STABILITY ANALYSIS OF RECURRENT NEURAL NETWORKS USING DISSIPATIVITY

By

NAM HOAI NGUYEN

Bachelor of Science in Automatic Control
Hanoi University of Science and Technology
Hanoi, Vietnam
2002

Master of Science in Measurement and Control Systems
Hanoi University of Science and Technology
Hanoi, Vietnam
2005

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2012

STABILITY ANALYSIS OF RECURRENT NEURAL NETWORKS

USING DISSIPATIVITY

Dissertation Approved:

Dr. Martin Hagan

Dissertation Advisor

Dr. Carl Latino

Dr. George Scheets

Dr. Lisa Mantini

Dr. Sheryl Tucker

Dean of the Graduate College

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

The objective of this research is to use dissipativity theory to analyze stability for a general class of discrete-time recurrent neural networks (RNNs). Then, a new training algorithm is proposed to train RNNs for stability.

Other efforts on the stability analysis of RNNs have been made in recent years. Jin and Gupta [1] found absolute stability conditions for RNNs based on Ostrowski's theorem. The networks they dealt with contained only a single layer without biases. Tanaka [2] analyzed the stability of neural network systems by using stability conditions, based on Lyapunov theory, of linear differential inclusions. The neural network systems investigated by Tanaka include a system that combines a neural network plant model and a neural network controller. However, there are no biases included in the neural networks. Suykens [3] found stability criteria for a class of RNNs that can be represented in a form he designated as NL_q. These criteria provide sufficient conditions for asymptotic stability. However, he did not deal with the case of nonzero biases.

Recently, Barabanov and Prokhorov proposed an approach for the stability analysis of RNNs with sector-type nonlinearities and nonzero biases based on the theory of absolute stability [4]. They later developed a new method based on reduction of dissipativity domain [5]. This method works effectively if the system has a convex Lyapunov function. Later, Liu proposed a generic network model, which is referred to as the discrete-time standard neural network model (DSNNM) [6]. The DSNNM represents a neural network model as the interconnection of a linear dynamic system and static nonlinear operators. Liu found some criteria for the globally asymptotic stability of equilibrium points of the DSNNM.

More recently, the method in [5] has been modified to speed up convergence by Jafari and Hagan [7]. Most recently, Kim and Braatz [8] used a modified Lure'-Postnikov function to obtain stability criteria for some classes of standard nonlinear operator forms. These methods can be applied to RNNs with nonzero biases. However, they have not yet been applied to the stability analysis of layered digital dynamic networks (LDDNs) [9].

A few authors in the past have applied dissipativity theory to continuous-time neural networks [10].We now want to apply this theory to discrete-time networks. In this work, a general class of discrete-time recurrent neural networks (LDDNs) will be considered, and dissipativity theory will be used to analyze stability of equilibrium points for LDDNs.

Dissipativity theory was first developed by Jan C. Willems [11, 12] in the 1970s. The other major authors of this theory are David Hill and Peter Moylan [13, 14]. The term "dissipativity" was inspired by the concept of passivity. A system for which the rate of increase in its stored energy is not greater than the absorbed input power is a passive system. In dissipativity theory, the stored energy is generalized by a storage function and the input power is generalized by a supply rate function.

One of the important results in passivity theory states that if two passive systems are connected in a feedback loop, then the resulting closed loop system is stable. A corresponding result from dissipativity theory states that if two dissipative systems are connected in a feedback loop, then the closed loop system is stable under certain conditions. This result will be used extensively throughout our work.

Dissipativity was first defined for continuous-time dynamic systems. Later, a discrete-time version of dissipativity was developed by W. Haddad [15]. However, to our knowledge, dissipativity theory has not yet been applied to discrete-time RNNs.

In the past, Liang Jin and Madan Gupta developed two stable dynamic backpropagation learning algorithms for a class of RNNs [16]. They used both local and global stability conditions to maintain the network stability during the training. Recently, Jason Horn, Orlando de Jesus and Martin Hagan [17] demonstrated that there are spurious valleys in the error surface of RNNs. These spurious valleys occur in unstable regions of the error surfaces and can cause difficulties in training. They suggested that one might be able to use a constrained optimization process to avoid the unstable region during training, but the constraints would be extremely complex for LDDNs.

In this work, several stability criteria based on dissipativity will be proposed. Then these novel criteria will be compared with those of Liu [6] and Barabanov and Prokhorov [18] on several test problems. Based on these criteria, we will propose a new training algorithm to train recurrent neural networks for stability. The new training algorithm will be tested on two examples of model reference control systems: a linear plant and a nonlinear plant.

This proposal includes seven chapters. The next chapter describes dissipativity theory for continuous-time systems. Chapter 3 is about the stability analysis of discrete-time recurrent neural networks using dissipativity. It presents some fundamental concepts and theorems, and gives a brief introduction to LDDNs. Next, a method is proposed to transform LDDNs into a standard interconnected system form. Then, sector conditions are introduced and some important lemmas and theorems are proposed. Finally, novel stability criteria for the equilibrium points of LDDNs, based on these lemmas and theorems, are found. In Chapter 4, existing stability criteria are reviewed and then compared with the novel criteria on a large number of test problems. The following chapter develops a framework for training recurrent neural networks for stability. A modified performance index is defined and a brief review of the first derivative of eigenvalue with respect to a matrix parameter is provided. Then, the first derivative of the maximum eigenvalue with respect to network weights is represented. In Chapter 6, we introduce neural network-based model reference control systems. The proposed training algorithm is applied to train neural network controllers for both a linear plant and a magnetic levitation system. The final chapter provides a summary and proposes future work.

CHAPTER 2

DISSIPATIVITY AND STABILITY FOR CONTINUOUS-TIME SYSTEMS

The theory of dissipativity was first developed by Willems [11], [12] for continuous-time dynamical systems. Recently, it has been extended to discrete-time dynamical systems [15], switched systems [19], and hybrid systems [20]. It has been applied to not only stability analysis [13], [14], [21], [22] but also controller synthesis [15], [20], [23], [24]. This chapter reviews dissipativity theory for continuous-time systems.

There are two settings in which dissipative dynamical systems have been defined: the input-output setting and the input-state-output setting. This chapter will concentrate on the second setting. The chapter will begin by introducing continuous-time dynamical systems. Next, continuous-time dissipative dynamical systems are defined, followed by an example. Finally, stability analysis of interconnected continuous-time dynamical systems is introduced.

2.1 Continuous-time dynamical systems

This section presents some definitions [11] concerning dynamical systems and dissipative systems. It also discusses the main properties of dissipative dynamical systems.

We begin by introducing some notation.

- R is the set of real numbers.
- $\mathbf{R}^n$ is the *n* dimensional Euclidean space.
- R_+ is the set of nonnegative real numbers.
- $\mathbf{R}_+^2 = \left\{(t_2, t_1) \in \mathbf{R}^2 | t_2 > t_1\right\}$.

- Shift operater $\sigma_T(.)$: Given a function $s(t)$, $t \in R$, then the shift operator is defined as $\sigma_T(s) = s(t+T)$.

Continuous-time dynamical systems were defined by Willems [11] as follows.

Definition 2.1 *A dynamical system Σ is defined as follows:*

- *X, U and Y are called the state space, the set of input values, and the set of output values, respectively.*
- *U^* is called the input space and it contains a class of U-valued functions on R.*
- *Y^* is called the output space and it contains a class of Y-valued functions on R.*
- *Assume that U^* and Y^* are closed under the shift operator.*
- *$\Phi : \boldsymbol{R}_+^2 \times X \times U^* \to X$ is called the state transtion function.*
  The following axioms hold:
  1. *Consistency: $\Phi(t_0, t_0, x_0, u) = x_0$ for all $t_0 \in R$, $x_0 \in X$, and $u \in U^*$.*
  2. *Determinism: $\Phi(t_1, t_0, x_0, u_1) = \Phi(t_1, t_0, x_0, u_2)$ for all $(t_1, t_0) \in \boldsymbol{R}_+^2$, $x_0 \in X$, and $u_1, u_2 \in U^*$ satisfying $u_1(t) = u_2(t)$ for $t_0 \leq t \leq t_1$.*
  3. *Semi-group property: $\Phi(t_2, t_0, x_0, u) = \Phi(t_2, t_1, \Phi(t_1, t_0, x_0, u), u)$ for $t_0 \leq t_1 \leq t_2$, $x_0 \in X$ and $u \in U^*$.*
  4. *Stationary: $\Phi(t_1 + T, t_0 + T, x_0, \sigma_T(u)) = \Phi(t_1, t_0, x_0, u)$ for all $(t_1, t_0) \in \boldsymbol{R}_+^2$, $T \in R$, $x_0 \in X$, and $u,\ \sigma_T(u) \in U$.*
- *$w : X \times U \to Y$ is called the read-out function.*
- *The Y-valued function $y(t) = w(\Phi(t, t_0, x_0, u), u(t))$ for $t \geq t_0$.*

It is assumed that a dynamical system Σ is given together with a real valued function $r(u, y) = r(u(t), y(t))$ called the supply rate function. The constraint of the supply rate function is that for any $(t_1, t_0) \in \mathbf{R}_+^2$, $u \in U$ and $y \in Y$ $r(u, y)$ satisfies $\int_{t_0}^{t_1} |r(u, y)|dt < \infty$.

2.2 Continuous-time dissipative dynamical systems

Continuous-time dissipative dynamical systems were defined by Willems [11] as follows.

Definition 2.2 *A dynamical system Σ with the supply rate function $r(u, y)$ is said to be dissipative if there exists a nonnegative function $S : X \to R_+$, called the storage function, such that for all $(t_1, t_0) \in \boldsymbol{R}_+^2$, $x_0 \in X$ and $u \in U$,*

$$S(x_1) \leq S(x_0) + \int_{t_0}^{t_1} r(u, y)dt \tag{2.1}$$

where $x_1 = \Phi(t_1, t_0, x_0, u)$ and $r(u, y) = r(u(t), y(t))$ with $y(t) = w(x(t), u)$.

The inequality (2.1) is called the dissipation inequality.

2.3 Properties of supply rate function

There are some key properties of supply rate functions that are explained in the following two lemmas.

