

Chapter 1

Time-Varying Reciprocal

Abstract Along with neural dynamics (based on analog solvers) widely arising in scientific computation and optimization fields in recent decades which attracts extensive interest and investigation of researchers, a special type of neural dynamics, called Zhang dynamics (ZD), has been formally proposed by Zhang et al. for real-time solution of time-varying problems. By following Zhang et al.'s neural-dynamics design method, the ZD model, which is based on an indefinite Zhang function (ZF), can guarantee the exponential convergence performance for time-varying problems solving. In this chapter, for time-varying reciprocal finding, we propose, generalize, develop, and investigate different indefinite ZFs as the error-monitoring functions, which can lead to different ZD models. In addition, for the goal of developing the floating-point processors or coprocessors for the future generation of computers, the MATLAB Simulink modeling and simulative verifications of such different ZD models are presented. The modeling results further substantiate the efficacy of the proposed ZD models for time-varying reciprocal finding.

1.1 Introduction and Preliminaries

The reciprocal computation, which is described in the form of $f(x) = ax - 1 = 0$, is considered to be an important operation in a floating-point divider/processor. Thus, many researches on the reciprocal computation are conducted and presented [1–6]. However, these researches are just for the static reciprocal computation, thereby making the corresponding methods less accurate enough to solve the time-varying reciprocal problem in the following form:

$$f(x(t), t) = a(t)x(t) - 1 = 0 \in \mathbb{R}, t \in [0, +\infty), \quad (1.1)$$

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 $a(t)$, both of which are assumed to be known numerically or could be measured accurately. In this chapter, we aim at finding the $x(t) \in \mathbb{R}$ to make (1.1) hold true at any time instant $t \in [0, +\infty)$. Furthermore, $x^*(t)$ is used to denote the theoretical time-varying reciprocal of $a(t)$ [i.e., mathematically, $x^*(t) = 1/a(t)$].

Remark 1.1 The above $x^*(t)$ 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. Specifically, 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. Thus, the computed result is less accurate and less effective, since the value of the theoretical time-varying reciprocal has actually changed to $x^*(t + \Delta t)$ after the computation. This is the so-called lagging-error phenomenon. Note that, as for other time-varying problems solving (via the conventional computation approaches), such types of lagging-error phenomena still exist. This propels us to develop and investigate an effective computation approach for real-time solution of various time-varying problems (e.g., the ones presented and investigated in this chapter as well as Chaps. 2–13).

Generally speaking, in the solving process of (1.1), a real-time solver first receives the specific data of $a(t)$ at one single time instant; then the solver does computations based on the present and/or the stored previous data; and finally, it outputs the result to the user. Note that, in this process, the solver cannot use the future data because they are unknown and have not come yet at the present time instant. Furthermore, at every single time instant, we, based on the present and/or previous data, compute the result for future. This is also because computation consumes time inevitably. As for the conventional approaches, the computation is based on the present data, and the computed result is directly used for future. Thus, there exists the lagging-error problem (see also Remark 1.1), when they are directly exploited to solve the time-varying reciprocal problem. In other words, these approaches are less effective on the time-varying reciprocal problem solving. This is the reason why we need to develop and investigate an effective approach for time-varying reciprocal finding (and further, for real-time solution of various time-varying problems).

Being different from the conventional neural-dynamics approach (i.e., gradient dynamics, GD), a special type of neural dynamics, called Zhang dynamics (ZD), has been formally proposed by Zhang et al. for various time-varying problems solving [7–13]. 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 (where the word “indefinite” here means that such an error-monitoring function can be positive, zero, negative or even lower-unbounded). This differs from the situation involved in the design of conventional approaches; for example, a norm-based positive-definite energy function is generally used in the GD design [8, 11, 12]. Thus, by making use of the time-derivative information of the time-varying coefficient(s) involved in the time-varying problem, the resultant ZD models can methodologically avoid the lagging errors generated by the conventional approaches. Note that such ZD models can guarantee much better convergence performance to the theoretical time-varying solution of the time-varying problem in an error-free manner. Besides, for better understanding and to lay a basis for further investigation, the concepts of ZD and ZF are presented as follows.

Concept 1.1 Zhang dynamics (ZD) has been generalized from Zhang neural network formally since 2008 [12], of which the state dimension can be multiple or one. It is viewed as a systematic approach to real-time solution of time-varying problems with scalar situation included as well. It differs from the conventional GD in terms of the problem to be solved, error function, design formula, dynamic equation, and the utilization of time-derivative information.

