28 June 2021

Please cite this article as: D. Chen, X. Cao, S. Li, A multi-constrained zeroing neural network for time-dependent nonlinear optimization with application to mobile robot tracking control, *Neurocomputing* (2021), doi: https://doi.org/10.1016/j.neucom.2021.06.089

A multi-constrained zeroing neural network for time-dependent nonlinear optimization with application to mobile robot tracking control[☆]

Dechao Chen[1], Xinwei Cao[2], Shuai Li[3]

[1]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, P.R. China*

[2]*School of Management, Shanghai University, Shanghai, 200444, P.R. China*

[3]*School of Engineering, Swansea University, Swansea SA1 7EN, United Kingdom*

Abstract

Because of the strong dynamic behavior and computing power, zeroing neural networks (ZNNs) have been developed to handle different time-dependent issues. However, due to the high nonlinearity and complexity, the research on finding a feasible ZNN to address time-dependent nonlinear optimization with multiple types of constraints still remains stagnant. To simultaneously handle multiple types of constraints for the time-dependent nonlinear optimization, this paper proposes a novel neural-network based model in a unified framework of ZNN. By using leveraging the Lagrange method, the time-dependent nonlinear optimization problem with multiple types of constraints is converted to a time-dependent equality system. The proposed multi-constrained ZNN (termed MZNN) inherently possesses the effectiveness of exponential convergence property by utilizing the time-derivative information. Theoretical analyses on the global stability and exponential convergence property are rigorously provided. Then, three general numerical examples in time-independent and time-dependent cases verify the computational performance of the proposed MZNN. An application based on the

[☆] This work is supported by the National Natural Science Foundation of China (with number 61906054), and also by the Natural Science Foundation of Zhejiang Provincial (with number LY21-F030006).

[*] Corresponding authors.

E-mail addresses: chdchao@hdu.edu.cn (D. Chen); shuaili@ieee.org (S. Li).

mobile robot for nonlinear optimization control sufficiently demonstrates the physical effectiveness of the proposed MZNN for the control of mobile robot with both performance-index optimization and multiple physical-limit constraints. Finally, comparisons with existing neural networks such as gradient neural network (GNN), and performance tests with investigation on computational complexity substantiate the superiority of the MZNN for the time-dependent nonlinear optimization subject to multiple types of constraints.

Keywords: Multi-constrained zeroing neural networks (MZNNs),
Time-dependent problem, Nonlinear optimization, Multiple constraints,
Robot control
2010 MSC: 62G35, 92B20, 93B51

1. Introduction

For simplification and also for low requirement of real-time computation, most researchers only focus on handling statical or termed time-independent problems in the past decade [1, 2]. Due to the increasingly containing complexity as well as the requirement of time-dependent calculation [3–5], the investigation on time-dependent problem becomes increasing desirable and significant. Being quite contrary to time-independent problems, it is more tough to handle time-dependent problems because the related problems, coefficients in addition to solutions are time-dependent [6–8]. Traditional statical approaches for the time-independent solution may become invalid because of unignorable time-delay (or termed lagging behind) errors [9, 10]. The conventional statical approaches usually find a solution at present moment on the basis of the present data and conditions. The veritable solution keep variation whereas a local time solution will no longer be a feasible solution [11]. Thus, to investigate effective methods for time-dependent issues is an interesting but also challenging research topic [12–14].

Time-dependent nonlinear optimization becomes common issue in fields of both academic and engineering such as robotic control [8, 14–19], signal process [20], and numerical algorithm [10, 21–23]. Owing to the advantages on the parallel processing and feasibility of hardware implementation, approaches by leveraging neural networks [24–33], especially recurrent neural networks (RNNs), were viewed as powerful approaches for time-dependent nonlinear optimization [34–36]. Particularly, on the basis of the scalar valued energy function, neural-network based approaches by utilizing negative-gradient have been developed and investigated [37, 38].

In contrast with gradient neural networks (GNNs) utilizing scalar-valued energy functions, zeroing neural networks (ZNNs) define an indefinite error function with scalar value (also vector value or matrix valued). In the meantime, the ZNNs fully explore the time-derivative data and conditions for solving time-dependent issues [39–43]. Despite the undeniable success research work in references [39–43] using different kinds neural network to address the time-dependent nonlinear optimization, a tough issue is still remnant in a unified framework of ZNN. Time-dependent nonlinear optimization with multiple types of constraints is hard to be addressed availably because of nonlinearity and complexity.

1.1. Related work

Researchers focus on solving the time-independent problems had a great success many years ago [1, 2]. Hu and Zhang [1] proposed a novel RNN model with the capacity of global convergence for addressing the time-independent convex-quadratic programming problems. Qin *et al.* [2] developed a new and effective one layer RNN to address the complex variable convex optimization with different kinds of constraints.

Because of the significance of time-dependent nonlinear optimization, various studies were reported [44–49]. Miao *et al.* [10] developed ZNNs for solving time-dependent quadratic-optimization issues with applications to the robot motion generation. The survey on the ZNNs is report in [50], which covers state-of-the-art studies on time-dependent problems solving. In addition, Li *et al.* [49] investigated general square-pattern-discretization formulas for the neural networks to handle the future optimization issue subject to linear equality constraint. Chen *et al.* [38] introduced neural network systems on the basis of gradient-direction strategy to solve Lyapunov-matrix issue. However, most GNNs produce the above mentioned time-delay errors, and also exhibit a divergency phenomenon when time-dependent nonlinear optimization is solved [51].

Within the framework of ZNN theory, extensive work on ZNNs has been generalized in terms of dynamic behaviour [52, 53], convergence performance [54], and robustness [55, 56]. Stanimirović *et al.* [52] introduced a so-called hyper power ZNN method to calculate the time-dependent outer-inverse. Guo *et al.* [53] developed a new time-dependent ZNN discrete algorithm for optimization issue. Being deferent from the conventional ZNNs with a

Table 1: Different Solution Properties via Neural Networks for Time-Dependent Optimizations

Neural network	Problem related	Equality constraint	Inequality constraint	Time dependent	Convergence performance
[1]	Quadratic optimization	Yes	Yes	No	Asymptotic convergence
[10]	Quadratic optimization	Yes	No	Yes	Exponential convergence
[21]	Nonlinear optimization	Yes	No	Yes	Exponential convergence
[37]	Quadratic optimization	Yes	Yes	No	Asymptotic convergence
[49]	Nonlinear optimization	Yes	No	Yes	Asymptotic convergence
[57]	Linear optimization	Yes	Yes	No	Asymptotic convergence
[58]	Quadratic optimization	No	No	Yes	Exponential convergence
[59]	Quadratic optimization	Yes	No	Yes	Asymptotic convergence
[60]	Quadratic optimization	Yes	No	Yes	Asymptotic convergence
Ours	**Nonlinear optimization**	**Yes**	**Yes**	**Yes**	**Exponential convergence**

fixed predefined parameters, Li *et al.* [56] designed a variable gain finite time convergent ZNN model to solve the convex quadratic optimization.

1.2. Contributions and organization

In this research, by following a unified framework of ZNNs, this paper develops a neural network (called MZNN) to address time-dependent nonlinear optimization. The proposed neural network is capable of simultaneously handling multiple types of constraints. The optimization issue with multiple types of constraints is skillfully converted to a time-dependent equality system by leveraging Lagrange method. The proposed MZNN becomes effectiveness for exponential-convergence performance by utilizing the time-derivative information. In addition, theoretical analyses are rigorously given. Three illustrated examples in both the time-independent and time-dependent cases verify the computational performance of the proposed model. A real-world mobile robot application on nonlinear optimization and tracking control demonstrates the effectiveness of the proposed MZNN. Finally, comparisons with GNN as well as performance tests with investigation on computational complexity substantiate the advantages of MZNN for time-dependent nonlinear optimization subject to multiple types of constraints. The main contributions are highlighted.

- This research simultaneously handles multiple types of constraints for time-dependent nonlinear optimization in a unified framework of ZNN to propose a neural network. The MZNN makes progresses on ZNN optimization from single type constraint to multiple types of constraints by taking advantages of fully exploitation of time-derivative information.

- The design of the MZNN for nonlinear optimization subject to different types of constraints is illustrated to fill the vacancy of the conventional ZNN for time-dependent nonlinear optimization.
- Theoretical analyses, illustrative examples and robot application substantiate global stability, exponential-convergence performance and physical validity of MZNN for time-dependent solution of nonlinear optimization.

For better illustration, comparisons on different solution properties the proposed MZNN with other existing neural networks on literatures [1, 10, 21, 37, 49, 57–60] are shown in Table 1. The studies in [1, 49] and the proposed MZNN involve the nonlinear optimization issue. However, both the neural-network models in [1] and [49] did not consider inequality type constraint being different from the proposed MZNN. In addition, the proposed MZNN is able to address the optimization issue under different types of constraints being superior to the existing ones in literatures [1, 10, 21, 37, 49, 57–60]. Unlike those in [10, 21, 37, 49, 57, 59, 60] with the solution properties of asymptotic convergence, the proposed MZNN has outstanding convergent performance, that is exponential-convergence performance. Such a convergent performance makes the proposed MZNN address time-dependent nonlinear optimization possessing a fast solution. The above discussions and comparisons are the significance and motivation of the proposed research on this field

The organization of this paper is presented. Firstly, the problem formulation with preliminaries of nonlinear optimization is presented in Section 2. Then, Section 3 illustrates the design process of the neural network in unified framework of ZNN with theoretical analyses provided. In Section 4, numerical examples, robot application, comparisons and tests are provided. Section 5 concludes the paper with future work.

2. Problem formulation

A general mathematical description of nonlinear optimization containing multiple types of constraints could be depicted in real time as

$$\begin{aligned} &\text{min. } f(\mathbf{x}(t),t)\\ &\text{s. t. } g(\mathbf{x}(t)) = A(t)\mathbf{x}(t)+\mathbf{b}(t)=0,\\ &\quad\; h(\mathbf{x}(t)) = C(t)\mathbf{x}(t)+\mathbf{d}(t)\leq 0, \end{aligned} \tag{1}$$

of which $f(\cdot,\cdot) : \mathbb{R}^n \to \mathbb{R}$ denotes an object function for optimization assumed to be second order differentiable. In addition, $f(\cdot,\cdot)$ is convex with respect to the time-dependent state-vector $\mathbf{x}(t) \in \mathbb{R}^n$ in time interval $[0, t_\mathrm{f}] \subseteq [0, +\infty)$. Object function $f(\cdot,\cdot)$ is termed time-dependent convex if, at each t_e, the objective function $f(\mathbf{x}(t_\mathrm{e}), t_\mathrm{e})$ being a common function as well as only related to $\mathbf{x}$, is convex [21]. In addition, $g(\mathbf{x}(t)) \in \mathbb{R}^m$ and $h(\mathbf{x}(t)) \in \mathbb{R}^p$ can be multiple types of constraints such as equality or inequality constraint, respectively, as two different types of constraints, with the rank of $A(t) \in \mathbb{R}^{m\times n}$ being always equal to m, and $\mathbf{b}(t) \in \mathbb{R}^m$, matrix $C(t) \in \mathbb{R}^{p\times n}$ and $\mathbf{d}(t) \in \mathbb{R}^p$. To solve general nonlinear optimization issue is that a solution $\mathbf{x}(t)$ is found which make problem (1) always hold true with time t being greater than or equal to zero.

