Hindawi
Complexity
Volume 2017, Article ID 6429725, 27 pages
https://doi.org/10.1155/2017/6429725

*Research Article*

# Conditions for Existence, Representations, and Computation of Matrix Generalized Inverses

**Predrag S. Stanimirović,[1] Miroslav Ćirić,[1] Igor Stojanović,[2] and Dimitrios Gerontitis[3]**

[1] *Faculty of Science and Mathematics, Department of Computer Science, University of Niš, Višegradska 33, 18000 Niš, Serbia*
[2] *Faculty of Computer Science, Goce Delčev University, Goce Delčev 89, 2000 Štip, Macedonia*
[3] *Aristoteleion Panepistimion, Thessalonikis, Greece*

Correspondence should be addressed to Predrag S. Stanimirović; pecko@pmf.ni.ac.rs

Received 3 January 2017; Accepted 18 April 2017; Published 5 June 2017

Academic Editor: Sigurdur F. Hafstein



Conditions for the existence and representations of {2}-, {1}-, and {1, 2}-inverses which satisfy certain conditions on ranges and/or null spaces are introduced. These representations are applicable to complex matrices and involve solutions of certain matrix equations. Algorithms arising from the introduced representations are developed. Particularly, these algorithms can be used to compute the Moore-Penrose inverse, the Drazin inverse, and the usual matrix inverse. The implementation of introduced algorithms is defined on the set of real matrices and it is based on the Simulink implementation of GNN models for solving the involved matrix equations. In this way, we develop computational procedures which generate various classes of inner and outer generalized inverses on the basis of resolving certain matrix equations. As a consequence, some new relationships between the problem of solving matrix equations and the problem of numerical computation of generalized inverses are established. Theoretical results are applicable to complex matrices and the developed algorithms are applicable to both the time-varying and time-invariant real matrices.

## 1. Introduction, Motivation, and Preliminaries

Let $\mathbb{C}^{m\times n}$ and $\mathbb{C}_r^{m\times n}$ (resp., $\mathbb{R}^{m\times n}$ and $\mathbb{R}_r^{m\times n}$) denote the set of complex (resp., real) $m\times n$ matrices and all complex (resp., real) $m\times n$ matrices of rank $r$. As usual, the notation $I$ denotes the unit matrix of an appropriate order. Further, by $A^*$, $\mathscr{R}(A)$, $\operatorname{rank}(A)$, and $\mathscr{N}(A)$ are denoted as the conjugate transpose, the range, the rank, and the null space of $A\in\mathbb{C}^{m\times n}$.

The problem of pseudoinverses computation leads to the, so-called, Penrose equations:

$$\begin{aligned}&(1)\ AXA=A,\\&(2)\ XAX=X,\\&(3)\ (AX)^*=AX,\\&(4)\ (XA)^*=XA.\end{aligned}\tag{1}$$

The set of all matrices obeying the conditions contained in $\mathcal{S}$ is denoted by $A\{\mathcal{S}\}$. Any matrix from $A\{\mathcal{S}\}$ is called the $\mathcal{S}$-inverse of $A$ and is denoted by $A^{(\mathcal{S})}$. $A\{\mathcal{S}\}_s$ is denoted as the set of all $\mathcal{S}$-inverses of $A$ of rank $s$. For any matrix $A$ there exists a unique element in the set $A\{1,2,3,4\}$, called the Moore-Penrose inverse of $A$, which is denoted by $A^\dagger$. The Drazin inverse of a square matrix $A\in\mathbb{C}^{n\times n}$ is the unique matrix $X\in\mathbb{C}^{n\times n}$ which fulfills matrix equation (2) in conjunction with

$$\begin{aligned}&\left(1^k\right)\ A^{l+1}X=A^l,\quad l\ge\operatorname{ind}(A),\\&(5)\quad AX=XA,\end{aligned}\tag{2}$$

and it is denoted by $X=A^{\mathrm{D}}$. Here, the notation $\operatorname{ind}(A)$ denotes the index of a square matrix $A$ and it is defined by $\operatorname{ind}(A)=\min\{j\mid \operatorname{rank}(A^j)=\operatorname{rank}(A^{j+1})\}$. In the case $\operatorname{ind}(A)=1$, the Drazin inverse becomes the group inverse $X=A^{\#}$. For other important properties of generalized inverses see [1, 2].

An element $X \in A\{S\}$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$ (resp., $\mathscr{N}(X) = \mathscr{N}(C)$) is denoted by $A^{(S)}_{\mathscr{R}(B),*}$ (resp., $A^{(S)}_{*,\mathscr{N}(C)}$). If $X$ satisfies both the conditions $\mathscr{R}(X) = \mathscr{R}(B)$ and $\mathscr{N}(X) = \mathscr{N}(C)$ it is denoted by $A^{(S)}_{\mathscr{R}(B),\mathscr{N}(C)}$. The set of all $\{S\}$-inverses of $A$ with the prescribed range $\mathscr{R}(B)$ (resp., prescribed kernel $\mathscr{N}(X) = \mathscr{N}(C)$) is denoted by $X = A\{S\}_{\mathscr{R}(B),*}$ (resp., $A\{S\}_{*,\mathscr{N}(C)}$). Definitions and notation used in the further text are from the books by Ben-Israel and Greville [1] and Wang et al. [2].

Full-rank representation of $\{2\}$-inverses with the prescribed range and null space is determined in the next proposition, which originates from [3].

**Proposition 1** (see [3]). *Let $A \in \mathbb{C}^{m\times n}_r$, let $T$ be a subspace of $\mathbb{C}^n$ of dimension $s \le r$, and let $S$ be a subspace of $\mathbb{C}^m$ of dimensions $m - s$. In addition, suppose that $R \in \mathbb{C}^{n\times m}$ satisfies $\mathscr{R}(R) = T$, $\mathscr{N}(R) = S$. Let $R$ have an arbitrary full-rank decomposition; that is, $R = FG$. If $A$ has a $\{2\}$-inverse $A^{(2)}_{T,S}$, then*

(1) *$GAF$ is an invertible matrix;*

(2) $A^{(2)}_{T,S} = F(GAF)^{-1}G$.

The Moore-Penrose inverse $A^{\dagger}$, the Drazin inverse $A^{\mathrm{D}}$, and the group inverse $A^{\#}$ are generalized inverses $A^{(2)}_{T,S}$ for appropriate choice of subspaces $T$ and $S$. For example, the following is valid for a rectangular matrix $A$ [2]:

$$\begin{aligned} A^{\dagger} &= A^{(2)}_{\mathscr{R}(A^*),\mathscr{N}(A^*)}, \\ A^{\mathrm{D}} &= A^{(2)}_{\mathscr{R}(A^k),\mathscr{N}(A^k)}, \quad k \ge \operatorname{ind} A, \\ A^{\#} &= A^{(2)}_{\mathscr{R}(A),\mathscr{N}(A)}. \end{aligned} \tag{3}$$

The full-rank representation $A^{(2)}_{T,S} = F(GAF)^{-1}G$ has been applied in numerical calculations. For example, such a representation has been exploited to define the determinantal representation of the $A^{(2)}_{T,S}$ inverse in [3] or the determinantal representation of the set $A\{2\}_s$ in [4]. A lot of iterative methods for computing outer inverses with the prescribed range and null space have been developed. An outline of these numerical methods can be found in [5–13].

A drawback of the representation given in Proposition 1 arises from the fact that it is based on the full-rank decomposition $R = FG$ and gives the representation of $A^{(2)}_{\mathscr{R}(R),\mathscr{N}(R)}$. Besides, it requires invertibility of $GAF$; in the opposite case, it is not applicable. Finally, representations of outer inverses with given only range or null space or the representations of inner inverses with the prescribed range and/or null space are not covered. For this purpose, our further motivation is well-known representations of generalized inverses $A^{(2)}_{T,S}$ and $A^{(1,2)}_{T,S}$, given by the Urquhart formula. The Urquhart formula was originated [14] and later extended in [2, Theorem 1.3.3] and [1, Theorem 13, P. 72]. We restate it for the sake of completeness.

**Proposition 2** (Urquhart formula). *Let $A \in \mathbb{C}^{m\times n}_r$, $U \in \mathbb{C}^{n\times p}$, $V \in \mathbb{C}^{q\times m}$, and $X = U(VAU)^{(1)}V$, where $(VAU)^{(1)}$ is a fixed but arbitrary element of $(VAU)\{1\}$. Then*

(1) *$X \in A\{1\}$ if and only if $\operatorname{rank}(VAU) = r$;*

(2) *$X \in A\{2\}$ and $\mathscr{R}(X) = \mathscr{R}(U)$ if and only if $\operatorname{rank}(VAU) = \operatorname{rank}(U)$;*

(3) *$X \in A\{2\}$ and $\mathscr{N}(X) = \mathscr{N}(V)$ if and only if $\operatorname{rank}(VAU) = \operatorname{rank}(V)$;*

(4) *$X = A^{(2)}_{\mathscr{R}(U),\mathscr{N}(V)}$ if and only if $\operatorname{rank}(VAU) = \operatorname{rank}(U) = \operatorname{rank}(V)$;*

(5) *$X = A^{(1,2)}_{\mathscr{R}(U),\mathscr{N}(V)}$ if and only if $\operatorname{rank}(VAU) = \operatorname{rank}(U) = \operatorname{rank}(V) = r$.*

Later, our motivation is the notion of a $(b, c)$-inverse of an element $a$ in a semigroup, introduced by Drazin in [15]. Following the result from [9], the representation of outer inverses given in Proposition 1 investigates $(R, R)$-inverses. Our tendency is to consider representations and computations of $(B, C)$-inverses, where $B$ and $C$ could be different.

Finally, our intention is to define appropriate numerical algorithms for computing generalized inverses

$$A^{(2)}_{T,S}, A^{(1)}_{T,*}, A^{(1)}_{*,S}, A^{(1)}_{T,S}, A^{(2)}_{T,*}, A^{(2)}_{*,S}, A^{(1,2)}_{T,*}, A^{(1,2)}_{*,S}, A^{(1,2)}_{T,S} \tag{4}$$

in both the time-varying and time-invariant cases. For this purpose, we observed that the neural dynamic approach has been exploited as a powerful tool in solving matrix algebra problems, due to its parallel distributed nature as well as its convenience of hardware implementation. Recently, many authors have shown great interest for computing the inverse or the pseudoinverse of square and full-rank rectangular matrices on the basis of gradient-based recurrent neural networks (GNNs) or Zhang neural networks (ZNNs). Neural network models for the inversion and pseudo-inversion of square and full-row or full-column rank rectangular matrices were developed in [16–18]. Various recurrent neural networks for computing generalized inverses of rank-deficient matrices were introduced in [19–23]. RNNs designed for calculating the pseudoinverse of rank-deficient matrices were created in [21]. Three recurrent neural networks for computing the weighted Moore-Penrose inverse were introduced in [22]. A feedforward neural network architecture for computing the Drazin inverse was proposed in [19]. The dynamic equation and induced gradient recurrent neural network for computing the Drazin inverse were defined in [24]. Two gradient-based RNNs for generating outer inverses with prescribed range and null space in the time-invariant case were introduced in [25]. Two additional dynamic state equations and corresponding gradient-based RNNs for generating the class of outer inverses of time-invariant real matrices were proposed in [26].

The global organization of the paper is as follows. Conditions for the existence and representations of generalized inverses included in (4) are given in Section 2. Numerical algorithms arising from the representations derived in Section 2 are defined in Section 3. In this way, Section 3 defines algorithms for computing various classes of inner and outer generalized inverses by means of derived solutions of certain matrix equations. Main particular cases are presented in the same section as well as the global computational complexity of introduced algorithms. Illustrative simulation and numerical examples are presented in Section 4.

## 2. Existence and Representations of Generalized Inverses

Theorem 3 provides a theoretical basis for computing outer inverses with the prescribed range space.

**Theorem 3.** *Let $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times k}$.*

(a) *The following statements are equivalent:*

(i) *There exists a $\{2\}$-inverse $X$ of $A$ satisfying $\mathcal{R}(X) = \mathcal{R}(B)$, denoted by $A^{(2)}_{\mathcal{R}(B),*}$.*
(ii) *There exists $U \in \mathbb{C}^{k\times m}$ such that $BUAB = B$.*
(iii) $\mathcal{N}(AB) = \mathcal{N}(B)$.
(iv) $\operatorname{rank}(AB) = \operatorname{rank}(B)$.
(v) *$B(AB)^{(1)}AB = B$, for some (equivalently every) $(AB)^{(1)} \in (AB)\{1\}$.*

(b) *If the statements in (a) are true, then the set of all outer inverses with the prescribed range $\mathcal{R}(B)$ is represented by*

$$A\{2\}_{\mathcal{R}(B),*} = \left\{B(AB)^{(1)} \mid (AB)^{(1)} \in (AB)\{1\}\right\} = \left\{BU \mid U \in \mathbb{C}^{k\times m},\ BUAB = B\right\}. \quad (5)$$

*Moreover,*

$$A\{2\}_{\mathcal{R}(B),*} = \left\{B(AB)^{(1)} + BY\left(I_m - AB(AB)^{(1)}\right) \mid Y \in \mathbb{C}^{k\times m}\right\}, \quad (6)$$

*where $(AB)^{(1)} \in (AB)\{1\}$ is arbitrary but fixed.*

*Proof.* (a) (i) $\Rightarrow$ (ii). Let $X \in \mathbb{C}^{n\times m}$ such that $XAX = X$ and $\mathcal{R}(X) = \mathcal{R}(B)$. Then $X = BU$ and $B = XW$, for some $U \in \mathbb{C}^{k\times m}$ and $W \in \mathbb{C}^{m\times k}$, so $B = XW = XAXW = XAB = BUAB$.

(ii) $\Rightarrow$ (iii). As we know, $\mathcal{N}(B) \subseteq \mathcal{N}(AB)$. On the other hand, taking into account $BUAB = B$ for some $U \in \mathbb{C}^{k\times m}$, it follows that $\mathcal{N}(AB) \subseteq \mathcal{N}(BUAB) = \mathcal{N}(B)$, and hence $\mathcal{N}(AB) = \mathcal{N}(B)$.

(iii) $\Rightarrow$ (v). Let $(AB)^{(1)}$ be an arbitrary $\{1\}$-inverse of $AB$. As $\mathcal{N}(AB) = \mathcal{N}(B)$ implies $B = VAB$, for some $V \in \mathbb{C}^{n\times m}$, it follows that

$$B = VAB = VAB(AB)^{(1)}AB = B(AB)^{(1)}AB. \quad (7)$$

(v) $\Rightarrow$ (i). Let $B = B(AB)^{(1)}AB$, for some $(AB)^{(1)} \in (AB)\{1\}$, and set $X = B(AB)^{(1)}$. Then

$$XAX = B(AB)^{(1)}AB(AB)^{(1)} = B(AB)^{(1)} = X, \quad (8)$$

and by $X = B(AB)^{(1)}$ and $B = B(AB)^{(1)}AB = XAB$ it follows that $X$ is a $\{2\}$-inverse of $A$ which satisfies $\mathcal{R}(X) = \mathcal{R}(B)$.

(iii) $\Rightarrow$ (v). This result is well-known.

(b) From the proofs of (i) $\Rightarrow$ (ii) and (iv) $\Rightarrow$ (i), and the fact that $B = BUAB$ implies $U \in (AB)\{1\}$, it follows that

$$\begin{aligned} A\{2\}_{\mathcal{R}(B),*} &\subseteq \left\{BU \mid U \in \mathbb{C}^{k\times m},\ BUAB = B\right\} \\ &\subseteq \left\{B(AB)^{(1)} \mid (AB)^{(1)} \in (AB)\{1\}\right\} \\ &\subseteq A\{2\}_{\mathcal{R}(B),*}, \end{aligned} \quad (9)$$

and hence (5) holds.

According to Theorem 1 [1, Section 2] (or [2, Theorem 1.2.5]), the condition (v) ensures consistency of the matrix equation $BUAB = B$ and gives its general solution

$$\left\{U \in \mathbb{C}^{k\times m} \mid BUAB = B\right\} = \left\{B^{(1)}B(AB)^{(1)} + Y - B^{(1)}BYAB(AB)^{(1)} \mid Y \in \mathbb{C}^{k\times m}\right\}, \quad (10)$$

whence we obtain

$$\begin{aligned} A\{2\}_{\mathcal{R}(B),*} &= \left\{BU \mid U \in \mathbb{C}^{k\times m},\ BUAB = B\right\} \\ &= \left\{B(AB)^{(1)} + BY\left(I_m - AB(AB)^{(1)}\right) \mid Y \in \mathbb{C}^{k\times m}\right\}. \end{aligned} \quad (11)$$

This proves is that (6) is true. □

*Remark 4.* Five equivalent conditions for the existence and representations of the class of generalized inverses $A^{(2)}_{T,*}$ were given in [27, Theorem 1]. Theorem 3 gives two new and important conditions (i) and (v). These conditions are related with solvability of certain matrix equations. Further, the representations of generalized inverses $A^{(2)}_{T,*}$ were presented in [27, Theorem 2]. Theorem 3 gives two new and important representations: the second representation in (5) and representation (6).

Theorem 5 provides a theoretical basis for computing outer inverses with the prescribed kernel. These results are new in the literature, according to our best knowledge.

**Theorem 5.** *Let $A \in \mathbb{C}^{m\times n}$ and $C \in \mathbb{C}^{l\times m}$.*

(a) *The following statements are equivalent:*

(i) *There exists a $\{2\}$-inverse $X$ of $A$ satisfying $\mathcal{N}(X) = \mathcal{N}(C)$, denoted by $A^{(2)}_{*,\mathcal{N}(C)}$.*
(ii) *There exists $V \in \mathbb{C}^{n\times l}$ such that $CAVC = C$.*
(iii) $\mathcal{R}(CA) = \mathcal{R}(C)$.
(iv) $\operatorname{rank}(CA) = \operatorname{rank}(C)$.
(v) *$CA(CA)^{(1)}C = C$, for some (equivalently every) $(CA)^{(1)} \in (CA)\{1\}$.*

(b) *If the statements in (a) are true, then the set of all outer inverses with the prescribed null space $\mathcal{N}(C)$ is represented by*

$$\begin{aligned} A\{2\}_{*,\mathcal{N}(C)} &= \left\{(CA)^{(1)} C \mid (CA)^{(1)} \in (CA)\{1\}\right\} \\ &= \left\{VC \mid V \in \mathbb{C}^{n\times l},\ CAVC = C\right\}. \end{aligned} \tag{12}$$

*Moreover,*

$$\begin{aligned} &A\{2\}_{*,\mathcal{N}(C)} \\ &= \left\{(CA)^{(1)} C + \left(I_l - (CA)^{(1)} CA\right) YC \mid Y \in \mathbb{C}^{n\times l}\right\}, \end{aligned} \tag{13}$$

*where $(CA)^{(1)}$ is an arbitrary fixed matrix from $(CA)\{1\}$.*

*Proof.* The proof is analogous to the proof of Theorem 3. □

Theorem 6 is a theoretical basis for computing a {2}-inverse with the prescribed range and null space.