Lemma 2.1 *If a dynamical system is dissipative with respect to the supply rate $r(u, y)$, then it is also dissipative with respect to the supply rate $\lambda r(u, y)$ where $\lambda > 0$.*

Proof. Assume a dynamical system Σ is dissipative with respect to the supply rate $r(u, y)$. Then there exists a storage function $S(x)$ such that the dissipation inequality (2.1) holds. This means that

$$S(x_1) \leq S(x_0) + \int_{t_0}^{t_1} r(u, y)dt \tag{2.2}$$

Multiplying both sides of (2.2) by λ, we get

$$\lambda S(x_1) \leq \lambda S(x_0) + \lambda \int_{t_0}^{t_1} r(u, y)dt$$

Let $r_1(u, y) = \lambda r(u, y)$ and $S_1(x) = \lambda S(x)$. Then $S_1(x)$ is a storage function and $r_1(u, y)$ is a supply rate function. It follows that

$$S_1(x_1) \leq S_1(x_0) + \int_{t_0}^{t_1} r_1(u, y)dt$$

Therefore the system is dissipative with respect to the supply rate $\lambda r(u, y)$. ■

Lemma 2.2 *If a dynamical system Σ is dissipative with respect to the supply rate $r_i(u, y)$ for $i = 1,\ 2, ..., n$, then it is also dissipative with respect to the supply rate $r(u, y) = \sum_{i=1}^{n} r_i(u, y)$.*

Proof. Since the system Σ is dissipative with respect to the supply rate $r_i(u, y)$ for $i = 1,\ 2, ..., n$, there exists a storage function $S_i(x)$ such that the dissipation inequality (2.1) holds. This means that

$$S_i(x_1) \leq S_i(x_0) + \int_{t_0}^{t_1} r_i(u, y)dt \tag{2.3}$$

for $i = 1,\ 2, ..., n$. Taking the summation on both sides of (2.3) where i goes from 1 to n, we get

$$\sum_{i=1}^{n} S_i(x_1) \leq \sum_{i=1}^{n} S_i(x_0) + \sum_{i=1}^{n} \int_{t_0}^{t_1} r_i(u, y)dt \tag{2.4}$$

The inequality (2.4) is equivalent to

$$\sum_{i=1}^{n} S_i(x_1) \leq \sum_{i=1}^{n} S_i(x_0) + \int_{t_0}^{t_1} \sum_{i=1}^{n} r_i(u, y)dt \tag{2.5}$$

Let's define $r(u, y) = \sum_{i=1}^{n} r_i(u, y)$ and $S(x) = \sum_{i=1}^{n} S_i(x)$.Then (2.5) is equivalent to

$$S(x_1) \leq S(x_0) + \int_{t_0}^{t_1} r(u, y)dt \tag{2.6}$$

Therefore the system Σ is dissipative with respect to the supply rate $r(u, y)$. ■

Based on these lemmas, we can find supply rate functions for continuous-time dynamical systems.

2.4 An example continuous-time dissipative dynamical system

Consider a mass, spring and damper system [15]. The equation of motion is

$$m\ddot{x}(t) + D\dot{x}(t) + Kx(t) = u(t) \tag{2.7}$$

where M is the mass, D is the damper constant, K is the spring stiffness, x is the position of the mass and u is the force acting on the mass. The initial conditions are $x(0) = x_0$ and $\dot{x}(0) = \dot{x}_0$. It is assumed that $M > 0$, $D \geq 0$ and $K \geq 0$.

The energy of the system is

$$V(x, \dot{x}) = 0.5m\dot{x}^2 + 0.5Kx^2$$

The time derivative of the energy is

$$\dot{V}(x, \dot{x}) = m\ddot{x}\dot{x} + Kx\dot{x} = u\dot{x} - D\dot{x}^2$$

Let's define $x_1 = x$, $x_2 = \dot{x}$ and $y = \dot{x}$ as state variables and the output of the system (2.7), respectively. Then

$$\dot{V}(\mathbf{x}) = uy - D\dot{x}^2 \tag{2.8}$$

where $\mathbf{x} = [x_1 \; x_2]$.

Integrating both sides of equation (2.8) from $t = 0$ to $t = T$ gives

$$V(\mathbf{x}(T)) = V(\mathbf{x}(0)) + \int_0^T uy dt - \int_0^T D\dot{x}^2 dt \tag{2.9}$$

where $V(\mathbf{x}(T))$ is the energy at $t = T$ and $V(\mathbf{x}(0))$ is the initial energy. The second term and the last term on the right side of (2.9) are the energy supplied by the external source and the energy dissipated by the damper, respectively.

From (2.9) we get

$$V(\mathbf{x}(T)) \leq V(\mathbf{x}(0)) + \int_0^T uy dt \tag{2.10}$$

since

$$\int_0^T D\dot{x}^2 dt \geq 0 \tag{2.11}$$

Let's define $S(\mathbf{x}) = V(\mathbf{x})$ and $r(u, y) = uy$. Then $S(\mathbf{x})$ is a storage function and $r(u, y)$ is a supply rate. The inequality (2.10) means that the system (2.7) is dissipative.

In this example, the storage function is the stored energy and the supply rate is the absorbed input power. The dissipativity of the system says that the change in its stored energy is not greater than the absorbed input power.

2.5 Stability of interconnected continuous-time dissipative dynamical systems

Consider dynamical systems Σ_1 and Σ_2 that are interconnected via constraints $\mathbf{u}_1 = -\mathbf{y}_2$ and $\mathbf{u}_2 = \mathbf{y}_1$, as shown in Fig. 2.1. Suppose that equilibrium points of systems Σ_1 and Σ_2 are located at the origin.

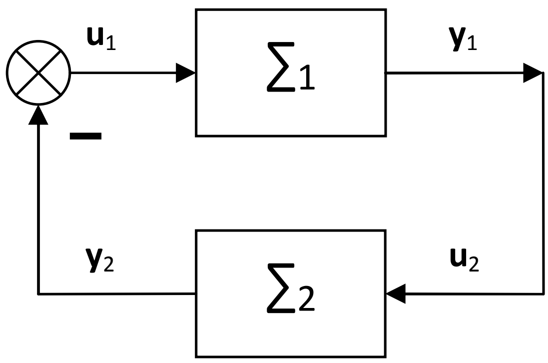

Figure 2.1: Interconnected continuous-time systems

Theorem 2.1 *[11] If the system Σ_1 is dissipative with respect to the supply rate $r_1(\boldsymbol{u}_1, \boldsymbol{y}_1)$, the system Σ_2 is dissipative with respect to the supply rate $r_2(\boldsymbol{u}_1, \boldsymbol{y}_2)$ and $r_1(\boldsymbol{u}_1, \boldsymbol{y}_2) + r_2(\boldsymbol{u}_1, \boldsymbol{y}_2) = 0$ then the origin of the feedback system is stable.*

Proof. Since the system Σ_1 is dissipative with respect to $r_1(u_1, y_1)$, there exists a storage function $S_1(\mathbf{x}_1) \geq 0$ such that $\dot{S}_1(\mathbf{x}_1) \leq r_1(u_1, y_1)$. Since the system Σ_2 is dissipative with respect to $r_2(u_2, y_2)$, there exists a storage function $S_2(\mathbf{x}_2) \geq 0$ such that $\dot{S}_2(\mathbf{x}_2) \leq r_2(u_2, y_2)$. Thus $S_1(\mathbf{x}_1) + S_2(\mathbf{x}_2) \geq 0$ and $\dot{S}_1(\mathbf{x}_1) + \dot{S}_2(\mathbf{x}_2) \leq r_1(u_1, y_1) + r_2(u_2, y_2)$. Since

$r_1(u_1, y_1) + r_2(u_2, y_2) = 0$, $\dot{S}_1(\mathbf{x}_1) + \dot{S}_2(\mathbf{x}_2) \leq 0$. Let's define $V(\mathbf{x}) = S_1(\mathbf{x}_1) + S_2(\mathbf{x}_2)$. Then $\dot{V}(\mathbf{x}) \leq 0$. Therefore the origin of the feedback system is stable. ■

In the next chapter, this theorem will be extended to the case where Σ_1 is static and Σ_2 is a discrete-time dynamical system.

CHAPTER 3

STABILITY ANALYSIS OF RECURRENT NEURAL NETWORKS USING DISSIPATIVITY

The previous chapter introduced the concept of dissipativity for continuous time dynamic systems and demonstrated how dissipativity can be used to prove stability for interconnected dissipative systems. In this chapter, we extend these ideas to analyze the stability of discrete-time recurrent neural networks. We begin by defining dissipativity for static and discrete-time dynamic systems and update the stability theorem for interconnected dissipative systems. Then we introduce a general framework for representing RNNs and show how this general framework can be represented in a standard form. From the standard form we can apply the stability theorem for interconnected dissipative systems. We will derive three different criteria for testing the stability of RNNs through different choices of supply rate functions.

3.1 Dissipative systems

3.1.1 Static systems

Consider the system

$$\mathbf{y} = \mathbf{f}(\mathbf{u}) \tag{3.1}$$

where $\mathbf{u} \in U \subseteq R^m,\ \mathbf{y} \in Y \subseteq R^l,\ \mathbf{f} : U \to Y$ and $\mathbf{0} \in U.$ Without loss of generality, let $\mathbf{f}(\mathbf{0}) = \mathbf{0}$.

Definition 3.1 *The system (3.1) is called a static system if the outputs* $\mathbf{y}$ *depend only on current values of inputs* $\boldsymbol{u}$.

3.1.2 Discrete-time dynamical systems

There are several possible representations of discrete-time dynamical systems. The representation used in [15] is

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{h}(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) \end{cases} \tag{3.2}$$

where $\mathbf{x}(k) \in D \subseteq R^n,\ \mathbf{u} \in U \subseteq R^m,\ \mathbf{y} \in Y \subseteq R^l,\ \mathbf{h} : D \times U \to R^n,\ \mathbf{g} : D \times U \to Y$, and D is open and contains $\mathbf{0}$. It is assumed that h and g are continuous mappings, and h has at least one equilibrium point. Without loss of generality, suppose $\mathbf{h}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$ and $\mathbf{g}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$.

3.1.3 Dissipative systems

Definition 3.2 *A function $r : U \times Y \to R$ is a supply rate for a given system if the following conditions are satisfied:*

1. $r(\boldsymbol{u}, \boldsymbol{y}) = 0$ *if and only if* $\boldsymbol{u} = \boldsymbol{0}$.
2. $\sum_{k=k_1}^{k=k_2} |r(\boldsymbol{u}(k), \boldsymbol{y}(k))| < \infty$, *for all* $k_1,\ k_2 \in Z_+$.

Definition 3.3 *The discrete-time dynamical system (3.2) is said to be dissipative with respect to the supply rate $r(\boldsymbol{u}, \boldsymbol{y})$ if there exists a continuous radially unbounded, positive definite function $V : D \to R$ satisfying $V(\boldsymbol{0}) = 0$ and the following inequality*

$$V(\boldsymbol{x}(k+1)) - V(\boldsymbol{x}(k_0)) \leq \sum_{i=k_0}^{k} r(\boldsymbol{u}(i), \boldsymbol{y}(i)) \tag{3.3}$$

holds for all $k_0,\ k \in Z_+$ such that $k \geq k_0$, where $\boldsymbol{x}(k)$ is the solution of system (3.2) with $\boldsymbol{u} \in U$. The function V is called a storage function. If the inequality in (3.3) is strict for $\boldsymbol{x}(k) \neq \boldsymbol{0}$, then system (3.2) is said to be strictly dissipative.

