

Article

Recurrent Neural Network Models Based on Optimization Methods

Predrag S. Stanimirović ^{1,2,*} , Spyridon D. Mourtas ^{2,3} , Vasilios N. Katsikis ³ , Lev A. Kazakovtsev ² 
and Vladimir N. Krutikov ⁴¹ Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia² Laboratory “Hybrid Methods of Modelling and Optimization in Complex Systems”, Siberian Federal University, Krasnoyarsk 660041, Russia³ Department of Economics, Division of Mathematics and Informatics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece⁴ Department of Applied Mathematics, Kemerovo State University, Krasnaya Street 6, Kemerovo 650043, Russia

* Correspondence: pecko@pmf.ni.ac.rs

Abstract: Many researchers have addressed problems involving time-varying (TV) general linear matrix equations (GLMEs) because of their importance in science and engineering. This research discusses and solves the topic of solving TV GLME using the zeroing neural network (ZNN) design. Five new ZNN models based on novel error functions arising from gradient-descent and Newton optimization methods are presented and compared to each other and to the standard ZNN design. Pseudoinversion is involved in four proposed ZNN models, while three of them are related to Newton’s optimization method. Heterogeneous numerical examples show that all models successfully solve TV GLMEs, although their effectiveness varies and depends on the input matrix.

Keywords: linear matrix equations; zeroing neural network; dynamic system; optimization methods

MSC: 65F20; 68T05



Citation: Stanimirović, P.S.;

Mourtas, S.D.; Katsikis, V.N.;

Kazakovtsev, L.A.; Krutikov, V.N.

Recurrent Neural Network Models

Based on Optimization Methods.

Mathematics **2022**, *10*, 4292. <https://doi.org/10.3390/math10224292>

Academic Editor: Danilo Costarelli

Received: 13 October 2022

Accepted: 14 November 2022

Published: 16 November 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction and Preliminaries

Novel zeroing neural network (ZNN) dynamical systems are presented using various error functions that arise from gradient-descent and Newton optimization methods. Introduced models are theoretically investigated and compared to each other and to the standard ZNN design. The proposed ZNN models are applied in solving time-varying (TV) general linear matrix equations (GLMEs) with arbitrary TV real input matrices. The TV GLME problem is expressed as the matrix equation

$$A(t)Y(t)C(t) = B(t), A(t) \in \mathbb{R}^{m \times n}, Y(t) \in \mathbb{R}^{n \times k}, C(t) \in \mathbb{R}^{k \times h}, B(t) \in \mathbb{R}^{m \times h}, \quad (1)$$
$$m \geq n, k \leq h,$$

in which $Y(t)$ is unknown matrix.

The solutions $Y(t)$ to the Equation (1) are extremely useful in science as well as in a wide range of engineering applications, including optimizing the manipulability of a robotic arm’s joint angles [1], making highly accurate predictions for missing quality of service data [2], tracking robotic motion and mobile objects [3], obtaining weights of feed-forward neural networks [4]. This paper presents and contrasts six ZNN models based on two new error functions for solving TV GLMEs that involve arbitrary matrices $A(t)$, $B(t)$, and $C(t)$. It is worth mentioning that the matrix pseudoinversion is involved in four of the proposed ZNN models, while three are related to Newton’s optimization method.

The ZNN concept was coined by Zhang et al. in [5] for determining online solutions to TV matrix inversion problems. Most ZNN-based dynamical systems belong to classes of

recurrent neural networks (RNNs) designed to approximate zeros of matrix, vector, or scalar equations. Besides ZNN, there is another type of RBBs, which is known as Gradient neural networks (GNN). As an outcome, numerous significant research results have been revealed in the scientific literature. ZNN and GNN dynamics for solving linear matrix equations of the form (1) were described and compared in [6]. Two Zhang functions and initiated ZNN dynamics for solving the TV GLME $A(t)Y(t) = B(t)$ were presented and investigated theoretically and numerically in [7]. A gradient-based neural dynamics for solving the matrix equation $AXB = D$ was investigated in [8]. Gradient and zeroing dynamical systems aimed to solve the system of linear equations in both the TV and constant environment are investigated in [9–11]. GNN neural-dynamic schemes are substantially designed for solving equations with time-invariant coefficient matrices [11]. A global overview of Zeroing dynamical systems (ZND), involving ZND aimed to solving linear matrix equations, was presented in [12]. Nonlinear Zeroing neural dynamical systems with finite-time convergence and noise-tolerant for solving linear matrix equations in time-varying scenarios were proposed and investigated in [13,14]. A ZNN design whose dynamics are based on varying gain parameter and which are suitable for solving TV GLME was proposed in [15]. A very popular class of ZNN dynamical systems is represented by ZNN models with a finite-time or pre-defined convergence time. A finite-time convergent ZNN dynamics with a robust activation was presented in [14] and applied in some engineering fields. Xiao in [16] discovered and investigated finite-time dynamical systems for solving time-dependent complex matrix equation $AX = B$. A desirable feature of ZNN dynamics is its ability to neutralize different types of noises that appear during the evolution of dynamic systems. A noise-tolerant ZNN design with nonlinear activation that involves appropriate integral term was proposed and investigated in [17]. Finite-convergent zeroing neural design in the complex domain based on the Tikhonov regularization on the matrix A and aimed at solving time-dependent matrix equations $AX = D$ was considered in [18]. Various and numerous areas of numerical linear algebra are studied by applying ZNN dynamics in the time-varying case. Solving tensor and matrix inversion on TV arrays [19], generalized eigenvalues problems [20], generalized inversion [21], matrix equations and matrix decompositions [3], quadratic optimization problems [22], and approximations of various matrix functions [23,24] are predominant utilizations of ZNNs. ZNN models based on two fuzzy-adaptive activations for resolving specific linear TV linear matrix equations (LME) were introduced in [25,26]. The hybrid models introduced in [27] are defined using suitable combinations of GNN and ZNN neural dynamics for solving matrix equations $BX = D$ or $CX = D$. Apart from different types of activation in dynamic systems, numerous studies have been devoted to the modification of the dynamic evolution of ZNN. High-order error function designs and high-order ZNNs which are designed using powers of standard Zhang functions and adopted for solving LME were proposed in [28].

Determining an error function (or Zhang function) $E(t)$, based on a suitable expression that reflects the problem being solved, is the first step in developing ZNN dynamics [29]. The second stage takes the dynamical development

$$\dot{E}(t) = \frac{dE(t)}{dt} = -\lambda \mathcal{F}(E(t)), \quad \lambda > 0 \tag{2}$$

in which $\dot{E}(t) \in \mathbb{R}^{m \times n}$ means the time derivative of $E(t) \in \mathbb{R}^{m \times n}$, a real number $\lambda > 0$ represents the scaling parameter required for accelerating the convergence speed, and $\mathcal{F}(\cdot) : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ denotes elementwise usage of appropriately defined odd-increasing activation function on $E(t)$. Target of our research is the linear ZNN design

$$\dot{E}(t) = -\lambda E(t), \quad \lambda > 0. \tag{3}$$

The system of first-order differential Equation (3), which defines the ZNN evolution law, has the analytical solution $E(t) = E(0)e^{-\lambda t}$ [12]. Thus, $E(t)$ in the standard ZNN is exponentially convergent to the zero 0 with the exponential convergence rate λ as $t \rightarrow +\infty$.

The ZNN model for solving the GLME (1) is defined using the matrix-valued Zhangian

$$E(t) = A(t)Y(t)C(t) - B(t), \quad A(t) \in \mathbb{R}^{m \times n}, Y(t) \in \mathbb{R}^{n \times k}, C(t) \in \mathbb{R}^{k \times h}, B(t) \in \mathbb{R}^{m \times h}. \quad (4)$$

The main drawback of the basic ZNN design (3) and (4), based on the standard error function $E(t)$, is its limited application to cases where the input matrices meet certain requirements. Motivated by that shortcoming, in this research we propose two new error functions derived from optimization methods. The error function $E_G(t)$ is defined by zeroing the gradient of the Frobenius matrix norm of $E(t)$, while the error function $E_N(t)$ is defined by the Newton step corresponding to the Frobenius matrix norm of $E(t)$. That is why we use the terms Gradient zeroing neural network (GZNN) and Newton zeroing neural network (NZNN) for dynamic systems that are defined based on the new error functions E_G and E_N , respectively.

Moreover, we noticed that the analysis of the complete set of solutions to the Equation (1) has not been investigated so far. In previous research, the solution based on invertible matrices A and C has been analyzed, which corresponds to the solution $Y(t) = A^{-1}(t)B(t)C^{-1}(t)$ [6,26]. Our idea is to look for least-squares solutions, under weaker assumptions that allow the coefficient-matrices to be singular.

The following highlights are key results of this work:

- A new and unique ZNN development, based on two novel error functions $E_G(t)$ and $E_N(t)$, is used to address the problem of solving TV GLMEs (1) with arbitrary TV real input matrices.
- Six ZNN models based on different error functions are presented and compared, five of which are new. Pseudoinversion is involved in three new ZNN models, while two are related to Newton’s optimization method.
- A precise convergence analysis of the proposed ZNNs is presented and the set of solutions is defined using known results from linear algebra.
- Six numerical examples confirm that all considered models can solve TV GLMEs, although their effectiveness varies depending on input matrices.

The following is the structure of the presentation. Section 2 presents the motivation and describes the main results. Section 3 defines six ZNN dynamical systems and analyses their solvability, general solution, sets of all solutions, and convergence properties. Section 4 introduces and analyses simulation results obtained through six numerical examples that involve input matrices of different types. Finally, in Section 5, there are closing remarks.

The paper follows classical notations: $I_n \in \mathbb{R}^{n \times n}$ signifies the identity matrix of the order n ; $I_{m,n} \in \mathbb{R}^{m \times n}$ signifies a $m \times n$ matrix with ones on the main diagonal and zeros elsewhere; $\mathbf{1}_n$ and $\mathbf{1}_{m,n}$, respectively, signify a vector in \mathbb{R}^n and a matrix in $\mathbb{R}^{m \times n}$ consisting of ones; $O_{m,n} \in \mathbb{R}^{m \times n}$ signifies the zero $m \times n$ array; \otimes designates the Kronecker product; $\text{vec}(\cdot)$ stands for the vectorization; \odot means the Hadamard product; $\|\cdot\|_F$ signifies the Frobenius norm; the operators $(\cdot)^T$, $(\cdot)^{-1}$ and $(\cdot)^\dagger$, respectively, signify the matrix transposition, inversion and the pseudoinversion.

2. Motivation and Description of Main Results

The traditional ZNN dynamics for solving (1) is based on the error function (Zhangian) $E(t)$ defined in (4), and follows the dynamic flow

$$\dot{E}(t) = -\lambda E(t) = -\lambda(A(t)Y(t)C(t) - B(t)). \quad (5)$$

The standard ZNN design (5) developed to solve the GLME (1) requires invertible matrices A and C to force $E(t)$ to zero, which the unique solution $Y(t) = A^{-1}(t)B(t)C^{-1}(t)$ [26,30]. Our goal is to resolve this restriction and propose dynamical evolutions based on error functions that tend to zero in more general cases, which include the singularity of matrix-coefficients $A(t)$ and $C(t)$. This research aims to find new neurodynamical models based on different error

functions. Our motivation in defining new Zhang functions has its origin in gradient-descent methods for solving the unconstrained minimization problem

$$\min f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n, \tag{6}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously-differentiable function bounded from below. The *gradient descent* (GD) iterative scheme for solving (6) is defined by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k, \tag{7}$$

whereby $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ is the gradient vector and α_k is appropriately defined step size. The convergence result

$$\lim_{k \rightarrow \infty} \|\mathbf{g}_k\| = 0 \tag{8}$$

for iterations (7) is widely known in the literature [31].

The Newton method with line search for solving (6) is defined by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k G_k^{-1} \mathbf{g}_k, \tag{9}$$

where G_k^{-1} denotes the inverse of the Hessian matrix $G_k = \nabla^2 f(\mathbf{x}_k)$.

