axioms

MDPI

Article

Application of Gradient Optimization Methods in Defining Neural Dynamics

Predrag S. Stanimirović [1,2], Nataša Tešić [3], Dimitrios Gerontitis [4], Gradimir V. Milovanović [5], Milena J. Petrović [6,*], Vladimir L. Kazakovtsev [2] and Vladislav Stasiuk [2]

[1] Faculty of Sciences and Mathematics, University of Niš, 18000 Niš, Serbia; pecko@pmf.ni.ac.rs
[2] Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia; vokz@bk.ru (V.L.K.); vstasyuk@sfu-kras.ru (V.S.)
[3] Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, 21000 Novi Sad, Serbia; natasa.tesic27@gmail.com
[4] Department of Information and Electronic Engineering, International Hellenic University, 57400 Thessaloniki, Greece; dimitrios_gerontitis@yahoo.gr or dimger@iee.ihu.gr
[5] Mathematical Institute, Serbian Academy of Sciences and Arts, Kneza Mihaila 35, 11000 Belgrade, Serbia; gvm@mi.sanu.ac.rs
[6] Faculty of Sciences and Mathematics, University of Pristina in Kosovska Mitrovica, Lole Ribara 29, 38220 Kosovska Mitrovica, Serbia
* Correspondence: milena.petrovic@pr.ac.rs; Tel.: +381-28-425-396

Abstract: Applications of gradient method for nonlinear optimization in development of Gradient Neural Network (GNN) and Zhang Neural Network (ZNN) are investigated. Particularly, the solution of the matrix equation $AXB = D$ which changes over time is studied using the novel GNN model, termed as GGNN(A, B, D). The GGNN model is developed applying GNN dynamics on the gradient of the error matrix used in the development of the GNN model. The convergence analysis shows that the neural state matrix of the GGNN(A, B, D) design converges asymptotically to the solution of the matrix equation $AXB = D$, for any initial state matrix. It is also shown that the convergence result is the least square solution which is defined depending on the selected initial matrix. A hybridization of GGNN with analogous modification GZNN of the ZNN dynamics is considered. The Simulink implementation of presented GGNN models is carried out on the set of real matrices.

Keywords: gradient neural network; generalized inverses; Moore–Penrose inverse; linear matrix equations

MSC: 68T05; 15A09; 65F20

check for updates

Citation: Stanimirović, P.S.; Tešić, N.; Gerontitis, D.; Milovanović, G.V.; Petrović, M.J.; Kazakovtsev, V.L.; Stasiuk V. Application of Gradient Optimization Methods in Defining Neural Dynamics. *Axioms* **2024**, *13*, 49. https://doi.org/10.3390/axioms13010049

Academic Editor: Feliz Manuel Minhós

Received: 1 November 2023
Revised: 24 December 2023
Accepted: 11 January 2024
Published: 14 January 2024

1. Introduction and Background

Recurrent neural networks (RNNs) are an important class of algorithms for computing matrix (generalized) inverses. These algorithms are used to find the solutions of matrix equations or to minimize certain nonlinear matrix functions. RNNs are divided into two subgroups: Gradient Neural Networks (GNNs) and Zhang Neural Networks (ZNNs). The GNN design is explicit and mostly applicable to time-invariant problems, which means that the coefficients of the equations that are addressed are constant matrices. ZNN models can be implicit and are able to solve time-varying problems, where the coefficients of the equations depend on the variable $t \in \mathbb{R}, t > 0$, representing time [1–3].

The Moore–Penrose inverse of $A \in \mathrm{R}^{p\times n}$ is the unique matrix $A^{\dagger} = X \in \mathrm{R}^{n\times p}$ which is the solution to the well-known Penrose equations [4,5]:

$$A = AXA, \qquad X = XAX, \qquad AX = (AX)^{\mathrm{T}}, \qquad XA = (XA)^{\mathrm{T}},$$

where $()^{\mathrm{T}}$ denotes the transpose matrix. The rank of a matrix A, i.e., the maximum number of linearly independent columns in A, is denoted by rank(A).

Applications of linear algebra tools and generalized inverses can be found in important areas such as the modeling of electrical circuits [6], the estimation of DNA sequences [7] and the balancing of chemical equations [8,9], as well as in other important research domains related to robotics [10] and statistics [11]. A number of iterative methods for solving matrix equations based on gradient values have been proposed [12–15].

In the following sections, we will focus on GNN and ZNN dynamical systems based on the gradient of the objective function and their implementation. The main goal of this research is the analysis of convergence and the study of analytic solutions.

Models with GNN neural designs for computing the inverse or the Moore–Penrose inverse and linear matrix equations were proposed in [16–19]. Further, various dynamical systems aimed at approximating the pseudo-inverse of rank-deficient matrices were developed in [16]. Wei, in [20], proposed three RNN models for the approximation of the weighted Moore–Penrose inverse. Online matrix inversion in a complex matrix case was considered in [21]. A novel GNN design based on nonlinear activation functions (AFs) was proposed and analyzed in [22,23] for solving the constant Lyapunov matrix equation online. A fast convergent GNN aimed at solving a system of linear equations was proposed and numerically analyzed in [24]. Xiao, in [25], investigated the finite-time convergence of an appropriately accelerated ZNN for the online solution of the time-varying complex matrix equation $A(t)X(t) = B(t)$. A comparison with the corresponding GNN design was considered. Two improved nonlinear GNN dynamical systems for approximating the Moore–Penrose inverse of full-row or full-column rank matrices were proposed and considered in [26]. GNN-type models for solving matrix equations and computing related generalized inverses were developed in [1,3,13,16,18,20,27–29]. The acceleration of GNN dynamics to a finite-time convergence has been investigated recently. A finite-time convergent GNN for approximating online solutions of the general linear matrix equation $AX(t)B + CX(t)D = B$ was proposed in [30]. This goal was achieved using two activation functions (AFs) in the construction of the GNN. The influence of AFs on the convergence performance of a GNN design for solving the matrix equation $AXB + X = C$ was investigated in [31]. A fixed-time convergent GNN for solving the Sylvester equation was investigated in [32]. Moreover, noise-tolerant GNN models equipped with a suitable activation function (AF) able to solve convex optimization problems were developed in [33].

Our goal is to solve the equation $AXB = D$ and apply its particular cases in computing generalized inverses in real time by improving the GNN model developed in [34]. The developed dynamical system is denoted by $\text{GNN}(A, B, D)$. Or motivation is to improve the GNN model denoted by $\text{GNN}(A, B, D)$ and develop a novel gradient-based GGNN model, termed $\text{GGNN}(A, B, D)$, utilizing a novel type of dynamical system. The proposed GGNN model is based on the standard GNN dynamics along the gradient of the standard error matrix. The convergence analysis reveals the global asymptotic convergence of $\text{GGNN}(A, B, D)$ without restrictions, while the output belongs to the set of general solutions to the matrix equation $AXB = D$.

In addition, we propose gradient-based modifications of the hybrid models developed in [35] as proper combinations of GNN and ZNN models for solving the matrix equations $BX = D$ and $XC = D$ with constant coefficients. Analogous hybridizations for approximating the matrix inverse were developed in [36], while two modifications of the ZNN design for computing the Moore–Penrose inverse were proposed in [37]. Hybrid continuous-gradient–Zhang neural dynamics for solving linear time-variant equations were investigated in [38,39]. The developed hybrid GNN-ZNN models in this paper are aimed at solving the matrix equations $AX = B$ and $XC = D$, denoted by $\text{HGZNN}(A, I, B)$ and $\text{HGZNN}(I, C, D)$, respectively.

The implementation was performed in MATLAB Simulink, and numerical experiments were performed with simulations of the GNN, GGNN and HGZNN models.

The GNN used to solve the general linear matrix equation $AXB = D$ is defined over the error matrix $E(t) = D - AV(t)B$, where $t \in [0, +\infty)$ is time, and $V(t)$ is an unknown state-variable matrix that approximates the unknown matrix X in $AXB = D$. The goal function is $\varepsilon(t) = ||D - AV(t)B||_F^2/2$, where $\| \cdot \|_F = \sqrt{\sum_{ij} a_{ij}^2}$ denotes the Frobenius norm of a matrix. The gradient of $\varepsilon(t)$ is equal to

$$\frac{\partial \varepsilon(t)}{\partial V} = \nabla \varepsilon = \frac{1}{2} \frac{\partial ||D - AV(t)B||_F^2}{\partial V} = -A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}}.$$

The GNN evolutionary design is defined by the dynamic system

$$\dot{V}(t) = \frac{\mathrm{d}V(t)}{\mathrm{d}t} = -\gamma \frac{\partial \varepsilon(t)}{\partial V}, \;\; V(0) = V_0, \tag{1}$$

where $\gamma > 0$ is a real parameter used to speed up the convergence, and $\dot{V}(t)$ denotes the time derivative of $V(t)$. Thus, the linear GNN aimed at solving $AXB = D$ is given by the following dynamics:

$$\dot{V}(t) = \gamma A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}}. \tag{2}$$

The dynamical flow (2) is denoted as $\mathrm{GNN}(A, B, D)$. The nonlinear $\mathrm{GNN}(A, B, D)$ for solving $AXB = D$ is defined by

$$\dot{V}(t) = \gamma A^{\mathrm{T}} \mathcal{F}(D - AV(t)B)B^{\mathrm{T}}. \tag{3}$$

The function array $\mathcal{F}(C) = \mathcal{F}([c_{ij}])$ is based on the appropriate odd and monotonically increasing activation function, which is applicable to the elements of a real matrix $C = (c_{ij}) \in \mathbb{R}^{m \times n}$, i.e., $\mathcal{F}(C) = [f(c_{ij})], i = 1, \ldots, m, j = 1, \ldots, n,$.

Proposition 1 restates restrictions on the solvability of $AXB = D$ and its general solution.

Proposition 1 ([4,5])**.** *If $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{p \times q}$ and $D \in \mathbb{R}^{m \times q}$, then the fulfillment of the condition*

$$AA^{\dagger}DB^{\dagger}B = D \tag{4}$$

is necessary and sufficient for the solvability of the linear matrix equation $AXB = D$. In this case, the set of all solutions is given by

$$X = \left\{ A^{\dagger}DB^{\dagger} + Y - A^{\dagger}AYBB^{\dagger} | \; Y \in \mathbb{R}^{n \times p} \right\}. \tag{5}$$

The following results from [34] describe the conditions of convergence and the limit of the unknown matrix $V(t)$ from (3) as $t \to +\infty$.

Proposition 2 ([34])**.** *Suppose the matrices $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{p \times q}$ and $D \in \mathbb{R}^{m \times q}$ satisfy (4). Then, the unknown matrix $V(t)$ from (3) converges as $t \to +\infty$ with the equilibrium state*

$$V(t) \to \tilde{V} = A^{\dagger}DB^{\dagger} + V(0) - A^{\dagger}AV(0)BB^{\dagger} \tag{6}$$

for any initial state-variable matrix $V(0) \in \mathbb{R}^{n \times p}$.

The research in [40] investigated various ZNN models based on optimization methods. The goal of the current research is to develop a GNN model based on the gradient $E_G(t)$ of $||E(t)||_F^2$ instead of the original goal function $E(t)$.

The obtained results are summarized as follows:

- A novel error function $E_G(t)$ is proposed for the development of the GNN dynamical evolution.
- The GNN design based on the error function $E_G(t)$ is developed and analyzed theoretically and numerically.
- A hybridization of GNN and ZNN dynamical systems based on the error matrix E_G is proposed and investigated.

The overall organization of this paper is as follows. The motivation and derivation of the GGNN and GZNN models are presented in Section 2. Section 3 is dedicated to the convergence analysis of GGNN dynamics. A numerical comparison of GNN and GGNN dynamics is given in Section 4. Neural dynamics based on the hybridization of GGNN and GZNN models for solving matrix equations are considered in Section 6. Numerical examples of hybrid models are analyzed in Section 6. Finally, the last section presents some concluding remarks and a vision of further research.

2. Motivation and Derivation of GGNN and GZNN Models

The standard GNN design (2) solves the GLME $AXB = D$ under constraint (4). Our goal is to resolve this restriction and propose dynamic evolutions based on error functions that tend to zero without restrictions.

Our goal is to define the GNN design for solving the GLME $AXB = D$ based on the error function

$$E_G(t) := \nabla\varepsilon(t) = A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}} = A^{\mathrm{T}}E(t)B^{\mathrm{T}}. \tag{7}$$

According to known results from nonlinear unconstrained optimization [41], the equilibrium points of (7) satisfy

$$E_G(t) := \nabla\varepsilon(t) = 0.$$

We continue the investigation from [40]. More precisely, we develop the GNN model based on the error function $E_G(t)$ instead of the error function $E(t)$. In this way, new neural dynamics are aimed at forcing the gradient E_G to zero instead of the standard goal function $E(t)$. It is reasonable to call such an RNN model a gradient-based GNN (abbreviated GGNN).

Proposition 3 gives the conditions for the solvability of the matrix equations $E(t) = 0$ and $E_G(t) = 0$ and the general solutions to these systems.

Proposition 3 ([40])**.** *Consider the arbitrary matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{k\times h}$ and $D \in \mathbb{R}^{m\times h}$. The following statements are true:*

(a) *The equation $E(t) = 0$ is solvable if and only if* (4) *is satisfied, and the general solution to $E(t) = 0$ is given by* (5).

(b) *The equation $E_G(t) = 0$ is always solvable, and its general solution coincides with* (5).

Proof. (a) This part of the proof follows from known results on the solvability and general solution of the matrix equation $AXB = D$ of generalized inverses [4] (p. 52, Theorem 1) and its application to the matrix equation $E(t) = 0 \Longleftrightarrow AV(t)B = D$.