Definition 3.4 *The static system (3.1) is said to be dissipative with respect to the supply rate $r(\boldsymbol{u}, \boldsymbol{y})$ if $r(\boldsymbol{u}, \boldsymbol{y}) \geq 0$ for all $\boldsymbol{u} \in U$. If the inequality is strict for $\boldsymbol{u} \neq \boldsymbol{0}$, then (3.1) is strictly dissipative with respect to $r(\boldsymbol{u}, \boldsymbol{y})$.*

Definition 3.5 *[15, p. 803] The system (3.2) is said to be zero-state observable if* $\boldsymbol{u}(k) = \boldsymbol{0}$ *and* $\boldsymbol{y}(k) = \boldsymbol{0}$ *implies* $\boldsymbol{x}(k) = \boldsymbol{0}$.

Theorem 3.1 *If the storage function* V *satisfies*

$$V(\boldsymbol{x}(n+1)) - V(\boldsymbol{x}(n)) \leq r(\boldsymbol{u}(n), \boldsymbol{y}(n))$$

for all $n \in Z_+$*, then (3.2) is dissipative with respect to supply rate* $r(\boldsymbol{u}, \boldsymbol{y})$*. Moreover, if the inequality is strict for* $\boldsymbol{x}(n) \neq \boldsymbol{0}$*, then (3.2) is strictly dissipative with respect to* $r(\boldsymbol{u}, \boldsymbol{y})$*.*

Proof. We will prove that the system is dissipative. The proof for the strictly dissipative case follows in a similar manner. Since the inequality holds for all $n \in Z_+$, it holds for $k_0, k_0+1, ..., k-1, k$ where $k_0 \in Z_+$. Thus we get

$$\begin{aligned} V(\mathbf{x}(k_0+1)) - V(\mathbf{x}(k_0)) &\leq r(\mathbf{u}(k_0), \mathbf{y}(k_0)) \\ V(\mathbf{x}(k_0+2)) - V(\mathbf{x}(k_0+1)) &\leq r(\mathbf{u}(k_0+1), \mathbf{y}(k_0+1)) \\ &\vdots \\ V(\mathbf{x}(k+1)) - V(\mathbf{x}(k)) &\leq r(\mathbf{u}(k), \mathbf{y}(k)) \end{aligned}$$

Adding these inequalities together, we have

$$V(\mathbf{x}(k+1)) - V(\mathbf{x}(k_0)) \leq \sum_{i=k_0}^{k} r(\mathbf{u}(i), \mathbf{y}(i))$$

Therefore, (3.2) is dissipative with respect to $r(\mathbf{u}, \mathbf{y})$. ■

3.2 Stability

In this section we will define stability for discrete-time systems and then update Theorem 2.1 on the stability of interconnected dissipative systems to include the case of static and discrete-time subsystems.

Consider system (3.2) with equilibrium point $\mathbf{x}_e = \mathbf{0}$.

Definition 3.6 *[15, p. 765] The equilibrium point $\boldsymbol{x}_e$ is said to be Lyapunov stable if for all $\epsilon > 0$ there exists $\delta > 0$ such that if $||\boldsymbol{x}(0)|| < \delta$ then $||\boldsymbol{x}(k)|| < \epsilon$ for all $k \in Z_+$. In addition, it is said to be globally asymptotically stable (GAS) if it is Lyapunov stable and for all $\boldsymbol{x}(0) \in R^n$, $\lim_{k\to\infty} \boldsymbol{x}(k) = \boldsymbol{0}$.*

3.2.1 STABILITY ANALYSIS OF RECURRENT NEURAL NETWORKS USING DISSIPATIVITY

Consider the interconnected system shown in Fig. 3.1, where subsystem Σ_1 is a static system of the form (3.1), subsystem Σ_2 is a discrete-time dynamical system of the form (3.2), $\mathbf{u}_1(k) = -\mathbf{y}_2(k)$ and $\mathbf{u}_2(k) = \mathbf{y}_1(k)$. Assume that $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ and the system Σ_2 has an equilibrium point $\mathbf{x}_e = \mathbf{0}$.

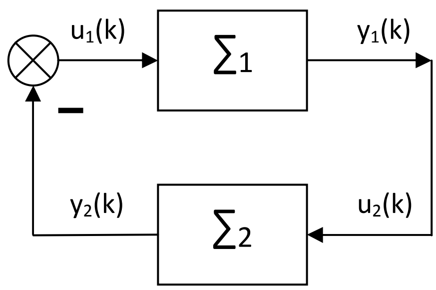

Figure 3.1: Interconnected systems

Theorem 3.2 *If subsystem Σ_1 is dissipative with respect to the supply rate $r_1(\boldsymbol{u}_1(k), \boldsymbol{y}_1(k))$, subsystem Σ_2 is dissipative with respect to the supply rate $r_2(\boldsymbol{u}_2(k), \boldsymbol{y}_2(k))$, and*

$$r_1(\boldsymbol{u}_1(k), \boldsymbol{y}_1(k)) + r_2(\boldsymbol{u}_2(k), \boldsymbol{y}_2(k)) \leq 0,$$

then the origin of the interconnected system is stable. Moreover, the origin of the system is globally asymptotic stable (GAS), if the system Σ_2 is zero state observable and one of the following additional conditions holds:

1. *Either Σ_1 is strictly dissipative with respect to the supply rate $r_1(\boldsymbol{u}_1(k), \boldsymbol{y}_1(k))$ or Σ_2 is strictly dissipative with respect to the supply rate $r_2(\boldsymbol{u}_2(k), \boldsymbol{y}_2(k))$.*

2. $r_1(\boldsymbol{u}_1(k), \boldsymbol{y}_1(k)) + r_2(\boldsymbol{u}_2(k), \boldsymbol{y}_2(k)) < 0$ *when the variables are not all equal to zero.*

Proof. Since Σ_1 is static and dissipative with respect to the supply rate $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k))$, $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) \geq 0$. Since subsystem Σ_2 is dissipative with respect to the supply rate $r_2(\mathbf{u}_2(k), \mathbf{y}_2(k))$, there exists a storage function $V_2(\mathbf{x}_2(k))$ such that

$$V_2(\mathbf{x}_2(k+1)) - V_2(\mathbf{x}_2(k)) \leq r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)).$$

Since $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) \geq 0$, $r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) \leq -r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) \leq 0$. Thus $V_2(\mathbf{x}_2(k+1)) - V_2(\mathbf{x}_2(k)) \leq 0$. $V_2(\mathbf{x}_2(k))$ is a Lyapunov function for the closed loop system, therefore the origin of the system is stable. If either the subsystem Σ_1 or the subsystem Σ_2 is strictly dissipative, then

$$V_2(\mathbf{x}_2(k+1)) - V_2(\mathbf{x}_2(k)) < 0.$$

If both subsystem Σ_1 and subsystem Σ_2 are dissipative, and

$$r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) + r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) < 0,$$

then

$$V_2(\mathbf{x}_2(k+1)) - V_2(\mathbf{x}_2(k)) < 0.$$

In either case, if $V_2(\mathbf{x}_2(k+1)) - V_2(\mathbf{x}_2(k)) = 0$ then $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) = r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) = 0$. So $\mathbf{u}_1(k) = \mathbf{0}$ and $\mathbf{u}_2(k) = \mathbf{0}$ by definition of supply rate. Thus $\mathbf{y}_2(k) = \mathbf{0}$ by constraints of interconnected systems. Since the system Σ_2 is zero state observable, $\mathbf{x}_2(k) = \mathbf{0}$. This shows that the origin of the system is GAS. ■

3.2.2 Example use of dissipativity on a simple network

To introduce our proposed method, we start with an example of a recurrent neuron with a zero bias

$$x(k+1) = tanh(wx(k)). \tag{3.4}$$

This system (3.4) has an equilibrium point $x_e = 0$, and the transfer function $tanh$ satisfies the condition $0 \leq \frac{tanh(u)}{u} \leq 1$. Now let's transform the system (3.4) into two interconnected subsystems as in Fig. 3.1, where the subsystem Σ_1 is $y_1(k) = tanh(u_1(k))$, the subsystem Σ_2 is $y_2(k) = -wu_2(k-1)$, and $u_1(k) = -y_2(k)$ and $u_2(k) = y_1(k)$.

Let's choose a supply rate

$$r_1(u_1(k), y_1(k)) = u_1^2(k) - y_1^2(k).$$

Then

$$r_1(u_1(k), y_1(k)) = u_1^2(k) - tanh^2(u_1(k)).$$

Thus $r_1(u_1(k), y_1(k)) \geq 0$. Therefore by definition 3.4, the system Σ_1 is dissipative with respect to the supply rate $r_1(u_1(k), y_1(k))$.

Let

$$r_2(u_2(k), y_2(k)) = -r_1(u_1(k), y_1(k)).$$

Then

$$\begin{aligned} r_2(u_2(k), y_2(k)) &= -u_1^2(k) + y_1^2(k) \\ &= u_2^2(k) - y_2^2(k) \\ &= u_2^2(k) - w^2 u_2^2(k-1). \end{aligned}$$

Let $x_2(k) = u_2(k-1)$. Then $x_2(k+1) = u_2(k)$ and $y_2(k) = -wx_2(k)$. If $u_2(k) = 0$ and $y_2(k) = 0$, then $x_2(k) = 0$. So the system Σ_2 is zero state observable. Choose $V_2(x_2(k)) = qx_2^2(k)$, where $q > 0$. Then $V_2(x_2(k)) = qu_2^2(k-1)$ and $V_2(x_2(k+1)) = qu_2^2(k)$.

We claim that if $w^2 < 1$, then the origin of (3.4) is GAS. Since $w^2 < 1$, there exists $q > 0$ such that $w^2 < q < 1$. So

$$(1-q)u_2^2(k) + (q - w^2)u_2^2(k-1) > 0.$$

It follows that

$$u_2^2(k) - w^2 u_2^2(k-1) > q u_2^2(k) - q u_2^2(k-1).$$

Thus

$$V_2(x_2(k+1)) - V_2(x_2(k)) < r_2(u_2(k), y_2(k)).$$

Therefore the system Σ_2 is strictly dissipative with respect to the supply rate $r_2(u_2(k), y_2(k))$. By Theorem 3.2, the origin is GAS for the closed loop system. In the next section, this result is generalized for LDDNs.

3.3 Stability analysis of Layered Digital Dynamic Networks (LDDNs)

In this section, we introduce a general framework for representing recurrent neural networks-the Layered Digital Dynamic Network. We then show how this class of network can be represented in a standard form, and how the standard form can be represented in the interconnected system form, so that Theorem 3.2 can be used to demonstrate stability. By selecting different supply rate functions, we develop three different criteria for determining stability of LDDNs.

