Mathematics, Pusan National University

Introduction to Zhang Neural Network And Solving Time-varying Matrix Equations

Taehyeong Kim

August 31, 2022

Content

About Me

Name	Taehyeong Kim
Advisor	Prof. Hyun-Min Kim
Position	Ph.D student in Mathematics
Major	Numerical linear algebra, Mathematical computing, Nonlinear matrix equation, Iterative method
Program	MATLAB
	Python

Motivation

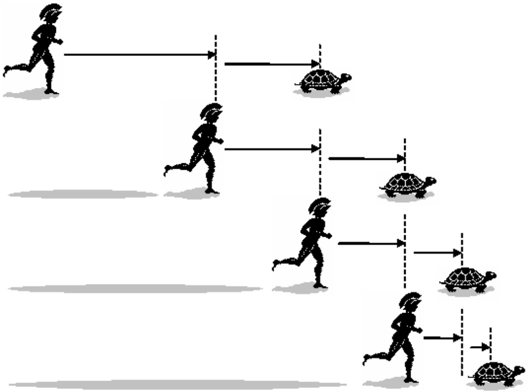

Figure 1: Zeno's paradoxes

In **mathematics**, Zeno's paradoxes is **false**.

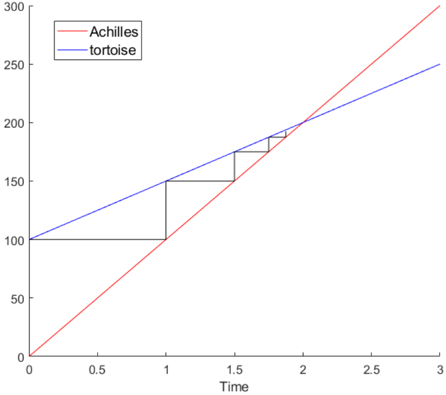

Figure 2: Obviously, Achilles can overtakes the tortoise!

But in **computer science**, Zeno’s paradox is **TRUE**!

Lagging error phenomenon

Consider the time-varying reciprocal problem in the following form:

$$f(x(t), t) = a(t)x(t) - 1 = 0 \in \mathbb{R}, t \in [0, -\infty) \quad (1)$$

where $a(t) \neq 0 \in \mathbb{R}$ denotes a smoothly time-varying scalar with $\dot{a}(t) \in \mathbb{R}$ denoting the time derivative of $a(t)$.

aim : Finding the $x(t) \in \mathbb{R}$ to make (1) hold true at any time $t \in [0, -\infty)$.
And denote $x^*(t)$ as the theoretical time-varying reciprocal of $a(t)$, i.e., mathematically, $x^*(t) = 1/a(t)$ in (1).

Lagging error phenomenon

Remark

This $x^*(t)$ is given symbolically for better understanding and solution comparison, whose the computation of $1/a(t)$ at every single time instant t is less practical in real-life applications. When we compute $1/a(t)$ at a time instant t, as the computation consumes time Δt inevitably, the value of $a(t)$ is changing during the computation procedure. This is the so-called **lagging error phenomenon**.

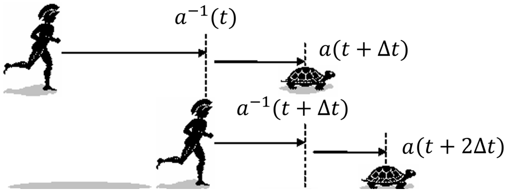

Figure 3: Achilles never can overtakes the tortoise... in computer!

Application

- Control theory : Real-time tracking
  - GPS

  - Robot arm

Zhang dynamics (ZD) has been formally proposed by Zhang et al. for various time-varying problems solving.

Concept of Zhang dynamics

Zhang dynamics(ZD) is a special type of neural dynamics that has been formally proposed by Zhang et al. for various time-varying problems solving.

According to Zhang et al.'s neural-dynamics design method, the ZD is designed based on an indefinite Zhang function (ZF) as the error-monitoring function.

To lay a basis for further discussion, the design procedure for a ZD model is presented as follows.

1. Define an indefinite ZF as the error-monitoring function to monitor the process of time-varying reciprocal finding.
2. To force $e(t)$ globally and exponentially converge to zero, we choose its time derivative $\dot{e}(t)$ via the following ZD design formula,

$$\dot{e}(t) = \frac{\mathrm{d}e(t)}{\mathrm{d}t} = -\gamma e(t), \tag{2}$$

   where design parameter $\gamma > 0 \in \mathbb{R}$.
3. By expanding the ZD design formula (2), the dynamic equation of a ZD model is thus established for time-varying reciprocal finding.

