2010 8th IEEE International Conference on
Control and Automation
Xiamen, China, June 9-11, 2010

More Than Newton Iterations Generalized from Zhang Neural Network for Constant Matrix Inversion Aided with Line-Search Algorithm

Yunong Zhang, *Member, IEEE*, Dongsheng Guo, Chenfu Yi, Lingfeng Li, and Zhende Ke

***Abstract*— Since 12 March 2001, Zhang *et al* have proposed a special class of recurrent neural networks for online time-varying problems solving, especially for matrix inversion. For possible hardware (e.g., digital-circuit) realization, such Zhang neural networks (ZNN) could also be reformulated in the discrete-time form, which incorporates Newton iteration as a special case. In this paper, for constant matrix inversion, we generalize and investigate more discrete-time ZNN models (which could also be termed as ZNN iterations) by using multiple-point backward-difference formulas. For fast convergence to the theoretical inverse, a line-search algorithm is employed to obtain an appropriate step-size value (in each iteration). Computer-simulation results demonstrate the efficacy of the presented new discrete-time ZNN models aided with a line-search algorithm, as compared to Newton iteration.**

I. Introduction

The problem of linear matrix equations solving is considered to be a very fundamental problem widely encountered in science and engineering fields. Being a sub-topic of linear matrix equation solving, online matrix inversion is usually viewed as an essential part of many solutions; e.g., robot control [1]-[3], signal processing [4], [5] and statistics [4], [6]. In mathematics, matrix inversion could be generally formulated as $AX = I$, where coefficient matrix $A \in R^{n\times n}$, identity matrix $I \in R^{n\times n}$, and $X \in R^{n\times n}$ is the unknown matrix to be obtained. In recent decades, the neural-dynamic approach based on recurrent neural networks (RNN) has been viewed as a powerful alternative to online linear matrix equations solving (including matrix inversion), owing to its potential high-speed parallel-processing nature [1], [7]-[9].

Since 12 March 2001 [10]-[14], Zhang *et al* have formally proposed a special class of recurrent neural networks for online time-varying problems solving (e.g., matrix inversion). Differing from scalar-valued norm-based energy-functions widely used in conventional gradient-based neural networks (GNN) [1], [7], [8], the design of Zhang neural network (ZNN) is based on a matrix-valued error-function [10]-[12], [15]-[18]. Note that, such a ZNN design method aims at making every entry of the matrix-valued error function converge to zero [10]-[12], [15], [17], [18]. In addition, ZNN models are depicted generally in implicit dynamics, instead of explicit dynamics usually associated with GNN and/or Hopfield-type neural networks [1], [7], [8], [19], [20].

This work is sponsored by the Program for New Century Excellent Talents in University (NCET-07-0887), the opening Fund of Laboratory Sun Yat-sen University and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, China.

The authors are with the School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510006, China (e-mails: zhynong@mail.sysu.edu.cn, ynzhang@ieee.org, gdongsh2008@126.com, yichf@mail2.sysu.edu.cn, st04llf@mail2.sysu.edu.cn, kezhende@126.com).

Moreover, for the purpose of possible hardware (e.g., digital-circuit) realization, the discrete-time ZNN models for matrix inversion are developed and investigated in [15]-[17]. It is worth mentioning that Newton iteration (or to say, method) could be viewed as a special case of the discrete-time ZNN model, when linear activation functions and step-size $h \equiv 1$ are used [15]-[17]. Recalling the effectiveness and efficiency of ZNN models on linear matrix equations solving (including constant and time-varying ones) [10]-[18], in this paper, we further develop and investigate more ZNN iterations using equally-spaced multiple-point backward-difference formulas [21] for constant matrix inversion. In addition, the line-search algorithm [6], [22] is adopted as well to obtain an appropriate step-size value for the discrete-time ZNN models. In view of the link between Newton iteration and discrete-time ZNN models, numerical simulations are performed comparatively for the efficacy of these models on constant matrix inversion.

II. ZNN Model and Newton Iteration

To solve for the inverse of time-varying nonsingular matrix $A(t) \in R^{n\times n}$, the neural-network design methods are usually based on the definition equation, $A(t)X(t) - I = 0$.