3.3.1 LDDNs

In this section, we want to describe the LDDN framework, first introduced in [9]. An example LDDN is shown in Fig. 3.2. The net input $n^m(k)$ for layer m of an LDDN can be computed

$$\begin{aligned}\mathbf{n}^m(k) = &\sum_{l \in L_m^f} \sum_{d \in DL_{m,l}} \mathbf{LW}^{m,l}(d)\mathbf{a}^l(k-d) \\ &+ \sum_{l \in I_m} \sum_{d \in DI_{m,l}} \mathbf{IW}^{m,l}(d)\mathbf{p}^l(k-d) + \mathbf{b}^m \end{aligned} \tag{3.5}$$

where $\mathbf{p}^l(k)$ is the lth input to the network at time k, $\mathbf{IW}^{m,l}$ is the input weight between input l and layer m, $\mathbf{LW}^{m,l}$ is the layer weight between layer l and layer m, $\mathbf{b}^m$ is the bias vector for layer m, $DL_{m,l}$ is the set of all delays in the tapped delay line between layer l

and layer m, I_m is the set of indices of input vectors that connect to layer m, and L_m^f is the set of indices of layers that connect directly forward to layer m. The output of layer m is

$$\mathbf{a}^m(k) = \mathbf{f}^m(\mathbf{n}^m(k)) \tag{3.6}$$

for $m = 1,\ 2,\ \cdots,\ M$. The set of M paired equations (3.5) and (3.6) describes the LDDN.

3.3.2 Transform LDDNs to a standard form

The LDDN is described by (3.5) and (3.6). Our goal in this section is to transform the LDDN into the form of (3.7), which we will call the standard form. It is assumed that the matrix $\mathbf{W}^2$ has the property that it can be transformed into a strictly triangular matrix through a re-ordering of the elements of the state vector $\mathbf{x}$. This guarantees that $\mathbf{x}(k)$ can be solved iteratively from some initial $\mathbf{x}(k_0)$. This is equivalent to the condition that the LDDN contains no zero-delay loops.

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{W}^1\mathbf{x}(k) + \mathbf{W}^2\mathbf{x}(k+1) + \mathbf{b}) \tag{3.7}$$

Assume that inputs $\mathbf{p}^l(k)$ are constant. Let S^m be the number of neurons in layer m. Let

$$\begin{aligned}
\mathbf{h}^m &= \sum_{l\in I_m}\sum_{d\in DI_{m,l}} \mathbf{IW}^{m,l}(d)\mathbf{p}^l(k-d) + \mathbf{b}^m,\\
\mathbf{b} &= [(\mathbf{h}^1)^T\ 0\cdots(\mathbf{h}^2)^T\ 0\cdots(\mathbf{h}^M)^T\cdots 0]^T_{(S\times 1)},\\
d^l &= max\left\{d\in DL_{m,l}|m = 1,\ 2,\ ...,\ M\right\},
\end{aligned}$$

where $S = \sum_{i=1}^{M} S^i d^i$.

Consider layer m for $m = 1,\ 2,\ ...,\ M$. Let $\mathbf{x}_1^m(k) = \mathbf{a}^m(k-1)$, $\mathbf{x}_2^m(k) = \mathbf{a}^m(k-2)$, ..., $\mathbf{x}_{d^m}^m(k) = \mathbf{a}^m(k-d^m)$. Then $\mathbf{x}_1^m(k+1) = \mathbf{a}^m(k)$, $\mathbf{x}_2^m(k+1) = \mathbf{a}^m(k-1)$, ..., $\mathbf{x}_{d^m}^m(k+1) = \mathbf{a}^m(k-d^m+1)$. Let $\mathbf{x}^m(k) = [\mathbf{x}_1^m(k)\ \mathbf{x}_2^m(k)\ ...\ \mathbf{x}_{d^m}^m(k)]^T_{(S^m d^m\times 1)}$. Let $\mathbf{x}(k) = [\mathbf{x}^1(k)\ \mathbf{x}^2(k)\ ...\ \mathbf{x}^M(k)]^T_{(S\times 1)}$. Then we get the standard form (3.7), where $\mathbf{W}^1$ and $\mathbf{W}^2$ are defined in (3.8) and (3.9), respectively.

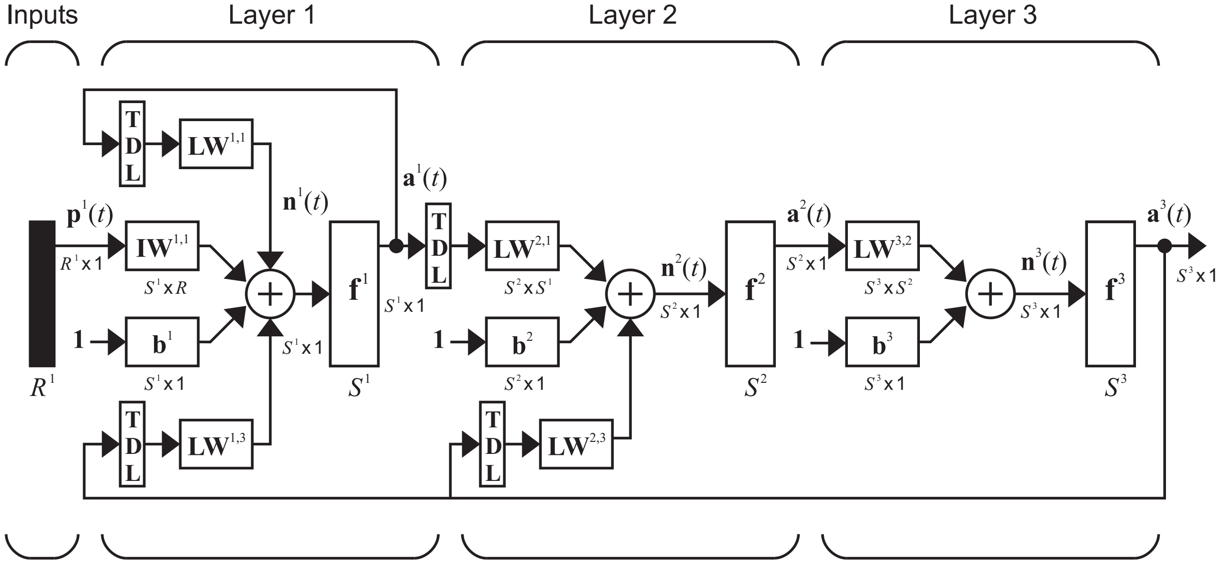

Figure 3.2: Example LDDN network

$$\mathbf{W}^1 = [\mathbf{W}^1_{i,j}]_{(S\times S)} \tag{3.8}$$

where

$$\mathbf{W}^1_{i,i} = \begin{bmatrix} \mathbf{LW}^{i,i}(1) & \mathbf{LW}^{i,i}(2) & \cdots & \mathbf{LW}^{i,i}(d^i-1) & \mathbf{LW}^{i,i}(d^i) \\ \mathbf{I}_{S^i} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{S^i} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{S^i} & \mathbf{0} \end{bmatrix}_{(S^i d^i \times S^i d^i)}$$

and

$$\mathbf{W}^1_{i,j} = \begin{bmatrix} \mathbf{LW}^{i,j}(1) & \mathbf{LW}^{i,j}(2) & \cdots & \mathbf{LW}^{i,j}(d^j-1) & \mathbf{LW}^{i,j}(d^j) \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix}_{(S^i d^i \times S^j d^j)}$$

if $i \neq j$.

$$\mathbf{W}^2 = [\mathbf{W}^2_{i,j}]_{(S\times S)} \tag{3.9}$$

where $\mathbf{W}^2_{i,j} = [\mathbf{0}]_{(S^i d^i \times S^j d^j)}$ if $i \leq j$, and

$$\mathbf{W}^2_{i,j} = \begin{bmatrix} \mathbf{LW}^{i,j}(0) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}_{(d^i \times d^j)}$$

if $i > j$. Also,

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_m \end{bmatrix}_{(S \times 1)} \qquad where\ \mathbf{f}_i = \begin{bmatrix} \mathbf{f}^i \\ \mathbf{id}^i \\ \vdots \\ \mathbf{id}^i \end{bmatrix}_{(d^i S^i \times 1)} \tag{3.10}$$

where $\mathbf{I}_{S^i}$ is an identity matrix with dimensions $(S^i \times S^i)$, $\mathbf{0}$ is a zero matrix with appropriate dimensions, $\mathbf{id}^i$ is a vector of identity functions, and $\mathbf{f}^i$ is a vector of transfer functions of layer i for $i = 1,\ 2,\ ...,\ M$. For an LDDN, the order in which the individual layer outputs must be computed to obtain the correct network output is called the simulation order (see [9]). Here we have assumed that the layers are numbered so that the simulation order (which need not be unique) increases with layer number.

Suppose that the LDDN of equations (3.5) and (3.6) has an equilibrium point. Then system (3.7) has an equilibrium point. Let $\mathbf{x}_e$ be that equilibrium point. Then $\mathbf{x}_e$ satisfies $\mathbf{x}_e = \mathbf{f}(\mathbf{W}^1\mathbf{x}_e + \mathbf{W}^2\mathbf{x}_e +\ \mathbf{b})$. Let $\mathbf{z}(k) = \mathbf{x}(k) - \mathbf{x}_e$. Then

$$\mathbf{z}(k+1) = \mathbf{f}(\mathbf{W}^1 z(k) + \mathbf{W}^2\mathbf{z}(k+1) + \mathbf{t}_e) - \mathbf{f}(\mathbf{t}_e) \tag{3.11}$$

where

$$\mathbf{t}_e = \mathbf{W}^1 x_e + \mathbf{W}^2\mathbf{x}_e + \mathbf{b}. \tag{3.12}$$

Therefore system (3.11) has an equilibrium point $\mathbf{z}_e = 0$. If this equilibrium point is GAS, then the equilibrium point of system (3.7) and that of the LDDN are also GAS. The next step is to transform system (3.11) into the interconnected system form shown in Fig. 3.1.