(b) According to [4] (p. 52, Theorem 1), the matrix equation

$$E_G(t) = 0 \Longleftrightarrow A^{\mathrm{T}}AVBB^{\mathrm{T}} = A^{\mathrm{T}}DB^{\mathrm{T}}$$

is consistent if and only if

$$A^{\mathrm{T}}A\left(A^{\mathrm{T}}A\right)^{\dagger}A^{\mathrm{T}}DB^{\mathrm{T}}\left(BB^{\mathrm{T}}\right)^{\dagger}BB^{\mathrm{T}} = A^{\mathrm{T}}DB^{\mathrm{T}}$$

is satisfied. Indeed, applying the properties $(A^{\mathrm{T}}A)^{\dagger}A^{\mathrm{T}} = A^{\dagger}$, $B^{T}(BB^{\mathrm{T}})^{\dagger} = B^{\dagger}$ and $A^{\mathrm{T}}AA^{\dagger} = A^{\mathrm{T}}$, $B^{\dagger}BB^{\mathrm{T}} = B^{\mathrm{T}}$ of the Moore–Penrose inverse [5] results in

$$A^{\mathrm{T}}A\left(A^{\mathrm{T}}A\right)^{\dagger}A^{\mathrm{T}}DB^{\mathrm{T}}\left(BB^{\mathrm{T}}\right)^{\dagger}BB^{\mathrm{T}} = A^{\mathrm{T}}AA^{\dagger}DB^{\dagger}BB^{\mathrm{T}} = A^{\mathrm{T}}DB^{\mathrm{T}}.$$

In addition, based on [4] (p. 52, Theorem 1), the general solution $V(t)$ to $E_G(t) = 0$ is

$$\begin{aligned} V &= \left(A^{\mathrm{T}}A\right)^{\dagger}A^{\mathrm{T}}DB^{\mathrm{T}}\left(BB^{\mathrm{T}}\right)^{\dagger} + Y - \left(A^{\mathrm{T}}A\right)^{\dagger}A^{\mathrm{T}}AYBB^{\mathrm{T}}\left(BB^{\mathrm{T}}\right)^{\dagger} \\ &= A^{\dagger}DB^{\dagger} + Y - A^{\dagger}AYBB^{\dagger}, \end{aligned} \tag{8}$$

which coincides with (5). □

In this way, the matrix equation $E(t) = 0$ is solvable under condition (4), while the equation $E_G(t) = 0$ is always consistent. In addition, the general solutions to equations $E(t) = 0$ and $E_G(t) = 0$ are identical [40].

The next step is to define the GGNN dynamics using the error matrix $E_G(t)$. Let us define the objective function $\varepsilon_G = ||E_G||_F^2/2$, whose gradient is equal to

$$\frac{\partial \varepsilon_G(V(t))}{\partial V} = \frac{\partial ||A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}}||_F^2}{\partial V} = -A^{\mathrm{T}}A\Big(A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}}\Big)BB^{\mathrm{T}}.$$

The dynamical system for the GGNN formula is obtained by applying the GNN evolution along the gradient of $\varepsilon_G(V(t))$ based on $E_G(t)$, as follows:

$$\begin{aligned}\dot{V}(t) &= -\gamma\frac{\partial \varepsilon_G}{\partial V}\\ &= \gamma A^{\mathrm{T}}A\Big(A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}}\Big)BB^{\mathrm{T}}.\end{aligned} \tag{9}$$

The nonlinear GGNN dynamics are defined as

$$\dot{V}(t) = \gamma A^{\mathrm{T}}A\mathcal{F}(A^{\mathrm{T}}(D - AV(t)B)B^{T})BB^{\mathrm{T}}, \tag{10}$$

in which $\mathcal{F}(C) = \mathcal{F}([c_{ij}])$ denotes the elementwise application of an odd and monotonically increasing function $f(\cdot)$, as mentioned in the previous section for the GNN model (3). Model (10) is termed GGNN(A, B, D). Three activation functions $f(\cdot)$ are used in numerical experiments:

1. Linear function
$$f_{lin}(x) = x; \tag{11}$$
2. Power-sigmoid activation function
$$f_{ps}(x, \rho, \varrho) = \begin{cases} x^{\rho} & \text{if} \quad |x| \geq 1 \\ \frac{1+e^{-\varrho}}{1-e^{-\varrho}} \cdot \frac{1+e^{-\varrho x}}{1-e^{-\varrho x}} & \text{if} \quad |x| < 1 \end{cases} \tag{12}$$
where $\varrho > 2$, and $\rho \geq 3$ is an odd integer;
3. Smooth power-sigmoid function
$$f_{sps}(x, \rho, \varrho) = \frac{1}{2}x^{\rho} + \frac{1+e^{-\varrho}}{1-e^{-\varrho}} \cdot \frac{1+e^{-\varrho x}}{1-e^{-\varrho x}}, \tag{13}$$
where $\varrho > 2$, and $\rho \geq 3$ is an odd integer.

Figure 1 represents the Simulink implementation of GGNN(A, B, D) dynamics (10).

On the other hand, the GZNN model, defined using the ZNN dynamics on the Zhangian matrix $E_G(t)$, is defined in [40] by the general evolutionary design

$$\dot{E}_G(t) = \frac{\mathrm{d}E_G(t)}{\mathrm{d}t} = -\gamma\mathcal{F}(E_G(t)). \tag{14}$$

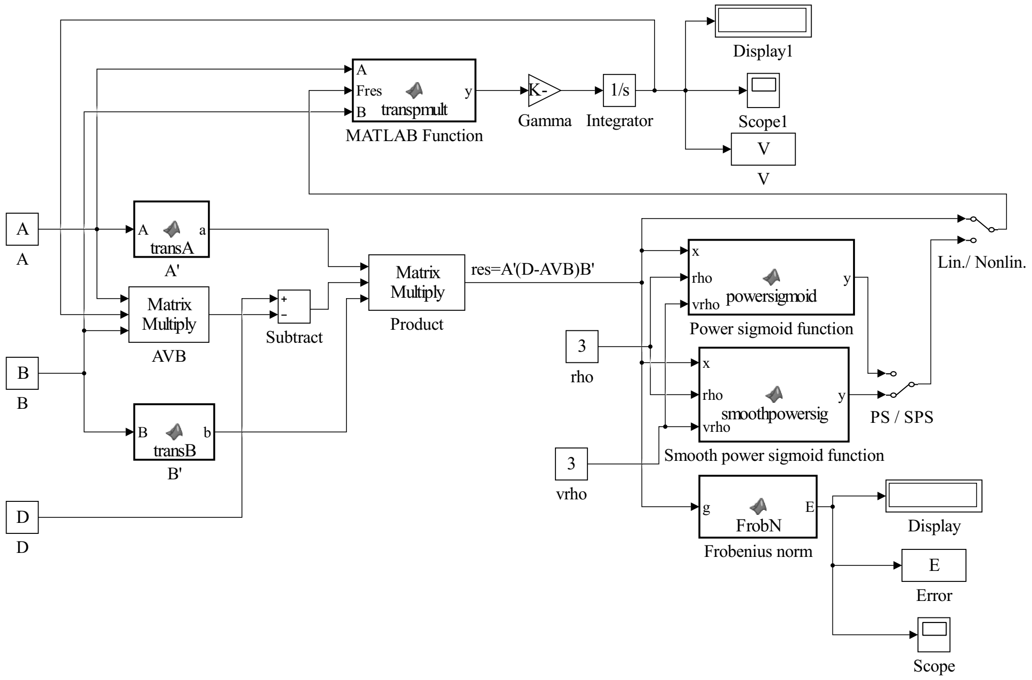


Figure 1. Simulink implementation of $\mathrm{GGNN}(A,B,D)$ evolution (10).

3. Convergence Analysis of GGNN Dynamics

In this section, we will analyze the convergence properties of the GGNN model given by dynamics (10).

Theorem 1. *Consider matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{p\times q}$ and $D \in \mathbb{R}^{m\times q}$. If an odd and monotonically increasing array activation function $\mathcal{F}(\cdot)$ based on an elementwise function $f(\cdot)$ is used, then the activation state matrix $V(t) \in \mathbb{R}^{n\times p}$ of the* $\mathrm{GGNN}(A,B,D)$ *model* (10) *asymptotically converges to the solution of the matrix equation $AXB = D$, i.e., $A^{\mathrm{T}}AV(t)BB^{\mathrm{T}} \to A^{\mathrm{T}}DB^{\mathrm{T}}$ as $t \to +\infty$, for an arbitrary initial state matrix $V(0)$.*

Proof. From statement (b) of Proposition 3, the solvability of $A^{\mathrm{T}}AVBB^{\mathrm{T}} = A^{\mathrm{T}}DB^{\mathrm{T}}$ is ensured. The substitution $V(t) = \bar{V}(t) + A^{\dagger}DB^{\dagger}$ transforms the dynamics (10) into

$$\begin{aligned}\frac{\mathrm{d}\bar{V}(t)}{\mathrm{d}t} = \frac{\mathrm{d}V(t)}{\mathrm{d}t} &= \gamma A^{\mathrm{T}}A\mathcal{F}\Big(A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}}\Big)BB^{\mathrm{T}}\\ &= \gamma A^{\mathrm{T}}A\mathcal{F}\Big(A^{\mathrm{T}}\Big(D - A\bar{V}(t)B - AA^{\dagger}DB^{\dagger}B\Big)B^{\mathrm{T}}\Big)BB^{\mathrm{T}}\\ &\overset{(4)}{=} \gamma A^{\mathrm{T}}A\mathcal{F}\Big(A^{\mathrm{T}}(D - A\bar{V}(t)B - D)B^{\mathrm{T}}\Big)BB^{\mathrm{T}}\\ &= -\gamma A^{\mathrm{T}}A\mathcal{F}\Big(A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}}\Big)BB^{\mathrm{T}}.\end{aligned} \tag{15}$$

The Lyapunov function candidate that measures the convergence performance is defined by

$$L(\bar{V}(t),t) = \frac{1}{2}||\bar{V}(t)||_F^2 = \frac{1}{2}\mathrm{Tr}\Big(\bar{V}(t)^{\mathrm{T}}\bar{V}(t)\Big). \tag{16}$$

The conclusion is $L(\bar{V}(t),t) \geq 0$. According to (16), assuming (15) and using $\mathrm{d}\,\mathrm{Tr}(X^{\mathrm{T}}X) = 2\mathrm{Tr}(X^{\mathrm{T}}\mathrm{d}X)$, in conjunction with the basic properties of the matrix trace function, one can express the time derivative of $L(\bar{V}(t),t)$ as follows:

$$\begin{aligned}\frac{\mathrm{d}L(\bar{V}(t),t)}{\mathrm{d}t} &= \frac{1}{2}\frac{\mathrm{dTr}\big(\bar{V}(t)^{\mathrm{T}}\bar{V}(t)\big)}{\mathrm{d}t} \\ &= \frac{1}{2}\cdot 2\cdot \mathrm{Tr}\left(\bar{V}(t)^{\mathrm{T}}\frac{\mathrm{d}\bar{V}(t)}{\mathrm{d}t}\right) \\ &= \mathrm{Tr}\Big[\bar{V}(t)^{\mathrm{T}}\Big(-\gamma A^{\mathrm{T}}A\mathcal{F}\Big(A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}}\Big)BB^{\mathrm{T}}\Big)\Big] \\ &= -\gamma\mathrm{Tr}\Big[\bar{V}(t)^{\mathrm{T}}A^{\mathrm{T}}A\mathcal{F}\Big(A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}}\Big)BB^{\mathrm{T}}\Big] \\ &= -\gamma\mathrm{Tr}\Big[BB^{\mathrm{T}}\bar{V}(t)^{\mathrm{T}}A^{\mathrm{T}}A\mathcal{F}\Big(A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}}\Big)\Big] \\ &= -\gamma\mathrm{Tr}\Big[\Big(A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}}\Big)^{\mathrm{T}}\mathcal{F}\Big(A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}}\Big)\Big].\end{aligned} \tag{17}$$

Since the scalar-valued function $f(\cdot)$ is odd and monotonically increasing, it follows that, for $W(t) = A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}}$,

$$\begin{aligned}\frac{\mathrm{d}L(\bar{V}(t),t)}{\mathrm{d}t} &= -\gamma\mathrm{Tr}\Big[(W^{\mathrm{T}}\mathcal{F}(W))\Big] \\ &= -\gamma\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}f(w_{ij})\begin{cases} <0 & \text{if} \quad W(t) := A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}} \neq 0 \\ =0 & \text{if} \quad W(t) := A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}} = 0,\end{cases}\end{aligned} \tag{18}$$

which implies

$$\frac{\mathrm{d}L(\bar{V}(t),t)}{\mathrm{d}t}\begin{cases} <0 & \text{if} \quad W(t) \neq 0 \\ =0 & \text{if} \quad W(t) = 0.\end{cases} \tag{19}$$

Observing the identity

$$\begin{aligned}W(t) &= A^{\mathrm{T}}A\bar{V}(t)BB^{\mathrm{T}} \\ &= A^{\mathrm{T}}A\Big(V(t) - A^{\dagger}DB^{\dagger}\Big)BB^{\mathrm{T}} \\ &= A^{\mathrm{T}}AV(t)BB^{\mathrm{T}} - A^{\mathrm{T}}DB^{\mathrm{T}} \\ &= A^{\mathrm{T}}(AV(t)B - D)B^{\mathrm{T}},\end{aligned}$$

and using the Lyapunov stability theory, $W(t) := A^{\mathrm{T}}(AV(t)B - D)B^{\mathrm{T}}$ globally converges to the zero matrix from an arbitrary initial value $V(0)$. □

Theorem 2. *The activation state-variable matrix $V(t)$ of the model* GGNN(A,B,D)*, defined by* (10)*, is convergent as $t \to +\infty$, and its equilibrium state is*

$$V(t) \to \tilde{V}(t) = A^{\dagger}DB^{\dagger} + V(0) - A^{\dagger}AV(0)BB^{\dagger} \tag{20}$$

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

Proof. From (10), the matrix $V_1(t) = (A^{\mathrm{T}}A)^{\dagger}A^{\mathrm{T}}AV(t)BB^{\mathrm{T}}(BB^{\mathrm{T}})^{\dagger}$ satisfies

$$\begin{aligned}\frac{\mathrm{d}V_1(t)}{\mathrm{d}t} &= (A^{\mathrm{T}}A)^{\dagger}A^{\mathrm{T}}A\frac{\mathrm{d}V(t)}{\mathrm{d}t}BB^{\mathrm{T}}(BB^{\mathrm{T}})^{\dagger} \\ &= \gamma(A^{\mathrm{T}}A)^{\dagger}A^{\mathrm{T}}A\Big[A^{\mathrm{T}}A\Big(A^{\mathrm{T}}(D - AV(t)B)B^{\mathrm{T}}\Big)BB^{\mathrm{T}}\Big]BB^{\mathrm{T}}(BB^{\mathrm{T}})^{\dagger}.\end{aligned}$$