Firstly, instead of using a scalar-valued norm-based positive energy-function $\|AX - I\|^2/2$ usually associated with GNN [1], [7], [8], the following matrix-valued indefinite error-function [10]-[12], [15], [17] could be defined to monitor online the time-varying matrix-inversion process:

$$E\big(X(t), t\big) := A(t)X(t) - I, \ t \geqslant 0. \quad (1)$$

Secondly, to make the error function $E(X(t), t)$ converge to zero [i.e., every entry $e_{ij}(t)$ of $E(X(t), t)$ converges to zero, $i, j = 1, 2, \cdots, n$], the ZNN design formula [10]-[12], [15], [17], [18] can be described as

$$\frac{dE\big(X(t), t\big)}{dt} = -\Gamma\mathcal{F}\big(E(X(t), t)\big), \quad (2)$$

where $\Gamma \in R^{n\times n}$ (usually $\Gamma = \gamma I$) is a positive-definite matrix used to scale the convergence rate of the neural-network solution, and $\mathcal{F}(\cdot) : R^{n\times n} \rightarrow R^{n\times n}$ denotes a matrix-valued activation-function array of neural networks. The processing array $\mathcal{F}(\cdot)$ here is made of n^2 monotonically-increasing odd activation-functions $f(\cdot)$ [14]. Generally speaking, any monotonically-increasing odd activation-function could be used for ZNN construction. We have generally discussed four basic types of activation functions, i.e., linear activation function, bipolar sigmoid activation function, power activation function, and power-sigmoid activation function [13]-[18].

Thirdly, by expanding ZNN design formula (2), the following ZNN model could be established:

$$A(t)\dot{X}(t) = -\dot{A}(t)X(t) - \Gamma\mathcal{F}\big(A(t)X(t) - I\big), \quad (3)$$

where $X(t)$, starting from initial condition $X(0) := X_0 \in R^{n\times n}$, is the activation state matrix corresponding to the time-varying theoretical inverse $A^{-1}(t)$, $\dot{X}(t)$ and $\dot{A}(t)$ denote the time-derivatives of $X(t)$ and $A(t)$, respectively. It is worth mentioning that the matrix-valued design-parameter $\Gamma \in R^{n\times n}$ could be set as large as the hardware permits [23] or selected appropriately for experimental purpose [13], [15], [17]. For simplicity, we could assume $\Gamma = \gamma I$ with the scalar-valued design parameter $\gamma > 0$. In addition, by considering the constant matrix inversion only [i.e., the time-derivative $\dot{A}(t) \equiv 0$], ZNN model (3) reduces to

$$A\dot{X}(t) = -\gamma\mathcal{F}\big(AX(t) - I\big). \quad (4)$$

Evidently, the above two continuous-time ZNN models are depicted in implicit dynamics [10]-[12], [15], [17], instead of explicit dynamics [1], [7], [8].

From the analysis results of [10]-[12], [15], [17], the following general observation on global exponential convergence can be summarized and presented for ZNN model (4).

Lemma 1: *Given a nonsingular matrix $A \in R^{n\times n}$, if a monotonically-increasing odd function array $\mathcal{F}(\cdot)$ is used, the state matrix $X(t)$ of ZNN model (4), starting from any initial state $X_0 \in R^{n\times n}$, converges to the theoretical inverse A^{-1} of matrix A. Moreover, if linear activation function $f(u) = u$ is used, the global exponential convergence with rate γ could be achieved for ZNN model (4).*

In addition, for the purpose of possible hardware implementation via digital circuits, it might be more preferable to discretize the continuous-time ZNN model (4) by employing the Euler forward-difference formula [15]-[17], $\dot{X}(t = k\tau) \approx (X((k+1)\tau) - X(k\tau))/\tau$, where $\tau > 0$ denotes the sampling gap, and $k = 0, 1, 2, \cdots$. In general, we have $X_k = X(t = k\tau)$ for presentation convenience [24]. Thus the presented ZNN model (4) could be reformulated as

$$AX_{k+1} = AX_k - h\mathcal{F}(AX_k - I),$$

where X_k corresponds to the kth iteration/sampling of $X(t = k\tau)$, and $h = \tau\gamma > 0$ is the step-size which should be selected appropriately for the convergence of X_k to the theoretical inverse A^{-1}. As matrix A is nonsingular, the above-mentioned implicit discrete-time ZNN model is

$$X_{k+1} = X_k - hA^{-1}\mathcal{F}(AX_k - I), \quad (5)$$

In view of the fact that the state $X(t)$ of ZNN model (4) converges to A^{-1}, the state X_k of discrete-time ZNN model (5) could be very close to A^{-1} after an appropriate number of iterations. Thus, A^{-1} in (5) could be replaced with X_k, which yields the following explicit equation [15]-[17]:

$$X_{k+1} = X_k - hX_k\mathcal{F}(AX_k - I). \quad (6)$$

When linear activation function $f(u) = u$ is used, the discrete-time ZNN model (6) reduces to

$$X_{k+1} = X_k - hX_k(AX_k - I). \quad (7)$$

In addition, for $h \equiv 1$, ZNN (7) could further reduce to

$$X_{k+1} = 2X_k - X_kAX_k, \quad (8)$$

which is exactly the Newton iteration for matrix inversion [15]-[17]. In other words, ZNN model (6) incorporates Newton iteration (8) as a special case for constant matrix inversion. For Newton iteration [as well as for ZNN models (6) and (7)], we have the following lemma [15], [17].