Concept 1.2 Zhang function (ZF), which is also referred to as Zhangian, is the design basis of ZD. It differs from the usual error/energy functions in the study of conventional approaches. Specifically, compared with the norm-based scalar-valued positive or at least lower-bounded energy function usually used in the GD design, ZF (1) is indefinite (i.e., can be positive, zero, or negative, in addition to being bounded, unbounded, or even lower unbounded), (2) can be matrix- or vector-valued (when solving a time-varying matrix- or vector-valued problem), and (3) can be real- or complex-valued (corresponding to a real- or complex-valued time-varying problem solving) to monitor and control the process of time-varying problems solving fully.

In this chapter, focusing on time-varying reciprocal finding, we propose, generalize, develop, and investigate different ZD models by defining different ZFs as the error-monitoring functions. In addition to the theoretical analyses and verifications of the convergence characteristics of the proposed ZD models, the MATLAB Simulink modeling [14–16] and illustrative examples are presented and investigated with the goal of developing the floating-point processors or coprocessors for the future generation of computers. From the modeling results, the efficacy of the proposed ZD models based on different ZFs for time-varying reciprocal finding is substantiated. To the best of the author’s knowledge, almost all reported computation approaches [1–6] are theoretically/intrinsically designed for static/time-invariant reciprocal finding. There is almost no other literature handling such a specific problem solving, i.e., real-time solution of time-varying reciprocal, at present stage.

1.2 ZFs and ZD Models

In this section, we introduce four different ZFs and propose the resultant ZD models for solving the time-varying reciprocal problem (1.1).

Because the ZF is the design basis for deriving a ZD model and for presentation convenience, we denote the ZF by $e(t)$ with $\dot{e}(t)$ being the time derivative of $e(t)$. Note that, in this chapter and also in Chaps. 2 and 3, $e(t)$ and $\dot{e}(t)$ are used as the notations of the scalar-valued ZF and its time derivative, respectively. Besides, to lay a basis for further discussion, the design procedure for a ZD model is presented as follows.

- First, we define an indefinite ZF as the error-monitoring function to monitor the process of time-varying reciprocal finding.
- Second, to force $e(t)$ globally and exponentially converge to zero, we choose its time derivative $\dot{e}(t)$ via the following ZD design formula: [7–13]:

$$\dot{e}(t) = \frac{de(t)}{dt} = -\gamma e(t), \quad (1.2)$$

where design parameter $\gamma > 0 \in \mathbb{R}$ corresponds to the reciprocal of a capacitance parameter, which should be set as large as the hardware would permit [10, 12, 13, 17], or selected appropriately for the simulative purpose.

- Finally, by expanding the ZD design formula (1.2), the dynamic equation of a ZD model is thus established for time-varying reciprocal finding.

For the excellent property of global and exponential convergence of the ZD design formula (1.2), we have the following theorem.

Theorem 1.1 *As for the ZD design formula (1.2) 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 (1.2), 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. The proof is thus complete. \square

Besides, it is worth pointing out here that the aforementioned design procedure for the scalar situation can also be generalized for deriving the ZD models to solve other time-varying problems with matrix or vector formulations (e.g., the ones presented and investigated in Chaps. 4–10).

Specifically, for real-time solution of time-varying reciprocal problem (1.1), in this chapter, we define the following four different ZFs:

$$e(t) = x(t) - \frac{1}{a(t)}, \quad (1.3)$$

$$e(t) = a(t) - \frac{1}{x(t)}, \quad (1.4)$$

$$e(t) = a(t)x(t) - 1, \quad (1.5)$$

$$e(t) = \frac{1}{a(t)x(t)} - 1. \quad (1.6)$$

According to the ZD design formula (1.2), different ZFs lead to different ZD models, which is detailed as below.

- Let us consider the ZD design formula (1.2) and ZF (1.3). 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

$$a^2(t)\dot{x}(t) = -\dot{a}(t) - \gamma \left(a^2(t)x(t) - a(t) \right). \quad (1.7)$$

Thus, we obtain ZD model (1.7) for time-varying reciprocal finding.

- Considering the ZD design formula (1.2) and ZF (1.4), we have

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

which is reformulated as

$$\dot{x}(t) = -\dot{a}(t)x^2(t) - \gamma \left(a(t)x^2(t) - x(t) \right). \quad (1.8)$$

Therefore, ZD model (1.8) for time-varying reciprocal finding is obtained.

- By combining the ZD design formula (1.2) and ZF (1.5), we have

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

and then

$$a(t)\dot{x}(t) = -\dot{a}(t)x(t) - \gamma \left(a(t)x(t) - 1 \right). \quad (1.9)$$

ZD model (1.9) for time-varying reciprocal finding is thus obtained.

