IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 2, 2020, accepted November 16, 2020, date of publication November 19, 2020,
date of current version December 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039253

Design and Validation of Zeroing Neural Network to Solve Time-Varying Algebraic Riccati Equation

HANG LIU[1], **TIE WANG**[1], **AND DONGSHENG GUO**[2], **(Member, IEEE)**
[1]Logistics Management Department, Qiqihar Medical University, Qiqihar 161006, China
[2]College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China

Corresponding author: Tie Wang (qqhr3977@163.com)

This work was supported in part by the Social Science Research Project of Qiqihar under Grant QSX2018-07YB, and in part by the Quanzhou City Science and Technology Program of China under Grant 2018C111R.

ABSTRACT Many control problems require solving the algebraic Riccati equation (ARE). Previous studies have focused more on solving the time-invariant ARE than on solving the time-varying ARE (TVARE). This paper proposes a typical recurrent neural network called zeroing neural network (ZNN) to determine the solution of TVARE. Specifically, the ZNN model, which is formulated as an implicit dynamic equation, is developed by defining an indefinite error function and using an exponential decay formula. Then, such a model is theoretically analyzed and proven to be effective in solving the TVARE. Computer simulation results with two examples also validate the efficacy of the proposed ZNN model.

INDEX TERMS Algebraic Riccati equation, time-varying, zeroing neural network (ZNN), theoretical analysis, simulation validation.

I. INTRODUCTION

In control areas, a typical nonlinear matrix equation termed algebraic Riccati equation (ARE) is frequently encountered and needs to be solved accurately [1]–[7]. At present, AREs with the continuous- and discrete-time forms are playing a remarkable role in many engineering applications of control problems [7]–[11], such as the linear-quadratic-Gaussian and H_∞ control problems. The solution of ARE is generally utilized as an essential part of the solution to the aforementioned control problems. For example, the positive definite solution of ARE has been used in each iteration of the homotopy algorithms for the fixed-architecture control [12]. The negative definite solution of ARE has been utilized in the Lyapunov equation approach to solve the matrix differential Riccati equation involved in the linear quadratic optimal control [13]. Thus, an effective algorithm/model for solving the ARE is worth designing and investigating due to its important role.

Several direct algorithms have been reported to solve the continuous- and discrete-time AREs [1]–[4], [14]–[19], such as the Schur and the generalized eigenvalue algorithms. However, computational disadvantages exist for the direct algorithms in several situations [20]. Different iterative algorithms for solving AREs that differ from the direct algorithms have been designed and investigated [4], [8]–[11], [20]–[25]. For example, the Kleinman algorithm, which is an effective Newton algorithm, has been developed to solve the continuous-time ARE arising in H_2 control [21]. The Lanzon algorithm has been provided for solving the continuous-time ARE arising in H_∞ control [20]. An accelerated iterative algorithm has been proposed and studied in [8] to obtain the positive definite solution of the discrete-time ARE. Such algorithm can also be used to compute the negative definite solution of the discrete-time ARE (once it exists). In addition to such iterative algorithms, the neurocomputing models have been developed to find the (positive/negative definite) solution of ARE [26], [27]. Notably, these algorithms and models have been established intrinsically to solve the time-invariant AREs, where ‘‘time-invariant’’ indicates that the coefficient matrices of ARE are constant.

In [28], the time-varying ARE (TVARE) was mentioned and presented for the first time, of which the positive definite time-varying solution was used to design the feedback controller for slowly varying linear system. However, to the best of the authors' knowledge, nearly no research result on seeking the solution of TVARE exists to date. That is, the study of solving TVARE is rare despite its importance in the control of time-varying systems. In general, the TVARE can be treated as a time-invariant ARE within a small time under the assumption of short-time invariance. Then, the

The associate editor coordinating the review of this manuscript and approving it for publication was Giambattista Gruosso.

aforementioned algorithms and models can be used to compute the solution of TVARE at each single time instant. However, this method of solving TVARE may fail to perform efficiently. In other words, the effectiveness of these algorithms and models in solving the TVARE cannot be guaranteed. Thus, developing and investigating an effective algorithm/model to solve the TVARE directly are imperative.

In this paper, motivated by the inspiring work [28] and considering the advances in neural network [29]–[31], we provide a new and efficient model to determine the solution of TVARE. Specifically, with the definition of an indefinite error function and the usage of an exponential decay formula [29]–[31], a typical recurrent neural network called zeroing neural network (ZNN) formulated into an implicit dynamic equation is developed and studied. Then, the convergence property of the proposed ZNN model is theoretically analyzed and proven. Specifically, the model can effectively solve the TVARE. Computer simulation results with two examples also validate the efficacy of the proposed ZNN model.

The remainder of the paper is organized as follows. Section II presents the preliminary of solving the TVARE. Section III describes the proposed ZNN model for solving the TVARE. Theoretical analyses on the model convergence are also discussed. Section IV shows the simulation results by using the proposed model. Section V provides the final remarks. The main contributions of this study are summarized and presented as follows.

1) The ZNN model is proposed and studied for solving the TVARE. The proposed model has not been reported and investigated in the existing literature.
2) This paper theoretically analyzes and proves that the proposed ZNN model can generate an exact solution of TVARE. Computer simulations are also conducted to demonstrate the efficacy of the proposed model.
3) This paper is the first attempt to provide an effective model for computing the solution of the TVARE. This improvement is crucial because it shows a potential for designing different models to solve additional (time-invariant and/or time-varying) nonlinear matrix equations/inequalities.

II. PROBLEM STATEMENT

Consider the following time-varying linear system [28]:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \tag{1}$$

where $u(t) \in R^m$ is the input or control vector and $x(t) \in R^n$ is the state vector with $\dot{x}(t)$ as its time derivative. The coefficient matrices $A(t) \in R^{n\times n}$ and $B(t) \in R^{n\times m}$ are assumed to be bounded at any time instant $t \geqslant 0$. In addition, their time derivatives, i.e., $\dot{A}(t)$ and $\dot{B}(t)$, are bounded at time $t \geqslant 0$.

In [28], the state feedback control of the system (1) is investigated by considering the control $u(t) = -B^{\mathrm{T}}(t)P(t)x(t)$, where $P(t) \in R^{n\times n}$ is the positive definite solution of the TVARE as follows:

$$C^{\mathrm{T}}(t)P(t) + P(t)C(t) - P(t)D(t)P(t) + Q(t) = 0. \tag{2}$$

where $C(t) = A(t) + \gamma I \in R^{n\times n}$, $D(t) = B(t)B^{\mathrm{T}}(t) \in R^{n\times n}$, and $Q(t) \in R^{n\times n}$ is a symmetric matrix. In addition, $\gamma > 0 \in R$ is the given parameter, and $I \in R^{n\times n}$ is the identity matrix. Notably, coefficient matrices $C(t)$, $D(t)$, and $Q(t)$ in (2) vary very slowly unlike the dynamics of the time-varying linear system (1).

With regard to the TVARE (2), the set of time-varying solutions can be either finite or infinite [4]. For presentation convenience, we denote $P^*(t) \in R^{n\times n}$ as a theoretical time-varying solution of (2), and $\dot{P}^*(t)$ as its time derivative. To guarantee the existence of $P^*(t)$ and $\dot{P}^*(t)$, the following solvability assumption of (2) is presented [4].