Lemma 2: *For a general constant nonsingular matrix $A \in R^{n\times n}$, we could choose initial state $X_0 = \alpha A^T$ with $\alpha = 2/tr(AA^T)$ for ZNN model (8) which, starting with $\|AX_0 - I\|_2 < 1$, quadratically converges to the inverse A^{-1}.*

III. New ZNN Iterations Aided with Line-Search Algorithm

In this section, more discrete-time ZNN models, which are different from Newton iteration (8) and ZNN models (6)-(7), are presented by employing more equally-spaced multiple-point backward-difference formulas [21]. Furthermore, the line-search algorithm [6], [22] is also used to obtain the appropriate step-size h for fast convergence.

A. More Iterations Generalized from ZNN Models

Based on the polynomial-interpolation theory [25], the Lagrange interpolating polynomial could be constructed by using corresponding discrete-time values of a target function. Then, the approximate first-order numerical-differentiation formulas could be derived in terms of multiple sampling-nodes [21]. In addition, the relatively high computational-precision of the numerical-differentiation approximation to $\dot{X}(t)$ could be achieved via the equally-spaced difference formulas involving multiple sampling-nodes [21]. To the best of the authors' knowledge, the approximation-precision order of the numerical-differentiation could be $\boldsymbol{O}(\tau^{n-1})$ [21], where τ is the sampling period of time, and n denotes the number of sampling-nodes. It is worth noting that the aforementioned Euler forward-difference formula $\dot{X}(t = k\tau) \approx (X((k+1)\tau) - X(k\tau))/\tau$ only has the precision order $\boldsymbol{O}(\tau)$ for approximating $\dot{X}(t)$ [which might imply that the approximation-precision order of Newton iteration (8) is $\boldsymbol{O}(\tau)$, as compared with the original continuous-time ZNN dynamics (4)]. To obtain higher approximation-precision [e.g., order $\boldsymbol{O}(\tau^2)$] and to further investigate ZNN models, many more new ZNN iterations generalized from continuous-time ZNN model (4) are presented below via equally-spaced multiple-point backward-difference formulas.

According to [21], [25], the equally-spaced three-point backward-difference formula can be written as

$$\begin{aligned}&\dot{X}((k+2)\tau)\\&\approx \frac{3X((k+2)\tau) - 4X((k+1)\tau) + X(k\tau)}{2\tau},\end{aligned} \quad (9)$$

where τ is the sampling gap, and $k = 0, 1, 2, \cdots$. It is worth mentioning here that $\dot{X}(t = (k+2)\tau)$ denotes the current state value for presentation convenience. As seen from (9), it has a precision order $\boldsymbol{O}(\tau^2)$ for approximating the corresponding derivative $\dot{X}(t)$ [25]. This means that formula (9) might be (much) better than the Euler forward-difference formula when we derive the discrete-time ZNN models. Thus we have the following proposition.

Proposition 1: Given a nonsingular matrix $A \in R^{n\times n}$, if a monotonically-increasing odd function array $\mathcal{F}(\cdot)$ is used, a new iteration model (involving three sampling-nodes and with two initial states X_0 and X_1) generalized form ZNN model (4) for the inverse of matrix A could be written as

$$X_{k+2} = \frac{4}{3}X_{k+1} - \frac{1}{3}X_k - hX_{k+1}\mathcal{F}\left(AX_{k+1} - I\right), \quad (10)$$

where $k = 0, 1, 2, \cdots$. $h > 0$ denotes a suitable step-size.

Proof: From the continuous-time ZNN model (4), we have

$$A\dot{X}\big((k+2)\tau\big) = -\gamma\mathcal{F}\Big(AX\big((k+2)\tau\big) - I\Big), \quad (11)$$

where k, τ, γ, $\mathcal{F}(\cdot)$ and I are defined as before. By using the presented backward-difference formula (9), we could discretize the continuous-time ZNN model (11) as

$$A(3X_{k+2} - 4X_{k+1} + X_k) = -2\gamma\tau\mathcal{F}\left(AX_{k+2} - I\right), \quad (12)$$

where $X_{k+2} := X((k+2)\tau)$ with $k = 0, 1, 2, \cdots$.