New error functions $E_G(t)$ and $E_N(t)$ are defined using analogies with gradient-descent and Newton method for nonlinear optimization. The residual matrix $E(t) = A(t)Y(t)C(t) - B(t)$ is the desired goal function which is forced to the zero matrix. The gradient of the Frobenius norm

$$\varepsilon_Y = \frac{\|E(t)\|_F^2}{2} = \frac{\|A(t)Y(t)C(t) - B(t)\|_F^2}{2}$$

is equal to [6]

$$\frac{\partial \varepsilon_Y}{\partial Y} = \nabla \varepsilon_Y = A^T(AYC - B)C^T = E_G(t).$$

Following iterations defined in [32], the gradient-descent iterations for minimizing the goal function ε_Y are defined by

$$Y_{k+1} = Y_k - \alpha_k A^T(AY_k C - B)C^T, \alpha_k > 0,$$

taking an arbitrary initial matrix $Y_0 \in \mathbb{R}^{n \times k}$. The Gradient neural network (GNN) dynamic evolution minimizes $\|AY(t)C - B\|_F^2$ and it is based on the direct proportionality between the time derivative $\dot{Y}(t)$ and the negative gradient of the goal function $\varepsilon_Y(t)$ [33,34]:

$$\dot{Y}(t) = -\lambda \frac{\partial \varepsilon_Y(t)}{\partial Y}, Y(0) = Y_0. \tag{10}$$

Our goal is to define the ZNN design for solving the TV GLME (1) based on the error function

$$E_G(t) := \nabla \varepsilon_Y(t) = A^T(AYC - B)C^T = A^T E(t) C^T, \tag{11}$$

which initiates the following dynamics, termed GZNN dynamics:

$$\dot{E}_G(t) = -\lambda E_G(t) = -\lambda (A^T(AYC - B)C^T). \tag{12}$$

Further, Hessian of ε_Y in the case $C := I$ is equal to $\nabla^2 \varepsilon_Y = \frac{\partial^2 \varepsilon_Y}{\partial Y^2} = A^T A$. In the case $A := I$, Hessian of ε_Y is equal to $\nabla^2 \varepsilon_Y = \frac{\partial^2 \varepsilon_Y}{\partial Y^2} = CC^T$. Following the results from [32], the Newton iterations with line search in the time-invariant case can be defined by

$$Y_{k+1} = Y_k - \alpha_k (A^T A)^{-1} A^T(AY_k C - B)C^T (CC^T)^{-1}, \alpha_k > 0.$$

Consider the real numbers $\delta_{A^T A}$ and δ_{CC^T} defined by

$$\delta_{A^T A} = \begin{cases} \varepsilon, & \text{rank}(A^T A) < n, \\ 0, & \text{rank}(A^T A) = n, \end{cases} \quad \delta_{CC^T} = \begin{cases} \varepsilon, & \text{rank}(CC^T) < k, \\ 0, & \text{rank}(CC^T) = k, \end{cases}$$

where $\varepsilon > 0$ is a small real regularization parameter. So, we define the Zhangian in the case of $\text{rank}(A) \leq n \leq m, \text{rank}(C) \leq k \leq h$ as

$$E_N(t) := \left(A^T(t)A(t) + \delta_{A^T A} I \right)^{-1} A^T(A(t)Y(t)C(t) - B(t))C^T(t) \left(C(t)C^T(t) + \delta_{CC^T} I \right)^{-1}. \tag{13}$$

Given the well-known limit representation of the Moore-Penrose inverse, restated in [35], it can be obtained

$$\lim_{\varepsilon \rightarrow 0} \left(A^T A + \delta_{A^T A} I \right)^{-1} A^T = A^\dagger, \quad \lim_{\varepsilon \rightarrow 0} C^T \left(CC^T + \delta_{CC^T} I \right)^{-1} = C^\dagger,$$

which implies

$$E_N(t) \rightarrow A^\dagger(t)(A(t)Y(t)C(t) - B(t))C^\dagger(t) = A^\dagger(t)E(t)C^\dagger(t)$$

as $\varepsilon \rightarrow 0$. The ZNN design for solving $E_N(t) = 0$ based on the error function $E_N(t)$, and termed NZNN, is defined by

$$\dot{E}_N(t) = -\lambda E_N(t). \tag{14}$$

Dynamics (14) forces $E_N(t)$ to zero, which (in view of (13)) coincides with the zero of $E_G(t)$.

Solvability and Solutions of Proposed Dynamical Systems

In this subsection we investigate conditions for solvability of the matrix equations $E(t) = 0, E_G(t) = 0$ and $E_N(t) = 0$ and their sets of solutions.

Lemma 1 defines conditions for the solvability and defines solutions of the linear matrix equation $AYC = B$. The results is obtained as a consequence of [35] (p. 52, Theorem 1) in the time-varying case.

Lemma 1 ([35] p. 52, Theorem 1). *For arbitrary $A(t) \in \mathbb{C}^{m \times n}, B(t) \in \mathbb{C}^{p \times q}, D(t) \in \mathbb{C}^{m \times q}$, the general linear matrix equation*

$$A(t)Y(t)C(t) = B(t) \tag{15}$$

is solvable if and only if

$$A(t)A^\dagger(t)B(t)C^\dagger(t)C(t) = B(t), \tag{16}$$

in which case its general solution is given by

$$A^\dagger(t)B(t)C^\dagger(t) + Q(t) - A^\dagger(t)A(t)Q(t)C(t)C^\dagger(t), \tag{17}$$

for arbitrary $Q(t) \in \mathbb{C}^{n \times p}$.

Corollary 1. *If (16) is satisfied, the set of all solutions to (15) is defined by*

$$\Theta_Q = \left\{ A^\dagger(t)B(t)C^\dagger(t) + Q(t) - A^\dagger(t)A(t)Q(t)C(t)C^\dagger(t) \mid Q(t) \in \mathbb{C}^{n \times p} \right\}. \tag{18}$$

The set Θ_Q is termed as solutions set in rest of the paper.

Lemma 2 restates results from [8] (Theorem 2.2). These results are used in Remark 1 to define the solution which will be used in numerical comparison of defined models.

Lemma 2 ([8] Theorem 2.2). Assume that constant matrices $A \in \mathbb{R}^{m \times n}$, $C \in \mathbb{R}^{k \times h}$ and $B \in \mathbb{R}^{m \times h}$ satisfy

$$AA^\dagger BC^\dagger C = B. \tag{19}$$

Then the activation state variables matrix $Y(t)$ of the dynamical system

$$\dot{Y}(t) = -\gamma A^\top (AV(t)C - B)C^\top \tag{20}$$

is convergent as $t \rightarrow +\infty$ and has the equilibrium state

$$Y(t) \rightarrow \tilde{Y} = A^\dagger CB^\dagger + Y(0) - A^\dagger AY(0)CC^\dagger \tag{21}$$

for every initial state matrix $Y(0) \in \mathbb{R}^{n \times p}$.

Remark 1. Theorem 2.2 in [8] claims that the activation state variables matrix $Y(t)$ of the GNN dynamics (20) for solving (15) in the time-invariant case converges as $t \rightarrow +\infty$ to the equilibrium state defined in (21).

Accordingly, we will use the equilibrium matrix

$$\tilde{Y}_{Y(0)}(t) = A^\dagger(t)B(t)C^\dagger(t) + Y(0) - A^\dagger(t)A(t)Y(0)C(t)C^\dagger(t) \in \Theta_Q, \tag{22}$$

for the comparison of the proposed models in performed numerical experiments.

Theorem 1 investigates conditions for the solvability of the equations $E(t) = 0$, $E_G(t) = 0$, $E_N(t) = 0$, and solutions sets to these equations.

Theorem 1. Consider arbitrary smooth TV matrices $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$. The following statements are valid.

- (a) The equation $E(t) = 0$ is solvable if and only if (16) holds, and the solutions set to $E(t) = 0$ is Θ_Q defined in (18).
- (b) If $\text{rank}(A) \leq n \leq m$, the equation $E_G(t) = 0$ is always solvable and the solutions set to $E_G(t) = 0$ is Θ_Q defined by (18).
- (c) If $\text{rank}(A) \leq n \leq m$, the equation $E_N(t) = 0$ is always solvable and the solutions set to $E_N(t) = 0$ is Θ_Q defined by (18).

Proof. (a) The error function $E(t)$ in the traditional ZNN (TZNN) model is defined by (4) and solves the equation $E(t) = 0$. This part of the proof follows from known results on the solvability and the general solution to the matrix Equation (15), given in Lemma 1, and its application to the matrix equation $E(t) = 0 \iff A(t)Y(t)C(t) = B(t)$.

(b) The dynamics (12) forces $E_G(t) = \nabla_{\varepsilon_Y} E(t) = A^\top (AYC - B)C^\top$ to zero. On the other hand, according to Lemma 1, the error function $E_G(t)$ vanishes in the case.

$$E_G(t) = 0 \iff A^\top AYCC^\top = A^\top BC^\top.$$

As a consequence, the matrix equation $E_G(t) = 0$ is consistent, its general solution is given by (17), and its solutions set is defined in (18). The consistency is confirmed by the identity

$$A^\top A (A^\top A)^\dagger A^\top BC^\top (CC^\top)^\dagger CC^\top = A^\top BC^\top.$$

In addition, the general solution to $E_G(t) = 0$ is given by (17). Indeed, using Lemma 1, the general solution to $E_G(t) = 0$ is equal to

$$\begin{aligned} Y &= (A^\top A)^\dagger A^\top BC^\top (CC^\top)^\dagger + Q - (A^\top A)^\dagger A^\top AQCC^\top (CC^\top)^\dagger \\ &= A^\dagger BC^\dagger + Q + A^\dagger AQCC^\dagger. \end{aligned} \tag{23}$$

- (c) In the case $\text{rank}(A) \leq n \leq m, \text{rank}(C) \leq k \leq h$, the Zhangian $E_N(t)$ becomes the zero matrix in the case

$$E_N(t) = 0 \iff \left(A^T A + \delta_{A^T A} I \right)^{-1} A^T (AYC - B) C^T \left(CC^T + \delta_{CC^T} I \right)^{-1} = 0. \quad (24)$$

The GLME $E_N(t) = 0$ is always solvable and its solutions set is equal to (18). This statement follows from

$$\begin{aligned} E_N(t) = 0 &\iff A^T (AYC - B) C^T = 0 \\ &\iff E_G(t) = 0. \end{aligned}$$

Following general results from [35] (p. 52, Theorem 1) it is straightforward to verify that (18) is the general set of solutions to $E_N(t) = 0$.

□

Remark 2. The general conclusion is that the neural dynamics (12) forces the residual matrix $E_G(t) = \nabla_{\varepsilon_Y(t)} = A^T (AYC - B) C^T$ to zero. According to (23), the ZNN dynamics (12) reaches the solution $Y(t)$ of the form (23) which coincides with the solution (18) to $E(t) = A(t)Y(t)C(t) - B(t) \rightarrow 0$. Finally, $E_N(t) \rightarrow 0$ is equivalent to $E_G(t) \rightarrow 0$.

In this way, we explain one of the main advantages of the proposed models: the matrix equation $E(t) = 0$ is solvable under the condition (16), while zeroing $E_G(t) = 0$ and $E_N(t) = 0$ are always consistent. In view of (18) and (23), the general solutions to all three equations $E(t) = 0, E_G(t) = 0$ and $E_N(t) = 0$ are identical.

Corollary 2. Let us consider the solutions set

$$\Lambda = \left\{ \tilde{Y}_{Y(0)}(t) \mid Y(0) \in \mathbb{R}^{n \times k} \right\},$$

generated by all possible initial states $Y(0)$ of the form (22). The following statements are valid.

- (a) $A\Lambda C = \{B\}$.
- (b) An arbitrary solution $\tilde{Y}_{Y(0)}(t) \in \Lambda$, defined in (22), is a least squares solution to (1).
- (c) The unique solution $\tilde{Y}_0(t) \in \Lambda$ produces the best-approximate (i.e., the minimal-norm least-squares) solution $A^\dagger B C^\dagger$ to (1).

Proof. Part (a) follows from Lemma 1.

(b) Following the results from [36,37],

$$\|AYC - B\|_2 \geq \|A A^\dagger B C^\dagger C - B\|_2,$$

for arbitrary solution Y to (1), where the equality is valid if $Y \in \tilde{Y}_{Y(0)}(t)$.

(c) Follows from the known result originated in [36,37]

$$\|A^\dagger B C^\dagger\|_2 \leq \|A^\dagger B C^\dagger + Y - A^\dagger A Y C^\dagger C\|_2.$$

The proof is complete. □

3. Various ZNN Models for Solving TV GLMEs

The current section defines and analyses six ZNN expansions for calculating online solutions of TV GLMEs, including TZNN which is based on the traditional ZNN approach, GZNN based on the gradient ZNN approach, NZNNV1, NZNNV2 and NZNNV3 which are based on the Newton’s optimization method, and DZNN based on the ZNN approach for finding the direct solution produced with the use of pseudoinverse. The TZNN and GZNN models provided here cover all potential scenarios for arbitrary TV real input matrices $A(t), B(t), C(t)$ for solving the TV GLME (1).