Theorem 1.1

As for the ZD design formula (2) which is also a dynamic system, starting from an initial error $e(0) \in \mathbb{R}$, the error function $e(t) \in \mathbb{R}$ globally and exponentially converges to zero with rate γ.

Proof.

For (2), by calculus, we obtain its analytical solution as $e(t) = e(0)exp(-\gamma t)$. Based on the definition of global and exponential convergence, we can draw the conclusion that, starting from any $e(0)$, $e(t)$ globally and exponentially converges to zero with rate γ, as time t tends to infinity. □

$$f(x(t),t) = a(t)x(t) - 1 = 0 \in \mathbb{R}, t \in [0, -\infty)$$

For real-time solution of time-varying reciprocal problem (1), we define the following four different ZFs:

$$e(t) = x(t) - \frac{1}{a(t)}, \tag{3}$$

$$e(t) = a(t) - \frac{1}{x(t)}, \tag{4}$$

$$e(t) = a(t)x(t) - 1, \tag{5}$$

$$e(t) = \frac{1}{a(t)x(t)} - 1. \tag{6}$$

$$\dot{e}(t) = \frac{\mathrm{d}e(t)}{\mathrm{d}t} = -\gamma e(t)$$

Example of ZD model

Let us consider the ZD design formula (2) and ZF (3). Then, we have

$$\dot{x}(t) + \frac{1}{a^2(t)}\dot{a}(t) = -\gamma\left(x(t) - \frac{1}{a(t)}\right),$$

which is rewritten as

$$a^2(t)\dot{x}(t) = -\dot{a}(t) - \gamma\left(a^2(t)x(t) - a(t)\right). \tag{7}$$

Thus, we obtain ZD model (7) for time-varying reciprocal finding.

For ZD model (7),

$$\dot{x}(t) = \left(1 - a^2(t)\right) \dot{x}(t) - \dot{a}(t) - \gamma \left(a^2(t)x(t) - a(t)\right).$$

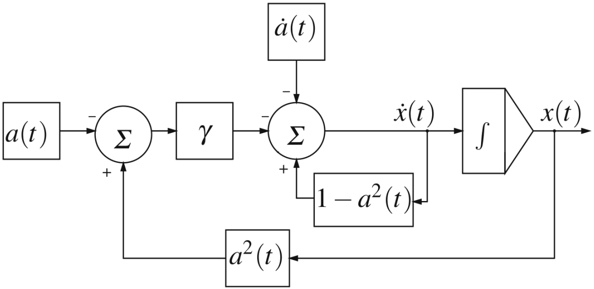

Figure 4: Block diagrams of ZD models (7) for time-varying reciprocal finding

Similarly, we obtain ZD models using ZFs equations (4)–(6), respectively.

ZF	ZD model
(3)	$a^2(t)\dot{x}(t) = -\dot{a}(t) - \gamma\left(a^2(t)x(t) - a(t)\right)$
(4)	$\dot{x}(t) = -\dot{a}(t)x^2(t) - \gamma\left(a(t)x^2(t) - x(t)\right)$
(5)	$a(t)\dot{x}(t) = -\dot{a}(t)x(t) - \gamma(a(t)x(t) - 1)$
(6)	$a(t)\dot{x}(t) = -\dot{a}(t)x(t) + \gamma\left(a(t)x(t) - a^2(t)x^2(t)\right)$

Table 1: Different ZFs resulting in different ZD models for time-varying reciprocal finding

Following proposition shows the convergence properties of the proposed ZD model (7) for time-varying reciprocal finding.

Proposition

Consider a smoothly time-varying scalar $a(t) \neq 0 \in \mathbb{R}$ involved in time-varying reciprocal problem (1). Starting from randomly-generated initial state $x(0) \neq 0 \in \mathbb{R}$ which has the same sign as $a(0)$, the neural state $x(t)$ of ZD model (7) derived from ZF (3) exponentially converges to the theoretical time-varying reciprocal $x^*(t)$ of $a(t)$ [i.e., $a^{-1}(t)$].