Let

$$\mathbf{u}_1(k) = -\mathbf{y}_2(k) = \mathbf{W}^1\mathbf{z}(k) + \mathbf{W}^2\mathbf{z}(k+1)$$
$$\mathbf{u}_2(k) = \mathbf{y}_1(k) = \mathbf{z}(k+1).$$

Then subsystem Σ_1 is

$$\mathbf{y}_1(k) = \mathbf{f}(\mathbf{u}_1(k) + \mathbf{t}_e) - \mathbf{f}(\mathbf{t}_e) = \mathbf{g}(u_1(k)), \tag{3.13}$$

and subsystem Σ_2 is

$$\mathbf{y}_2(k) = -\mathbf{W}^1\mathbf{u}_2(k-1) - \mathbf{W}^2\mathbf{u}_2(k), \tag{3.14}$$

and the constraints are

$$\mathbf{u}_1(k) = -\mathbf{y}_2(k),$$
$$\mathbf{u}_2(k) = \mathbf{y}_1(k).$$

Define $\mathbf{x}_2(k) = \mathbf{u}_2(k-1)$, then $\mathbf{x}_2(k+1) = \mathbf{u}_2(k)$ and $\mathbf{y}_2(k) = -\mathbf{W}^1\mathbf{x}_2(k) - \mathbf{W}^2\mathbf{u}_2(k)$. In this case, subsystem Σ_1 is a static system of the form (3.1), and Σ_2 is a dynamic system of the form (3.2). If $\mathbf{u}_2(k) = \mathbf{0}$ and $\mathbf{y}_2(k) = \mathbf{0}$ then $\mathbf{x}_2(k) = \mathbf{0}$, so the system Σ_2 is zero state observable. Since $\mathbf{z}_e = \mathbf{0}$ is the equilibrium point of (3.11), it is also the equilibrium point of subsystems Σ_1 and Σ_2. Therefore, if the origin of the interconnected systems is GAS, then the equilibrium point of the LDDN is also GAS. To analyze stability of the equilibrium point, we follow Theorem 3.2. The following sections will perform this analysis and will develop several stability criteria.

3.3.3 Sector conditions

Consider a scalar static system of the form (3.1), $y = f(u)$. The function f is said to lie inside a sector $[\alpha, \beta]$ (written as $f \in [\alpha, \beta]$) if $\alpha \leq \frac{f(u)}{u} \leq \beta$, $\forall u \neq 0$ and $f(0) = 0$. This is called a sector condition. An example function $f(u)$ (solid curve) and its bounds αu and βu (dashed lines) are shown in Fig. 3.3.

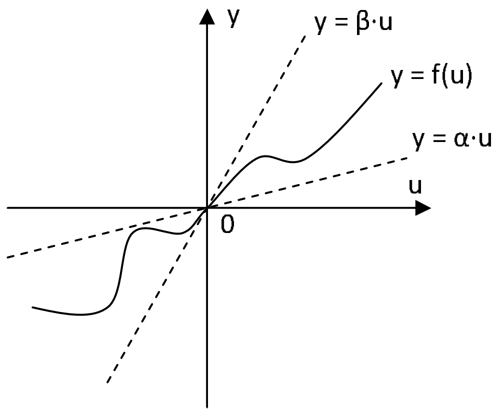

Figure 3.3: The function *f(u)* satisfying sector conditions

3.3.4 Supply rate using sector conditions - static/scalar

It is possible to use sector conditions to design supply rate functions for static systems. In this section we consider the scalar version of the static system (3.1).

Lemma 3.1 *If the function $f \in [\alpha, \beta]$, and the supply rate is chosen as either*

$$r(u, y) = \beta^2 u^2 - y^2$$

(using the sector upper bound) or

$$r(u, y) = (\beta u - y)(y - \alpha u),$$

(using the sector upper and lower bounds), then the scalar static system $y = f(u)$ is dissipative with respect to the supply rate $r(u, y)$.

Proof. From the sector condition, it follows that $\beta^2 u^2 \geq y^2$, $(\beta u - y)(y - \alpha u) \geq 0$. Thus $r(u, y) = \beta^2 u^2 - y^2 \geq 0$ and $r(u, y) = (\beta u - y)(y - \alpha u) \geq 0$. In either case, $r(u, y) \geq 0$. Thus the static system is dissipative with respect to the supply rate $r(u, y)$ by definition 3.4. ■

3.3.5 Supply rate using sector conditions - static/vector

In the case that system (3.1) is multi-input multi-output, let's assume $\mathbf{u} = [u_1\ u_2 \cdots u_n]^T$, $\mathbf{y} = [y_1\ y_2 \cdots y_n]^T$, and $\mathbf{f} = [f_1\ f_2 \cdots f_n]^T$. We will consider functions $\mathbf{f}$ such that $y_i = f_i(u_i)$. Suppose that $f_i \in [\alpha_i, \beta i]$, for $i = 1, 2, ..., n$. Let $\mathbf{A} = diag(\alpha_i)$ and $\mathbf{B} = diag(\beta_i)$.

Lemma 3.2 *If the supply rate is chosen as either $r(\boldsymbol{u},\boldsymbol{y}) = \boldsymbol{u}^T\boldsymbol{B}^2\boldsymbol{u} - \boldsymbol{y}^T\boldsymbol{y}$ or $r(\boldsymbol{u},\boldsymbol{y}) = (\boldsymbol{Bu} - \boldsymbol{y})^T\boldsymbol{T}(\boldsymbol{y} - \boldsymbol{Au}) + \boldsymbol{u}^T\Lambda\boldsymbol{y}$ where $\boldsymbol{T}$ is a positive definite diagonal matrix and Λ is a diagonal matrix with non-negative elements, then the system $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{u})$ is dissipative with respect to the supply rate $r(\boldsymbol{u},\boldsymbol{y})$.*

Proof. Let $\mathbf{T} = diag(t_i)$ and $\Lambda = diag(\lambda_i)$ where $t_i > 0,\ \lambda_i \geq 0$ for $i = 1,\ 2, ...,\ n$. Since $\mathbf{f}_i \in [\alpha_i, \beta i]$, $u_i^2\beta_i^2 - y_i^2 \geq 0$, so

$$\mathbf{u}^T\mathbf{B}^2\mathbf{u} - \mathbf{y}^T\mathbf{y} = \sum_{i=1}^{n}(u_i^2\beta_i^2 - y_i^2) \geq 0.$$

Since $t_i > 0,\ \lambda_i \geq 0$ and $f_i \in [\alpha_i, \beta i]$, $(\beta_i u_i - y_i)t_i(y_i - \alpha_i u_i) + u_i\lambda_i y_i \geq 0$, so

$$(\mathbf{Bu} - \mathbf{y})^T\mathbf{T}(\mathbf{y} - \mathbf{Au}) + \mathbf{u}^T\Lambda\mathbf{y} = \sum_{i=1}^{n}[(\beta_i u_i - y_i)t_i(y_i - \alpha_i u_i) + u_i\lambda_i y_i] \geq 0.$$

In either case, $r(\mathbf{u}, \mathbf{y}) \geq 0$. Thus the system is dissipative with respect to the supply rate $r(\mathbf{u}, \mathbf{y})$ by definition 3.4. ■

3.3.6 Selecting the supply rate for LDDNs

Consider an LDDN that has been put into the standard form (3.11) and then put into the interconnected systems form of (3.13) and (3.14). Assume that the function $g_i \in [\alpha_i, \beta_i]$ for $i = 1,\ 2,\ ...,\ S$. Let $\mathbf{A} = diag(\alpha_i)$ and $\mathbf{B} = diag(\beta_i)$. Choose $V_2(x_2(k)) = x_2^T(k)\mathbf{Q}x_2(k)$, where $\mathbf{Q}$ is a positive definite matrix.

Supply rate using sector upper bounds

Choose $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) = \mathbf{u}_1^T(k)\mathbf{B}^2\mathbf{u}_1(k) - \mathbf{y}_1^T(k)\mathbf{y}_1(k)$. Then the system Σ_1 is dissipative with respect to the supply rate $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k))$ by Lemma 3.2. Choose $r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) =$

$-r_1(\mathbf{u}_1(k), \mathbf{y}_1(k))$. Then $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) + r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) = 0$.

Using the constraints of interconnected systems, we get

$$r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) = \mathbf{u}_2^T(k)\mathbf{u}_2(k) - \mathbf{y}_2^T(k)\mathbf{B}^2\mathbf{y}_2(k).$$

Thus

$$\begin{aligned} r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) &= \mathbf{u}_2^T(k)\mathbf{u}_2(k) \\ &\quad - [\mathbf{W}^1\mathbf{u}_2(k-1) + \mathbf{W}^2\mathbf{u}_2(k)]^T\mathbf{B}^2[\mathbf{W}^1\mathbf{u}_2(k-1) + \mathbf{W}^2\mathbf{u}_2(k)] \\ &= \mathbf{u}_2^T(k)\mathbf{u}_2(k) - [\mathbf{u}_2^T(k-1)(\mathbf{W}^1)^T\mathbf{B}^2\mathbf{W}^1\mathbf{u}_2(k-1) \\ &\quad + \mathbf{u}_2^T(k)(\mathbf{W}^2)^T\mathbf{B}^2\mathbf{W}^1\mathbf{u}_2(k-1) \\ &\quad + \mathbf{u}_2^T(k-1)(\mathbf{W}^1)^T\mathbf{B}^2\mathbf{W}^2\mathbf{u}_2(k) \\ &\quad + \mathbf{u}_2^T(k)(\mathbf{W}^2)^T\mathbf{B}^2\mathbf{W}^2\mathbf{u}_2(k)]. \end{aligned}$$

Let

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I} - \mathbf{Q} - (\mathbf{W}^2)^T\mathbf{B}^2\mathbf{W}^2 & -(\mathbf{W}^2)^T\mathbf{B}^2\mathbf{W}^1 \\ -(\mathbf{W}^1)^T\mathbf{B}^2\mathbf{W}^2 & \mathbf{Q} - (\mathbf{W}^1)^T\mathbf{B}^2\mathbf{W}^1 \end{bmatrix} \tag{3.15}$$

.

Theorem 3.3 *If a positive definite matrix $\mathbf{Q}$ can be found such that the matrix $\mathbf{P}_1$ is positive definite, then the equilibrium point of the LDDN is GAS.*

Proof. Since $\mathbf{P}_1$ is positive definite, $[\mathbf{u}_2^T(k)\ \mathbf{u}_2^T(k-1)]\mathbf{P}_1[\mathbf{u}_2^T(k)\ \mathbf{u}_2^T(k-1)]^T > 0$. Thus

$$\begin{aligned} &\mathbf{u}_2^T(k)[\mathbf{I} - \mathbf{Q} - (\mathbf{W}^2)^T\mathbf{B}^2\mathbf{W}^2]\mathbf{u}_2(k) - \mathbf{u}_2^T(k)(\mathbf{W}^2)^T\mathbf{B}^2\mathbf{W}^1\mathbf{u}_2(k-1) \\ &\quad - \mathbf{u}_2^T(k-1)(\mathbf{W}^1)^T\mathbf{B}^2\mathbf{W}^2\mathbf{u}_2(k) + \mathbf{u}_2^T(k-1)[\mathbf{Q} - (\mathbf{W}^1)^T\mathbf{B}^2\mathbf{W}^1]\mathbf{u}_2(k-1) > 0. \end{aligned}$$

It follows that $V_2(\mathbf{x}_2(k+1)) - V_2(\mathbf{x}_2(k)) < r_2(\mathbf{u}_2(k), \mathbf{y}_2(k))$. Thus the system Σ_2 is strictly dissipative with respect to the supply rate $r_2(\mathbf{u}_2(k), \mathbf{y}_2(k))$. In addition, the system Σ_1 is dissipative with respect to the supply rate $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k))$, as proved above. Therefore the origin of the interconnected system is GAS by Theorem 3.2. Consequently, the equilibrium point of the LDDN is GAS. ■

Supply rate using sector lower and upper bounds

Choose $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) = (\mathbf{y}_1(k) - \mathbf{A}\mathbf{u}_1(k))^T\mathbf{T}(\mathbf{B}\mathbf{u}_1(k)) - \mathbf{y}_1(k)) + \mathbf{u}_1^T(k)\Lambda\mathbf{y}_1(k)$, where $\mathbf{T}$ is a positive definite diagonal matrix and Λ is an diagonal matrix with non-negative elements. Then the system Σ_1 is dissipative with respect to the supply rate $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k))$ by Lemma 3.2.