From the perspective of real-world optimization, a mobile robot usually requires to minimize the velocity norm because of the energy reduction in motion control. For better understanding physical nature of (1), it can be specifically illustrated as a robot time-dependent nonlinear optimization and tracking control with performance-index optimization and physical-limit constraints [55]. Firstly, a performance-index optimization is constructed related to an object function $f(\cdot,\cdot)$ as

$$\text{min. } f(\mathbf{x}(t), t) = \frac{\|\dot{\Theta}(t)\|_\mathrm{E}^2}{2}, \tag{2}$$

where $\mathbf{x}(t) = \dot{\Theta}(t)$ is the vector-valued combined velocity of mobile robot, and $\|\cdot\|_\mathrm{E}$ is Euclidean norm. From the perspective of real-world optimization, a mobile robot usually requires to minimize the velocity norm because the energy consumption is closely related to the robot control signals, i.e., the combined velocity $\dot{\Theta}(t)$. The smaller value of $\dot{\Theta}(t)$, the lower energy consumption of the mobile robot. Therefore, to reduce the energy consumption in robot motion control, the optimization for performance-index (2) is chosen. In addition, physical-limits of robot-signals are considered

$$\dot{\Theta}^- \leq \dot{\Theta}(t) \leq \dot{\Theta}^+, \tag{3}$$

of whice $\dot{\Theta}^-$ is the lower bound and $\dot{\Theta}^+$ is the upper bound of physical-limits of the vector-valued of combine-velocity, respectively. Hence, a relatively complete description with physical essence for nonlinear optimization and tracking control of mobile robots with performance-index optimization and physical-limit constraints is given by

$$\begin{aligned}
&\text{min. } \frac{\|\dot{\Theta}(t)\|_{\mathrm{E}}^2}{2}\\
&\text{s. t. } \mathcal{R}(\vartheta(t), \theta(t))\dot{\Theta}(t) = \dot{\mathbf{r}}_{\mathrm{md}}(t) - \delta\Lambda(\mathbf{r}_{\mathrm{m}}(t) - \mathbf{r}_{\mathrm{md}}(t)),\\
&\qquad \dot{\Theta}^{-} \leq \dot{\Theta}(t) \leq \dot{\Theta}^{+},
\end{aligned} \tag{4}$$

of which matrix $\mathcal{R}(\cdot,\cdot)$ is a matrix-valued integration of the geometrical relation information; $\vartheta(t)$ is a heading angle of mobile base; $\theta(t)$ is the vector-valued joint angle; δ is the predefined parameter; $\Lambda(\cdot)$ is the active function vector. The elements of active function are monotonically-increasing odd functions; $\mathbf{r}_{\mathrm{m}}(t)$ is the actual end-effector three dimensional Cartesian position; and $\mathbf{r}_{\mathrm{m}}(t)$ is the user predefined end-effector three dimensional position vector in Cartesian space [55]. The above problem description (4) is the time-dependent nonlinear optimization and tracking control of mobile robots with physical nature for primal mathematical nonlinear optimization problem (1).

The solution $\mathbf{x}(t)$ of general optimization (1) can be found by defining a Lagrange function [61] as

$$\begin{aligned}
\mathcal{L}(\mathbf{x}(t), \lambda(t), \mu(t), t) &= f(\mathbf{x}(t), t) + \lambda^{\mathrm{T}}(t)g(\mathbf{x}(t)) + \mu^{\mathrm{T}}(t)h(\mathbf{x}(t))\\
&= f(\mathbf{x}(t), t) + \lambda^{\mathrm{T}}(t)(A(t)\mathbf{x}(t) + \mathbf{b}(t))\\
&\quad + \mu^{\mathrm{T}}(t)(C(t)\mathbf{x}(t) + \mathbf{d}(t)),
\end{aligned} \tag{5}$$

where variables $\lambda(t) = [\lambda_1(t), \lambda_2(t), \cdots, \lambda_m(t)]^{\mathrm{T}} \in \mathbb{R}^m$ as well as $\mu(t) = [\mu_1(t), \mu_2(t), \cdots, \mu_p(t)]^{\mathrm{T}} \in \mathbb{R}^p$ are Lagrange multipliers related to multiple types of constraints such as equality or inequality. The superscript $^{\mathrm{T}}$ is a transpose operation. From Lagrange method [61], the solution $\mathbf{x}(t)$ to general optimization (1) is identical to the solution to the set of equations depicted

$$\begin{cases}
\frac{\partial \mathcal{L}(\mathbf{x}(t), \lambda(t), \mu(t), t)}{\partial \mathbf{x}} = 0,\\
g(\mathbf{x}(t)) = 0,\\
h(\mathbf{x}(t)) \leq 0, \mu(t) \geq 0, \text{ and } \mu^{\mathrm{T}}(t)h(\mathbf{x}(t)) = 0,
\end{cases} \tag{6}$$

which is further rewritten

$$\begin{cases}
\frac{\partial f(\mathbf{x}(t), t)}{\partial \mathbf{x}} + A^{\mathrm{T}}(t)\lambda(t) + \left(\frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\right)^{\mathrm{T}} \mu(t) = 0,\\
g(\mathbf{x}(t)) = 0,\\
h(\mathbf{x}(t)) \leq 0, \mu(t) \geq 0, \text{ and } \mu^{\mathrm{T}}(t)h(\mathbf{x}(t)) = 0.
\end{cases} \tag{7}$$

To ensure the existence of solution as well as to be optimal for optimization (1) the following theoretical basis are provided.

Lemma 1. *[62]: If and only if two vectors $\lambda^*(t) \in \mathbb{R}^m$ and $\mu^*(t) \in \mathbb{R}^p$ exist such that the integrated vector $\mathbf{y}^*(t) = [\mathbf{x}^{*T}(t), \lambda^{*T}(t), \mu^{*T}(t)]^T \in \mathbb{R}^{n+m+p}$ satisfies a Karush-Kuhn-Tucker (KKT) condition as*

$$\begin{cases} \frac{\partial f(\mathbf{x}(t),t)}{\partial \mathbf{x}}|_{\mathbf{x}(t)=\mathbf{x}^*(t)} + A^T(t)\lambda^*(t) + C^T(t)\mu^*(t) = 0, \\ A(t)\mathbf{x}^*(t) + \mathbf{b}(t) = 0, \\ C(t)\mathbf{x}^*(t) + \mathbf{d}(t) \leq 0, \mu^*(t) \geq 0, \\ \mu^{*T}(t)(C(t)\mathbf{x}^*(t) + \mathbf{d}(t)) = 0, \end{cases} \tag{8}$$

vector $\mathbf{x}^*(t) \in$ is thus optimal solution for optimization (1).

Proof. Generalized from [62]. □

Lemma 2. *[63]: Provided that object function $f(\mathbf{x}(t), t)$ is time-dependent convex at each t_e with the domain of $\mathbf{x}(t_e)$ denoted as Ω_e being a convex set for each t_e for $\mathbf{x}_1(t_e)$, and $\mathbf{x}_2(t_e)$ in domain and $0 \leq \nu \leq 1$ for object function satisfying convexity inequality as*

$$\begin{aligned} &f(\nu\mathbf{x}_1(t_e) + (1-\nu)\mathbf{x}_2(t_e), t_e) \\ &\leq \nu f(\mathbf{x}_1(t_e), t_e) + (1-\nu) f(\mathbf{x}_2(t_e), t_e), \end{aligned} \tag{9}$$

for any two points $\mathbf{x}_1(t_e)$ and $\mathbf{x}_2(t_e)$ in domain Ω_e, and the line segment belonging to Ω_e, that is, $\omega\mathbf{x}_1(t_e) + (1-\omega)\mathbf{x}_2(t_e) \in \Omega_e$ for all $0 \leq \omega \leq 1$, then $\mathbf{x}^*(t)$ is an optimal solution to optimization (1) if and only if $\mathbf{x}^*(t)$ is a KKT point.

Proof. Generalized from [63]. □

The following lemma is the preliminary for the conversion of optimization (1) to an equality system in a unified design framework of ZNN.

Lemma 3. *Solving the set of equations (7) for optimization (1) is identical to solving a set of equation as*

$$\begin{cases} \frac{\partial f(\mathbf{x}(t),t)}{\partial \mathbf{x}} + A^T(t)\lambda(t) + \left(\frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\right)^T \mu(t) = 0, \\ g(\mathbf{x}(t)) = 0, \\ \Upsilon^+(-h(\mathbf{x}(t)) - \mu(t)) = -h(\mathbf{x}(t)), \end{cases} \tag{10}$$

where the ith element of function mapping $\Upsilon^{+}(\cdot):\mathbb{R}^{p}\rightarrow\mathbb{R}^{p}$ *is given by*

$$\Upsilon_{i}^{+}(v_i(t))=\begin{cases}v_i(t), & \textit{if}\ \ v_i(t)\geq 0,\\ 0, & \textit{if}\ \ v_i(t)<0,\end{cases}\tag{11}$$

of which $\mathbf{v}(t)\in\mathbb{R}^{p}$ *is a vector.*

Proof. One can prove that solving

$$h(\mathbf{x}(t))\leq 0,\mu(t)\geq 0,\ \text{and}\ \mu^{\mathrm{T}}(t)h(\mathbf{x}(t))=0,\tag{12}$$

is identical to solving

$$\Upsilon^{+}(-h(\mathbf{x}(t))-\mu(t))=-h(\mathbf{x}(t)),\tag{13}$$

which is segmented into two parts.

Part I (Sufficiency): Denote vectors $h(\mathbf{x}(t))$ and $\mu(t)$ as

$$h(\mathbf{x}(t))=\begin{bmatrix}h_1(\mathbf{x}(t))\\ h_2(\mathbf{x}(t))\\ \vdots\\ h_p(\mathbf{x}(t))\end{bmatrix},\ \text{and}\ \mu(t)=\begin{bmatrix}\mu_1(t)\\ \mu_2(t)\\ \vdots\\ \mu_p(t)\end{bmatrix},$$

with $i=1,2,\cdots,p$. Assumed that $h_i(\mathbf{x}(t))\leq 0$, $\mu_i(t)\geq 0$ together with $\sum_{i=1}^{p}\mu_i(t)h_i(\mathbf{x}(t))=0$, it is obtained

$$\mu_i(t)h_i(\mathbf{x}(t))\leq 0,$$

which is further obtained

$$\sum_{i=1}^{p}\mu_i(t)h_i(\mathbf{x}(t))\leq 0.$$

It can be found that $\sum_{i=1}^{p}\mu_i(t)h_i(\mathbf{x}(t))=0$ holds true, which thus yields $\mu_i(t)h_i(\mathbf{x}(t))=0$, otherwise $\sum_{i=1}^{p}\mu_i(t)h_i(\mathbf{x}(t))<0$. Hence, it obtains $\mu_i(t)=0$ or $h_i(\mathbf{x}(t))$ for all i. *Situation i):* If $\mu_i(t)=0$, then

$$\Upsilon_{i}^{+}(-h_i(\mathbf{x}(t))-\mu_i(t))=\Upsilon_{i}^{+}(-h_i(\mathbf{x}(t)))=-h_i(\mathbf{x}(t))$$

with $-h_i(\mathbf{x}(t)) \geq 0$. It makes that (13) could be derived from (12). *Situation ii):* If $h_i(\mathbf{x}(t)) = 0$, then

$$\Upsilon_i^+(-h_i(\mathbf{x}(t)) - \mu_i(t)) = \Upsilon_i^+(-\mu_i(t)) = -h_i(\mathbf{x}(t)) = 0$$

with $-\mu_i(t) \leq 0$. It also makes that (13) could be deduced from (12).