Solvability Assumption: *The time-varying coefficient matrices $D(t)$ and $Q(t)$ in the TVARE (2) satisfy*

$$D(t) = D^{\mathrm{T}}(t), D(t) \geqslant 0, \text{ and } Q(t) = Q^{\mathrm{T}}(t), Q(t) \geqslant 0 \tag{3}$$

at any time instant $t \geqslant 0$.

In this paper, our objective is to develop a new neurodynamic model for determining an exact time-varying solution of the TVARE (2) under the assumption of (3).

III. MAIN RESULTS

To solve the TVARE (2), the indefinite error function $E(t) \in R^{n\times n}$ that must converge to zero is defined as follows:

$$E(t) = C^{\mathrm{T}}(t)P(t) + P(t)C(t) - P(t)D(t)P(t) + Q(t). \tag{4}$$

To make $E(t)$ be convergent to zero, the following decay formula [29]–[31] is utilized:

$$\dot{E}(t) = -\lambda\Phi(E(t)), \tag{5}$$

where $\dot{E}(t)$ is the time derivative of $E(t)$, $\lambda > 0 \in R$ is a design parameter and $\Phi(\cdot) : R^{n\times n} \to R^{n\times n}$ is an activation function array. Notably, each element of $\Phi(\cdot)$, i.e., $\phi(\cdot)$, is a monotonically increasing odd function, such as the linear, power-sigmoid, or hyperbolic-sine function [29]–[31]. Substituting (4) into (5) yields the result as follows:

$$\begin{aligned}
C^{\mathrm{T}}(t)\dot{P}(t) &+ \dot{P}(t)C(t) - \dot{P}(t)D(t)P(t) - P(t)D(t)\dot{P}(t)\\
&= P(t)\dot{D}(t)P(t) - \dot{C}^{\mathrm{T}}(t)P(t) - P(t)\dot{C}(t) - \dot{Q}(t)\\
&\quad -\lambda\Phi\big(C^{\mathrm{T}}(t)P(t) + P(t)C(t) - P(t)D(t)P(t) + Q(t)\big),
\end{aligned}$$

which is further changed to the following formulation:

$$\begin{aligned}
\big(C^{\mathrm{T}}(t) &- P(t)D(t)\big)\dot{P}(t) + \dot{P}(t)\big(C(t) - D(t)P(t)\big)\\
&= \big(P(t)\dot{D}(t) - \dot{C}^{\mathrm{T}}(t)\big)P(t) - P(t)\dot{C}(t) - \dot{Q}(t)\\
&\quad -\lambda\Phi\big((C^{\mathrm{T}}(t) - P(t)D(t))P(t) + P(t)C(t) + Q(t)\big),
\end{aligned} \tag{6}$$

with $\dot{C}(t)$, $\dot{D}(t)$, $\dot{P}(t)$, and $\dot{Q}(t)$ as time derivatives of $C(t)$, $D(t)$, $P(t)$, and $Q(t)$, respectively.

Let us define the following matrices:

$$\begin{aligned}
M(t) &= I \otimes (C^{\mathrm{T}}(t) - P(t)D(t)) \in R^{n^2\times n^2},\\
N(t) &= (C(t) - D(t)P(t))^{\mathrm{T}} \otimes I \in R^{n^2\times n^2},
\end{aligned}$$

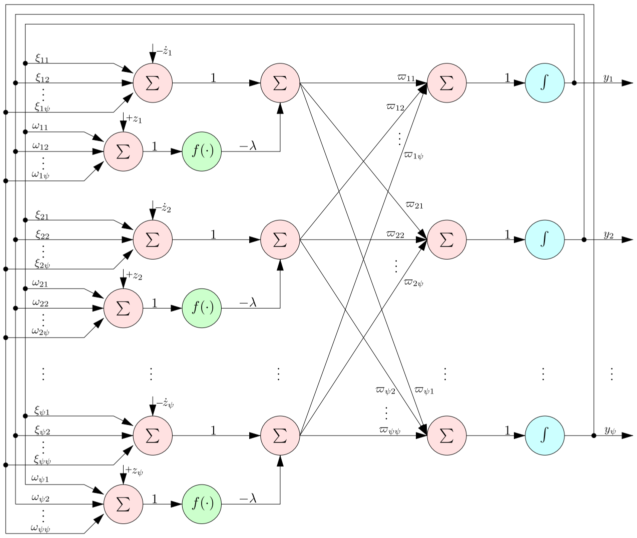

FIGURE 1. Structure of the neurons in the proposed ZNN model (7) for solving the TVARE (2).

$$
\begin{aligned}
U(t) &= I \otimes (P(t)\dot{D}(t) - \dot{C}^{\mathrm{T}}(t)) \in R^{n^2\times n^2},\\
V(t) &= \dot{C}^{\mathrm{T}}(t) \otimes I \in R^{n^2\times n^2},\\
W(t) &= C^{\mathrm{T}}(t) \otimes I \in R^{n^2\times n^2},
\end{aligned}
$$

with symbol $\otimes$ being the Kronecker product [32]. In addition, let vectors $y(t)$ and $z(t)$ be $y(t) = \mathrm{vec}(P(t)) \in R^{n^2}$ and $z(t) = \mathrm{vec}(Q(t)) \in R^{n^2}$ with $\dot{y}(t)$ and $\dot{z}(t)$ corresponding to their time derivatives, where $\mathrm{vec}(\cdot)$ denotes the vectorization operator for a matrix. On the basis of the Kronecker-product and vectorization technique [32], the dynamic equation (6) is thus reformulated as follows:

$$
\begin{aligned}
\big(M(t)+N(t)\big)\dot{y}(t) = \big(U(t)-V(t)\big)y(t) - \dot{z}(t)&\\
-\lambda\mathcal{F}\big((M(t)+W(t))y(t)+z(t)\big),& \qquad (7)
\end{aligned}
$$

where $\mathcal{F}(\cdot): R^{n^2} \to R^{n^2}$ denotes the vectorization of $\Phi(\cdot)$ and is also a monotonically increasing odd function array. Hereafter, (7) is termed the ZNN model that is proposed in this paper for solving the TVARE (2). For better understanding on the structure, the proposed ZNN model (7) is presented in the following ith neuron form:

$$
y_i = \int \sum_{k=1}^{\psi} \varpi_{ik}\Big(\sum_{j=1}^{\psi} \xi_{kj}y_j - \dot{z}_k - \lambda f\Big(\sum_{j=1}^{\psi} \omega_{kj}y_j + z_k\Big)\Big)\mathrm{d}t,
$$

where $\psi = n^2$, y_i is the ith (with $i = 1, 2, \cdots, \psi$) element of (7) and $f(\cdot)$ is a processing element of $\mathcal{F}(\cdot)$. The weights ϖ_{ik}, ξ_{kj}, and ω_{kj} represent the ikth element of the inverse of $(M(t)+N(t))$, the kjth element of $(U(t)-V(t))$, and the kjth element of $((M(t)+W(t))$. The thresholds $\dot{z}_k$ and z_k represent the kth elements of $\dot{z}(t)$ and $z(t)$. Therefore, the structure of (7) is shown in Fig. 1.

With regard to the proposed ZNN model (7), it is depicted in an implicit dynamics [i.e., $(M(t)+N(t))\dot{y}(t) = \cdots$] and employs the derivative information of coefficients [i.e., $\dot{C}(t)$, $\dot{D}(t)$, and $\dot{Q}(t)$]. The effectiveness of (7) in solving (2) is theoretically analyzed via the following theorem.

