Solution of Nonlinear Equations by Continuous- and Discrete-time Zhang Dynamics and More Importantly Their Links to Newton Iteration

Yunong Zhang[1], Peng Xu[1], and Ning Tan[2]
[1]School of Information Science and Technology [2]School of Software
Sun Yat-Sen University, Guangzhou 510275, China
Emails: zhynong@mail.sysu.edu.cn, ynzhang@ieee.org

***Abstract*—Different from gradient-based dynamics (GD), a special class of neural dynamics has been found, developed, generalized and investigated by Zhang *et al*, e.g., for online solution of time-varying and/or static nonlinear equations. The resultant Zhang dynamics (ZD) is designed based on the elimination of an indefinite error-function (instead of the elimination of a square-based positive or at least lower-bounded energy-function usually associated with GD and/or Hopfield-type neural newtorks). In this paper, discrete-time ZD models (different from our previous research on continuous-time ZD models) are developed and investigated. In terms of nonlinear-equations solving, the Newton iteration (also termed, Newton-Raphson iteration) is found to be a special case of the ZD models (by focusing on the static-problem solving, utilizing the linear activation function and fixing the step-size to be 1). Noticing this new relation and explanation, we conduct computer-simulation, testing and comparisons for such discrete-time ZD models (including Newton iteration) for nonlinear equations solving. The numerical results substantiate the theoretical analysis, explanation, unification and efficacy of the discrete-time ZD models on nonlinear equations solving.**

***Keywords*—neural dynamics; discrete-time model; nonlinear equation; Newton-Raphson iteration; activation function**

I. Introduction

The online solution of nonlinear equations is considered to be a basic issue widely encountered in science and engineering. Many numerical algorithms (e.g., [1]–[5] and references therein) have thus been employed for solving such nonlinear equations, especially since computers were widely used.

In recent decades, due to the in-depth research in artificial neural networks, various dynamic and analog solvers in the form of neural-dynamics and recurrent neural nets have been proposed, developed, investigated and implemented on specific architectures (see [6]–[10] and references therein). Suitable for analog VLSI implementation [6] and in view of distributed parallel-processing properties, the neural-dynamic (ND) approach is now regarded as a powerful alternative for online problems solving [7]–[10]. However, it is worth mentioning again that most reported computational-schemes are related to gradient methods and/or designed theoretically for solving time-invariant (or termed, static) problems [6]–[10], which may be applied directly to the time-varying environments.

Different from the above gradient-based dynamic (GD) approach, a new class of neural-dynamics has recently been proposed by Zhang *et al* [11]–[15] (formally since March 2001) for time-varying problems solving (e.g., time-varying Sylvester equation solving, matrix inversion and optimization). By following and generalizing the Zhang *et al*'s design method, a neural-dynamics has recently been introduced to handle nonlinear equations [16], [17]. The proposed Zhang dynamics (or termed, ZD, for presentation convenience) is designed based on the elimination of an indefinite error-function (instead of the elimination of a square-based positive or at least lower-bounded energy-function usually associated with conventional gradient-based, Hopfield-type and/or Lyapunov design-&-analysis methods [1]–[3], [6]–[17]).

In this paper, as an extension, link and also a breakthrough of our previous work, the discrete-time ZD models are developed and investigated for such nonlinear-equations' solving. Comparing with the Newton-Raphson iteration for solving the same equation problem, we find that the Newton-Raphson iteration is a special case of the ZD models when focusing on the static-problem solving, utilizing the linear activation function and fixing the step-size to be 1. By doing so, a new reasonable explanation about the origin of Newton iteration is given, which differs very much from the standard textbook explanations – geometric construction and Taylor expansion [1]–[3], [18], [19]. Noticing this relation and new explanation, we conduct in this paper computer-simulations, testing and comparisons about the discrete-time ZD models (including Newton iteration) for nonlinear equations solving.