Part II (Necessity): Assumed that $\Upsilon^+(-h(\mathbf{x}(t)) - \mu(t)) = -h(\mathbf{x}(t))$, and the elements $\Upsilon_i^+(-h_i(\mathbf{x}(t)) - \mu_i(t)) = -h_i(\mathbf{x}(t))$ always hold true, it has $-h_i(\mathbf{x}(t)) \geq 0$. It obtains $h_i(\mathbf{x}(t)) \leq 0$. *Situation i):* If $-h_i(\mathbf{x}(t)) - \mu_i(t) \geq 0$, then

$$-h_i(\mathbf{x}(t)) - \mu_i(t) = -h_i(\mathbf{x}(t))$$

with $\Upsilon_i^+(-h_i(\mathbf{x}(t)) - \mu_i(t)) = -h_i(\mathbf{x}(t))$. Hence, it obtains $\mu_i(t) = 0$ as well as $h_i(\mathbf{x}(t)) \leq 0$ with $-h_i(\mathbf{x}(t)) - \mu_i(t) \geq$. *Situation ii:* If $-h_i(\mathbf{x}(t)) - \mu_i(t) \leq 0$, then

$$\Upsilon_i^+(-h_i(\mathbf{x}(t)) - \mu_i(t)) = -h_i(\mathbf{x}(t)) = 0,$$

that yields $\mu_i(t) \geq 0$ with $-h_i(\mathbf{x}(t)) - \mu_i(t) \leq 0$. By summing up both two situations, it has

$$\mu_i(t) = 0, \text{ and } h_i(\mathbf{x}(t)) \leq 0,$$

and $\mu_i(t)h_i(\mathbf{x}(t)) = 0$, or

$$\mu_i(t) \geq 0, \text{ and } h_i(\mathbf{x}(t)) = 0,$$

and further obtains $\mu_i(t)h_i(\mathbf{x}(t)) = 0$. Hence, it has $\sum_{i=1}^{p} \mu_i(t)h_i(\mathbf{x}(t)) = 0$ and $\mu^{\mathrm{T}}(t)h(\mathbf{x}(t)) = 0$ with $h(\mathbf{x}(t)) \leq 0$ and $\mu(t) \geq 0$. The proof is thus completed. □

3. MZNN design in ZNN unified framework

In this section, a novel neural network, termed MZNN, in ZNN design framework is proposed. In addition, theoretical analyses on stability as well as convergence performance are presented.

3.1. MZNN model design

Firstly, for time-dependent nonlinear optimization, in ZNN design framework [42, 43], an indefinite error function in vector form can be defined on the basis of Lemma 3 as

$$\mathbf{e}(t) = \begin{bmatrix} \frac{\partial f(\mathbf{x}(t),t)}{\partial \mathbf{x}} + A^{\mathrm{T}}(t)\lambda(t) + \left(\frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\right)^{\mathrm{T}} \mu(t) \\ -g(\mathbf{x}(t)) \\ -\Upsilon^+(-h(\mathbf{x}(t)) - \mu(t)) - h(\mathbf{x}(t)) \end{bmatrix} \tag{14}$$

with $\mathbf{e}(t) \in \mathbb{R}^{n+m+p}$. Note system structure (1) is the problem to be solved. The error function (14) is an intermediate function that is utilized to develop the neural network and find the solution. According to the Leibniz formula [64], it has a product rule of derivative as

$$\frac{\mathrm{d}(u(t)v(t))}{\mathrm{d}t} = \frac{\mathrm{d}u(t)}{\mathrm{d}t}v(t) + u(t)\frac{\mathrm{d}v(t)}{\mathrm{d}t}$$

with $u(t)$ and $v(t)$ being two time-involved variables. Afterwards, it has

$$\frac{\mathrm{d}((\partial h(\mathbf{x}(t))/\partial \mathbf{x})^{\mathrm{T}}\mu(t))}{\mathrm{d}t} = \left(\frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\right)^{\mathrm{T}} \dot{\mu}(t) + \frac{\mathrm{d}^{\mathrm{T}}(\partial h(\mathbf{x}(t))/\partial \mathbf{x})}{\mathrm{d}t}\mu(t).$$

It can be found that

$$\frac{\mathrm{d}(\partial h(\mathbf{x}(t))/\partial \mathbf{x})}{\mathrm{d}t} = \sum_{i=1}^{p} \frac{\partial^2 h(\mathbf{x}(t))}{\partial \mathbf{x}\partial \mathbf{x}_i}\dot{\mathbf{x}}_i(t),$$

holds true. Hence, one obtains

$$\begin{aligned}\frac{\mathrm{d}^{\mathrm{T}}(\partial h(\mathbf{x}(t))/\partial \mathbf{x})}{\mathrm{d}t}\mu(t) &= \sum_{i=1}^{p}\left(\frac{\partial^2 h(\mathbf{x}(t))}{\partial \mathbf{x}\partial \mathbf{x}_i}\dot{\mathbf{x}}_i(t)\mu(t)\right)\\ &= \sum_{i=1}^{p}\dot{\mathbf{x}}_i(t)\left(\frac{\partial^2 h(\mathbf{x}(t))}{\partial \mathbf{x}\partial \mathbf{x}_i}\right)\mu(t).\end{aligned}$$

Make a variable substitution

$$\left(\frac{\partial^2 h(\mathbf{x}(t))}{\partial \mathbf{x}\partial x_i}\right)\mu(t) = \varphi_i(t),$$

with

$$\Psi(t) = [\varphi_1(t), \varphi_2(t), \cdots, \varphi_i(t), \cdots, \varphi_n(t)].$$

Thus, it has

$$\frac{\mathrm{d}^{\mathrm{T}}(\partial h(\mathbf{x}(t))/\partial \mathbf{x})}{\mathrm{d}t}\mu(t) = \sum_{i=1}^{p}\dot{\mathbf{x}}_i(t)\varphi_i(t) = \Psi(t)\dot{\mathbf{x}}(t),$$

which yields

$$\frac{\mathrm{d}((\partial h(\mathbf{x}(t))/\partial \mathbf{x})^{\mathrm{T}}\mu(t))}{\mathrm{d}t} = \Psi(t)\dot{\mathbf{x}}(t) + \left(\frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\right)^{\mathrm{T}}\dot{\mu}(t).$$

On the basis of the above substitution, time-derivative of (14) is obtained

$$\dot{\mathbf{e}}(t) = \left[\dot{\mathbf{e}}_1(t), \dot{\mathbf{e}}_2(t), \dot{\mathbf{e}}_3(t)\right]^{\mathrm{T}}$$

with $\dot{\mathbf{e}}_1(t) \in \mathbb{R}^n$, $\dot{\mathbf{e}}_2(t) \in \mathbb{R}^m$ and $\dot{\mathbf{e}}_3(t) \in \mathbb{R}^p$ being defined by

$$\begin{aligned}
\dot{\mathbf{e}}_1(t) &= \frac{\partial^2 f(\mathbf{x}(t), t)}{\partial \mathbf{x}^2}\dot{\mathbf{x}}(t) + \dot{A}^{\mathrm{T}}(t)\lambda(t) + A^{\mathrm{T}}(t)\dot{\lambda}(t) \\
&\quad + \Psi(t)\dot{\mathbf{x}}(t) + \left(\frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\right)^{\mathrm{T}} \dot{\mu}(t), \\
\dot{\mathbf{e}}_2(t) &= -A(t)\dot{\mathbf{x}}(t) - \dot{A}(t)\mathbf{x}(t) - \dot{\mathbf{b}}(t), \\
\dot{\mathbf{e}}_3(t) &= -\Phi(t)\Upsilon^{-}\left(-\frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) - \dot{\mu}(t)\right) \\
&\quad - \frac{\partial h(\mathbf{x}(t))}{\partial \mathbf{x}}\dot{\mathbf{x}}(t),
\end{aligned}$$

of which $\Phi(t) \in \mathbb{R}^{p\times p}$ is given by

$$\Phi(t) = \mathrm{diag}(\phi(-h(\mathbf{x}(t)) - \mu(t))),$$

where $\mathrm{diag}(\mathbf{v}) : \mathbb{R}^p \to \mathbb{R}^{p\times p}$ is to generate a $p \times p$ dimensional-square matrix. Elements of vector $\mathbf{v} \in \mathbb{R}^p$ are on the diagonal. In addition, elements of $\phi(\cdot) : \mathbb{R}^p \to \mathbb{R}^p$ are denoted

$$\phi_i(v_i(t)) = \begin{cases} 0, & \text{if} \quad v_i(t) \leq 0, \\ 1, & \text{if} \quad v_i(t) > 0. \end{cases}$$

The ith element of function mapping $\Upsilon^{-}(\cdot) : \mathbb{R}^p \to \mathbb{R}^p$ is given by

$$\Upsilon_i^{-}(v_i(t)) = \begin{cases} -v_i(t), & \text{if} \quad v_i(t) \leq 0, \\ 0, & \text{if} \quad v_i(t) > 0. \end{cases}$$

In the unified ZNN design framework, it can be applied the ZNN design formula [42, 43] as

$$\dot{\mathbf{e}}(t) = -\zeta\Gamma(\mathbf{e}(t)),$$

with ζ denoting a ZNN predefined parameter for practitioners to adjust the convergence rate. In addition, $\Gamma(\cdot) : \mathbb{R}^{n+m+p} \to \mathbb{R}^{n+m+p}$ is an active function and its elements are all monotonically-increasing odd functions. A novel

MZNN with dynamical expression for addressing the time-dependent nonlinear optimization (1) under multiple types of constraints is proposed

$$\begin{bmatrix} Q(t) & A^{\mathrm{T}}(t) & C^{\mathrm{T}}(t) \\ -A(t) & 0 & 0 \\ M(t) & 0 & \Phi(t) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\lambda}(t) \\ \dot{\mu}(t) \end{bmatrix} \\ = -\zeta\Gamma\left(\begin{bmatrix} \mathbf{e}_1(t) \\ \mathbf{e}_2(t) \\ \mathbf{e}_3(t) \end{bmatrix}\right) - \begin{bmatrix} \mathbf{r}_1(t) \\ \mathbf{r}_2(t) \\ \mathbf{r}_3(t) \end{bmatrix}, \tag{15}$$

where is rewritten in a compact matrix form

$$W(t)\dot{\mathbf{y}}(t) = -\zeta\Gamma(\mathbf{e}(t)) - \mathbf{r}(t), \tag{16}$$

in which $W(t)$ and $\dot{\mathbf{y}}(t)$ are given by

$$W(t) = \begin{bmatrix} Q(t) & A^{\mathrm{T}}(t) & C^{\mathrm{T}}(t) \\ -A(t) & 0 & 0 \\ M(t) & 0 & \Phi(t) \end{bmatrix}, \ \dot{\mathbf{y}}(t) = \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\lambda}(t) \\ \dot{\mu}(t) \end{bmatrix}$$

with $Q(t) = \partial^2 f(\mathbf{x}(t), t)/\partial \mathbf{x}^2 + \Psi(t)$, $M(t) = \Phi(t)C(t) - C(t)$, $\mathbf{r}_1(t) = \dot{A}^{\mathrm{T}}(t)\lambda(t)$, $\mathbf{r}_2(t) = -\dot{A}(t)\mathbf{x}(t) - \dot{\mathbf{b}}(t)$, $\mathbf{r}_3(t) = 0$ and $\mathbf{r}(t) = [\mathbf{r}_1(t), \mathbf{r}_2(t), \mathbf{r}_3(t)]^{\mathrm{T}}$.