3.1. Traditional ZNN Model

Considering smooth arbitrary TV matrices $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$, the error function in the traditional ZNN (TZNN) model is established as

$$E(t) = A(t)Y(t)C(t) - B(t), \tag{25}$$

under conditions

$$\{A, B, C\} \in \mathcal{D} = \left\{ A \in \mathbb{R}^{m \times n}, C \in \mathbb{R}^{k \times h}, B \in \mathbb{R}^{m \times h}, m = n = \text{rank}(A), k = h = \text{rank}(C) \right\}. \tag{26}$$

It is important to note that the desired solution is $Y(t) \in \mathbb{R}^{n \times k}$. The time derivative of $E(t)$ defined in (25) is equal to

$$\dot{E}(t) = \dot{A}(t)Y(t)C(t) + A(t)\dot{Y}(t)C(t) + A(t)Y(t)\dot{C}(t) - \dot{B}(t). \tag{27}$$

Therefore, the following implicit dynamics can be obtained considering the linear ZNN flow (2)

$$\begin{aligned} A(t)\dot{Y}(t)C(t) &= -\lambda(A(t)Y(t)C(t) - B(t)) - \dot{A}(t)Y(t)C(t) - A(t)Y(t)\dot{C}(t) + \dot{B}(t), \\ \{A, B, C\} &\in \mathcal{D}. \end{aligned} \tag{28}$$

which is just the linear ZNN model (8) in [6]. Using the Kronecker product in conjunction with the vectorization, the dynamics (28) is modified and, as a result, we obtain the following ZNN model in vector form

$$\begin{aligned} (C^T(t) \otimes A(t))\text{vec}(\dot{Y}(t)) &= \\ \text{vec}(-\lambda(A(t)Y(t)C(t) - B(t)) - \dot{A}(t)Y(t)C(t) - A(t)Y(t)\dot{C}(t) + \dot{B}(t)), & \\ \{A, B, C\} \in \mathcal{D}. & \end{aligned} \tag{29}$$

The ZNN model (29) will be termed as TZNN model. The solution to the dynamical system (29) can be approximated using the standard Matlab ode solver. If $C^T(t) \otimes A(t)$ is a nonsingular mass matrix, the exponential convergence speed achieved by the TZNN design (29) to the exact TV solution of the TV GLME (1) is proved in [6] (Theorem 1).

In [6], the authors vectorized the TV GLME $A(t)Y(t)C(t) = B(t)$ into its equivalent form $(C^T(t) \otimes A(t))\text{vec}(Y(t)) = \text{vec}(B(t))$. We restate the main result from [6] (Theorem 1) in the form adopted for (1) to improve readability of the paper. This result defines Unique-solution condition that requires invertibility of the mass matrix $(C^T(t) \otimes A(t))$.

Lemma 3 ([6] Theorem 1). *Assume that time-varying coefficient matrices in (1) are smooth. If the following Unique-solution condition*

$$(C^T(t) \otimes A(t))^T(C^T(t) \otimes A(t)) \geq \delta I, \forall t \geq 0, \tag{30}$$

is satisfied for a real number $\delta > 0$, then the matrix $Y(t)$ of the Zhang neural network (28), starting from arbitrary initial state $Y(0)$, converges exponentially to the theoretical solution of (1) with the exponential convergence rate λ .

Vectorized ZNN design (29) exploits the same mass matrix $C^T(t) \otimes A(t)$ and requires its invertibility. Using $(C^T(t) \otimes A(t))^{-1} = C^T(t)^{-1}A(t)^{-1}$, we conclude that the Unique-solution condition requires invertibility of $A(t)$ and $C(t)$. Therefore, if $C^T(t) \otimes A(t)$ is nonsingular then (16) is satisfied.

Corollary 3. *Let $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ be differentiable. If the condition (26) is satisfied, starting from arbitrary initial value $Y(0)$, the TZNN model (40) converges exponentially and globally to $A^{-1}(t)B(t)C^{-1}(t)$.*

3.2. Gradient ZNN (GZNN) Model

The gradient-based ZNN (GZNN) approach considers arbitrary smooth TV matrices $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ for solving

$$A^T(t)A(t)Y(t)C(t)C^T(t) = A^T(t)B(t)C^T(t), \tag{31}$$

where $Y(t) \in \mathbb{R}^{n \times k}$ is the desired solution. According to general requirements of the ZNN design (2) [6,30], we consider the constraints

$$\{A, B, C\} \in \mathcal{D}_G = \left\{ A \in \mathbb{R}^{m \times n}, C \in \mathbb{R}^{k \times h}, B \in \mathbb{R}^{m \times h}, m \geq n \geq \text{rank}(A(t)), h \geq k \geq \text{rank}(C(t)) \right\}. \tag{32}$$

The error function in GZNN design will be defined by

$$E_G(t) = A^T(t)(A(t)Y(t)C(t) - B(t))C^T(t) = A^T(t)E(t)C^T(t), \quad \{A, B, C\} \in \mathcal{D}_G. \tag{33}$$

The time derivative of (33) is

$$\dot{E}_G(t) = \dot{A}^T(AYC - B)C^T + A^T(\dot{A}YC + A\dot{Y}C + AY\dot{C} - \dot{B})C^T + A^T(AYC - B)\dot{C}^T. \tag{34}$$

Then, considering (2), it can be obtained:

$$A^T A \dot{Y} C C^T = -\lambda A^T(AYC - B) - \dot{A}^T(AYC - B)C^T - A^T(\dot{A}YC + AY\dot{C} - \dot{B})C^T - A^T(AYC - B)\dot{C}^T, \quad \{A, B, C\} \in \mathcal{D}_G. \tag{35}$$

Using the vectorization in conjunction with the Kronecker product, the dynamics (35) is modified as

$$\begin{aligned} & (CC^T \otimes A^T A) \text{vec}(\dot{Y}) = \\ & \text{vec}(-\lambda A^T(AYC - B) - \dot{A}^T(AYC - B)C^T - A^T(\dot{A}YC + AY\dot{C} - \dot{B})C^T - A^T(AYC - B)\dot{C}^T), \\ & \{A, B, C\} \in \mathcal{D}_G. \end{aligned} \tag{36}$$

Replacements

$$\begin{aligned} W(t) &= CC^T \otimes A^T A \\ r(t) &= \text{vec}(-\lambda A^T(AYC - B) - \dot{A}^T(AYC - B)C^T \\ &\quad - A^T(\dot{A}YC + AY\dot{C} - \dot{B})C^T - A^T(AYC - B)\dot{C}^T) \\ \dot{y}(t) &= \text{vec}(\dot{Y}(t)), \quad y = \text{vec}(Y(t)) \end{aligned} \tag{37}$$

lead to the dynamical evolution

$$W(t)\dot{y}(t) = r(t). \tag{38}$$

The mass matrix $W(t)$ in (38) is singular in rank conditions $\text{rank}(A) < \min\{m, n\}$ or $\text{rank}(C) < \min\{k, h\}$. The Tikhonov regularization is the principle widely exploited to solve such singularity problems. As a result, $W(t)$ is regularized by the matrix

$$M(t) = \begin{cases} W(t), & \text{rank}(A) = \min\{m, n\} \ \& \ \text{rank}(C) = \min\{k, h\} \\ W(t) + \beta I_{kn}, & \text{rank}(A) < \min\{m, n\} \ \text{or} \ \text{rank}(C) < \min\{k, h\} \end{cases} \tag{39}$$

where $\beta \geq 0$ is a small regularization quantity. Then the following ZNN model is used instead of (38):

$$M(t)\dot{y}(t) = r(t), \tag{40}$$

where $M(t)$ is a nonsingular mass matrix. The ZNN (40) is named GZNN, and it is solvable using a proper ode Matlab solver. A global convergence with exponential speed of the GZNN design (40) to the exact TV solution of the TV GLME (31) is certified in Theorem 2.

Theorem 2. Let $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ be differentiable. If the condition (32) is satisfied, starting from arbitrary initial value $\mathbf{y}(0)$, the vectorized GZNN model (40) converges exponentially and globally to one element from the set Θ_Q defined in (18), which solves the TV GLME (31).

Proof. To obtain the solution $\mathbf{y}(t)$ to the TV GLMEs (31), the Zhangian matrix is defined by (33) inline with its time derivative (34). The ZNN expansion based on (33) leads to the model (35). Furthermore, from the derivation procedure, (40) is an equivalent form of (35), which represents the standard GZNN model. Since a regularization parameter β is involved in the definition of the mass matrix $M(t)$ in (40), $M(t)$ is invertible. From [6] (Theorem 1), restated in Lemma 3, the unknown matrix $Y(t)$ in (35) converges to the exact solution as $t \rightarrow \infty$. According to Theorem 1(b), the matrix equation $E_G(t) = 0$ is always solvable and its solutions are included in Θ_Q . \square

3.3. Newton ZNN Model (Version 1)

Applying the limit representation of the pseudoinverse [35]

$$\lim_{\epsilon \rightarrow 0} \left(A^T A + \delta_{A^T A} I \right)^{-1} A^T = A^\dagger, \quad \lim_{\epsilon \rightarrow 0} C^T \left(C C^T + \delta_{C C^T} I \right)^{-1} = C^\dagger,$$

the error function (13) of the Newton ZNN version 1 (NZNNV1) model will be defined by

$$E_N(t) = A^\dagger(t)(A(t)Y(t)C(t) - B(t))C^\dagger(t), \quad \{A, B, C\} \in \mathcal{D}_G. \tag{41}$$

So, the first variant of the Newton ZNN approach considers the smooth arbitrary TV matrices $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ for solving the following GLMEs:

$$A^\dagger(t)A(t)Y(t)C(t)C^\dagger(t) = A^\dagger(t)B(t)C^\dagger(t), \quad \{A, B, C\} \in \mathcal{D}_G, \tag{42}$$

where the desired solution is $Y(t) \in \mathbb{R}^{n \times k}$.

The time derivative of $E_N(t)$ is given as

$$\dot{E}_N(t) = \dot{A}^\dagger(AYC - B)C^\dagger + A^\dagger(\dot{A}YC + A\dot{Y}C + AY\dot{C} - \dot{B})C^\dagger + A^\dagger(AYC - B)\dot{C}^\dagger. \tag{43}$$

The time derivative of the pseudoinverse of $P(t) \in \mathbb{R}^{m \times n}$ is equal to

$$\dot{P}^\dagger = -P^\dagger \dot{P} P^\dagger + P^\dagger (P^\dagger)^T \dot{P}^T (I - P P^\dagger) + (I - P^\dagger P) \dot{P}^T (P^\dagger)^T P^\dagger. \tag{44}$$

Then, considering the linear ZNN design (2), it is obtained

$$\begin{aligned} & \dot{A}^\dagger(AYC - B)C^\dagger + A^\dagger(\dot{A}YC + A\dot{Y}C + AY\dot{C} - \dot{B})C^\dagger + A^\dagger(AYC - B)\dot{C}^\dagger \\ & = -\lambda A^\dagger(AYC - B)C^\dagger, \quad \{A, B, C\} \in \mathcal{D}_G, \end{aligned} \tag{45}$$

or equivalently

$$\begin{aligned} A^\dagger A \dot{Y} C C^\dagger & = -\lambda A^\dagger(AYC - B)C^\dagger - \dot{A}^\dagger(AYC - B)C^\dagger - A^\dagger(\dot{A}YC + AY\dot{C} - \dot{B})C^\dagger \\ & \quad - A^\dagger(AYC - B)\dot{C}^\dagger, \quad \{A, B, C\} \in \mathcal{D}_G, \end{aligned} \tag{46}$$

Applying the vectorization and the Kronecker product, the (46) dynamics is transformed into

$$\begin{aligned} & \left((CC^\dagger) \otimes (A^\dagger A) \right) \text{vec}(\dot{Y}) \\ &= \text{vec} \left(-\lambda A^\dagger (AYC - B)C^\dagger - \dot{A}^\dagger (AYC - B)C^\dagger - A^\dagger (\dot{A}YC + AY\dot{C} - \dot{B})C^\dagger \right. \\ & \quad \left. - A^\dagger (AYC - B)\dot{C}^\dagger \right), \\ & \{A, B, C\} \in \mathcal{D}_G, \end{aligned} \tag{47}$$

and setting

$$\begin{aligned} W(t) &= ((CC^\dagger) \otimes (A^\dagger A)) \\ r(t) &= \text{vec} \left(-\lambda A^\dagger (AYC - B)C^\dagger - \dot{A}^\dagger (AYC - B)C^\dagger \right. \\ & \quad \left. - A^\dagger (\dot{A}YC + AY\dot{C} - \dot{B})C^\dagger - A^\dagger (AYC - B)\dot{C}^\dagger \right) \\ \dot{y}(t) &= \text{vec}(\dot{Y}(t)), \quad y(t) = \text{vec}(Y(t)), \end{aligned} \tag{48}$$

the subsequent ZNN design is generated:

$$W(t)\dot{y}(t) = r(t). \tag{49}$$

The mass matrix $W(t)$ in (49) is singular under the rank conditions $\text{rank}(A) < \min\{m, n\}$ or $\text{rank}(C) < \min\{k, h\}$. The regularization principle leads to the regular mass matrix

$$M(t) = \begin{cases} W(t), & \text{rank}(A) = \min\{m, n\} \ \& \ \text{rank}(C) = \min\{k, h\} \\ W(t) + \beta I_{kn}, & \text{rank}(A) < \min\{m, n\} \ \text{or} \ \text{rank}(C) < \min\{k, h\} \end{cases} \tag{50}$$

such that $\beta \geq 0$ defines the ridge regression parameter. Accordingly, the ZNN dynamical system

$$M(t)\dot{y}(t) = r(t) \tag{51}$$

is used instead of (49), where $M(t)$ is the nonsingular mass matrix. The differential system (51) is termed NZNNV1, and it is solvable using ode Matlab solvers. Theorem 3 proves the exponential convergence rate of the NZNNV1 evolution (51) to the accurate TV solution of TV GLMEs (42).