II. Problems and Continuous-Time ZD Solvers

Our main purpose in this paper is, firstly, to solve the following general form of nonlinear equation (which is also the conventional problem formulation in textbooks and papers):

$$f(x) = 0, \tag{1}$$

where $x \in R$, and $f(\cdot) : R \to R$ is assumed a continuously-differentiable function [16]. For ease of presentation, let x^* denote a theoretical solution (or termed, root, zero) of nonlinear equation (1). Moreover, it is worth pointing out that, as usual and before, our original purpose in this kind of ZD research [17] is about the more general dynamic form of (1); i.e., the following time-varying nonlinear equation (which, to the best

978–1–4244–4657–5/09/$25.00 © 2009 IEEE

knowledge of the authors, seldom appears in literature):

$$f\big(x(t), t\big) = 0. \tag{2}$$

In the ensuing parts, we briefly review the continuous-time ZD models for solving such nonlinear-equations (1) and (2). This could make readers well prepared for the derivation of discrete-time ZD models (including Newton iteration as a special case) in the following section. Due to space limitation, we only show below the differential equations of continuous-time ZD models but with their detailed derivation procedures omitted (see [16], [17] if necessary and/or if interested).

To solve the nonlinear equation (1), by following Zhang *et al*'s neural-dynamic design method [11]–[17], the following ZD could be established (with $t \in [0, +\infty)$ being the time parameter, $f'(x) := df(x)/dx$ here, and $\dot{x} := dx/dt$):

$$f'(x)\dot{x}(t) = -\gamma\phi\big(f(x)\big) \quad \text{or,} \quad \dot{x}(t) = -\gamma\frac{\phi\big(f(x)\big)}{f'(x)}. \tag{3}$$

On the other hand, to solve the time-varying nonlinear equation (2), the following ZD [17] could be established (with $f'_x := \partial f(x,t)/\partial x$ and $f'_t := \partial f(x,t)/\partial t$ here):

$$\begin{aligned} &f'_x\dot{x}(t) = -\Big(\gamma\phi\big(f(x(t),t)\big) + f'_t\Big) \text{ or,} \\ &\dot{x}(t) = -\Big(\gamma\phi\big(f(x(t),t)\big) + f'_t\Big)/f'_x. \end{aligned} \tag{4}$$

The neural state $x(t)$ in ZD (3) or (4), which could start from randomly-generated initial condition $x(0) \in R$, corresponds to the online solution of theoretical root x^* of nonlinear equation (1) [or theoretical time-varying root $x^*(t)$ of time-varying nonlinear equation (2)]. Moreover, in ZD (3) and (4), the design-parameter $\gamma > 0$ is used to scale the convergence rate of the ZD solution. In analog circuits or VLSI [6], [20], the parameter γ, being the reciprocal of a capacitance parameter, can be set as large as hardware permits or selected appropriately (e.g., between 10^3 and 10^8) for simulation purposes. In addition, activation-function $\phi(\cdot) : R \to R$ in ZD (3) or (4) should be monotonically-increasing and odd. In this paper, the following three types of $\phi(\cdot)$ are recommended [7], [9], [11]–[17], [19]:

i) widely-investigated linear activation function $\phi(u) = u$;

ii) BP-type bipolar-sigmoid function (with $\xi > 2 \in R$)

$$\phi(u) = \frac{1 - \exp(-\xi u)}{1 + \exp(-\xi u)};$$

iii) our preferred power-sigmoid activation function

$$\phi(u) = \begin{cases} u^p, & \text{if } |u| \geqslant 1 \\ \frac{1+\exp(-\xi)}{1-\exp(-\xi)} \cdot \frac{1-\exp(-\xi u)}{1+\exp(-\xi u)}, & \text{otherwise} \end{cases} \tag{5}$$

with $\xi \geqslant 2 \in R$ and odd integer $p \geqslant 3$ (e.g., 3,5,7,$\cdots$).

Evidently, as for differential equations (3) and (4) of Zhang dynamics, different values of parameter γ and different types of activation function $\phi(\cdot)$ lead to different ZD performance. Moreover, to lay the necessary basis of further investigation on discrete-time ZD models (and, more importantly, their link to Newton iteration [21]–[23]), we present the following lemma [16] about ZD (3) which solves nonlinear equation (1).