- With the ZD design formula (1.2) and ZF (1.6) combined, we have

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

which is rewritten as

$$a(t)\dot{x}(t) = -\dot{a}(t)x(t) + \gamma \left(a(t)x(t) - a^2(t)x^2(t) \right). \quad (1.10)$$

Therefore, we come to ZD model (1.10) for time-varying reciprocal finding.

As a result, we have obtained four different types of ZD models [i.e. (1.7)–(1.10)] for time-varying reciprocal finding, which correspond to four different types of ZFs [i.e., (1.3)–(1.6)]. For readers' convenience and also for comparison, such four different ZD model based on four different ZFs for time-varying reciprocal finding are listed in Table 1.1.

Table 1.1 Different ZFs resulting in different ZD models for time-varying reciprocal finding

ZF	ZD model
(1.3)	$a^2(t)\dot{x}(t) = -\dot{a}(t) - \gamma (a^2(t)x(t) - a(t))$
(1.4)	$\dot{x}(t) = -\dot{a}(t)x^2(t) - \gamma (a(t)x^2(t) - x(t))$
(1.5)	$a(t)\dot{x}(t) = -\dot{a}(t)x(t) - \gamma (a(t)x(t) - 1)$
(1.6)	$a(t)\dot{x}(t) = -\dot{a}(t)x(t) + \gamma (a(t)x(t) - a^2(t)x^2(t))$

1.3 Theoretical Results and Analyses

In this section, four propositions (viewed as the special cases of 1.1) are presented, which show the convergence properties of the proposed ZD models (1.7)–(1.10) for time-varying reciprocal finding.

Proposition 1.1 *Consider a smoothly time-varying scalar $a(t) \neq 0 \in \mathbb{R}$ involved in time-varying reciprocal problem (1.1). 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 (1.7) derived from ZF (1.3) exponentially converges to the theoretical time-varying reciprocal $x^*(t)$ of $a(t)$ [i.e., $a^{-1}(t)$].*

Proof We use the well-known Lyapunov method to prove the exponential convergence of ZD model (1.7).

First, starting with ZF (1.3), we define a Lyapunov candidate

$$V(x(t), t) = \frac{1}{2} \left(x(t) - \frac{1}{a(t)} \right)^2 \geq 0,$$

where $V(x(t), t) = 0$ for any $x(t) = a^{-1}(t)$, and $V(x(t), t) > 0$ for any $x(t) \neq a^{-1}(t)$. Then, we derive its time derivative as

$$\begin{aligned} \dot{V}(x(t), t) &= \frac{dV(x(t), t)}{dt} = \left(x(t) - \frac{1}{a(t)} \right) \left(\dot{x}(t) + \frac{1}{a^2(t)} \dot{a}(t) \right) \\ &= -\gamma \left(x(t) - \frac{1}{a(t)} \right)^2 = -2\gamma V(x(t), t). \end{aligned}$$

Since $V(x(t), t) \geq 0$, then $\dot{V}(x(t), t) = -2\gamma V(x(t), t) \leq 0$, which guarantees the (final) negative-definiteness of $\dot{V}(x(t), t)$.

Furthermore, from $\dot{V}(x(t), t) = -2\gamma V(x(t), t)$, we have

$$V(x(t), t) = V(x(0), 0) \exp(-2\gamma t).$$

That is,

$$\frac{1}{2} \left(x(t) - \frac{1}{a(t)} \right)^2 = \frac{1}{2} \left(x(0) - \frac{1}{a(0)} \right)^2 \exp(-2\gamma t).$$

Thus, we have

$$\left| x(t) - \frac{1}{a(t)} \right| = \left| x(0) - \frac{1}{a(0)} \right| \exp(-\gamma t),$$

where symbol $|\cdot|$ denotes the absolute value of a scalar. With $\alpha = |x(0) - 1/a(0)|$, the above equation is further rewritten as

$$\left| x(t) - \frac{1}{a(t)} \right| = \alpha \exp(-\gamma t),$$

which means that $x(t)$ exponentially converges to $a^{-1}(t)$ with the convergence rate $\gamma > 0$. That is, 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 (1.7) exponentially converges to the theoretical time-varying reciprocal $x^*(t) = a^{-1}(t)$ of $a(t)$ involved in time-varying Eq. (1.1). The proof is thus complete. \square

As for ZD models (1.8)–(1.10), we also have the following convergence results, with the related proofs being generalized from the proof of Proposition 1.1 (and being left to interested readers to complete as a topic of exercise).