As matrix A is nonsingular (i.e., invertible), the discrete-time ZNN model (12) could be written in the following form:

$$3X_{k+2} = 4X_{k+1} - X_k - 2\gamma\tau A^{-1}\mathcal{F}\left(AX_{k+2} - I\right). \quad (13)$$

In view of the convergence of $X(t)$ to A^{-1} in (11) based on Lemma 1 [in other words, $X(t) \approx A^{-1}$ for time instant t being large enough], we may simply replace A^{-1} in ZNN model (13) with X_{k+2}. Then, we have

$$3X_{k+2} = 4X_{k+1} - X_k - 2\gamma\tau X_{k+2}\mathcal{F}\left(AX_{k+2} - I\right). \quad (14)$$

However, X_{k+2} in (14) is the unknown matrix to be iteratively solved. In order to make (14) computable, similar to the idea of Gauss-Seidel method [25], it seems reasonable that X_{k+2} on the right hand side of (14) be replaced by X_{k+1}. Thus, discrete-time ZNN model (14) becomes

$$X_{k+2} = \frac{4}{3}X_{k+1} - \frac{1}{3}X_k - hX_{k+1}\mathcal{F}\left(AX_{k+1} - I\right), \quad (15)$$

where $h = 2\gamma\tau/3 > 0$ denotes the step-size, which would be selected appropriately via the line-search algorithm [6], [22] (presented in the ensuing subsection) for the convergence to the theoretical inverse A^{-1}. Thus the proof is complete. □

For presentation convenience, (15) is called in this paper a three-point ZNN iteration model. In addition, if the linear activation function $f(u) = u$ is used, the three-point discrete-time ZNN model (15) reduces to

$$X_{k+2} = \frac{4}{3}X_{k+1} - \frac{1}{3}X_k - hX_{k+1}\left(AX_{k+1} - I\right). \quad (16)$$

Evidently, the three-point ZNN iteration model (16) [as a sub-case of (15)] is different from Newton iteration (8), in the sense that (16) is a three-point iteration while Newton iteration (8) is two-point iteration. It is worth mentioning here that, the two initial states X_0 and X_1 of the presented three-point ZNN iteration model (16) [or (15)] could be chosen respectively as $X_0 = 2A^T/tr(AA^T)$ [15], [17] and $X_1 = 2X_0 - X_0AX_0$ [of which the latter, for simplicity, is generated directly from Newton iteration (8) with $k = 0$].

Furthermore, by employing more equally-spaced multiple-point backward-difference formulas [21], we could have the following proposition for constant matrix inversion.

Proposition 2: Given a nonsingular matrix $A \in R^{n\times n}$, if a monotonically-increasing odd function array $\mathcal{F}(\cdot)$ is used, many more iteration models (involving multiple sampling-points) generalized form ZNN model (4) for the inverse of matrix A could be achieved, such as

- the four-point ZNN iteration model (or termed the generalized four-point Newton iteration)

$$X_{k+3} = \frac{18}{11}X_{k+2} - \frac{9}{11}X_{k+1} + \frac{2}{11}X_k - hX_{k+2}\mathcal{F}\left(AX_{k+2} - I\right),$$

- the five-point ZNN iteration model (or termed the generalized five-point Newton iteration)

$$X_{k+4} = \frac{48}{25}X_{k+3} - \frac{36}{25}X_{k+2} + \frac{16}{25}X_{k+1} - \frac{3}{25}X_k - hX_{k+3}\mathcal{F}(AX_{k+3} - I),$$

- the six-point ZNN iteration model (or termed the generalized six-point Newton iteration)

$$X_{k+5} = \frac{300}{137}X_{k+4} - \frac{300}{137}X_{k+3} + \frac{200}{137}X_{k+2} - \frac{75}{137}X_{k+1} + \frac{12}{137}X_k - hX_{k+4}\mathcal{F}(AX_{k+4} - I),$$

- and the seven-point ZNN iteration model (or termed the generalized seven-point Newton iteration)

$$X_{k+6} = \frac{360}{147}X_{k+5} - \frac{450}{147}X_{k+4} + \frac{400}{147}X_{k+3} - \frac{225}{147}X_{k+2} + \frac{72}{147}X_{k+1} - \frac{10}{147}X_k - hX_{k+5}\mathcal{F}(AX_{k+5} - I).$$

Proof: Following the similar proof procedure of Proposition 1, with the aid of the equally-spaced multiple-point backward-difference formulas shown in Appendix. □

The above-mentioned multiple-point ZNN iterations [including three-point iteration (16)], which are generalized from continuous-time ZNN model (4), are clearly different from Newton iteration (8). Before ending this subsection, it is worth mentioning that the initial values of the above multiple-point ZNN iteration models could normally be chosen via Newton iteration (8) starting from $X_0 = 2A^T/tr(AA^T)$.