Theorem 3. *Let $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ be differentiable. If the condition (16) is satisfied, the NZNNV1 model (51), starting from any initial value $y(0)$, exponentially converges to one element from the set Θ_Q defined in (18), which solves the TV GLME (42).*

Proof. To acquire the solution $y(t)$ to (42), the Zhangian matrix is defined as in (41), in conjunction with the initiated ZNN dynamics. The model (45) is obtained by adopting the linear design for zeroing (41), which produces the NZNN design. From [6] (Theorem 1), restated in Lemma 3, each error matrix equation in the error matrix equation group (45) converges to the accurate solution as $t \rightarrow \infty$. According to Theorem 1(c), the matrix equation $E_N(t) = 0$ is always solvable and its solutions are included in Θ_Q . Consequently, the solution of (45) converges to the solution of TV GLMEs (42) as $t \rightarrow \infty$. The derivation procedure of (51) confirms that it is an equivalent form of (45). \square

3.4. Newton ZNN Model (Version 2)

This subsection presents the second version of the Newton ZNN model for solving the TV GLMEs of (42). The Newton ZNN version 2 (NZNNV2) model includes two error functions. More precisely, the error function (41) is converted as follows:

$$E_1(t) = X(t)(A(t)Y(t)C(t) - B(t))Z(t), \quad \{A, B, C\} \in \mathcal{D}_G. \tag{52}$$

The matrix $X(t) \in \mathbb{R}^{n \times m}$ in (52) is defined as the zero of the error function for finding the pseudoinverse of an arbitrary TV matrix $A(t)$ (see [38])

$$E_2(t) = A^T(t)(I_m - A(t)X(t)), \quad m \geq n \geq \text{rank}(A(t)), \quad (53)$$

while $Z(t) \in \mathbb{R}^{h \times k}$ is the zero of the error function for finding the pseudoinverse of an arbitrary TV matrix $C(t)$

$$E_3(t) = (I_h - Z(t)C(t))C^T(t), \quad h \geq k \geq \text{rank}(C(t)). \quad (54)$$

The derivative of (52) is

$$\begin{aligned} \dot{E}_1(t) &= \dot{X}(AY - B)Z + X(\dot{A}YC + A\dot{Y}C + AY\dot{C} - \dot{B})Z + X(AYC - B)\dot{Z}, \\ &\{A, B, C\} \in \mathcal{D}_G, \end{aligned} \quad (55)$$

while the derivative of (53) is

$$\dot{E}_2(t) = \dot{A}^T(I_m - AX) - A^T\dot{A}X - A^T A\dot{X}, \quad m \geq n \geq \text{rank}(A), \quad (56)$$

and

$$\dot{E}_3(t) = (I_h - ZC)\dot{C}^T - \dot{Z}CC^T - Z\dot{C}C^T, \quad h \geq k \geq \text{rank}(C), \quad (57)$$

Then, considering the linear ZNN design (2), it can be obtained:

$$\begin{cases} \dot{X}(AY - B)Z + X(\dot{A}YC + A\dot{Y}C + AY\dot{C} - \dot{B})Z + X(AYC - B)\dot{Z} = -\lambda X(AYC - B)Z, \\ \dot{A}^T(I_m - AX) - A^T\dot{A}X - A^T A\dot{X} = -\lambda A^T(I_m - AX), \\ (I_h - ZC)\dot{C}^T - \dot{Z}CC^T - Z\dot{C}C^T = -\lambda(I_h - ZC)C^T \end{cases} \quad (58)$$

or in equivalent form

$$\begin{cases} XA\dot{Y}CZ + \dot{X}(AY - B)Z + X(AYC - B)\dot{Z} = -\lambda X(AYC - B)Z - X(\dot{A}YC + AY\dot{C} - \dot{B})Z, \\ -A^T A\dot{X} = -\lambda A^T(I_m - AX) - \dot{A}^T(I_m - AX) + A^T\dot{A}X, \\ -\dot{Z}CC^T = -\lambda(I_h - ZC)C^T - (I_h - ZC)\dot{C}^T + Z\dot{C}C^T. \end{cases} \quad (59)$$

The dynamics (59) are modified using vectorization and the Kronecker product as follows:

$$\begin{cases} (CZ)^T \otimes (XA)\text{vec}(\dot{Y}) + ((AY - B)Z)^T \otimes I_n \text{vec}(\dot{X}) + I_k \otimes (X(AYC - B))\text{vec}(\dot{Z}) \\ = \text{vec}(-\lambda X(AYC - B)Z - X(\dot{A}YC + AY\dot{C} - \dot{B})Z), \\ (-I_m \otimes A^T A)\text{vec}(\dot{X}) = \text{vec}(-\lambda A^T(I_m - AX) - \dot{A}^T(I_m - AX) + A^T\dot{A}X), \\ (-CC^T \otimes I_h)\text{vec}(\dot{Z}) = \text{vec}(-\lambda(I_h - ZC)C^T - (I_h - ZC)\dot{C}^T + Z\dot{C}C^T). \end{cases} \quad (60)$$

Setting

$$\begin{aligned}
 w_1(t) &= (CZ)^T \otimes (XA) \\
 w_2(t) &= ((AY - B)Z)^T \otimes I_n, \\
 w_3(t) &= I_k \otimes (X(AYC - B)), \\
 w_4(t) &= -I_m \otimes A^T A, \\
 w_5(t) &= -CC^T \otimes I_h, \\
 r_1(t) &= \text{vec}(-\lambda X(AYC - B)Z - X(\dot{A}YC + AY\dot{C} - \dot{B})Z) \\
 r_2(t) &= \text{vec}(-\lambda A^T(I_m - AX) - \dot{A}^T(I_m - AX) + A^T\dot{A}X) \\
 r_3(t) &= \text{vec}(-\lambda(I_h - ZC)C^T - (I_h - ZC)\dot{C}^T + Z\dot{C}C^T) \\
 W(t) &= \begin{bmatrix} w_1 & w_2 & w_3 \\ O_{mn, kn} & w_4 & O_{mn, kh} \\ O_{kh, kn} & O_{kh, mn} & w_5 \end{bmatrix}, \quad r(t) = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \quad \dot{\mathbf{y}}(t) = \begin{bmatrix} \text{vec}(\dot{Y}) \\ \text{vec}(\dot{X}) \\ \text{vec}(\dot{Z}) \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \text{vec}(Y) \\ \text{vec}(X) \\ \text{vec}(Z) \end{bmatrix},
 \end{aligned} \tag{61}$$

the following dynamical system is generated

$$W(t)\dot{\mathbf{y}}(t) = r(t), \tag{62}$$

with the mass matrix $W(t)$. To solve the problem with singularity of $W(t)$ in the case $\text{rank}(A) < \min\{m, n\}$ or $\text{rank}(C(t)) < \min\{k, h\}$, the following nonsingular mass matrix is used:

$$M(t) = \begin{cases} W(t), & \text{rank}(A) = \min\{m, n\} \ \& \ \text{rank}(C) = \min\{k, h\} \\ W(t) + \beta I_{mkn}, & \text{rank}(A) < \min\{m, n\} \ \text{or} \ \text{rank}(C) < \min\{k, h\}, \end{cases} \tag{63}$$

wherein $\beta \geq 0$. Then the following ZNN model is used instead of (62):

$$M(t)\dot{\mathbf{y}}(t) = r(t), \tag{64}$$

where $M(t)$ is a nonsingular mass matrix. The NZNNV2 design (64) is solvable with an ode solver available in Matlab. The global and exponential convergence rate of the NZNNV2 flow (64) to the exact TV solution of (42) is verified in Theorem 4.

Theorem 4. Let $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ be differentiable. If the condition (16) is satisfied, the NZNNV2 model (64) initialized by an arbitrary initial value $\mathbf{y}(0)$, exponentially converges to the exact solution of the TV GLMEs (42) in the form (18).

Proof. Similar to the proof of Theorem 3. \square

3.5. Newton ZNN Model (Version 3)

This subsection presents the third version of the Newton ZNN model for solving the TV GLMEs of (65). The Newton approach considers the smooth arbitrary TV matrices $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ for solving the following GLMEs:

$$\begin{aligned}
 (A^T A + \gamma I_n)^{-1} A^T A Y C C^T (C C^T + \gamma I_k)^{-1} &= (A^T A + \gamma I_n)^{-1} A^T B C^T (C C^T + \gamma I_k)^{-1}, \\
 \{A, B, C\} &\in \mathcal{D}_G,
 \end{aligned} \tag{65}$$

where $Y \in \mathbb{R}^{n \times k}$ is the desired solution. Note that $\gamma \geq 0$ denotes the regularization parameter.

Based on the aforementioned, the error function in the Newton ZNN version 3 (NZNNV3) model will be defined by:

$$E_N(t) = (A^T A + \gamma I_n)^{-1} A^T (AYC - B) C^T (C C^T + \gamma I_k)^{-1}, \quad \{A, B, C\} \in \mathcal{D}_G, \tag{66}$$

Given that the time derivative of the inverse of $P(t)$ is

$$\dot{P}^{-1}(t) = -P^{-1}\dot{P}P^{-1}, \tag{67}$$

and setting

$$S_1(t) = (A^T A + \gamma I_n)^{-1}, \quad m \geq n \geq \text{rank}(A), \tag{68}$$

$$S_2(t) = (CC^T + \gamma I_k)^{-1}, \quad h \geq k \geq \text{rank}(C), \tag{69}$$

with

$$\dot{S}_1(t) = -\left(A^T A + \gamma I_n\right)^{-1} \left(\dot{A}^T A + A^T \dot{A}\right) \left(A^T A + \gamma I_n\right)^{-1}, \quad m \geq n \geq \text{rank}(A) \tag{70}$$

and

$$\dot{S}_2(t) = -\left(CC^T + \gamma I_k\right)^{-1} \left(\dot{C}C^T + CC^T\dot{C}\right) \left(CC^T + \gamma I_k\right)^{-1}, \quad h \geq k \geq \text{rank}(C), \tag{71}$$

the time derivative of (41) is given by

$$\begin{aligned} \dot{E}_N(t) = & \dot{S}_1 A^T (AYC - B)C^T S_2 + S_1 \dot{A}^T (AYC - B)C^T S_2 \\ & + S_1 A^T (\dot{A}YC + A\dot{Y}C + AY\dot{C} - \dot{B})C^T S_2 \\ & + S_1 A^T (AYC - B)\dot{C}^T S_2 \\ & + S_1 A^T (AYC - B)C^T \dot{S}_2, \quad \{A, B, C\} \in \mathcal{D}_G. \end{aligned} \tag{72}$$