According to the basic properties of the Moore–Penrose inverse [5], it follows that

$$(BB^{\mathrm{T}})^{\mathrm{T}}BB^{\mathrm{T}}(BB^{\mathrm{T}})^{\dagger} = (BB^{\mathrm{T}})^{\mathrm{T}} = BB^{\mathrm{T}}, \qquad (A^{\mathrm{T}}A)^{\dagger}A^{\mathrm{T}}A(A^{\mathrm{T}}A)^{\mathrm{T}} = (A^{\mathrm{T}}A)^{\mathrm{T}} = A^{\mathrm{T}}A$$

which further implies

$$\begin{aligned}\frac{\mathrm{d}V_1(t)}{\mathrm{d}t} &= \gamma A^{\mathrm{T}}A\Big(A^{\mathrm{T}}(D-AV(t)B)B^{\mathrm{T}}\Big)BB^{\mathrm{T}}\\ &= \frac{\mathrm{d}V(t)}{\mathrm{d}t}.\end{aligned}$$

Consequently, $V_2(t) = V(t) - V_1(t)$ satisfies $\frac{\mathrm{d}V_2(t)}{\mathrm{d}t} = \frac{\mathrm{d}V(t)}{\mathrm{d}t} - \frac{\mathrm{d}V_1(t)}{\mathrm{d}t} = 0$, which implies

$$\begin{aligned}V_2(t) &= V_2(0)\\ &= V(0) - V_1(0)\\ &= V(0) - (A^TA)^{\dagger}A^{\mathrm{T}}AV(0)BB^{\mathrm{T}}(BB^{\mathrm{T}})^{\dagger}\\ &= V(0) - A^{\dagger}AV(0)BB^{\dagger},\ \ t \geq 0.\end{aligned} \tag{21}$$

Furthermore, from Theorem 1, $A^{\mathrm{T}}AV(t)BB^{\mathrm{T}} \to A^{\mathrm{T}}DB^{\mathrm{T}}$, and $V_1(t)$ converges to

$$\begin{aligned}V_1(t) = (A^{\mathrm{T}}A)^{\dagger}A^{\mathrm{T}}AV(t)BB^{\mathrm{T}}(BB^{\mathrm{T}})^{\dagger} &\to (A^{\mathrm{T}}A)^{\dagger}A^{\mathrm{T}}DB^{\mathrm{T}}(BB^{\mathrm{T}})^{\dagger}\\ = A^{\dagger}DB^{\dagger}\end{aligned}$$

as $t \to +\infty$. Therefore, $V(t) = V_1(t) + V_2(t)$ converges to the equilibrium state

$$\tilde{V}(t) = A^{\dagger}DB^{\dagger} + V_2(t) = A^{\dagger}DB^{\dagger} + V(0) - A^{\dagger}AV(0)BB^{\dagger}.$$

The proof is finished. □

4. Numerical Experiments on GNN and GGNN Dynamics

The numerical examples in this section are based on the Simulink implementation of the GGNN formula in Figure 1.

The parameter γ, initial state $V(0)$ and parameters ρ and ϱ of the nonlinear activation functions (12) and (13) are entered directly into the model, while matrices A, B and D are defined from the workspace. It is assumed that $\rho = \varrho = 3$ in all examples. The `ode15s` differential equation solver is used in the configuration parameters. In all examples, V^* denotes the theoretical solution.

The blocks *powersig*, *smoothpowersig* and *transpmult* include the codes described in [34,42].

Example 1. *Let us consider the idempotent matrix A from* [43,44],

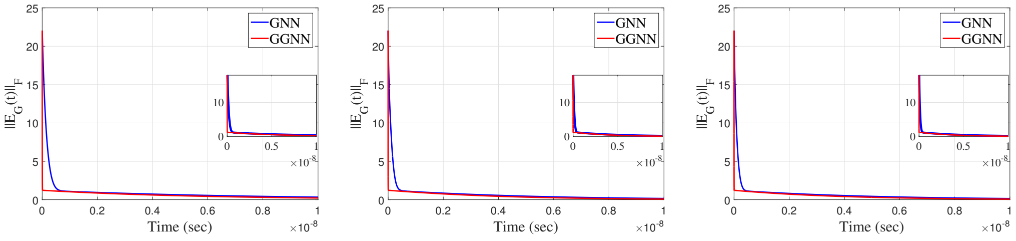
$$A = \begin{bmatrix}1 & 0 & 1 & 1\\ 0 & 1 & 1 & 2\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$$

of $\mathrm{rank}(A) = 2$*, and the theoretical Moore–Penrose inverse*

$$V^* = A^{\dagger} = \frac{1}{3}\begin{bmatrix}2 & -1 & 0 & 0\\ -1 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}.$$

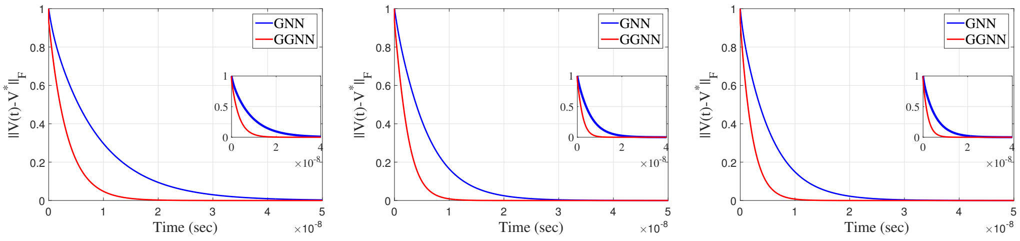
The matrix equation corresponding to the Moore–Penrose inverse is $A^{\mathrm{T}}AX = A^{\mathrm{T}}$ [16], which implies the error function $E(t) = A^{\mathrm{T}}(I - AX)$. The corresponding GNN model is defined by GNN$(A^{\mathrm{T}}A, I_4, A^{\mathrm{T}})$, where I_4 denotes the identity and zero 4×4 matrix. Constraint (4) reduces to the condition $AA^{\dagger}A^{\mathrm{T}} = A^{\mathrm{T}}$, which is not satisfied. The input parameters of GNN$(A^{\mathrm{T}}A, I_4, A^{\mathrm{T}})$ are $\gamma = 10^8$, $V(0) = O_4$, where O_4 denotes the zero 4×4 matrix. The corresponding GGNN$((A^{\mathrm{T}}A)^2$, $I, A^{\mathrm{T}}AA^{\mathrm{T}})$ design is based on the error matrix $E_G(t) = A^{\mathrm{T}}AA^{\mathrm{T}}(I - AV)$. The Simulink imple-

mentation of GGNN(A, B, D) from Figure 1 and the Simulink implementation of GNN(A, B, D) from [34] export, in this case, the graphical results presented in Figures 2 and 3, which display the behaviors of the norms $||E_G(t)||_F = ||A^{\mathrm{T}}AA^{\mathrm{T}}(I - AV(t))||_F$ and $||V(t) - V^*||_F$, respectively. It is observable that the norms generated by the application of the GGNN formula vanish faster to zero than the corresponding norms in the GNN model. The graphs in the presented figures strengthen the fast convergence of the GGNN dynamical system and its important role, which can include the application of this specific model (10) to problems that require the computation of the Moore–Penrose inverse.



(**a**) (**b**) (**c**)

Figure 2. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E_G(t)\|_F$ in GGNN$((A^{\mathrm{T}}A)^2, I, A^{\mathrm{T}}AA^{\mathrm{T}})$ compared to GNN$(A^{\mathrm{T}}A, I_4, A^{\mathrm{T}})$ in Example 1.



(**a**) (**b**) (**c**)

Figure 3. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|V(t) - V^*\|_F$ in GGNN$((A^{\mathrm{T}}A)^2, I, A^{\mathrm{T}}AA^{\mathrm{T}})$ compared to GNN$(A^{\mathrm{T}}A, I_4, A^{\mathrm{T}})$ in Example 1.

Example 2. *Let us consider the matrices*

$$A = \begin{bmatrix} -8 & 8 & -4 \\ 11 & 4 & -7 \\ 1 & -4 & 3 \\ 0 & 12 & -10 \\ 6 & 12 & -12 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} -84 & 2524 & 304 \\ -2252 & -623 & 2897 \\ 484 & -885 & -701 \\ -1894 & 2278 & 2652 \\ -2778 & 1524 & 3750 \end{bmatrix}.$$

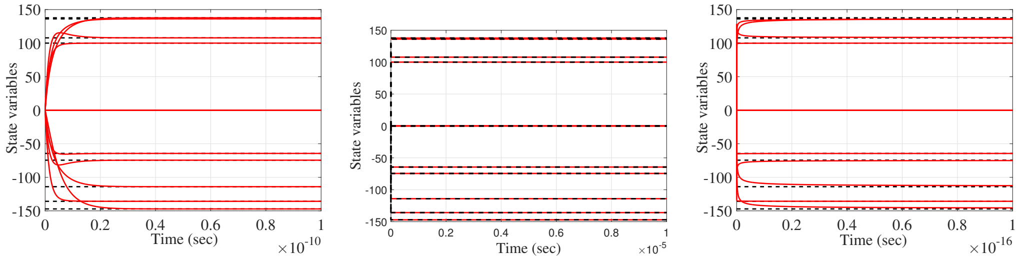
The exact minimum-norm least-squares solution is

$$V^* = A^{\dagger}DB^{\dagger} = \begin{bmatrix} -\frac{7409}{65} & -\frac{9564}{65} & \frac{8953}{65} & 0 \\ -\frac{968}{13} & \frac{1770}{13} & \frac{1402}{13} & 0 \\ \frac{6503}{65} & -\frac{4187}{65} & -\frac{8826}{65} & 0 \end{bmatrix}.$$

The ranks of the input matrices are equal to $r = \mathrm{rank}(A) = 2$, $\mathrm{rank}(D) = 2$ and $\mathrm{rank}(B) = 3$. Constraint (4) is satisfied in this case. The linear GGNN(A, B, D) formula (10) is applied to solve the matrix equation $AXB = D$. The gain parameter of the model is $\gamma = 10^9$, $V(0) = 0$, and the stopping time is $t = 0.00001$, which gives

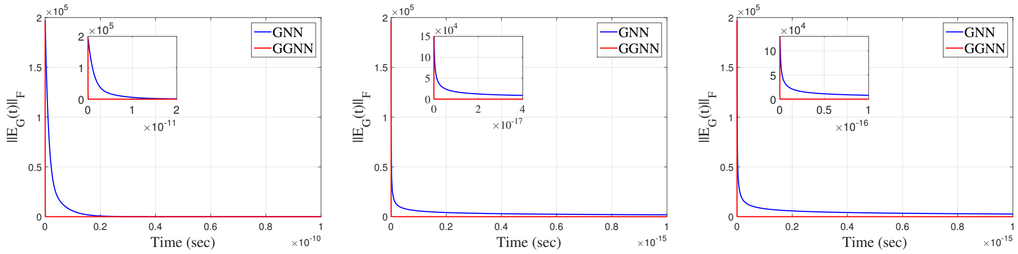
$$X = \begin{bmatrix} -113.9846 & -147.1385 & 137.7385 & 0 \\ -74.4615 & 136.1538 & 107.8462 & 0 \\ 100.0462 & -64.4154 & -135.7846 & 0 \end{bmatrix} \approx A^{\dagger}DB^{\dagger}.$$

The elementwise trajectories of the state variables v_{ij} of the state matrix $V(t)$ are shown in Figure 4a–c with solid red lines for linear, power-sigmoid and smooth power-sigmoid activation functions, respectively. The fast convergence of elementwise trajectories to the corresponding black dashed trajectories of the theoretical solution V^* is notable. In addition, faster convergence caused by the nonlinear AFs f_{ps} and f_{sps} is noticeable in Figure 4b,c. The trajectories in the figures indicate the usual convergence behavior, so the system is globally asymptotically stable. The norms of the error matrix E_G of both models GNN and GGNN under linear and nonlinear AFs are shown in Figure 5a–c. The power-sigmoid and smooth power-sigmoid activation functions show superiority in their convergence speed compared with linear activation. On each graph in Figure 5a–c, the Frobenius norm $\|E_G(t)\|_F$ of the error matrix $E_G(t)$ in the GGNN formula vanishes faster to zero than that in the GNN model. Moreover, in each graph in Figure 6a–c, the Frobenius norm $\|E(t)\|_F$ in the GGNN formula vanishes faster to zero than that in the GNN model, which strengthens the fact that the proposed dynamical system (10) initiates accelerated convergence compared to (3).



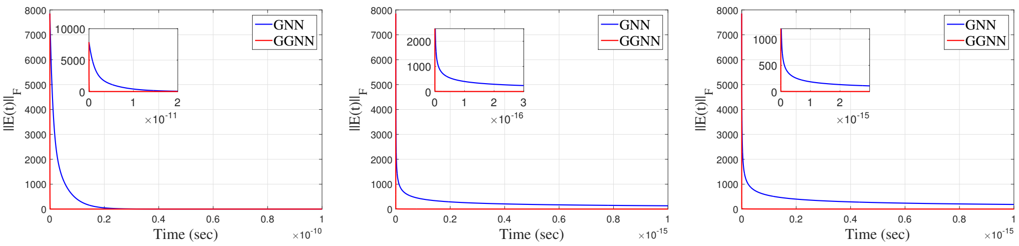
(**a**) (**b**) (**c**)

Figure 4. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. Elementwise convergence trajectories $v_{ij} \in V(t)$ of the GGNN(A, B, D) network in Example 2.



(**a**) (**b**) (**c**)

Figure 5. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E_G(t)\|_F$ in GGNN(A, B, D) compared to GNN(A, B, D) in Example 2.



(**a**) (**b**) (**c**)

Figure 6. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E(t)\|_F$ in GGNN(A, B, D) compared to GNN(A, B, D) in Example 2.

