# A zeroing neural network for solving discrete time-varying minimization with different adjustable parameter

Haolin He
School of Information and Communication Engineering,
University of Electronic Science and Technology of China, Chengdu, 611731, P. R. China

## ABSTRACT

It is popular approach to employ zeroing neural networks (ZNN) to handle time-varying problems. However, most of the existing methods have limited flexibility in the discretization process, which seriously affects the final effect of the model. To solve this problem, in this paper, we present a new discrete-time ZNN (DTZNN) for handling discrete time-varying minimization problem. In detail, the new DTZNN is proposed with four-instant general discretization formula, which has a adjustable parameter $d$. Generally speaking, in the process of discretization, different values of $d$ and time interval can influence the residual error observably. In order to verify the effectiveness of our method, we conduct multiple numerical experiments. The numerical experiment results show that the new DTZNN has effectiveness and superiority for handling discrete time-varying minimization problem.

**Keywords:** Zeroing neural network, discrete time-varying minimization problem, four-instant general discretization formula.

## 1. INTRODUCTION

In recent years, researches in minimization have rising heat, due to its essential position in artificial intelligence fields, such as the control of mechanical arms [1], health big data digitization [2] and logistics engineering [3]. Many researchers have developed numerous algorithms based on traditional machine learning techniques and the latest deep learning techniques. Although there are many minimization algorithms, a considerable part of them are designed to solve time-invariant problems. But in reality there are still time-varying problems that need to be addressed, and with multiple time-varying applications in both industry and research fields, it is of great importance to develop minimization model to deal with time-varying problems.

Among many minimization algorithms, ZNN is widely used due to its special structure. At the same time, due to the particularity of its own structure, there are natural similarities between the types of problems it can deal with and time-varying problems, so it has been receiving extensive attention. Specifically, zeroing neural network (ZNN), a specific kind of recurrent neural network (RNN), is a product of Hopfield neural network research. Recurrent neural network is a kind of neural network with a special structure, and its main processing core is a computing unit of cyclic operation, which can realize parameter sharing and state storage. It is capable of receiving long sequences of information, such as a piece of text or a piece of speech. After receiving the sequence information, the cyclic neural network can perform operations in sequence, and at the same time, it can save the above information to an intermediate state, and comprehensively consider when processing the following information to complete context awareness. Zeroing neural network is a special case of neural network with this structure, which has the structural characteristics of RNN and also has some independent designs. For its parallel distributed design and practical hardware implementation, ZNN is suggested as a methodical strategy to address time-varying problems. For instance, in [4], to solve the Lyapunov equation, researchers suggest using a convergence-accelerated zeroing neural network, which is frequently seen in the sciences and engineering. And researchers in [5] discover that when used to control a nonlinear dynamical system, ZNN outperforms controller based on linear quadratic regulator (LQR).

*E-mail: harland0108@std.uestc.edu.cn

Seventh International Conference on Mechatronics and Intelligent Robotics (ICMIR 2023), edited by Srikanta Patnaik, Tao Shen, Proc. of SPIE Vol. 12779, 127792Q · 0277-786X · doi: 10.1117/12.2688661

These studies have fully proved that the neural network processing of zeroing neural network (ZNN) can handle time-varying problems well. Hence, zeroing neural network (ZNN) is created to address a variety of time-varying issues in practical engineering applications [6]. The iteration of a neural network is discrete in all real-world engineering issues. This characteristic of ZNN enables it to handle some engineering problems involving time-varying, but there is another point that cannot be ignored in the application process, which is the processing of nonlinear time-varying, that is, discrete time-varying problems. To make ZNN play a role in practical engineering problems and reduce time-varying nonlinearity, discrete-time ZNN (DTZNN), which originates from discretization of continuous-time zeroing neural network (CTZNN), can be thought of as a useful tool for resolving discrete time-varying issues [7].

After paying attention to the need to solve the discretization problem, researchers have proposed many types of algorithms, including further deformation of DTZNN and derivation algorithm from CTZNN to DTZNN. During the process of DTZNN building, it is necessary to discretize the CTZNN using the proper formula for discretization. Up to now, an Euler-type DTZNN using the Euler forward-difference formula has been presented [8]. In order to carry out the discrete-time varying nonlinear minimization, researchers have also developed Taylor-type DTZNN based on Taylor-type differential method [9-10]. Additionally, a general three-step technique is created by the researchers to discretize the CTZNN into a DTZNN [11].