Choose $r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) = -r_1(\mathbf{u}_1(k), \mathbf{y}_1(k))$. Then $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) + r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) = 0$. Using the constraints of interconnected systems, we get

$$r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) = (\mathbf{u}_2(k) + \mathbf{A}\mathbf{y}_2(k))^T\mathbf{T}(\mathbf{B}\mathbf{y}_2(k)) + \mathbf{u}_2(k)) + \mathbf{y}_2^T(k)\Lambda\mathbf{u}_2(k).$$

Thus

$$\begin{aligned}
r_2(\mathbf{u}_2(k), \mathbf{y}_2(k)) &= [(\mathbf{I} - \mathbf{A}\mathbf{W}^2)\mathbf{u}_2(k) - \mathbf{A}\mathbf{W}^1\mathbf{u}_2(k-1)]^T \\
&\times \mathbf{T}[(\mathbf{I} - \mathbf{B}\mathbf{W}^2)\mathbf{u}_2(k) - \mathbf{B}\mathbf{W}^1\mathbf{u}_2(k-1)] \\
&+ [-\mathbf{W}^1\mathbf{u}_2(k-1) - \mathbf{W}^2\mathbf{u}_2(k)]^T\Lambda\mathbf{u}_2(k) \\
&= \mathbf{u}_2^T(k)(\mathbf{I} - \mathbf{A}\mathbf{W}^2)^T\mathbf{T}(\mathbf{I} - \mathbf{B}\mathbf{W}^2)u_2(k) \\
&+ \mathbf{u}_2^T(k-1)(\mathbf{W}^1)^T\mathbf{B}\mathbf{T}\mathbf{A}\mathbf{W}^1\mathbf{u}_2(k-1) \\
&- \mathbf{u}_2^T(k)(\mathbf{I} - \mathbf{A}\mathbf{W}^2)^T\mathbf{T}(\mathbf{B}\mathbf{W}^1)\mathbf{u}_2(k-1) \\
&- \mathbf{u}_2^T(k-1)(\mathbf{A}\mathbf{W}^1)^T\mathbf{T}(\mathbf{I} - \mathbf{B}\mathbf{W}^2)\mathbf{u}_2(k) \\
&- \mathbf{u}_2^T(k-1)(\mathbf{W}^1)^T\Lambda\mathbf{u}_2(k) \\
&- \mathbf{u}_2^T(k)(\mathbf{W}^2)^T\Lambda\mathbf{u}_2(k).
\end{aligned}$$

Let

$$\mathbf{P}_2 = \begin{bmatrix} \mathbf{P11} & \mathbf{P12} \\ \mathbf{P21} & \mathbf{P22} \end{bmatrix} \tag{3.16}$$

where

$$\begin{aligned}
\mathbf{P11} =& \mathbf{T} - \frac{1}{2}(\mathbf{W}^2)^T(\mathbf{B}+\mathbf{A})\mathbf{T} - \frac{1}{2}\mathbf{T}(\mathbf{B}+\mathbf{A})\mathbf{W}^2 + (\mathbf{W}^2)^T\mathbf{BTAW}^2 \\
& - \mathbf{Q} - \frac{1}{2}((\mathbf{W}^2)^T\Lambda + \Lambda\mathbf{W}^2), \\
\mathbf{P12} =& -\frac{1}{2}\mathbf{T}(\mathbf{A}+\mathbf{B})\mathbf{W}^1 + (\mathbf{W}^2)^T\mathbf{BTAW}^1 - \frac{1}{2}\Lambda\mathbf{W}^1, \\
\mathbf{P21} =& \mathbf{P12}^T, \\
\mathbf{P22} =& \mathbf{Q} + (\mathbf{W}^1)^T\mathbf{BTAW}^1.
\end{aligned}$$

Theorem 3.4 *The equilibrium point of the LDDN is GAS if a positive definite matrix* $\boldsymbol{Q}$*, a positive definite diagonal matrix* $\boldsymbol{T}$ *and a positive semi-definite diagonal matrix* Λ *can be found such that the matrix* $\mathbf{P}_2$ *is positive definite.*

Proof. The proof is similar to the proof of Theorem 3.3, but the matrix $\mathbf{P}_2$ is used in place of the matrix $\mathbf{P}_1$. ■

Supply rate using general quadratic form

Choose $r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) = \mathbf{v}^T\mathbf{F}\mathbf{v}$ where $\mathbf{v} = [\mathbf{u}_1(k)\ \mathbf{y}_1(k)]^T$ and

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix}.$$

Assume that $\mathbf{F}$ satisfies the condition

$$r_1(\mathbf{u}_1(k), \mathbf{y}_1(k)) \geq 0. \tag{3.17}$$

Let's define

$$\mathbf{P}_3 = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}$$

where

$$
\begin{aligned}
\mathbf{P}_{11} =& -\mathbf{F}_{22} - 0.5 * (\mathbf{F}_{21} + \mathbf{F}_{12}^T)\mathbf{W}^2 - 0.5(\mathbf{W}^2)^T(\mathbf{F}_{12} + \mathbf{F}_{21}^T) - (\mathbf{W}^2)^T\mathbf{F}_{11}\mathbf{W}^2 - \mathbf{Q} \\
\mathbf{P}_{12} =& -(\mathbf{W}^2)^T\mathbf{F}_{11}\mathbf{W}^1 - 0.5 * (\mathbf{F}_{21} + \mathbf{F}_{12}^T)\mathbf{W}^1 \\
\mathbf{P}_{21} =& -(\mathbf{W}^1)^T\mathbf{F}_{11}\mathbf{W}^2 - 0.5 * (\mathbf{W}^1)^T(\mathbf{F}_{12} + \mathbf{F}_{21}) \\
\mathbf{P}_{22} =& \mathbf{Q} - (\mathbf{W}^1)^T\mathbf{F}_{11}\mathbf{W}^1.
\end{aligned}
$$

Theorem 3.5 *The equilibrium point of the LDDN is GAS if there exists a matrix $\boldsymbol{F}$ satisfying condition (3.17) and a positive definite matrix $\boldsymbol{Q}$ such that the matrix $\boldsymbol{P}_3$ is positive definite.*

Proof. The proof is similar to the proof of Theorem 3.3, but the matrix $\mathbf{P}_3$ is used in place of the matrix $\mathbf{P}_1$. ■

We can see that Theorems 3.3 and 3.4 are special cases of Theorem 3.5.

- If $\mathbf{F}_{11} = \mathbf{B}^2$, $\mathbf{F}_{22} = -\mathbf{I}$ and $\mathbf{F}_{12} = \mathbf{F}_{21} = \mathbf{0}$ then $\mathbf{P}_3 = \mathbf{P}_1$.
- If $\mathbf{F}_{11} = -\mathbf{BTA}$, $\mathbf{F}_{22} = -\mathbf{T}$, $\mathbf{F}_{12} = \mathbf{BT} + \Lambda$ and $\mathbf{F}_{21} = \mathbf{TA}$ then $\mathbf{P}_3 = \mathbf{P}_2$.

Since the output $\mathbf{y}_1(k)$ is a static function of $\mathbf{u}_1(k)$, the matrix $\mathbf{F}$ doesn't need to be positive definite. As we can see in Theorem 3.3, $\mathbf{F}_{11} = \mathbf{B}^2$ is a positive definite matrix, $\mathbf{F}_{22} = -\mathbf{I}$ is negative definite and $\mathbf{F}_{12} = \mathbf{F}_{21} = \mathbf{0}$. So in this case the matrix $\mathbf{F}$ is not positive definite. Based on this theorem we may find other stability criteria for the standard form.

Conclusions

In this section, we have found three new conditions for globally asymptotic stability of equilibrium points of LDDNs. The different conditions were derived by selecting different supply rate functions. In Theorem 3.3, we used the upper bound on the sector condition of the static subsystem. In Theorem 3.4, we used both the sector upper and lower bounds to define the supply rate. In Theorem 3.5, we used a general quadratic supply rate.

3.3.7 Stability analysis of other types of Recurrent Neural Networks

Now we will apply Theorems 3.3, 3.4 and 3.5 to analyze the stability analysis of other types of RNNs. First we consider the network given by Barabanov [4, p. 292]. This network can be transformed into standard form (3.7) as follows. Let

$$\mathbf{x}(k) = [\mathbf{x}_1(k)\ \mathbf{x}_2(k)\ ...\ \mathbf{x}_n(k)]^T,$$

$$\mathbf{b} = [\mathbf{b}_1\ \mathbf{b}_2\ ...\ \mathbf{b}_n]^T,$$

$$\mathbf{W}^1 = \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{V}_n \\ \mathbf{0} & \mathbf{W}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{W}_n \end{bmatrix},$$

and

$$\mathbf{W}^2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{V}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{n-1} & \mathbf{0} \end{bmatrix}.$$

The next step is to apply Theorems 3.3, 3.4 and 3.5 to check stability of the equilibrium point of this network. Some examples are shown in the next chapter.

Next, we consider a class of RNNs given in [1, p. 955], [25, p. 1105], [26, p. 1373] and [27, p. 130]. The network has the form

$$\mathbf{z}(k+1) = \mathbf{D}\mathbf{z}(k) + \mathbf{E}\mathbf{g}(\mathbf{W}\mathbf{z}(k) + \mathbf{s}_1) + \mathbf{s}_2 \tag{3.18}$$

where $\mathbf{D}$, $\mathbf{E}$ and $\mathbf{W}$ are matrices with appropriate sizes, $\mathbf{s}_1$ and $\mathbf{s}_2$ are bias vectors with appropriate size and $\mathbf{g}$ is a vector of activation functions in the network. We can transform this system into standard form (3.7) by defining new state variables as follows:

$$\mathbf{x}_1(k+1) = \mathbf{g}(\mathbf{W}\mathbf{z}(k) + \mathbf{s}_1)$$
$$\mathbf{x}_2(k) = \mathbf{z}(k)$$

Then

$$\mathbf{x}_1(k+1) = \mathbf{g}(\mathbf{W}\mathbf{x}_2(k) + \mathbf{s}_1)$$
$$\mathbf{x}_2(k+1) = \mathbf{D}x_2(k) + \mathbf{E}\mathbf{x}_1(k+1) + \mathbf{s}_2).$$

So we have

$$\mathbf{W}^1 = \begin{bmatrix} \mathbf{0} & \mathbf{W} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \mathbf{W}^2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{E} & \mathbf{0} \end{bmatrix}, \tag{3.19}$$

$\mathbf{f} = [\mathbf{g}\ \mathbf{id}]^T$ and $\mathbf{b} = [\mathbf{s}_1\ \mathbf{s}_2]^T$, where $\mathbf{0}$ is a zero matrix with appropriate size. Thus system (3.18) is in standard form (3.7).