All graphs shown in Figures 5 and 6 confirm the applicability of the proposed GGNN design compared to the traditional GNN design, even if constraint (4) holds.

Example 3. *Let us explore the behavior of GNN and GGNN dynamics for computing the Moore–Penrose inverse of the matrix*

$$A = \begin{bmatrix} 9 & 3 & -3 \\ -1 & 1 & 0 \\ 4 & 7 & 2 \\ 2 & 4 & -4 \\ 13 & 5 & 8 \end{bmatrix}.$$

The Moore–Penrose inverse of A is equal to

$$A^{\dagger} = \begin{bmatrix} \frac{9908}{127779} & -\frac{18037}{766674} & -\frac{6874}{127779} & -\frac{2663}{383337} & \frac{29941}{766674} \\ -\frac{5690}{127779} & \frac{14426}{383337} & \frac{16741}{127779} & \frac{25130}{383337} & -\frac{6392}{383337} \\ -\frac{3517}{42593} & \frac{1979}{255558} & \frac{1073}{42593} & -\frac{7373}{127779} & \frac{15049}{255558} \end{bmatrix}$$
$$\approx \begin{bmatrix} 0.0775 & -0.0235 & -0.0538 & -0.0069 & 0.0390 \\ -0.0445 & 0.0376 & 0.1310 & 0.0655 & -0.0167 \\ -0.0826 & 0.0077 & 0.0252 & -0.0577 & 0.0589 \end{bmatrix}.$$

The rank of the input matrix is equal to $r = \operatorname{rank}(A) = 3$. Consequently, the matrix A is left invertible and satisfies $A^{\dagger}A = I$. The error matrix $E(t) = I - VA$ initiates the GNN(I, A, I) dynamics for computing $A^{\dagger}$. The gradient-based error matrix

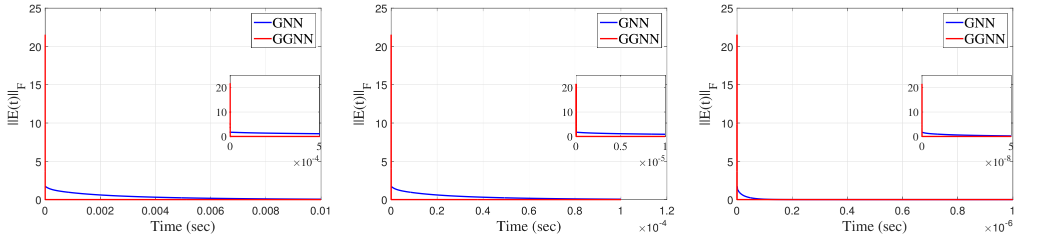
$$E_G(t) = (I - V(t)A)A^{\mathrm{T}}.$$

initiates the GGNN$(I, AA^{\mathrm{T}}, A^{\mathrm{T}})$ design.

The gain parameter of the model is $\gamma = 100$, and the initial state is $V(0) = 0$ with a stop time $t = 0.00001$.

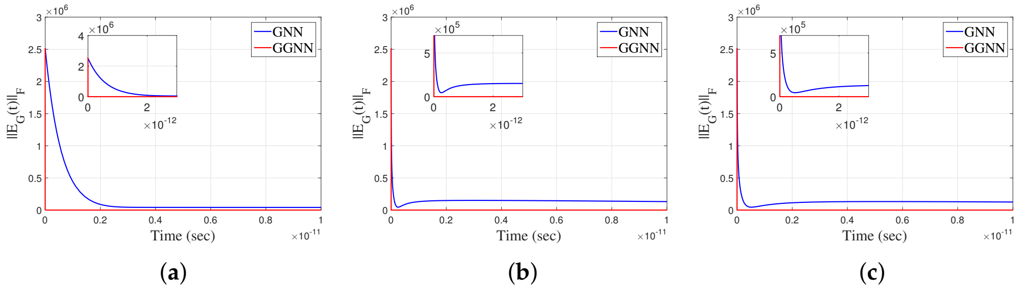
The Frobenius norms of the error matrix $E(t)$ generated by the linear GNN and GGNN models for different values of γ ($\gamma = 10^2, \gamma = 10^3, \gamma = 10^6$) are shown in Figure 7a–c. The graphs in these figures confirm an increase in the convergence speed, which is caused by the increase in the gain parameter γ. Because of that, the considered time intervals are $[0, 10^{-2}]$, $[0, 10^{-3}]$ and $[0, 10^{-6}]$, respectively. In all three scenarios, a faster convergence of the GGNN model is observable compared to the GNN design. The values of the norm $\|E_G\|_F$ generated by both the GNN and GGNN models with linear and two nonlinear activation functions are shown in Figure 8a–c. Like the conclusion in the previous example, the perception is that the GGNN converges faster compared to the GNN model.

In addition, the graphs in Figure 8b,c, corresponding to the power-sigmoid and smooth power-sigmoid AFs, respectively, show a certain level of instability in convergence, as well as an increase in the value of $\|E_G(t)\|_F$.



(**a**) (**b**) (**c**)

Figure 7. (**a**) $\gamma = 10, t \in [0, 10^{-2}]$. (**b**) $\gamma = 10^3, t \in [0, 10^{-3}]$. (**c**) $\gamma = 10^6, t \in [0, 10^{-6}]$. $\|E(t)\|_F$ for different γ in GGNN$(I, AA^{\mathrm{T}}, A^{\mathrm{T}})$ compared to GNN(I, A, I) in Example 3.



(**a**) (**b**) (**c**)

Figure 8. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E_G(t)\|_F$ in GGNN$(I, AA^{\mathrm{T}}, A^{\mathrm{T}})$ compared to GNN(I, A, I) in Example 3.

Example 4. *Consider the matrices*

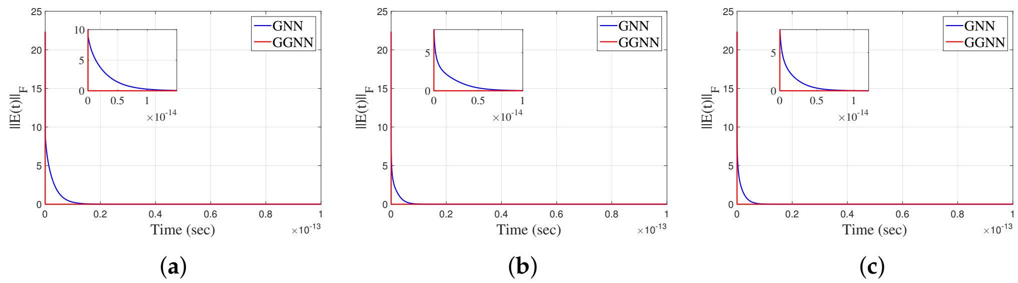
$$A = \begin{bmatrix} 15 & -352 & -45 & -238 & 42 \\ -5 & 14 & 8 & 132 & -65 \\ 235 & -65 & 44 & 350 & -73 \end{bmatrix}, \ D = \begin{bmatrix} -4 & 4 & 16 \\ 3 & 1 & -9 \\ 1 & -7 & 2 \\ 2 & 2 & -4 \\ 4 & 1 & -5 \end{bmatrix}, \ A_1 = DA,$$

which dissatisfy $\mathrm{rank}(A_1) = \mathrm{rank}(D) = 3$*. Now, we apply the GNN and GGNN formulae to solve the matrix equation* $A_1X = D$*. The standard error function is defined as* $E(t) = D - A_1V(t)$*. So, we consider GNN*(A_1, I_3, D)*. The error matrix for the corresponding GGNN model is* $E_G(t) = A_1^{\mathrm{T}}(D - A_1V(t))$*, which initiates the GGNN*$(A_1^{\mathrm{T}}A_1, I_3, A_1^{\mathrm{T}}D)$ *flow. The gain parameter of the model is* $\gamma = 10^9$*, and the final time is* $t = 0.00001$*. The zero initial state* $V(0) = 0$ *generates the best approximate solution* $X = A_1^{\dagger}D = (DA)^{\dagger}D$ *of the matrix equation* $A_1X = D$*, given by*

$$X = A_1^{\dagger}D = \begin{bmatrix} -\frac{133851170015}{180355524917879} & -\frac{1648342203725}{180355524917879} & \frac{608888775010}{180355524917879} \\ -\frac{508349079720}{180355524917879} & -\frac{691967699675}{180355524917879} & -\frac{48398092277}{180355524917879} \\ -\frac{68130232042}{180355524917879} & -\frac{242513061343}{180355524917879} & \frac{82710890618}{180355524917879} \\ -\frac{31936168532}{180355524917879} & \frac{727110260384}{180355524917879} & \frac{134047117682}{180355524917879} \\ -\frac{172434574901}{180355524917879} & -\frac{1350198643304}{180355524917879} & \frac{225136761416}{180355524917879} \end{bmatrix}$$

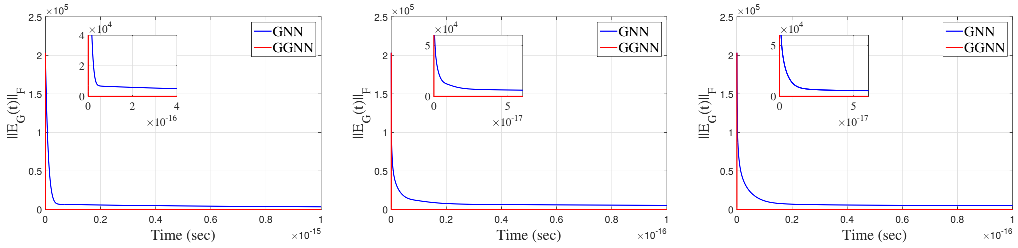
$$\approx \begin{bmatrix} -0.000742 & -0.00914 & 0.00338 \\ -0.00282 & -0.00384 & -0.000268 \\ -0.000378 & -0.00134 & 0.000459 \\ -0.000177 & 0.00403 & 0.000743 \\ -0.000956 & -0.00749 & 0.00125 \end{bmatrix}.$$

The Frobenius norms of the error matrix $E(t) = D - A_1V(t)B$ *in the GNN and GGNN models for both linear and nonlinear activation functions are shown in Figure 9a–c, and the error matrix* $E_G(t) = A_1^{\mathrm{T}}(D - A_1V(t))$ *in both models for linear and nonlinear activation functions are shown in Figure 10a–c. It is observable that the GGNN converges faster than GNN.*



(**a**) (**b**) (**c**)

Figure 9. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E(t)\|_F$ in GGNN$(A_1^{\mathrm{T}}A_1, I_3, A_1^{\mathrm{T}}D)$ compared to GNN(A_1, I_3, D) in Example 4.



(**a**) (**b**) (**c**)

Figure 10. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E_G(t)\|_F$ in GGNN$(A_1^{\mathrm{T}}A_1, I_3, A_1^{\mathrm{T}}D)$ compared to GNN(A_1, I_3, D) in Example 4.

Example 5. *Tables 1 and 2 show the results obtained during experiments we conducted with nonsquared matrices, where $m \times n$ is the dimension of the matrix. Table 1 lists the input data that were used to perform experiments with the Simulink model and generated the results in Table 2. The best cases in Table 2 are marked in bold text.*

Table 1. Input data.

Matrix A			**Matrix B**			**Matrix D**			**Input and Residual Norm**		
m	n	rank(A)	p	q	rank(B)	m	q	rank(D)	γ	t_f	$\|AA^{\dagger}DB^{\dagger}B-D\|_F$
10	8	8	9	7	7	10	7	7	10^4	0.5	1.051
10	8	6	9	7	7	10	7	7	10^4	0.5	1.318
10	8	6	9	7	5	10	7	7	10^4	0.5	1.81
10	8	6	9	7	5	10	7	5	10^4	5	2.048
10	8	1	9	7	2	10	7	1	10^4	5	2.372
20	10	10	8	5	5	20	5	5	10^6	5	1.984
20	10	5	8	5	5	20	5	5	10^6	5	2.455
20	10	5	8	5	2	20	5	5	10^6	1	3.769
20	10	2	8	5	2	20	5	2	10^6	1	2.71
20	15	15	5	2	2	20	2	2	10^8	1	1.1
20	15	10	5	2	2	20	2	2	10^8	1	1.158
20	15	10	5	2	1	20	2	2	10^8	1	2.211
20	15	5	5	2	1	20	2	2	10^8	1	1.726

Table 2. Experimental results based on data presented in Table 1.

$\|\|E(t)\|\|_F$**(GNN)**	$\|\|E(t)\|\|_F$**(GGNN)**	$\|\|E_G(t)\|\|_F$**(GNN)**	$\|\|E_G(t)\|\|_F$**(GGNN)**	**CPU(GNN)**	**CPU(GGNN)**
1.051	1.094	$\mathbf{2.52 \times 10^{-9}}$	0.02524	**5.017148**	13.470995
1.318	1.393	$\mathbf{3.122 \times 10^{-7}}$	0.03661	22.753954	**10.734163**
1.811	1.899	**0.0008711**	0.03947	15.754537	**15.547785**
2.048	2.082	$\mathbf{1.96 \times 10^{-10}}$	0.00964	**9.435709**	17.137916
2.372	**2.3722**	$\mathbf{1.7422 \times 10^{-15}}$	2.003×10^{-15}	21.645386	**13.255210**
1.984	**1.984**	2.288×10^{-14}	$\mathbf{9.978 \times 10^{-15}}$	21.645386	**13.255210**
2.455	**2.455**	1.657×10^{-11}	$\mathbf{1.693 \times 10^{-14}}$	50.846893	**19.059385**
3.769	**3.769**	6.991×10^{-11}	$\mathbf{4.071 \times 10^{-14}}$	42.184748	**13.722390**
2.71	**2.71**	1.429×10^{-14}	$\mathbf{1.176 \times 10^{-14}}$	148.484258	**13.527065**
1.1	**1.1**	1.766×10^{-13}	$\mathbf{5.949 \times 10^{-15}}$	218.169376	**17.5666568**
1.158	**1.158**	2.747×10^{-10}	$\mathbf{2.981 \times 10^{-13}}$	45.505618	**12.441782**
2.211	**2.211**	7.942×10^{-12}	$\mathbf{8.963 \times 10^{-14}}$	194.605133	**14.117241**
1.726	**1.726**	8.042×10^{-15}	$\mathbf{3.207 \times 10^{-15}}$	22.340501	**11.650829**

The numerical results arranged in Table 2 are divided into two parts by a horizontal line. The upper part corresponds to the test matrices of dimensions ≤ 10, while the lower part corresponds to the dimensions $m, n \geq 10$. Considering the first two columns, it is observable from the upper part that the GGNN generates smaller values $||E(t)||_F$ compared to the GGNN. The values of $||E(t)||_F$ in the lower part generated by the GNN and GGNN are equal. Considering the third and fourth columns, it is observable from the upper part that the GGNN generates smaller values $||E_G(t)||_F$ compared to the GGNN. On the other hand, the values of $||E_G(t)||_F$ in the lower part, generated by the GGNN, are smaller than the corresponding values generated by the GNN. The last two columns show that the GGNN requires less CPU time compared to the GNN. The general conclusion is that the GGNN model is more efficient in rank-deficient test matrices of larger order $m, n \geq 10$.