Remark 1. *The parameter ζ is a pivotal parameter of MZNN (16). It can be predefined by the users and developers. In theoretical, any values that satisfy the condition $\zeta > 0 \in \mathbb{R}$ could be viable. In the perspective of convergence performance, the value of user-predefined parameter ζ could be set as appropriately large enough as the hardware that permits in engineering [65].*

The proposed MZNN (16) is in compact matrix form containing implicit output $\mathbf{y}(t)$. Such compact matrix form (16) of the proposed MZNN can not be readily implemented and realized by hardware units. To make MZNN (16) more computable, it can be equivalently written as

$$W(t)\dot{\mathbf{y}}(t) = -(\zeta\Gamma(\mathbf{e}(t)) + \mathbf{r}(t)).$$

The above equation can be pre-multiplication both term using pseudo-inverse matrix of $W(t)$, i.e., $W^{\dagger}(t)$. Hence, the MZNN is reformulated as the following explicit and computable expression

$$\dot{\mathbf{y}}(t) = -W^{\dagger}(t)(\zeta\Gamma(\mathbf{e}(t)) + \mathbf{r}(t)), \tag{17}$$

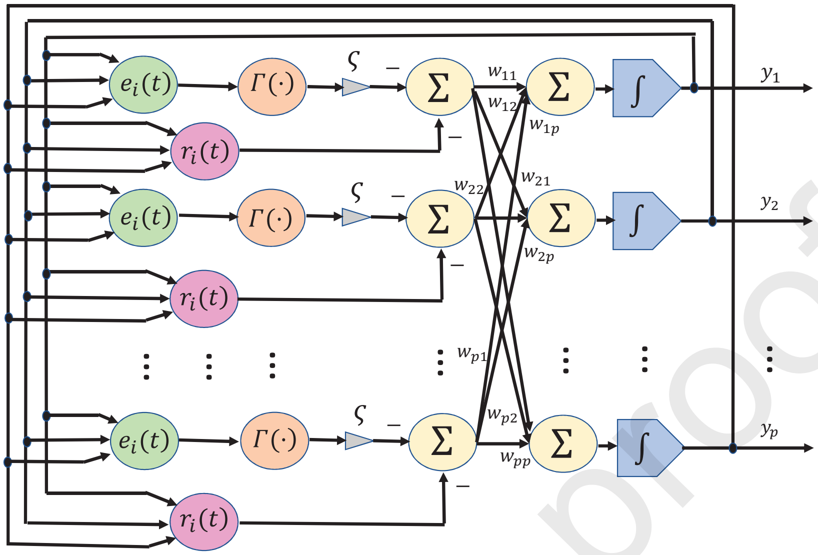


Figure 1: Neural network structure with the neurons in the proposed MZNN (16) for solution to optimization (1) subject to multiple types of constraints.

which the ith $(i = 1, 2, \cdots, n+m+p)$ neuron of neural network is given by

$$y_i = \int \sum_{j=1}^{p} w_{ij}(-\zeta\Gamma(e_i(t)) - r_i(t))\mathrm{d}t, \tag{18}$$

of which y_i is the ith neuron of MZNN (16), and w_{ij} is the ijth element of matrix $W^{\dagger}(t)$.

A neural network structure of the proposed MZNN (16) for handling optimization (1) subject to multiple types of constraints is presented in Fig. 1. The initial state $\mathbf{y}(0)$ of vector $\mathbf{y}(t)$ is the input of MZNN (16), and y_i is the ith neuron of MZNN (16) with its steady state being the output of the MZNN for addressing time-dependent nonlinear optimization. In addition, the ijth element of matrix $W^{\dagger}(t)$, i.e., w_{ij}, is the time-dependent weight of MZNN (16). The neuron output y_i contains the feedback information for MZNN (16). Because there exist feedback connections among neurons, the proposed MZNN (16) is a typical kind of RNNs taking full advantages of the time-derivative information of parameters. Other operations including multiplications and integrator could be realized by utilizing logarithmic amplifiers

Algorithm 1 Design of MZNN (16) for Time-Dependent Nonlinear Optimization

```
Input: Predefined ZNN parameter, i.e., $\zeta$;
Input: Activation-function vector mapping, i.e., $\Lambda(\cdot)$;
  Initialize: Neural network state vector, i.e., $\mathbf{y}(0)$;
  Formulate: Time-dependent object function, i.e., $f(\mathbf{x}(t), t)$;
  Formulate: Time-dependent equality constraint, i.e., $g(\mathbf{x}(t))$;
  Formulate: Time-dependent inequality constraint, i.e., $h(\mathbf{x}(t))$;
  Set: Neural network evolution duration $T_\mathrm{d}$, sampling period $\tau$;
  if $t \leq T_\mathrm{d}$ then
    Calculate: Time-dependent weight matrix $W(t)$;
      Calculate: Time-dependent coefficient vectors $\mathbf{e}(t)$ and $\mathbf{r}(t)$;
      Calculate: Time-derivative of state vector $\dot{\mathbf{y}}(t)$ via dynamical
      equation (17);
      Update: Neural network state vector $\mathbf{y}(t)$ in next iteration by equation
      $\mathbf{y}(t+\tau) = \mathbf{y}(t) + \tau\dot{\mathbf{y}}(t)$;
    Output: Next iteration neural network state $\mathbf{y}(t+\tau)$;
  else if $t > T_\mathrm{d}$ then
    Stop: Neural network evolution process, calculation finished;
  end if
```

[66]. Provided that involved operations could be achieved by applying operational amplifiers, a hardware version of the proposed MZNN (16) would be readily obtained. Moreover, to clarify the processing steps of the proposed MZNN (16) in a clear manner, an algorithm description is presented in Algorithm 1.

3.2. Theoretical analyses

To show global stability and outstanding convergence performance of the proposed MZNN (16), theoretical analyses are given. As theoretical basis, literatures [16, 42, 43] might be used for better understanding the preliminaries.

Theorem 1. *(Global Stability and Convergence of MZNN): As for optimization (1) is under multiple types of constraints. Assumed that a positive user-predefined parameter $\zeta > 0$ and an odd as well as monotone increasing active function $\Gamma(\cdot)$ are applied, from an arbitrary neural-network state $\mathbf{y}(0)$, the closed-loop MZNN (16) is globally stable with the first n elements of state $\mathbf{y}(t)$ converging to an exact time-dependent solution $\mathbf{x}^*(t)$ of (1) in the sense of Lyapunov.*

Proof. As for (1) subject to multiple types of constraints, a neurodynamical equation of MZNN (16) can be given by

$$\dot{\mathbf{e}}(t) = -\zeta\Gamma(\mathbf{e}(t)), \tag{19}$$

and the ith sub-element of (19) is obtained

$$\dot{e}_i(t) = -\zeta\Gamma(e_i(t)), \tag{20}$$

of which user-predefined parameter $\zeta > 0$, $\Gamma(\cdot)$ is an odd as well as monotone increasing active function and with index $i = 1, 2, \cdots, p$. Define a Lyapunov function candidate

$$\mathcal{L}(t) = \frac{e_i^2(t)}{2}. \tag{21}$$

Because of $\mathcal{L}(t) > 0$ for $e_i(t) \neq 0$, and $\mathcal{L}(t) = 0$ for $e_i(t) = 0$ only, therefore, $\mathcal{L}(t)$ becomes positive definite. After that, the time-derivative of $\mathcal{L}(t)$ can be computed

$$\dot{\mathcal{L}}(t) = \frac{\mathrm{d}\mathcal{L}(t)}{\mathrm{d}t} = e_i(t)\dot{e}_i(t) = -\eta e_i(t)\Gamma(e_i(t)).$$

Because $\Gamma(\cdot)$ is an odd as well as monotone increasing active function, it has

$$-\Gamma(e_i(t)) = \Gamma(-e_i(t)),$$

that it obtains

$$-\eta e_i(t)\Gamma(e_i(t)) \begin{cases} < 0, & \text{if } e_i(t) \neq 0, \\ = 0, & \text{if } e_i(t) = 0. \end{cases}$$

Hence, it asserts the result that $\dot{\mathcal{L}}(t)$ is negative definite when $t \in [0, +\infty)$ and user-predefined parameter $\zeta > 0$ both hold true. According to Lyapunov stability theory [16], MZNN (16) is proven to be globally stable and its all elements of error function $e_i(t)$ are globally convergent to 0. In other words, the first n elements of $\mathbf{y}(t)$ is convergent to an time-dependent solution $\mathbf{x}^*(t)$ of optimization (1). The proof is completed. □

Theorem 2. *(Exponential Convergence Performance of MZNN): As for optimization (1) is under multiple types of constraints. Assumed that a positive user-predefined parameter $\zeta > 0$ and a linear-type active function, i.e., $\Gamma(e_i(t)) = e_i(t)$, are utilized, from an arbitrary state $\mathbf{y}(0)$, the first n elements of $\mathbf{y}(t)$ of the proposed MZNN (16) is exponentially convergent to a time-dependent solution $\mathbf{x}^*(t)$ of (1).*

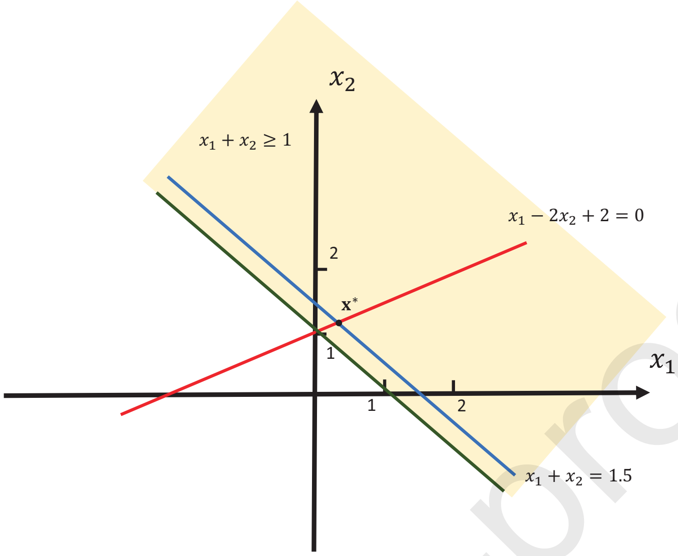


Figure 2: Region and theoretical solution to time-independent nonlinear optimization (22).

Proof. Let us review the ith sub-system of (19):

$$\dot{e}_i(t) = -\zeta\Gamma(e_i(t)),$$

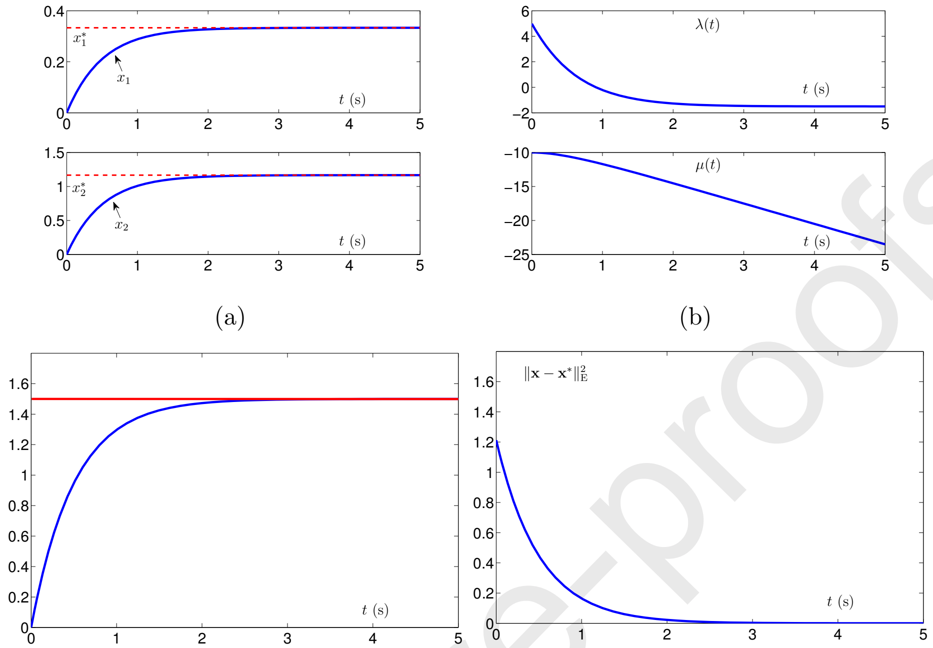
where a linear-type active function, i.e., $\Gamma(e_i(t)) = e_i(t)$ is utilized. It has

$$\dot{e}_i(t) = -\zeta e_i(t),$$

that an analytical solution is acquired

$$e_i(t) = e_i(0)\exp(-\zeta t).$$

Equation (5) points out that all elements of $\mathbf{e}(t)$ are exponentially convergent to zero. In addition, the convergence rate is related the user-predefined parameter ζ for the proposed MZNN (16). The first n elements of $\mathbf{y}(t)$ of MZNN (16) is exponentially convergent to a time-dependent solution $\mathbf{x}^*(t)$ of (1). The proof is completed. □.