B. Line-Search Algorithm

As mentioned before, the step-size h should be selected appropriately for the convergence of X_k to the theoretical inverse A^{-1}. In this subsection, the line-search algorithm [6], [22] is presented to obtain the appropriate step-size value of h for the discrete-time ZNN models [e.g., (16)].

The line-search algorithm is mainly about the determination of the search-direction and the appropriate step-size. In other words, in each iteration, a search direction is obtained first, and then different step lengths are tried along the direction for a better solution point [6], [22]. The line-search algorithm employed in this paper could thus be divided into the following two main steps.

Step 1. Search-direction determination: According to ZNN design formula (2) [10]-[12], [15], [17], we could obtain the search-direction as the convergence-direction of the matrix-valued error $E = AX(t) - I$. This would guarantee that the state matrix $X(t)$ of (4) [as well as (16)] converges to the theoretical inverse A^{-1} . Therefore, the step-size h could be searched along this convergence-direction for the above presented multiple-point ZNN iteration models.

Step 2. Appropriate step-size determination: Along the search-direction determined in Step 1, the appropriate step-size h is obtained by trying different step lengths via multiplication and/or division technique. That is, when the error E decreases, the step-size is multiplied by a real number $\eta > 1$; when E increases, the step-size is divided by $\eta > 1$. For example, η could be 2 or $2n$, where n is the dimension of matrix A. Note that, in this paper, we prefer the latter; i.e., $\eta = 2n$ is selected.

In detail, the following computation procedure is used to obtain the appropriate step-size in the kth iteration for the three-point ZNN iteration model (16), as an example.

- If the kth iteration-error $\|E_{k+1}\|_F$ decreases (i.e., $\|E_{k+1}\|_F < \|E_k\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm [14]), the current state-values $\|E_{k+1}\|_F$, X_{k+1} and h are saved, and then step-size h is multiplied by η. The computation procedure used to determine the appropriate step-size h could be described as follows.

```
While 1
    h = h × η
    Calculate X_{k+1}, E_{k+1}
     if E_{k+1} ⩽ bestERR
        Save to bestERR, bestX, bestH
     else
        Restore bestERR, bestX, bestH
        break
     end
  end
```

- If $\|E_{k+1}\|_F$ increases (i.e., $\|E_{k+1}\|_F \geqslant \|E_k\|_F$), the step-size h is divided by $\eta > 1$. It is worth mentioning that, when the step-size h is divided by η for too many times, it would make $h \to 0$ and may yield wrong information. Thus, to avoid this undesired result, we restore the already saved best values "bestERR", "bestX", "bestH" if $h \leqslant 10^{-20}$. The computation procedure could be shown in the box on next column.

```
While 1
    h = h ÷ η
    Calculate X_{k+1}, E_{k+1}
     if E_{k+1} ⩽ bestERR
        Save to bestERR, bestX, bestH
        break
     end
     if h ⩽ 10^{-20}
        Restore bestERR, bestX, bestH
        break
     end
  end
```

Following the above discussion and analysis, the appropriate step-size h could be obtained in each iteration for the three-point ZNN iteration model (16) (as well as other multiple-point ZNN iteration models). In the ensuing section, four different-dimensional matrices are simulated to demonstrate the performance of the presented ZNN iteration model (16) aided with line-search algorithm.

IV. Illustrative Simulation Results

For illustrative purposes, computer-verification examples are conducted and discussed to show the convergence of the presented three-point discrete-time ZNN model (16) aided with line-search algorithm for online matrix inversion. For comparison, Newton iteration (8) is simulated as well.

Let us consider the matrix inversion problem $AX = I$ with the following matrix $A \in R^{3\times 3}$:

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

By simple algebraic manipulations, we could obtain the theoretical inverse as

$$A^{-1} = \begin{bmatrix} 1.333 & 5.000 & -1.667 \\ 2.667 & 11.000 & -3.333 \\ 1.667 & 7.000 & -2.333 \end{bmatrix},$$

which is given for comparative purposes (or to say, to check the correctness of neural-network solutions).