5. Mixed GGNN-GZNN Model for Solving Matrix Equations

The gradient-based error matrix for solving the matrix equation $AX = B$ is defined by

$$E_{G_{A,I,B}}(t) = A^{\mathrm{T}}(AV(t) - B).$$

The GZNN design (14) corresponding to the error matrix $E_{A,I,B}$, designated $\mathrm{GZNN}(A, I, B)$, is of the form:

$$\dot{E}_{G_{A,I,B}}(t) = -\gamma \mathcal{F}\Big(A^{\mathrm{T}}(AV(t) - B)\Big). \tag{22}$$

Now, the scalar-valued norm-based error function corresponding to $E_{G_{A,I,B}}(t)$ is given by

$$\varepsilon(t) = \varepsilon(V(t)) = \frac{1}{2}||E_{G_{A,I,B}}(t)||_F = \frac{||A^{\mathrm{T}}(AV(t) - B)||_F}{2}.$$

The following dynamic state equation can be derived using the $\mathrm{GGNN}(A, I, B)$ design formula based on (10):

$$\dot{V}(t) = -\gamma A^{\mathrm{T}} A \mathcal{F}\Big(A^{\mathrm{T}}(AV(t) - B)\Big). \tag{23}$$

Further, using a combination of $\dot{E}_{G_{A,I,B}}(t) = A^{\mathrm{T}} A\dot{V}(t)$ and the GNN dynamics (23), it follows that

$$\dot{E}_{G_{A,I,B}}(t) = A^{\mathrm{T}} A\dot{V}(t) = -\gamma A^{\mathrm{T}} A A^{\mathrm{T}} A \mathcal{F}\Big(A^{\mathrm{T}}(AV(t) - B)\Big). \tag{24}$$

The next step is to define the new hybrid model based on the summation of the right-hand sides in (22) and (24), as follows:

$$\dot{E}_{G_{A,I,B}}(t) = -\gamma\Big(\Big(A^{\mathrm{T}} A\Big)^2 + I\Big)\mathcal{F}\Big(A^{\mathrm{T}}(AV(t) - B)\Big). \tag{25}$$

The model (25) is derived from the combination of the model $\mathrm{GGNN}(A, I, B)$ and the model $\mathrm{GZNN}(A, I, B)$. Hence, it is equally justified to use the term Hybrid GGNN (abbreviated HGGNN) and Hybrid GZNN (abbreviated HGZNN) model. But model (25) is implicit, so it is not a type of GGNN dynamics. On the other hand, it is designed for time-invariant matrices, which is not in accordance with the common nature of GZNN models, because usually, the GZNN is used in the time-varying case. A formal comparison of (25) and $\mathrm{GZNN}(A, I, B)$ reveals that both these methods possess identical left-hand sides, and the right-hand side of (25) can be derived by multiplying the right-hand side of $\mathrm{GZNN}(A, I, B)$ by the term $\big(A^{\mathrm{T}} A\big)^2 + I$.

Formally, (25) is closer to GZNN dynamics, so we will denote the model (25) by $\mathrm{HGZNN}(A, I, B)$, considering that this model is not the exact GZNN neural dynamics and is applicable to time-invariant case. This is the case of the constant coefficient matrices A, I and B. Figure 11 represents the Simulink implementation of $\mathrm{HGZNN}(A, I, B)$ dynamics (25).

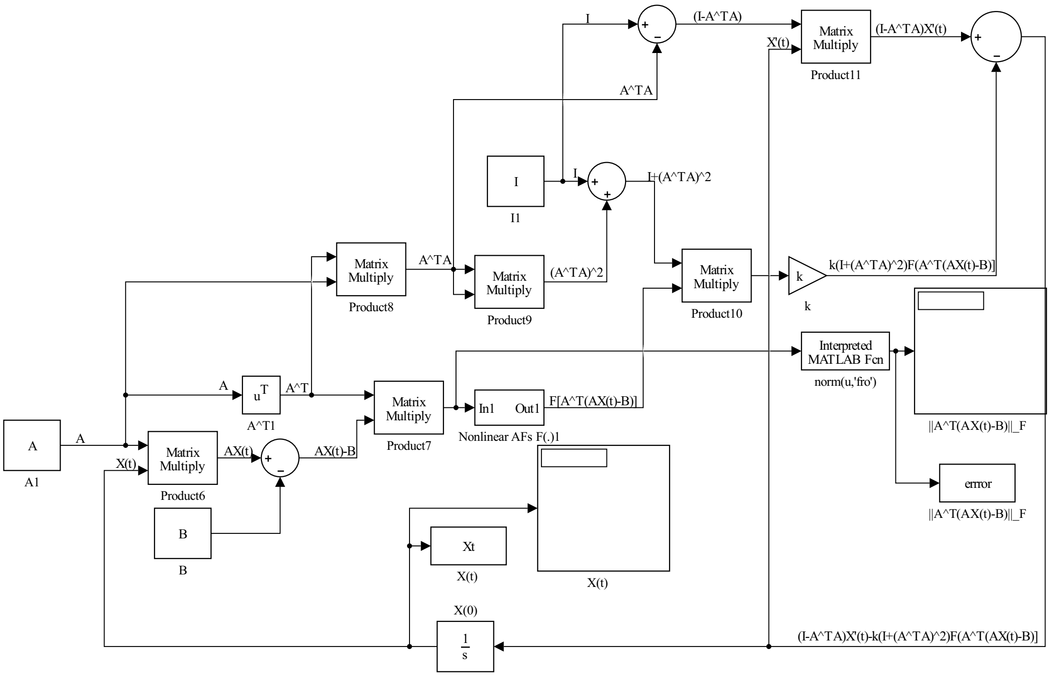


Figure 11. Simulink implementation of (25).

Now, we will take into account the process of solving the matrix equation $XC = D$. The error matrix for this equation is defined by

$$E_{G_{I,C,D}}(t) = (V(t)C - D)C^{\mathrm{T}}.$$

The GZNN design (14) corresponding to the error matrix $E_{I,C,D}$, denoted by $\mathrm{GZNN}(I, C, D)$, is of the form:

$$\dot{E}_{G_{I,C,D}}(t) = \dot{V}CC^{\mathrm{T}} = -\gamma\mathcal{F}\Big((V(t)C - D)C^{\mathrm{T}}\Big). \tag{26}$$

On the other hand, the GGNN design formula (10) produces the following dynamic state equation:

$$\dot{V}(t) = -\gamma\mathcal{F}\Big((V(t)C - D)C^{\mathrm{T}}\Big)CC^{\mathrm{T}}, \;\; V(0) = V_0. \tag{27}$$

The GGNN model (27) is denoted by $\mathrm{GGNN}(I, C, D)$. It implies

$$\dot{E}_{G_{I,C,D}}(t) = \dot{V}(t)CC^{\mathrm{T}} = -\gamma\mathcal{F}\Big((V(t)C - D)C^{\mathrm{T}}\Big)CC^{\mathrm{T}}CC^{\mathrm{T}}. \tag{28}$$

A new hybrid model based on the summation of the right-hand sides in (26) and (28) can be proposed as follows:

$$\dot{E}_{G_{I,C,D}}(t) = -\gamma\mathcal{F}\Big((V(t)C - D)C^{\mathrm{T}}\Big)\Big(I + \big(CC^{\mathrm{T}}\big)^2\Big). \tag{29}$$

The Model (29) will be denoted by $\mathrm{HGZNN}(I, C, D)$. This is the case with the constant coefficient matrices I, C and D.

For the purposes of the proof of the following results, we will use $ECR(\mathcal{M})$ to denote the exponential convergence rate of the model $\mathcal{M}$. With $\lambda_{\min}(K)$ and $\lambda_{\max}(K)$, we denote the smallest and largest eigenvalues of the matrix K, respectively. Continuing the previous work, we use three types of activation functions $\mathcal{F}$: linear, power-sigmoid and smooth power-sigmoid.

The following theorem determines the equilibrium state of $\mathrm{HGZNN}(A, I, B)$ and defines its global exponential convergence.

Theorem 3. *Let $A \in \mathbb{R}^{k\times n}, B \in \mathbb{R}^{k\times m}$ be given and satisfy $AA^{\dagger}B = B$, and let $V(t) \in \mathbb{R}^{n\times m}$ be the state matrix of* (25)*, where $\mathcal{F}$ is defined by f_{lin}, f_{ps} or f_{sps}.*

(a) *Then, $V(t)$ achieves global convergence and satisfies $AV(t) \to B$ when $t \to +\infty$, starting from any initial state $X(0) \in \mathbb{R}^{n\times m}$. The state matrix $V(t) \in \mathbb{R}^{n\times m}$ of* HGZNN(A, I, B) *is stable in the sense of Lyapunov.*

(b) *The exponential convergence rate of the* HGZNN(A, I, B) *model* (25) *in the linear case is equal to*

$$ECR(\mathrm{HGZNN}(A,I,B)) = \gamma\left(1+\sigma^4_{\min}(A)\right), \tag{30}$$

where $\sigma_{\min}(A) = \lambda_{\min}(A^{\mathrm{T}}A)$ is the minimum singular value of A.

(c) *The activation state variable matrix $V(t)$ of the model* HGZNN(A, I, B) *is convergent when $t \to +\infty$ with the equilibrium state matrix*

$$V(t) \to \tilde{V}_{V(0)} = A^{\dagger}B + (I - A^{\dagger}A)V(0). \tag{31}$$

Proof. (a) The assumption $AA^{\dagger}B = B$ provides the solvability of the matrix equation $AX = B$.

The appropriate Lyapunov function is defined as

$$\mathcal{L}(t) = \frac{1}{2}||E_{G_{A,I,B}}(t)||_F^2 = \frac{1}{2}\mathrm{Tr}\left(\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}} E_{G_{A,I,B}}(t)\right).$$

Hence, from (25) and $\mathrm{d}\,\mathrm{Tr}(V^{\mathrm{T}}V) = 2\mathrm{Tr}(V^{\mathrm{T}}\mathrm{d}V)$, it holds that

$$\begin{aligned}\dot{\mathcal{L}}(t) &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Tr}\left(\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}} E_{G_{A,I,B}}(t)\right)\\ &= \mathrm{Tr}\left(\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}} \dot{E}_{A,I,B}(t)\right)\\ &= \mathrm{Tr}\left(\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}}\left(-\gamma\left(\left(A^{\mathrm{T}}A\right)^2 + I\right)\mathcal{F}\left(E_{G_{A,I,B}}(t)\right)\right)\right)\\ &= -\gamma\mathrm{Tr}\left(\left(\left(A^{\mathrm{T}}A\right)^2 + I\right)\mathcal{F}\left(E_{G_{A,I,B}}(t)\right)\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}}\right).\end{aligned}$$

According to similar results from [45], one can verify the following inequality:

$$\dot{\mathcal{L}}(t) \leq -\gamma\mathrm{Tr}\left(\left(\left(A^{\mathrm{T}}A\right)^2 + I\right)E_{G_{A,I,B}}(t)\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}}\right).$$

We also consider the following inequality from [46], which is valid for a real symmetric matrix K and a real symmetric positive-semidefinite matrix L of the same size:

$$\lambda_{\min}(K)\mathrm{Tr}(L) \leq \mathrm{Tr}(KL) \leq \lambda_{\max}(K)\mathrm{Tr}(L). \tag{32}$$

Now, the following can be chosen: $K = \left(A^{\mathrm{T}}A\right)^2 + I$ and $L = E_{G_{A,I,B}}(t)\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}}$. Consider $\lambda_{\min}\left(\left(A^{\mathrm{T}}A\right)^2\right) = \lambda^2_{\min}(A^{\mathrm{T}}A) = \sigma^4_{\min}(A)$, where $\lambda_{\min}(A)$ is the minimum eigenvalue of A, and $\sigma_{\min}(A) = \sqrt{\lambda_{\min}(A^{\mathrm{T}}A)}$ is the minimum singular value of A. Then, $1 + \sigma^4_{\min}(A) \geq 1$ is the minimum nonzero eigenvalue of $\left(A^{\mathrm{T}}A\right)^2 + I$, which implies

$$\dot{\mathcal{L}}(t) \leq -\gamma\left(1+\sigma^4_{\min}(A)\right)\mathrm{Tr}\left(E_{G_{A,I,B}}(t)\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}}\right). \tag{33}$$

From (33), it can be concluded

$$\dot{\mathcal{L}}(t)\begin{cases} <0 & \text{if } E_{G_{A,I,B}}(t)\neq 0\\ =0 & \text{if } E_{G_{A,I,B}}(t)=0. \end{cases} \tag{34}$$

According to (34), the Lyapunov stability theory confirms that $E_{A,I,B}(t) = AV(t) - B = 0$ is a globally asymptotically stable equilibrium point of the HGZNN(A, I, B) model (25). So, $E_{A,I,B}(t)$ converges to the zero matrix, i.e., $AV(t) \to B$, from any initial state $X(0)$.