(a) (b)

(c) (d)

Figure 3: Transient states and solutions synthesized by the proposed MZNN (16) for handling time-independent nonlinear optimization (22). (a) Neural network states $\mathbf{x}$ and solutions $\mathbf{x}^*$. (b) Neural network states λ and μ. (c) Profiles of equality constraint $g(\mathbf{x}) = x_1 + x_2 = 1.5$. (d) Profile of residual error $\|\mathbf{x} - \mathbf{x}^*\|_{\mathrm{E}}^2$.

4. Verifications, comparisons and tests

Three illustrative examples and real-world mobile robot application by utilizing the proposed MZNN (16) to address the time-independent nonlinear optimization (1) subject to multiple types of constraints in both the time-independent and time-dependent cases are provided. In addition, comparisons with other neural networks such as GNN model are conducted. Finally, tests under different initial neural network states are provided.

4.1. Example 1

Firstly, consider time-independent nonlinear optimization under multiple types of constraints

$$\begin{aligned} &\text{min. } x_1^2 + 4x_2^2 + 4x_1 - 8x_2 - 4x_1x_2 + 4 \\ &\text{s. t. } x_1 + x_2 = 1.5, \\ &\qquad x_1 + x_2 \geq 1. \end{aligned} \tag{22}$$

Problem (22) is rewritten in a matrix form of nonlinear optimization involving time-independent coefficients

$$\begin{aligned} f(\mathbf{x}) &= x_1^2 + 4x_2^2 + 4x_1 - 8x_2 - 4x_1x_2 + 4, \\ \mathbf{x} &= \left[x_1, x_2\right]^{\mathrm{T}}, \ A = \left[1, 1\right], \ \mathbf{b} = \left[-1.5\right], \\ C &= \left[-1, -1\right], \ \mathbf{d} = \left[1\right], \end{aligned}$$

For the convenience of the presentation, the region of the time-independent nonlinear optimization problem (22) is illustrated in Fig. 2. The object function $f(\mathbf{x})$ in this example is a typical quadratic nonlinear function respective to x_1 and x_2. Problem (22) is a typical time-independent nonlinear optimization problem. Therefore, the illustrative results of Fig. 2 is the case of time-independent nonlinear optimization. The linear equations $x_1 - 2x_2 + 2 = 0$ and $x_1 + x_2 = 1.5$ in Fig. 2 is illustrated to find the region and theoretical solution $\mathbf{x}^*$. Note that the time-independent nonlinear optimization (22) is under multiple types of constraints as $x_1 + x_2 >= 1$ (inequality constraint) and $x_1 + x_2 = 1.5$ (equality constraint). As one can readily find in Fig. 2, the unique optimal solution (or to say, theoretical solution) $\mathbf{x}^* = [0.3333, 1.1667]^{\mathrm{T}}$ for the nonlinear optimization problem (22) exist within the region formed by constraints $x_1 + x_2 >= 1$ (illustrated via yellow) and $x_1 + x_2 = 1.5$ (illustrated via blue). In other words, the constraints equations $x_1 + x_2 = 1.5$ and $x_1 + x_2 >= 1$ contain intersection and are reasonable. The illustration can be used to examine and compare solution generated by MZNN (16). The initial states vector is set as $[0, 0, 5, -10]^{\mathrm{T}}$. The duration of the solving process is set as $T_{\mathrm{d}} = 5$ s. In addition, the parameter is set as $\zeta = 2$. A linear activation function $\Gamma(\cdot)$ is explored. The corresponding transient states and solutions synthesized by the proposed MZNN (16) for (22) are illustrated in Fig. 3. Firstly, Fig. 3(a) illustrates that neural states $\mathbf{x}$ of the proposed MZNN (16) converge to the theoretical solution $\mathbf{x}^* = [0.3333, 1.1667]^{\mathrm{T}}$ with exponential convergence rate. Other neural network states λ and μ are presented in Fig.

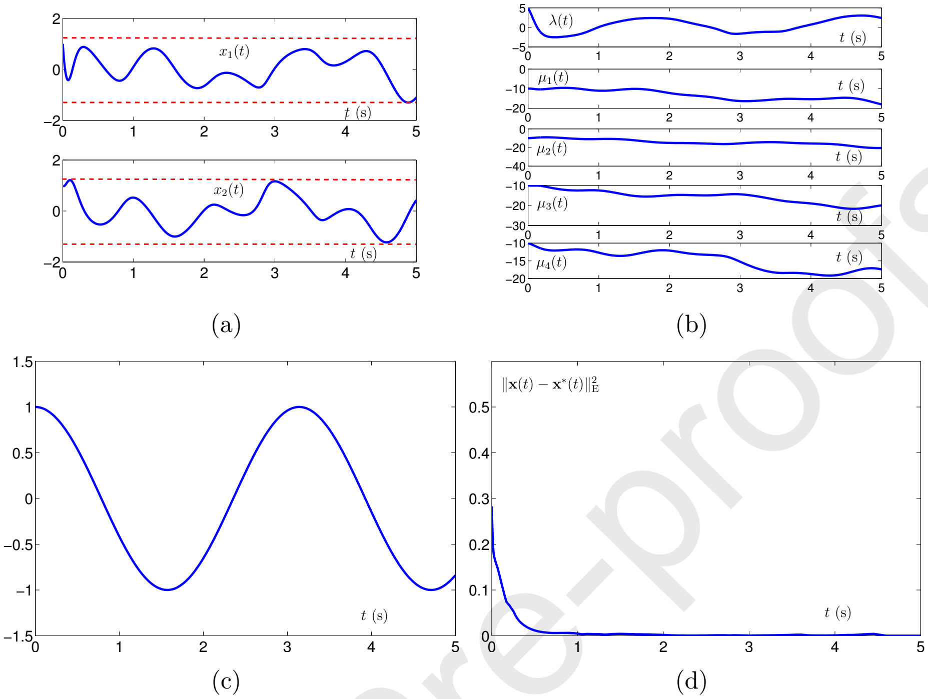
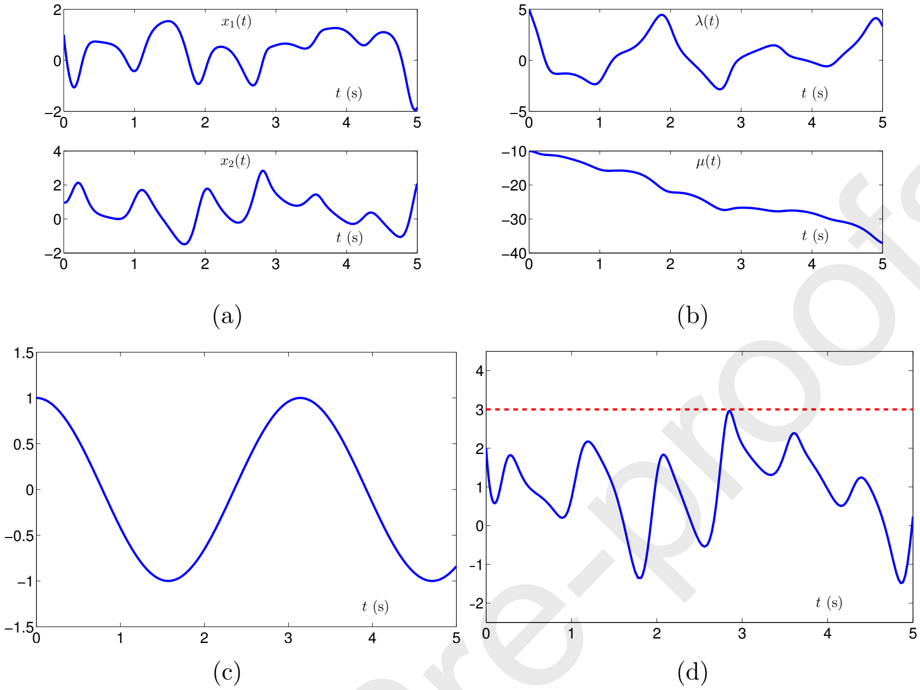


Figure 4: Transient states and solutions synthesized by the proposed MZNN (16) for (23). (a) Neural network states $\mathbf{x}(t)$ with inequality constraint $-1.3 \le x_1(t),\ x_2(t) \le 1.3$. (b) Neural network states $\lambda(t)$ and $\mu(t)$. (c) Profiles of equality constraint $g(\mathbf{x}(t)) = \sin(4t)x_1(t) + \cos(4t)x_2(t) = \cos(2t)$. (d) Profile of residual error $\|\mathbf{x}(t) - \mathbf{x}^*(t)\|_{\mathrm{E}}^2$.

3(b). As shown in Fig. 3(c), it can be found that the profiles of $g(\mathbf{x})$ complies with the equality constraint $x_1 + x_2 = 1.5$. The residual errors synthesized by the proposed MZNN (16) during the solving processes shown in Fig. 3(d) converge to zero within around 2.5 s illustrating the exponential convergence property. The above numerical results imply that the steady-state solution $\mathbf{x}$ illustrated in Fig. 3 is an optimal solution to (22) subject to multiple types of constraints in the time-independent case.



(a)

(b)

(c)

(d)

Figure 5: Transient states and solutions synthesized by the proposed MZNN (16) for (24) with another inequality constraint. (a) Neural network states $\mathbf{x}(t)$. (b) Neural network states $\lambda(t)$ and $\mu(t)$. (c) Profiles of equality constraint $g(\mathbf{x}(t)) = \sin(4t)x_1(t) + \cos(4t)x_2(t) = \cos(2t)$. (d) Profile of inequality constraint $h(\mathbf{x}(t)) = x_1(t) + x_2(t) \leq 3$.

4.2. Example 2

Consider time-dependent nonlinear optimization under multiple types of constraints

$$\begin{aligned} \text{min.}\ & (\sin(t)/8 + 1/2)x_1^2(t) + (\cos(t)/8 + 1/2)x_2^2(t) \\ & + \cos(t)x_1(t)x_2(t) + \sin(3t)x_1(t) + \cos(3t)x_2(t) \\ \text{s. t.}\ & \sin(4t)x_1(t) + \cos(4t)x_2(t) = \cos(2t), \\ & -1.3 \leq x_1(t),\ x_2(t) \leq 1.3. \end{aligned} \tag{23}$$

Problem (23) is reformulated as a matrix form of nonlinear optimization involving time-dependent coefficients

$$\begin{aligned}
f(\mathbf{x}(t),t) &= (\sin(t)/8+1/2)x_1^2(t)+(\cos(t)/8+1/2)x_2^2(t)\\
&\quad+\cos(t)x_1(t)x_2(t)+\sin(3t)x_1(t)+\cos(3t)x_2(t),\\
\mathbf{x}(t) &= \left[x_1(t),x_2(t)\right]^{\mathrm{T}},\ A(t)=\left[\sin(4t),\cos(4t)\right],\\
\mathbf{b}(t) &= \left[-\cos(2t)\right],\ C(t)=\begin{bmatrix} I\\ -I\end{bmatrix},\\
\mathbf{d}(t) &= \left[-1.3,-1.3,-1.3,-1.3\right]^{\mathrm{T}}.
\end{aligned}$$

In this time-dependent case, the initial states is set as $[1,1,5,-10,-10,-10,-10]^{\mathrm{T}}$. In addition, the parameter is set as $\zeta=5$. The corresponding transient states and solutions synthesized by the proposed MZNN (16) for (23) are illustrated in Fig. 4. First, Fig. 4(a) illustrates that neural states $\mathbf{x}(t)$ of the proposed MZNN (16) strictly complying with the inequality constraint $-1.3\leq x_1(t),\ x_2(t)\leq 1.3$ for (23). Other neural network states $\lambda(t)$ and $\mu(t)$ are presented in Fig. 4(b). As illustrated in Fig. 4(c), it can be readily found that the time-dependent profiles of $g(\mathbf{x}(t))$ complies with equality-type constraint $\sin(4t)x_1(t)+\cos(4t)x_2(t)=\cos(2t)$. The residual errors generated via MZNN (16) during the solving processes in Fig. 4(d) exhibit the exponential convergent performance. It can be seen that the convergent time is within about 1 s. The above numerical results imply that solution $\mathbf{x}(t)$ illustrated in Fig. 4 are feasible solution to (23) under multiple types of constraints in the time-dependent case.