All the above-mentioned discretization methods for CTZNN have achieved good results in practice, but there is a key problem, that is, the operational flexibility in the discretization process is limited. Intuitively speaking, an adjustable discretization process will also have a great impact on the final effect. In this paper, to further optimize and add more flexibility to the related DTZNN, we present the general four-instant discretization formula [12], which has an adjustable parameter $d$ , for the discretization process. These parameters on the one hand represent the flexibility of the discrete process, and on the other hand have an impact on the final result. On this basis, we conducted several numerical verification experiments. First, we designed control variable experiments to explore the flexibility of the method. And we designed experiments to explore the trend of the residual as d changes, in order to confirm the impact of the method on the final effect. The results show that with the change of $d$ , the convergence curve of the residual error also changes. For the design of ZNN, the new four-instant discretization formula offers greater operability, so that the new ZNN theory can deal with more kinds of engineering problems.

## 2. PROBLEM FORMULATION

Before discretization, the CTZNN approach can be used to create the CTZNN model initially. The following time-varying minimization issue [9] is initially thought of as

$$\min_{\mathbf{x}(t)\in\mathbb{R}^n} f(\mathbf{x}(t),t)\in\mathbb{R}, t\in[t_0,t_{\mathrm{f}}]\subseteq[0,+\infty), \tag{1}$$

where the second-order differentiable and bounded $f(\cdot,\cdot):\mathbb{R}^n\times[0,+\infty)\to\mathbb{R}$ is a time-varying nonlinear mapping function. In ZNN theory, $f(\mathbf{x}(t),t)$ is also considered as a time-varying signal, while $x(t)$ is always considered as neural state.

Calculating the gradient of $f(\mathbf{x}(t),t)$ is an important step to address the minimization problem in CTZNN. By using the matrix differentiation rule, gradient $\psi(\mathbf{x}(t),t)$ is derived:

$$\psi(\mathbf{x}(t),t)=\frac{\partial f(\mathbf{x}(t),t)}{\partial \mathbf{x}(t)}\in\mathbb{R}^n, \tag{2}$$

which derives from $f(\mathbf{x}(t),t)$ and is an array of nonlinear differentiable mapping functions. It can be concluded from the idea of derivative that, to solve the minimization problem, $\psi(\mathbf{x}(t),t)$ is required to be zero.

After that, we use the ZNN design formula, $\mathbf{e}(t)=\mathrm{d}\mathbf{e}(t)/\mathrm{d}t=-\gamma\mathbf{e}(t)$ [13], where the vector-valued error function is defined as $\mathbf{e}(t)=\psi(\mathbf{x}(t),t)$ , and design parameter $\gamma>0$ is used to adjust the convergence rate of ZNN. The dynamical formula of CTZNN below is obtained by expanding the ZNN design formula:

$$H(\mathbf{x}(t),t)\dot{\mathbf{x}}(t)+\psi_t^{'}(\mathbf{x}(t),t)=-\gamma\psi(\mathbf{x}(t),t). \tag{3}$$

In equation , $H(\mathbf{x}(t),t)$ and $\psi_t^{'}(\mathbf{x},t)$ are defined as

$$H(\mathbf{x}(t),t)=\frac{\partial\psi(\mathbf{x}(t),t)}{\partial\mathbf{x}^{\mathrm{T}}}\in\mathbb{R}^{n\times n}$$

and

$$\psi_t^{'}(\mathbf{x},t)=\frac{\partial\psi(\mathbf{x}(t),t)}{\partial t}\in\mathbb{R}^{n}.$$

Thus, is translated as follow for the nonsingular condition of $H(\mathbf{x}(t),t)$ that we investigate in the research:

$$\dot{\mathbf{x}}(t)=-H^{-1}(\mathbf{x}(t),t)(\gamma\psi(\mathbf{x}(t),t)+\psi_t^{'}(\mathbf{x}(t),t)). \tag{4}$$

After discreting continous time-varying signal in equation on the time interval $[t_0,t_{\mathrm{f}}]$, the equation that follows can be formed as

$$\min_{\mathbf{x}_{k+1}\in\mathbb{R}^n} f(\mathbf{x}_{k+1},t_{k+1})\in\mathbb{R}, \tag{5}$$

Since it is a given that sampling at time instant $t=(k+1)\delta$ (abbreviated as $t_{k+1}$) creates and measures the signal $f(\mathbf{x}_{k+1},t_{k+1})$ from the signal $f(\mathbf{x}(t),t)$ with a smooth time variation. Additionally, the update index is indicated by $k=0,1,2\cdots$, while the sampling gap is indicated by $\delta>0$.