Lemma 1. Consider a solvable nonlinear equation $f(x) = 0$, where $f(\cdot)$ is continuously differentiable $\{$denoted by $f' \in C[x^* - \delta, x^* + \delta]$, i.e., with at least the first-order derivative continuous at some interval $[x^* - \delta, x^* + \delta]$ containing $x^*\}$. If a monotonically-increasing odd activation function $\phi(\cdot)$ is employed, then the neural state $x(t)$ of ZD (3), starting from randomly-generated initial condition $x(0) \in R$, could converge to theoretical root x^* of static nonlinear equation $f(x) = 0$ [of which the specific value of x^*, in the situation of not less than two zeros existing, depends on the sufficient closeness of initial state $x(0)$ to $x^* \in R$].

III. Discrete-Time ZD Solvers

For possible hardware (digital circuits) implementation of Zhang dynamics to solve nonlinear equation (1), we could discretize the continuous-time ZD model (3) by using Euler forward-difference rule [1]–[3], [22]:

$$\dot{x}(t) \approx (x_{k+1} - x_k)/\tau,$$

where $\tau > 0$ denotes the sampling gap, and x_k denotes the kth iteration/sampling of $x(t = k\tau)$ with $k = 1, 2, \cdots$. The discrete-time ZD solver, resulting from (3), is thus generated:

$$(x_{k+1} - x_k)/\tau = -\gamma\frac{\phi\big(f(x_k)\big)}{f'(x_k)},$$

which could be rewritten further as

$$x_{k+1} = x_k - h\frac{\phi\big(f(x_k)\big)}{f'(x_k)}, \tag{6}$$

where parameter $h := \tau\gamma > 0$ denotes the step-size (or termed, learning rate, step-length) which should be set appropriately for the convergence of (6) to theoretical root x^* of nonlinear equation (1). Once more, about discrete-time ZD solver (6), different values and choices of step-size h and activation function $\phi(\cdot)$ lead to different performance. We will investigate.

A. Link and New Explanation to Newton-Raphson Iteration

Let us look at discrete-time ZD solver (6): if linear activation function $\phi(u) = u$ is used, ZD solver (6) reduces to

$$x_{k+1} = x_k - h\frac{f(x_k)}{f'(x_k)}; \tag{7}$$

and, if $h \equiv 1$, discrete-time ZD solver (7) becomes further

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \tag{8}$$

which is exactly the Newton-Raphson iteration presented in textbooks and literature (see [1]–[3], [18], [19] and references therein) for solving nonlinear equation (1)! In other words, we have discovered (again in view of [22], [23]) that a general form of Newton iteration for solving nonlinear equation (1) can be given by discretizing the continuous-time Zhang dynamics (3). Evidently, this paper shows another (new) explanation to Newton iteration; i.e., the Newton iteration (also termed Newton-Raphson iteration usually) is one of the special cases of Zhang dynamics (by focusing on the static equation solving, discretizing it via Euler forward difference, using a linear activation function and fixing the step-size to be 1).

However, traditionally, the Newton iteration for nonlinear equation solving has a standard textbook explanation; i.e., via Taylor expansion [1]–[3], [18], [19]. For readers' convenience, such a derivation/construction/explanation procedure is repeated and compared below [1]–[3], [18].

According to the Taylor expansion of nonlinear function $f(x)$ in (1) around the initial point of $x_0 \in R$, we have

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + f''(c)(x - x_0)^2/2, \quad (9)$$

where c is located somewhere between x_0 and x. Let $x = x^*$ in the above equation. Noting that $f(x^*) = 0$, we can get $0 = f(x^*) = f(x_0)+f'(x_0)(x^*-x_0)+f''(c)(x^*-x_0)^2/2$. Assume that x_0 is close enough to x^*, and the last term of the right-hand side of equation (9) is so small (as compared with the sum of the first two terms) that it can be neglected. In this way, the following equation is obtained: $0 \approx f(x_0) + f'(x_0)(x^* - x_0)$, which yields $x^* \approx x_0 - f(x_0)/f'(x_0)$. Letting x_{k+1} denote the next approximation of x^* and then substituting x_k for x_0 into $x^* \approx x_0 - f(x_0)/f'(x_0)$, the Newton iteration (8) is thus derived/constructed/explained approximately.