4.3. Example 3

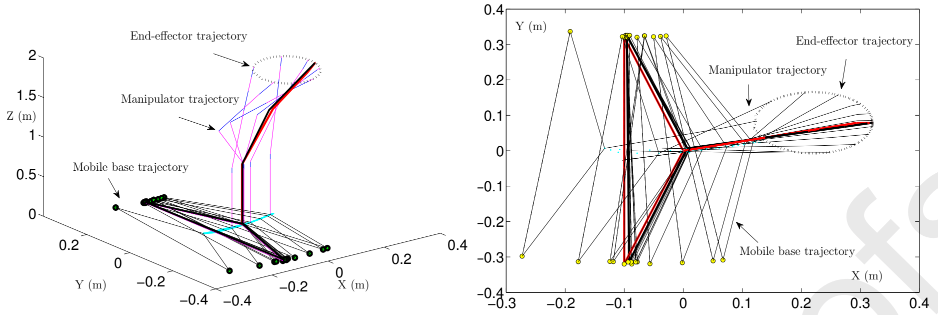
Furthermore, optimization (23) under another inequality constraint can be considered

$$\text{s. t. } h(\mathbf{x}(t),t)=x_1(t)+x_2(t)\leq 3, \tag{24}$$

where is rewritten in a matrix form of nonlinear optimization under inequality constraint with coefficients

$$C(t)=\left[1,1\right],\ \mathbf{d}(t)=\left[-3\right].$$

The objective optimization function and equality constraint is the same as (23). The initial states vector in this case is set to be $[1,1,5,-10]^{\mathrm{T}}$. The corresponding transient states and solutions synthesized by the proposed



(a) (b)

Figure 6: Real-world application results for addressing (4) of mobile robot with performance-index optimization and physical-limit constraints tracking a circle path using MZNN (16). (a) Three-dimensional motion trajectories of mobile robot. (b) Top view of motion trajectories of mobile robot.

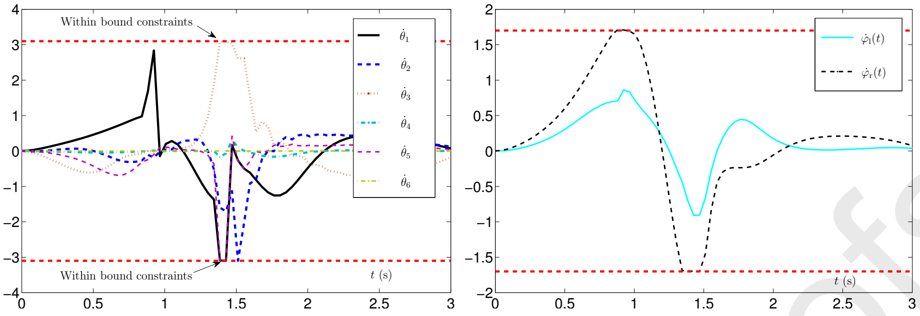
Table 2: Time and Space Complexity Synthesized by the Proposed MZNN (16) for (22)

Run time (s)	Time complexity	Storage element	Space complexity
0.9×10^{-5}	$O(T_\mathrm{d}/\tau)$	16	$(n+m+p)^2$

MZNN (16) for (24) with another inequality constraint are illustrated in Fig. 5. Firstly, Fig. 5(a) as well as 5(b) illustrate all states $\mathbf{x}(t)$, $\lambda(t)$ and $\mu(t)$ of the proposed MZNN (16) for solving (23). One could readily found in Fig. 5(c) that profiles of $g(\mathbf{x}(t))$ also complies with equality-type constraint $\sin(4t)x_1(t)+\cos(4t)x_2(t)=\cos(2t)$. The profile of $h(\mathbf{x}(t))$ strictly complies with inequality-type constraint as $x_1(t)+x_2(t)\leq 3$. The results can be readily found in Fig. 5(d). These results imply that solution $\mathbf{x}(t)$ illustrated in Fig. 5 is a feasible solution to (24) under another inequality constraint in the time-dependent case.

4.4. Application to robot nonlinear optimization and tracking control

In this part, a real-world application, i.e., the mobile robot nonlinear optimization control, is conducted to demonstrate effectiveness of the proposed proposed MZNN (16) for addressing the time-dependent nonlinear optimization problem (1). The actuator of mobile robot is expected to finish the tracking task of circle path. Simultaneously, the mobile robot is expected to con-



(a) (b)

Figure 7: Robot variables for addressing (4) of mobile robot under performance-index optimization and physical-limit constraints to track a circle path via MZNN (16). (a) Profiles of manipulator joint control signals $\dot{\theta}(t)$. (b) Profiles of driving wheels control signals $\dot{\phi}(t)$.

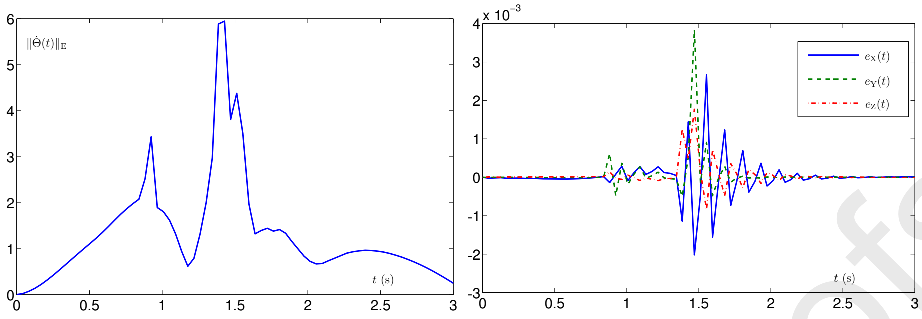
Table 3: Time and Space Complexity Synthesized by the Proposed MZNN (16) for (23)

Run time (s)	Time complexity	Storage element	Space complexity
1.0×10^{-5}	$O(T_{\rm d}/\tau)$	49	$(n+m+p)^2$

sider performance-index optimization and physical limits (4). The design circle path is mathematically expressed as $\mathbf{r}_{\rm md}(t) = [r_{\rm mdX}(t), r_{\rm mdY}(t), r_{\rm mdZ}(t)]^{\rm T}$, and elements in X, Y, and Z axes is given by

$$\begin{cases} r_{\rm mdX}(t) = & \kappa\cos(2\pi\sin^2(0.5\pi t/T_{\rm d})) - \kappa + 0.6891, \\ r_{\rm mdY}(t) = & \kappa\cos(\pi/6)\sin(2\pi\sin^2(0.5\pi t/T_{\rm d})) + 0.0069, \\ r_{\rm mdZ}(t) = & \kappa\sin(\pi/6)\sin(2\pi\sin^2(0.5\pi t/T_{\rm d})) + 0.1778, \end{cases}$$

of which the parameter is set as $\kappa = 0.1$ m in the robot tracking application. The upper bound of the physical limit for robot variables $\dot{\Theta}(t)$ is set as $\dot{\Theta}^+ = [1.7, 1.7, 3.1, 3.1, 3.1, 3.1, 3.1, 3.1]$ rad/s. The lower bound of the physical limit of robot variables $\dot{\Theta}(t)$ is set as $\dot{\Theta}^- = -[1.7, 1.7, 3.1, 3.1, 3.1, 3.1, 3.1, 3.1]$ rad/s. The robot experimental results for addressing problem (4) with both performance-index optimization and physical limits to track a circle path via MZNN (16) are presented in Fig. 6 through Fig. 8. Firstly, it can be found from Fig. 6(a) that the mobile robot successfully finishes desired



(a) (b)

Figure 8: Real-world application results on performance for (4) of mobile robot under performance-index optimization and physical-limit constraints to track a circle path using MZNN (16). (a) Profile of Euclidean norm of control signals $\|\dot{\Theta}(t)\|_{\mathrm{E}}$. (b) Profile of position error.

Table 4: Time and Space Complexity Synthesized by the Proposed MZNN (16) for Optimization with Another Inequality Constraint (24)

Run time (s)	Time complexity	Storage element	Space complexity
1.0×10^{-5}	$O(T_{\mathrm{d}}/\tau)$	16	$(n+m+p)^2$

tracking control task illustrated which can be found from trajectories in three-dimensional space. Top view of mobile robot tracking process can be seen from Fig. 6(b) showing the success of tracking task. Application results demonstrate that expected mobile robot task has been successfully finished. The corresponding motion control signals via MZNN (16) are illustrated in Fig. 7. Fig. 7(a) shows that profiles of mobile robot variable $\dot{\theta}(t)$ could be strictly constrained by physical limits in velocity level that depicted as an inequality constraints. Moreover, the profiles of driving wheels variables $\dot{\phi}(t)$ are complied with the physical limits within the boundary (see Fig. 7(b)). As clearly shown in the figure, the variable $\dot{\theta}(t)$ could be strictly constrained by physical limits in velocity level that depicted as an inequality constraints $\dot{\theta}^{-} \leq \dot{\theta}(t) \leq \dot{\theta}^{+}$. In addition, the profiles of driving wheels variables $\dot{\phi}(t)$ are complied with the physical limits within the boundary depicted as $\dot{\phi}^{-} \leq \dot{\phi}(t) \leq \dot{\phi}^{+}$. Note that the physical limits contain the boundary. That is to say that the lower and upper limits for the robot

Table 5: Time and Space Complexity Synthesized by the Proposed MZNN (16) for Mobile Robot Nonlinear Optimization and Tracking Control (4)

Run time (s)	Time complexity	Storage element	Space complexity
1.1×10^{-5}	$O(T_\mathrm{d}/\tau)$	169	$(n+m+p)^2$

control signal are also feasible. Therefore, the control signals that almost overlap the bound constraints are feasible in the proposed approach, which is also accord with theoretical analysis. The Euclidean norm of mobile robot variables $\|\dot{\Theta}(t)\|_\mathrm{E}$ is illustrated in Fig. 8(a). Note that the value of $\|\dot{\Theta}(t)\|_\mathrm{E}$ is related to the energy consumption and optimization. The relatively tiny position error shown in Fig. 8(b) verifies high tracking control accuracy when mobile robot considers performance-index optimization and physical limits. Both the experiment results and theoretical results are complete in terms of stability and convergence. Due to the complexity and nonlinearity, it is almost impossible to guarantee a time-dependent optimal solution for every time-independent nonlinear optimization.