To solve the discrete-time nonlinear minimization shown in equation , equation is discretized to generate following equation as

$$\dot{\mathbf{x}}_k=-H^{-1}(\mathbf{x}_k,t_k)(\gamma\psi(\mathbf{x}_k,t_k)+\psi_{t_k}^{'}(\mathbf{x}_k),t_k)). \tag{6}$$

In order to solve the time-varying minimization issue using DTZNN, $\mathbf{x}_{k+1}$ need to be generated between $t_k$ and $t_{k+1}$. To realize this, proper theory for discretization and differentiation is required to ensure the computing efficiency, while flexibility should also be considered. In this paper, for the first time, we use the four-instant discretization formula in the differential step [12], which is shown in following Lemma 1.

**Lemma 1** *There is an O truncation Error* $O(\delta^2)$ *in the second-order derivative elimination (SODE) method's four-instant discretization formula. The effective domain of parameter* $d$ *is* $d\in(-\infty,-1)\cup(1/3,+\infty)$ *for the ZNN discretized by such a formula.*

The general four-instant discretization formula is given as follow:

$$\begin{aligned}\dot{\mathbf{x}}_k&=\frac{2d}{(3d-1)\delta}\mathbf{x}_{k+1}-\frac{3d+3}{2(3d-1)\delta}\mathbf{x}_k\\&\quad+\frac{2}{(3d-1)\delta}\mathbf{x}_{k-1}+\frac{-1-d}{2(3d-1)\delta}\mathbf{x}_{k-2}+O(\delta^2).\end{aligned} \tag{7}$$

After that, the Taylor-type DTZNN as follow can be created by discretizing CTZNN in equation using the aforementioned equation

$$
\begin{aligned}
\mathbf{x}_{k+1} = &\frac{(3d+3)}{4d}\mathbf{x}_k - \frac{1}{d}\mathbf{x}_{k-1} + \frac{d+1}{4d}\mathbf{x}_{k-2} \\
&-\frac{(3d-1)\delta}{2d}H^{-1}(\mathbf{x}_k,t_k)\gamma\psi(\mathbf{x}_k,t_k) \\
&-\frac{(3d-1)\delta}{2d}H^{-1}(\mathbf{x}_k,t_k)\psi_t^{'}(\mathbf{x}_k,t_k) \\
&-O(\delta^3),
\end{aligned} \tag{8}
$$

where $d$ is a parameter that can be adjusted within $(-\infty,-1)\cup(1/3,+\infty)$. When $d$ is set to be 1, the following equation of $\mathbf{x}_{k+1}$ is derived as

$$
\begin{aligned}
\mathbf{x}_{k+1} = &1.5\mathbf{x}_k - \mathbf{x}_{k-1} + 0.5\mathbf{x}_{k-2} \\
&-H^{-1}(\mathbf{x}_k,t_k)\gamma\psi(\mathbf{x}_k,t_k) \\
&-H^{-1}(\mathbf{x}_k,t_k)\psi_t^{'}(\mathbf{x}_k,t_k) - O(\delta^3),
\end{aligned} \tag{9}
$$

which is the exact outcome of utilizing Taylor-type differentiation formula for discretization [8]. From this result, it can be deduced that for the four-instant general discretization formula, the Taylor-type differentiation formula is a specific instance. So compared with Taylor-type differentiation formula, the four-instant general discretization formula gives the novel DTZNN more tunability and flexibility in various engineering problems.

## 3. NUMERICAL EXPERIMENTS AND VERIFICATIONS

In our proposed method, the main focus is on the flexibility embodied in the discretization process, which affects the generalization of the algorithm and the final experimental effect. In order to further verify the actual effect of the algorithm, we designed two parts of experiments. The first part of the experiment focuses on the relationship between the selection of different values and the order of the residuals. Specifically, we verify the flexibility of the algorithm by observing the order of the residuals under different values. In the second part, we mainly focus on the impact of the value on the final effect, and explore an appropriate value strategy.

We present a number of numerical experiments in order to further investigate numerous properties of the novel DTZNN. During the experiments, we used the control variate method by selecting different values of $\delta$ and $d$. To some extent, the experimental findings reflect the features and advantages of the new DTZNN.