The strategy for stability analysis of RNNs is to transform the network into the interconnected systems form and then to apply Theorem 3.2. If the network can be represented in standard form (3.7), then we can use Theorems 3.3, 3.4 and 3.5.

3.4 Conclusions

This chapter has developed new criteria for analyzing the stability of recurrent neural networks. A general framework for representing recurrent networks was presented, and we demonstrated how these networks could be put into a standard form and then into an interconnected systems form. This enabled us to use Theorem 3.2 to derive stability conditions. The key to the development of the stability criteria was to establish sector conditions on the static subsystem. By using the sector bounds, we were able to define different supply rate functions. It is the flexibility in selecting the supply rate function that makes the use of dissipativity theory so attractive for stability analysis of recurrent neural networks.

In the next chapter we will demonstrate the performance of our new stability criteria on a variety of recurrent networks. We will also compare our new criteria with other state-of-the-art methods.

CHAPTER 4

COMPARISON OF STABILITY CRITERIA ON TEST PROBLEMS

In this chapter, we will compare our new dissipativity-based (DB) criteria with Liu's criterion [6, p.1382] and Barabanov's LMI criterion [18, p. 4554] on 23 test problems. The test neural networks are in the form of (5) in [4, p. 294], (3.7) or (4.3). We have chosen some networks that have been introduced in previous papers by other authors. We have also used new networks that we have developed. We have chosen networks based on the difficulty of determining the stability of their equilibrium points.

4.1 Sector conditions

We will use networks with activation function $a = tanh(n)$. Therefore, as part of the stability analysis, we will need to find upper and lower bounds of the sector of the following function

$$f(u) = tanh(u + t) - tanh(t), \tag{4.1}$$

where t is a constant. In order to determine sector upper bounds, we will determine the argument that maximizes the value of function $\frac{f(u)}{u}$. The following lemma can be used to determine the lower bound of the function (4.1).

Lemma 4.1 *[4, p. 298] Consider a function (4.1). If* $|u + t| \leq r$*, then* $\frac{f(u)}{u} \geq \nu = \frac{tanh(r)-tanh(|t|)}{r-|t|}$*. If* $|t| = r$*, then* $\nu = \frac{d(tanh(u))}{du}$ *at* $u = r$*.*

There are several ways to find r, depending on the type of system. We will discuss this later.

Since we will be comparing our DB criteria with those of Liu and Barabanov, we will give a brief description of their criteria in the next two sections.

4.2 Existing stability criteria

4.2.1 Liu's criterion

Liu's model is called the Discrete-time Delayed Standard Neural Network Model (DDSNNMs) [6, p. 1378]:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}_l\mathbf{x}(k) + \mathbf{B}_p\Phi(\xi(k)) + \mathbf{B}_{pd}\Phi(\xi(k-h)) \\ \xi(k) = \mathbf{C}_q\mathbf{x}(k) + \mathbf{D}_p\Phi(\xi(k)) + \mathbf{D}_{pd}\Phi(\xi(k-h)) \end{cases} \tag{4.2}$$

Assume $\Phi_i(\xi_i) \in [q_i, u_i],\ u_i > q_i \geq 0$ and h is constant. Define $\mathbf{Q} = diag(q_i)$ and $\mathbf{U} = diag(u_i)$.

Theorem 4.1 *[6, p. 1382] The origin of the DDSNNM (4.2) is globally asymptotically stable, if there exist symmetric positive-definite matrices* $\boldsymbol{P}$ *and* Γ*, and diagonal positive semi-definite matrices* Λ *and* $\boldsymbol{T}$*, such that the following matrix is negative definite*

$$\bar{\boldsymbol{G}} = \begin{bmatrix} \bar{\boldsymbol{G}}_{11} & \bar{\boldsymbol{G}}_{12} & \bar{\boldsymbol{G}}_{13} \\ \bar{\boldsymbol{G}}_{21} & \bar{\boldsymbol{G}}_{22} & \bar{\boldsymbol{G}}_{23} \\ \bar{\boldsymbol{G}}_{31} & \bar{\boldsymbol{G}}_{32} & \bar{\boldsymbol{G}}_{33} \end{bmatrix}$$

where

$$
\begin{aligned}
\bar{\boldsymbol{G}}_{11} =& \boldsymbol{A}_l^T \boldsymbol{P} \boldsymbol{A}_l - \boldsymbol{P} - 2\boldsymbol{C}_q^T \boldsymbol{T}\boldsymbol{Q}\boldsymbol{U}\boldsymbol{C}_q \\
\bar{\boldsymbol{G}}_{12} =& \boldsymbol{A}_l^T \boldsymbol{P} \boldsymbol{B}_p + \boldsymbol{C}_q^T \Lambda - 2\boldsymbol{C}_q^T \boldsymbol{T}\boldsymbol{Q}\boldsymbol{U}\boldsymbol{D}_p + \boldsymbol{C}_q^T (\boldsymbol{Q} + \boldsymbol{U})\boldsymbol{T} \\
\bar{\boldsymbol{G}}_{13} =& \boldsymbol{A}_l^T \boldsymbol{P} \boldsymbol{B}_{pd} - 2\boldsymbol{C}_q^T \boldsymbol{T}\boldsymbol{Q}\boldsymbol{U}\boldsymbol{D}_{pd} \\
\bar{\boldsymbol{G}}_{21} =& \bar{\boldsymbol{G}}_{12}^T \\
\bar{\boldsymbol{G}}_{31} =& \bar{\boldsymbol{G}}_{13}^T \\
\bar{\boldsymbol{G}}_{32} =& \bar{\boldsymbol{G}}_{23}^T \\
\bar{\boldsymbol{G}}_{22} =& \boldsymbol{B}_p^T \boldsymbol{P} \boldsymbol{B}_p + \Lambda \boldsymbol{D}_p + \boldsymbol{D}_p^T \Lambda + \Gamma - 2\boldsymbol{D}_p^T \boldsymbol{T}\boldsymbol{Q}\boldsymbol{U}\boldsymbol{D}_p \\
& - 2\boldsymbol{T} + \boldsymbol{D}_p^T (\boldsymbol{Q} + \boldsymbol{U})\boldsymbol{T} + \boldsymbol{T}(\boldsymbol{Q} + \boldsymbol{U})\boldsymbol{D}_p \\
\bar{\boldsymbol{G}}_{23} =& \boldsymbol{B}_p^T \boldsymbol{P} \boldsymbol{B}_{pd} + \Lambda \boldsymbol{D}_{pd} - 2\boldsymbol{D}_p^T \boldsymbol{T}\boldsymbol{Q}\boldsymbol{U}\boldsymbol{D}_{pd} + \boldsymbol{T}(\boldsymbol{Q} + \boldsymbol{U})\boldsymbol{D}_{pd} \\
\bar{\boldsymbol{G}}_{33} =& \boldsymbol{B}_{pd}^T \boldsymbol{P} \boldsymbol{B}_{pd} - \Gamma - 2\boldsymbol{D}_{pd}^T \boldsymbol{T}\boldsymbol{Q}\boldsymbol{U}\boldsymbol{D}_{pd}
\end{aligned}
$$

4.2.2 Barabanov and Prokhorov's LMI criterion

Consider the RNN [18, p. 4553]

$$\mathbf{x}(k+1) = \mathbf{D}\mathbf{x}(k) + \mathbf{E}tanh(\mathbf{W}\mathbf{x}(k) + \mathbf{s}_1). \tag{4.3}$$

Let $\mathbf{z}$ be the equilibrium point of (4.3). Then

$$\mathbf{z} = \mathbf{D}\mathbf{z} + \mathbf{E}tanh(\mathbf{W}\mathbf{z} + \mathbf{s}_1)$$

Define $\mathbf{y} = \mathbf{x} - \mathbf{z}$ and $\mathbf{c} = \mathbf{W}\mathbf{z} + \mathbf{s}_1$. Then

$$\begin{cases} \mathbf{y}(k+1) = \mathbf{D}\mathbf{y}(k) + \mathbf{E}\eta(k) \\ \eta(k) = tanh(\sigma(k) + \mathbf{c}) - tanh(\mathbf{c}) \end{cases} \tag{4.4}$$

where $\sigma(k) = \mathbf{W}\mathbf{y}(k)$. Assume $\eta_i(\sigma_i) \in [\nu_i, \mu_i]$ for $i = 1, 2, ..., m$ where m is the length of $\mathbf{s}_1$. Let $\mathbf{N} = diag(\mu_i)$ and $\mathbf{M} = diag(\nu_i)$.

Lemma 4.2 *[18, p. 4554] If* $|c_i| \geq |c_j|$ *then*

$$\beta_{ij} = \frac{1 - tanh(\alpha sign(c_i)c_j)}{1 - tanh(\alpha|c_i|)}$$

otherwise

$$\beta_{ij} = \frac{1 - tanh(\alpha sign(c_j)c_i)}{1 - tanh(\alpha|c_j|)}$$

where $\alpha > 0$*.*

By Lemma 4.2, $\eta^T \mathbf{G}(\alpha \mathbf{W}\mathbf{y} - \eta) \geq 0$, where $\mathbf{G} = \{g_{ij}\}$ is a symmetric positive definite matrix that satisfies $g_{ij} < 0$ for $i \neq j$,

$$g_{ii} + \sum_{j=1, j\neq i}^{m} \beta_{ij} g_{ij} > 0 \tag{4.5}$$

for all $i = 1, ..., m$.