(b) From (a), it follows that

$$\begin{aligned}\dot{\mathcal{L}} &\leq -\gamma\left(1+\sigma^4_{\min}(A)\right)\mathrm{Tr}\left(\left(E_{G_{A,I,B}}(t)\right)^{\mathrm{T}}E_{G_{A,I,B}}(t)\right)\\ &= -\gamma\left(1+\sigma^4_{\min}(A)\right)||E_{G_{A,I,B}}(t)||_F^2\\ &= -\frac{\gamma}{2}\left(1+\sigma^4_{\min}(A)\right)\mathcal{L}(t).\end{aligned}$$

This implies

$$\begin{aligned}\mathcal{L} \leq \mathcal{L}(0)e^{-\gamma\left(1+\sigma^4_{\min}(A)\right)t} &\iff\\ ||E_{G_{A,I,B}}(t)||_F^2 \leq ||E_{G_{A,I,B}}(0)||_F^2\, \mathrm{e}^{-\gamma\left(1+\sigma^4_{\min}(A)\right)} &\iff\\ ||E_{G_{A,I,B}}(t)||_F \leq ||E_{G_{A,I,B}}(0)||_F\, \mathrm{e}^{-\gamma/2\left(1+\sigma^4_{\min}(A)\right)},&\end{aligned}$$

which confirms the convergence rate (30) of HGZNN(A, I, B).

(c) This part of the proof can be verified with the particular case $B := I, D := B$ of Theorem 2.

□

Theorem 4. *Let* $C \in \mathbb{R}^{m\times l}, D \in \mathbb{R}^{n\times l}$ *be given and satisfy* $DC^{\dagger}C = D$*, and let* $V(t) \in \mathbb{R}^{n\times m}$ *be the state matrix of* (29)*, where* $\mathcal{F}$ *is defined by* f_{lin}*,* f_{ps} *or* f_{sps}*.*

(a) *Then,* $V(t)$ *achieves global convergence* $V(t)C \to D$ *when* $t \to +\infty$*, starting from any initial state* $V(0) \in \mathbb{R}^{n\times m}$*. The state matrix* $V(t) \in \mathbb{R}^{n\times m}$ *of* HGZNN(I, C, D) *is stable in the sense of Lyapunov.*

(b) *The exponential convergence rate of the* HGZNN(I, C, D) *model* (29) *in the linear case is equal to*

$$ECR(\mathrm{HGZNN}(I, C, D)) = \gamma\left(1+\sigma^4_{min}(C)\right). \tag{35}$$

(c) *The activation state variable matrix* $V(t)$ *of the model* HGZNN(I, C, D) *is convergent when* $t \to +\infty$ *with the equilibrium state matrix*

$$V(t) \to \tilde{V}_{V(0)} = DC^{\dagger} + V(0)(I - CC^{\dagger}). \tag{36}$$

Proof. (a) The assumption $DC^{\dagger}C = D$ ensures the solvability of the matrix equation $XC = D$.

Let us define the Lyapunov function by

$$\mathcal{L}(t) = \frac{1}{2}||E_{G_{I,C,D}}(t)||_F^2 = \frac{1}{2}\mathrm{Tr}\left(\left(E_{G_{I,C,D}}(t)\right)^{\mathrm{T}}E_{G_{I,C,D}}(t)\right).$$

Hence, from (29) and $\mathrm{d}\,\mathrm{Tr}(X^{\mathrm{T}}X) = 2\mathrm{Tr}(X^{\mathrm{T}}\mathrm{d}X)$, it holds that

$$\begin{aligned}\dot{\mathcal{L}}(t) &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{Tr}\Big(\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}E_{G_{I,C,D}}(t)\Big)\\ &= \mathrm{Tr}\Big(\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}\dot{E}_{G_{I,C,D}}(t)\Big)\\ &= \mathrm{Tr}\Big(\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}\Big(-\gamma\Big(\Big(CC^{\mathrm{T}}\Big)^2+I\Big)\mathcal{F}\Big(E_{G_{I,C,D}}(t)\Big)\Big)\Big)\\ &= -\gamma\mathrm{Tr}\Big(\Big(\Big(CC^{\mathrm{T}}\Big)^2+I\Big)\mathcal{F}\Big(E_{G_{I,C,D}}(t)\Big)\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}\Big).\end{aligned}$$

Following the principles from [45], one can verify the following inequality:

$$\dot{\mathcal{L}}(t) \leq -\gamma\mathrm{Tr}\Big(\Big(\Big(CC^{\mathrm{T}}\Big)^2+I\Big)E_{G_{I,C,D}}(t)\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}\Big).$$

Consider the inequality (32) with the particular settings $K = \big(CC^{\mathrm{T}}\big)^2 + I$, $L = E_{G_{I,C,D}}(t)\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}$. Let $\lambda_{\min}\Big(\big(CC^{\mathrm{T}}\big)^2\Big)$ be the minimum eigenvalue of $\big(CC^{\mathrm{T}}\big)^2$. Then, $1+\sigma^4_{\min}(C)) \geq 1$ is the minimal nonzero eigenvalue of $\big(CC^{\mathrm{T}}\big)^2 + I$, which implies

$$\dot{\mathcal{L}}(t) \leq -\gamma\Big(1+\sigma^4_{\min}(C)\Big)\mathrm{Tr}\Big(E_{G_{I,C,D}}(t)\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}\Big). \tag{37}$$

From (37), it can be concluded

$$\dot{\mathcal{L}}(t)\begin{cases} <0 & \text{if } E_{G_{I,C,D}}(t) \neq 0\\ =0 & \text{if } E_{G_{I,C,D}}(t) = 0.\end{cases} \tag{38}$$

According to (38), the Lyapunov stability theory confirms that $E_{G_{I,C,D}}(t) = V(t)C - D = 0$ is a globally asymptotically stable equilibrium point of the HGZNN(A, I, B) model (29). So, $E_{G_{I,C,D}}(t)$ converges to the zero matrix, i.e., $V(t)C \to D$, from any initial state $V(0)$.

(b) From (a), it follows

$$\begin{aligned}\dot{\mathcal{L}} &\leq -\gamma\Big(1+\sigma^4_{\min}(C)\Big)\mathrm{Tr}\Big(\Big(E_{G_{I,C,D}}(t)\Big)^{\mathrm{T}}E_{G_{I,C,D}}(t)\Big)\\ &= -\gamma\Big(1+\sigma^4_{\min}(C)\Big)||E_{G_{I,C,D}}(t)||_F^2\\ &= -\frac{\gamma}{2}\Big(1+\sigma^4_{\min}(C)\Big)\mathcal{L}(t).\end{aligned}$$

This implies

$$\begin{aligned}\mathcal{L} \leq \mathcal{L}(0)e^{-2\gamma\left(1+\sigma^4_{\min}(C)\right)t} &\iff\\ ||E_{G_{I,C,D}}(t)||_F^2 \leq ||E_{G_{I,C,D}}(0)||_F^2 e^{-2\gamma\left(1+\sigma^4_{\min}(C)\right)} &\iff\\ ||E_{G_{I,C,D}}(t)||_F \leq ||E_{G_{I,C,D}}(0)||_F e^{-\gamma\left(1+\sigma^4_{\min}(C)\right)},&\end{aligned}$$

which confirms the convergence rate (35) of HGZNN(I, C, D).

(c) This part of the proof can be verified with the particular case $A := I, B := C$ of Theorem 2.

□

Corollary 1. (a) *Let the matrices* $A \in \mathbb{R}^{k\times n}, B \in \mathbb{R}^{k\times m}$ *be given and satisfy* $AA^{\dagger}B = B$, *and let* $V(t) \in \mathbb{R}^{n\times m}$ *be the state matrix of* (25), *with an arbitrary nonlinear activation* $\mathcal{F}$. *Then,* $ECR(\text{GZNN}(A, I, B)) = \gamma$ *and* $ECR(\text{GGNN}(A, I, B)) = \gamma\sigma_{\min}(A)$.
(b) *Let the matrices* $C \in \mathbb{R}^{m\times l}, D \in \mathbb{R}^{n\times l}$ *be given and satisfy* $DC^{\dagger}C = D$, *and let* $V(t) \in \mathbb{R}^{n\times m}$ *be the state matrix of* (29) *with an arbitrary nonlinear activation* $\mathcal{F}$. *Then,* $ECR(\text{GZNN}(I, C, D)) = \gamma$ *and* $ECR(\text{GGNN}(I, C, D)) = \gamma\sigma_{\min}(C)$.

From Theorem 3 and Corollary 1(a), it follows that

$$\frac{ECR(\text{HGZNN}(A, I, B))}{ECR(\text{GZNN}(A, I, B))} = 1 + \sigma^4_{\min}(A) \geq 1. \tag{39}$$

$$\frac{ECR(\text{HGZNN}(A, I, B))}{ECR(\text{GGNN}(A, I, B))} = \frac{1 + \sigma^4_{\min}(A)}{\sigma^2_{\min}(A)} > 1. \tag{40}$$

$$\frac{ECR(\text{GZNN}(A, I, B))}{ECR(\text{GGNN}(A, I, B))} = \frac{1}{\sigma^2_{\min}(A)} \begin{cases} < 1, & \sigma_{\min}(A) > 1 \\ \geq 1, & \sigma_{\min}(A) \leq 1 \end{cases}. \tag{41}$$

Similarly, according to Theorem 4 and Corollary 1(b), it can be concluded that

$$\frac{ECR(\text{HGZNN}(I, C, D))}{ECR(\text{GZNN}(I, C, D))} = 1 + \sigma^4_{\min}(C) \geq 1. \tag{42}$$

$$\frac{ECR(\text{HGZNN}(I, C, D))}{ECR(\text{GGNN}(I, C, D))} = \frac{1 + \sigma^4_{\min}(C)}{\sigma^2_{\min}(C)} > 1. \tag{43}$$

$$\frac{ECR(\text{GZNN}(I, C, D))}{ECR(\text{GGNN}(I, C, D))} = \frac{1}{\sigma^2_{\min}(C)} \begin{cases} < 1, & \sigma_{\min}(C) > 1 \\ \geq 1, & \sigma_{\min}(C) \leq 1 \end{cases}. \tag{44}$$

Remark 1. (a) *According to* (40), *it follows that* $ECR(\text{HGZNN}(A, I, B)) > ECR(\text{GZNN}(A, I, B))$. *According to* (39), *it is obtained*

$$ECR(\text{HGZNN}(A, I, B)) \begin{cases} = ECR(\text{GZNN}(A, I, B)), & \sigma_{\min}(A) = 0 \\ > ECR(\text{GZNN}(A, I, B)), & \sigma_{\min}(A) > 0. \end{cases}$$

According to (41), *it follows*

$$ECR(\text{GZNN})(\text{A, I, B}) \begin{cases} < ECR(\text{GGNN}(A, I, B)), & \sigma_{\min}(A) > 1 \\ \geq ECR(\text{GGNN}(A, I, B)), & \sigma_{\min}(A) \leq 1 \end{cases}.$$

As a result, the following conclusions follow:

- HGZNN(A, I, B) *is always faster than* GGNN(A, I, B);
- HGZNN(A, I, B) *is faster than* GZNN(A, I, B) *in the case where* $\sigma_{\min}(A) > 0$;
- GZNN(A, I, B) *is faster than* GGNN$(A, I, B))$ *in the case where* $\sigma_{\min}(A) < 1$.

(b) *According to* (43), *it follows that* $ECR(\text{HGZNN}(I, C, D)) > ECR(\text{GZNN}(I, C, D))$. *According to* (42), *it follows that*

$$ECR(\text{HGZNN}(I, C, D)) \begin{cases} = ECR(\text{GZNN}(I, C, D)), & \sigma_{\min}(C) = 0 \\ > ECR(\text{GZNN}(I, C, D)), & \sigma_{\min}(C) > 0. \end{cases}$$

According to (41) *and* (44), *it can be verified*

$$ECR(\text{GZNN})(\text{I, C, D}) \begin{cases} < ECR(\text{GGNN}(I, C, D)), & \sigma_{\min}(C) > 1 \\ \geq ECR(\text{GGNN}(I, C, D)), & \sigma_{\min}(C) \leq 1 \end{cases}.$$

As a result, the following conclusions follow:

- $\text{HGZNN}(I, C, D)$ *is always faster than* $\text{GGNN}(I, C, D)$*;*
- $\text{HGZNN}(I, C, D)$ *is faster than* $\text{GZNN}(I, C, D)$ *in the case where* $\sigma_{\min}(C) > 0$*;*
- $\text{GZNN}(I, C, D)$ *is faster than* $\text{GGNN}(I, C, D))$ *in the case where* $\sigma_{\min}(C) < 1$*.*

Remark 2. *The particular HGZNN*$(A^{\mathrm{T}}A, I, A^{\mathrm{T}})$ *and GGNN*$(A^{\mathrm{T}}A, I, A^{\mathrm{T}})$ *designs define the corresponding modifications of the improved GNN design proposed in [26] if* $A^{\mathrm{T}}A$ *is invertible. In the dual case, HGZNN*$(I, CC^{\mathrm{T}}, C^{\mathrm{T}})$ *and GGNN*$(I, CC^{\mathrm{T}}, C^{\mathrm{T}})$ *define the corresponding modifications of the improved GNN design proposed in [26] if* CC^{T} *is invertible.*

Regularized HGZNN Model for Solving Matrix Equations

The convergence of $\text{HGZNN}(A, I, B)$ (resp. $\text{HGZNN}(I, C, D)$), as well as $\text{GGNN}(A, I, B)$ (resp. $\text{GGNN}(I, C, D)$), can be improved in the case where $\sigma_{\min}(A) > 0$ (resp. $\sigma_{\min}(C) > 0$). There exist two possible situations when the acceleration terms $A^{\mathrm{T}}A$ and CC^{T} improve the convergence. The first case assumes the invertibility of A (resp. C), and the second case assumes the left invertibility of A (resp. right invertibility of C). Still, in some situations, the matrices A and C could be rank-deficient. Hence, in the case where A and C are square and singular, it is useful to use the invertible matrices $A_1 := A + \lambda I$ and $C_1 := C + \lambda I$, $\lambda > 0$ instead of A and C and to consider the models $\text{HGZNN}(A_1, I, B)$ and $\text{HGZNN}(I, C_1, D)$. The following presents the convergence results considering the nonsingularity of A_1 and C_1.