Theorem: Given a solvable TVARE (2), the neural state of the proposed ZNN model (7) using a suitable activation function array is convergent to an exact solution of (2).

Proof: For the TVARE (2), let us define $P^*(t) \in R^{n\times n}$ as a theoretical solution, which satisfies the following equation:

$$
C^{\mathrm{T}}(t)P^*(t) + P^*(t)C(t) - P^*(t)D(t)P^*(t) + Q(t) = 0. \quad (8)
$$

To lay a basis for further discussion, let matrices $G(t)$ and $G^*(t)$ be $G(t) = P(t)D(t)P(t) \in R^{n\times n}$ and $G^*(t) = P^*(t)D(t)P^*(t) \in R^{n\times n}$. Differentiating (8) yields

$$
\begin{aligned}
\dot{C}^{\mathrm{T}}(t)P^*(t) + C^{\mathrm{T}}(t)\dot{P}^*(t) + \dot{P}^*(t)C(t) + P^*(t)\dot{C}(t)&\\
-\dot{G}^*(t) + \dot{Q}(t) = 0,& \qquad (9)
\end{aligned}
$$

with $\dot{P}^*(t)$ and $\dot{G}^*(t)$ as time derivatives of $P^*(t)$ and $G^*(t)$.

By recalling the previous derivation, the proposed ZNN model (7) is transformed into the following formulation:

$$
\begin{aligned}
&\dot{C}^{\mathrm{T}}(t)P(t) + C^{\mathrm{T}}(t)\dot{P}(t) + \dot{P}(t)C(t) + P(t)\dot{C}(t)\\
&\quad -\dot{G}(t) + \dot{Q}(t) = -\lambda\Phi\big(C^{\mathrm{T}}(t)P(t) + P(t)C(t)\\
&\quad -G(t) + Q(t)\big), \qquad\qquad (10)
\end{aligned}
$$

with $\dot{G}(t)$ as the time derivative of $G(t)$. By substituting (8) and (9) into (10), the following result is obtained:

$$
\begin{aligned}
&\dot{C}^{\mathrm{T}}(t)\big(P(t) - P^*(t)\big) + C^{\mathrm{T}}(t)\big(\dot{P}(t) - \dot{P}^*(t)\big)\\
&\quad +\big(\dot{P}(t) - \dot{P}^*(t)\big)C(t) + \big(P(t) - P^*(t)\big)\dot{C}(t)\\
&\quad -\big(\dot{G}(t) - \dot{G}^*(t)\big) = -\lambda\Phi\big(C^{\mathrm{T}}(t)(P(t) - P^*(t))\\
&\quad +(P(t) - P^*(t))C(t) - (G(t) - G^*(t))\big).
\end{aligned}
$$

By defining the matrix $\tilde{E}(t) = C^{\mathrm{T}}(t)(P(t) - P^*(t)) + (P(t) - P^*(t))C(t) - (G(t) - G^*(t)) \in R^{n\times n}$ with $\dot{\tilde{E}}(t)$ as its time derivative, the above equation is rewritten as follows:

$$
\dot{\tilde{E}}(t) = -\lambda\Phi(\tilde{E}(t)). \quad (11)
$$

As to (11), the ijth differential equation is formulated as

$$
\dot{\tilde{e}}_{ij}(t) = -\lambda\phi(\tilde{e}_{ij}(t)), \quad (12)
$$

with $i, j \in \{1, \cdots, n\}$. Then, the Lyapunov function is defined as $v(t) = \tilde{e}^2_{ij}(t)/2$ to analyze the dynamic system (12). The following time derivative of $v(t)$ is further obtained:

$$
\dot{v}(t) = \tilde{e}_{ij}(t)\dot{\tilde{e}}_{ij}(t) = -\lambda\tilde{e}_{ij}(t)\phi(\tilde{e}_{ij}(t)).
$$

Because $\phi(\cdot)$ is a odd function, we have

$$
\dot{v}(t) = -\lambda\tilde{e}_{ij}(t)\phi(\tilde{e}_{ij}(t)) \begin{cases} = 0, & \text{if } \tilde{e}_{ij}(t) = 0,\\ < 0, & \text{if } \tilde{e}_{ij}(t) \neq 0, \end{cases}
$$

which indicates that $\dot{v}(t)$ is negative definite. On the basis of Lyapunov theory [30], the equilibrium point of (12) is asymptotically stable. Thus, as time $t \to +\infty$, $\tilde{e}_{ij}(t) \to 0$ for all $i, j \in \{1, \cdots, n\}$. This result also denotes that $\tilde{E}(t) = [\tilde{e}_{ij}(t)] \in R^{n\times n}$ in (11) converges to zero as t evolves.

Recalling the definitions of $G(t)$ and $G^*(t)$, we can reformulated $\tilde{E}(t)$ as follows:

$$\tilde{E}(t) = C^{\mathrm{T}}(t)\big(P(t) - P^*(t)\big) + \big(P(t) - P^*(t)\big)C(t) \\ -\big(P(t)D(t)P(t) - P^*(t)D(t)P^*(t)\big). \quad (13)$$

By defining the difference $\Delta(t)$ as $\Delta(t) = P(t) - P^*(t) \in R^{n\times n}$, (13) is changed to the following form:

$$\begin{aligned}\tilde{E}(t) &= C^{\mathrm{T}}(t)\Delta(t) + \Delta(t)C(t) - \big((\Delta(t) + P^*(t))D(t)(\Delta(t) \\ &\quad +P^*(t)) - P^*(t)D(t)P^*(t)\big) \\ &= \big(C^{\mathrm{T}}(t) - P^*(t)D(t)\big)\Delta(t) + \Delta(t)\big(C(t) \\ &\quad -D(t)P^*(t)\big) - \Delta(t)D(t)\Delta(t),\end{aligned}$$

which is similar to the formulation of the TVARE (2).

As analyzed above, $\tilde{E}(t) \to 0$ with $t \to +\infty$. The following result is thus obtained:

$$\lim_{t\to+\infty}\Big(\big(C^{\mathrm{T}}(t) - P^*(t)D(t)\big)\Delta(t) + \Delta(t)\big(C(t) \\ -D(t)P^*(t)\big) - \Delta(t)D(t)\Delta(t)\Big) = 0,$$

which further yields

$$\lim_{t\to+\infty}\Big(\Delta(t)\Big) = 0 \;\Rightarrow\; \lim_{t\to+\infty}\Big(P(t) - P^*(t)\Big) = 0.$$

This result denotes that the state matrix $P(t)$ of (10) will converge to a theoretical solution $P^*(t)$ of the TVARE (2). That is, $P(t) \to P^*(t)$ as time t evolves. Notably, the vectorization form of (10) is exactly the proposed ZNN model (7), wherein the state vector $y(t)$ is the vectorization of $P(t)$; i.e., $y(t) = \mathrm{vec}(P(t))$. By summarizing the above results, it is concluded that the neural state of the proposed ZNN model (7) with appropriate activation function array is convergent to an exact solution of the TVARE (2). The proof is thus completed. ■

The above theorem theoretically guarantees that the proposed ZNN model (7) is effective in solving the TVARE (2). The following propositions are presented to further show the convergence characteristic of (7), where $S^+ \in R^{n\times n}$ denotes a given positive definite matrix and $S^- \in R^{n\times n}$ denotes a given negative definite matrix.