**Theorem 6.** *Let $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times k}$, and $C \in \mathbb{C}^{l\times m}$.*

(a) *The following statements are equivalent:*

(i) *There exists a {2}-inverse $X$ of $A$ satisfying $\mathcal{R}(X) = \mathcal{R}(B)$ and $\mathcal{N}(X) = \mathcal{N}(C)$.*
(ii) *There exist $U \in \mathbb{C}^{k\times l}$ such that $BUCAB = B$ and $CABUC = C$.*
(iii) *There exist $U, V \in \mathbb{C}^{k\times l}$ such that $BUCAB = B$ and $CABVC = C$.*
(iv) *There exist $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times l}$ such that $BUAB = B$, $CAVC = C$, and $BU = VC$.*
(v) *There exist $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times l}$ such that $CABU = C$ and $VCAB = B$.*
(vi) *$\mathcal{N}(CAB) = \mathcal{N}(B)$, $\mathcal{R}(CAB) = \mathcal{R}(C)$.*
(vii) *$\operatorname{rank}(CAB) = \operatorname{rank}(B) = \operatorname{rank}(C)$.*
(viii) *$B(CAB)^{(1)}CAB = B$ and $CAB(CAB)^{(1)}C = C$, for some (equivalently every) $(CAB)^{(1)} \in (CAB)\{1\}$.*

(b) *If the statements in (a) are true, then the unique {2}-inverse of $A$ with the prescribed range $\mathcal{R}(B)$ and null space $\mathcal{N}(C)$ is represented by*

$$A^{(2)}_{\mathcal{R}(B),\mathcal{N}(C)} = B(CAB)^{(1)}C = BUC, \tag{14}$$

*for arbitrary $(CAB)^{(1)} \in (CAB)\{1\}$ and arbitrary $U \in \mathbb{C}^{k\times l}$ satisfying $BUCAB = B$ and $CABUC = C$.*

*Proof.* (a) (i) ⇒ (ii). Let $X \in \mathbb{C}^{n\times m}$ be such that $XAX = X$, $\mathcal{R}(X) = \mathcal{R}(B)$, and $\mathcal{N}(X) = \mathcal{N}(C)$. Then there exists $U \in \mathbb{C}^{k\times l}$ such that $X = BUC$. Also, $B$ and $C$ satisfy $B = XW$ and $C = VX$, for some $W \in \mathbb{C}^{m\times k}$, $V \in \mathbb{C}^{l\times n}$. This further implies

$$\begin{aligned} B &= XW = XAXW = XAB = BUCAB, \\ C &= VX = VXAX = CAX = CABUC. \end{aligned} \tag{15}$$

(ii) ⇒ (vi). According to $CABUC = C$, for some $U \in \mathbb{C}^{k\times l}$, it follows that

$$\mathcal{R}(C) = \mathcal{R}(CABUC) \subseteq \mathcal{R}(CAB) \subseteq \mathcal{R}(C), \tag{16}$$

and thus $\mathcal{R}(CAB) = \mathcal{R}(C)$. Further, by $B = BUCAB$, for some $U \in \mathbb{C}^{k\times l}$, it follows that

$$\mathcal{N}(B) \subseteq \mathcal{N}(CAB) \subseteq \mathcal{N}(BUCAB) = \mathcal{N}(B), \tag{17}$$

which yields $\mathcal{N}(CAB) = \mathcal{N}(B)$.

(vi) ⇒ (viii). Let $(CAB)^{(1)}$ be an arbitrary {1}-inverse of $CAB$. Since $\mathcal{R}(CAB) = \mathcal{R}(C)$ implies $C = CABW$, for some $W \in \mathbb{C}^{k\times m}$, it follows that

$$\begin{aligned} C &= CABW = CAB(CAB)^{(1)} CABW \\ &= CAB(CAB)^{(1)} C. \end{aligned} \tag{18}$$

Similarly, $\mathcal{N}(CAB) = \mathcal{N}(B)$ implies $B = VCAB$, for some $V \in \mathbb{C}^{n\times l}$ and

$$\begin{aligned} B &= VCAB = VCAB(CAB)^{(1)} CAB \\ &= B(CAB)^{(1)} CAB. \end{aligned} \tag{19}$$

(viii) ⇒ (i). Let $CAB(CAB)^{(1)}C = C$, for some $(CAB)^{(1)} \in (CAB)\{1\}$, and set $X = B(CAB)^{(1)}C$. Then

$$\begin{aligned} XAX &= B(CAB)^{(1)} CAB(CAB)^{(1)} C = B(CAB)^{(1)} C \\ &= X \end{aligned} \tag{20}$$

and by $X = B(CAB)^{(1)}C$, $B = B(CAB)^{(1)}CAB = XAB$, and $C = CAB(CAB)^{(1)}C = CAX$ it follows that $X$ is a {2}-inverse of $A$ which satisfies $\mathcal{R}(X) = \mathcal{R}(B)$, $\mathcal{N}(X) = \mathcal{N}(C)$.

(vi) ⇔ (vii). This statement follows from [2, Theorem 1.1.3, P. 3].

(ii) ⇒ (iii). This is evident.

(iii) ⇒ (ii). Let $U, V \in \mathbb{C}^{k\times l}$ be arbitrary matrices such that $BUCAB = B$ and $CABVC = C$. Then

$$BUC = BUCABVC = BVC, \tag{21}$$

whence

$$\begin{aligned} B &= BUCAB = BVCAB, \\ C &= CABVC = CABUC. \end{aligned} \tag{22}$$

Thus, (ii) holds.

(ii) ⇒ (iv). $U \in \mathbb{C}^{k\times l}$ such that $BUCAB = B$ and $CABUC = C$. Then

$$\begin{aligned} B &= B(UC)AB, \\ C &= CA(BU)C, \\ B(UC) &= (BU)C, \end{aligned} \tag{23}$$

which means that (iv) is true.

(iv) ⇒ (v). Let $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times l}$ such that $BUAB = B$, $CAVC = C$, and $BU = VC$. Then

$$\begin{aligned} B &= BUAB = VCAB, \\ C &= CAVC = CABU, \end{aligned} \tag{24}$$

which confirms (v).

(v) ⇒ (iv). Let $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times l}$ such that $CABU = C$ and $VCAB = B$. Then

$$\begin{aligned} VC &= VCABU = BU, \\ B &= VCAB = BUAB, \\ C &= CABU = CAVC, \end{aligned} \tag{25}$$

and hence (iv) holds.

(iv) ⇒ (i). Let $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times l}$ such that $BUAB = B$, $CAVC = C$, and $BU = VC$, and set $X = BU = VC$. Then

$$XAX = BUABU = BU = X; \tag{26}$$

by $X = BU$ and $B = BUAB = XAB$ it follows that $\mathscr{R}(X) = \mathscr{R}(B)$, and by $C = CAVC = CAX$ it follows that $\mathscr{N}(X) = \mathscr{N}(C)$. Therefore, (i) is true.

(b) According to the proofs of (i)⇒(ii) and (iv)⇒(i) and the fact that $C = CABUC$ and $BUCAB = B$, for $U \in \mathbb{C}^{k\times l}$, imply $U \in (CAB)\{1\}$, it follows that

$$A^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)} = BUC = B(CAB)^{(1)}C, \tag{27}$$

and hence (14) holds. □

*Remark 7.* After a comparison of Theorem 6 with the Urquhart formula given in Proposition 2, it is evident that conditions (vi) and (vii) of Theorem 6 could be derived using the Urquhart results. All other conditions are based on the solutions of certain matrix equations, and they are new.

In addition, comparing the representations of Theorem 6 with the full-rank representation restated from [3] in Proposition 1, it is remarkable that the representations given in Theorem 6 do not require computation of a full-rank factorization $R = FG$ of the matrix $R$. More precisely, representations of $A^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)}$ from Theorem 6 boil down to the full-rank factorization of $A^{(2)}_{\mathscr{R}(F),\mathscr{N}(G)}$ from Proposition 1 in the case when $BC = R$ is a full-rank factorization of $R$ and $CAB$ is invertible.

It is worth mentioning that Drazin in [15] generalized the concept of the outer inverse with the prescribed range and null space by introducing the concept of a $(b,c)$-inverse in a semigroup. In the matrix case, this concept can be defined as follows. Let $A \in \mathbb{C}^{m\times n}$, $X \in \mathbb{C}^{n\times m}$, $B \in \mathbb{C}^{n\times k}$, and $C \in \mathbb{C}^{l\times m}$. Then, we call $X$ a $(B,C)$-inverse of $A$ if the following two relations hold:

$$\begin{aligned} &XAB = B, \\ &CAX = C \end{aligned} \tag{28}$$

$$X = BU = VC, \quad \text{for some } U \in \mathbb{C}^{k\times m},\ V \in \mathbb{C}^{n\times l}. \tag{29}$$

It is easy to see that $X$ is a $(B,C)$-inverse of $A$ if and only if $X$ is a $\{2\}$-inverse of $A$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$ and $\mathscr{N}(X) = \mathscr{N}(C)$.

The next theorem can be used for computing a $\{1\}$-inverse $X$ of $A$ satisfying $\mathscr{R}(X) \subseteq \mathscr{R}(B)$.

**Theorem 8.** *Let $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times k}$.*

(a) *The following statements are equivalent:*

(i) *There exists a $\{1\}$-inverse $X$ of $A$ satisfying $\mathscr{R}(X) \subseteq \mathscr{R}(B)$.*

(ii) *There exists $U \in \mathbb{C}^{k\times m}$ such that $ABUA = A$.*

(iii) $\mathscr{R}(AB) = \mathscr{R}(A)$.

(iv) $AB(AB)^{(1)}A = A$*, for some (equivalently every) $(AB)^{(1)} \in (AB)\{1\}$.*

(v) $\operatorname{rank}(AB) = \operatorname{rank}(A)$.

(b) *If the statements in (a) are true, then the set of all inner inverses of $A$ whose range is contained in $\mathscr{R}(B)$ is represented by*

$$\begin{aligned} &\{X \in A\{1\} \mid \mathscr{R}(X) \subseteq \mathscr{R}(B)\} \\ &\quad = \left\{B(AB)^{(1)} \mid (AB)^{(1)} \in (AB)\{1\}\right\} \\ &\quad = \left\{BU \mid U \in \mathbb{C}^{k\times m},\ ABUA = A\right\}. \end{aligned} \tag{30}$$

*Moreover,*

$$\begin{aligned} \{X \in A\{1\} \mid \mathscr{R}(X) \subseteq \mathscr{R}(B)\} = \big\{&B(AB)^{(1)}AA^{(1)} \\ &+ BY - B(AB)^{(1)}ABYAA^{(1)} \mid Y \in \mathbb{C}^{k\times m}\big\}, \end{aligned} \tag{31}$$

*where $(AB)^{(1)} \in (AB)\{1\}$ and $A^{(1)} \in A\{1\}$ are arbitrary but fixed.*

*Proof.* (a) (i) ⇒ (ii). Let $X \in \mathbb{C}^{n\times m}$ such that $AXA = A$ and $\mathscr{R}(X) \subseteq \mathscr{R}(B)$. Then $X = BU$, for some $U \in \mathbb{C}^{k\times m}$, so $A = AXA = ABUA$.

(ii) ⇒ (iii). Let $ABUA = A$, for some $U \in \mathbb{C}^{k\times m}$. Then $\mathscr{R}(A) = \mathscr{R}(ABUA) \subseteq \mathscr{R}(AB)$. Since the opposite inclusion always holds, we conclude that $\mathscr{R}(AB) = \mathscr{R}(A)$.

(iii) ⇒ (iv). Let $(AB)^{(1)}$ be an arbitrary $\{1\}$-inverse of $AB$. By $\mathscr{R}(AB) = \mathscr{R}(A)$ it follows that $A = ABV$, for some $V \in \mathbb{C}^{k\times n}$, so we have that

$$A = ABV = AB(AB)^{(1)}ABV = AB(AB)^{(1)}A. \tag{32}$$

(iv) ⇒ (i). Let $AB(AB)^{(1)}A = A$, for some $(AB)^{(1)} \in (AB)\{1\}$, and set $X = B(AB)^{(1)}$. It is clear that $AXA = A$, and by $X = B(AB)^{(1)}$ we obtain the fact that $\mathscr{R}(X) \subseteq \mathscr{R}(B)$.

(iii) ⇔ (v). This follows from [2, Theorem 1.1.3, P. 3].

(b) On the basis of the fact that $A = ABUA$ implies $U \in (AB)\{1\}$ and the arguments used in the proofs of (i) ⇒ (ii) and (iv) ⇒ (i), we have that

$$\begin{aligned}\{X \in A\{1\} \mid \mathcal{R}(X) \subseteq \mathcal{R}(B)\} &\subseteq \left\{BU \mid U \in \mathbb{C}^{k\times m}, ABUA = A\right\} \\ &\subseteq \left\{B(AB)^{(1)} \mid (AB)^{(1)} \in (AB)\{1\}\right\} \\ &\subseteq \{X \in A\{1\} \mid \mathcal{R}(X) \subseteq \mathcal{R}(B)\},\end{aligned} \tag{33}$$

which confirms that (30) is true.

Once again, according to Theorem 1 [1, Section 2] (or Theorem 1.2.5 [2]) we have that

$$\left\{U \in \mathbb{C}^{k\times m} \mid ABUA = A\right\} = \left\{(AB)^{(1)} AA^{(1)} + Y - (AB)^{(1)} ABYAA^{(1)} \mid Y \in \mathbb{C}^{k\times m}\right\}, \tag{34}$$

where $(AB)^{(1)} \in (AB)\{1\}$ and $(A)^{(1)} \in A\{1\}$ are arbitrary elements, whence we obtain that

$$\begin{aligned}\{X \in A\{1\} \mid \mathcal{R}(X) \subseteq \mathcal{R}(B)\} &= \left\{BU \mid U \in \mathbb{C}^{k\times m},\ ABUA = A\right\} = \left\{B(AB)^{(1)}AA^{(1)} + BY \right. \\ &\left. - B(AB)^{(1)} ABYAA^{(1)} \mid Y \in \mathbb{C}^{k\times m}\right\},\end{aligned} \tag{35}$$

and hence (31) is true. □

Theorem 9 can be used for computing a {1}-inverse $X$ of $A$ satisfying $\mathcal{N}(C) \subseteq \mathcal{N}(X)$. Its proof is dual to the proof of Theorem 8.

**Theorem 9.** *Let $A \in \mathbb{C}^{m\times n}$ and $C \in \mathbb{C}^{l\times m}$.*

(a) *The following statements are equivalent:*

(i) *There exists a {1}-inverse $X$ of $A$ satisfying $\mathcal{N}(C) \subseteq \mathcal{N}(X)$.*
(ii) *There exists $V \in \mathbb{C}^{n\times l}$ such that $AVCA = A$.*
(iii) *$\mathcal{N}(CA) = \mathcal{N}(A)$.*
(iv) *$A(CA)^{(1)}CA = A$, for some (equivalently every) $(CA)^{(1)} \in (CA)\{1\}$.*
(v) $\operatorname{rank}(CA) = \operatorname{rank}(A)$.

(b) *If the statements in (a) are true, then the set of all inner inverses of $A$ whose null space is contained in $\mathcal{N}(C)$ is represented by*

$$\begin{aligned}\{X \in A\{1\} \mid \mathcal{N}(C) \subseteq \mathcal{N}(X)\} &= \left\{(CA)^{(1)} C \mid (CA)^{(1)} \in (CA)\{1\}\right\} \\ &= \left\{VC \mid V \in \mathbb{C}^{n\times l},\ AVCA = A\right\}.\end{aligned} \tag{36}$$

*Moreover,*

$$\{X \in A\{1\} \mid \mathcal{N}(C) \subseteq \mathcal{N}(X)\} = \left\{A^{(1)}A(CA)^{(1)}C + YC - A^{(1)}AYCA(CA)^{(1)}C \mid Y \in \mathbb{C}^{n\times l}\right\}, \tag{37}$$

*where $(CA)^{(1)} \in (CA)\{1\}$ and $A^{(1)} \in A\{1\}$ are arbitrary but fixed.*

Theorem 10 provides several equivalent conditions for the existence and representations for computing a {1, 2}-inverse with the prescribed range.

**Theorem 10.** *Let $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times k}$.*

(a) *The following statements are equivalent:*

(i) *There exists a {1, 2}-inverse $X$ of $A$ satisfying $\mathcal{R}(X) = \mathcal{R}(B)$, denoted by $A^{(1,2)}_{\mathcal{R}(B),*}$.*
(ii) *There exist $U, V \in \mathbb{C}^{k\times m}$ such that $BUAB = B$ and $ABVA = A$.*
(iii) *There exists $W \in \mathbb{C}^{k\times m}$ such that $BWAB = B$ and $ABWA = A$.*
(iv) *$\mathcal{N}(AB) = \mathcal{N}(B)$ and $\mathcal{R}(AB) = \mathcal{R}(A)$.*
(v) $\operatorname{rank}(AB) = \operatorname{rank}(A) = \operatorname{rank}(B)$.
(vi) *$B(AB)^{(1)}AB = B$ and $AB(AB)^{(1)}A = A$, for some (equivalently every) $(AB)^{(1)} \in (AB)\{1\}$.*

(b) *If the statements in (a) are true, then the set of all {1, 2}-inverses with the prescribed range $\mathcal{R}(B)$ is represented by*

$$\begin{aligned}A\{1,2\}_{\mathcal{R}(B),*} &= A\{2\}_{\mathcal{R}(B),*} \\ &= \{X \in A\{1\} \mid \mathcal{R}(X) \subseteq \mathcal{R}(B)\}.\end{aligned} \tag{38}$$

*Proof.* (a) First we note that the implication (i) ⇒ (vi) and the equivalences (ii) ⇔ (iv) and (iv) ⇔ (vi) follow directly from Theorems 3 and 8. Also, (iv) ⇔ (v) follows from Theorem 1.1.3 [2] (or Example 10 [1, Section 1]).

(vi) ⇒ (iii). If we set $W = (AB)^{(1)}$, where $(AB)^{(1)} \in (AB)\{1\}$ is an arbitrary element, then (vi) implies that $BWAB = B$ and $ABWA = A$.

(iii) ⇒ (i). If $W \in \mathbb{C}^{k\times m}$ such that $BWAB = B$ and $ABWA = A$, then by Theorem 3 we obtain the fact that $X = BW$ is a {2}-inverse of $A$ satisfying $\mathcal{R}(X) = \mathcal{R}(B)$, and clearly $X$ is also a {1}-inverse of $A$.

(iii) ⇒ (ii). This implication is evident.

(b) If the statements in (a) hold, then the statements of Theorems 3 and 8 also hold, and from these two theorems it follows directly that (38) is valid. □

Theorem 11 provides several equivalent conditions for the existence and representations of $A^{(1,2)}_{*,\mathcal{N}(C)}$.

**Theorem 11.** *Let $A \in \mathbb{C}^{m\times n}$ and $C \in \mathbb{C}^{l\times m}$.*

(a) *The following statements are equivalent:*

(i) *There exists a {1, 2}-inverse $X$ of $A$ satisfying $\mathcal{N}(X) = \mathcal{N}(C)$, denoted by $A^{(1,2)}_{*,\mathcal{N}(C)}$.*
(ii) *There exist $U, V \in \mathbb{C}^{n\times l}$ such that $CAUC = C$ and $AVCA = A$.*

(iii) *There exists* $W \in \mathbb{C}^{n\times l}$ *such that* $CAWC = C$ *and* $AWCA = A$.
(iv) $\mathcal{N}(CA) = \mathcal{N}(A)$ *and* $\mathcal{R}(CA) = \mathcal{R}(C)$.
(v) $\text{rank}(CA) = \text{rank}(A) = \text{rank}(C)$.
(vi) $CA(CA)^{(1)}C = C$ *and* $A(CA)^{(1)}CA = A$, *for some (equivalently every)* $(CA)^{(1)} \in (CA)\{1\}$.