Corollary 2. *Let* $A \in \mathbb{R}^{n\times n}$*,* $B \in \mathbb{R}^{n\times m}$ *be given and* $V(t) \in \mathbb{R}^{n\times m}$ *be the state matrix of* (25)*, where* $\mathcal{F}$ *is defined by* f_{lin}*,* f_{ps} *or* f_{sps}*. Let* $\lambda > 0$ *be a selected real number. Then, the following statements are valid:*

(a) *The state matrix* $V(t) \in \mathbb{R}^{n\times m}_r$ *of the model* $\text{HGZNN}(A_1, I, B)$ *converges globally to*

$$\tilde{V}_{V(0)} = A_1^{-1}B,$$

when $t \to +\infty$*, starting from any initial state* $X(0) \in \mathbb{R}^{n\times m}$*, and the solution is stable in the sense of Lyapunov.*

(b) *The exponential convergence rate of* $\text{HGZNN}(A_1, I, B)$ *in the case where* $\mathcal{F} = I$ *is equal to*

$$ECR(\text{HGZNN}(A_1, I, B)) = \gamma\Big(1 + \sigma^4_{\min}(A + \lambda I)\Big).$$

(c) *Let* $\tilde{V}_{V(0)}$ *be the limiting value of* $V(t)$ *when* $t \to +\infty$*. Then,*

$$\lim_{\lambda\to 0} \tilde{V}_{V(0)} = \lim_{\lambda\to 0}(A + \lambda I)^{-1}B. \tag{45}$$

Proof. Since $A + \lambda I$ is invertible, it follows that $V = (A + \lambda I)^{-1}B$.

From (31) and the invertibility of $A + \lambda I$, we conclude the validity of (a). In this case, it follows that

$$\begin{aligned}\tilde{V}_{V(0)} &= (A + \lambda I)^{-1}B + (I - (A + \lambda I)^{-1}(A + \lambda I))V(0)\\ &= (A + \lambda I)^{-1}B + (I - I)V(0)\\ &= (A + \lambda I)^{-1}B.\end{aligned}$$

The part (b) is proved analogously to the proof of Theorem 3. The last part (c) follows from (a). □

Corollary 3. *Let $C \in \mathbb{R}^{m\times m}$, $D \in \mathbb{R}^{n\times m}$ be given and $V(t) \in \mathbb{R}^{n\times m}$ be the state matrix of* (29)*, where $\mathcal{F} = I, \mathcal{F} = \mathcal{F}_{ps}$ or $\mathcal{F} = \mathcal{F}_{sps}$. Let $\lambda > 0$ be a selected real number. Then, the following statements are valid:*

(a) *The state matrix $V(t) \in \mathbb{R}_r^{n\times m}$ of* $\mathrm{HGZNN}(I, C_1, D)$ *converges globally to*

$$\tilde{V}_{V(0)} = D(C + \lambda I)^{-1},$$

when $t \to +\infty$, starting from any initial state $X(0) \in \mathbb{R}^{n\times m}$, and the solution is stable in the sense of Lyapunov.

(b) *The exponential convergence rate of* $\mathrm{HGZNN}(I, C_1, D)$ *in the case where $\mathcal{F} = I$ is equal to*

$$ECR(\mathrm{HGZNN}(I, C_1, D)) = \gamma\Big(1 + \sigma^4_{\min}(1)\Big).$$

(c) *Let $\tilde{V}_{V(0)}$ be the limiting value of $V(t)$ when $t \to +\infty$. Then,*

$$\lim_{\lambda\to 0} \tilde{V}_{V(0)} = \lim_{\lambda\to 0} D(C + \lambda I)^{-1}. \tag{46}$$

Proof. It can be proved analogously to Corollary 2. □

Remark 3. (a) *According to* (40)*, it can be concluded that*

$$ECR(\mathrm{HGZNN}(A_1, I, B)) > ECR(\mathrm{GZNN}(A_1, I, B)).$$

Based on (39) *it can be concluded*

$$ECR(\mathrm{HGZNN}(A_1, I, B)){>}ECR(\mathrm{GZNN}(A_1, I, B)).$$

According to (41)*, one concludes*

$$ECR(\mathrm{GZNN}(A_1, I, B)){<}ECR(\mathrm{GGNN}(A_1, I, B)).$$

(b) *According to* (43)*, it can be concluded*

$$ECR(\mathrm{HGZNN}(I, C_1, D)) > ECR(\mathrm{GZNN}(I, C_1, D)).$$

According to (42)*, it follows*

$$ECR(\mathrm{HGZNN}(I, C_1, D)) > ECR(\mathrm{GZNN}(I, C_1, D)).$$

Based on (41) *and* (44)*, it can be concluded*

$$ECR(\mathrm{GZNN}(I, C_1, D)){<}ECR(\mathrm{GGNN}(I, C_1, D)).$$

6. Numerical Examples on Hybrid Models

In this section, numerical examples are presented based on the Simulink implementation of the HGZNN formula. The previously mentioned three types of activation functions $f(\cdot)$ in (11), (12) and (13) will be used in the following examples. The parameters γ, the initial state $V(0)$ and the parameters ρ and ϱ of the nonlinear activation functions (12) and (13) are entered directly into the model, while the matrices A, B, C and D are defined from the workspace. We assume that $\rho = \varrho = 3$ in all examples. The ordinary differential equation solver in the configuration parameters is ode15s.

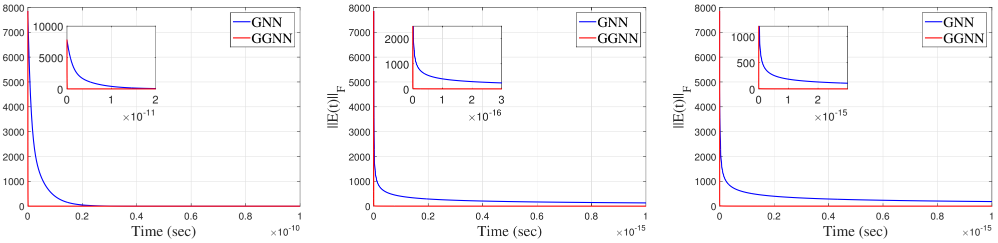
We present numerical examples in which we compare Frobenius norms $||E_G||_F$ and $||A^{-1}B - V(t)||_F$, which are generated by HGZNN, GZNN and GGNN.

Example 6. *Consider the matrix*

$$A = \begin{bmatrix} 0.49 & 0.276 & 0.498 & 0.751 & 0.959 \\ 0.446 & 0.68 & 0.96 & 0.255 & 0.547 \\ 0.646 & 0.655 & 0.34 & 0.506 & 0.139 \\ 0.71 & 0.163 & 0.585 & 0.699 & 0.149 \\ 0.755 & 0.119 & 0.224 & 0.891 & 0.258 \end{bmatrix}.$$

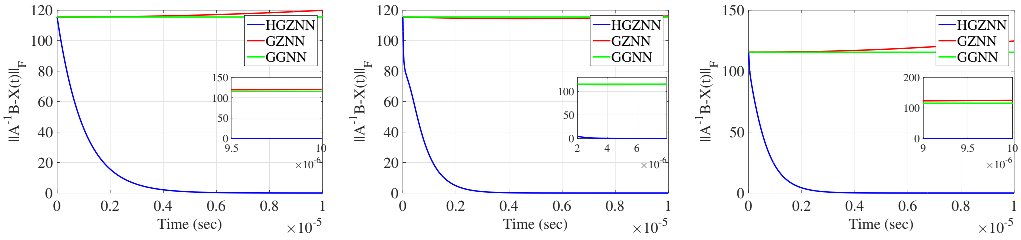
In this example, we compare the HGZNN(A, I, I) model with GZNN(A, I, I) and GGNN(A, I, I), considering all three types of activation functions. The gain parameter of the model is $\gamma = 10^6$, the initial state $V(0) = 0$, and the final time is $t = 0.00001$.

The Frobenius norm of the error matrix E_G in the HGZNN, GZNN and GGNN models for both linear and nonlinear activation functions are shown in Figure 12a–c, and the error matrices $A^{-1}B - V(t)$ of both models for linear and nonlinear activation functions are shown in Figure 13a–c. On each graph, the Frobenius norm of the error from the HGZNN formula vanishes faster to zero than those from the GZNN and GGNN models.



(**a**) (**b**) (**c**)

Figure 12. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E_{A,I,B}\|_F$ of HGZNN(A, I, I) compared to GGNN(A, I, I) and GZNN(A, I, I) in Example 6.



(**a**) (**b**) (**c**)

Figure 13. (**a**) Linear activation. (**b**) Power–sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|A^{-1}B - V(t)\|_F$ of HGZNN(A, I, I) compared to GGNN(A, I, I) and GZNN(A, I, I) in Example 6.

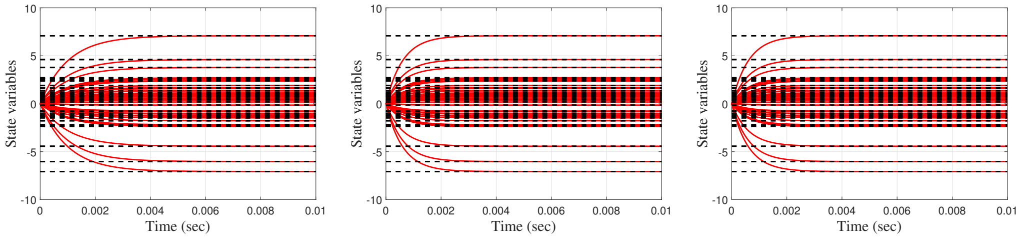
Example 7. *Consider the matrices*

$$A = \begin{bmatrix} 0.0818 & 0.0973 & 0.0083 & 0.0060 & 0.0292 & 0.0372 \\ 0.0818 & 0.0649 & 0.0133 & 0.0399 & 0.0432 & 0.0198 \\ 0.0722 & 0.0800 & 0.0173 & 0.0527 & 0.0015 & 0.0490 \\ 0.0150 & 0.0454 & 0.0391 & 0.0417 & 0.0984 & 0.0339 \\ 0.0660 & 0.0432 & 0.0831 & 0.0657 & 0.0167 & 0.0952 \\ 0.0519 & 0.0825 & 0.0803 & 0.0628 & 0.0106 & 0.0920 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.1649 & 0.1813 & 0.0851 & 0.1197 & 0.0138 & 0.1437 & 0.1558 \\ 0.1965 & 0.1759 & 0.0625 & 0.0942 & 0.0639 & 0.1937 & 0.0847 \\ 0.1460 & 0.1636 & 0.0323 & 0.1392 & 0.1062 & 0.1063 & 0.0182 \\ 0.0688 & 0.0521 & 0.0358 & 0.1400 & 0.1309 & 0.0650 & 0.0533 \\ 0.1168 & 0.1189 & 0.0846 & 0.1277 & 0.0815 & 0.0211 & 0.0307 \\ 0.0216 & 0.0045 & 0.0188 & 0.0067 & 0.1640 & 0.1222 & 0.0562 \end{bmatrix}.$$

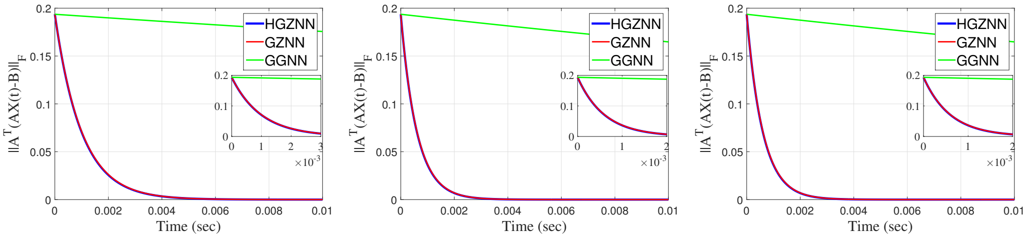
In this example, we compare the HGZNN(A, I, B) model with GZNN(A, I, B) and GGNN(A, I, B), considering all three types of activation functions. The gain parameter of the model is $\gamma = 1000$, the initial state $V(0) = 0$, and the final time is $t = 0.01$.

The elementwise trajectories of the state variable are shown with red lines in Figure 14a–c, for linear, power-sigmoid and smooth power-sigmoid activation functions, respectively. The solid red lines corresponding to HGZNN(A, I, B) converge to the black dashed lines of the theoretical solution X. It is observable that the trajectories indicate the usual convergence behavior, so the system is globally asymptotically stable. The error matrices E_G of the HGZNN, GZNN and GGNN models for both linear and nonlinear activation functions are shown in Figure 15a–c, and the residual matrices $A^{-1}B - X(t)$ of both models for linear and nonlinear activation functions are shown in Figure 16a–c. In each graph, for both error cases, the Frobenius norm of the error of the HGZNN formula is similar to the Frobenius norm of the error of the GZNN model, and they both converges faster to zero than the GGNN model.



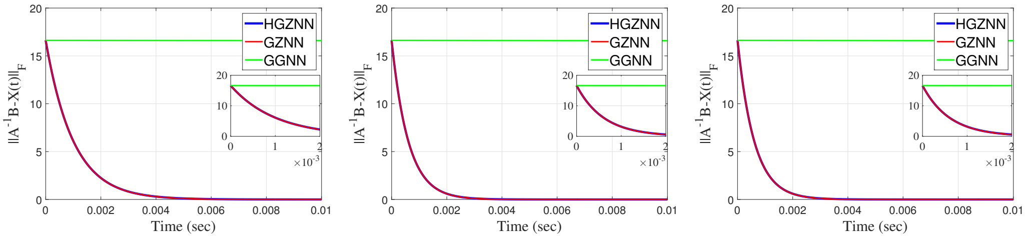
(**a**) (**b**) (**c**)

Figure 14. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. Elementwise convergence trajectories of the HGZNN(A, I, B) network in Example 7.



(**a**) (**b**) (**c**)

Figure 15. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. $\|E_{A,I,B}\|_F$ of HGZNN(A, I, B) compared to GGNN(A, I, B) and GZNN(A, I, B) in Example 7.



(**a**) (**b**) (**c**)

Figure 16. (**a**) Linear activation. (**b**) Power-sigmoid activation. (**c**) Smooth power–sigmoid activation. Frobenius norm of error matrix $A^{-1}B - X(t)$ of HGZNN(A, I, B) compared to GGNN(A, I, B) and GZNN(A, I, B) in Example 7.