Time-varying quadratic matrix equation

Consider a time-varying quadratic matrix equation

$$\mathcal{F}(t) = A(t)(X(t))^2 + B(t)X(t) + C(t) = 0 \tag{8}$$

where $A(t), B(t), C(t) \in \mathbb{R}^{n\times n}$ are given and $X(t) \in \mathbb{R}^{n\times n}$ is unknown matrix.

We will compare three methods for solving (9). These are Fixed point iteration(FPI), Newton's method(NM), and ZNN.

In this experiments, we set $A(t), B(t), C(t)$ as following:

$$
A(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

$$
B(t) = \begin{bmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{bmatrix},
$$

$$
C(t) = \begin{bmatrix} \cos(t)^2 - 2\cos(t)\sin(t) - \sin(t)^2 & \sin(t)^2 - \cos(t)^2 - 2\cos(t)\sin(t) \\ 2\cos(t)\sin(t) + \cos(t)^2 - \sin(t)^2 & \cos(t)^2 - 2\cos(t)\sin(t) - \sin(t)^2 \end{bmatrix}.
$$

Then the solution matrix is $S(t) = \begin{bmatrix} \sin(t) & \cos(t) \\ -\cos(t) & \sin(t) \end{bmatrix}$.

We use the following error function for each method:

for fixed time t,

$$Error(t) = \|S(t_{cal}) - X(t)\|_F$$

where $t_{cal} = t$ + calculation time of each method.

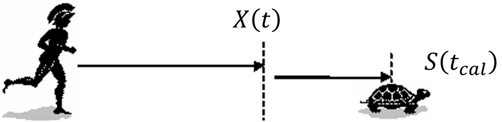

Figure 5: Time passes even while the algorithm is running.

For fixed time t, find $X(t)$ for fixed $A(t), B(t), C(t)$ using Newton's method.

Algorithm 1: Newton's method(NM)

Input: $A(t), B(t), C(t)$, tolerence: tol
Output: solution: X, calculation time: t_{cal}

$X \leftarrow \mathrm{zeros}(2, 2)$ `// Starting NM with zero initial matrix.`

tic `// Calculate start. Time is still running.`

while $res > tol$ **do**

$\quad \mathrm{vec}\, H = -(I \otimes (AX + B) + X^\top \otimes A)\, \mathrm{vec}(AX^2 + BX + C)$

$\quad X_{new} \leftarrow X + H$

$\quad res \leftarrow \|X_{new} - X\|_F$

$\quad X \leftarrow X_{new}$

end

$t_{cal} \leftarrow t + toc$ `// Calculation end.`

Numerical Experiments

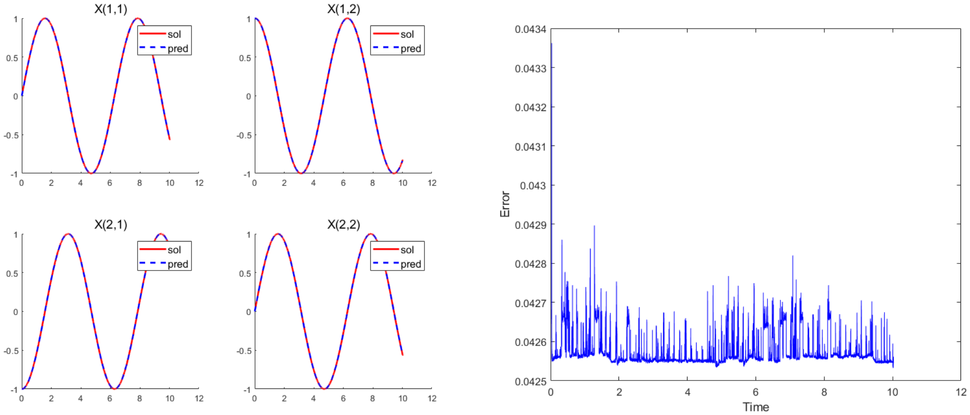

Figure 6: Result of Newton's method

For fixed time t, find $X(t)$ for fixed $A(t), B(t), C(t)$ using Fixed point iteration.