(b) *If the statements in (a) are true, then the set of all* $\{1,2\}$*-inverses with the range* $\mathcal{R}(B)$ *is given by*

$$\begin{aligned} A\{1,2\}_{*,\mathcal{N}(C)} &= A\{2\}_{*,\mathcal{N}(C)} \\ &= \{X \in A\{1\} \mid \mathcal{N}(C) \subseteq \mathcal{N}(X)\}. \end{aligned} \tag{39}$$

Theorem 12 is a theoretical basis for computing a $\{1,2\}$-inverse with the predefined range and null space.

**Theorem 12.** *Let* $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times k}$, *and* $C \in \mathbb{C}^{l\times m}$.

(a) *The following statements are equivalent:*

(i) *There exists a* $\{1,2\}$*-inverse* $X$ *of* $A$ *satisfying* $\mathcal{R}(X) = \mathcal{R}(B)$ *and* $\mathcal{N}(X) = \mathcal{N}(C)$, *denoted by* $A^{(1,2)}_{\mathcal{R}(B),\mathcal{N}(C)}$.
(ii) *There exist* $U \in \mathbb{C}^{k\times m}$ *and* $V \in \mathbb{C}^{n\times l}$ *such that* $BUAB = B$, $ABUA = A$, $CAVC = C$, *and* $AVCA = A$.
(iii) $\mathcal{N}(AB) = \mathcal{N}(B)$, $\mathcal{R}(AB) = \mathcal{R}(A)$, $\mathcal{R}(CA) = \mathcal{R}(C)$, *and* $\mathcal{N}(CA) = \mathcal{N}(A)$.
(iv) $\text{rank}(AB) = \text{rank}(A) = \text{rank}(B)$, $\text{rank}(CA) = \text{rank}(A) = \text{rank}(C)$.
(v) $\text{rank}(CAB) = \text{rank}(C) = \text{rank}(B) = \text{rank}(A)$.
(vi) $B(AB)^{(1)}AB = B$, $AB(AB)^{(1)}A = A$, $CA(CA)^{(1)}C = C$, *and* $A(CA)^{(1)}CA = A$, *for some (equivalently every)* $(AB)^{(1)} \in (AB)\{1\}$ *and* $(CA)^{(1)} \in (CA)\{1\}$.

(b) *If the statements in (a) are true, then the unique* $\{1,2\}$*-inverse of* $A$ *with the prescribed range* $\mathcal{R}(B)$ *and null space* $\mathcal{N}(C)$ *is represented by*

$$\begin{aligned} A^{(1,2)}_{\mathcal{R}(B),\mathcal{N}(C)} &= B(AB)^{(1)}A(CA)^{(1)}C = BUAVC \\ &= B(CAB)^{(1)}C, \end{aligned} \tag{40}$$

*for arbitrary* $(AB)^{(1)} \in (AB)\{1\}$, $(CA)^{(1)} \in (CA)\{1\}$, *and* $(CAB)^{(1)} \in (CAB)\{1\}$ *and arbitrary* $U \in \mathbb{C}^{k\times m}$ *and* $V \in \mathbb{C}^{n\times l}$ *satisfying* $BUAB = B$ *and* $CAVC = C$.

*Proof.* (a) The equivalence of the statements (i)–(iv) and (vi) follows immediately from Theorem 10 and its dual. The equivalence (i) ⇔ (v) follows immediately from part (4) of the famous Urquhart formula [2, Theorem 1.3.7].

(b) Let $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times l}$ be arbitrary matrices satisfying $BUAB = B$ and $CAVC = C$, and set $X = BUAVC$. Seeing that $U \in (AB)\{1\}$ and $V \in (CA)\{1\}$, according to (v) we obtain the fact that $ABUA = A$ and $AVCA = A$. This implies that

$$\begin{aligned} XAX &= BUAVCABUAVC = BUAVC = X, \\ AXA &= ABUAVCA = AVCA = A, \\ \mathcal{R}(X) &= \mathcal{R}(BUAVC) \subseteq \mathcal{R}(B), \\ \mathcal{N}(C) &\subseteq \mathcal{N}(BUAVC) = \mathcal{N}(X), \\ \mathcal{R}(B) &= \mathcal{R}(BUAB) = \mathcal{R}(BUAVCAB) = \mathcal{R}(XAB) \\ &\subseteq \mathcal{R}(X), \\ \mathcal{N}(X) &\subseteq N(CAX) = \mathcal{N}(CABUAVC) = \mathcal{N}(CAVC) \\ &= \mathcal{N}(C), \end{aligned} \tag{41}$$

which means that $X$ is a $\{1,2\}$-inverse of $A$ satisfying $\mathcal{R}(X) = \mathcal{R}(B)$ and $\mathcal{N}(X) = \mathcal{N}(C)$, and hence the second equality in (40) is true.

The same arguments confirm the validity of the first equality in (40). □

**Corollary 13.** *Theorem 6 is equivalent to Theorem 12 in the case* $\text{rank}(CAB) = \text{rank}(B) = \text{rank}(C) = \text{rank}(A)$.

*Proof.* According to assumptions, the output of Theorem 6 becomes $A^{(1,2)}_{\mathcal{R}(B),\mathcal{N}(C)}$. Then the proof follows from the uniqueness of this kind of generalized inverses. □

*Remark 14.* It is evident that only conditions (v) of Theorem 12 can be derived from the Urquhart results. All other conditions are based on the solutions of certain matrix equations and they are introduced in Theorem 12. Also, the first two representations in (40) are introduced in the present research.

## 3. Algorithms and Implementation Details

The representations presented in Section 2 provide two different frameworks for computing generalized inverses. The first approach arises from the direct computation of various generalizations or certain variants of the Urquhart formula, derived in Section 2. The second approach enables computation of generalized inverses by means of solving certain matrix equations.

The dynamical-system approach is one of the most important parallel tools for solving various basic linear algebra problems. Also, Zhang neural networks (ZNN) as well as gradient neural networks (GNN) have been simulated for finding a real-time solution of linear time-varying matrix equation $AXB = C$. Simulation results confirm the efficiency of the ZNN and GNN approach in solving both time-varying and time-invariant linear matrix equations. We refer to [28, 29] for further details. In the case of constant coefficient matrices $A, B, C$, it is necessary to use the linear GNN of the form

$$\dot{X} = -\gamma A^T (AXB - C) B^T. \tag{42}$$

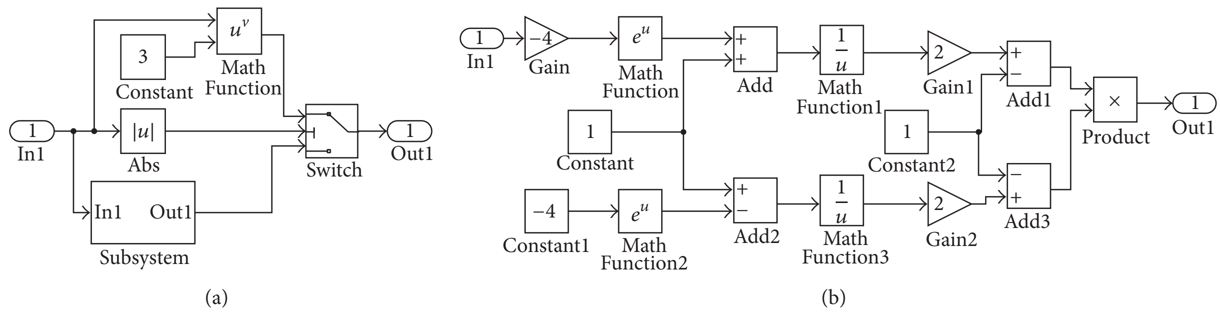


Figure 1: Block for the implementation of the power-sigmoid activation function (a) and its subsystem (b).

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$ and $B(t) \in \mathbb{C}^{n\times k}$.
(1) Verify $\operatorname{rank}(A(t)B(t)) = \operatorname{rank}(B(t))$.
If these conditions are satisfied then continue.
(2) Solve the matrix equation $B(t)U(t)A(t)B(t) = B(t)$ with respect to $U(t) \in \mathbb{C}^{k\times m}$.
(3) Return $X(t) = B(t)U(t) = A(t)^{(2)}_{\mathscr{R}(B),*}$.

Algorithm 1: Computing an outer inverse with the prescribed range.

The generalized nonlinearly activated GNN model (GGNN model) is applicable in both time-varying and time-invariant case and possesses the form

$$\dot{X}(t) = -\gamma A(t)^{T} \mathscr{F}(A(t)X(t)B(t) - C(t))B(t)^{T}, \quad (43)$$

where $\mathscr{F}(C)$ is an odd and monotonically increasing function element-wise applicable to elements of a real matrix $C = (c_{kj}) \in \mathbb{R}^{n\times m}$; that is, $\mathscr{F}(C) = (f(c_{kj}))$, wherein $f(\cdot)$ is an odd and monotonically increasing function. Also, the scaling parameter $\gamma$ could be chosen as large as possible in order to accelerate the convergence. The convergence could be proved only for the situation with constant coefficient matrices $A, B, C$.

Besides the linear activation function, $f(x) = x$, in the present paper we use the power-sigmoid activation function

$$f(x) = \begin{cases} x^{p}, & \text{if } |x| \geq 1, \\ \dfrac{1+\exp(-q)}{1-\exp(-q)}\dfrac{1-\exp(-qx)}{1+\exp(-qx)}, & \text{otherwise}, \end{cases} \quad q \geq 2, p \geq 3. \quad (44)$$

Theorem 3 provides not only criteria for the existence of an outer inverse $A(t)^{(2)}_{\mathscr{R}(B),*}$ with the prescribed range, but also a method for computing such an inverse. Namely, the problem of computing a $\{2\}$-inverse $X$ of $A$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$ boils down to the problem of computing a solution to the matrix equation $BUAB = B$, where $U$ is an unknown matrix taking values in $\mathbb{C}^{k\times m}$. If $U$ is an arbitrary solution to this equation, then a $\{2\}$-inverse $X$ of $A$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$ can be computed as $X = BU$.

The Simulink implementation of Algorithm 1 in the set of real matrices is based on GGNN model (43) for solving the matrix equation $B(t)U(t)A(t)B(t) = B(t)$ and it is presented in Figure 5. The Simulink Scope and Display Block denoted by $U(t)$ display input signals corresponding to the solution $U(t)$ of the matrix equation $B(t)U(t)A(t)B(t) = B(t)$ with respect to the time $t$. The underlying GGNN model in Figure 5 is

$$\dot{U}(t) = -\gamma B(t)^{\mathrm{T}} \mathscr{F}(B(t)U(t)A(t)B(t) - B(t)) \cdot (A(t)B(t))^{\mathrm{T}}. \quad (45)$$

The Display Block denoted by $BU$ displays inputs signals corresponding to the solution $X(t) = B(t)U(t)$.

The block subsystem implements the power-sigmoid activation function and it is presented in Figure 1.

Theorem 5 reduces the problem of computing a $\{2\}$-inverse $X$ of $A$ satisfying $\mathscr{N}(X) = \mathscr{N}(B)$ to the problem of computing a solution to the matrix equation $CAVC = C$, where $V$ is an unknown matrix taking values in $\mathbb{C}^{n\times l}$. Then $X := A^{(2)}_{*,\mathscr{N}(C)} = VC$.

The Simulink implementation of Algorithm 2 which is based on the GGNN model for solving $C(t)A(t)V(t)C(t) = C(t)$ and computing $X(t) = V(t)C(t)$ is presented in Figure 6. The underlying GGNN model in Figure 6 is

$$\dot{V}(t) = -\gamma (C(t)A(t))^{\mathrm{T}} \cdot \mathscr{F}(C(t)A(t)V(t)C(t) - C(t))C(t)^{\mathrm{T}}. \quad (46)$$

The *Display* Block denoted by $V(t)$ displays input signals corresponding to the solution $V(t)$ of the matrix equation $CAV(t)C = C$ with respect to simulation time. The *Display* Block denoted by $ATS2$ displays input signals corresponding to the solution $X(t) = V(t)C(t)$.

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$ and $C(t) \in \mathbb{C}^{l\times m}$.
(1) Verify $\operatorname{rank}(C(t)A(t)) = \operatorname{rank}(C(t))$.
If these conditions are satisfied then continue.
(2) Solve the matrix equation $C(t)A(t)V(t)C(t) = C(t)$ with respect to an unknown matrix $V(t) \in \mathbb{C}^{n\times l}$.
(3) Return $X(t) = V(t)C(t) = A(t)^{(2)}_{*,\mathcal{N}(C)}$.

Algorithm 2: Computing an outer inverse with the prescribed null space.

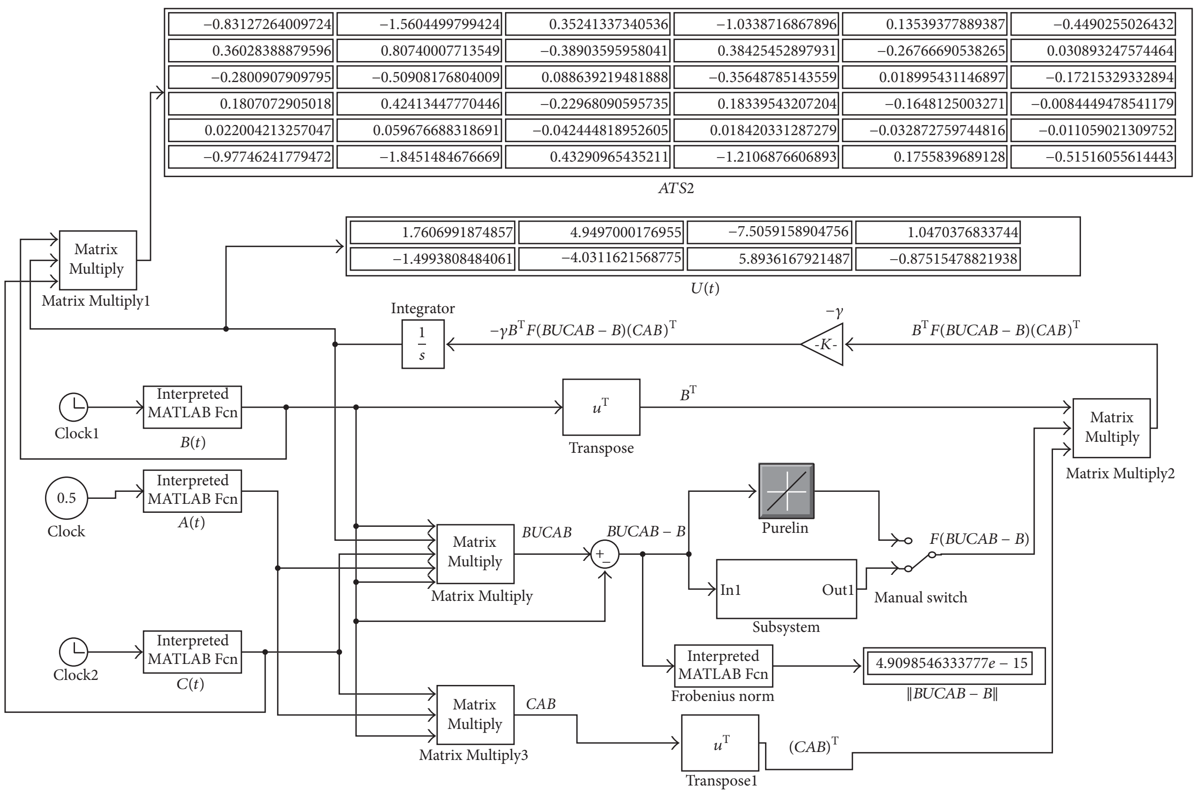


Figure 2: GGNN model for computing $B(t)U(t)C(t)A(t)B(t) = B(t)$, $X(t) = B(t)U(t)C(t)$.

Theorem 6 provides a powerful representation of a $\{2\}$-inverse $X$ of $A$ satisfying $\mathcal{R}(X) = \mathcal{R}(B)$ and $\mathcal{N}(X) = \mathcal{N}(C)$. Also, it suggests the following procedure for computing those generalized inverses. First, it is necessary to verify whether $\operatorname{rank}(CAB) = \operatorname{rank}(B) = \operatorname{rank}(C)$. If this is true, then by Theorem 6 it follows that the equations $BUCAB = B$ and $CABVC = C$ are solvable and have the same sets of solutions. We compute an arbitrary solution $U$ of the equation $BUCAB = B$, and then $X = BUC$ is the desired $\{2\}$-inverse of $A$.

The Simulink implementation of the GGNN model for solving $B(t)U(t)C(t)A(t)B(t) = B(t)$ and computing the outer inverse $X(t) = B(t)U(t)C(t)$ defined in Algorithm 3 is presented in Figure 2. The underlying GGNN model in Figure 2 is

$$\dot{U}(t) = -\gamma B(t)^{\mathrm{T}} \cdot \mathcal{F}\left(B(t)U(t)C(t)A(t)B(t) - B(t)\right) \cdot \left(C(t)A(t)B(t)\right)^{\mathrm{T}}. \quad (47)$$

The implementation of the dual approach, based on the solution of $C(t)A(t)BV(t)C(t) = C(t)$ and generating the outer inverse $X(t) = B(t)V(t)C(t)$, is presented in Figure 4. The underlying GGNN model in Figure 4 is

$$\dot{V}(t) = -\gamma\left(C(t)A(t)B(t)\right)^{\mathrm{T}}\mathcal{F}\left(C(t)A(t)B(t)V(t) \cdot C(t) - C(t)\right)C(t)^{\mathrm{T}}. \quad (48)$$

Theorem 8 can be used in a similar way to Theorem 3: if the equation $ABUA = A$ is solvable and its solution $U$ is computed, then a $\{1\}$-inverse $X$ of $A$ satisfying $\mathcal{R}(X) \subseteq \mathcal{R}(B)$ is computed as $X = BU$. Corresponding computational procedure is given in Algorithm 4.

Similarly, Theorem 9 can be used for computing a $\{1\}$-inverse $X$ of $A$ satisfying $\mathcal{N}(C) \subseteq \mathcal{N}(X)$, as it is presented in Algorithm 5.

An algorithm for computing a $\{1,2\}$-inverse with the prescribed range is based on Theorem 10. According to this

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$, $B(t) \in \mathbb{C}^{n\times k}$ and $C(t) \in \mathbb{C}^{l\times m}$.
(1) Verify $\operatorname{rank}(C(t)A(t)B(t)) = \operatorname{rank}(B(t)) = \operatorname{rank}(C(t))$.
If these conditions are satisfied then continue.
(2) Solve the matrix equation $B(t)U(t)C(t)A(t)B(t) = B(t)$ with respect to an unknown matrix $U(t) \in \mathbb{C}^{k\times m}$.
(3) Return $X(t) = B(t)U(t)C(t) = A(t)^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)}$.

Algorithm 3: Computing a {2}-inverse with the prescribed range and null space.

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$ and $B(t) \in \mathbb{C}^{n\times k}$.
(1) Check the condition $\operatorname{rank}(A(t)B(t)) = \operatorname{rank}(A(t))$.
If this condition is satisfied then continue.
(2) Solve the matrix equation $A(t)B(t)U(t)A(t) = A(t)$ with respect to $U(t) \in \mathbb{C}^{k\times m}$.
(3) Return a {1}-inverse $X(t) = B(t)U(t)$ of $A(t)$ satisfying $\mathscr{R}(X) \subseteq \mathscr{R}(B)$.