Then, considering the linear ZNN dynamics (2), it can be obtained:

$$\begin{aligned} & \dot{S}_1 A^T (AYC - B)C^T S_2 + S_1 \dot{A}^T (AYC - B)C^T S_2 \\ & + S_1 A^T (\dot{A}YC + A\dot{Y}C + AY\dot{C} - \dot{B})C^T S_2 \\ & + S_1 A^T (AYC - B)\dot{C}^T S_2 + S_1 A^T (AYC - B)C^T \dot{S}_2 \\ & = -\lambda S_1 A^T (AYC - B)C^T S_2, \end{aligned} \tag{73}$$

or equivalently

$$\begin{aligned} S_1 A^T A\dot{Y}CC^T S_2 = & -\lambda S_1 A^T (AYC - B)C^T S_2 \\ & - \dot{S}_1 A^T (AYC - B)C^T S_2 - S_1 \dot{A}^T (AYC - B)C^T S_2 \\ & - S_1 A^T (\dot{A}YC + AY\dot{C} - \dot{B})C^T S_2 - S_1 A^T (AYC - B)\dot{C}^T S_2 \\ & - S_1 A^T (AYC - B)C^T \dot{S}_2, \end{aligned} \tag{74}$$

The vectorization and the Kronecker product transform (46) into the equivalent form

$$\begin{aligned} ((CC^T S_2)^T \otimes (S_1 A^T A))\text{vec}(\dot{Y}) = & \text{vec}(-\lambda S_1 A^T (AYC - B)C^T S_2 \\ & - \dot{S}_1 A^T (AYC - B)C^T S_2 - S_1 \dot{A}^T (AYC - B)C^T S_2 \\ & - S_1 A^T (\dot{A}YC + AY\dot{C} - \dot{B})C^T S_2 \\ & - S_1 A^T (AYC - B)\dot{C}^T S_2 - S_1 A^T (AYC - B)C^T \dot{S}_2). \end{aligned} \tag{75}$$

Then setting

$$\begin{aligned}
 W(t) &= ((CC^T S_2)^T \otimes (S_1 A^T A)) \\
 r(t) &= \text{vec}(-\lambda S_1 A^T (AYC - B) C^T S_2 \\
 &\quad - \dot{S}_1 A^T (AYC - B) C^T S_2 - S_1 \dot{A}^T (AYC - B) C^T S_2 \\
 &\quad - S_1 A^T (\dot{A}YC + AY\dot{C} - \dot{B}) C^T S_2 \\
 &\quad - S_1 A^T (AYC - B) C^T S_2 - S_1 A^T (AYC - B) C^T \dot{S}_2) \\
 \dot{\mathbf{y}}(t) &= \text{vec}(\dot{Y}(t)), \quad \mathbf{y}(t) = \text{vec}(Y(t)),
 \end{aligned}
 \tag{76}$$

the following system of differential equations is obtained:

$$W(t)\dot{\mathbf{y}}(t) = r(t). \tag{77}$$

Since $W(t)$ is singular in the case $\text{rank}(A) < \min\{m, n\}$ or $\text{rank}(C) < \min\{k, h\}$, the utilization of the Tikhonov principle leads to the invertible mass matrix

$$M(t) = \begin{cases} W(t), & \text{rank}(A) = \min\{m, n\} \ \& \ \text{rank}(C) = \min\{k, h\} \\ W(t) + \beta I_{kn}, & \text{rank}(A) < \min\{m, n\} \ \text{or} \ \text{rank}(C) < \min\{k, h\} \end{cases}
 \tag{78}$$

in which it is $\beta \geq 0$. As a result, the following dynamical evolution is used instead of (77):

$$M(t)\dot{\mathbf{y}}(t) = r(t), \tag{79}$$

where $M(t)$ is a nonsingular mass matrix. The ZNN flow (79) will be denoted as NZNNV3, and it is solvable using one of the ode Matlab solvers.

Theorem 5. *Let $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ be differentiable. If the condition (16) is satisfied, the NZNNV3 model (79), starting from arbitrary initial state $\mathbf{y}(0)$, exponentially converges to the theoretical TV solution of the TV GLMEs (65) in the form (18).*

Proof. Similar to the proof of Theorem 3. \square

3.6. Direct ZNN Model

The direct approach considers the smooth arbitrary TV matrices $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ for solving the GLME

$$Y(t) = A^\dagger(t)B(t)C^\dagger(t), \quad \{A, B, C\} \in \mathcal{D}_G, \tag{80}$$

where the desired solution is $Y(t) \in \mathbb{R}^{n \times k}$. The direct approach always calculates the solution produced directly by the pseudoinverse of $A(t)$ and $C(t)$. As a result, the direct ZNN (DZNN) model includes three error functions. The first error function in the DZNN model is defined by

$$E_1(t) = Y(t) - X(t)B(t)Z(t), \quad \{A, B, C\} \in \mathcal{D}_G, \tag{81}$$

wherein $X(t) \in \mathbb{R}^{n \times m}$ is the desired zero of the second error function for calculating $A^\dagger(t)$ (see [38]):

$$E_2(t) = A^T(t)(I_m - A(t)X(t)), \quad m \geq n \geq \text{rank}(A(t)), \tag{82}$$

and $Z(t) \in \mathbb{R}^{h \times k}$ is the zero of the third error function for finding $C^\dagger(t)$:

$$E_3(t) = (I_h - Z(t)C(t))C^T(t), \quad h \geq k \geq \text{rank}(C(t)). \tag{83}$$

The time derivative of (81) is defined by

$$\dot{E}_1(t) = \dot{Y} - \dot{X}BZ - X\dot{B}Z - XB\dot{Z}, \{A, B, C\} \in \mathcal{D}_G, \tag{84}$$

the time derivative of (82) is

$$\dot{E}_2(t) = \dot{A}^T(I_m - AX) - A^T\dot{A}X - A^T A\dot{X}, m \geq n \geq \text{rank}(A) \tag{85}$$

and the time derivative of (83) is given by

$$\dot{E}_3(t) = (I_h - ZC)\dot{C}^T - \dot{Z}CC^T - Z\dot{C}C^T, h \geq k \geq \text{rank}(C). \tag{86}$$

Then, considering the ZNN model (2), it can be obtained

$$\begin{cases} \dot{Y} - \dot{X}BZ - X\dot{B}Z - XB\dot{Z} = -\lambda(Y - XBZ), \\ \dot{A}^T(I_m - AX) - A^T\dot{A}X - A^T A\dot{X} = -\lambda A^T(I_m - AX), \\ (I_h - ZC)\dot{C}^T - \dot{Z}CC^T - Z\dot{C}C^T = -\lambda(I_h - ZC)C^T, \end{cases} \tag{87}$$

or in equivalent form

$$\begin{cases} \dot{Y} - \dot{X}BZ - XB\dot{Z} = -\lambda(Y - XBZ) + X\dot{B}Z, \\ -A^T A\dot{X} = -\lambda A^T(I_m - AX) - \dot{A}^T(I_m - AX) + A^T\dot{A}X, \\ -\dot{Z}CC^T = -\lambda(I_h - ZC)C^T - (I_h - ZC)\dot{C}^T + Z\dot{C}C^T. \end{cases} \tag{88}$$

Using the vectorization and the Kronecker product, (88) is transformed into

$$\begin{cases} I_{nk}\text{vec}(\dot{Y}) - ((BZ)^T \otimes I_n)\text{vec}(\dot{X}) - (I_k \otimes (XB))\text{vec}(\dot{Z}) = \text{vec}(-\lambda(Y - XBZ) + X\dot{B}Z) \\ (-I_m \otimes A^T A)\text{vec}(\dot{X}) = \text{vec}(-\lambda A^T(I_m - AX) - \dot{A}^T(I_m - AX) + A^T\dot{A}X) \\ (-CC^T \otimes I_h)\text{vec}(\dot{Z}) = \text{vec}(-\lambda(I_h - ZC)C^T - (I_h - ZC)\dot{C}^T + Z\dot{C}C^T). \end{cases} \tag{89}$$

Setting

$$\begin{aligned} w_1(t) &= I_{nk} \\ w_2(t) &= -(BZ)^T \otimes I_n, \\ w_3(t) &= -I_k \otimes (XB), \\ w_4(t) &= -I_m \otimes A^T A, \\ w_5(t) &= -CC^T \otimes I_h, \\ r_1(t) &= \text{vec}(-\lambda(Y - XBZ) + X\dot{B}Z) \\ r_2(t) &= \text{vec}(-\lambda A^T(I_m - AX) - \dot{A}^T(I_m - AX) + A^T\dot{A}X) \\ r_3(t) &= \text{vec}(-\lambda(I_h - ZC)C^T - (I_h - ZC)\dot{C}^T + Z\dot{C}C^T) \end{aligned} \tag{90}$$

$$W(t) = \begin{bmatrix} w_1 & w_2 & w_3 \\ O_{mn, kn} & w_4 & O_{mn, kh} \\ O_{kh, kn} & O_{kh, mn} & w_5 \end{bmatrix}, r(t) = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \dot{\mathbf{y}} = \begin{bmatrix} \text{vec}(\dot{Y}) \\ \text{vec}(\dot{X}) \\ \text{vec}(\dot{Z}) \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \text{vec}(Y) \\ \text{vec}(X) \\ \text{vec}(Z) \end{bmatrix},$$

the following differential system with the mass matrix $W(t)$ is obtained

$$W(t)\dot{\mathbf{y}}(t) = r(t). \tag{91}$$

Since $W(t)$ is singular when $\text{rank}(A(t)) < \min\{m, n\}$ or $\text{rank}(C(t)) < \min\{k, h\}$, the following regular mass matrix is used:

$$M(t) = \begin{cases} W(t), & \text{rank}(A(t)) = \min\{m, n\} \ \& \ \text{rank}(C(t)) = \min\{k, h\} \\ W(t) + \beta I_{mnkn}, & \text{rank}(A(t)) < \min\{m, n\} \ \text{or} \ \text{rank}(C(t)) < \min\{k, h\} \end{cases} \tag{92}$$

such that $\beta \geq 0$. As a result, the following ZNN model is used instead of (91):

$$M(t)\dot{y}(t) = r(t), \tag{93}$$

where $M(t)$ is a nonsingular mass matrix. The system (93) is termed as DZNN model, and it is solvable with an ode Matlab solver.

Theorem 6. Let $A(t) \in \mathbb{R}^{m \times n}$, $C(t) \in \mathbb{R}^{k \times h}$ and $B(t) \in \mathbb{R}^{m \times h}$ be differentiable. If the condition (16) is satisfied, the DZNN model (93) starting from any initial value $y(0)$, converges exponentially to the theoretical TV solution of the TV GLMEs (80) in the form (18).

Proof. Analogous to the proof of Theorem 3. \square

4. Simulation Examples

This section compares performances of the TZNN model (29), the GZNN design (40), the NZNNV1 model (51), the NZNNV2 model (64), the NZNNV3 model (79), and the DZNN model (93) on six numerical examples (NE), involving square or rectangular, as well as singular or nonsingular input TV matrices. The diagram in Figure 1 is presented to understand better the solutions produced by these six models.

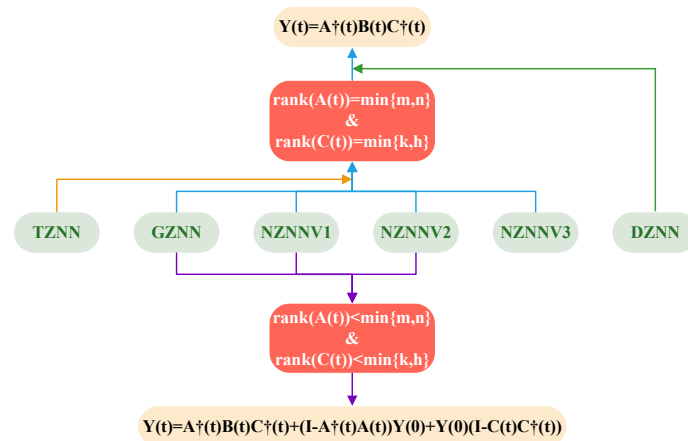


Figure 1. Solutions produced by proposed ZNN models.

This diagram shows how the GZNN, NZNNV1, NZNNV2, and NZNNV3 models generate various solutions for different initial conditions. In contrast, the TZNN and DZNN models generate just the pseudoinverse solution of the TV GLME (1) for any initial conditions.

As preliminaries, a few parameters and symbols must be defined, as well as some additional information for the subsequent NEs. The considered time interval is $[0, 10]$ in all NE, while the ZNN scaling parameter is assigned to $\lambda = 10$, and the regularization parameter is $\beta = 1e - 8$. The initial condition (IC1) in all NEs is

$$Y(0) = A^T(0)I_{m,h}C^T(0)$$

for all models, with $X(0) = A^T(0)$ and $Z(0) = C^T(0)$ for the NZNNV2 model (64), the NZNNV3 model (79), and the DZNN model (93). The second initial condition (IC2) for the last three NEs has been set as

$$Y(0) = (1 : k) \odot \mathbf{1}_{n \times k}$$

with $X(0) = A^T(0)$ and $Z(0) = C^T(0)$ for the NZNNV2 model (64), the NZNNV3 model (79) and the DZNN model (93), and $\gamma = 1e - 1$ for the NZNNV3 model (79). In the figures legends, the notation TZNN, GZNN, NZNNV1, NZNNV2, NZNNV3 and DZNN denote the solutions or errors produced by the corresponding models solutions,

while the notations $Y_1^*(t), Y_2^*(t), Y^*(t)$ refer to $Y_1^*(t) = A^\dagger BC^\dagger$ and $Y_2^*(t) = \tilde{Y}_{Y(0)}(t)$, and $Y^*(t) = Y_1^* = Y_2^*$.