For the simulation of ZNN (16), the two initial states (i.e., X_0 and X_1) and the initial step-size h are chosen respectively as $X_0 = 2A^T/tr(AA^T)$, $X_1 = 2X_0 - X_0AX_0$ and $h = 1$. In addition, the real-number $\eta = 2n$ is set for adjusting the value of h. Fig. 1 illustrates the state matrix X_i (i.e., the ith state-value, where $i = 0, 1, 2, \cdots$) of ZNN (16) aided with line-search algorithm. It can be seen that the neural-network state X_i of the three-point ZNN iteration model (16) converges to the theoretical inverse A^{-1}. For comparison, Newton iteration (8) is employed as well to solve the same matrix-inversion problem with the initial state $X_0 = 2A^T/tr(AA^T)$. Its convergent performance could also be seen in Fig. 1, which shows that the neural state X_i of Newton iteration (8) could also converge to the theoretical inverse A^{-1}. It is worth noting here that, at a beginning stage [e.g., $i \in (0, 15)$], ZNN iteration model (16) has (much) superior convergence, as compared to Newton iteration (8). This could also be seen in Fig. 2, which illustrates the residual-errors $\|AX_i - I\|_F$ of the above two models [i.e., ZNN iteration model (16) and Newton iteration (8)].

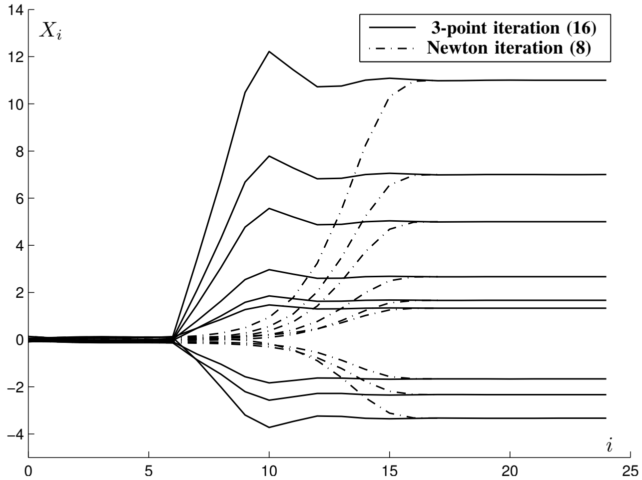


Fig. 1. State trajectories synthesized by three-point ZNN iteration model (16) and Newton iteration (8) for constant matrix inversion.

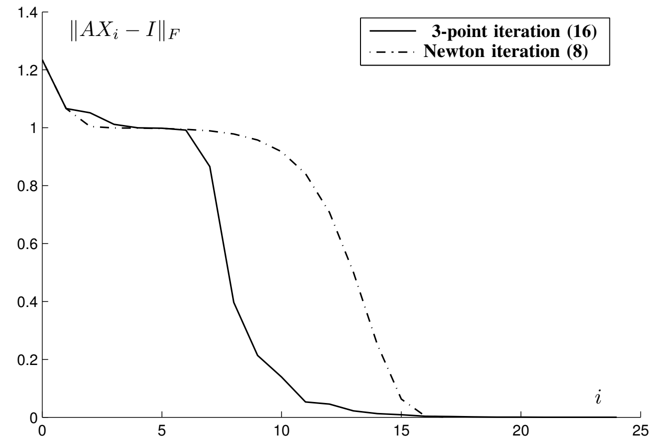


Fig. 2. Comparison of residual-errors $\|AX_i - I\|_F$ synthesized by three-point ZNN model (16) and Newton iteration (8) for matrix inversion.

To further investigate ZNN iteration model (16) and its performance for matrix inversion, illustrative examples based on three randomly-generated matrices with different dimensions (i.e., $A \in R^{5\times5}$, $R^{30\times30}$ or $R^{100\times100}$) are conducted. The simulation-results are shown in Fig. 3, which further demonstrates the superior convergence of ZNN iteration model (16) aided with a line-search algorithm.

V. CONCLUSIONS

For the purpose of hardware (e.g., digital-circuit) realization, from the continuous-time ZNN model (4) we could generalize the discrete-time ZNN models [e.g., (6)], which incorporates Newton iteration (8) as a special case. By employing more equally-spaced multiple-point backward-difference formulas, the new multiple-point ZNN iterations have been developed and investigated for online constant matrix inversion. In addition, to obtain the appropriate step-size, the line-search algorithm is employed. Computer-simulation results via different-dimensional matrices have demonstrated the effectiveness of the presented ZNN iterations aided with a line-search algorithm (which may thus be termed the generalized multiple-point Newton iterations).