Algorithm 4: Computing a {1}-inverse $X$ of $A$ satisfying $\mathscr{R}(X) \subseteq \mathscr{R}(B)$.

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$ and $C(t) \in \mathbb{C}^{l\times m}$.
(1) Check the condition $\operatorname{rank}(C(t)A(t)) = \operatorname{rank}(A(t))$.
If this condition is satisfied then continue.
(2) Solve the matrix equation $A(t)V(t)C(t)A(t) = A(t)$ with respect to an unknown matrix $V(t) \in \mathbb{C}^{n\times l}$.
(3) Return a {1}-inverse $X(t) = V(t)C(t)$ of $A(t)$ satisfying $\mathscr{N}(C) \subseteq \mathscr{N}(X)$.

Algorithm 5: Computing a {1}-inverse $X$ of $A$ satisfying $\mathscr{N}(C) \subseteq \mathscr{N}(X)$.

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$ and $B(t) \in \mathbb{C}^{n\times k}$.
(1) Check the condition $\operatorname{rank}(A(t)B(t)) = \operatorname{rank}(A(t)) = \operatorname{rank}(B(t))$.
If these conditions are satisfied then continue.
(2) If the previous condition is satisfied, then solve the matrix equation $B(t)U(t)A(t)B(t) = B(t)$ with respect to an unknown matrix $U(t) \in \mathbb{C}^{k\times m}$.
(3) Return a {1, 2}-inverse $X(t) = B(t)U(t)$ of $A(t)$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$.

Algorithm 6: Computing a {1, 2}-inverse with the prescribed range.

theorem we first check the condition $\operatorname{rank}(AB) = \operatorname{rank}(A) = \operatorname{rank}(B)$. If it is satisfied, then the equation $BUAB = B$ is solvable and we compute an arbitrary solution $U$ to this equation, after which we compute a {2}-inverse $X$ of $A$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$ as $X = BU$. By Theorem 10, $X$ is also a {1}-inverse of $A$. Algorithm 1 differs from Algorithm 6 only in the first step. Therefore, the implementation of Algorithm 6 uses the Simulink implementation of Algorithm 1 in the case when $\operatorname{rank}(AB) = \operatorname{rank}(A) = \operatorname{rank}(B)$.

Similarly, Theorem 11 provides an algorithm for computing $A^{(1,2)}_{*,\mathscr{N}(C)}$. The implementation of Algorithm 7 uses the Simulink implementation of Algorithm 2 in the case $\operatorname{rank}(CA) = \operatorname{rank}(C) = \operatorname{rank}(A)$.

Theorem 12 suggests the following procedure for computing a {1, 2}-inverse $X$ of $A$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$ and $\mathscr{N}(X) = \mathscr{N}(C)$. First we check the condition $\operatorname{rank}(CAB) = \operatorname{rank}(B) = \operatorname{rank}(C) = \operatorname{rank}(A)$. If this is true, then the equations $BUAB = B$ and $CAVC = C$ are solvable, and we compute an arbitrary solution $U$ to the first one and an arbitrary solution $V$ of the second one. According to Theorem 12, $X = BUAVC$ is a {1, 2}-inverse $X$ of $A$ with $\mathscr{R}(X) = \mathscr{R}(B)$ and $\mathscr{N}(X) = \mathscr{N}(C)$.

The Simulink implementation of Algorithm 8 based on the GGNN models for solving $B(t)U(t)A(t)B(t) = B(t)$ and $C(t)A(t)V(t)C(t) = C(t)$ and computing $X(t) = B(t)U(t)A(t)V(t)C(t)$ is presented in Figure 8. In this case, it is necessary to implement two parallel GGNN models of the form

$$\begin{aligned} \dot{U}(t) &= -\gamma B(t)^{\mathrm{T}} \mathscr{F}(B(t)U(t)A(t)B(t) - B(t)) \\ &\quad \cdot (A(t)B(t))^{\mathrm{T}}, \\ \dot{V}(t) &= -\gamma (C(t)A(t))^{\mathrm{T}} \\ &\quad \cdot \mathscr{F}(C(t)A(t)V(t)C(t) - C(t))C(t)^{\mathrm{T}}. \end{aligned} \tag{49}$$

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$ and $C(t) \in \mathbb{C}^{l\times m}$.
(1) Check the condition $\operatorname{rank}(C(t)A(t)) = \operatorname{rank}(A(t)) = \operatorname{rank}(C(t))$.
If these conditions are satisfied then continue.
(2) Solve the matrix equation $C(t)A(t)W(t)C(t) = C(t)$ with respect to an unknown matrix $W(t) \in \mathbb{C}^{l\times m}$.
(3) Return a $\{1,2\}$-inverse $X(t) = V(t)C(t)$ of $A(t)$ satisfying $\mathcal{N}(X) = \mathcal{N}(C)$.

Algorithm 7: Computing a $\{1,2\}$-inverse with the prescribed null space.

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$, $B(t) \in \mathbb{C}^{n\times k}$ and $C(t) \in \mathbb{C}^{l\times m}$.
*Require*: Verify $\operatorname{rank}(C(t)A(t)B(t)) = \operatorname{rank}(B(t)) = \operatorname{rank}(C(t)) = \operatorname{rank}(A(t))$.
If these conditions are satisfied then continue.
(1) Solve the matrix equation $B(t)U(t)A(t)B(t) = B(t)$ with respect to an unknown matrix $U(t) \in \mathbb{C}^{k\times m}$.
(2) Solve the matrix equation $C(t)A(t)V(t)C(t) = C(t)$ with respect to an unknown matrix $V(t) \in \mathbb{C}^{n\times l}$.
(3) Return $X(t) = B(t)U(t)A(t)V(t)C(t) = A(t)^{(1,2)}_{\mathcal{R}(B),\mathcal{N}(C)}$.

Algorithm 8: Computing a $\{1,2\}$-inverse with the prescribed range and null space.

There is also an alternative way to compute a $\{1,2\}$-inverse $X$ of $A$ with $\mathcal{R}(X) = \mathcal{R}(B)$ and $\mathcal{N}(X) = \mathcal{N}(C)$. Namely, first we check whether $\operatorname{rank}(CAB) = \operatorname{rank}(B) = \operatorname{rank}(C) = \operatorname{rank}(A)$. If this is true, then by Theorem 12 it follows that there exists a $\{2\}$-inverse of $A$ with the prescribed range $\mathcal{R}(B)$ and null space $\mathcal{N}(C)$, and each such inverse is also a $\{1\}$-inverse of $A$. Therefore, to compute a $\{1,2\}$-inverse of $A$ having the range $\mathcal{R}(B)$ and null space $\mathcal{N}(C)$ we have to compute a $\{2\}$-inverse $X$ of $A$ with $\mathcal{R}(X) = \mathcal{R}(B)$ and $\mathcal{N}(X) = \mathcal{N}(C)$ in exactly the same way as in Algorithm 3. In other words, we compute an arbitrary solution $U$ to the equation $BUCAB = B$, and then $X = BUC$ is the desired $\{1,2\}$-inverse of $A$.

*3.1. Complexity of Algorithms.* The general computational pattern for commuting generalized inverses is based on the general representation $B(CAB)^{(1)}C$, where the matrices $A, B, C$ satisfy various conditions imposed in the proposed algorithms.

The first approach is based on the computation of an involved inner inverse $(CAB)^{(1)}$, and it can be described in three main steps:

(1) Compute the matrix product $P = CAB$.
(2) Compute an inner inverse $U = P^{(1)}$ of $P$, for example, $U = P^{\dagger}$.
(3) Compute the generalized inverse as the matrix product $BUC$.

The second general computational pattern for computing generalized inverses can be described in three main steps:

(1) Compute matrix products included in the required linear matrix equation.
(2) Solve the generated matrix equation with respect to the unknown matrix $U$.
(3) Compute the generalized inverse of $A$ as the matrix product which includes $U$.

According to the first approach, the complexity of computing generalized inverses can be estimated as follows:

(1) Complexity of the matrix product $P = CAB$
+(2) Complexity to compute an inner inverse of $P$
+(3) Complexity to compute the matrix product $BUC$

According to the second approach, the complexity of computing generalized inverses can be expressed according to the rule:

(1) Complexity of the matrix product $P$ included in required matrix equation which should be solved.
+(2) Complexity to solve the linear matrix generated in (1)
+(3) Complexity of matrix products required in final representation

Let us compare complexities of two representations from (14). Two possible approaches are available. The first approach assumes computation $A^{(2)}_{\mathcal{R}(B),\mathcal{N}(C)} = B(CAB)^{(1)}C$ and the second one assumes $A^{(2)}_{\mathcal{R}(B),\mathcal{N}(C)} = BUC$, where $BUCAB = B$. Complexity of computing the $B(CAB)^{(1)}C$ is

(1) complexity of the matrix product $P = CAB$,
+(2) complexity of computation of $P^{(1)}$,
+(3) complexity of matrix products required in final representation $BP^{(1)}C$.

Complexity of computing the second expression in (14) is

(1) complexity of matrix products $P = CAB$,
+(2) complexity to solve appropriate linear matrix equation $BUP = B$ with respect to $U$,
+(3) complexity of the matrix product $BUC$.

*3.2. Particular Cases.* The main particular cases of Theorem 6 can be derived directly and listed as follows.

(a) In the case $\mathrm{rank}(CAB) = \mathrm{rank}(B) = \mathrm{rank}(C) = \mathrm{rank}(A)$ the outer inverse $A^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)}$ becomes $A^{(1,2)}_{\mathscr{R}(B),\mathscr{N}(C)}$.

(b) If $A$ is nonsingular and $B = C = I$, then the outer inverse $A^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)}$ becomes the usual inverse $A^{-1}$.

Then the matrix equation $BUCAB = B$ becomes $UA = I$ and $A^{-1} = U$.

(c) In the case $B = C = A^*$ or when $BC = A^*$ is a full-rank factorization of $A^*$, it follows that $A^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)} = A^{\dagger}$.

(d) The choice $m = n$, $B = C = A^l$, $l \geq \mathrm{ind}(A)$, or the full-rank factorization $BC = A^l$ implies $A^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)} = A^{\mathrm{D}}$.

(e) The choice $m = n$, $B = C = A$, or the full-rank factorization $BC = A$ produces $A^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)} = A^{\#}$.

(f) In the case $m = n$ when $A$ is invertible, the inverse matrix $A^{-1}$ can be generated by two choices: $B = C = A^*$ and $B = C = I$.

(g) Theorem 6 and the full-rank representation of $\{2,4\}$- and $\{2,3\}$-inverses from [30] are a theoretical basis for computing $\{2,4\}$- and $\{2,3\}$-inverses with the prescribed range and null space.

(h) Further, Theorems 3 and 5 provide a way to characterize $\{1,2,4\}$- and $\{1,2,3\}$-inverses of a matrix.

**Corollary 15.** *Let $A \in \mathbb{C}^{m\times n}$ and $C \in \mathbb{C}^{l\times m}$.*

(a) *The following statements are equivalent:*

(i) *There exists a $\{2,4\}$-inverse $X$ of $A$ satisfying $\mathscr{R}(X) = \mathscr{R}((CA)^*)$ and $\mathscr{N}(X) = \mathscr{N}(C)$.*

(ii) *There exist $U \in \mathbb{C}^{l\times l}$ such that $(CA)^*UCA(CA)^* = (CA)^*$ and $CA(CA)^*UC = C$.*

(iii) *There exist $U, V \in \mathbb{C}^{l\times l}$ such that $(CA)^*UCA(CA)^* = (CA)^*$ and $CA(CA)^*VC = C$.*

(iv) *There exist $U \in \mathbb{C}^{l\times m}$ and $V \in \mathbb{C}^{n\times l}$ such that $(CA)^*UA(CA)^* = (CA)^*$, $CAVC = C$, and $(CA)^*U = VC$.*

(v) *There exist $U \in \mathbb{C}^{l\times m}$ and $V \in \mathbb{C}^{n\times l}$ such that $CA(CA)^*U = C$ and $VCA(CA)^* = (CA)^*$.*

(vi) $\mathscr{N}(CA(CA)^*) = \mathscr{N}((CA)^*)$, $\mathscr{R}(CA(CA)^*) = \mathscr{R}(C)$.

(vii) $\mathrm{rank}(CA(CA)^*) = \mathrm{rank}((CA)^*) = \mathrm{rank}(C)$.

(viii) $(CA)^*(CA(CA)^*)^{(1)}CA(CA)^* = (CA)^*$ *and* $CA(CA)^*(CA(CA)^*)^{(1)}C = C$, *for some (equivalently every)* $(CA(CA)^*)^{(1)} \in (CA(CA)^*)\{1\}$.

(b) *If the statements in (a) are true, then the unique $\{2,4\}$-inverse of $A$ with the prescribed range $\mathscr{R}((CA)^*)$ and null space $\mathscr{N}(C)$ is represented by*

$$\begin{aligned} A^{(2,4)}_{\mathscr{R}((CA)^*),\mathscr{N}(C)} &= (CA)^* \left(CA\,(CA)^*\right)^{(1)} C \\ &= (CA)^* UC, \end{aligned} \tag{50}$$

*for arbitrary $(CA(CA)^*)^{(1)} \in (CA(CA)^*)\{1\}$ and arbitrary $U \in \mathbb{C}^{l\times l}$ satisfying $(CA)^*UCA(CA)^* = (CA)^*$ and $CA(CA)^*UC = C$.*

*Proof.* (a) This part of the proof is particular case $B = (CA)^*$ of Theorem 6.

(b) According to general representation of outer inverses with prescribed range and null space, it follows that $X \coloneqq (CA)^*(CA(CA)^*)^{(1)}C = A^{(2)}_{\mathscr{R}((CA)^*),\mathscr{N}(C)}$. Now, it suffices to verify that $X$ satisfies Penrose equation (4). For this purpose, it is useful to use known result

$$A\left(A^*A\right)^{(1)} A^* = AA^{\dagger}, \tag{51}$$

which implies

$$\begin{aligned} XA &= (CA)^* \left(CA\,(CA)^*\right)^{(1)} CA = (CA)^* \left((CA)^*\right)^{\dagger} \\ &= (CA)^{\dagger}\, CA \end{aligned} \tag{52}$$

and later $XA = (XA)^*$. Hence, (50) holds. □

**Corollary 16.** *Let $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times k}$.*

(a) *The following statements are equivalent:*

(i) *There exists a $\{2,3\}$-inverse $X$ of $A$ satisfying $\mathscr{R}(X) = \mathscr{R}(B)$ and $\mathscr{N}(X) = \mathscr{N}((AB)^*)$.*

(ii) *There exist $U \in \mathbb{C}^{k\times k}$ such that $BU(AB)^*AB = B$ and $(AB)^*ABU(AB)^* = (AB)^*$.*

(iii) *There exist $U, V \in \mathbb{C}^{k\times k}$ such that $BU(AB)^*AB = B$ and $(AB)^*ABV(AB)^* = (AB)^*$.*

(iv) *There exist $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times k}$ such that $BUAB = B$, $(AB)^*AV(AB)^* = (AB)^*$, and $BU = V(AB)^*$.*

(v) *There exist $U \in \mathbb{C}^{k\times m}$ and $V \in \mathbb{C}^{n\times k}$ such that $(AB)^*ABU = (AB)^*$ and $V(AB)^*AB = B$.*

(vi) $\mathscr{N}((AB)^*AB) = \mathscr{N}(B)$, $\mathscr{R}((AB)^*AB) = \mathscr{R}((AB)^*)$.

(vii) $\mathrm{rank}((AB)^*AB) = \mathrm{rank}(B) = \mathrm{rank}((AB)^*)$.

(viii) $B((AB)^*AB)^{(1)}(AB)^*AB = B$ *and* $(AB)^*AB((AB)^*AB)^{(1)}(AB)^* = (AB)^*$, *for some (equivalently every)* $((AB)^*AB)^{(1)} \in (CAB)\{1\}$.

(b) *If the statements in (a) are true, then the unique $\{2,3\}$-inverse of $A$ with the prescribed range $\mathscr{R}(B)$ and null space $\mathscr{N}((AB)^*)$ is represented by*

$$\begin{aligned} A^{(2,3)}_{\mathscr{R}(B),\mathscr{N}((AB)^*)} &= B\left((AB)^*\,AB\right)^{(1)} (AB)^* \\ &= BU\,(AB)^*, \end{aligned} \tag{53}$$

*for arbitrary* $((AB)^*AB)^{(1)} \in ((AB)^*AB)\{1\}$ *and arbitrary* $U \in \mathbb{C}^{k\times k}$ *satisfying* $BU(AB)^*AB = B$ *and* $(AB)^*ABU(AB)^* = (AB)^*$.

Corollary 17 shows the equivalence between the first representation given in (53) of Corollary 16 and Corollary 1 from [31].

**Corollary 17.** *Let* $A \in \mathbb{C}^{m\times n}$ *and* $B \in \mathbb{C}^{n\times k}$ *satisfy* $\operatorname{rank}(AB) = \operatorname{rank}(B)$. *Then*

$$A^{(2,3)}_{\mathscr{R}(B),\mathscr{R}(AB)^{\perp}} = B(AB)^{(1,3)}. \tag{54}$$

*Proof.* It suffices to verify

$$((AB)^* AB)^{(1)} (AB)^* = (AB)^{(1,3)}. \tag{55}$$

Indeed, since $\operatorname{rank}((AB)^*AB) = \operatorname{rank}(AB)$, it follows that

$$AB((AB)^* AB)^{(1)} (AB)^* = AB. \tag{56}$$

Now, the proof can be completed using the evident fact that $AB((AB)^*AB)^{(1)}(AB)^*$ is the Hermitian matrix. □

In dual case, Corollary 18 is an additional result to Corollary 1 from [31].

**Corollary 18.** *Let* $A \in \mathbb{C}^{m\times n}$ *and* $C \in \mathbb{C}^{l\times m}$ *satisfy* $\operatorname{rank}(CA) = \operatorname{rank}(C)$. *Then*

$$A^{(2,4)}_{\mathscr{N}(CA)^{\perp},\mathscr{N}(C)} = (CA)^{(1,4)} C. \tag{57}$$

*Proof.* In this case, the identity

$$(CA)^* (CA(CA)^*)^{(1)} = (CA)^{(1,4)} \tag{58}$$

can be verified similarly. □

**Theorem 19.** *Let* $A \in \mathbb{C}^{m\times n}$. *Then*

$$\begin{aligned} A\{1,2,4\} &= A\{2\}_{\mathscr{R}(A^*),*} = A\{1,2\}_{\mathscr{R}(A^*),*} \\ &= \{A^*U \mid U \in \mathbb{C}^{m\times m},\ A^*UAA^* = A^*\}. \end{aligned} \tag{59}$$

*Proof.* The equalities

$$\begin{aligned} A\{2\}_{\mathscr{R}(A^*),*} &= A\{1,2\}_{\mathscr{R}(A^*),*} \\ &= \{A^*U \mid U \in \mathbb{C}^{m\times m},\ A^*UAA^* = A^*\} \end{aligned} \tag{60}$$

follow immediately from Theorem 3.