4.1. Example 1

This NE is about a TV GLME that considers the following square matrices of dimensions 4×4 :

$$A(t) = \begin{bmatrix} 3 + \cos(t) & 1 + 1/2 \sin(t) & \dots & 1 + 1/n \sin(t) \\ 1 + 1/2 \sin(t) & 3 + \cos(t) & \dots & 1 + 1/(n - 1) \sin(t) \\ \dots & \dots & \dots & \dots \\ 1 + 1/m \sin(t) & 1 + 1/(m - 1) \sin(t) & \dots & 3 + \cos(t) \end{bmatrix},$$

$$C(t) = \begin{bmatrix} 5 + \cos(t) & -3 + \sin(t) \\ 3 + \cos(t) & 2 + \sin(t) \end{bmatrix},$$

$$B(t) = \begin{bmatrix} 7 + \cos(t) & 6 + \sin(2t) \\ 5 + \sin(t) & -4 - \sin(t) \\ 1 + \cos(t) & -6 + \cos(t) \\ 3 - \cos(t) & -2 + \cos(t) \end{bmatrix},$$

such that $\text{rank}(A(t)) = 4$. The input matrices $A(t), C(t)$ and $B(t)$ are nonsingular matrices.

4.2. Example 2

This NE is about a TV GLME that considers rectangular matrices

$$A(t) = \begin{bmatrix} 6 - \cos(\pi t) & -3 - \cos(\pi t) & 1 - \cos(\pi t) \\ 5 - \cos(\pi t) & -8 - \cos(\pi t) & 2 - \cos(\pi t) \end{bmatrix}^T,$$

$$C(t) = \begin{bmatrix} 5 + \cos(t) & -3 + \sin(t) & -2 + \sin(t) & -1 + \sin(t) \\ 3 + \cos(t) & 2 + \sin(t) & 3 + \sin(t) & 4 + \sin(t) \\ 5 + \cos(t) & 7 + \sin(t) & 3 + \sin(t) & 2 + \sin(t) \end{bmatrix},$$

$$B(t) = [2A(:,1) \quad 3A(:,2) \quad -3A(:,2) \quad -2A(:,1)],$$

such that $\text{rank}(A(t)) = 2$. That is, the input $A(t)$ is a full-column rank 3×2 matrix, and the input $C(t)$ is a full-row rank 3×4 matrix.

4.3. Example 3

This NE is about a TV GLME that takes into account the rectangular matrices below:

$$A(t) = \begin{bmatrix} 3 + \cos(t) & 1 + 1/2 \sin(t) & \dots & 1 + 1/n \sin(t) \\ 1 + 1/2 \sin(t) & 3 + \cos(t) & \dots & 1 + 1/(n - 1) \sin(t) \\ \dots & \dots & \dots & \dots \\ 1 + 1/m \sin(t) & 1 + 1/(m - 1) \sin(t) & \dots & 3 + \cos(t) \end{bmatrix},$$

$$C(t) = \begin{bmatrix} 5 + \cos(t) & -3 + \sin(t) & -2 + \sin(t) & -1 + \sin(t) \\ 3 + \cos(t) & 2 + \sin(t) & 3 + \sin(t) & 4 + \sin(t) \end{bmatrix},$$

$$B(t) = \begin{bmatrix} \text{diag}(1 : k) \\ \mathbf{1}_k \\ 2 \times \mathbf{1}_{2,k} \end{bmatrix} + \sin(t),$$

where $\text{rank}(A(t)) = \min\{m, n\}$ and $m = 10, n = 6$ and $k = 4$. That is, the input matrix $A(t)$ is a full-column rank 10×6 matrix, and the input $C(t)$ is a full-row rank 2×4 matrix.

4.4. Example 4

This example concerns a TV GLME that takes into account the input matrices

$$A(t) = \begin{bmatrix} 1 + \sin(t) \\ 1 + 1/2 \sin(t) \\ 1 + 1/3 \sin(t) \\ 1 + 1/4 \sin(t) \end{bmatrix} \odot \mathbf{1}_4,$$

$$C(t) = \begin{bmatrix} 5 + \cos(t) & -3 + \sin(t) \\ 5 + \cos(t) & -3 + \sin(t) \end{bmatrix},$$

$$B(t) = [3A(:,1) \quad 2A(:,4)]$$

such that $A(t)$ and $C(t)$ are singular and of dimensions 4×4 and 2×2 , respectively, with $\text{rank}(A(t)) = \text{rank}(C(t)) = 1$.

4.5. Example 5

This NE is about a TV GLME that takes the rectangular matrices below into account:

$$A(t) = [6 - \cos(\pi t) \quad -5 - \cos(\pi t) \quad 1 + \sin(\pi t) \quad 2 + \sin(t) \quad -2 + \sin(t)]^T \odot [1 \quad 2] \odot \mathbf{1}_{5,2},$$

$$C(t) = \begin{bmatrix} 5 & -3 & -2 & -1 & 2 & -3 \\ 8 & -4 & -2 & 7 & 2 & -3 \\ 5 & 7 & 3 & 2 & -3 & 2 \end{bmatrix} + \sin(t),$$

$$B(t) = [5A(:,1) \quad 2A(:,2) \quad \mathbf{1}_{5 \times 4}]$$

where $\text{rank}(A(t)) = 1$ and $\text{rank}(C(t)) = 1$. The input matrix $A(t)$ is a 5×2 rank-deficient matrix, and $C(t)$ is a full-row rank 3×6 matrix.

4.6. Example 6

This NE is about a TV GLME that takes the rectangular matrices below into consideration:

$$A(t) = [1 + \sin(t) \quad 1 + 1/2 \sin(t) \quad \dots \quad 1 + 1/m \sin(t)] \odot \mathbf{1}_{m \times n},$$

$$C(t) = [5 + \cos(t) \quad -3 + \sin(t) \quad -2 + \sin(t) \quad -1 + \sin(t)] \mathbf{1}_{k \times h},$$

$$B(t) = [3A(:,1) \quad 2A(:,1) \quad 5A(:,1) \quad \mathbf{1}_m],$$

where $\text{rank}(A(t)) = 1$ with $m = 8$ and $n = 5$, and $\text{rank}(C(t)) = 1$ with $k = 3$ and $h = 4$. The input matrix $A(t)$ is rank-deficient and of size 8×5 , and $C(t)$ is a rank deficient 3×4 matrix.

4.7. General Discussion

Since $A(t)$ and $C(t)$ in Example 1 are nonsingular, the solvability condition (16) is satisfied. As a consequence, the TV GLME (1) is solvable with the unique solution $Y_1^*(t) = A^\dagger BC^\dagger$. On the other hand, the solvability condition (16) for the matrix equation $E(t) = 0$ is not satisfied in all another Examples 2–6. So, the equation $E(t) = 0$ is not solvable in Examples 2–6. On the other hand, according to Theorem 1, the matrix equations $E_G(t) = 0$ and $E_N(t) = 0$ are solvable in all examples.

In this subsection, the performance of six proposed ZNN models for solving TV GLME (1) is investigated through six NEs. All the models generate the solution $Y_1^*(t)$, i.e., the pseudoinverse solution of the TV GLME (1), under the IC1 in all NEs, while the GZNN, NZNNV1, NZNNV2, and NZNNV3 models generate the solution $Y_2^*(t)$ under the IC2 in NEs Sections 4.4–4.6. More precisely, NEs Sections 4.1–4.3, respectively, deal with square nonsingular $A(t)$ and $C(t)$, a rectangular full column rank $A(t)$, and a rectangular full row rank $C(t)$. As a result, all the models generate the pseudoinverse solution of the TV GLME (1), i.e., $Y^*(t) = Y_1^*(t) = Y_2^*(t)$, for any initial condition. It is worth noting that the TZNN is only applicable in NE Section 4.1, which has square nonsingular input

matrices $A(t)$ and $C(t)$. To substantiate the claim mentioned above, the TZNN has been used in NE Section 4.2, where Figure 2b demonstrates that the TZNN error matrix does not converge to zero. Furthermore, NE Section 4.4 deals with square singular $A(t)$ and $C(t)$, and NEs Sections 4.5 and 4.6 deal with rectangular rank deficient $A(t)$ and $C(t)$. As a result, the GZNN, NZNNV1, NZNNV2, and NZNNV3 generate the solution $Y_1^*(t)$ under the IC1 and the solution $Y_2^*(t)$ under the IC2. It is worth noting that $Y_1^*(t) \neq Y_2^*(t)$ in NEs Sections 4.4–4.6 under the IC2.

Figures 2–4 follow the next schedule: first rows in the figures display the convergence of the error functions involved in the tested models, i.e., (25) of TZNN, (33) of GZNN, (41) of NZNNV1, (52) and (53) of NZNNV2 (NZNNV2-E1 and NZNNV2-E2), (66) of NZNNV3, (81) and (82) of DZNN (DZNN-E1 and DZNN-E2) Frobenius norms; the second rows in figures exhibit the convergence behavior of the models on Error 1: $\|Y(t) - Y_2^*(t)\|_F$, where $Y(t)$ refers to solutions generated by considered models; the third rows in the figures display trajectories of generated solutions $Y(t)$; the fourth rows show the convergence of the models on Error 2: $\|A(t)Y(t)C(t) - B(t)\|_F$ in NE Section 4.1, and Error 2: $\|A^T(t)(A(t)Y(t)C(t) - B(t))C^T(t)\|_F$ in NEs Sections 4.2–4.6. Note also that Figure 2 contains results about NEs Sections 4.1–4.3, Figure 3 contains results about NEs Sections 4.4–4.6 under the IC1, and Figure 4 contains results about NEs Sections 4.4–4.6 under the IC2.

The following can be deduced from the NEs. The exponential convergence of the models that is proven in Theorems 2–6 can be observed in the first row figures. Particularly, Figure 2a–c present the exponential convergence of the models in NEs Sections 4.1–4.3, respectively, whereas Figures 3a–c and 4a–c present the exponential convergence of the models in NEs Sections 4.4–4.6 under IC1 and IC2, respectively. More particularly, while all models begin with an initial value other than the optimal one, all NEs have received differentiable $A(t)$, $B(t)$, and $C(t)$ matrices that meet the condition (16). In Figures 2a–c, 3a–c and 4a–c, the error functions convergence begins at $t = 0$ in the range $[10^1, 10^6]$, but it ends before $t = 2$ with lowest values in the range $[10^{-10}, 10^{-1}]$. It is also crucial to note that when the value of the design parameter λ is higher, the models will converge faster and the overall error will be lowered even further. Furthermore, all other figures behave in the same way due to the convergence tendency of the error functions. In other words, the graphs associated with the models in the remaining figures begin at $t = 0$ in a very different value from the objective and reach it before $t = 2$. As a result, it is evident that Theorems 2–6 are proven true.

In general, according to the first row figures, all the ZNN models converge to zero, where the NZNNV1 and NZNNV3 models have the fastest convergence rate, and the TZNN and DZNN have the second and third fastest, respectively. In contrast, the GZNN and NZNNV2 models have the slowest and almost similar convergence rates. According to the second row figures under IC1 and fourth row figures, the NZNNV1 and NZNNV3 models have the fastest convergence rate, the GZNN has the second fastest, and the NZNNV2 model has the slowest. The NZNNV2 and DZNN models have the lowest overall errors when $A(t)$ and $C(t)$ are rectangular, specifically in NEs Sections 4.2 and 4.5. The second row figures under IC2 show that the NZNNV1, NZNNV2, NZNNV3, and GZNN models perform similarly to the second row figures under IC1. However, the DZNN model does not converge to zero, as expected. Finally, the third row figures demonstrate that under IC1, all models solutions match $Y^*(t)$, however under IC2, the DZNN model solutions match $Y_1^*(t)$, while the NZNNV1, NZNNV2, NZNNV3, and GZNN models solutions match $Y_2^*(t)$.