B. Theoretical Analysis

With the general form of discrete-time ZD solver (6) and its link to Newton iteration presented in the previous subsection, some design consideration, details and theoretical results are given in this subsection (by the fixed-point iteration approach).

Lemma 2. Assume that $g(x), g'(x) \in C[a, b]$, $g(x) \in [a, b]$ for all $x \in [a, b]$, and there exists a positive constant $K < 1$ such that $|g'(x)| \leqslant K < 1$ for all $x \in [a, b]$. Then, $g(x)$ has an attractive unique fixed point $p \in [a, b]$, where, if $g(p) = p$, we call p a fixed point of $g(x)$. [1]–[3].

Theorem. Assume that $f(x)$ of nonlinear equation (1) is sufficiently continuously differentiable on $[x^*-\delta, x^*+\delta]$, with x^* a root of order q (if $q = 1$, x^* is a simple root). When the step-size h of discrete-time ZD solver (6) satisfies $0 < h < 2q/\theta$ [with $\theta := \phi'(0)$], there generally exists such a positive number δ such that $\{x_k\}$ will converge to x^* for any initial state $x_0 \in [x^* - \delta, x^* + \delta]$.

Proof. Following the proof approach of [1]–[3], about discrete-time ZD solver (6), we know that the fix-point iteration function is $g(x) = x - h\phi\big(f(x)\big)/f'(x)$. By applying Lemma 2, the key of the proof is to analyze $g'(x)$:

$$\begin{aligned} g'(x) &= 1 - h\frac{\phi'\big(f(x)\big)\big(f'(x)\big)^2 - \phi\big(f(x)\big)f''(x)}{\big(f'(x)\big)^2} \\ &= 1 - h\phi'\big(f(x)\big) + h\frac{\phi\big(f(x)\big)f''(x)}{\big(f'(x)\big)^2}. \end{aligned} \quad (10)$$

i) If x^* is a simple root, we have $f(x^*) = 0$ and $f'(x^*) \neq 0$. Substituting x^* into (10), in view of $\phi\big(f(x^*)\big) = 0$ (as $\phi(\cdot)$ is odd), we have

$$g'(x^*) = 1 - h\phi'(0) := 1 - h\theta.$$

So, when $0 < h < 2/\theta$, the above expression $1 - h\theta$ satisfies $-1 < 1 - h\theta < 1$, i.e., $|g'(x^*)| < 1$. As $g'(x)$ is continuous at and around the point x^*, it is possible to find a $\delta > 0$ such that $|g'(x)| < 1$ for any $x \in [x^* - \delta, x^* + \delta]$. By Lemma 2, it is then proved [2] that, if initial state $x_0 \in [x^* - \delta, x^* + \delta]$, the sequence $\{x_k\}$ ($k = 0, 1, 2, \cdots$) resulting from the fixed-point iteration function $g(x)$ will converge to x^* [which is a simple root of nonlinear equation $f(x) = 0$].

ii) Let us investigate the situation that x^* is a multiple root of order $q \geqslant 2$; i.e., $f(x^*) = 0$, $f'(x^*) = 0, \cdots, f^{(q-1)}(x^*) = 0$, but $f^{(q)}(x^*) \neq 0$ [1]–[3]. So, we can denote $f(x)$ as $f(x) = (x-x^*)^q m(x)$ with $m(x^*) \neq 0$ [2]. Now let us analyze $g'(x)$ in (10), and then we have

$$\begin{aligned} g'(x^*) &= \lim_{x\to x^*} g'(x) \\ &= 1 - h\phi'(0) + h\lim_{x\to x^*}\frac{\phi\big(f(x)\big)f''(x)}{\big(f'(x)\big)^2}, \end{aligned}$$