Proposition 1: Given a solvable TVARE (2), the proposed ZNN model (7) possesses the following properties.

1) *The state vector $y(t)$ of (7), which starts from an initial state $y(0)$ selected as $y(0) = \mathrm{vec}(S^+)$, is convergent to the positive definite theoretical solution of (2).*
2) *The state vector $y(t)$ of (7), which starts from an initial state $y(0)$ selected as $y(0) = \mathrm{vec}(S^-)$, is convergent to the negative definite theoretical solution of (2).*

Proposition 2: By selecting different activation function arrays, the proposed ZNN model (7) possesses the following properties [29]–[31].

1) *In the case of selecting the linear activation function array, the exponential convergence with rate λ is achieved for (7), which corresponds to the exponential convergence of $P(t)$ to $P^*(t)$ of (2).*
2) *In the case of selecting the hyperbolic-sine, power-sum, or power-sigmoid activation function array, superior convergence is achieved for (7) in comparison with the linearly-activated ZNN model.*

Remarks: The selection of the initial state $y(0)$ is important to the computation of the proposed ZNN model (7), leading to different computational results of (7). It follows from Proposition 1 that $y(0)$ can be selected according to different situations. For instance, to obtain the positive definite solution of the TVARE (2) involved in the control of slowly-varying systems [28], we can simply set $y(0)$ to be $y(0) = \mathrm{vec}(I)$ (i.e., the vectorization of the identity matrix), or directly set $y(0)$ to be $y(0) = \mathrm{vec}(P^*(0))$, where $P^*(0)$ is determined by using the MATLAB routine "CARE" to solve (2) at $t = 0$ s. Besides, it follows from Proposition 2 that the activation function has a significant effect on the convergence characteristic of (7). In the ensuing simulation part, the linear, hyperbolic-sine, power-sum, and power-sigmoid activation functions presented in previous studies [29]–[31] are employed for investigating the proposed ZNN model (7) to solve the TVARE (2). As a matter of fact, by referring to [33]–[35], designing more different types of activation functions for further enhancing the convergence performance of (7) can be a future research direction.

IV. SIMULATION VALIDATION

In this section, simulation results are presented to validate the efficacy of the proposed ZNN model (7) for TVARE solving. The simulations are performed by using MATLAB R2008a on a digital computer with an Intel(R) Core(TM) i3-3110M @2.40 GHz CPU, 4 GB memory, and Windows 7 OS.

Example 1: In this example, the TVARE (2) is considered with the following matrices:

$$\begin{aligned}C(t) &= \begin{bmatrix}\sin(t) & \cos(t)\\ -\cos(t) & \sin(t)\end{bmatrix} + 5I \\ &= \begin{bmatrix}5+\sin(t) & \cos(t)\\ -\cos(t) & 5+\sin(t)\end{bmatrix}, \\ D(t) &= \begin{bmatrix}d_{11}(t) & 0\\ 0 & d_{22}(t)\end{bmatrix} \text{ and } Q(t) = \begin{bmatrix}q_{11}(t) & 0\\ 0 & q_{22}(t)\end{bmatrix}.\end{aligned}$$

where $d_{11}(t) = (4 + \exp(-t) - \cos(t))^2$, $d_{22}(t) = (2 + 1/(t + 1) + \sin(t))^2$, $q_{11}(t) = 3 + (1/(t + 1))^2 + \sin(t)$, and $q_{22}(t) = 6 + (2 + \exp(-t))^2 - \cos(t)$. This situation corresponds to the time-varying linear system (1) with the coefficients as follows:

$$\begin{aligned}A(t) &= \begin{bmatrix}\sin(t) & \cos(t)\\ -\cos(t) & \sin(t)\end{bmatrix}, \\ B(t) &= \begin{bmatrix}4+\exp(-t)-\cos(t) & 0\\ 0 & 2+1/(t+1)+\sin(t)\end{bmatrix}.\end{aligned}$$

To solve the TVARE (2) with above matrices, the proposed ZNN model (7) is simulated and studied. The related simulation results are shown in Figs. 2 through 5.

Figure 2 presents the results of solving the TVARE (2) via the proposed ZNN model (7) using $\lambda = 1$ and linear activation function array and starting from six initial states selected as $y(0) = \mathrm{vec}(S^+)$. In the figure, the residual error

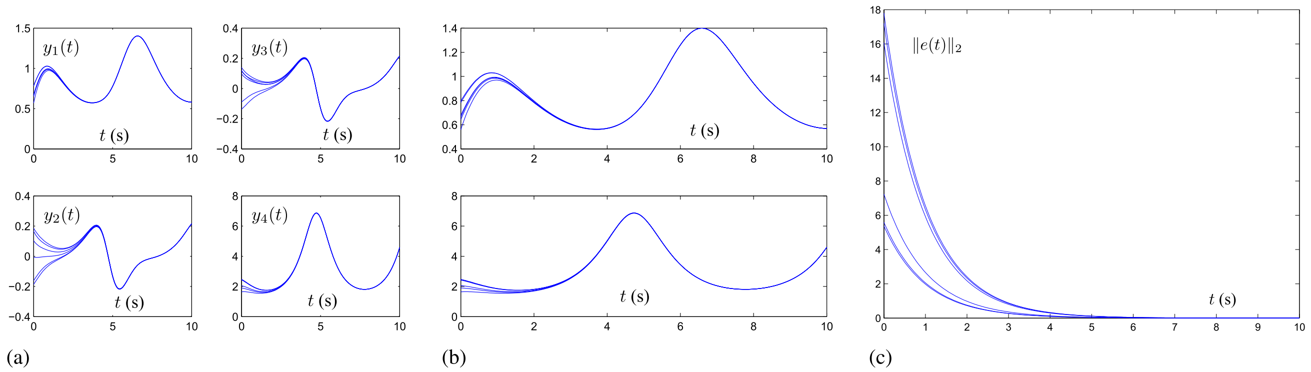

FIGURE 2. **Simulation results of solving the TVARE (2) via the proposed ZNN model (7) using $\lambda = 1$ and linear activation function array and starting from six initial states selected as $y(0) = \mathrm{vec}(S^+)$, where the positive definite matrices (denoted as S^+) are randomly generated. (a) Trajectories of $y(t) = \mathrm{vec}(P(t))$. (b) Eigenvalues of $P(t)$. (c) Residual errors.**