Graphs included in Figures 3a–c and 4a–c, on the other hand, show that the IC1 and IC2 have no effect on the models' convergence, proving the Theorems 2–6 claim that the models' performance is unaffected by the initial condition.

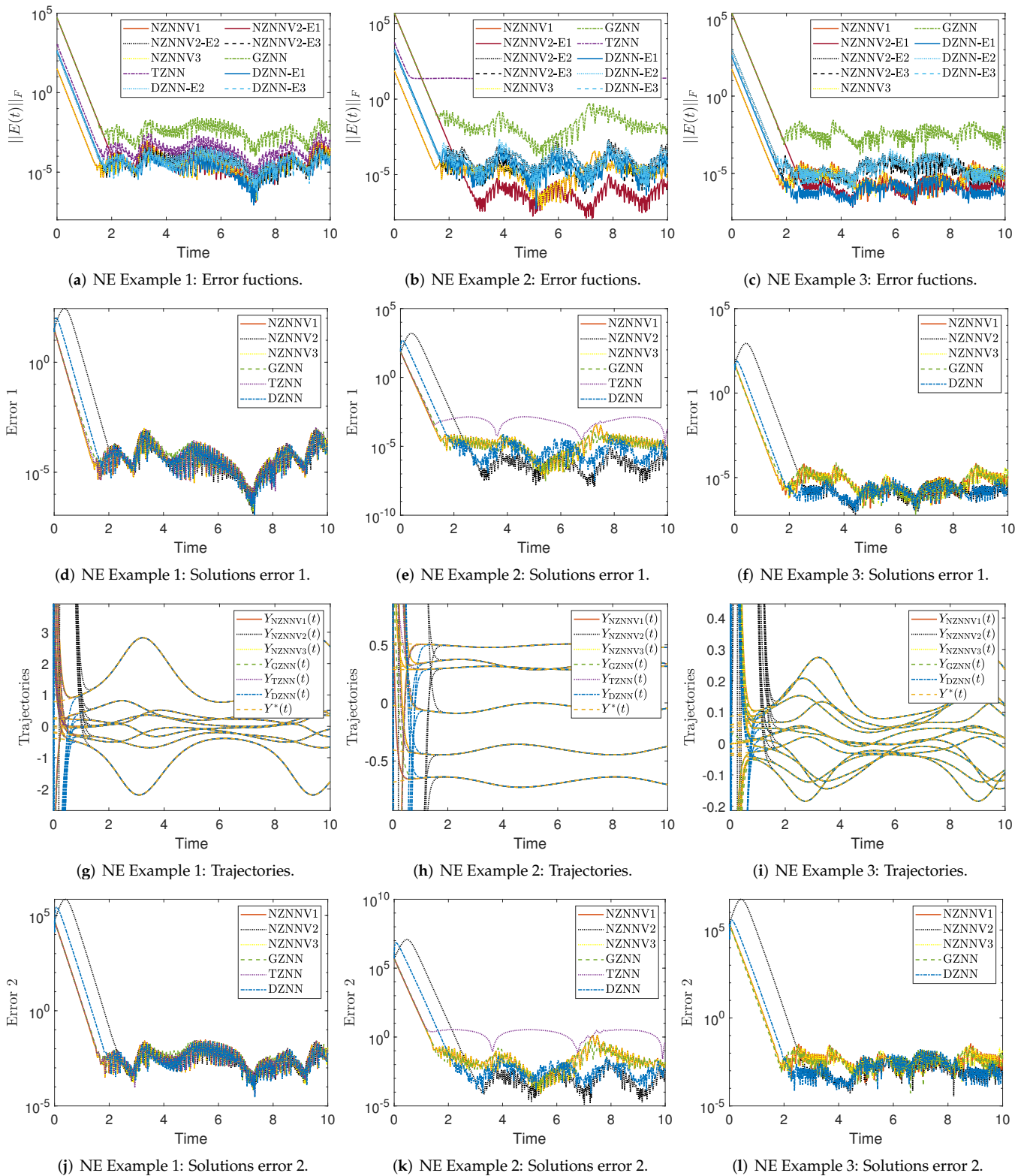


Figure 2. Errors and trajectories in NE Sections 4.1–4.3.

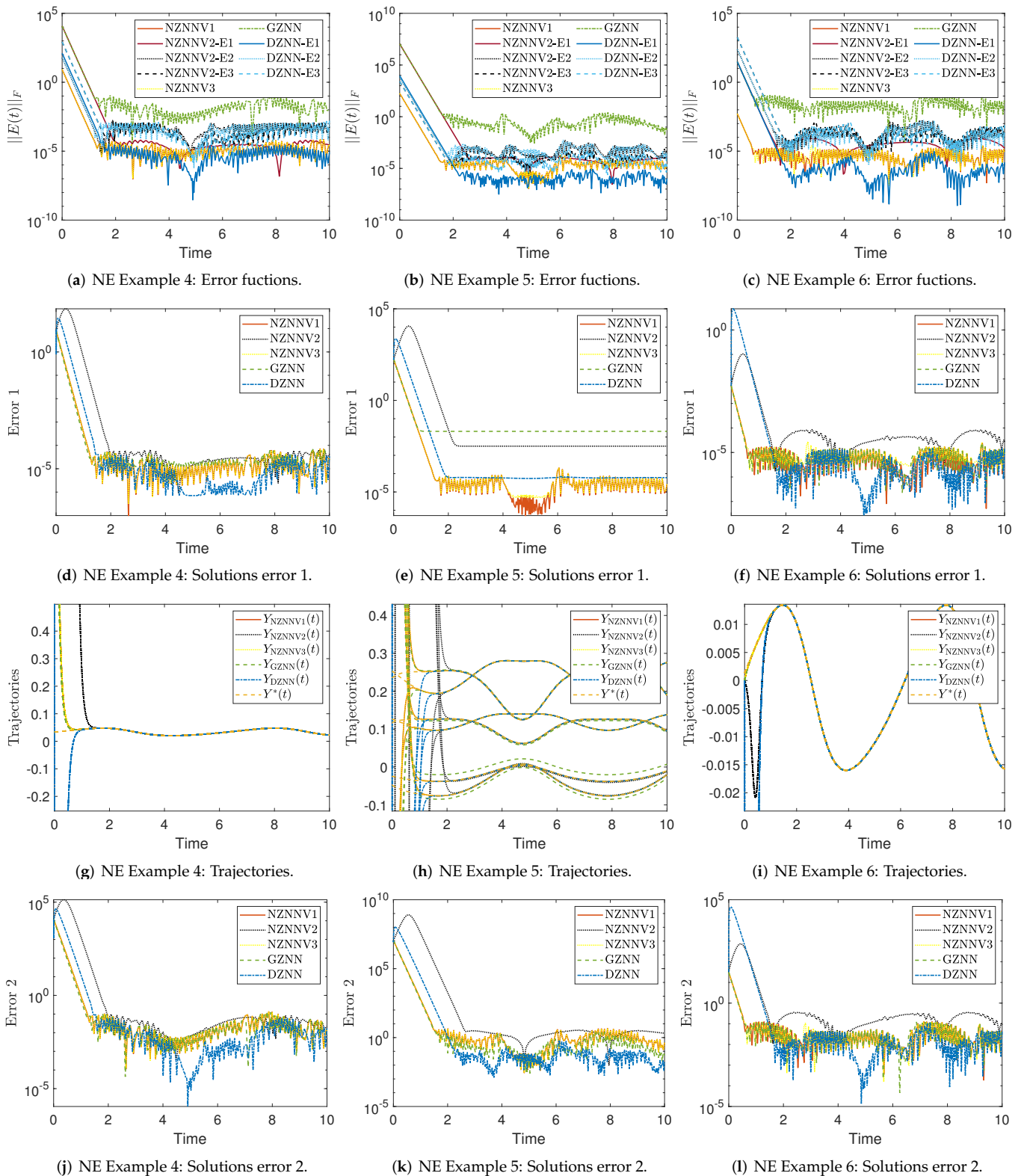


Figure 3. Errors and trajectories in NE Sections 4.4–4.6 under IC1.

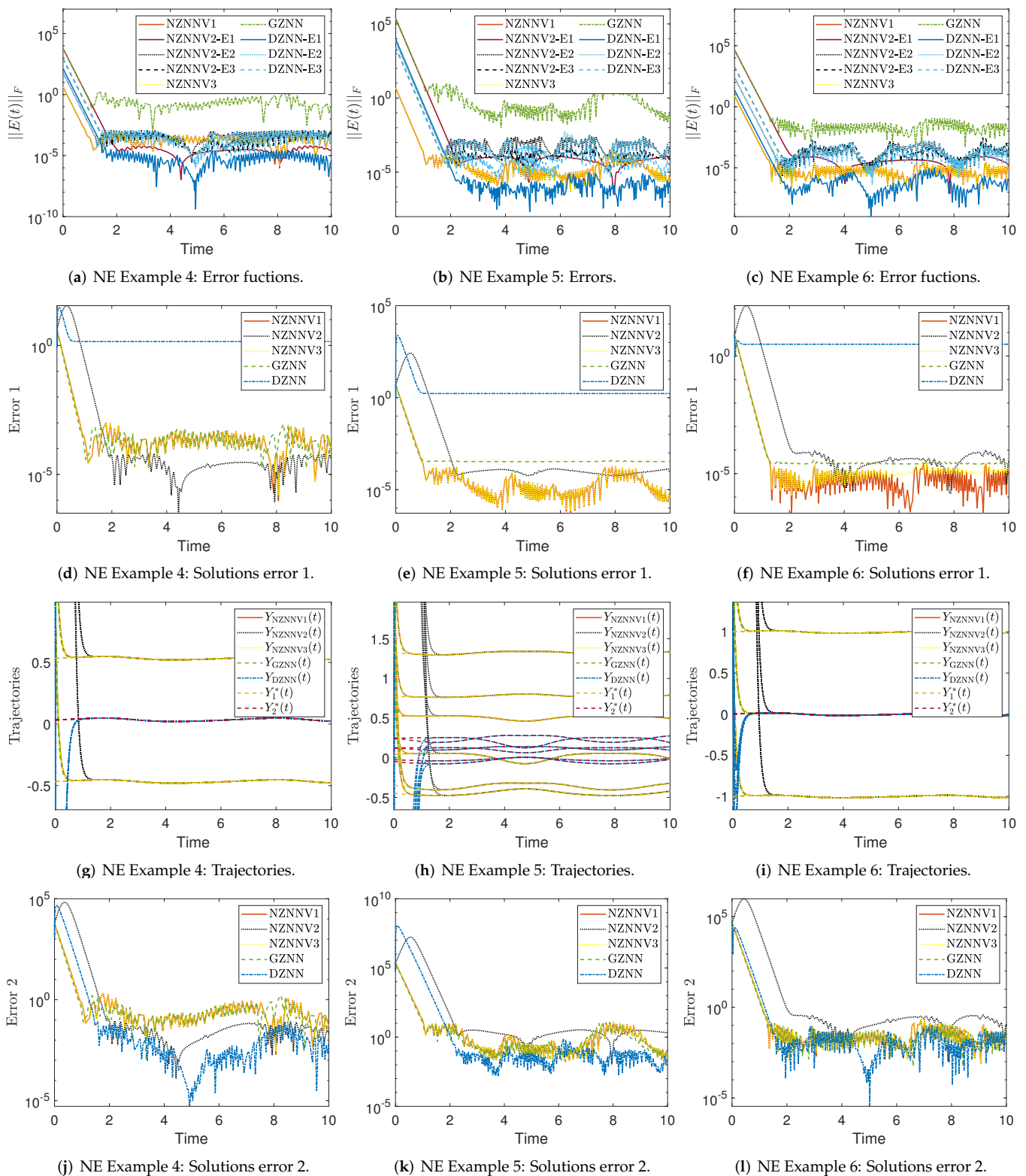


Figure 4. Errors and trajectories in NE Sections 4.4–4.6 under IC2.

To summarize, the TZNN is only applicable in the case where $m = n = \text{rank}(A(t))$ and $k = h = \text{rank}(C(t))$. It is important to mention that $A(t)$ becomes $C(t)$ and $C(t)$ becomes $A(t)$ when $n < m$ and $k < h$. As a result, the cases where $n < m$ and $k < h$ is disregarded in the NEs. The GZNN has a lower overall error than NZNNV1 and NZNNV3, although they have the same convergence speed. Even though the NZNNV2 has the

slowest convergence speed, its overall error is between NZNNV1 and GZNN. Compared to GZNN, NZNNV1, and NZNNV3, the DZNN model's convergence speed is slightly slower, but its overall error is the lowest. Remember that the DZNN model's primary distinction from all other models is that, regardless of the initial conditions, it generates only a single solution based on the pseudoinversion. That is, all the models work excellently in solving the TV GLMEs (1).