of which the last term can be derived (via L'Hopital's rule) as

$$\begin{aligned} &\lim_{x\to x^*}\frac{\phi\big(f(x)\big)f''(x)}{\big(f'(x)\big)^2} = \lim_{x\to x^*}\frac{\phi\big(f(x)\big)\big((x-x^*)^q m(x)\big)''}{\Big(\big((x-x^*)^q m(x)\big)'\Big)^2} \\ &= \lim_{x\to x^*}\phi\big(f(x)\big)\Big[\frac{q(q-1)(x-x^*)^{q-2}m(x)}{\Big(q(x-x^*)^{q-1}m(x) + (x-x^*)^q m'(x)\Big)^2} \\ &\quad + \frac{2q(x-x^*)^{q-1}m'(x) + (x-x^*)^q m''(x)}{\Big(q(x-x^*)^{q-1}m(x) + (x-x^*)^q m'(x)\Big)^2}\Big] \\ &= \lim_{x\to x^*}\frac{\phi\big(f(x)\big)\Big(q - 1 + 2\frac{(x-x^*)m'(x)}{m(x)} + \frac{(x-x^*)^2 m''(x)}{qm(x)}\Big)}{q(x-x^*)^q m(x)\Big(1 + (x-x^*)\frac{m'(x)}{qm(x)}\Big)^2} \\ &= \left(\lim_{x\to x^*}\frac{\phi\big(f(x)\big)}{q(x-x^*)^q m(x)}\right)(q-1) \\ &= (q-1)\lim_{x\to x^*}\frac{\Big(\phi\big(f(x)\big)\Big)'}{\big(q(x-x^*)^q m(x)\big)'} \\ &= (q-1)\lim_{x\to x^*}\Big[\frac{\phi'\big(f(x)\big)q(x-x^*)^{q-1}m(x)}{q^2(x-x^*)^{q-1}m(x) + q(x-x^*)^q m'(x)} \\ &\quad + \frac{\phi'\big(f(x)\big)(x-x^*)^q m'(x)}{q^2(x-x^*)^{q-1}m(x) + q(x-x^*)^q m'(x)}\Big] \\ &= (q-1)\lim_{x\to x^*}\frac{\phi'\big(f(x)\big)\Big(qm(x) + (x-x^*)m'(x)\Big)}{q^2 m(x) + q(x-x^*)m'(x)} \\ &= (q-1)\lim_{x\to x^*}\frac{\phi'\big(f(x)\big)}{q} = \frac{(q-1)\phi'(0)}{q} := \frac{(q-1)\theta}{q}. \end{aligned}$$

Thus, we have $g'(x^*) = 1 - h\theta + h(q-1)\theta/q = 1 - h\theta/q$. So, similar to the above analysis on the simple-root situation, in this q-order multiple-root situation, if step-size h is chosen such that $0 < h < 2q/\theta$, then $|g'(x^*)| < 1$. As $g'(x)$ is continuous at and around the point x^*, it is possible to find a $\delta > 0$ such that $|g'(x)| < 1$ for any $x \in [x^* - \delta, x^* + \delta]$ as well. By Lemma 2, it is proved again that, if initial state $x_0 \in [x^* - \delta, x^* + \delta]$, the sequence $\{x_k\}$ ($k = 0, 1, 2, \cdots$)

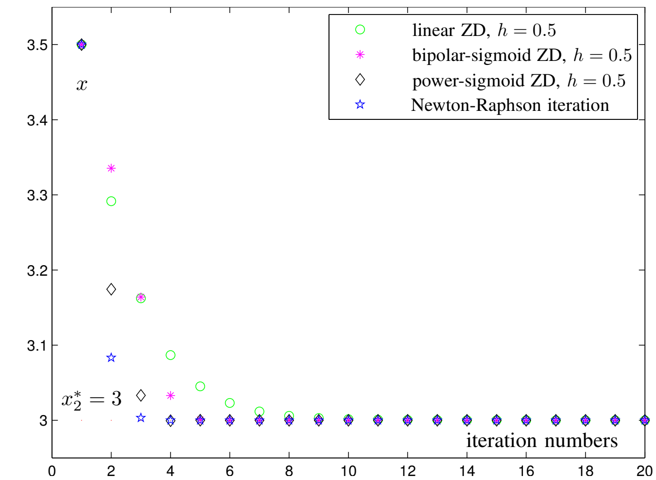

Fig. 1. Convergent performance of ZD (6) using different activation functions with $h = 0.5$ and Newton iteration solving $(x-3)(x-1) = 0$.