Algorithm 2: Fixed point iteration(FPI)

Input: $A(t), B(t), C(t)$, tolerence: tol
Output: solution: X, calculation time: t_{cal}

$X \leftarrow \text{zeros}(2,2)$ `// Starting FPI with zero initial matrix.`

tic `// Calculate start. Time is still running.`

while $res > tol$ **do**

$X_{new} \leftarrow (-B - AX)^{-1}C$

$res \leftarrow \|X_{new} - X\|_F$

$X \leftarrow X_{new}$

end

$t_{cal} \leftarrow t + toc$ `// Calculation end.`

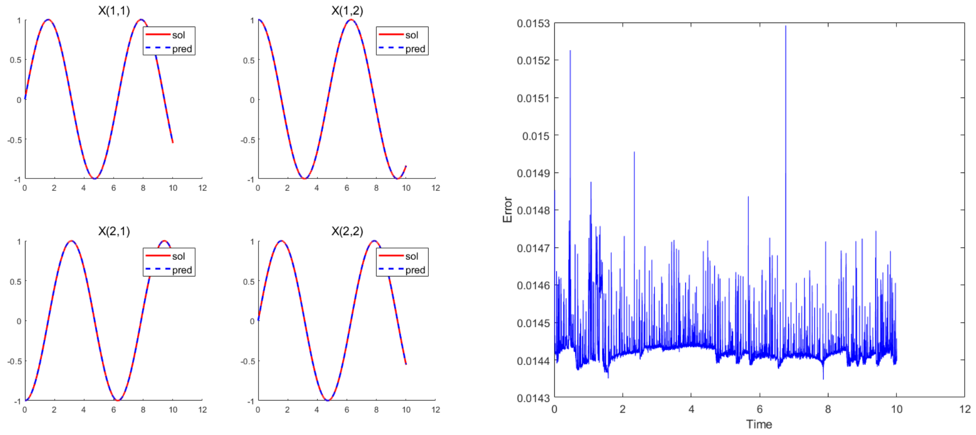

Figure 7: Result of Fixed point iteration

Let ZF as below:

$$E(t) = A(t)(X(t))^2 + B(t)X(t) + C(t) \tag{9}$$

And considering ZD design formula (2)

$$\begin{aligned}\dot{E}(t) &= \frac{dE(t)}{dt} \\ &= \dot{A}(t)(X(t))^2 + A\dot{X}(t)X(t) + AX(t)\dot{X}(t) + \dot{B}(t)X(t) + B(t)\dot{X}(t) + \dot{C}(t) \\ &= -\gamma(A(t)(X(t))^2 + B(t)X(t) + C(t))\end{aligned}$$

Then, we can obtain ZD model using ZF equation,

$$\begin{aligned}\dot{X}(t) =& (I - A(t)X(t) - B(t))\dot{X}(t) - A(t)\dot{X}(t)X(t) - \dot{A}(t)(X(t))^2 \\ &- \dot{B}(t)X(t) - \gamma(A(t)(X(t))^2 + B(t)X(t) + C(t))\end{aligned}$$