Let $X \in A\{1,2,4\}$, that is, $A = AXA$, $X = XAX$, and $(XA)^* = XA$, and set $U = X^*X$. Then

$$\begin{aligned} X &= XAX = (XA)^* X = A^*X^*X = A^*U, \\ A^*UAA^* &= A^*X^*XAA^* = A^*X^*(XA)^*A^* \\ &= (AXAXA)^* = A^*. \end{aligned} \tag{61}$$

Conversely, let $X = A^*U$ and $A^*UAA^* = A^*$, for some $U \in \mathbb{C}^{m\times m}$. According to (5) we have that $X \in A\{1,2\}$. On the other hand, by $X = A^*U$ and $A^*UAA^* = A^*$ it follows that $XAA^* = A^*$, and it is well-known that it is equivalent to $X \in A\{1,4\}$. Thus, $X \in A\{1,2,4\}$. □

The following theorem can be verified in a similar way.

**Theorem 20.** *Let* $A \in \mathbb{C}^{m\times n}$. *Then*

$$\begin{aligned} A\{1,2,3\} &= A\{2\}_{*,\mathscr{N}(A^*)} = A\{1,2\}_{*,\mathscr{N}(A^*)} \\ &= \{VA^* \mid V \in \mathbb{C}^{n\times n},\ A^*AVA^* = A^*\}. \end{aligned} \tag{62}$$

## 4. Numerical Examples

All numerical experiments are performed starting from the zero initial condition. *MATLAB* and the Simulink version is 8.4 (R2014b).

*Example 21.* Consider

$$\begin{aligned} A &= \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 & 0 \\ -1 & -1 & -1 & 0 & 2 & -1 \\ -1 & -1 & 0 & -1 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{6\times 6}_5, \\ B &= \begin{bmatrix} 0.793372 & 0.265655 \\ 0.140305 & 0.633824 \\ 0.329002 & 0.184927 \\ 0.141169569 & 0.427424 \\ 0.0468532 & 0.0979332 \\ 0.89494969 & 0.253673 \end{bmatrix} \in \mathbb{R}^{6\times 2}_2, \\ C &= \begin{bmatrix} 0.714297 & 0.734462 & 0.790305 & 1.1837035 & 0.850446 & 1.143219 \\ 0.596075 & 0.5652303 & 0.745458 & 1.011021 & 0.785712 & 1.013570 \\ 0.780387 & 0.931596 & 0.630581 & 1.23033 & 0.723199 & 1.0876717 \\ 0.298214 & 0.30235998 & 0.337657 & 0.496275 & 0.361875 & 0.482631 \end{bmatrix} \in \mathbb{R}^{4\times 6}_4. \end{aligned} \tag{63}$$

(a) This part of the example illustrates results of Theorem 6 and it is based on the implementation of Algorithm 3. The matrices $A, B, C$ satisfy $\text{rank}(B) = 2$, $\text{rank}(C) = 4$, and $\text{rank}(CAB) = 2$. Since the conditions in (vii) of Theorem 6 are not satisfied, there is no an exact solution of the system of matrix equations $BUCAB = B$ and $CABUC = C$. The outer inverse $X = B(CAB)^{(1)}C$ can be computed using the RNN approach, as follows. The Simulink implementation of Algorithm 3, which is based on the GGNN model for solving the matrix equation $B(t)U(t)C(t)A(t)B(t) = B(t)$, gives the result which is presented in Figure 2. The display denoted by $U(t)$ denotes an approximate solution of the matrix equation $BU(t)CAB = B$. The time interval is $[0, 0.5]$, the solver is `ode15s`, the power-sigmoid activation is selected, and $\gamma = 10^6$.

*Step 1.* Solve the matrix equation $B(t)U(t)C(t)A(t)B(t) = B(t)$ with respect to $U(t)$ using an appropriate adaptation of the GGNN approach developed in [28, 29] and restated in (43). In the particular case, the model becomes

$$\dot{U}(t) = -\gamma B(t)^{\mathrm{T}} \cdot \mathcal{F}\left(B(t)U(t)C(t)A(t)B(t) - B(t)\right) \cdot \left(C(t)A(t)B(t)\right)^{\mathrm{T}}. \tag{64}$$

The matrix $B$ is of full-column rank, and it possesses the left inverse $B_l^{-1}$. Therefore, the matrix equation $BUCAB = B$ is equivalent to the equation $UCAB - I = 0$. Then the GGNN model (64) reduces to the well-known GNN model for computing the pseudoinverse of $CAB$. The GNN models for computing the pseudoinverse of rank-deficient matrices were introduced and described in [21]. We further confirm the results derived in *MATLAB* Simulink by means of the programming package *Mathematica*. *Mathematica* gives

$$(CAB)^{\dagger} = \begin{bmatrix} 1.76069 & 4.94967 & -7.50589 & 1.04706 \\ -1.49938 & -4.03114 & 5.8936 & -0.875175 \end{bmatrix}, \tag{65}$$

which coincides with the result displayed in $U(t)$ in Figure 2.

*Step 2.* The matrix $X(t) = B(t)U(t)C(t)$ is showed in Figure 2, in the display denoted by $ATS2$. The residual norm of $X$ is equal to $\|XAX - X\|_2 = 6.5360016 * 10^{-15}$.

As a confirmation, *Mathematica* gives

$$X = B(CAB)^{\dagger}C = \begin{bmatrix} -0.83127 & -1.56045 & 0.352412 & -1.03387 & 0.135393 & -0.449024 \\ 0.360282 & 0.807398 & -0.389035 & 0.384252 & -0.267666 & 0.0308925 \\ -0.28009 & -0.509081 & 0.0886387 & -0.356487 & 0.0189951 & -0.172153 \\ 0.180706 & 0.424133 & -0.22968 & 0.183394 & -0.164812 & -0.00844537 \\ 0.0220041 & 0.0596765 & -0.0424447 & 0.0184201 & -0.0328727 & -0.0110591 \\ -0.977459 & -1.84514 & 0.432908 & -1.21068 & 0.175583 & -0.515159 \end{bmatrix}, \tag{66}$$

which coincides with the contents of the *Display* Block denoted as ATS2 in Figure 2. Further, the matrix $U = (CAB)^{\dagger}$ is an approximate solution of the matrix equations $CABUC = C$ and $BUCAB = B$. Also, $X = BUC$ is an approximate solution of (28), since

$$\begin{aligned} \|CABUC - C\| &= \|CAX - C\| = 2.23452290 * 10^{-14}, \\ \|BUCAB - B\| &= \|XAB - B\| = 9.4574123 * 10^{-15}. \end{aligned} \tag{67}$$

Therefore, the equations in (28) are satisfied. In addition, (29) is satisfied by the definition of $X$. Therefore, $X$ is an approximate $(B, C)$-inverse of $A$.

Trajectories of the entries in the matrix $B(t)U(t)C(t)$ generated inside the time $[0, 5 * 10^{-2}]$, using $\gamma = 10^6$ and `ode15s` solver, are presented in Figure 3.

(b) Dual approach in Theorem 6, as well as in the implementation of Algorithm 3, is based on the solution of $C(t)A(t)V(t)C(t) = C(t)$ and the associated outer inverse $X_1(t) = B(t)V(t)C(t)$. The Simulink implementation of the GGNN model which is based on the matrix equation $CABV(t)C = C$ and the matrix product $X_1(t) = BV(t)C$ gives the result which is presented in Figure 4. The display denoted by $V(t)$ represents an approximate solution of the matrix equation $CABV(t)C = C$. The time interval is $[0, 0.5]$, the solver is `ode15s`, the linear activation is selected, and $\gamma = 10^{11}$.

Since the matrix $C$ is right invertible, the matrix equation $CABV(t)C = C$ gives the dual form of the matrix equation for computing $(CAB)^{\dagger}$; that is, $CABV(t) = I$.

Therefore, both $X$ and $X_1$ are approximations of the same outer inverse of $A$, equal to $B(CAB)^{\dagger}C$. To that end, it can be verified that $X$ and $X_1$ satisfy $\|X - X_1\| = 4.143699 * 10^{-11}$.

(c) The goal of this part of the example is to illustrate Theorem 3 and Algorithm 1. The matrices $A$ and $B$ satisfy $\text{rank}(AB) = \text{rank}(B)$, so that it is justifiable to search for a solution $U(t)$ of the matrix equation $BU(t)AB = B$ and the initiated outer inverse $X = BU$. In order to highlight the

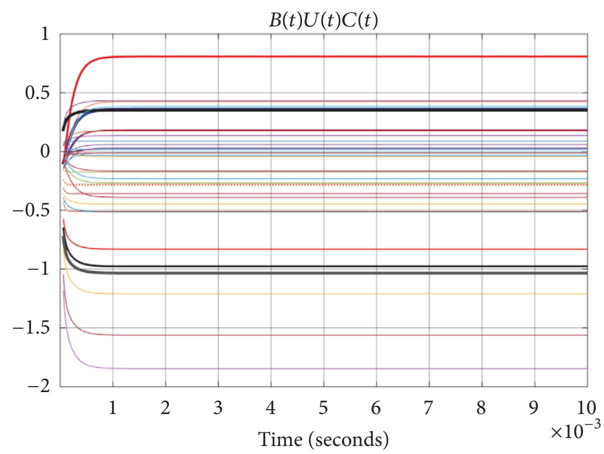


FIGURE 3: Trajectories of elements of the matrix $BUC$.

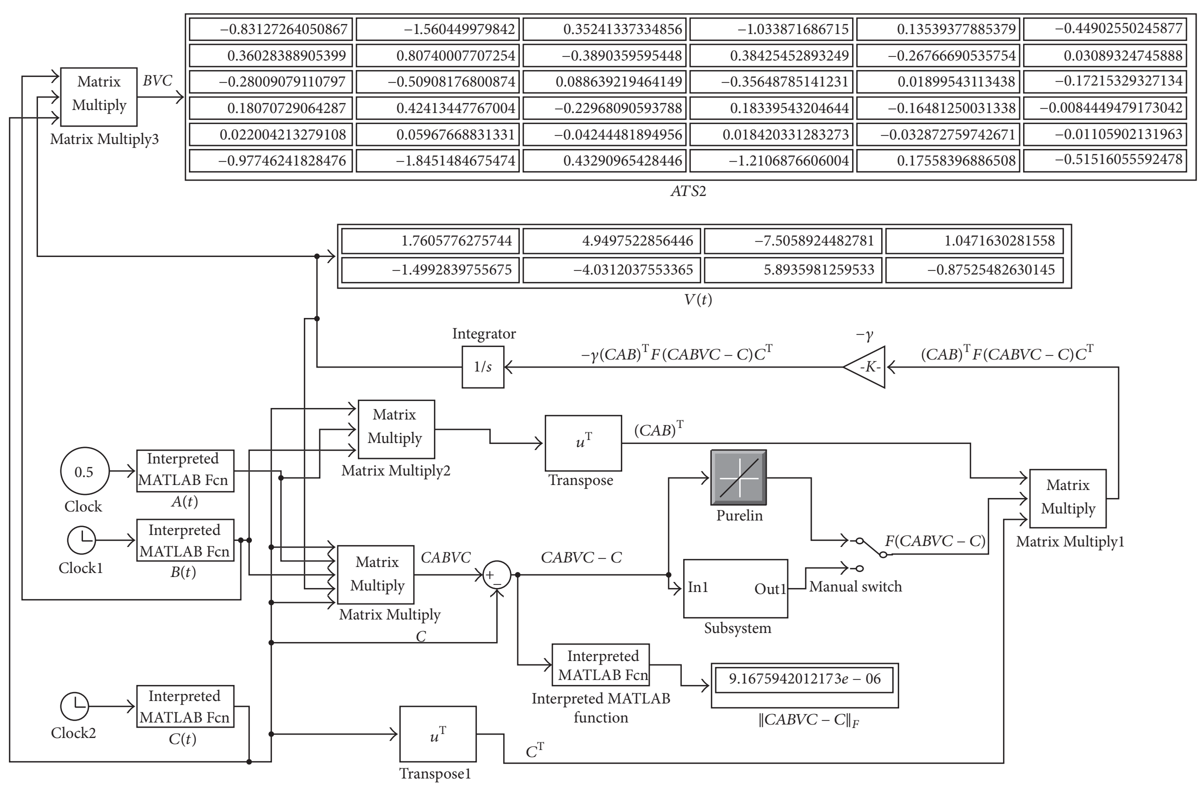


FIGURE 4: Simulink implementation of the GNN model for computing $CABV(t)C = C$, $X_1 = BVC$.

results derived by the implementation of Algorithm 1 it is important to mention that

$$(AB)^{\dagger} = \begin{bmatrix} 0.167297 & -0.167297 & 0.00708203 & -0.123801 & -0.236308 & 0.239756 \\ -0.203528 & 0.203528 & -0.279822 & -0.0731705 & -0.112743 & -0.385548 \end{bmatrix},$$

$$B(AB)^{\dagger} = \begin{bmatrix} 0.0786607 & -0.0786607 & -0.0687173 & -0.117658 & -0.217431 & 0.0877929 \\ -0.105529 & 0.105529 & -0.176364 & -0.0637471 & -0.104615 & -0.21073 \\ 0.0174033 & -0.0174033 & -0.0494166 & -0.0542619 & -0.098595 & 0.00758195 \\ -0.0633756 & 0.0633756 & -0.118603 & -0.0487517 & -0.0815486 & -0.130946 \\ -0.0120938 & 0.0120938 & -0.027072 & -0.0129663 & -0.0221131 & -0.0265246 \\ 0.0980931 & -0.0980931 & -0.0646451 & -0.129357 & -0.240083 & 0.116766 \end{bmatrix} \in A\{2\}_{\mathcal{R}(B),*}. \tag{68}$$

On the other hand, the Simulink implementation gives another element $BU(t)$ from $A\{2\}_{\mathcal{R}(B),*}$, different from $X_1 = (AB)^{\dagger}$. The matrix $BU(t)$ is presented in Figure 5. The display denoted by $U(t)$ represents an approximate solution of the matrix equation $BU(t)AB = B$. The time interval is $[0, 10^{-2}]$ and the solver is `ode15s`.

(d) The goal of this part of the example is to illustrate Theorem 5 and Algorithm 2. Since $\operatorname{rank}(CA) = \operatorname{rank}(C)$, it is justifiable to search for a solution of the matrix equation $CAV(t)C = C$. The Simulink implementation of the GGNN model which is based on the matrix equation $C(t)A(t)V(t)C(t) = C(t)$ gives the result which is presented in Figure 6. The display denoted by $V(t)$ represents an approximation of $V(t)$. The display denoted by $ATS2$ represents the matrix product $X = V(t)C(t)$. The time interval is $[0, 1]$ and the solver is `ode15s`. The activation is achieved by the power-sigmoid function. The corresponding outer inverse of $A$ is $X = VC \in A\{2\}_{*,\mathcal{N}(C)}$.

It is important to mention that the results $V(t)$ and $X = V(t)C$ given by the implementation of Algorithm 2 are different from the pseudoinverse of $CA$ and $(CA)^{\dagger}C$, since

$$(CA)^{\dagger} = \begin{bmatrix} 120140. & 129792. & 27421.9 & -618952. \\ -90013.5 & -47865.2 & -6777.93 & 329013. \\ -52937.6 & -1464.19 & 3452.26 & 120689. \\ 23062. & -103225. & -30460.2 & 230800. \\ -36793.4 & -112814. & -28769.9 & 388910. \\ 66669. & 217503. & 55777.9 & -740399. \end{bmatrix},$$

$$(CA)^{\dagger} C = \begin{bmatrix} 0.800499 & 0.290122 & -0.192667 & -0.201861 & -0.0498093 & -0.0355252 \\ -0.714584 & -0.247028 & -0.0707528 & -0.0994969 & -0.155725 & -0.111067 \\ -0.615629 & -0.552896 & 0.153409 & -0.291012 & 0.0511091 & -0.0352418 \\ 0.408051 & 0.436046 & -0.154673 & 0.37013 & -0.115624 & -0.15416 \\ -0.373293 & -0.441438 & 0.209825 & -0.0172156 & 0.0895808 & -0.149952 \\ 0.580871 & 0.558288 & -0.208561 & -0.0619027 & -0.0250655 & 0.339354 \end{bmatrix} \in A\{2\}_{*,\mathcal{N}(C)}. \tag{69}$$

*Example 22.* The aim of the present example is a verification of Theorem 6 and Algorithm 3 in the important case $B = C = A^{\mathrm{T}}$. For this purpose, we consider the same matrix $A$ as in Example 21. The *Mathematica* function `Pseudoinverse` gives the following exact Moore-Penrose inverse of $A$:

$$A^{\dagger} = \begin{bmatrix} \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 \\ -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & -\frac{1}{6} & -\frac{1}{3} & \frac{5}{12} & \frac{1}{12} \\ 0 & 0 & -\frac{1}{3} & -\frac{1}{6} & \frac{1}{12} & \frac{5}{12} \end{bmatrix}. \tag{70}$$

It can be approximated using the Simulink implementation of Algorithm 3 corresponding to the choice $B = C = A^{\mathrm{T}}$. Indeed, according to Example 21, the Simulink implementation of Algorithm 3 approximates the outer inverse $A^{\mathrm{T}}(A^{\mathrm{T}}AA^{\mathrm{T}})^{\dagger}A^{\mathrm{T}} = A^{\dagger}$. The implementation and generated results are presented in Figure 7. The GGNN model underlying the implementation is

$$\begin{aligned} \dot{U}(t) &= -\gamma A(t) \\ &\cdot \mathcal{F}\left(A(t)^{\mathrm{T}} U(t) A(t)^{\mathrm{T}} A(t) A(t)^{\mathrm{T}} - A(t)^{\mathrm{T}}\right) \\ &\cdot \left(A(t)^{\mathrm{T}} A(t) A(t)^{\mathrm{T}}\right)^{\mathrm{T}}. \end{aligned} \tag{71}$$

The display denoted by $U(t)$ represents an approximate solution of the matrix equation $A^{\mathrm{T}}U(t)A^{\mathrm{T}}AA^{\mathrm{T}} = A^{\mathrm{T}}$ and the display denoted by $MP$ represents an approximation of $A^{\dagger}$. The time interval is $[0, 0.001]$, the solver is `ode15s`, and the scaling parameter is assigned to $\gamma = 10^8$.