Proposition 1.2 *Consider a smoothly time-varying scalar $a(t) \neq 0 \in \mathbb{R}$ involved in time-varying reciprocal problem (1.1). 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 (1.8) derived from ZF (1.4) converges to the theoretical time-varying reciprocal $x^*(t)$ of $a(t)$ [i.e., $a^{-1}(t)$], with the error defined in ZF (1.4) exponentially convergent to zero.*

Proposition 1.3 *Consider a smoothly time-varying scalar $a(t) \neq 0 \in \mathbb{R}$ involved in time-varying reciprocal problem (1.1). 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 (1.9) derived from ZF (1.5) converges to the theoretical time-varying reciprocal $x^*(t)$ of $a(t)$ [i.e., $a^{-1}(t)$], with the error defined in ZF (1.5) exponentially convergent to zero.*

Proposition 1.4 *Consider a smoothly time-varying scalar $a(t) \neq 0 \in \mathbb{R}$ involved in time-varying reciprocal problem (1.1). 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 (1.10) derived from ZF (1.6) converges to the theoretical time-varying reciprocal $x^*(t)$ of $a(t)$ [i.e., $a^{-1}(t)$], with the error defined in ZF (1.6) exponentially convergent to zero.*

1.4 Simulink Modeling

For possible hardware implementation based on digital circuits and also for the goal of developing the floating-point processors or coprocessors for the future generation of computers, the MATLAB Simulink modeling of the proposed ZD models (1.7)–(1.10) is investigated and presented in this section. Before doing this, we need to transform some of such ZD models into the following explicit forms, with the corresponding block diagrams depicted in Fig. 1.1.

- For ZD model (1.7),

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

- For ZD model (1.8), it is already in the explicit form, and does not need to be transformed.
- For ZD model (1.9),

$$\dot{x}(t) = (1 - a(t)) \dot{x}(t) - \dot{a}(t)x(t) - \gamma (a(t)x(t) - 1).$$

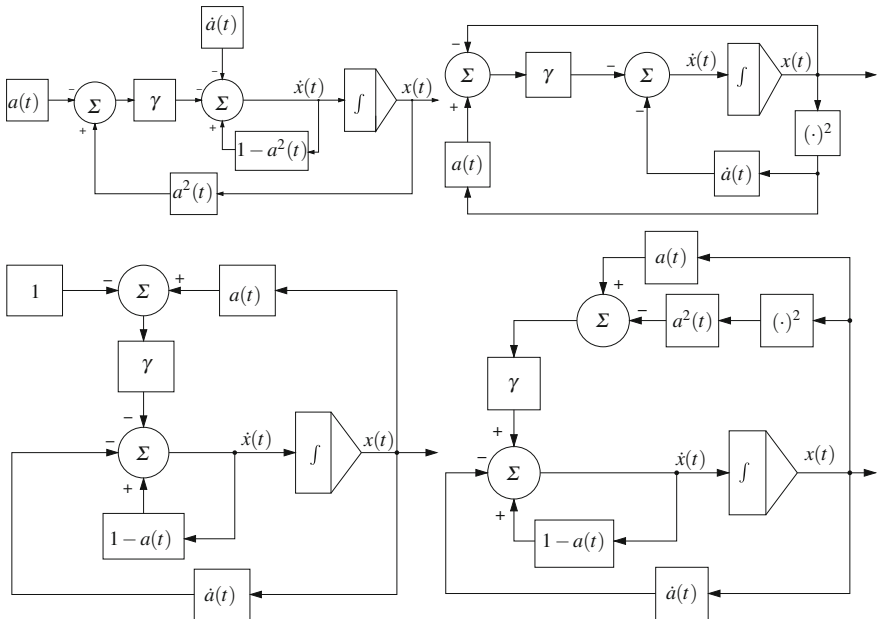


Fig. 1.1 Block diagrams of ZD models (1.7)–(1.10) for time-varying reciprocal finding

- For ZD model (1.10),

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

Therefore, the overall Simulink models of the proposed ZD models are shown in Figs. 1.2 and 1.3, in which $a(t)$ is generated by employing the ‘‘MATLAB Function’’ block using the ‘‘Clock’’ block as its input.

1.5 Illustrative Examples

In the previous sections, we have proposed the ZD models based on different ZFs for time-varying reciprocal finding, together with corresponding propositions and theoretical analyses. Based on the aforementioned Simulink models depicted in Figs. 1.2 and 1.3, the ensuing illustrative examples are shown to substantiate the efficacy of the proposed ZD models (1.7)–(1.10).