Remark 4. *In this remark, we analyze the answer to the question, "how are the system parameters selected to obtain better performance?" The answer is complex and consists of several parts.*

1. *The gain parameter γ is the parameter with the most influence on the behavior of the observed dynamic systems. The general rule is "the parameter γ should be selected as large as possible". The numerical confirmation of this fact is investigated in Figure 7.*

2. *The influence of γ and AFs is indisputable. The larger the value of γ, the faster the convergence. And, clearly, AFs increase convergence compared to the linear models. In the presented numerical examples, we investigate the influence of three AFs: linear, power-sigmoid and smooth power-sigmoid.*
3. *The right question is as follows: what makes the* GGNN *better than the* GNN *under fair conditions that assume an identical environment during testing? Numerical experiments show better performance of the* GGNN *design compared to the* GNN *with respect to all three tested criteria: $\|E(t)\|_F$, $\|E_G(t)\|_F$ and $\|V(t) - V^*\|_F$. Moreover, Table 2 in Example 5 is aimed at convergence analysis. The general conclusion from the numerical data arranged in Table 2 is that the GGNN model is more efficient compared to the GNN in rank-deficient test matrices of larger order $m, n \geq 10$.*
4. *The convergence rate of the linear hybrid model* $\mathrm{HGZNN}(A, I, B))$ *depends on γ and the singular value $\sigma_{\min}(A)$, while the convergence rate of the hybrid model* $\mathrm{HGZNN}(I, C, D)$ *depends on γ and $\sigma_{\min}(C)$.*
5. *The convergence of the linear regularized hybrid model* $\mathrm{HGZNN}(A + \lambda I, I, B))$ *depends on γ, $\sigma_{\min}(A)$ and the regularization parameter $\lambda > 0$, while the convergence of the linear regularized hybrid model* $\mathrm{HGZNN}(I, C + \lambda I, D))$ *depends on γ, $\sigma_{\min}(C)$ and λ.*

In conclusion, it is reasonable to analyze the system parameter selections to obtain better performance. But the best performance is not defined.

7. Conclusions

We show that the error functions which make the basis of GNN and ZNN dynamical evolutions can be defined using the gradient of the Frobenius norm of the traditional error function $E(t)$. The result of such a strategy is the usage of the error function $E_G(t)$ for the basis of GNN dynamics, which results in the proposed GGNN model. The results related to the GNN model (called $\mathrm{GNN}(A, B, D)$) for solving the general matrix equation $AXB = D$ are extended in the GGNN model (called $\mathrm{GGNN}(A, B, D)$) in both theoretical and computational directions. In a theoretical sense, the convergence of the defined GGNN model is considered. It is shown that the neural state matrix $V(t)$ of the $\mathrm{GGNN}(A, B, D)$ model asymptotically converges to the solution of the matrix equation $AXB = D$ for an arbitrary initial state matrix $V(0)$ and coincides with the general solution of the linear matrix equation. A number of applications of GNN(A, B, D) are considered. All applications are globally convergent. Several particular appearances of the general matrix equation are observed and applied for computing various classes of generalized inverses. Illustrative numerical examples and simulation results were obtained using Matlab Simulink implementation and are presented to demonstrate the validity of the derived theoretical results. The influence of various nonlinear activations on the GNN models is considered in both the theoretical and computational directions. From the presented examples, it can be concluded that the GGNN model is faster and has a smaller error compared to the GNN model.

Further research can be oriented to the definition of finite-time convergent GGNN or GZNN models, as well as the definition of a noise-tolerant GGNN or GZNN design.

Author Contributions: Conceptualization, P.S.S. and G.V.M.; methodology, P.S.S., N.T., D.G. and V.S.; software, D.G., V.L.K. and N.T.; validation, G.V.M., M.J.P. and P.S.S.; formal analysis, M.J.P., N.T. and D.G.; investigation, M.J.P., G.V.M. and P.S.S.; resources, D.G., N.T., V.L.K. and V.S.; data curation, M.J.P., V.L.K., V.S., D.G. and N.T.; writing—original draft preparation, P.S.S., D.G. and N.T.; writing—review and editing, M.J.P. and G.V.M.; visualization, D.G. and N.T.; supervision, G.V.M.; project administration, M.J.P.; funding acquisition, G.V.M., M.J.P. and P.S.S. All authors have read and agreed to the published version of the manuscript.

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

Data Availability Statement: Data results are available on reader request.

Acknowledgments: Predrag Stanimirović is supported by the Science Fund of the Republic of Serbia (No. 7750185, Quantitative Automata Models: Fundamental Problems and Applications—QUAM).

Dimitrios Gerontitis receives financial support from the "Savas Parastatidis" named scholarship granted provided by the Bodossaki Foundation. Milena J. Petrović acknowledges support from a project supported by Ministry of Education and Science of Republic of Serbia, Grant No. 174025.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of the data; in the writing of the manuscript, or in the decision to publish the results.

References

1. 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; pp. 1–6. [CrossRef]
2. Zhang, Y.; Yi, C.; Guo, D.; Zheng, J. Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation. *Neural Comput. Appl.* **2011**, *20*, 1–7. [CrossRef]
3. Zhang, Y.; Xu, P.; Tan, L. Further studies on Zhang neural-dynamics and gradient dynamics for online nonlinear equations solving. In Proceedings of the 2009 IEEE International Conference on Automation and Logistics, Shenyang, China, 5–7 August 2009; pp. 566–571. [CrossRef]
4. Ben-Israel, A.; Greville, T.N.E. *Generalized Inverses: Theory and Applications*, 2nd ed.; CMS Books in Mathematics; Springer: New York, NY, USA, 2003.
5. Wang, G.; Wei, Y.; Qiao, S. *Generalized Inverses: Theory and Computations*; Science Press, Springer: Beijing, China, 2018.
6. Dash, P.; Zohora, F.T.; Rahaman, M.; Hasan, M.M.; Arifuzzaman, M. Usage of Mathematics Tools with Example in Electrical and Electronic Engineering. *Am. Sci. Res. J. Eng. Technol. Sci. (ASRJETS)* **2018**, *46*, 178–188.
7. Qin, F.; Lee, J. Dynamic methods for missing value estimation for DNA sequences. In Proceedings of the 2010 International Conference on Computational and Information Sciences, IEEE, Chengdu, China, 9–11 July 2010; pp. 442–445. [CrossRef]
8. Soleimani, F.; Stanimirović, P.S.; Soleimani, F. Some matrix iterations for computing generalized inverses and balancing chemical equations. *Algorithms* **2015**, *8*, 982–998. [CrossRef]
9. Udawat, B.; Begani, J.; Mansinghka, M.; Bhatia, N.; Sharma, H.; Hadap, A. Gauss Jordan method for balancing chemical equation for different materials. *Mater. Today Proc.* **2022**, *51*, 451–454. [CrossRef]
10. Doty, K.L.; Melchiorri, C.; Bonivento, C. A theory of generalized inverses applied to robotics. *Int. J. Robot. Res.* **1993**, *12*, 1–19. [CrossRef]
11. Li, L.; Hu, J. An efficient second-order neural network model for computing the Moore–Penrose inverse of matrices. *IET Signal Process.* **2022**, *16*, 1106–1117. [CrossRef]
12. Wang, X.; Tang, B.; Gao, X.G.; Wu, W.H. Finite iterative algorithms for the generalized reflexive and anti-reflexive solutions of the linear matrix equation $AXB = C$. *Filomat* **2017**, *31*, 2151–2162. [CrossRef]
13. Ding, F.; Chen, T. Gradient based iterative algorithms for solving a class of matrix equations. *IEEE Trans. Autom. Control* **2005**, *50*, 1216–1221. [CrossRef]
14. Ding, F.; Zhang, H. Gradient-based iterative algorithm for a class of the coupled matrix equations related to control systems. *IET Control Theory Appl.* **2014**, *8*, 1588–1595. [CrossRef]
15. Zhang, H. Quasi gradient-based inversion-free iterative algorithm for solving a class of the nonlinear matrix equations. *Comput. Math. Appl.* **2019**, *77*, 1233–1244. [CrossRef]
16. Wang, J. Recurrent neural networks for computing pseudoinverses of rank-deficient matrices. *SIAM J. Sci. Comput.* **1997**, *18*, 1479–1493. [CrossRef]
17. Fa-Long, L.; Zheng, B. Neural network approach to computing matrix inversion. *Appl. Math. Comput.* **1992**, *47*, 109–120. [CrossRef]
18. Wang, J. A recurrent neural network for real-time matrix inversion. *Appl. Math. Comput.* **1993**, *55*, 89–100. [CrossRef]
19. Wang, J. Recurrent neural networks for solving linear matrix equations. *Comput. Math. Appl.* **1993**, *26*, 23–34. [CrossRef]
20. Wei, Y. Recurrent neural networks for computing weighted Moore–Penrose inverse. *Appl. Math. Comput.* **2000**, *116*, 279–287. [CrossRef]
21. Xiao, L.; Zhang, Y.; Li, K.; Liao, B.; Tan, Z. FA novel recurrent neural network and its finite-time solution to time-varying complex matrix inversion. *Neurocomputing* **2019**, *331*, 483–492. [CrossRef]
22. Yi, C.; Chen, Y.; Lu, Z. Improved gradient-based neural networks for online solution of Lyapunov matrix equation. *Inf. Process. Lett.* **2011**, *111*, 780–786. [CrossRef]
23. Yi, C.; Qiao, D. Improved neural solution for the Lyapunov matrix equation based on gradient search. *Inf. Process. Lett.* **2013**, *113*, 876–881.
24. Xiao, L.; Li, K.; Tan, Z.; Zhang, Z.; Liao, B.; Chen, K.; Jin, L.; Li, S. Nonlinear gradient neural network for solving system of linear equations. *Inf. Process. Lett.* **2019**, *142*, 35–40. [CrossRef]
25. Xiao, L. A finite-time convergent neural dynamics for online solution of time-varying linear complex matrix equation. *Neurocomputing* **2015**, *167*, 254–259. [CrossRef]

26. Lv, X.; Xiao, L.; Tan, Z.; Yang, Z.; Yuan, J. Improved Gradient Neural Networks for solving Moore–Penrose Inverse of full-rank matrix. *Neural Process. Lett.* **2019**, *50*, 1993–2005. [CrossRef]
27. Wang, J. Electronic realisation of recurrent neural network for solving simultaneous linear equations. *Electron. Lett.* **1992**, *28*, 493–495. [CrossRef]
28. Zhang, Y.; Chen, K.; Tan, H.Z. Performance analysis of gradient neural network exploited for online time-varying matrix inversion. *IEEE Trans. Autom. Control.* **2009**, *54*, 1940–1945. [CrossRef]
29. Wang, J.; Li, H. Solving simultaneous linear equations using recurrent neural networks. *Inf. Sci.* **1994**, *76*, 255–277. [CrossRef]
30. Tan, Z.; Chen, H. Nonlinear function activated GNN versus ZNN for online solution of general linear matrix equations. *J. Frankl. Inst.* **2023**, *360*, 7021–7036. [CrossRef]
31. Tan, Z.; Hu, Y.; Chen, K. On the investigation of activation functions in gradient neural network for online solving linear matrix equation. *Neurocomputing* **2020**, *413*, 185–192. [CrossRef]
32. Tan, Z. Fixed-time convergent gradient neural network for solving online sylvester equation. *Mathematics* **2022**, *10*, 3090. [CrossRef]
33. Wang, D.; Liu, X.-W. A gradient-type noise-tolerant finite-time neural network for convex optimization. *Neurocomputing* **2022**, *49*, 647–656. [CrossRef]
34. Stanimirović, P.S.; Petković, M.D. Gradient neural dynamics for solving matrix equations and their applications. *Neurocomputing* **2018**, *306*, 200–212. [CrossRef]
35. Stanimirović, P.S.; Katsikis, V.N.; Li, S. Hybrid GNN-ZNN models for solving linear matrix equations. *Neurocomputing* **2018**, *316*, 124–134. [CrossRef]
36. Sowmya, G.; Thangavel, P.; Shankar, V. A novel hybrid Zhang neural network model for time-varying matrix inversion. *Eng. Sci. Technol. Int. J.* **2022**, *26*, 101009. [CrossRef]
37. Wu, W.; Zheng, B. Improved recurrent neural networks for solving Moore–Penrose inverse of real-time full-rank matrix. *Neurocomputing* **2020**, *418*, 221–231. [CrossRef]
38. Zhang, Y.; Wang, C. Gradient-Zhang neural network solving linear time-varying equations. In Proceedings of the 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA), Chengdu, China, 16–19 December 2022; pp. 396–403. [CrossRef]
39. Wang, C., Zhang, Y. Theoretical Analysis of Gradient-Zhang Neural Network for Time-Varying Equations and Improved Method for Linear Equations. In *Neural Information Processing*; ICONIP 2023, Lecture Notes in Computer Science; Luo, B., Cheng, L., Wu, Z.G., Li, H., Li, C., Eds.; Springer: Singapore, 2024; Volume 14447. [CrossRef]
40. 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. [CrossRef]
41. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer: New York, NY, USA, 1999.
42. Stanimirović, P.S.; Petković, M.D.; Gerontitis, D. Gradient neural network with nonlinear activation for computing inner inverses and the Drazin inverse. *Neural Process. Lett.* **2017**, *48*, 109–133. [CrossRef]
43. Smoktunowicz, A.; Smoktunowicz, A. Set-theoretic solutions of the Yang–Baxter equation and new classes of R-matrices. *Linear Algebra Its Appl.* **2018**, *546*, 86–114. [CrossRef]
44. Baksalary, O.M.; Trenkler, G. On matrices whose Moore–Penrose inverse is idempotent. *Linear Multilinear Algebra* **2022**, *70*, 2014–2026. [CrossRef]
45. Wang, X.Z.; Ma, H.; Stanimirović, P.S. Nonlinearly activated recurrent neural network for computing the Drazin inverse. *Neural Process. Lett.* **2017**, *46*, 195–217. [CrossRef]
46. Wang, S.D.; Kuo, T.S.; Hsu, C.F. Trace bounds on the solution of the algebraic matrix Riccati and Lyapunov equation. *IEEE Trans. Autom. Control* **1986**, *31*, 654–656. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.