APPENDIX

It follows from [21] that the equally-spaced multiple-point backward-difference formulas could be formulated as follows (which involve three to seven sampling-points):

- the three-point backward-difference formula

$$\dot{X}((k+2)\tau) \approx \frac{1}{2\tau}\Big(3X((k+2)\tau) - 4X((k+1)\tau) + X(k\tau)\Big),$$

- the four-point backward-difference formula

$$\dot{X}((k+3)\tau) \approx \frac{1}{6\tau}\Big(11X((k+3)\tau) - 18X((k+2)\tau) + 9X((k+1)\tau) - 2X(k\tau)\Big),$$

- the five-point backward-difference formula

$$\dot{X}((k+4)\tau) \approx \frac{1}{12\tau}\Big(25X((k+4)\tau) - 48X((k+3)\tau) + 36X((k+2)\tau) - 16X((k+1)\tau) + 3X(k\tau)\Big),$$

- the six-point backward-difference formula

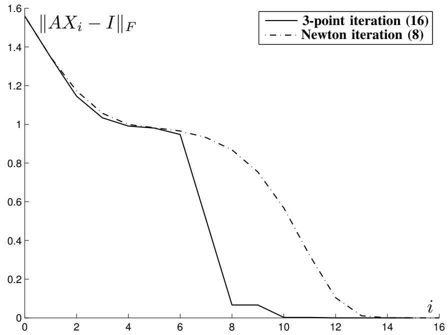
$$\dot{X}((k+5)\tau) \approx \frac{1}{60\tau}\Big(137X((k+5)\tau) - 300X((k+4)\tau) + 300X((k+3)\tau) - 200X((k+2)\tau) + 75X((k+1)\tau) - 12X(k\tau)\Big),$$

- and the seven-point backward-difference formula

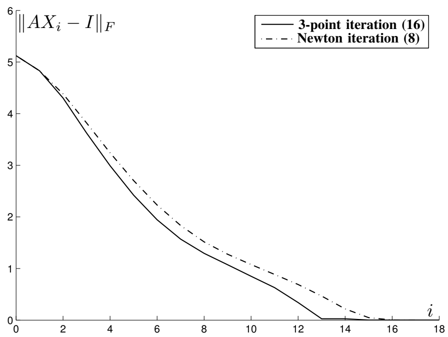
$$\dot{X}((k+6)\tau) \approx \frac{1}{60\tau}\Big(147X((k+6)\tau) - 360X((k+5)\tau) + 450X((k+4)\tau) - 400X((k+3)\tau) + 225X((k+2)\tau) - 72X((k+1)\tau) + 10X(k\tau)\Big).$$

ACKNOWLEDGMENTS

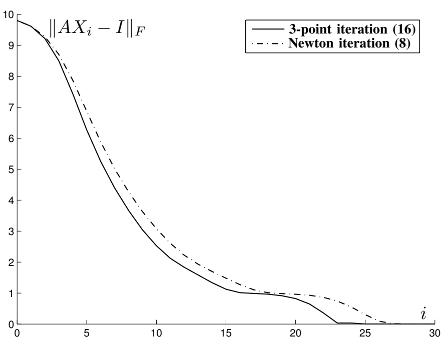
With many thanks to the school, university and government for their generous supports, Yunong Zhang, the corresponding author, would also like to express his gratitude to the National University of Ireland at Maynooth, the University of Strathclyde, the National University of Singapore, the Chinese University of Hong Kong, the South China University of Technology, and the Huazhong University of Science and Technology, which Yunong had been with from 1992 to 2006. Yunong was also supported by those research fellowships, assistantship and studentships, related to this research explicitly or implicitly. His web-page is now at http://sist.sysu.edu.cn/~zhynong/. In addition to the above thanks, Yunong would like to share the following thoughts with research colleagues and students: "A generation of people may only do things of that generation".



(a) Residual-errors for $A \in R^{5\times 5}$



(b) Residual-errors for $A \in R^{30\times 30}$



(c) Residual-errors for $A \in R^{100\times 100}$

Fig. 3. Comparison of residual-errors synthesized by ZNN iteration model (16) and Newton iteration (8) for matrix inversion with different dimensions.

REFERENCES

[1] Y. Zhang, "Revisit the analog computer and gradient-based neural system for matrix inversion," in *Proceedings of IEEE International Symposium on Intelligent Control*, Limassol, June 2005, pp. 1411-1416.

[2] R. H. Sturges Jr, "Analog matrix inversion," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 157-162, April 1988.

[3] Y. Zhang and H. Peng, "Zhang neural network for linear time-varying equation solving and its robotic application," in *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, HongKong, August 2007, pp. 3543-3548.

[4] Y. Zhang, W. E. Leithead, and D. J. Leith, "Time-series Gaussian process regression based on toeplitz computation of O(N^2) operations and O(N)-level storage," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, December 2005, pp. 3711-3716.