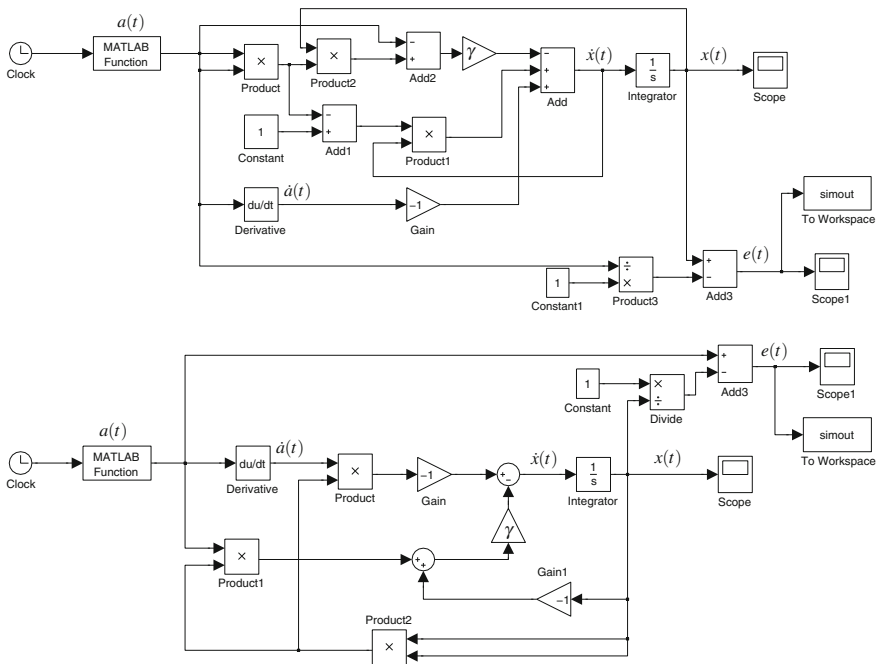


Fig. 1.2 Simulink modeling of ZD models (1.7) and (1.8) for time-varying reciprocal finding

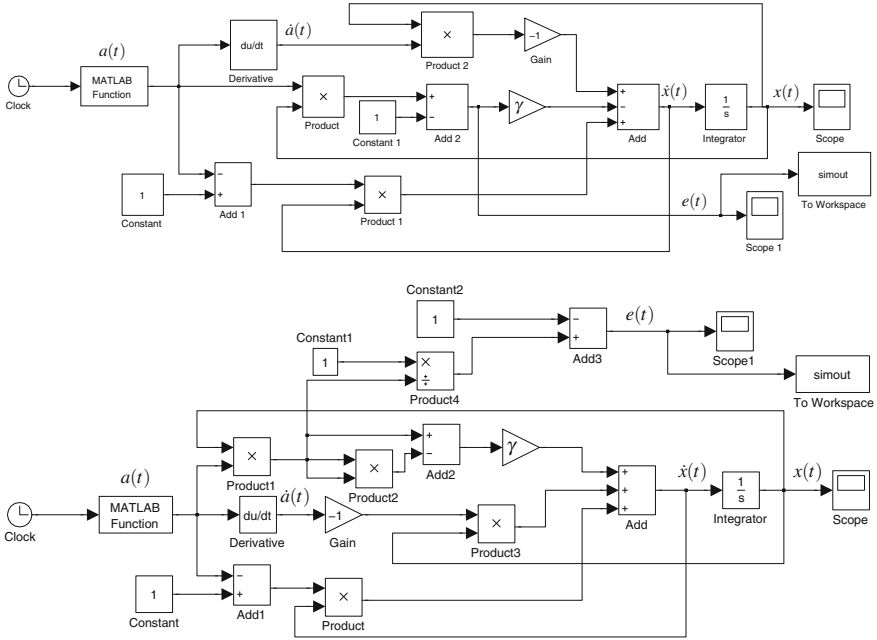


Fig. 1.3 Simulink modeling of ZD models (1.9) and (1.10) for time-varying reciprocal finding