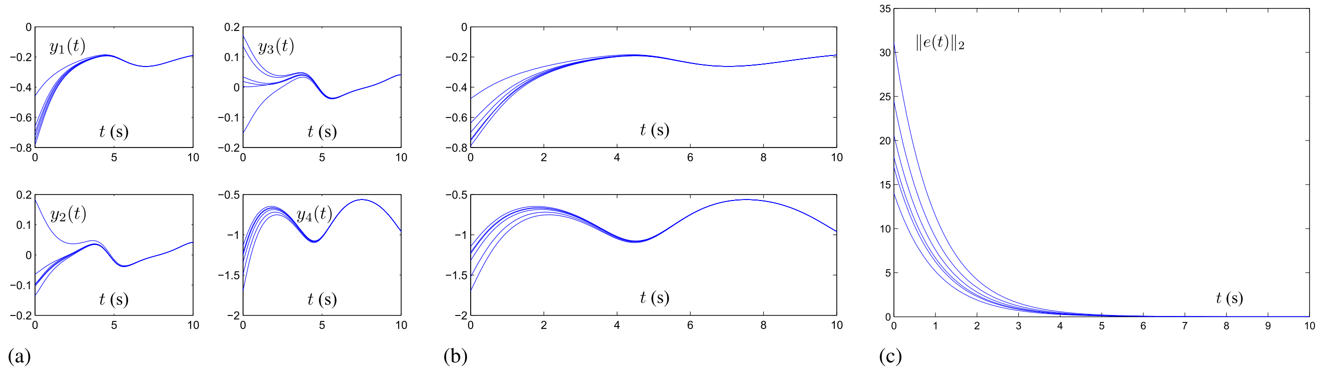

FIGURE 3. **Simulation results of solving the TVARE (2) via the proposed ZNN model (7) using $\lambda = 1$ and linear activation function array and starting from six initial states selected as $y(0) = \mathrm{vec}(S^-)$, where the negative definite matrices (denoted as S^-) are randomly generated. (a) Trajectories of $y(t) = \mathrm{vec}(P(t))$. (b) Eigenvalues of $P(t)$. (c) Residual errors.**

is defined as $\|e(t)\|_2 = \|(M(t) + W(t))y(t) + z(t)\|_2 = \|C^{\mathrm{T}}(t)P(t) + P(t)C(t) - P(t)D(t)P(t) + Q(t)\|_{\mathrm{F}}$ with $\|\cdot\|_2$ and $\|\cdot\|_{\mathrm{F}}$ corresponding to the two norms and the Frobenius norm. As shown in Fig. 2(a), all the state trajectories of (7) converge to a time-varying solution. Fig. 2(b) denotes that such a solution is positive definite in terms of its eigenvalues being greater than zero. In addition, as seen from Fig. 2(c), all the residual errors of (7) are convergent to zero, thereby indicating that the positive definite solution computed by (7) satisfies $C^{\mathrm{T}}(t)P(t) + P(t)C(t) - P(t)D(t)P(t) + Q(t) = 0$. These simulation results verify that the proposed ZNN model (7) with $y(0) = \mathrm{vec}(S^+)$ can generate an exact positive definite solution of the TVARE (2).

Figure 3 shows the results of solving the TVARE (2) via the proposed ZNN model (7) using $\lambda = 1$ and linear activation function array and starting from six initial states selected as $y(0) = \mathrm{vec}(S^-)$. As presented in Fig. 3(a) and (b), all the state trajectories of (7) are convergent to a time-varying negative definite solution (as its eigenvalues are smaller than zero). Fig. 3(c) indicates that all the residual errors of (7) converge to zero, thereby showing that the negative definite solution computed by (7) also satisfies $C^{\mathrm{T}}(t)P(t)+P(t)C(t)-P(t)D(t)P(t) + Q(t) = 0$. These simulation results verify that the proposed ZNN model (7) with $y(0) = \mathrm{vec}(S^-)$ can generate an exact negative definite solution of the TVARE (2).

Figure 4 illustrates the results of solving the TVARE (2) via the proposed ZNN model (7) by increasing the value of λ (i.e., from 1 to 10). As seen from the figure, 1) starting from $y(0) = \mathrm{vec}(S^+)$, the state trajectory of (7) converges to the positive definite solution; and 2) starting from $y(0) = \mathrm{vec}(S^-)$, the state trajectory of (7) converges to the negative definite solution. These phenomena coincide with the theoretical results given in Proposition 1. Notably, the solutions at steady-state in Fig. 4 are the same as those in Figs. 2(a) and 3(a), thus meaning that such solutions computed by (7) also make $C^{\mathrm{T}}(t)P(t) + P(t)C(t) - P(t)D(t)P(t) + Q(t) = 0$ hold true. Fig. 4 further shows that all the residual errors of (7) converge more rapidly, and the steady-state errors are in the orders of 10^{-5} and 10^{-6}. These simulation results indicate that the proposed ZNN model (7) is effective in solving the TVARE (2). Moreover, by comparing Fig. 4 with Figs. 2 and 3, we find that, with the increase of λ, the convergence performance of (7) will be enhanced (in terms of its convergence time being shortened by increasing the λ value). Thus, for the proposed ZNN model (7), λ should be selected as a sufficiently large value to meet the practical requirement.