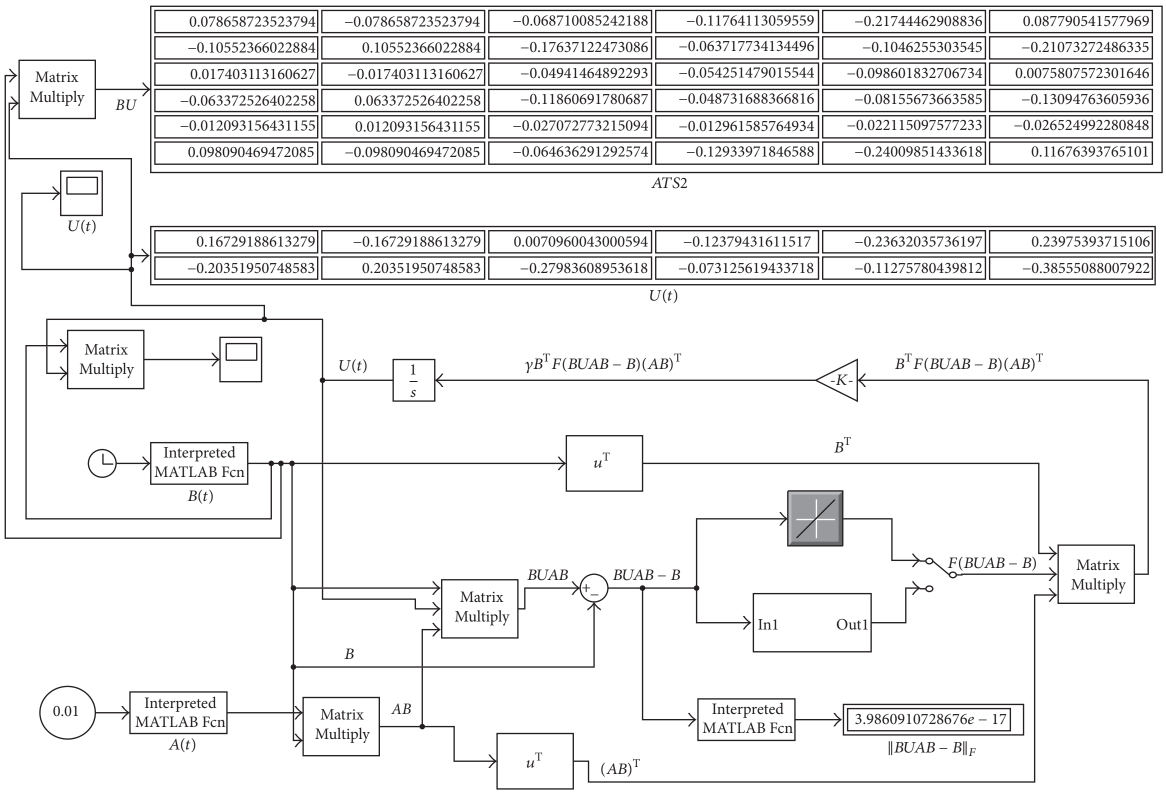


Figure 5: Simulink implementation of the GNN model for computing $BUAB = B$, $X = BU \in A\{2\}_{\mathcal{R}(B),*}$.

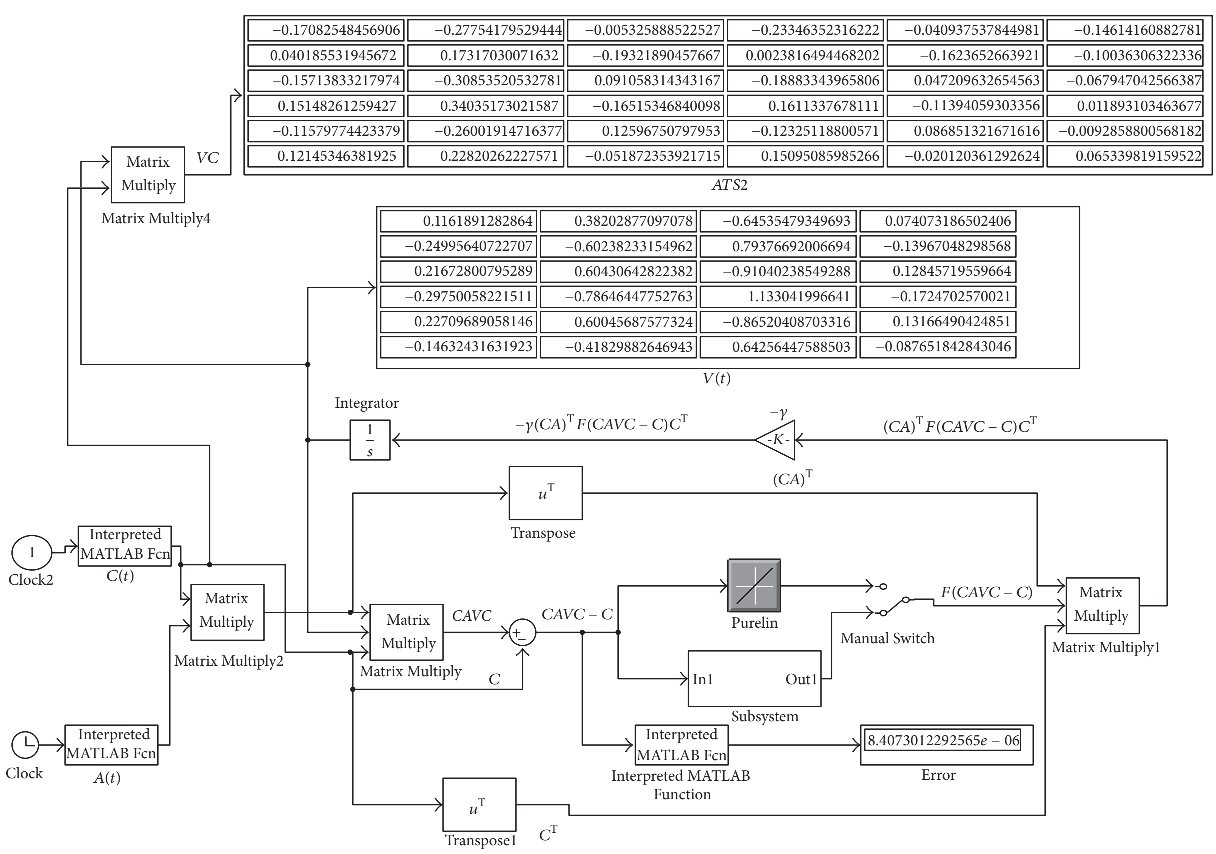


Figure 6: Simulink implementation of the GNN model for computing $CAVC = C$, $X = VC \in A\{2\}_{*,\mathcal{N}(C)}$.

*Example 23.* Let us consider the same matrix $A$ as in Example 21 and

$$B = \begin{bmatrix} 0.895516 & 0.0576096 & 0.25043 & 0.475532 & 0.862471 \\ 0.792079 & 0.248449 & 0.880375 & 0.567239 & 0.9282 \\ 0.808897 & 0.602233 & 0.0492111 & 0.88686 & 0.769442 \\ 0.258699 & 0.711749 & 0.961789 & 0.556687 & 0.880079 \\ 0.665172 & 0.164182 & 0.33616 & 0.892039 & 0.564932 \\ 0.640587 & 0.578898 & 0.278248 & 0.873279 & 0.660159 \end{bmatrix} \in \mathbb{R}_5^{6\times 5},$$
$$C = \begin{bmatrix} 0.351124 & 0.472523 & 0.796377 & 0.810286 & 0.484798 & 0.286383 \\ 0.505833 & 0.717046 & 0.246185 & 0.810956 & 0.22764 & 0.363135 \\ 0.499275 & 0.417029 & 0.442484 & 0.596716 & 0.573046 & 0.798864 \\ 0.513633 & 0.380053 & 0.317329 & 0.991615 & 0.917641 & 0.774303 \\ 0.969499 & 0.291356 & 0.926272 & 0.736567 & 0.609807 & 0.807355 \end{bmatrix} \in \mathbb{R}_5^{5\times 6}. \quad (72)$$

The matrices $B$ and $C$ are generated with the purpose of illustrating Theorem 12 and Algorithms 8 and 9. Conditions (iv) and (v) of Theorem 12 are satisfied. Therefore, it is expectable that the results generated by Algorithms 8 and 9 are the same.

The Simulink implementation of Algorithm 9 generates results presented in Figure 8. The simulation is performed within the time interval which is $[0, 10]$, the scaling constant is $\gamma = 10^7$, and the selected solver is `ode15s`.

The Simulink implementation of Algorithm 8 generates the results presented in Figure 9. The time interval is $[0, 0.5]$, $\gamma = 10^{11}$, and the solver is `ode15s`.

As a verification, *Mathematica* gives the following result:

$$X_1 = A^{(1,2)}_{\mathscr{R}(B),\mathscr{N}(C)} = B(AB)^{\dagger} A (CA)^{\dagger} C$$
$$= \begin{bmatrix} 0.0923811 & -0.407619 & -0.25 & -0.25 & -7.882583475 * 10^{-15} & -1.78745907 * 10^{-14} \\ -0.500161 & -0.00016084 & -0.25 & -0.25 & -1.776356839 * 10^{-15} & -3.44169138 * 10^{-15} \\ 4.4104 & -2.59426 & -14.9626 & -3.15303 & 3.96324 & 12.6509 \\ 4.73567 & -2.269 & -15.4626 & -2.65303 & 3.96324 & 12.6509 \\ 3.75197 & -3.2527 & -15.6292 & -3.48636 & 4.62991 & 12.9842 \\ 4.31832 & -2.68635 & -15.7959 & -3.3197 & 4.29657 & 13.3175 \end{bmatrix}. \quad (73)$$

Let us observe that $X = A^{(1,2)}_{\mathscr{R}(B),\mathscr{N}(C)} = B(CAB)^{\dagger}C$ and $X_1 = B(AB)^{\dagger}A(CA)^{\dagger}C$ are very close with respect to the Frobenius norm, since $\|X - X_1\| = 4.710014456589536 * 10^{-12}$. In the case $U = (CAB)^{\dagger}$ and $X = BUC$, the matrix equations $CAX = CABUC = C$ and $XAB = BUC = B$ are satisfied, since

$$\|CABUC - C\| = 1.631647583439993 * 10^{-13},$$
$$\|BUCAB - B\| = 2.405407190529498 * 10^{-13}. \quad (74)$$

*Example 24.* (a) Consider the time-varying symmetric matrix $S_5$, belonging to $n \times n$ matrices $S_n$ of rank $n-1$ from [32]:

$$S_5(t) = \begin{bmatrix} t+1 & t & t & t & t+1 \\ t & t-1 & t & t & t \\ t & t & t+1 & t & t \\ t & t & t & t-1 & t \\ t+1 & t & t & t & t+1 \end{bmatrix}. \quad (75)$$

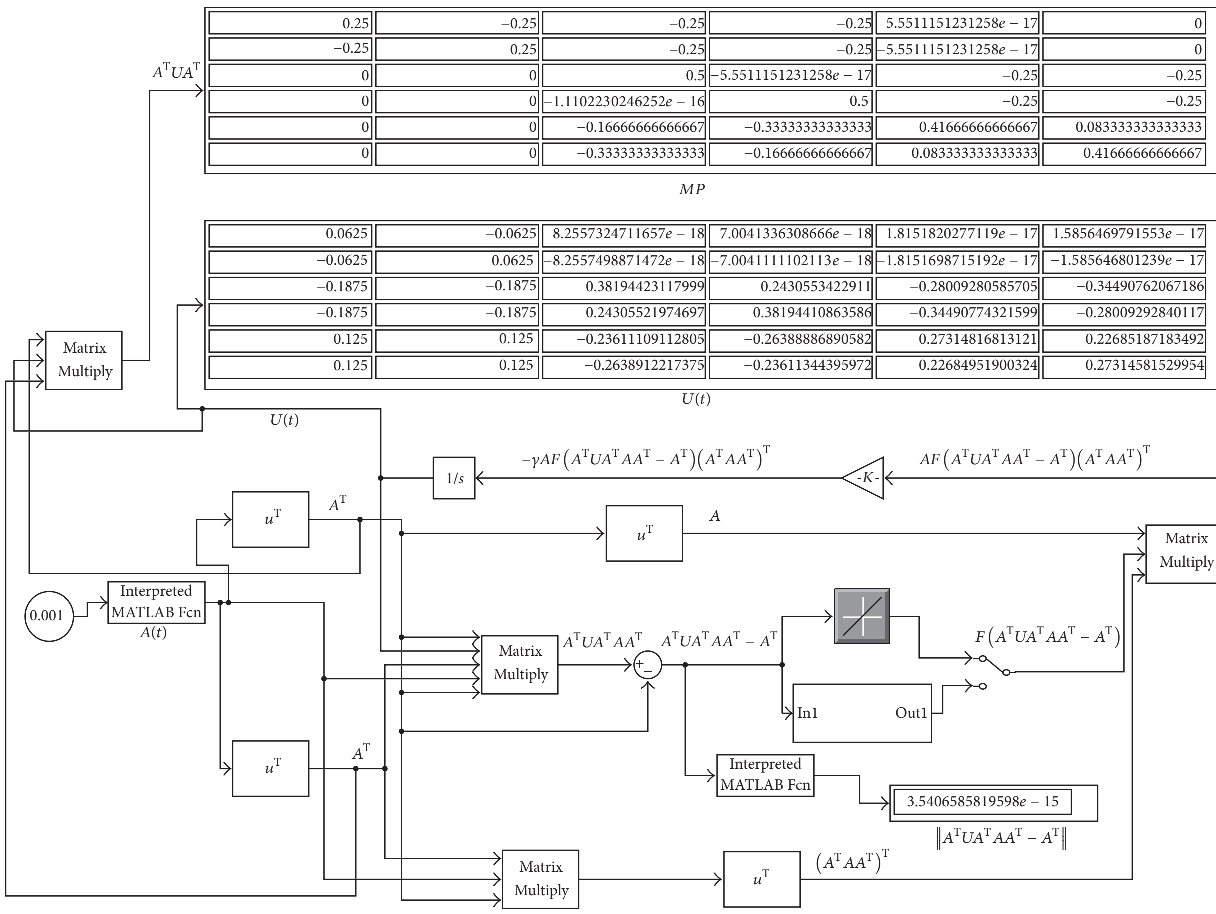


Figure 7: Simulink implementation of the GNN model for computing $A^{\dagger}$ using Algorithm 3.

The Moore-Penrose inverse of $S_5(t)$ is equal to

$$S_5(t)^{\dagger} = \begin{bmatrix} \frac{1-t}{4} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & \frac{1-t}{4} \\ \frac{t}{2} & -t-1 & t & -t & \frac{t}{2} \\ -\frac{t}{2} & t & 1-t & t & -\frac{t}{2} \\ \frac{t}{2} & -t & t & -t-1 & \frac{t}{2} \\ \frac{1-t}{4} & \frac{t}{2} & -\frac{t}{2} & \frac{t}{2} & \frac{1-t}{4} \end{bmatrix}. \quad (76)$$

Figure 10 shows the Simulink adopted computation of $S_5(t)^{\dagger}$ in the time period $[0, 5*10^{-7}]$ using the solver `ode15s` and the parameter $\gamma = 10^8$.

Trajectories of approximations of the entries in the matrix $S_5(t)^{\dagger}$ inside the time $[0, 5*10^{-7}]$ and generated using $\gamma = 10^8$ are presented in Figure 11. It is evident that these trajectories follow the graphs of the corresponding different expressions (representing entries) in $S_5^{\dagger}$.

(b) Now, consider the following matrices $B(t)$ and $C(t)$ in conjunction with $S_5(t)$:

$$B(t) = \begin{bmatrix} 2t+1 & t & t \\ t & 2t-1 & t \\ t & t & 2t+1 \\ t & t & t \\ 2t+1 & t & t \end{bmatrix},$$
$$C(t) = \begin{bmatrix} t^2+1 & t^2 & t^2 & t^2 & t^2+1 \\ t^2 & t^2-1 & t^2 & t^2 & t^2 \\ t^2 & t^2 & t^2+1 & t^2 & t^2 \end{bmatrix}. \quad (77)$$

The outer inverse $S_5(t)^{(2)}_{\mathscr{R}(B),\mathscr{N}(C)}$ of $S_5(t)$ corresponding to $B(t)$ and $C(t)$ is equal to

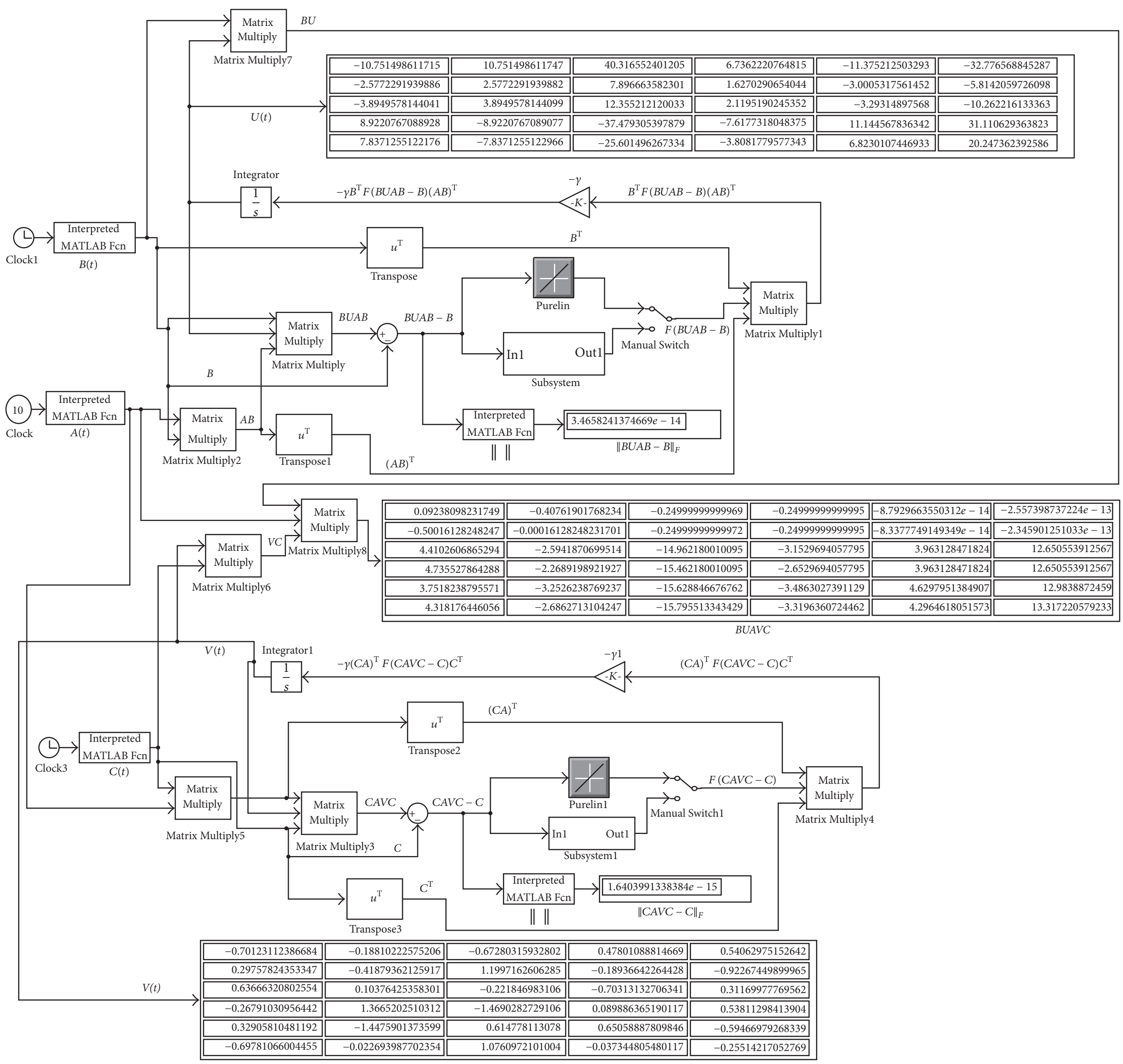


Figure 8: Simulink implementation of Algorithm 9.

$$
S_5(t)^{(2)}_{\mathscr{R}(B),\mathcal{N}(C)} = B\left(CS_5(t)B\right)^{-1}C
$$

$$
= \begin{bmatrix}
\frac{-10t^5+6t^4+3t^3-t^2+t+1}{-63t^5+42t^4-t^3+2t^2+8t+4} & \frac{t\left(15t^4+t^3+6t^2-2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(-15t^4+13t^3-8t^2+4t+2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t^2\left(-20t^3+6t^2+3t-1\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{-10t^5+6t^4+3t^3-t^2+t+1}{-63t^5+42t^4-t^3+2t^2+8t+4} \\
\frac{t\left(-2t^4+3t^3+8t^2-3t-2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{-66t^5+13t^4-25t^3+2t^2+12t+4}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(3t^4+t^3+16t^2-8t-4\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t^2\left(67t^3-9t^2-18t-4\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(-2t^4+3t^3+8t^2-3t-2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} \\
\frac{t\left(-16t^4+13t^3-6t^2+3t+2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(39t^4-7t^3+8t^2-4\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{-24t^5+7t^4+15t^3-6t^2+4t+4}{-63t^5+42t^4-t^3+2t^2+8t+4} & \frac{t^2\left(-31t^3+13t^2+2t-4\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(-16t^4+13t^3-6t^2+3t+2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} \\
\frac{t\left(2t^4-3t^3+t^2-5t+1\right)}{-63t^5+42t^4-t^3+2t^2+8t+4} & -\frac{t\left(3t^4+29t^3+6t^2+18t+4\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(3t^4+t^3-2t^2+10t-4\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t^3\left(4t^2+33t-1\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(2t^4-3t^3+t^2-5t+1\right)}{-63t^5+42t^4-t^3+2t^2+8t+4} \\
\frac{-10t^5+6t^4+3t^3-t^2+t+1}{-63t^5+42t^4-t^3+2t^2+8t+4} & \frac{t\left(15t^4+t^3+6t^2-2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t\left(-15t^4+13t^3-8t^2+4t+2\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{t^2\left(-20t^3+6t^2+3t-1\right)}{63t^5-42t^4+t^3-2t^2-8t-4} & \frac{-10t^5+6t^4+3t^3-t^2+t+1}{-63t^5+42t^4-t^3+2t^2+8t+4}
\end{bmatrix}. \quad (78)
$$

Its computation in the time period $[0, 5 * 10^{-2}]$ using solver `ode15s` and the parameter $\gamma = 10^{11}$ is presented in Figure 12.