Take into account the subsequent discrete time-varying minimization problem, where each computation time period $[k\delta,(k+1)\delta]\subseteq[0,10]$ should compute $\mathbf{x}_{k+1}$ with update index $k=0,1,2\cdots$:

$$
\min_{\mathbf{x}_{k+1}\in\mathbb{R}^n} f(\mathbf{x}_{k+1},t_{k+1})\in\mathbb{R}, \tag{10}
$$

in which $\mathbf{x}_k = \left[x_1(t_k),x_2(t_k),x_3(t_k),x_4(t_k)\right]^{\mathrm{T}}$.

In the following two experiments, we created two time-varying mapping functions as examples to show more general experimental findings as

$$
\begin{aligned}
f_1(\mathbf{x}_k,t_k) = &(3+\sin(t_k))x_1(t_k)^2 + 2x_2(t_k)^2 \\
&+(x_3(t_k)-e^{-t_k})^2 + (x_4(t_k)+e^{-t_k})^2 \\
&+(x_1(t_k)+\sin(t_k))x_3(t_k) \\
&+0.1(t_k-1)x_3(t_k)x_4(t_k)
\end{aligned} \tag{11}
$$

and

$$f_2(\mathbf{x}_k, t_k) = (2+\cos(t_k))x_1(t_k)^2 + x_2(t_k)^2 + 3x_3(t_k)^2 + x_4(t_k)^2 - \cos(t_k)x_1(t_k)x_3(t_k) - \exp(-t_k)x_2(t_k) + 3x_3(t_k). \tag{12}$$

### 3.1 First Experiment

The first experiment is designed to analyze the evolution of residual error. In this experiment, we study the change of error with the change of $\delta$ and prove the order of residual error to be $O(\delta^3)$.

In DTZNN, $h = \gamma\delta > 0$ is often used to represent the iteration step size. In this experiment, to control the variable, $d$ is set to be constant 1 and $h$ is set to be constant 0.3. The initial vector $\mathbf{x}_0 \in [-0.25, 0.25]^4$ is a vector which is generated randomly. Besides, the following two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are generated by $\mathbf{x}_0$ according to the Newton iteration, for its structure is the most basic. The formula of Newton iteration is shown in equation :

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H^{-1}(\mathbf{x}_k, t_k)\psi(\mathbf{x}_k, t_k). \tag{13}$$

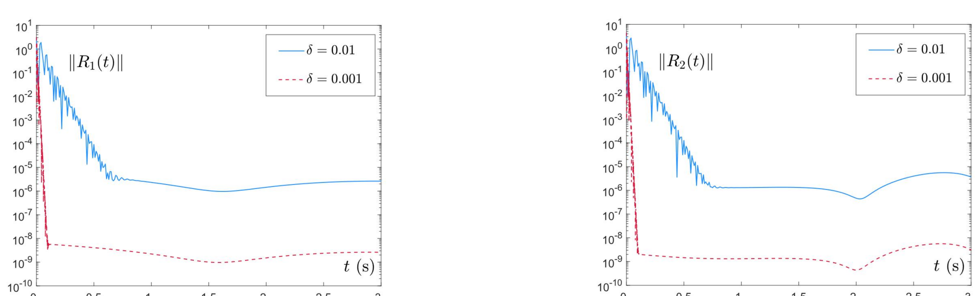
After that, $\mathbf{x}_4$ and following neural states are generated according to equation . To study the change of error with the change of $\delta$ ,the value of $\delta$ is set to be 0.01 and 0.001 respectively, and applied in the iteration process of $f_1(\mathbf{x}_k, t_k)$ and $f_2(\mathbf{x}_k, t_k)$ for comparison. In Fig. 1, the residual error is displayed, in which $t = (k+1)\delta$ . $\|R_1(t)\|$ and $\|R_2(t)\|$ are defined as

$$\|R_1(t)\| = \psi_1(\mathbf{x}(t), t) = \frac{\partial f_1(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}$$

and

$$\|R_2(t)\| = \psi_2(\mathbf{x}(t), t) = \frac{\partial f_2(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)}.$$

Fig. 1 (a) shows the semilog plot of the errors of $f_1(\mathbf{x}_k, t_k)$ when $\delta$ changes. Additionally, Fig1. (b) displays a semilog plot showing the error of $f_2(\mathbf{x}_k, t_k)$ when $\delta$ changes. From Fig. 1, it can be observed that with $t$ increasing, residual error decreases faster when $\delta = 0.001$ than when $\delta = 0.01$. From equation , it is clear that the order of the residual error is $O(\delta^3)$ . Besides, from Fig. 1, it is apparent that the residual error associated with $\delta = 0.001$ is $1/1000$ of the error associated with $\delta = 0.01$ when the curve tends to be flat and $t$ is established. Therefore, the order of residual error in equation can be proved.