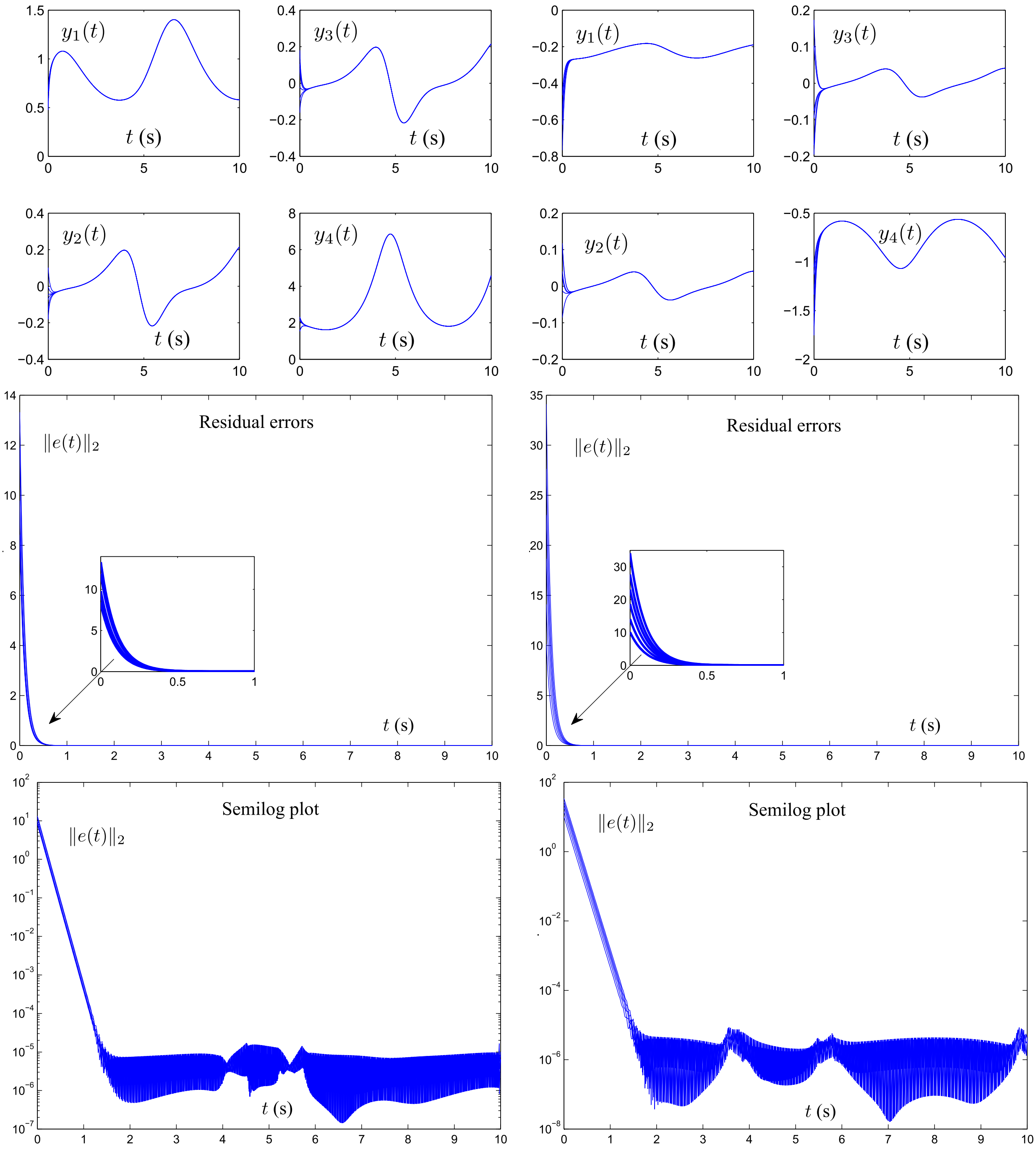

FIGURE 4. Simulation results of solving the TVARE (2) via the proposed ZNN model (7) using $\lambda = 10$ and linear activation function array and starting from twelve randomly-generated initial states, with the fist six initial states selected as $y(0) = \mathrm{vec}(S^{+})$ and the rest selected as $y(0) = \mathrm{vec}(S^{-})$.

For further investigation, the proposed ZNN model (7) is simulated by using $y(0) = [1, 0, 0, 2]^{\mathrm{T}}$, different values of λ, and different activation function arrays. The related simulation results are presented in Fig. 5, and they denote that (7) effectively solves the TVARE (2) in terms of the convergence of residual errors. Moreover, from Figs. 2 through 5, the observations are perceived as follows, which coincide with the theoretical results given in Proposition 2.

- The convergence time for (7) is shortened by 10 times as λ increases by 10 times, thus denoting that (7) possesses the property of exponential convergence.
- Increasing λ effectively enhances the convergence performance of (7). thus indicating again the importance of λ in (7) for TVARE solving.
- By selecting the hyperbolic-sine, power-sum, and power-sigmoid activation function arrays, (7) possesses better convergence that the linearly-activated model.

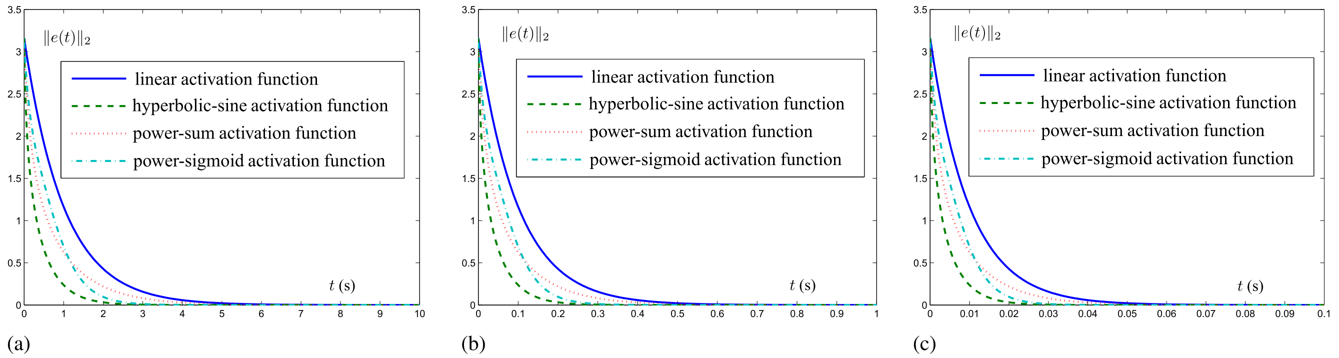

FIGURE 5. Residual errors of solving the TVARE (2) via the proposed ZNN model (7) using $y(0) = [1, 0, 0, 2]^{\mathrm{T}}$, different values of λ, and different activation function arrays. (a) Using $\lambda = 1$. (b) Using $\lambda = 10$. (c) Using $\lambda = 100$.

On the basis of these observations, increasing the value of λ and selecting a suitable activation function array result in superior performance of (7).

In summary, Figs. 2 through 5 verify the efficacy of the proposed ZNN model (7) for solving the TVARE (2).

Example 2: When the coefficients in (2) are the constant ones, the TVARE reduces to a time-invariant ARE. In this example, the proposed ZNN model (7) is investigated to solve the TVARE with the following constant matrices [27]:

$$C(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix},$$

$$Q(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.5 \end{bmatrix},$$

$$\text{and } D(t) = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 3.3333 & 1 \\ 0 & 0 & 14.2857 \end{bmatrix}.$$

For solving such a (time-invariant) ARE, the MATLAB routine "CARE" is employed, and the solution is obtained as follows:

$$\mathrm{vec}(P_{\mathrm{care}}) = \begin{bmatrix} 0.332420792902192 \\ 0.109354093718442 \\ -0.012253454448820 \\ 0.109354093718442 \\ 0.378968000158956 \\ -0.005913282457315 \\ -0.012253454448820 \\ -0.005913282457315 \\ 0.038757975507853 \end{bmatrix} \in R^9.$$

By using and simulating the proposed ZNN model (7) to solve the TVARE (2) with above constant matrices (Notably, in this situation, $\dot{C}(t) = \dot{D}(t) = \dot{Q}(t) = 0$), the simulation results are presented in Figs. 6 and 7.

Figure 6 shows the results of solving the time-invariant ARE (i.e., (2) with constant matrices) via the proposed ZNN model (7) using $\lambda = 1$ and linear activation function array and starting from the initial state selected as $y(0) = \mathrm{vec}(0.5I)$. As shown in Fig. 6(a) and (b), the state trajectory of (7) is convergent to the positive definite solution (as its eigenvalue is greater than zero). Furthermore, the residual error in Fig. 6(c) converges to zero, showing that the solution in Fig. 6(a) satisfies $C^{\mathrm{T}}(t)P(t) + P(t)C(t) - P(t)D(t)P(t) + Q(t) = 0$. Notably, the solution computed by (7) at $t = 10$ s is given as follows:

$$\mathrm{vec}(P_{\mathrm{neuro}}) = y(t = 10) = \begin{bmatrix} 0.332436225382166 \\ 0.109359112049686 \\ -0.012256722725030 \\ 0.109359112049686 \\ 0.379027247306223 \\ -0.005918968020323 \\ -0.012256722725030 \\ -0.005918968020323 \\ 0.038788489811289 \end{bmatrix}.$$

Thus, the error between $\mathrm{vec}(P_{\mathrm{care}})$ and $\mathrm{vec}(P_{\mathrm{neuro}})$ is obtained as follows:

$$|\mathrm{vec}(P_{\mathrm{care}}) - \mathrm{vec}(P_{\mathrm{neuro}})| = \begin{bmatrix} 1.54324799739869 \\ 0.50183312446106 \\ 0.32682762102828 \\ 0.50183312445967 \\ 5.92471472662948 \\ 0.56855630077134 \\ 0.32682762102811 \\ 0.56855630077186 \\ 3.05143034359245 \end{bmatrix} \times 10^{-5},$$

where symbol $|\cdot|$ returns to the absolute value for each element of the vector. This small error indicates that the solution via the MATLAB routine "CARE" and the solution via the proposed ZNN model (7) are almost the same as each other. These results verify the efficacy of (7) for solving the time-invariant ARE (being a special case of the TVARE (2)).