[5] R. J. Steriti and M. A. Fiddy, "Regularized image reconstruction using SVD and a neural network method for matrix inversion," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 3074-3077, October 1993.

[6] W. E. Leithead and Y. Zhang, "O(N^2)-operation approximation of covariance matrix inverse in Gaussian process regression based on quasi-Netwon BFGS method," *Communications in Statistics - Simulation and Computation*, vol. 36, no. 2, pp. 367-380, March 2007.

[7] S. J. Yoo, Y. H. Choi, and J. B. Park, "Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 53, no. 6, pp. 1381-1394, June 2006.

[8] J. J. Rubio and W. Yu, "Stability analysis of nonlinear system identification via delayed neural networks," *IEEE Transactions on Circuits and Systsem-II: Express Briefs*, vol. 54, no. 2, pp. 161-165, February 2007.

[9] J. -S. Wang and Y. -P. Chen, "A fully automated recurrent neural network for unknown dynamic system identification and control," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 6, pp. 1363-1372, June 2006.

[10] Y. Zhang and S. S. Ge, "A general recurrent neural network model for time-varying matrix inversion," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, December 2003, pp. 6169-6174.

[11] Y. Zhang and S. S. Ge, "Design and analysis of a general recurrent neural network model for time-varying matrix inversion," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1477-1490, November 2005.

[12] Y. Zhang, D. Jiang, and J. Wang, "A recurrent neural network for solving Sylvester equation with time-varying coefficients," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1053-1063, September 2002.

[13] Y. Zhang, X. Guo, W. Ma, K. Chen, and B. Cai, "MATLAB Simulink modeling and simulation of Zhang neural network for online time-varying matrix inversion," in *Proceedings of IEEE International Conference on Networking, Sensing and Control*, Sanya, April 2008, pp. 1480-1485.

[14] Y. Zhang, Z. Chen, K. Chen, and B. Cai, "Zhang neural network without using time-derivative information for constant and time-varying matrix inversion," in *Proceedings of IEEE International Joint Conference on Neural Networks*, HongKong, June 2008, pp. 142-146.

[15] Y. Zhang, W. Ma, and C. Yi, "The link between Newton iteration for matrix inversion and Zhang neural network (ZNN)," in *Proceedings of IEEE Conference on Industrial Technology*, Chengdu, April 2008, pp. 1-6.

[16] Y. Zhang, B. Cai, M. Liang, and W. Ma, "On the variable step-size of discrete-time Zhang neural network and Newton iteration for constant matrix inversion," in *Proceedings of the 2nd International Symposium on Intelligent Information Technology Application*, Shanghai, December 2008, vol 1, pp. 34-38.

[17] Y. Zhang, W. Ma, and B. Cai, "From Zhang neural network to Newton iteration for matrix inversion," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 56, no. 7, pp. 1405-1415, July 2009.

[18] W. Ma, Y. Zhang, and J. Wang, "MATLAB Simulink modeling and simulation of Zhang neural networks for online time-varying sylvester equation solving," in *Proceedings of IEEE International Joint Conference on Neural Network*, HongKong, June 2008, pp. 285-289.

[19] H. Huang, D. W. C. Ho, and J. Lam, "Stochastic stability analysis of fuzzy Hopfield neural networks with time-varying delays," *IEEE Transactions on Circuits and SystemsłII: Express Briefs*, vol. 52, no. 5, pp. 251-255, May 2005.

[20] Y. Zhang, Z. Li, K. Chen, and B. Cai, "Common nature of learning exemplified by BP and Hopfield neural networks for solving online a system of linear equations," in *Proceedings of IEEE International Conference on Networking, Sensing and Control*, Sanya, April 2008, pp. 832-836.

[21] Y. Zhang, D. Guo, S. Xu, and H. Li, "Verification and practice on first-order numerical differentiation formulas for unknown target functions," *Journal of Gansu Sciences*, vol. 21, no. 1, pp. 13-18, March 2009.

[22] Y. Zhang and W. E. Leithead, "Exploiting Hessian matrix and trust-region algorithm in hyperparameters estimation of Gaussian process," *Applied Mathematics and Computation*, vol. 171, no. 2, pp. 1264-1281, March 2005.

[23] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.

[24] S. K. Mitra, *Digital Signal Processing-A Computrer-Based Approach, 3rd ed.* Beijing: Tsinghua University Press, 2006.

[25] J. H. Mathews and K. K. Fink, *Numerical Methods Using Matlab, 4th ed.* Beijing: Publishing House of Electronics Industry, 2005.