*Example 25.* Here we discuss the behaviour of Algorithm 3 in the case when the condition $\mathrm{rank}(CAB) = \mathrm{rank}(B) = \mathrm{rank}(C)$

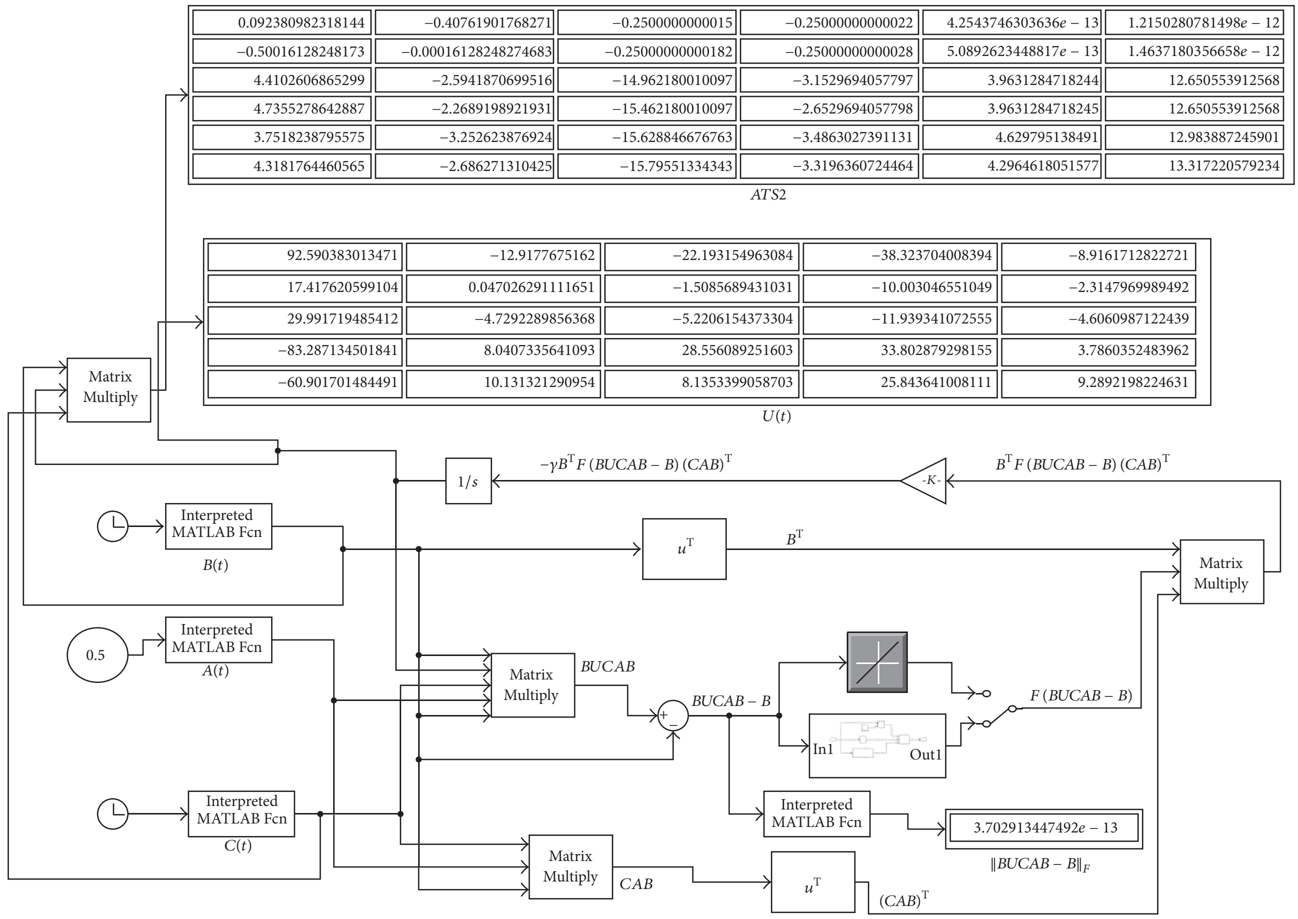


Figure 9: Simulink implementation of Algorithm 8.

is not satisfied. For this purpose, let us consider the matrices

$$A = \begin{bmatrix} 5 & -8 & -16 & 24 & 0 \\ 6 & -11 & -18 & 24 & 0 \\ -7 & 14 & 26 & -36 & 0 \\ -4 & 8 & 16 & -23 & 0 \\ 2 & -6 & -10 & 12 & -3 \end{bmatrix},$$

$$B = \begin{bmatrix} 18 & -34 & -52 & 72 & -8 \\ 36 & -72 & -108 & 144 & 0 \\ -36 & 82 & 130 & -168 & 8 \\ -18 & 43 & 70 & -90 & 8 \\ -36 & 70 & 100 & -132 & 2 \end{bmatrix}, \tag{79}$$

$$C = \begin{bmatrix} -2 & 10 & 12 & -12 & 4 \\ 4 & -6 & -12 & 16 & 0 \\ 2 & -6 & -4 & 4 & -4 \\ 4 & -10 & -12 & 14 & -4 \\ 6 & -15 & -22 & 26 & -4 \end{bmatrix}.$$

These matrices do not satisfy the requirement $\operatorname{rank}(CAB) = \operatorname{rank}(B) = \operatorname{rank}(C)$ of Algorithm 3, since

$$\begin{aligned} \operatorname{rank}(A) &= 5, \\ \operatorname{rank}(B) &= 4, \\ \operatorname{rank}(C) &= 3, \\ \operatorname{rank}(CAB) &= 2. \end{aligned} \tag{80}$$

On the other hand, the conditions $\operatorname{rank}(AB) = \operatorname{rank}(B)$ and $\operatorname{rank}(CA) = \operatorname{rank}(C)$ are valid, so that the conditions required in Algorithms 1 and 2 hold. An application of Algorithm 3 in the time $[0, 10^{-9}]$, based on the scaling constant $\gamma = 10^7$ and the `ode15s` solver, gives the results for $U(t)$ and $X = BUC$ as it is presented in Figure 13.

An application of the dual case of Algorithm 3 in the time $[0, 10^{-8}]$, based on the scaling constant $\gamma = 10^7$ and the `ode15s` solver, gives the results for $V(t)$ and $X = BVC$ as it is presented in Figure 14.

Trajectories of the elements of the matrix $B(t)U(t)C(t)$ in the period of time $[0, 10^{-9}]$ are presented in Figure 15.

According to the obtained results, the following can be concluded.

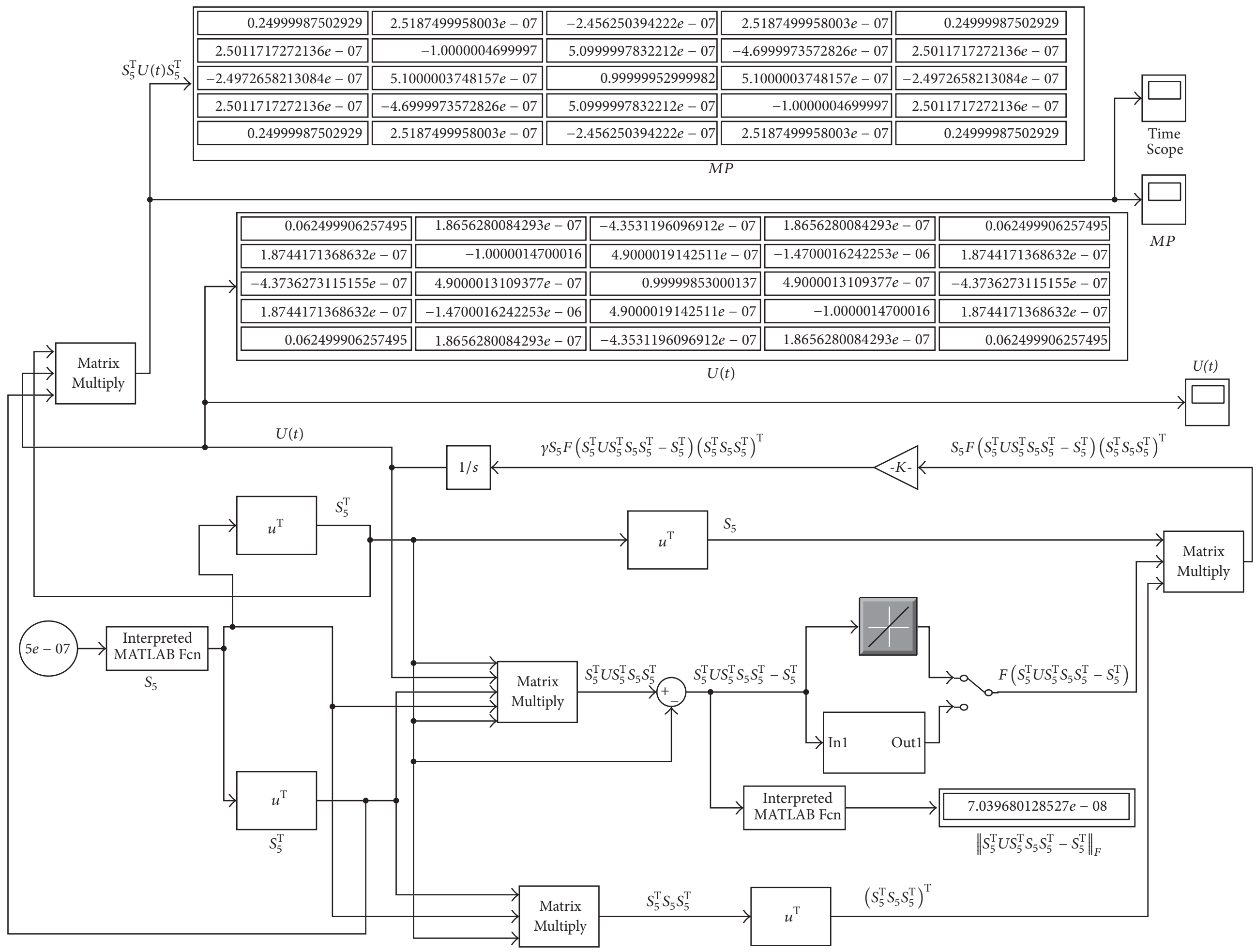


FIGURE 10: The Simulink adopted for computation of $S_5(t)^\dagger$.

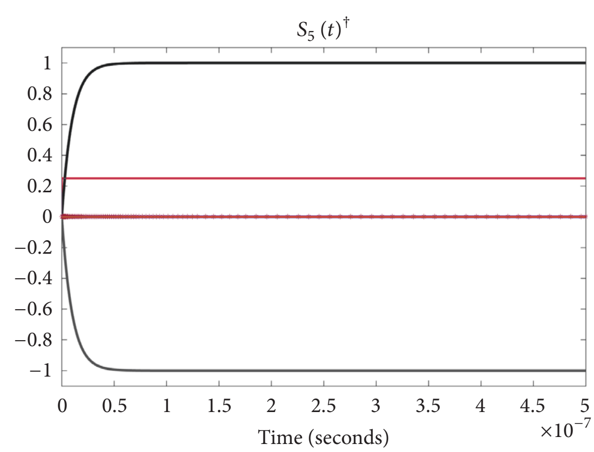


FIGURE 11: Trajectories of elements of the matrix $S_5(t)^\dagger$.

(1) The matrix equation $BUCAB = B$ is not satisfied, since $\|BUCAB - B\| = 39.53256$. This fact is expectable since the conditions $\operatorname{rank}(CAB) = \operatorname{rank}(B) = \operatorname{rank}(C)$ are not satisfied nor is the matrix $B$ invertible. Similarly, the matrix equation $BUCAB = B$ is not satisfied, since $\|CABVC - C\| = 27.412588$.

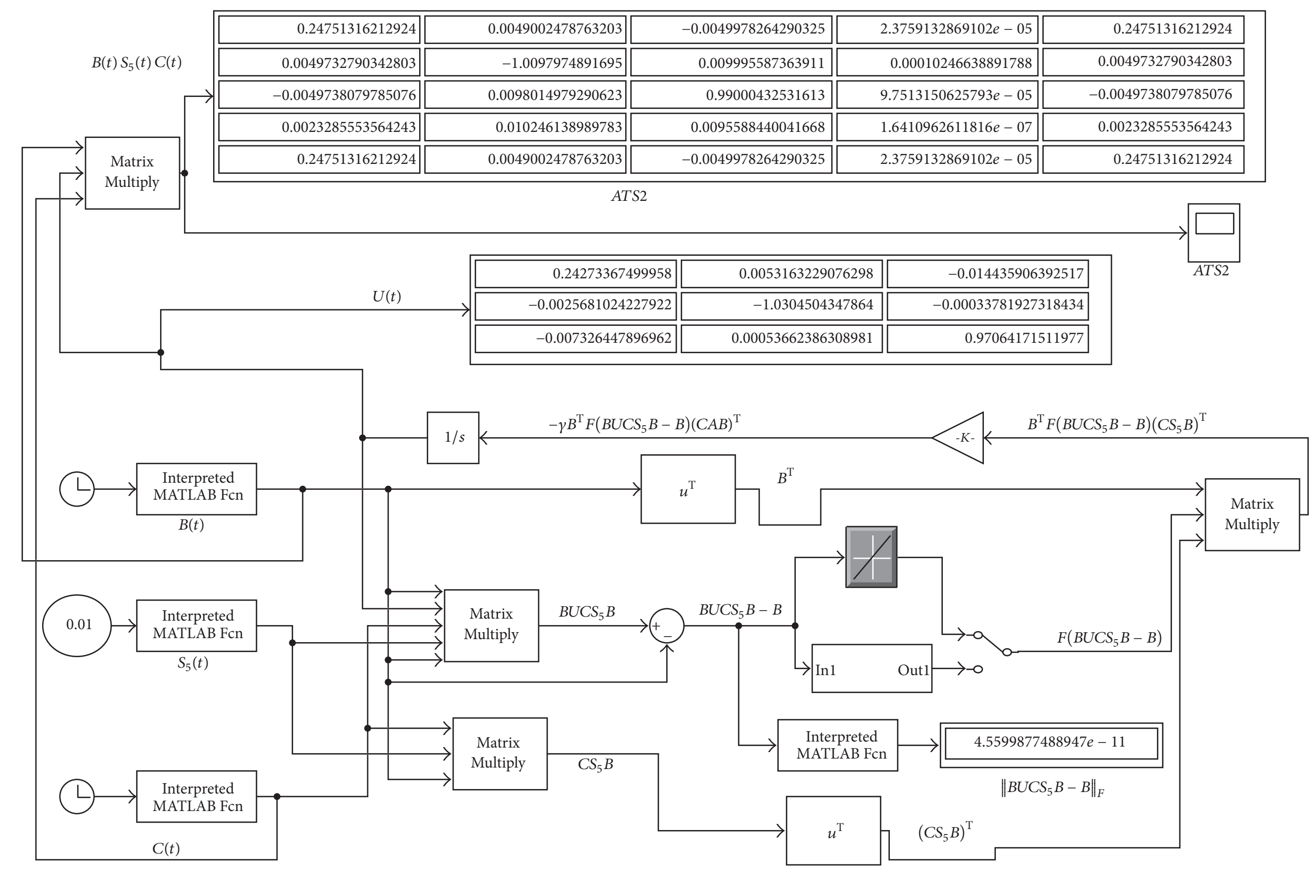


FIGURE 12: The Simulink model for computing $B(t)(C(t)S_5(t)B(t))^{(1)}C(t)$.

(2) Both the matrices $U$ and $V$ are approximations of $(CAB)^{\dagger}$, since

$$(CAB)^{\dagger}=\begin{bmatrix} 0.0345588 & 0. & 0.0321078 & 0.0154412 & -0.0178922 \\ -0.0116558 & 0. & -0.0105664 & -0.00501089 & 0.00610022 \\ 0.0227669 & 0. & 0.0216776 & 0.0105664 & -0.0116558 \\ 0.000272331 & 0. & -0.000272331 & -0.000272331 & -0.000272331 \\ -0.0230392 & 0. & -0.0214052 & -0.0102941 & 0.0119281 \end{bmatrix}. \quad (81)$$

This means that the solutions of the matrix equations $BUCAB = B$ and $CABVC = C$ given by the GNN model approximate the solution of the GNN model corresponding to the matrix equations $UCAB = I$ and $CABV = I$, respectively, which is equal to $(CAB)^{\dagger}$.

(3) Accordingly, the output denoted by $ATS2$ approximates the outer inverse

$$X = B(CAB)^{\dagger}C = \begin{bmatrix} -0.537582 & 1.57435 & 1.91993 & -2.15033 & 0.614379 \\ -0.492157 & 0.0127451 & -2.01373 & 2.03137 & 1.01961 \\ 0.38366 & 1.01928 & 4.28693 & -4.46536 & -1.12418 \\ 0.237582 & 0.375654 & 1.98007 & -2.04967 & -0.614379 \\ 0.715033 & -1.37974 & -0.667974 & 0.860131 & -1.04575 \end{bmatrix} \quad (82)$$

exactly in five decimals. In conclusion, the Simulink implementation of Algorithm 3 computes the outer inverse $X = B(CAB)^{\dagger}C$ which satisfies condition (29) from the definition of the $(B,C)$-inverse, but not condition (28) from the same definition. In other words, $X$ satisfies neither $\mathcal{R}(X) = \mathcal{R}(B)$ nor $\mathcal{N}(X) = \mathcal{N}(C)$.