Figure 7 presents the results of solving the time-invariant ARE via the proposed ZNN model (7) by changing the value of λ from 1 to 10, which denotes the model effectiveness for time-invariant ARE solving. That is, the state trajectory of (7) converges to the positive definite solution rapidly and the corresponding residual error converges to zero. In addition, comparing Fig. 6 with Fig. 7 shows that the convergence performance of (7) is enhanced with the increase of λ. Besides, the following solution computed by (7) at $t = 10$ s is

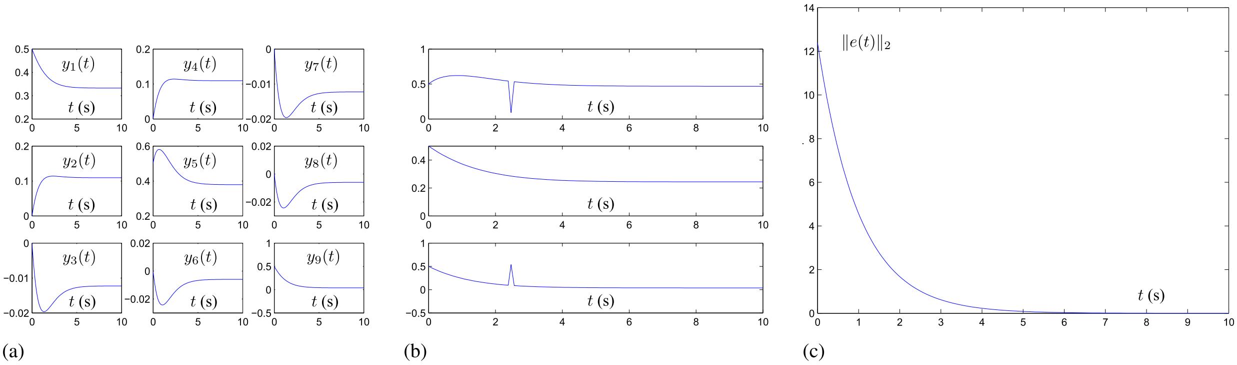

FIGURE 6. Simulation results of solving the time-invariant ARE (i.e., (2) with constant coefficient matrices) via the proposed ZNN model (7) using $\lambda = 1$ and linear activation function array and starting from the initial state selected as $y(0) = \mathrm{vec}(0.5I)$. (a) Trajectory of $y(t) = \mathrm{vec}(P(t))$. (b) Eigenvalue of $P(t)$. (c) Residual error.

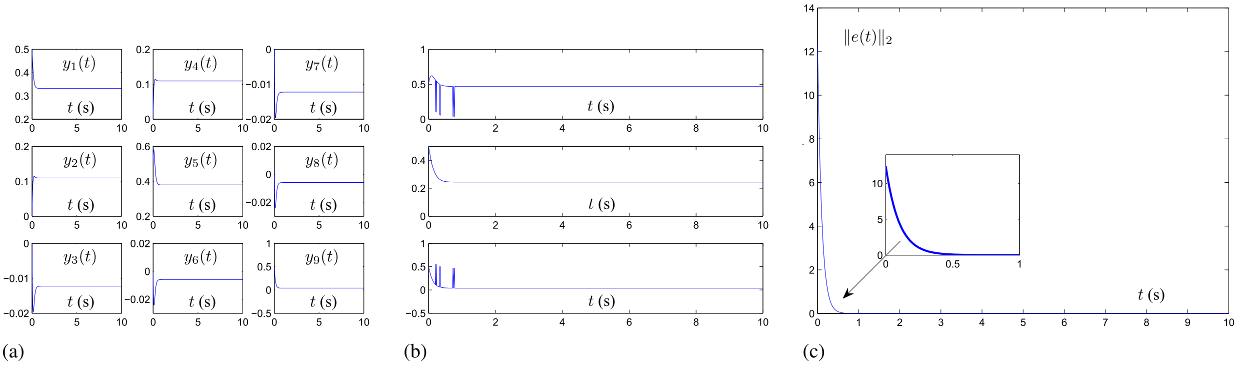

FIGURE 7. Simulation results of solving the time-invariant ARE via the proposed ZNN model (7) using $\lambda = 10$ and linear activation function array and starting from the initial state selected as $y(0) = \mathrm{vec}(0.5I)$. (a) Trajectory of $y(t) = \mathrm{vec}(P(t))$. (b) Eigenvalue of $P(t)$. (c) Residual error.

provided:

$$\mathrm{vec}(P_{\mathrm{neuro}}) = y(t = 10) = \begin{bmatrix} 0.332420818173419 \\ 0.109354101949185 \\ -0.012253459803822 \\ 0.109354101949184 \\ 0.378968097227928 \\ -0.005913291768627 \\ -0.012253459803821 \\ -0.005913291768627 \\ 0.038758025461878 \end{bmatrix}.$$

This solution is quite similar to the solution via the MATLAB routine "CARE", with their small difference/error being presented as follows:

$$|\mathrm{vec}(P_{\mathrm{care}}) - \mathrm{vec}(P_{\mathrm{neuro}})| = \begin{bmatrix} 2.52712262782673 \\ 0.82307429483119 \\ 0.53550017198312 \\ 0.82307428372896 \\ 9.70689716939255 \\ 0.93113120946259 \\ 0.53550016781978 \\ 0.93113120512578 \\ 4.99540243059649 \end{bmatrix} \times 10^{-8}.$$

Evidently, this error is smaller than the previous one, i.e., 10^{-8} versus 10^{-5}. This means that by increasing the value of λ, the time-invariant ARE can be solved more accurately via the proposed ZNN model (7). These results indicate again that the proposed ZNN model (7) is effective in solving the time-invariant ARE and that the design parameter λ, which should be selected as a sufficiently large value, is an important factor affecting the performance of (7).

In summary, Figs. 6 and 7 verify that the proposed ZNN model (7) can generate an exact positive definite solution of the time-invariant ARE, further showing the efficacy of (7) for solving the TVARE (2).

V. CONCLUSION

In this paper, the ZNN model (7) was proposed and studied to solve the TVARE (2). Such a model which is depicted in an implicit dynamics and employs the derivative information of coefficients was theoretically proven to be capable of generating an exact solution of (2). Simulation results further verified the efficacy of the proposed ZNN model (7) for TVARE solving. By following this paper, the proposed ZNN model (7) will be further investigated for the control of slowly-varying systems [28]. Besides, designing a new ZNN model for solving the TVARE (2) in noisy environments [31] will be another future research direction.

ACKNOWLEDGMENT

The authors thank the editors and anonymous reviewers for the time and effort spent in handling this paper and constructive comments provided to improve the presentation and quality.

REFERENCES

[1] S. Bittanti, A. J. Laub, and J. C. Willems, *The Riccati Equation*. Berlin, Germany: Springer-Verlag, 1991.

[2] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Oxford, U.K.: Clarendon Press, 1995.

[3] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory*. Basel, Switzerland: Birkhäuser, 2003.

[4] W. S. Levine, *The Control Systems Handbook: Control System Advanced Methods*. New York, NY, USA: CRC Press, 2010.