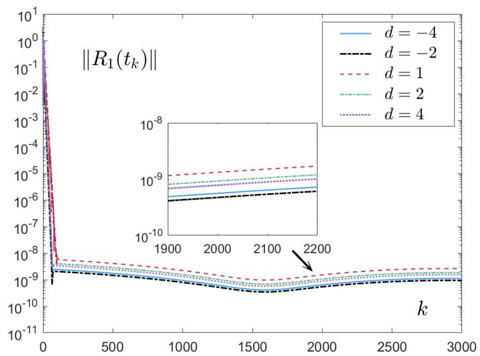

(a) Semilog plot of $\|R_1(t)\|$ with $\delta = 0.01$ and $\delta = 0.001$ (b) Semilog plot of $\|R_2(t)\|$ with $\delta = 0.01$ and $\delta = 0.001$

Fig. 1: Semilog plot of residual errors of two time-varying signal $f_1(\mathbf{x}_k, t_k)$ and $f_2(\mathbf{x}_k, t_k)$ , where $t = (k+1)\delta$ and $h = 0.3$ , using formula shown in equation .

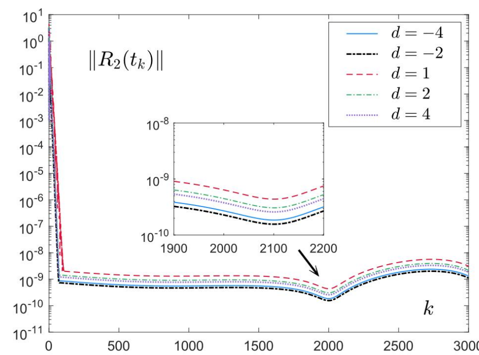
### 3.2 Second Experiment

The second experiment is aimed to study the variation of residual error with $d$ . Same as the first experiment, a vector named $\mathbf{x}_0 \in [-0.25, 0.25]^4$ is produced at random. In addition, $\mathbf{x}_0$ generates the other two vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$ , in accordance with the Newton iteration. After that, equation produces $\mathbf{x}_4$ and the subsequent neural states.

In contrast to the first experiment, to control variables, $\delta$ is fixed to be constant at 0.001 and $h$ is set to be constant at 0.3 in the second experiment. We no longer take $t$ as the horizontal axis, but $k$ , which is the update index. From Lemma 1, the area in which parameter $d$ is effective is $d \in (-\infty, -1) \cup (1/3, +\infty)$ . Therefore, to ensure the universality of results, we take -4, -2, 1, 2 and 4 respectively as the different values of $d$ . By applying different values of $d$ , different residual error curves are got, shown in Fig. 2.




(a)Semilog plot of $\|R_1(t_k)\|$ with $d = -4, -2, 1, 2,$ and 4




(b) Semilog plot of $\|R_2(t_k)\|$ with $d = -4, -2, 1, 2,$ and 4

Fig. 2: Semilog plot of residual errors of two time-varying signal $f_1(\mathbf{x}_k, t_k)$ and $f_2(\mathbf{x}_k, t_k)$ , where $\delta = 0.001$ and $h = 0.3$ , using formula shown in equation .

The errors of $f_1(\mathbf{x}_k, t_k)$ when $d$ varies is shown in a semilog plot Fig. 2 (a), and the semilog plot of the errors of $f_2(\mathbf{x}_k, t_k)$ when $d$ changes is shown in Fig. 2 (b). In order to better observe the law that error changes with $d$ , the curve corresponding to $d = 1$ can be regarded as a benchmark curve, because when $d = 1$ , the trend of error change is the result of iteration using Taylor formula, shown in equation .

By adjusting the value of $d$ while ensuring that other parameter values remain unchanged, the experimental results show some characteristics of novel DTZNN as follows.

- When $d$ is not taken as $1$ , the error decreases faster than when $d$ is taken as $1$ .
- When $d$ is negative, error decreases faster than when $d$ is positive.
- When is positive, the error drops more quickly the higher the absolute value of $d$ .
- When is negative, the error drops more quickly the smaller the absolute value of $d$ .

The above four properties demonstrate the effectiveness of our proposed method, and the choice of $d$ value will indeed have a significant impact on the final effect. Users can choose $d$ in practical application according to the characteristics of the algorithm, which takes into account the flexibility and effectiveness of the algorithm. For example, according to the experimental results, it is preferred to set $d$ as a negative number and the absolute value is as small as possible during the use of the algorithm.