5. Conclusions

The problem of solving TV GLMEs with random TV real input matrices is resolved in this paper by applying the ZNN design. Six ZNN models for calculating the online solution to general TV GLMEs are defined, analyzed, and compared, including TZNN, which is based on the traditional ZNN approach, GZNN which is based on the gradient ZNN approach, NZNNV1, NZNNV2, and NZNNV3 which are based on the Newton's optimization method, and DZNN which is based on the ZNN approach for finding the direct solution produced with the use of the pseudoinverse. Six numerical examples involving square or rectangular, and singular or nonsingular matrices show that all models successfully solve TV GLMEs. However, their effectiveness varies and depends on the input matrix, while the NZNNV1 and NZNNV3 models converge to exact solutions faster than the other models.

There are a few prospective study areas that can be explored.

1. The streams of NZNNV1, NZNNV2, NZNNV3, and DZNN models accelerated by appropriate nonlinear activations, as well as nonlinear NZNNV1, NZNNV2, NZNNV3, and DZNN model flows with a finite convergence, can all be investigated.
2. Another research stream is to use carefully selected parameters defined in fuzzy environments. Such research will be a continuation of research presented in [25,26,38,39].
3. The presented TZNN, GZNN, NZNNV1, NZNNV2, NZNNV3, and DZNN models are not noise-tolerant, so all noise types have a significant influence on the accuracy of the suggested ZNNs. Consequently, related future investigation can be oriented on adjusting the proposed dynamical systems to appropriate more general integration-enhanced and noise-tolerant ZNN classes.
4. Numerous results concerning the solution of important equations of a special type have appeared in recent years. For example, there is a number of recent papers concerning the Sylvester equation [40,41] or related to Lyapunov equation [42–44]. Studying such equations can be one of the goals of our research.

Author Contributions: P.S.S.: conceptualization, methodology, validation, formal analysis, investigation, writing—original draft, writing-review & editing. S.D.M.: data curation, validation, investigation, formal analysis, writing-review & editing. V.N.K. (Vasilios N. Katsikis): conceptualization, methodology, validation, formal analysis, investigation, writing—original draft. L.A.K.: methodology, validation, investigation. V.N.K. (Vladimir N. Krutikov): methodology, formal analysis, investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Predrag Stanimirović acknowledges support of Ministry of Education, Science and Technological Development, Republic of Serbia, Grant No. 451-03-68/2022-14/200124, and support of Science Fund of the Republic of Serbia, (No. 7750185, Quantitative Automata Models: Fundamental Problems and Applications—QUAM).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jin, L.; Li, S.; La, H.M.; Luo, X. Manipulability optimization of redundant manipulators using dynamic neural networks. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4710–4720. [[CrossRef](#)]
2. Luo, X.; Zhou, M.; Xia, Y.; Zhu, Q.; Ammari, A.C.; Alabdulwahab, A. Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 524–537. [[CrossRef](#)] [[PubMed](#)]
3. Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S.; Zhang, Y. Solving complex-valued time-varying linear matrix equations via QR decomposition with applications to robotic motion tracking and on angle-of-arrival localization. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 3415–3424. [[CrossRef](#)] [[PubMed](#)]
4. Simos, T.E.; Katsikis, V.N.; Mourtas, S.D. Multi-input bio-inspired weights and structure determination neuronet with applications in European Central Bank publications. *Math. Comput. Simul.* **2022**, *193*, 451–465. [[CrossRef](#)]
5. Zhang, Y.; Ge, S.S. Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans. Neural Netw.* **2005**, *16*, 1477–1490. [[CrossRef](#)]
6. Zhang, Y.; Chen, K. Comparison on Zhang neural network and gradient neural network for time-varying linear matrix equation $AXB = C$ solving. In Proceedings of the 2008 IEEE International Conference on Industrial Technology, Chengdu, China, 21–24 April 2008.
7. Xiao, L.; Zhang, Y. From Different Zhang Functions to Various ZNN Models Accelerated to Finite-Time Convergence for Time-Varying Linear Matrix Equation. *Neural Process. Lett.* **2014**, *39*, 309–326. [[CrossRef](#)]
8. Stanimirović, P.; Petković, M. Gradient neural dynamics for solving matrix equations and their applications. *Neurocomputing* **2018**, *306*, 200–212. [[CrossRef](#)]
9. Stanimirović, P.S.; Katsikis, V.N.; Jin, L.; Mosić, D. Properties and computation of continuous-time solutions to linear systems. *Appl. Math. Comput.* **2021**, *405*, 16. [[CrossRef](#)]
10. Li, J.; Shi, Y.; Sun, Y.; Fan, J.; Mirza, A. Noise-tolerant Zeroing Neural Dynamics for solving hybrid multilayered time-varying linear equation system. *Secur. Commun. Netw.* **2022**, *2022*, 6040463. [[CrossRef](#)]
11. Guo, D.; Yi, C.; Zhang, Y. Zhang neural network versus gradient-based neural network for time-varying linear matrix equation solving. *Neurocomputing* **2011**, *74*, 3708–3712. [[CrossRef](#)]
12. Jin, L.; Li, S.; Liao, B.; Zhang, Z. Zeroing neural networks: A survey. *Neurocomputing* **2017**, *267*, 597–604. [[CrossRef](#)]
13. Lin, J.; Chen, W.; Zhao, L.; Chen, L.; Tang, Z. A nonlinear zeroing neural network and its applications on time-varying linear matrix equations solving, electronic circuit currents computing and robotic manipulator trajectory tracking. *Comput. Appl. Math.* **2022**, *41*, 319. [[CrossRef](#)]
14. Xiao, L.; Jia, L.; Zhang, Y.; Hu, Z.; Dia, J. Finite-time convergence and robustness analysis of two nonlinear activated ZNN models for time-varying linear matrix equations. *IEEE Access* **2019**, *7*, 135133–135144. [[CrossRef](#)]
15. Zhang, Z.; Deng, X.; Qu, X.; Liao, B.; Kong, L.-D.; Li, L. A varying-gain recurrent neural network and its application to solving online time-varying matrix equation. *Comput. Appl. Math.* **2016**, *4*, 77940–77952. [[CrossRef](#)]
16. Xiao, L. A finite-time convergent neural dynamics for online solution of time-varying linear complex matrix equation. *Neurocomputing* **2015**, *167*, 254–259. [[CrossRef](#)]
17. Li, X.; Yu, J.; Li, S.; Ni, L. A nonlinear and noise-tolerant znn model solving for time-varying linear matrix equation. *Neurocomputing* **2018**, *317*, 70–78. [[CrossRef](#)]
18. Wang, X.; Liang, L.; Che, M. Finite-time convergent complex-valued neural networks for the time-varying complex linear matrix equations. *Eng. Lett.* **2018**, *26*, 432–440.
19. Mo, C.; Gerontitis, D.; Stanimirović, P. Solving the time-varying tensor square root equation by varying-parameters finite-time Zhang neural network. *Neurocomputing* **2021**, *445*, 309–325. [[CrossRef](#)]
20. Wang, X.; Che, M.; Wei, Y. Recurrent neural network for computation of generalized eigenvalue problem with real diagonalizable matrix pair and its applications. *Neurocomputing* **2016**, *216*, 230–241. [[CrossRef](#)]
21. Li, X.; Li, S.; Xu, Z.; Zhou, X. A Vary-parameter convergence-accelerated recurrent neural network for online solving dynamic matrix pseudoinverse and its robot application. *Neural Process. Lett.* **2021**, *53*, 1287–1304. [[CrossRef](#)]
22. Zhang, Z.; Yang, S.; Zheng, L. A penalty strategy combined varying-parameter recurrent neural network for solving time-varying multi-type constrained quadratic programming problems. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 2993–3004. [[CrossRef](#)]
23. Katsikis, V.N.; Stanimirović, P.S.; Mourtas, S.D.; Li, S.; Cao, X. *Generalized Inverses: Algorithms and Applications*, Ivan Kyrchei, I., Ed.; Towards Higher Order Dynamical Systems, Mathematics Research Developments, Nova Science Publishers, Inc.: New York, NY, USA, 2021; pp. 207–239.
24. Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S.; Zhang, Y. Continuous-time varying complex QR decomposition via zeroing neural dynamics. *Neural Process. Lett.* **2021**, *53*, 3573–3590. [[CrossRef](#)]
25. Dai, J.; Yang, X.; Xiao, L.; Jia, L.; Li, Y. ZNN with fuzzy adaptive activation functions and its application to time-varying linear matrix equation. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2560–2570. [[CrossRef](#)]
26. Dai, J.; Yang, X.; Xiao, L.; Jia, L.; Li, Y. A fuzzy adaptive zeroing neural network with superior finite-time convergence for solving time-variant linear matrix equations. *Knowl.-Based Syst.* **2022**, *242*, 108405. [[CrossRef](#)]
27. Stanimirović, P.S.; Katsikis, V.N.; Li, S. Hybrid GNN-ZNN models for solving linear matrix equations. *Neurocomputing* **2018**, *316*, 124–134. [[CrossRef](#)]

28. Xiao, L.; Tan, H.; Dai, J.; Jia, L.; Tang, W. High-order error function designs to compute time-varying linear matrix equations. *Inf. Sci.* **2021**, *576*, 173–186. [[CrossRef](#)]
29. Zhang, Y.; Guo, D. *Zhang Functions and Various Models*; Springer: Berlin/Heidelberg, Germany, 2015.
30. Zhang, Y.; Yi, C. *Zhang Neural Networks and Neural-Dynamic Method*; Nova Science Publishers, Inc.: New York, NY, USA, 2011.
31. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer: New York, NY, USA, 1999.
32. Djordjević, D.S.; Stanimirović, P.S. Iterative methods for computing generalized inverses related with optimization methods. *J. Aust. Math. Soc.* **2005**, *78*, 257–272. [[CrossRef](#)]
33. Wang, J. A recurrent neural network for real-time matrix inversion. *Appl. Math. Comput.* **1993**, *55*, 89–100. [[CrossRef](#)]
34. Wang, J. Recurrent neural networks for computing pseudoinverse of rank-deficient matrices. *SIAM J. Sci. Comput.* **1997**, *18*, 1479–1493. [[CrossRef](#)]
35. Ben-Israel, A.; Greville, T.N.E. *Generalized Inverses: Theory and Applications*, 2nd ed.; CMS Books in Mathematics; Springer: New York, NY, USA, 2003.
36. Maher, P.J. Some operator inequalities concerning generalized inverses. *Ill. J. Math.* **1990**, *34*, 503–514. [[CrossRef](#)]
37. Penrose, R. A generalized inverse for matrices. *Proc. Camb. Phil. Soc.* **1955**, *51*, 406–413. [[CrossRef](#)]
38. Katsikis, V.N.; Stanimirović, P.S.; Mourtas, S.D.; Xiao, L.; Karabasević, D.; Stanujkić, D. Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 3426–3435. [[CrossRef](#)]
39. Dai, J.; Luo, L.; Xiao, L.; Jia, L.; Li, X. An intelligent fuzzy robustness ZNN model with fixed-time convergence for time-variant Stein matrix equation. *Int. J. Intell. Syst.* **2022**. doi: 10.1002/int.23058. [[CrossRef](#)]
40. Xiao, L.; Zhang, Y.; Dai, J.; Li, J.; Li, W. New Noise-Tolerant ZNN Models With Predefined-Time Convergence for Time-Variant Sylvester Equation Solving. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 3629–3640. [[CrossRef](#)]
41. Xiao, L.; He, Y. A Noise-Suppression ZNN Model With New Variable Parameter for Dynamic Sylvester Equation. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7513–7522. [[CrossRef](#)]
42. He, Y.; Xiao, L.; Sun, F.; Wang, Y. A variable-parameter ZNN with predefined-time convergence for dynamic complex-valued Lyapunov equation and its application to AOA positioning. *Appl. Soft Comput.* **2022**, *130*, 109703. 09703. [[CrossRef](#)]
43. Liao, B.; Hua, C.; Cao, X.; Katsikis, V.N.; Li, S. Complex Noise-Resistant Zeroing Neural Network for Computing Complex Time-Dependent Lyapunov Equation. *Mathematics* **2022**, *10*, 2817. [[CrossRef](#)]
44. Sun, M.; Liu, J. A novel noise-tolerant Zhang neural network for time-varying Lyapunov equation. *Adv. Contin. Discret. Model.* **2020**, *2020*, 116. [[CrossRef](#)]