Remark 2. *It is worth pointing out here that the proposed MZNN (16) can be easily and potentially applied to other different scenes with conditional constraints, for instances, robot-based transportation, robot-based water jet cutting and unmanned system patrol in addition to the above mobile robot tracking control.*

Remark 3. *The investigation on the computational complexity of the proposed MZNN for addressing time-dependent nonlinear optimization under multiple types of constraints is an important issue. Note that the investigation on computational complexity falls into two aspects, i.e., time complexity and space complexity. The time and space complexity of the proposed MZNN for handling the numerical studies and robot application are conducted and summarized in Table 2 through Table 5. Note that run time is the MATLAB run time for the proposed MZNN per iteration, and the storage element is the total number of operator storage element in MATLAB for the proposed MZNN per iteration [67]. As it can be found in the tables, the run time for all examples are about* 10^{-5} *s per iteration with the time complexity being of* $O(T_d/\tau)$ *manner, where symbol* τ *is the sampling period. In addition, the number of* *the storage element is different according to the dimension of the involved problem with the same space complexity being of* $(n + m + p)^2$ *manner. An algorithm for neural network model calculation with the computational time being about 1 ms (or to say, 1000 iterations per second) is acceptable for the existing robotic systems [68]. Hence, the proposed MZNN (16) with such computational complexity has a potentially high enough computational speed that can achieve the online time-dependent nonlinear optimization control in practical robot applications.*

Remark 4. *To sufficiently demonstrated rigor of this work, on the one hand, theoretical analyses on the global stability (i.e., Theorem 1 and its proof) and exponential convergence property (i.e., Theorem 2 and its proof) are rigorously presented in this paper. On the other hand, numerical verifications (i.e., three different time-dependent nonlinear optimization examples), real-world robot application, comparisons with other existing models and performance tests fully verify the effectiveness in addition to superiority of MZNN for time-dependent nonlinear optimization under multiple types of constraints.*

4.5. Comparisons and tests

To show the advantage of the proposed MZNN (16) for (1) under multiple types of constraints, comparisons and tests are conducted. Conventionally, different GNNs were developed as an alternative for time-dependent handling. As for handling (1) subject to multiple types of constraints, an energy function in scalar form could be defined as $\varepsilon(t) = \|\mathbf{e}(t)\|_{\mathrm{E}}^2/2$. The GNN is thus developed

$$\dot{\mathbf{y}}(t) = -\rho\frac{\partial\varepsilon(t)}{\partial\mathbf{y}(t)} = -\rho\left(\frac{\partial\mathbf{e}(t)}{\partial\mathbf{y}(t)}\right)^{\mathrm{T}}\mathbf{e}(t), \tag{25}$$

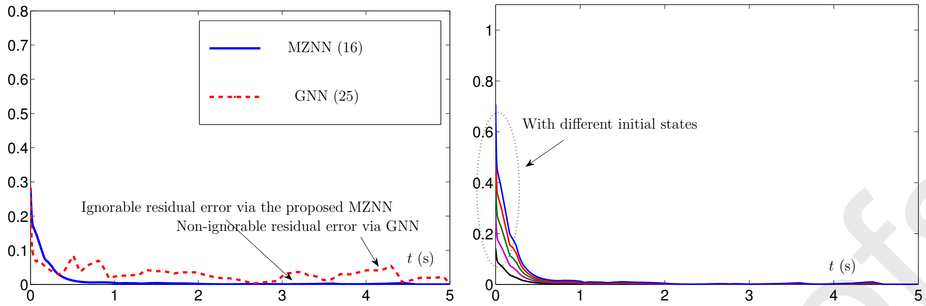
of which ρ is a user-predefined parameter for GNN (25). In contrast with GNN (25) utilizing scalar-valued energy functions in Euclidean norm depicted in $\varepsilon(t) = \|\mathbf{e}(t)\|_{\mathrm{E}}^2/2$, ZNN defines an indefinite error function with scalar value depicted in $\mathbf{e}(t)$ (also could be the vector value or matrix valued). In the meantime, the ZNN fully explores the time-derivative data and conditions for solving time-dependent issues but GNN does not. The compared result illustrated by the residual errors via the proposed MZNN (16) and the GNN (25) for (23) is shown in Fig. 9(a). As shown in this figure, residual error via the proposed MZNN (16) illustrates the exponential convergence property, and rapidly converges to zero. However, the residual error via the GNN (25) exhibits deviations in the steady state with a non-ignorable value. The large residual error via GNN (25) means that the solution always lags behind the theoretical solution, which is unacceptable in some computational applications. Hence, the proposed MZNN (16) is superior to the GNN (25) in terms of the convergence property and accuracy.

In addition, the global convergence property of the proposed MZNN (16) is investigated by conducting numerical studies with different values of initial neural network states in Fig. 9(b). To intuitively and effectively observe the transient state of convergence process, the initial values of neural network states $\mathbf{y}(0)$ are set away from the initial theoretical values $\mathbf{y}_{\mathrm{t}}(0)$ of neural network states in the test. In other words, as the input information of system, the initial values of neural network states are set as $\mathbf{y}(0) = \mathbf{y}_{\mathrm{t}}(0) + \Delta o$, of which vector $\Delta o \in \mathbb{R}^{n+m+p}$ is an initial offset between the theoretical and user defined neural network states. If the initial values of neural network states $\mathbf{y}(0)$ are exactly set on the initial theoretical values $\mathbf{y}_{\mathrm{t}}(0)$ of neural network states, the transient neural network states of the convergence process would not be easily observed from residual errors. In this situation, the dynamic property of MZNN could not be analyzed easily. Hence, the initial values of neural network states are set away from the initial theoretical values $\mathbf{y}_{\mathrm{t}}(0)$ of neural network states in the test. As seen from the figure, starting from different neural network states, the residual errors generated by MZNN (16) show almost the same exponential convergence property being consistent with the theoretical results. In other words, the MZNN (16) possesses the global convergence property.

Finally, to verify the advantages of computational complexity of the proposed MZNN (16), comprehensive comparisons with other kinds of neural networks on time and space complexity are further investigated and presented. As it can be found in Table 6, for the same optimization (24) in practice under multiple types of constraints, different neural networks possess the same manner of time complexity but different manners of space complexity. As compared with other kinds of neural networks [8, 38, 51, 56, 57, 69–71], the proposed MZNN (16) possesses the lowest time and space complexity being of $O(T_{\mathrm{d}}/\tau)$ manner and $(n+m+p)^2$ manner, respectively.

5. Conclusion and Future Work

In a unified design framework of ZNN, this paper has developed a novel MZNN, for time-dependent nonlinear optimization under multiple type-



(a) (b)

Figure 9: Comparisons the proposed MZNN (16) and via GNN (25) as well as test results of residual errors with different values of initial neural network states. (a) Profiles of residual errors via different neural network models. (b) Profiles of residual errors with different initial states.

s of constraints. The optimization with multiple types of constraints has been skillfully converted to a time-dependent equality system via Lagrange method. The proposed MZNN has inherently shown the effectiveness of exponential convergence property by utilizing the time-derivative information. Theoretically, the proposed MZNN has been proven to be globally stable with the neural network state converging to a time-dependent solution of optimization problem. In addition, the neural network state has been further proven to be exponential convergence to a time-dependent solution. Numerical verifications with three general nonlinear optimization examples, real-world robot application, comparisons with other existing neural networks and performance tests have been conducted in this work. All results have verified the stability and exponential convergence property for addressing time-dependent nonlinear optimization under multiple types of constraints. Practically, the proposed MZNN has been applied to the real-world robot nonlinear optimization control. The related results have shown that the mobile robot has successfully finished tracking control task with related control signals constrained by physical limits that has verified the effectiveness as well as superiority of MZNN.

The limitations and future directions of the work lie in the following facts. i) The optimization object function is required to be second order differentiable and convex with respect to the time-dependent state vector. An in-depth investigation on other kinds of time-dependent nonlinear optimization objective function could be an interesting and open research direction.

Table 6: Comparisons on Time and Space Complexity Synthesized by Different Neural Networks for (24) in Practice Subject to Multiple Types of Constraints

Neural network	Time complexity	Storage element	Space complexity
GNN [38, 51]	$O(T_{\mathrm{d}}/\tau)$	64	$(n+m+p)^3$
RNN [56, 69]	$O(T_{\mathrm{d}}/\tau)$	16	$(n+m+p)^2$
VPCNN [70, 71]	$O(T_{\mathrm{d}}/\tau)$	16	$(n+m+p)^2$
PDNN [8, 57]	$O(T_{\mathrm{d}}/\tau)$	64	$(n+m+p)^3$
Proposed (16)	$O(T_{\mathrm{d}}/\tau)$	16	$(n+m+p)^2$

Note: The GNN, RNN, VPCNN and PDNN are the abbreviations of gradient neural network, recurrent neural network, varying-parameter convergent-differential neural network and primal-dual neural network.

ii) The robustness of MZNN to handle time-dependent disturbances is not considered in this pioneering work, which would also be another interesting research direction. iii) An hardware implementation is needed. Implementation of MZNN on hardware by applying embedded systems to develop progressive controllers for real-world applications would also be a further work.

References

[1] X. Hu, B. Zhang, A new recurrent neural network for solving convex quadratic programming problems with an application to the k-winners-take-all problem, IEEE Trans. Neural Netw. 20 (4) (2009) 654–664.

[2] S. Qin, J. Feng, J. Song, X. Wen, C. Xu, A one-layer recurrent neural network for constrained complex-variable convex optimization, IEEE Trans. Neural Netw. Learn. Syst. 29 (3) (2018) 534–544.

[3] H. Huang, T. Zhang, C. Yang, and C. L. Philip Chen, Motor learning and generalization using broad learning adaptive neural control, IEEE Trans. Ind. Electron. 67 (10) (2020) 8608–8617.

[4] B. Xu, C. Yang, Y. Pan, Global neural dynamic surface tracking control of strict-feedback systems with application to hypersonic flight vehicle, IEEE Trans. Neural Netw. Learn. Syst. 26 (10) (2015) pp. 2563–2575.

[5] X. Luo, Z. Liu, L. Jin, Y. Zhou, M. Zhou, Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis, IEEE Trans. Neural Netw. Learn. Syst. to be published. doi: 10.1109/TNNLS.2020.3041360.

[6] D. Chen, S. Li, Q. Wu, X. Luo, Super-twisting ZNN for coordinated motion control of multiple robot manipulators with external disturbances suppression, Neurocomputing 371 (2020) pp. 78–90.

[7] H. Huang, C. Yang, C. L. Philip Chen, Optimal robot-environment interaction under broad fuzzy neural adaptive control, IEEE Trans. Cybern. to be published. doi: 10.1109/TCYB.2020.2998984.

[8] D. Chen, S. Li, W. Li, Q. Wu, A multi-level simultaneous minimization scheme applied to jerk bounded redundant robot manipulators, IEEE Trans. Autom. Sci. Eng. 17 (1) (2020) 463–474.

[9] Q. Liu, J.Wang, A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming, IEEE Trans. Neural Netw. 19 (4) (2008) 558–570.

[10] P. Miao, Y. Shen, Y. Huang, Y. Wang, Solving time-varying quadratic programs based on finite-time Zhang neural networks and their application to robot tracking, Neural Process. Lett. 26 (3) (2015) 693–703.

[11] Y. Shi, C. Mou, Y. Qi, B. Li, S. Li, and B. Yang, Design, analysis and verification of recurrent neural dynamics for handling time-variant augmented Sylvester linear system, Neurocomputing 426 (2021) 274–284.

[12] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive parameter estimation and control design for robot manipulators with finite-time convergence, IEEE Trans. Ind. Electron. 65 (10) (2018) 8112–8123.

[13] C. Yang, D. Huang, W. He, and L. Cheng, Neural control of robot manipulators with trajectory tracking constraints and input saturation, IEEE Trans. Neural Netw. Learn. Syst., to be published, doi: 10.1109/TNNLS.2020.3017202.