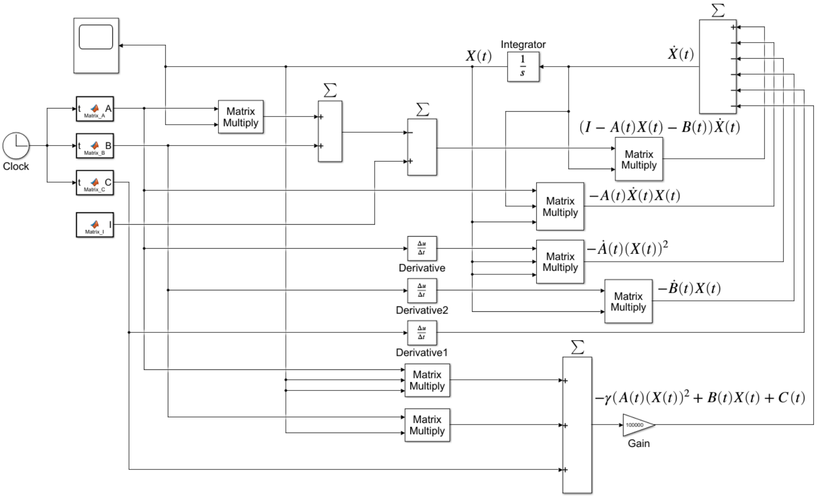

Figure 8: ZNN Simulink Model for Solving QME

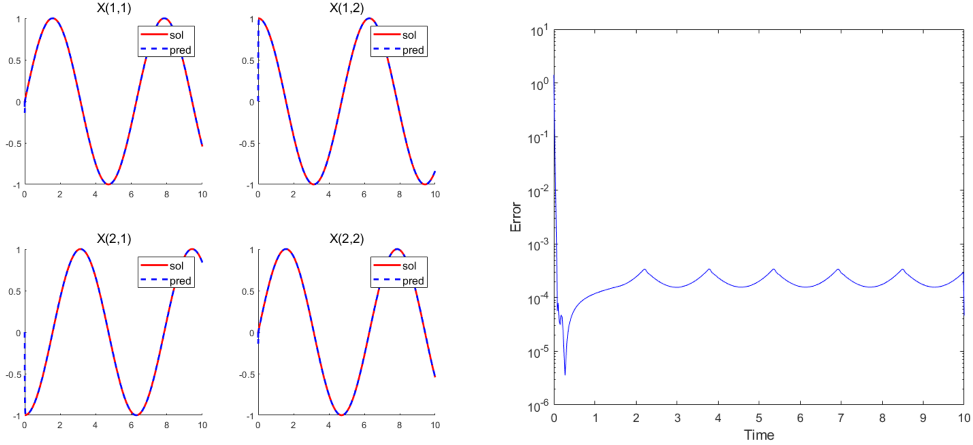

Figure 9: Result of Zhang Neural Network

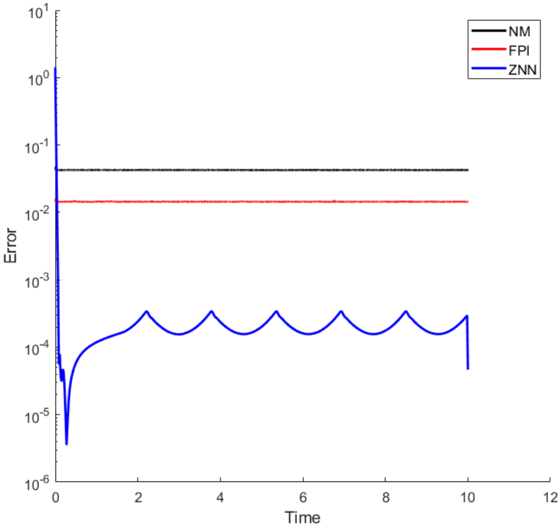

Figure 10: Error comparison for each method

Reference I

[1] PA Fuhrmann. “A functional approach to the Stein equation”. In: *Linear algebra and its applications* 432.12 (2010), pp. 3031–3071.

[2] Chuanqing Gu and Huiyan Xue. “A shift-splitting hierarchical identification method for solving Lyapunov matrix equations”. In: *Linear algebra and its applications* 430.5-6 (2009), pp. 1517–1530.

[3] Dongsheng Guo and Yunong Zhang. “Zhang neural network, Getz–Marsden dynamic system, and discrete-time algorithms for time-varying matrix inversion with application to robots’ kinematic control”. In: *Neurocomputing* 97 (2012), pp. 22–32.

[4] Dongsheng Guo et al. “Case study of Zhang matrix inverse for different ZFs leading to different nets”. In: *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2014, pp. 2764–2769.

[5] Yunong Zhang and Shuzhi Sam Ge. “Design and analysis of a general recurrent neural network model for time-varying matrix inversion”. In: *IEEE Transactions on Neural Networks* 16.6 (2005), pp. 1477–1490.

[6] Yunong Zhang, Danchi Jiang, and Jun Wang. "A recurrent neural network for solving Sylvester equation with time-varying coefficients". In: *IEEE Transactions on Neural Networks* 13.5 (2002), pp. 1053–1063.

[7] Yunong Zhang and Zhan Li. "Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints". In: *Physics Letters A* 373.18-19 (2009), pp. 1639–1643.

[8] Yunong Zhang, Weimu Ma, and Binghuang Cai. "From Zhang neural network to Newton iteration for matrix inversion". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 56.7 (2008), pp. 1405–1415.

[9] Yunong Zhang and Chenfu Yi. *Zhang neural networks and neural-dynamic method*. Nova Science Publishers, Inc., 2011.

[10] Yunong Zhang et al. "Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation". In: *Neural Computing and Applications* 20.1 (2011), pp. 1–7.

[11] Yunong Zhang et al. "Different Zhang functions leading to different Zhang-dynamics models illustrated via time-varying reciprocal solving". In: *Applied Mathematical Modelling* 36.9 (2012), pp. 4502–4511.

[12] Bin Zhou, James Lam, and Guang-Ren Duan. "On Smith-type iterative algorithms for the Stein matrix equation". In: *Applied Mathematics Letters* 22.7 (2009), pp. 1038–1044.

Thank you!