Example 1.1 Let us consider the following time-varying reciprocal problem [in which $a(t) = \sin(3t) + \cos(\sin(2t)) + 2$ is involved]:

$$f(x(t), t) = (\sin(3t) + \cos(\sin(2t)) + 2)x(t) - 1 = 0. \quad (1.11)$$

The proposed ZD models (1.7)–(1.10) are exploited to solve this problem (1.11). It is easy to see that the theoretical initial reciprocal value is $x^*(0) = 1/a(0) \approx 0.333$. For convenience of observation, the initial state (or to say, starting value) $x(0)$ is randomly generated within $[0.2, 0.4]$, i.e., $x(0) \in [0.2, 0.4]$. The simulation results based on the Simulink models are presented in Fig. 1.4. As shown in the figure, with design parameter $\gamma = 10$, the states $x(t)$ of the proposed ZD models all converge to the theoretical time-varying reciprocal $x^*(t) = a^{-1}(t)$, i.e., the theoretical time-varying solution of (1.11), in a rather short time (i.e., in less than 1 s). Through this example, we have primarily shown the efficacy of the proposed ZD models (1.7)–(1.10) for solving the time-varying reciprocal problem.

Example 1.2 In this example, we are considering a more complicated situation of the time-varying reciprocal problem, i.e.,

$$f(x(t), t) = (\sin(\cos(4t)) + \exp(-\sin(3t)) + \cos(t) + 2)x(t) - 1 = 0, \quad (1.12)$$

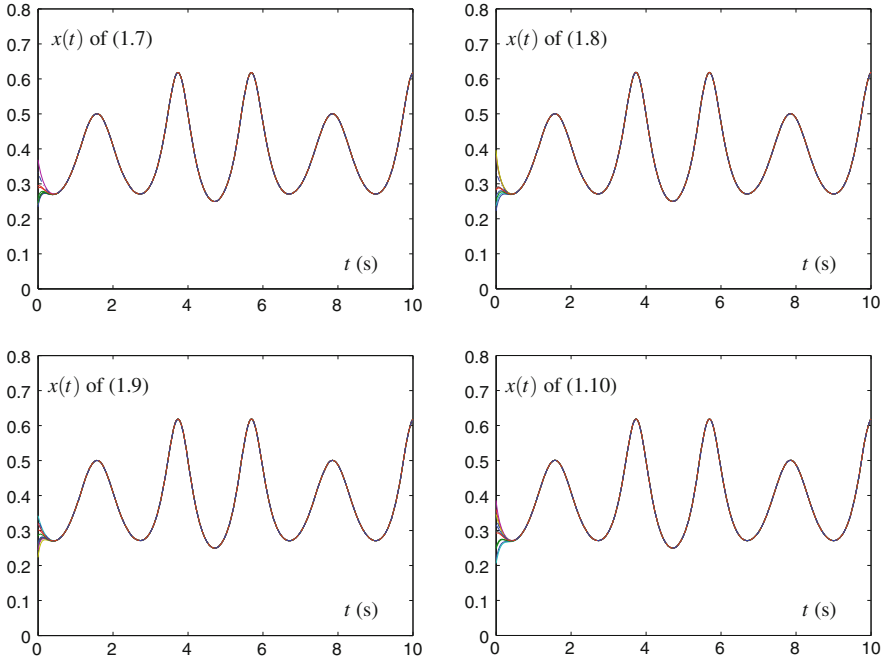


Fig. 1.4 State trajectories of ZD models (1.7)–(1.10) with design parameter $\gamma = 10$ for solving the time-varying reciprocal problem (1.11), where *dash-dotted curves* correspond to the theoretical time-varying reciprocal $a^{-1}(t)$, i.e., the theoretical solution of (1.11)

so as to investigate the general applicability of the proposed ZD models. In other words, $a(t) = \sin(\cos(4t)) + \exp(-\sin(3t)) + \cos(t) + 2$.

Similar to the way of Example 1.1, the proposed ZD models (1.7)–(1.10) are exploited to solve this reciprocal problem depicted in (1.12), and the corresponding simulation results are illustrated in Fig. 1.5. As shown in Fig. 1.5, with design parameter $\gamma = 10$ and the initial state $x(0)$ randomly generated within $[0.1, 0.4]$, the state trajectories of the proposed ZD models (1.7)–(1.10) all fit well with the theoretical time-varying reciprocal $x^*(t)$, i.e., the theoretical time-varying solution of (1.12), rapidly (i.e., in less than 1 s).

From the above two examples, we can draw the conclusion that, either for a simple problem (1.11) or a more complicated problem (1.12), the proposed ZD models (1.7)–(1.10) based on different ZFs (1.3)–(1.6) can all solve the time-varying reciprocal problem efficiently. That is, with appropriate values of design parameter γ and initial state $x(0)$, the neural states of the proposed ZD models all converge to the time-varying theoretical solution of the time-varying reciprocal problem rapidly and accurately. Thus, the efficacy of the proposed ZD models (1.7)–(1.10) for time-varying reciprocal finding is substantiated well.