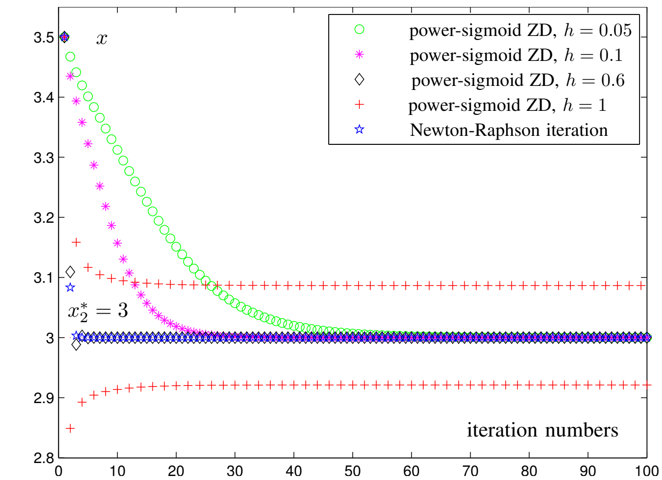

Fig. 2. Convergent performance of ZD (6) using power-sigmoid activation function with different h and Newton iteration solving $(x-3)(x-1) = 0$.

resulting from the fixed-point iteration function $g(x)$ converges to x^* (being a multiple root in the second situation).

By summarizing both situations, the proof is complete. □

Remark. It is worth pointing out that, if the step-size h of discrete-time ZD solver (6) satisfies $0 < h < 2q/\theta$ (where q is the order of a root), there generally exists an initial state x_0 close enough to such a root, which ensures the convergence of discrete-time ZD solver (6) to x^*. The step size h and initial state x_0 need to be selected appropriately to ensure (and better expedite) the convergence: a large step-size may make ZD (6) converge fast to a root of (1), but too large step-sizes may result in divergence. The (optimal) choosing of initial state x_0 is also an interesting issue [1], [22]–[24].

IV. Computer-Simulation, Testing and Comparison

The previous sections have proposed the continuous-time and discrete-time ZD models and their links to Newton iteration for online solution of nonlinear equations. In this section, computer-simulation results and observations are provided for the substantiation of such theoretical results, proposed link (to Newton method) and efficacy of ZD (6) with the aid of suitable initial state x_0, activation function $\phi(\cdot)$ and step-size h.

Example 1. Consider the following nonlinear equation

$$f(x) = (x-3)(x-1) = 0. \tag{11}$$

Evidently, the theoretical root (or termed, zero, solution) of equation (11) could be written as: $x_1^* = 1$ and $x_2^* = 3$ (both of which are simple roots) for the verification of ZD solutions. By applying ZD (6) to solve the above equation (11), the computer-simulation results are shown in Figure 1. In this example, we choose the initial state $x_0 = 3.5$ close to $x_2^* = 3$, and ZD (6) converges to $x_2^* = 3$, in addition to setting step-size $h = 0.5$. From the figure, we find that, after a sufficient number of iterations, ZD (6) could always converge to theoretical root $x_2^* = 3$ even when different activation functions are used. In addition, when $h \equiv 0.5$, the fastest convergence (within about 4 iterations) is achieved when power-sigmoid activation function (5) is exploited. In addition, the Newton-Raphson iteration [i.e., linear ZD (6)-(8) with $h = 1$] is also simulated, of which the superior performance is depicted via blue pentagrams in Figure 1.