From the results shown in first experiment and second experiment, it is clear that the order of residual error is $O(\delta^3)$. In addition, the experiment also verified the flexibility of our proposed method. The main process is to specialize the value of $d$ to obtain a fixed Taylor-type DTZNN. Experiments show that as a special case of the novel DTZNN, the Taylor-type DTZNN (in which $d = 1$) shown in equation performs worse than other cases. Such experimental results verify the effectiveness of our idea of focusing on flexibility in the discrete process, and the adjustment of the $d$ value also provides a comparison standard. Therefore, the result shows that the new DTZNN has effectiveness dealing with discrete time-varying minimization problem.

## 4. CONCLUSION

To solve a range of time-varying difficulties in real-world engineering applications, the zeroing neural network (ZNN) was developed. However, these methods do not pay much attention to the flexibility of the discrete process, which will affect the final effect of the model. In order to solve discrete time-varying minimization problems, we have developed a novel discrete-time ZNN (DTZNN) in this paper. To get the novel DTZNN, continuous time-varying minimization problem has been firstly considered. After applying ZNN design formula, we use a different way to discretize CTZNN. For the first time, four-instant discretization formula with an adjustable parameter $d$ has been used. To verify the performance of novel DTZNN, two time-varying mapping functions have been designed as examples. What's more, two numerical experiments are designed to study the properties of the model. In the first experiment, different values of $\delta$ is chosen to observe the order of residual error. In the second numerical experiments, $\delta$ is fixed to be constant at 0.001, and different values of $d$ have been selected respectively to observe its impact on the residual error. The outcomes of numerical experiments prove the order of residual error to be $O(\delta^3)$ in novel DTZNN, and the flexibility of the new DTZNN. To sum up, the numerical experiments demonstrate the usefulness of novel DTZNN in solving discrete time-varying minimization problem.

## REFERENCES


[1] Xu, R., Luo, J., Wang, M., Manipulability optimization for coordinated motion control of multi-arm space robots, IFAC-Papers OnLine, 53(2), pp. 9853-9858 (2020).

[2] Zhang, J., Artificial intelligence and machine learning algorithm optimization applied in health big data digitization, AIAM2021: 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture, pp. 2477-2480 (2021).

[3] Bu. S., Logistics engineering optimization based on machine learning and artificial intelligence technology", Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 40(2), pp. 2505-2516 (2021).

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

[5] Sanjay, C., Srinivasan, K., Analysis of the Zhang neural network and its application for the control of nonlinear dynamical systems, IFAC-PapersOnLine, 54(14), pp. 346-351 (2021).

[6] Zhang, Y., Li, Z., Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints, Physics Letters A, 373(18-19), pp. 1639-1643 (2009).

[7] Jin. L., Zhang, Y., Continuous and discrete Zhang dynamics for real-time varying nonlinear optimization", Numerical Algorithms, 73(1), pp. 115-140 (2016).

[8] Zhang, Y., Fang, Y., Liao, B., Qiao, T., Tan, H., New DTZNN model for future minimization with cube steady-state error pattern using Taylor finite-difference formula, 2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP), pp. 128-133 (2015).

[9] Sun, Z., Shi, T., Jin, L., Zhang, B., Pang, Z., Yu, J., Discrete-time zeroing neural network of O($\tau^4$) pattern for online solving time-varying nonlinear optimization problem: application to manipulator motion generation, Journal of the Franklin Institute, 358(14), pp. 7203-7220 (2021).

[10] Zhang, Y., Jin, L., Guo, D., Yin, Y., Chou, Y., Taylor-type 1-step-ahead numerical differentiation rule for first-order derivative approximation and ZNN discretization, Journal of Computational and Applied Mathematics, 273, pp. 29-40 (2015).

[11] Hu, C., Kang, X., Zhang, Y., Three-step general discrete-time Zhang neural network design and application to time-variant matrix inversion, Neurocomputing, 306, pp. 108-118 (2018).
[12] Yin, Z., Liu, K., Chen, Y., Xue, R., Li, J., Shi, Y., Performance analyses of four-instant discretization formulas with application to generalized-Sylvester-type future matrix equation, IEEE Access, vol. 7, pp. 152258-152266, (2019).
[13] Jin, L., Li, S., Liao, B., Zhang, Z., Zeroing neural networks: a survey", Neurocomputing, vol. 267, pp. 597-604, (2017).