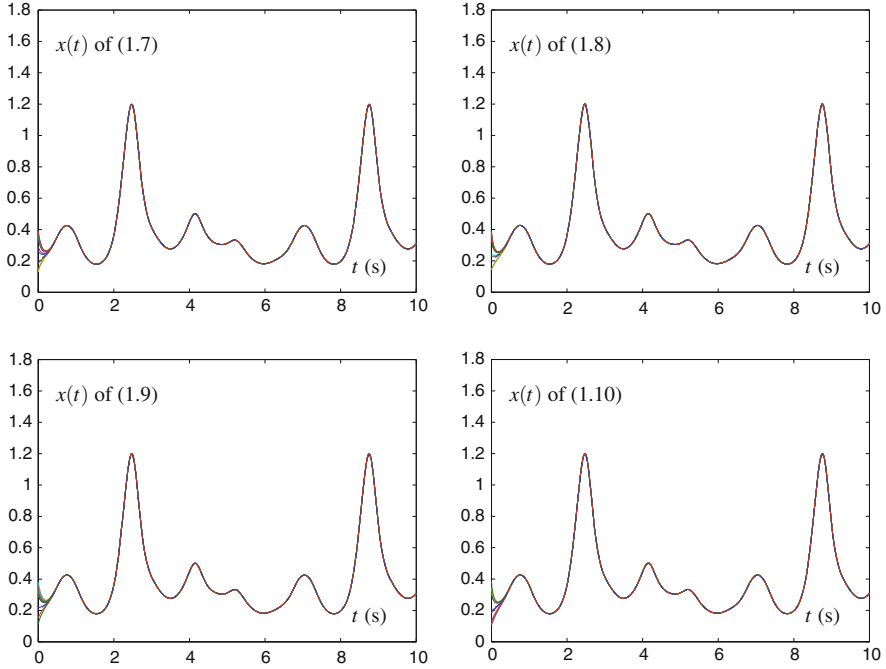


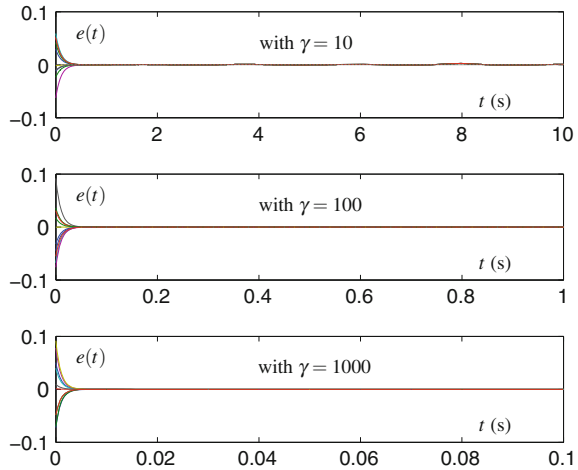
Fig. 1.5 State trajectories of ZD models (1.7)–(1.10) with design parameter $\gamma = 10$ for solving the time-varying reciprocal problem (1.12), where *dash-dotted curves* correspond to the theoretical time-varying reciprocal $a^{-1}(t)$, i.e., the theoretical solution of (1.12)

Example 1.3 As mentioned previously, the value of design parameter γ may affect the convergence performance of the proposed ZD models (1.7)–(1.10). Let us consider the following time-varying reciprocal problem [in which $a(t) = \sin(t) \cos(2t) + 2$]:

$$f(x(t), t) = (\sin(t) \cos(2t) + 2) x(t) - 1 = 0. \quad (1.13)$$

In this example, ZD model (1.7) is exploited to solve (1.13) with the initial state $x(0)$ randomly generated within $[0.4, 0.6]$. The computational errors $e(t) = x(t) - 1/(\sin(t) \cos(2t) + 2)$ with respect to different values of γ are displayed in Fig. 1.6. As seen from the figure, the convergence time of the computational error $e(t)$ to zero is becoming much shorter (i.e., from about 0.6 s to about 0.006 s) when the γ value increases from 10 to 1000. This simulation result indicates that design parameter γ plays an important role in the proposed ZD model (1.7), and should be set as large as the hardware would permit, or selected appropriately large for simulative purposes. Furthermore, the exponential-convergence characteristics of ZD model (1.7) can also be seen comparatively from Fig. 1.6. Note that the same conclusion applies to other

Fig. 1.6 Computational errors $e(t) = x(t) - 1/(\sin(t) \cos(2t) + 2)$ of ZD model (1.7) for solving the time-varying reciprocal problem (1.13) with different values of γ



ZD models [i.e., (1.8)–(1.10)], of which the related modeling results are omitted due to results similarity. Besides, the modeling testings of ZD models (1.8)–(1.10) are left to interested readers to complete as a topic of exercise.