For the same nonlinear equation (11) solving, it is also worth investigating the convergence properties of ZD (6) with different values of step-size h. The results are shown in Figure 2, where power-sigmoid activation function (5) is exploited with $\xi = 4$ and $p = 3$. In this case, $\theta = \phi'(0) = \xi(1+\exp(-\xi))/\big(2(1-\exp(-\xi))\big) = 2.0746$, and thus h should satisfy $0 < h < 2/\theta = 0.9640$. Choose the initial state $x_0 = 3.5$, and compare among different step-size values (e.g., $h = 0.05$, $h = 0.1$, $h = 0.6$ and $h = 1$). As we can see from Figure 2, the convergence speed could be expedited by appropriately increasing the step-size h. For example, in the case of $h = 0.6$, theoretical root $x_2^* = 3$ is reached by power-sigmoid ZD (6) with about 4 iterations; whereas, in the cases of $h = 0.1$ and $h = 0.05$, root $x_2^* = 3$ is reached with about 30 and 70 iterations, respectively. Furthermore, too large values of h may lead to cyclic oscillations or even divergence [1], e.g., when $h = 1 > 0.9640$ as illustrated in Figure 2.

Example 2. Consider the following nonlinear equation

$$f(x) = (x+2)(x-4)^5 = 0. \tag{12}$$

Different from (11), this nonlinear equation (12) theoretically has a simple root $x_1^* = -2$ and a multiple root $x_2^* = 4$ of order five (i.e., $q = 5$ here). By starting from initial state $x_0 = 4.5$, we examine the effectiveness of ZD (6) on solving for the multiple root $x_2^* = 4$. As shown in Figure 3, employing power-sigmoid activation function with $\xi = 4$ and $p = 3$ (in this case, h should satisfy $0 < h < 2q/\theta = 4.8201$) and setting step-size $h = 0.8$, we can find that the discrete-time ZD solver (6) converges to $x_2^* = 4$ in a fewer iterations as compared with traditional Newton iteration (8). This demonstrates the superior effectiveness of discrete-time ZD solver (6) in multiple roots finding, in addition to the important link to Newton iteration.

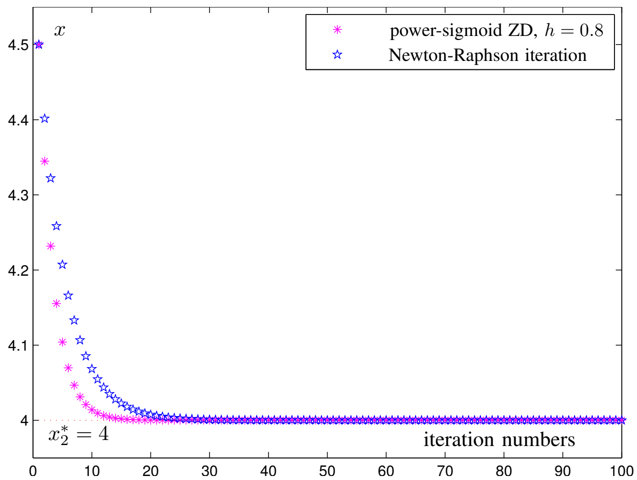

Fig. 3. Convergent performance of ZD (6) using power-sigmoid activation function with $h = 0.8$ and Newton iteration solving $(x + 2)(x - 4)^5 = 0$

V. Conclusions

A special class of neural dynamics (i.e., Zhang dynamics, ZD) has been found, developed and analyzed since 2001 for online solution of time-varying problems solving. Compared with conventional gradient-dynamics, the Zhang dynamics is designed base on the elimination of indefinite error-functions (instead of positive or lower-bounded energy-functions). In this paper, as an extension and generalization of our previous work, the discrete-time ZD solvers are generalized and investigated for solving the static (or termed, time-invariant) nonlinear equations. Moreover, we discover that the ZD models contains Newton iteration as a special case in some sense, which further presents another new reasonable explanation/interpretation about the origin of Newton iteration. Both theoretical analysis and simulation results demonstrate the efficacy and superiority of ZD-type solvers (continuous-time and/or discrete-time) on static nonlinear equations solving as well.