(4) Observations 2 and 3 finally imply that the GGNN model can be used for online time-varying pseudo-inversion of both the matrices $A$ and $CAB$.

## 5. Conclusion

The contribution of the present paper is both theoretical and computationally applicable. Conditions for the existence and

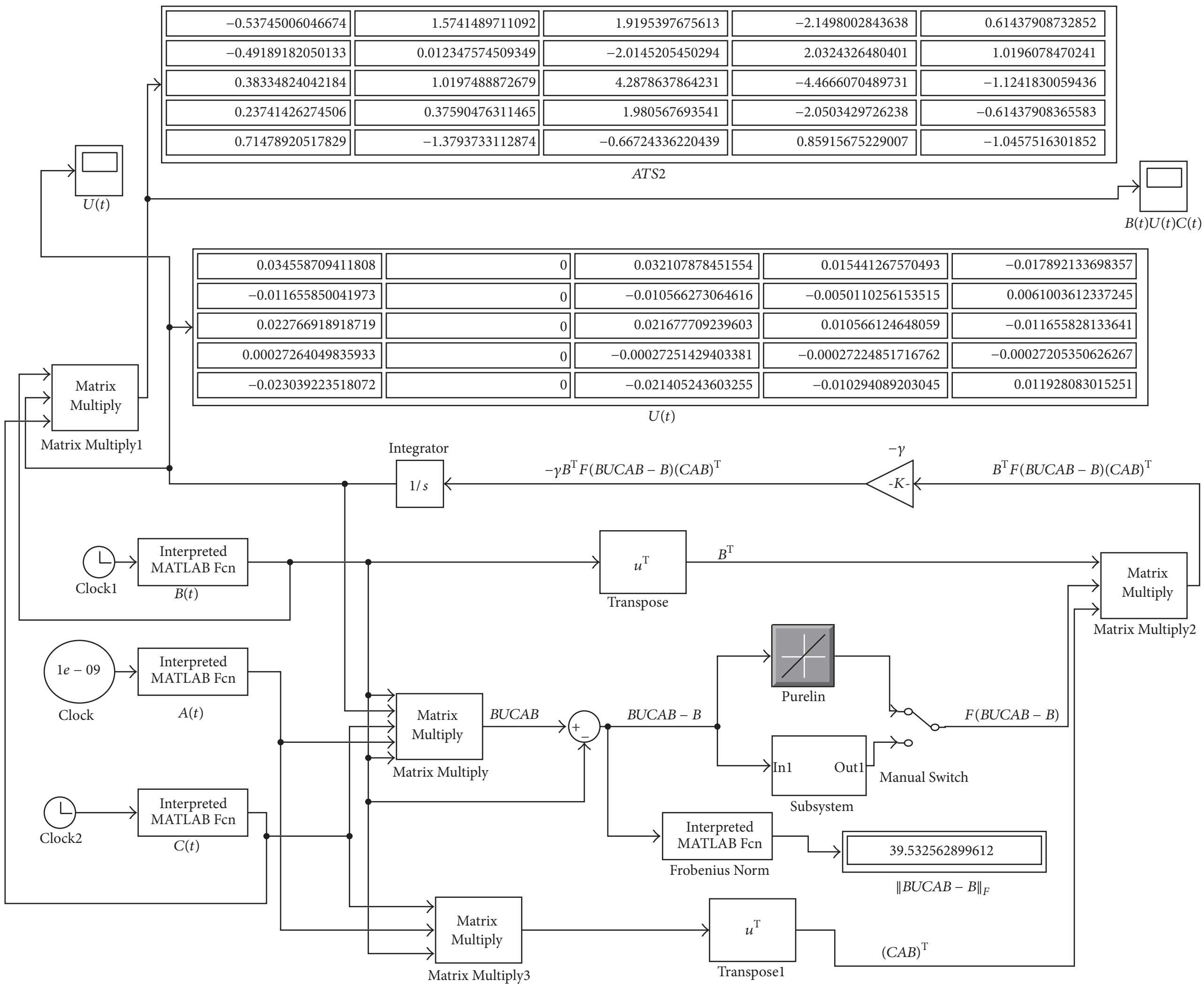


Figure 13: The implementation of Algorithm 3 when its conditions are not satisfied.

representations of {2}-, {1, 2}-, and {1}-inverses with some assumptions on their ranges and null spaces are proposed. A new computational framework for these generalized inverses is proposed. This approach arises from the derived general representations and involves solutions of certain matrix equations. In general, the methods and algorithms proposed in the present paper are aimed at computation of various classes of generalized inverses of the form $B(CAB)^{(1)}C$, where $(CAB)^{(1)}$ are solutions of the proposed matrix equations solvable under specified conditions.

Our decision is to apply the GGNN approach in finding solutions of required matrix equations. Also, we use Simulink implementation of the underlying RNN models. This decision allows us to extend derived algorithms to time-varying matrices. Also, such an approach makes it possible to compute two types of generalized inverses, namely, inner and/or outer inverses of $A$ and inner inverses of the matrix product $CAB$. Illustrative numerical examples and simulation examples are presented to demonstrate validity of the derived theoretical results and proposed methods.

It is worth mentioning that the blurring process which is applied on the original image $F$ and produces the blurred image $G$ is expressed in the form of a certain matrix equation of the form

$$G = H_c F H_r^T, \quad G \in \mathbb{R}^{m_1 \times m_2},\ H_c \in \mathbb{R}^{m_1 \times r},\ F \in \mathbb{R}^{r \times s},\ H_r \in \mathbb{R}^{m_2 \times s}, \tag{83}$$

wherein it is assumed that $s = m_2 + n_1 - 1$, $r = m_1 + n_2 - 1$, where $n_1$ (resp., $n_2$) is the length of the horizontal (resp., vertical) blurring in pixels. Solutions of these types of matrix equations which are based on the pseudoinverse of $H_c$ and $H_r$ and least squares solutions were investigated in [33–35]. Possible application of the proposed algorithms in finding least squares solutions of matrix equation (83) could be useful for further research.

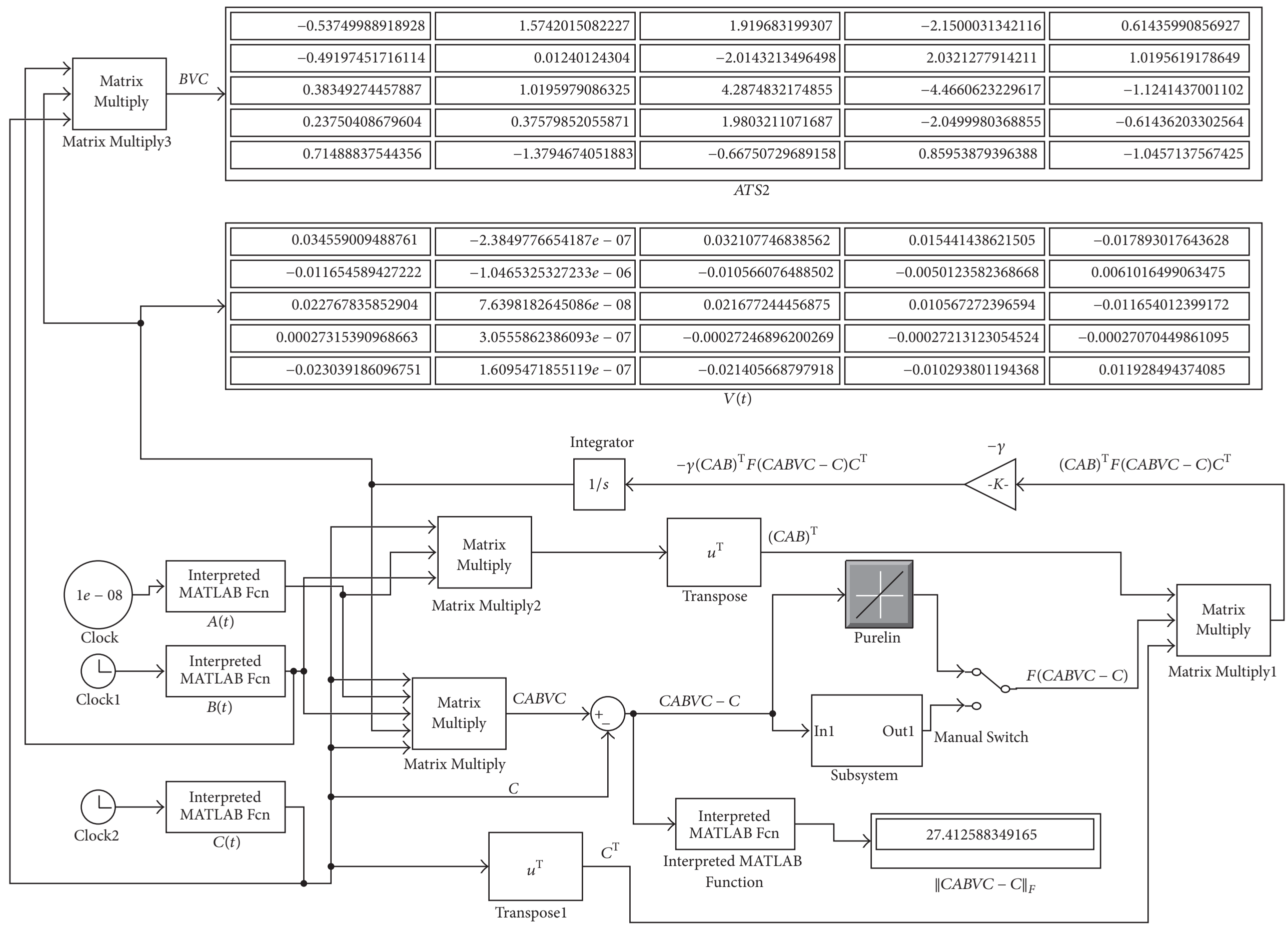


FIGURE 14: Dual implementation of Algorithm 3 when its conditions are not satisfied.

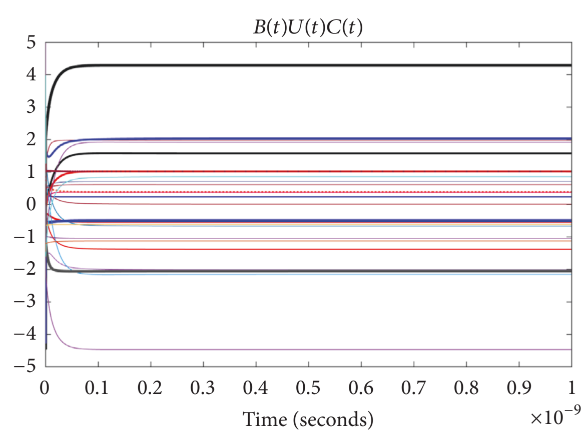


FIGURE 15: Trajectories of elements in $B(t)U(t)C(t)$ in the period of time $[0, 10^{-9}]$.

*Require*: Time varying matrices $A(t) \in \mathbb{C}^{m\times n}$, $B(t) \in \mathbb{C}^{n\times k}$ and $C(t) \in \mathbb{C}^{l\times m}$.
*Require*: Verify $\operatorname{rank}(C(t)A(t)B(t)) = \operatorname{rank}(B(t)) = \operatorname{rank}(C(t)) = \operatorname{rank}(A(t))$.
If these conditions are satisfied then continue.
(1) Solve the matrix equation $B(t)U(t)C(t)A(t)B(t) = B(t)$ with respect to an unknown matrix $U(t) \in \mathbb{C}^{k\times m}$.
(2) Return $X(t) = B(t)U(t)C(t) = A(t)^{(1,2)}_{\mathscr{R}(B),\mathscr{N}(C)}$.

ALGORITHM 9: Alternative computing of a $\{1, 2\}$-inverse with the prescribed range and null space.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The first and second author gratefully acknowledge support from the project supported by Ministry of Education and Science of Republic of Serbia, Grant no. 174013. The first and third author gratefully acknowledge support from the project "Applying Direct Methods for Digital Image Restoring" of the Goce Delčev University.

## References

[1] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory And Applications*, Springer, New York, NY, USA, 2nd edition, 2003.

[2] G. Wang, Y. Wei, and S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, New York, NY, USA, 2003.

[3] X. Sheng and G. Chen, "Full-rank representation of generalized inverse $A^{(2)}_{T,S}$ and its application," *Computers & Mathematics with Applications*, vol. 54, no. 11-12, pp. 1422–1430, 2007.

[4] P. Stanimirović, S. Bogdanović, and M. Ćirić, "Adjoint mappings and inverses of matrices," *Algebra Colloquium*, vol. 13, no. 3, pp. 421–432, 2006.

[5] Y.-L. Chen and X. Chen, "Representation and approximation of the outer inverse of $A^{(2)}_{T,S}$ a matrix $A$," *Linear Algebra and its Applications*, vol. 308, no. 1–3, pp. 85–107, 2000.

[6] X. Liu, H. Jin, and Y. Yu, "Higher-order convergent iterative method for computing the generalized inverse and its application to Toeplitz matrices," *Linear Algebra and Its Applications*, vol. 439, no. 6, pp. 1635–1650, 2013.

[7] X. Liu and Y. Qin, "Successive matrix squaring algorithm for computing the generalized inverse $A^{(2)}_{T,S}$," *Journal of Applied Mathematics*, vol. 2012, Article ID 262034, 12 pages, 2012.

[8] P. S. Stanimirović and D. S. Cvetković-Ilić, "Successive matrix squaring algorithm for computing outer inverses," *Applied Mathematics and Computation*, vol. 203, no. 1, pp. 19–29, 2008.

[9] P. S. Stanimirović and F. Soleymani, "A class of numerical algorithms for computing outer inverses," *Journal of Computational and Applied Mathematics*, vol. 263, pp. 236–245, 2014.

[10] Y. Wei, "A characterization and representation of the generalized inverse $A^{(2)}_{T,S}$ and its applications," *Linear Algebra and Its Applications*, vol. 280, no. 2-3, pp. 87–96, 1998.

[11] Y. Wei and H. Wu, "The representation and approximation for the generalized inverse $A^{(2)}_{T,S}$," *Applied Mathematics and Computation*, vol. 135, no. 2-3, pp. 263–276, 2003.

[12] Y. Wei and H. Wu, "$(T, S)$ splitting methods for computing the generalized inverse $A^{(2)}_{T,S}$ and rectangular systems," *International Journal of Computer Mathematics*, vol. 77, no. 3, pp. 401–424, 2001.

[13] H. Yang and D. Liu, "The representation of generalized inverse $A^{(2,3)}_{T,S}$ and its applications," *Journal of Computational and Applied Mathematics*, vol. 224, no. 1, pp. 204–209, 2009.

[14] N. S. Urquhart, "Computation of generalized inverse matrices which satisfy specified conditions," *SIAM Review*, vol. 10, pp. 216–218, 1968.

[15] M. P. Drazin, "A class of outer generalized inverses," *Linear Algebra and Its Applications*, vol. 436, no. 7, pp. 1909–1923, 2012.

[16] J. Jang, S. Lee, and S. Shin, "An optimization network for matrix inversion," in *Neural Information Processing Systems*, pp. 397–401, College Park, Md, USA, 1988.

[17] L. Fa-Long and B. Zheng, "Neural network approach to computing matrix inversion," *Applied Mathematics and Computation*, vol. 47, no. 2-3, pp. 109–120, 1992.

[18] J. Wang, "A recurrent neutral network for real-time matrix inversion," *Applied Mathematics and Computation*, vol. 55, no. 1, pp. 89–100, 1993.

[19] A. Cichocki, T. Kaczorek, and A. Stajniak, "Computation of the drazin inverse of a singular matrix making use of neural networks," *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 40, pp. 387–394, 1992.

[20] A. Cichock and D. Rolf Unbehauen, "Neural networks for solving systems of linear equations and related problems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, no. 2, pp. 124–138, 1992.

[21] J. Wang, "Recurrent neural networks for computing pseudoinverses of rank-deficient matrices," *SIAM Journal on Scientific Computing*, vol. 18, no. 5, pp. 1479–1493, 1997.

[22] Y. Wei, "Recurrent neural networks for computing weighted Moore-Penrose inverse," *Applied Mathematics and Computation*, vol. 116, no. 3, pp. 279–287, 2000.

[23] Y. Xia, T. Chen, and J. Shan, "A novel iterative method for computing generalized inverse," *Neural Computation*, vol. 26, no. 2, pp. 449–465, 2014.

[24] P. S. Stanimirović, I. S. Živković, and Y. Wei, "Recurrent neural network for computing the Drazin inverse," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2830–2843, 2015.

[25] I. S. Živković, P. S. Stanimirović, and Y. Wei, "Recurrent neural network for computing outer inverse," *Neural Computation*, vol. 28, no. 5, pp. 970–998, 2016.

[26] P. S. Stanimirović, I. S. Živković, and Y. Wei, "Neural network approach to computing outer inverses based on the full rank representation," *Linear Algebra and Its Applications*, vol. 501, pp. 344–362, 2016.

[27] C.-G. Cao and X. Zhang, "The generalized inverse $A^{(2)}_{T,*}$ and its applications," *Journal of Applied Mathematics and Computing*, vol. 11, no. (1-2), pp. 155–164, 2003.

[28] K. Chen, S. Yue, and Y. Zhang, "MATLAB simulation and comparison of zhang neural network and gradient neural network for online solution of linear time-varying matrix equation $AXB - C = 0$," in *Proceeding of the International Conference on Intelligent Computing (ICIC '08)*, D. S. Huang, D. C. Wunsch, D. S. Levine, and K. H. Jo, Eds., vol. 5227 of *LNAI*, pp. 68–75, Shanghai, China, 2008.

[29] Y. Zhang and K. Chen, "Comparison on zhang neural network and gradient neural network for time-varying linear matrix equation solving AXB = C Solving," in *Proceeding of the International Conference on Industrial Technology (IEEE ICIT '08)*, April 2008.

[30] P. S. Stanimirović, D. S. Cvetković-Ilić, S. Miljković, and M. Miladinović, "Full-rank representations of $\{2, 4\}$, $\{2, 3\}$—inverses and successive matrix squaring algorithm," *Applied Mathematics and Computation*, vol. 217, no. 22, pp. 9358–9367, 2011.

[31] S. Srivastava and D. K. Gupta, "A new representation for $A^{(2,3)}_{T,S}$," *Applied Mathematics and Computation*, vol. 243, pp. 514–521, 2014.

[32] G. Zielke, "Report on test matrices for generalized inverses," *Computing*, vol. 36, no. 1-2, pp. 105–162, 1986.

[33] P. S. Stanimirović, I. Stojanović, V. N. Katsikis, D. Pappas, and Z. Zdravev, "Application of the least squares solutions in image deblurring," *Mathematical Problems in Engineering*, vol. 2015, Article ID 298689, 18 pages, 2015.

[34] P. S. Stanimirović, S. Chountasis, D. Pappas, and I. Stojanović, "Removal of blur in images based on least squares solutions," *Mathematical Methods in the Applied Sciences*, vol. 36, no. 17, pp. 2280–2296, 2013.

[35] P. Stanimirović, I. Stojanović, S. Chountasis, and D. Pappas, "Image deblurring process based on separable restoration methods," *Computational and Applied Mathematics*, vol. 33, no. 2, pp. 301–323, 2014.