1.6 Summary

In this chapter, by following Zhang et al.’s neural-dynamics design method and based on the Zhang function (ZF) as the error-monitoring function, a special type of neural dynamics, called Zhang dynamics (ZD), has been presented and investigated for real-time solution of time-varying reciprocal problem, which is in the time-varying form of $f(x(t), t) = a(t)x(t) - 1 = 0$ [i.e., (1.1)]. Specifically, based on different ZFs (1.3)–(1.6), different ZD models (1.7)–(1.10) have thus been proposed, generalized, developed, and investigated for time-varying reciprocal finding. Moreover, theoretical analyses have been given to substantiate the exponential convergence of the proposed ZD models. For possible hardware implementations based on digital circuits and for the goal of developing the floating-point processors or coprocessors for the future generation of computers, the MATLAB Simulink modeling of such proposed ZD models has been presented and investigated in this chapter. Through three illustrative examples, the efficacy of the proposed ZD models (1.7)–(1.10) has been further substantiated.

References

1. Pineiro J, Bruguera JD (2002) High-speed double-precision computation of reciprocal, division, squareroot, and inverse square root. *IEEE Trans Comput* 51(12):1377–1388
2. Hanrot G, Rivat J, Tenenbaum G, Zimmermann P (2003) Density results on floating-point invertible numbers. *Theor Comput Sci* 291(2):135–141
3. Kucukkabak U, Akkas A (2004) Design and implementation of reciprocal unit using table look-up and Newton-Raphson iteration. In: *Proceedings of euromicro symposium on digital system design*, pp 249–253
4. Croot E, Li R-C, Zhu HJ (2004) The *abc* conjecture and correctly rounded reciprocal square roots. *Theor Comput Sci* 315(2–3):405–417
5. Antelo E, Lang T, Montuschi P, Sannarelli A (2005) Low latency digit-recurrence reciprocal and square-root reciprocal algorithm and architecture. In: *Proceedings of the 17th IEEE symposium on computer arithmetic*, pp 147–154
6. Agrawal G, Khandelwal A, Swartzlander EE Jr (2007) An improved reciprocal approximation algorithm for a Newton Raphson divider. In: *Proceedings of advanced signal processing algorithms, architectures, and implementations XVII*, pp 1–12
7. Zhang Y, Jiang D, Wang J (2002) A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Trans Neural Netw* 13(5):1053–1063
8. Zhang Y, Yi C, Guo D, Zheng J (2011) Comparison on Zhang neural dynamics and gradient-based neural dynamics for online solution of nonlinear time-varying equation. *Neural Comput Appl* 20(1):1–7
9. Zhang Y, Li Z (2009) Zhang neural network for online solution of time-varying convex quadratic program subject to time-varying linear-equality constraints. *Phys Lett A* 373(18–19):1639–1643
10. Zhang Y, Ma W, Cai B (2009) From Zhang neural network to Newton iteration for matrix inversion. *IEEE Trans Circuits Syst I-Regul Pap* 56(7):1405–1415
11. Zhang Y, Ge SS (2005) Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans Neural Netw* 16(6):1477–1490
12. Zhang Y, Yi C (2011) *Zhang neural networks and neural-dynamic method*. Nova Science Publishers, New York
13. Zhang Y, Li F, Yang Y, Li Z (2012) Different Zhang functions leading to different Zhang-dynamics models illustrated via time-varying reciprocal solving. *Appl Math Model* 36(9):4502–4511
14. Zhang Y, Guo X, Ma W (2008) Modeling and simulation of Zhang neural network for online time-varying equations solving based on MATLAB Simulink. In: *Proceedings of the 7th international conference on machine learning and cybernetics*, pp 805–810
15. Tan N, Chen K, Shi Y, Zhang Y (2009) Modeling, verification and comparison of Zhang neural net and gradient neural net for online solution of time-varying linear matrix equation. In: *Proceedings of the 4th IEEE conference on industrial electronics and applications*, pp 3698–3703
16. Ansari MS, Rahman SA (2011) DVCC-based non-linear feedback neural circuit for solving system of linear equations. *Circuits, Syst Signal Process* 30:1029–1045
17. Mead C (1989) *Analog VLSI and neural systems*. Addison-Wesley Longman, Boston