[5] A. Ferrante and L. Ntogramatzidis, "The generalised discrete algebraic Riccati equation in linear-quadratic optimal control," *Automatica*, vol. 49, no. 2, pp. 471–478, Feb. 2013.

[6] I. R. Petersen, "A survey of Riccati equation results in negative imaginary systems theory and quantum control theory," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9561–9566, Jul. 2017.

[7] A. Ferrante and L. Ntogramatzidis, "On the reduction of the continuous-time generalized algebraic Riccati equation: An effective procedure for solving the singular LQ problem with smooth solutions," *Automatica*, vol. 93, pp. 554–558, Jul. 2018.

[8] M. M. Lin and C.-Y. Chiang, "An accelerated technique for solving one type of discrete-time algebraic Riccati equations," *J. Comput. Appl. Math.*, vol. 338, pp. 91–110, Aug. 2018.

[9] M. A. Hernández-Verón and N. Romero, "Existence, localization and approximation of solution of symmetric algebraic Riccati equations," *Comput. Math. Appl.*, vol. 76, no. 1, pp. 187–203, Jul. 2018.

[10] A.-G. Wu, H.-J. Sun, and Y. Zhang, "Two iterative algorithms for stochastic algebraic Riccati matrix equations," *Appl. Math. Comput.*, vol. 339, pp. 410–421, Dec. 2018.

[11] S. Miyajima, "Verified computation for the hermitian positive definite solution of the conjugate discrete-time algebraic Riccati equation," *J. Comput. Appl. Math.*, vol. 350, pp. 80–86, Apr. 2019.

[12] E. G. Collins, W. M. Haddad, and L. T. Watson, "Fixed-architecture, robust control design using fixed-structure multipliers," *IFAC Proc. Volumes*, vol. 29, no. 1, pp. 1056–1061, Jun. 1996.

[13] Z. Gajic, "Solving the matrix differential Riccati equation: A Lyapunov equation approach," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 191–194, Jan. 2010.

[14] A. Laub, "A Schur method for solving algebraic Riccati equations," *IEEE Trans. Autom. Control*, vol. AC-24, no. 6, pp. 913–921, Dec. 1979.

[15] S. L. Osburn and D. S. Bernstein, "A generalized eigenvalue problem for solving the discrete-time Riccati equation with singular dynamics and singular measurement noise," in *Proc. Amer. Control Conf. (ACC)*, 1994, pp. 1155–1156.

[16] A. J. Laub and P. Gahinet, "Numerical improvements for solving Riccati equations," *IEEE Trans. Autom. Control*, vol. 42, no. 9, pp. 1303–1308, 1997.

[17] B. N. Datta, *Numerical Methods for Linear Control Systems*. New York, NY, USA: Academic, 2004.

[18] T. Haqiri and F. Poloni, "Methods for verified stabilizing solutions to continuous-time algebraic Riccati equations," *J. Comput. Appl. Math.*, vol. 313, pp. 515–535, Mar. 2017.

[19] A. S. A. Dilip, H. K. Pillai, and R. M. Jungers, "On discrete algebraic Riccati equations: A rank characterization of solutions," *Linear Algebra Appl.*, vol. 527, pp. 184–215, Aug. 2017.

[20] A. Lanzon, Y. Feng, B. D. O. Anderson, and M. Rotkowitz, "Computing the positive stabilizing solution to algebraic Riccati equations with an indefinite quadratic term via a recursive method," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2280–2291, Nov. 2008.

[21] D. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Trans. Autom. Control*, vol. AC-13, no. 1, pp. 114–115, Feb. 1968.

[22] L. Dieci, "Some numerical considerations and Newton's method revisited for solving algebraic Riccati equations," *IEEE Trans. Autom. Control*, vol. 36, no. 5, pp. 608–616, May 1991.

[23] P. Benner and R. Byers, "An exact line search method for solving generalized continuous-time algebraic Riccati equations," *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 101–107, 1998.

[24] R. Z. Anjani and A. Suparwanto, "Newton modification for solving discrete algebraic Riccati equation," in *Proc. 5th Int. Conf. Instrum., Control, Autom. (ICA)*, Aug. 2017, pp. 104–108.

[25] S. Miyajima, "Fast verified computation for stabilizing solutions of discrete-time algebraic Riccati equations," *J. Comput. Appl. Math.*, vol. 319, pp. 352–364, Aug. 2017.

[26] L.-X. Wang and J. M. Mendel, "Parallel structured networks for solving a wide variety of matrix algebra problems," *J. Parallel Distrib. Comput.*, vol. 14, no. 3, pp. 236–247, Mar. 1992.

[27] F. M. Ham and E. G. Collins, "A neurocomputing approach for solving the algebraic matrix Riccati equation," in *Proc. Int. Conf. Neural Netw. (ICNN)*, 1996, pp. 617–622.

[28] E. W. Kamen, P. P. Khargonekar, and A. Tannenbaum, "Control of slowly-varying linear systems," *IEEE Trans. Autom. Control*, vol. 34, no. 12, pp. 1283–1285, 1989.

[29] Y. Zhang and S. S. Ge, "Design and analysis of a general recurrent neural network model for time-varying matrix inversion," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1477–1490, Nov. 2005.

[30] L. Jin, S. Li, B. Liao, and Z. Zhang, "Zeroing neural networks: A survey," *Neurocomputing*, vol. 267, pp. 597–604, Dec. 2017.

[31] D. Guo, S. Li, and P. S. Stanimirovic, "Analysis and application of modified ZNN design with robustness against harmonic noise," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4627–4638, Jul. 2020.

[32] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[33] L. Jin, S. Li, H. Wang, and Z. Zhang, "Nonconvex projection activated zeroing neurodynamic models for time-varying matrix pseudoinversion with accelerated finite-time convergence," *Appl. Soft Comput.*, vol. 62, pp. 840–850, Jan. 2018.

[34] L. Xiao, Z. Zhang, and S. Li, "Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 11, pp. 2210–2220, Nov. 2019.

[35] W. Li, L. Xiao, and B. Liao, "A finite-time convergent and noise-rejection recurrent neural network and its discretization for dynamic nonlinear equations solving," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3195–3207, Jul. 2020.

HANG LIU received the B.E. degree in electrical engineering and automation from the Harbin University of Science and Technology, China, in 2011. He is currently pursuing the M.S. degree in control engineering with the College of Information Science and Engineering, Qiqihar University, Heilongjiang, China.

He is currently an Assistant Professor with the Logistics Management Department, Qiqihar Medical University, Heilongjiang. His current research interests include neural networks and robot control.

TIE WANG received the B.S. degree from Qiqihar University, Heilongjiang, China, in 2005, and the M.S. degree from Jilin University, Jilin, China, in 2010.

He is currently an Associate Researcher with the Logistics Management Department, Qiqihar Medical University, Heilongjiang. His current research interests include neural networks, information processing, and intelligent reform.

DONGSHENG GUO (Member, IEEE) received the B.S. degree in automation from Sun Yat-sen University, Guangzhou, China, in 2010, and the Ph.D. degree in communication and information systems from the School of Information Science and Technology, Sun Yat-sen University, in 2015.

He is currently an Associate Professor with the College of Information Science and Engineering, Huaqiao University, Xiamen, China. He is also with the Fujian Engineering Research Center of Motor Control and System Optimal Schedule, Xiamen, for cooperative research. His current research interests include neural networks, robot control, and numerical methods.