Acknowledgment

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 NUIM, University of Strathclyde, NUS, CUHK, SCUT and HUST which he had been with from 1992 to 2006. He had been supported by such research fellowships, assistantship and studentships, which had related to this research explicitly and implicitly. His web-page was also available at http://www.ee.sysu.edu.cn/teacher/detail.asp?sn=129. In addition, with the following Einstein's words, Yunong would like to thank and encourage the student-coauthors for their important contribution and ideas: "*Science is not just a collection of laws, a catalogue of unrelated facts. It is a creation of the human mind with its freely invented ideas and concepts ... ...*"

References

[1] J. H. Mathews and K. D. Fink, Nemerical Methods Using MATLAB, Beijing: Publishing House of Electronics Industry, 2005, pp. 41-90.

[2] C. Lin, Numerical Computation Methods, Beijing: Science Press, 2005, pp. 18-53.

[3] J. Feng, G. Che, and Y. Nie, Principles of Numerical Analysis, Beijing: Science Press, 2001, pp. 182-222.

[4] J. R. Sharma, "A composite third order Newton-Steffensen method for solving nonlinear equations," Applied Mathematics and Computation, vol. 169, pp. 242–246, 2005.

[5] C. Chun, "Construction of Newton-like iteration methods for solving nonlinear equations," Numerische Mathematik, vol. 104, pp. 297–315, 2006.

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

[7] Y. Zhang and J. Wang, "Global exponential stability of recurrent neural networks for synthesizing linear feedback control systems via pole assignment," IEEE Transactions on Neural Networks, vol. 13, no. 3, pp. 633-644, 2002.

[8] Y. Zhang and J. Wang, "A dual neural network for convex quadratic programming subject to linear equality and inequality constraints," Physics Letters A, vol. 298, pp. 271-278, 2002.

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

[10] Y. Zhang, "Towards piecewise-linear primal neural networks for optimization and redundant robotics," In Proceedings of IEEE International Conference on Networking, Sensing and Control, pp. 374-379, 2006.

[11] 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, 2002.

[12] 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, 2005.

[13] Y. Zhang and Z. Li, "Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints," Physics Letters A, vol. 373, no. 18-19, pp. 1639-1643, 2009.

[14] Y. Zhang, N. Tan, B. Cai, and Z. Chen, "MATLAB Simulink modeling of Zhang neural network solving for time-varying pseudoinverse in comparison with gradient neural network," In Proceedings of the 2nd International Symposium on Intelligent Information Technology Application, vol. 1, pp. 39-43, 2008.

[15] 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 International Joint Conference on Neural Networks, pp. 142-146, 2008.

[16] Y. Zhang, C. Yi, and W. Ma, "Comparison on gradient-based neural dynamics and Zhang neural dynamics for online solution of nonlinear equations," In LNCS Proceedings of International Symposium on Intelligence Computation and Applications, vol. 5370, pp. 269-279, 2008.

[17] Y. Zhang, C. Yi, and J. Zheng, "Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation," Neural Computing and Applications, submitted, July 2008.

[18] S. Abbasbandy, "Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method," Applied Mathematics and Computation, vol. 145, no. 2-3, pp. 887-893, 2003.

[19] M. Basto, V. Semiao, and F. L. Calheiros, "A new iterative method to compute nonlinear equations," Applied Mathematics and Computation, vol. 173, no. 1, pp. 468-483, 2006.

[20] A. Zou and Y. Zhang, Basis-Function Neural Networks and Their Application, Guangzhou: Sun Yat-Sen University Press (http://press.sysu.edu.cn/ShowBookDetail.asp?BookID=1121), 2009.

[21] 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, 2007.

[22] Y. Zhang, W. Ma, and C. Yi, "The link between Newton iteration for matrix inveration and Zhang neural network (ZNN)," In Proceedings of IEEE International Conference on Industrial Technology, pp. 1-6, 2008.

[23] 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, vol. 1, pp. 34-38, 2008.

[24] P. Kornerupa and J.-M. Muller, Choosing starting values for certain Newton-Raphson iterations, Theoretical Computer Science, vol. 351, no. 1, pp. 101-110, 2006.