[14] C. Yang, J. Luo, Y. Pan, Z. Liu, C.-Y. Su, Personalized variable gain control with tremor attenuation for robot teleoperation, IEEE Trans. Syst., Man, Cybern., Syst. 48 (10) (2018) 1759–1770.

[15] H. Wang, Adaptive visual tracking for robotic systems without image-space velocity measurement, Automatica 55 (2015) 294–301.

[16] W. He, B. Huang, Y. Dong, Z. Li, C.-Y. Su, Adaptive neural network control for robotic manipulators with unknown deadzone, IEEE Trans. Cybern. 48 (9) (2018) 2670–2682.

[17] D. Chen, S. Li, A recurrent neural network applied to optimal motion control of mobile robots with physical constraints, Appl. Soft Comput. 85 (2019) 105880.

[18] D. Chen, Y. Zhang, S. Li, Tracking control of robot manipulators with unknown models: a Jacobian-matrix-adaption method, IEEE Trans. Ind. Informat. 14 (7) (2018) 3044–3053.

[19] D. Chen, S. Li, Q. Wu, A novel supertwisting zeroing neural network with application to mobile robot manipulators, IEEE Trans. Neural Netw. Learn. Syst. 32 (4) (2021) 1776–1787.

[20] J. Na, J. Yang, X. Wu, Y. Guo, Robust adaptive parameter estimation of sinusoidal signals, Automatica 53 (2015) 376–384.

[21] J. Li, M. Mao, F. Uhlig, Y. Zhang, Z-type neural-dynamics for time-varying nonlinear optimization under a linear equality constraint with robot application, J. Comput. Appl. Math. 327 (2018) 155–166.

[22] J. Li, Y. Shi, and H. Xuan, Unified model solving nine types of time-varying problems in the frame of zeroing neural network, IEEE Trans. Neural Netw. Learn. Syst. 32 (5) (2021) 1896–1905.

[23] A. H. Khan, S. Li, X. Luo, Obstacle avoidance and tracking control of redundant robotic manipulator: an RNN-based metaheuristic approach, IEEE Trans. Ind. Informat. 16 (7) (2020) 4670–4680.

[24] C. Yang, X. Wang, L. Cheng, H. Ma, Neural-learning-based telerobot control with guaranteed performance, IEEE Trans. Cybern. 47 (10) (2017) 3148–3159.

[25] B. Xu, Y. Shou, J. Luo, H. Pu, Z. Shi, Neural learning control of strict-feedback systems using disturbance observer, IEEE Trans. Neural Netw. Learn. Syst. 30 (5) (2019) 1296–1307.

[26] G. Peng, C. L. P. Chen, C. Yang, Neural networks enhanced optimal admittance control of robot-environment interaction using reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst. to be published. doi: 10.1109/TNNLS.2021.3057958.

[27] D. Chen, S. Li, F.-J. Lin, Q. Wu, New super-twisting zeroing neural-dynamics model for tracking control of parallel robots: a finite-time and robust solution, IEEE Trans. Cybern. 50 (6) (2020) 2651–2660.

[28] D. Chen, S. Li, Q. Wu, X. Luo, New disturbance rejection constraint for redundant robot manipulators: an optimization perspective, IEEE Trans. Ind. Informat. 16 (4) (2020) 2221–2232.

[29] H. Wang, P. X. Liu, S. Li, D. Wang, Adaptive neural output-feedback control for a class of nonlower triangular nonlinear systems with unmodeled dynamics, IEEE Trans. Neural Netw. Learn. Syst. 29 (8) (2018) 3658–3668.

[30] H. Wang, X. Liu, K. Liu, Robust adaptive neural tracking control for a class of stochastic nonlinear interconnected systems, IEEE Trans. Neural Netw. Learn. Syst. 27 (3) (2016) 510–523.

[31] R. Rakkiyappan, S. Dharani, J. Cao, Synchronization of neural networks with control packet loss and time-varying delay via stochastic sampled-data controller, IEEE Trans. Neural Netw. Learn. Syst. 26 (12) (2015) 3215–3226.

[32] D. Huang. C. Yang, Y. Pan, L. Cheng, Composite learning enhanced neural control for robot manipulator with output error constraints, IEEE Trans. Ind. Informat., 17 (1) (2020) 209–218.

[33] Q. Wei, B. Li, R. Song, Discrete-time stable generalized self-learning optimal control with approximation errors, IEEE Trans. Neural Netw. Learning Syst. 29 (4) (2018) 1226–1238.

[34] D. Zheng, W.-F. Xie, X. Ren, J. Na, Identification and control for singularly perturbed systems using multitime-scale neural networks, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2017) 321–333.

[35] H. Lu, L. Jin, J. Zhang, Z. Sun, S. Li, Z. Zhang, New joint-drift-free scheme aided with projected ZNN for motion generation of redundant

robot manipulators perturbed by disturbances, IEEE Trans. Syst., Man, Cybern., Syst. to be published. doi: 10.1109/TSMC.2019.2956961.

[36] Z. Zhang, L.-D. Kong, L. Zheng, Power-type varying-parameter RNN for solving TVQP problems: design, analysis, and applications, IEEE Trans. Neural Netw. Learn. Syst. 30 (8) (2019) 2419–2433.

[37] A. Nazemi, M. Nazemi, A gradient-based neural network method for solving strictly convex quadratic programming problems, Cogn. Comput. 6 (3) (2014) 484–495.

[38] Y. Chen, C. Yi, D. Qiao, Improved neural solution for the Lyapunov matrix equation based on gradient search, Inf. Process. Lett. 13 (22–24) (2013) 876–881.

[39] F. Xu, Z. Li, Z. Nie, H. Shao, D. Guo, Zeroing neural network for solving time-varying linear equation and inequality systems, IEEE Trans. Neural Netw. Learn. Syst. 30 (8) (2019) 2346–2357.

[40] N. Tan, P. Yu, Robust model-free control for redundant robotic manipulators based on zeroing neural networks activated by nonlinear functions, Neurocomputing 438 (2021) 44–54.

[41] W. Li, A recurrent neural network with explicitly definable convergence time for solving time-variant linear matrix equations, IEEE Trans. Ind. Informat. 14 (12) (2018) 5289–5298.

[42] Y. Zhang, D. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, IEEE Trans. Neural Netw. 13 (5) (2002) 1053–1063.

[43] Y. Zhang, L. Xiao, Z. Xiao, M. Mao, Zeroing Dynamics, Gradient Dynamics, and Newton Iterations. Boca Raton, FL, USA: CRC Press, 2015.

[44] Y.-J. Liu, Y. Gao, S. Tong, C. L. P. Chen, A unified approach to adaptive neural control for nonlinear discrete-time systems with nonlinear deadzone input, IEEE Trans. Neural Netw. Learn. Syst. 27 (1) (2016) 139–150.

[45] Y.-J. Liu, M. Gong, S. Tong, C. L. P. Chen, D.-J. Li, Adaptive fuzzy output feedback control for a class of nonlinear systems with full state constraints, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 2607–2617.

[46] R. Rakkiyappan, J. Cao, G. Velmurugan, Existence and uniform stability analysis of fractional-order complex-valued neural networks with time delays, IEEE Trans. Neural Netw. Learn. Syst. 26 (1) (2015) 84–97.

[47] H. Wang, H. R. Karimi, P. X. Liu, H. Yang, Adaptive neural control of nonlinear systems with unknown control directions and input dead-zone, IEEE Trans. Syst., Man, Cybern., Syst. 48 (11) (2018) 1897–1907.

[48] H. Li, S. Shao, S. Qin, Y. Yang, Neural networks with finite-time convergence for solving time-varying linear complementarity problem, Neurocomputing 439 (2021) 146–158.

[49] J. Li, Y. Zhang, M. Mao, General square-pattern discretization formulas via second-order derivative elimination for zeroing neural network illustrated by future optimization, IEEE Trans. Neural Netw. Learn. Syst. 30 (3) (2019) 891–901.

[50] L. Jin, S. Li, B. Liao, Z. Zhang, Zeroing neural networks: A survey, Neurocomputing 267 (2017) 597–604.

[51] C. Yi, Y. Chen, Z. Lu, Improved gradient-based neural networks for online solution of Lyapunov matrix equation, Inf. Process. Lett. 111 (16) (2011) 780–786.

[52] P. S. Stanimirović, S. Srivastava, D.K. Gupta, From Zhang neural network to scaled hyperpower iterations, J. Comput. Appl. Math. 331 (2018) 133–155.

[53] D. Guo, L. Yan, Z. Nie, Design, analysis, and representation of novel five-step DTZD algorithm for time-varying nonlinear optimization, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2018) 4248–4260.

[54] L. Xiao, B. Liao, A convergence-accelerated Zhang neural network and its solution application to Lyapunov equation, Neurocomputing, 193 (2016) 213–218.

[55] D. Chen, Y. Zhang, Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2018) 4385–4397.

[56] W. Li, Z. Su, Z. Tan, A recurrent neural network with explicitly definable convergence time for solving time-variant linear matrix equations, IEEE Trans. Ind. Informat. 14 (12) (2018) 5289–5298.

[57] Y. Zhang, Z. Li, H.-Z. Tan, Z. Fan, On the simplified LVI-based primal-dual neural network for solving LP and QP problems, 2007 IEEE International Conference on Control and Automation, Guangzhou, China 3129–3134, 2007.

[58] L. Jin, Y. Zhang, S. Li, Y. Zhang, Modified ZNN for time-varying quadratic programming with inherent tolerance to noises and its application to kinematic redundancy resolution of robot manipulators, IEEE Trans. Ind. Electron. 63 (11) (2016) 6978–6988.

[59] Y. Zhang, H. Gong, M. Yang, J. Li, X. Yang, Stepsize range and optimal value for Taylor-Zhang discretization formula applied to zeroing neurodynamics illustrated via future equality-constrained quadratic programming, IEEE Trans. Neural Netw. Learn. Syst. 30 (3) (2019) 959–966.

[60] L. Xiao, K. Li, M. Duan, Computing time-varying quadratic optimization with finite-time convergence and noise tolerance: a unified framework for zeroing neural network, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3360–3369.

[61] J. H. Mathews, K. K. Fink, Numerical Methods Using MATLAB. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.

[62] X. Huang, X. Lou, B. Cui, A novel neural network for solving convex quadratic programming problems subject to equality and inequality constraints, Neurocomputing 214 (2016) 23–31.

[63] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, New York, 2004.

[64] J. C. Polking, A Leibniz formula for some differentiation operators of fractional order, Indiana U. Math. J., 21 (11) (1972) 1019–1029.

[65] C. Mead, Analog VLSI and Neural Systems, Reading, MA: Addison-Wesley, 1989.

[66] Y. N. Bapat, Electronic Circuits and Systems: Analog and Digital, Tata McGraw-Hill, New Delhi, 1992.

[67] Y.W. Kwon, H. Bang, The Finite Element Method Using MATLAB, CRC Press, Florida, 2000.

[68] H. Zhang, C. Reardon, and L. E. Parker, Real-time multiple human perception with color-depth cameras on a mobile robot, IEEE Trans. Cybern. 43 (5) (2013) 1429–1441.

[69] L. Xiao, Z. Zhang, Z. Zhang, W. Li, S. Li, Design, verification and robotic application of a novel recurrent neural network for computing dynamic Sylvester equation. Neural Netw. 105 (2018) 185-196.

[70] Z. Zhang, T. Fu, Z. Yan, L. Jin, L. Xiao, Y. Sun, Z. Yu, Y. Li, A varying-parameter convergent-differential neural network for solving joint-angular-drift problems of redundant robot manipulators. IEEE/ASME Trans. Mech. 23 (2) (2018) 679-689.

[71] Z. Zhang, Y. Lu, L. Zheng, S. Li, Z. Yu, Y. Li, A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality. IEEE Trans. Autom. Control 63 (12) (2018) 4110-4125.