Theorem 4.2 *[18, p. 4554] Consider the system (4.3). Assume* $\boldsymbol{D} + \boldsymbol{EMW}$ *is stable. If there exists diagonal positive definite matrix* Γ *and symmetric positive definite matrices* $\boldsymbol{H}$ *and* $\boldsymbol{G}$*, such that* $\boldsymbol{G}$ *satisfies the condition (4.5) and*

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}$$

is negative definite, then the equilibrium point is GAS, where

$$\begin{aligned}
\Phi_{11} =& \boldsymbol{D}^T \boldsymbol{HD} - \boldsymbol{H} - \boldsymbol{W}^T \boldsymbol{N}\Gamma \boldsymbol{MW} \\
\Phi_{12} =& \boldsymbol{W}^T((\boldsymbol{M} + \boldsymbol{N})\Gamma + \boldsymbol{G}\alpha)/2 + \boldsymbol{D}^T \boldsymbol{HD} \\
\Phi_{21} =& \Phi_{12}^T \\
\Phi_{22} =& \boldsymbol{E}^T \boldsymbol{HE} - \Gamma - \boldsymbol{G}
\end{aligned}$$

To compute the lower bound matrix $\mathbf{M}$, we will use Lemma 4.1. Thus we have to find r_j such that $|\sigma_j + c_j| \leq r_j$ for $j = 1, 2, ..., m$. In the general case for $\mathbf{D} \neq \mathbf{0}$ we can choose $\mathbf{M} = \mathbf{0}$. If $\mathbf{D} = \mathbf{0}$, then $|y_i| \leq \sum_{j=1}^{m} |E_{ij}|$ for $i = 1, 2, ..., n$. Let's define $\gamma_i = \sum_{j=1}^{m} |E_{ij}|$ and $r_j = \sum_{i=1}^{n} |W_{ji}||\gamma_i|$. Therefore $|\sigma_j + c_j| \leq r_j$.

4.3 Description of test problems

We will compare our new DB stability criteria with those of Liu and Barabanov on 22 different test networks. All of the networks have a GAS equilibrium point. For the first 12 networks, the DB-based criteria are able to prove stability. For the next 10 networks, DB criteria are not able to detect stability. The final test network (the 23rd test) is an LDDN network that cannot be put into Liu's or Barabanov's form, so their methods cannot be applied to this network.

Even in those cases where the methods are not able to prove stability, we would like to measure how close they come. Each of the criteria involves finding matrices that are definite. For the three DB criteria, we attempt to find matrices $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$ that are positive definite. (For the test problems, we will use only $\mathbf{P}_2$, from Theorem 3.4, which has produced the best results.) For Liu's and Barabanov's methods, we are looking for $\bar{\mathbf{G}}$ and Φ matrices that are negative definite. To measure how close we come to finding matrix $\mathbf{P}_2$ that is positive definite, we will find its minimum eigenvalue, which we will label p_2. If p_2 is positive, then GAS is proved. If it is negative, then we cannot prove stability. However, even in this case, the closer the value is to zero, the closer the algorithm has come to identifying stability. This concept applies also to the maximum eigenvalues of $\bar{\mathbf{G}}$ and Φ, which we will label $\bar{g}$ and φ. If these values are negative, then GAS is proved. If they are positive, we have not proven stability. However, the closer $\bar{g}$ and φ are to zero, the closer the algorithm has come to identifying stability.

All of the 23 test problems are described in the appendix. They are represented in either the standard form (3.7), in which case $\mathbf{W}^1$, $\mathbf{W}^2$, $\mathbf{b}$ and $\mathbf{f}$ will be provided, or in the Barabanov form (4.3), in which case $\mathbf{W}$, $\mathbf{E}$ and $\mathbf{s}_1$ will be given ($\mathbf{D} = \mathbf{0}$ for the Barabanov test problems). On the first 12 test problems, all matrices for the DB criterion, Liu and Barabanov's criteria are also provided in the appendix.

In the next section we analyze stability of neural networks with stable equilibrium points where our DB methods can prove stability. In the following section we focus on neural networks with stable equilibrium points where these methods cannot prove stability. Finally, we will analyze the stability of an LDDN, where Liu's and Barabanov's methods cannot be applied.

4.4 Stable equilibria that can be proved stable

In this section, we analyze the stability of equilibrium points for test problems 1 to 12. The first four test problems were first presented in previously published papers. Test problem 1 is a special case of standard form (3.7) where $\mathbf{W}^2 = \mathbf{0}$, test problem 2 has the form of (4.3) and test problems 3 and 4 are have the form of (5) in [4, p. 294]. The remaining 8 test problems are in the form of (4.3) with $\mathbf{D} = \mathbf{0}$.

We will use the new DB criteria, Liu's criterion and Barabanov's criterion to check stability of equilibria. Eigenvalues p_2, $\bar{g}$, and φ for each test problem are shown in Table 4.1. The equilibrium point is proven to be GAS if $p_2 > 0$, $\bar{g} < 0$ or $\varphi < 0$. Thus the criteria $\mathbf{P}_2$, $\bar{\mathbf{G}}$ and Φ all proved that the equilibrium points of test problems 1 through 12 are GAS.

From this table we can see that our DB criterion is as tight as both Liu's criterion [6, p. 1382] and Barabanov's LMI criterion [18, 4554] on the first 12 test problems. Fig. 4.1 and Fig. 4.2 are representative of the types of responses that we have in these problems. We have oscillatory responses as well as over-damped responses. Table 4.2 gives an approximate measure of how quickly the systems converge to the equilibrium point from a random initial condition. Because the systems are nonlinear, these convergence times will change with the initial conditions, but these numbers are representative. In the next section we will investigate systems with GAS equilibria for which the DB stability criteria cannot determine stability.

TP.	p_2	$\bar{g}$	φ
1	0.34743	-0.57805	-0.49272
2	14.625	-147.31	-128.83
3	7.1617	-4.7596	-1.5471
4	34.342	-45.769	-75.599
5	5.9322	-61.345	-42.564
6	1.8062	-7.5758	-8.5027
7	1.1116	-34.491	-22.097
8	29.101	-136.46	-212.8607
9	59.747	-242.07	-393.64
10	19.376	-119.87	-156.07
11	6.8088	-30.292	-16.997
12	0.2592	-12.421	-8.8994

Table 4.1: Eigenvalues of matrices

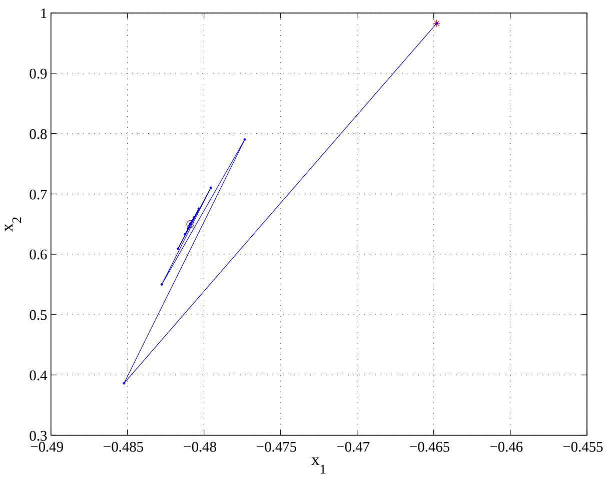

Figure 4.1: Trajectory for test problem 1

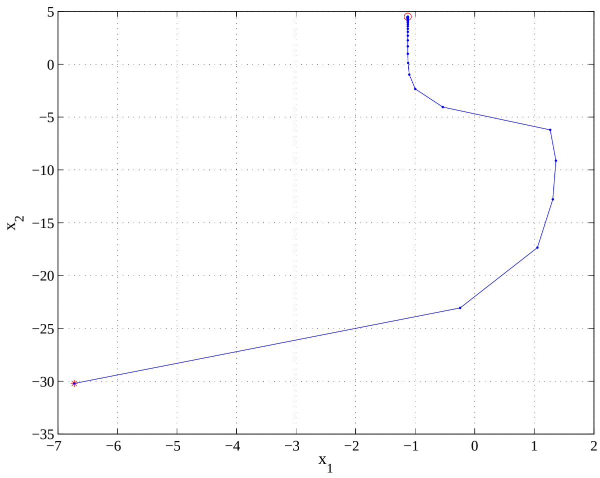

Figure 4.2: Trajectory for test problem 2

TP.	N
01	15
02	14
03	28
04	35
05	06
06	07
07	05
08	10
09	07
10	07
11	03
12	08

Table 4.2: Time steps N to convergence

4.5 Stable equilibria that cannot be proved stable

In this section, we will use our DB criterion, Liu's criterion and Barabanov's LMI criterion to check stability of equilibrium points of test problems 13 to 22. All of these test neural networks are in the form of (4.3) with $\mathbf{D} = \mathbf{0}$, except test problem 13, which is a special case of (3.7) where $\mathbf{W}^2 = \mathbf{0}$. The equilibrium points here are all GAS. The best values of p_2, $\bar{g}$ and φ are shown in Table 4.3. Barabanov's LMI criterion was the only one that was

TP.	p_2	$\bar{g}$	φ
13	$-4.35 * 10^{-8}$	$1.16 * 10^{-7}$	4.0163
14	$-4.16 * 10^{-4}$	$6.57 * 10^{-8}$	-2.8767
15	$-2.52 * 10^{-4}$	$7.67 * 10^{-5}$	-0.24913
16	$-5.63 * 10^{-4}$	$5.63 * 10^{-8}$	$1.4 * 10^{-4}$
17	$-3.18 * 10^{-5}$	$3.72 * 10^{-8}$	$4.47 * 10^{-5}$
18	-0.0033	$4.67 * 10^{-7}$	2.7698
19	-0.0123	$3.03 * 10^{-7}$	2.53
20	$-1.80 * 10^{-4}$	$2.29 * 10^{-6}$	$7.98 * 10^{-5}$
21	$-2.45 * 10^{-4}$	$4.09 * 10^{-7}$	$3.64 * 10^{-5}$
22	$-1.84 * 10^{-5}$	$1.71 * 10^{-8}$	$8.57 * 10^{-6}$

Table 4.3: Eigenvalues of matrices

able to prove stability of equilibria in any of the test problems, and it only worked for test problems 14 and 15. In these two networks, the matrix $\mathbf{W}$ is an identity matrix and the bias vector $\mathbf{s}_1 = \mathbf{0}$.

Fig. 4.3 and Fig. 4.4 are representative of the responses of test problems 13 through 22. All of these responses are oscillatory. Table 4.4 shows approximate convergence times from random initial conditions. We can see that, although all of the responses are oscillatory, there is a wide range of response times - some of them in the same range as the first 12 systems (see Table 4.2). It seems clear that when the system is of higher dimension, has oscillatory behavior and has slow convergence, it becomes more difficult for all methods to determine stability. However, these parameters do not completely determine the success of the various methods.

In terms of the eigenvalues, $|\bar{g}|$ is smallest for all test problems, except 13, 14 and 15. For test problems 18 and 19, the oscillation is longer (approximately 600 time steps) and the size of matrices **E** and **W** is bigger (2×10). In these cases, $|\bar{g}| < |p_2| < |\varphi|$.

The weight matrices and bias vectors of test problems 20, 21 and 22 are the same size, but N is 40, 10 and 5, respectively. In this case, $|\bar{g}|$, $|p_2|$, and $|\varphi|$ became smaller, as shown in Table 4.3.

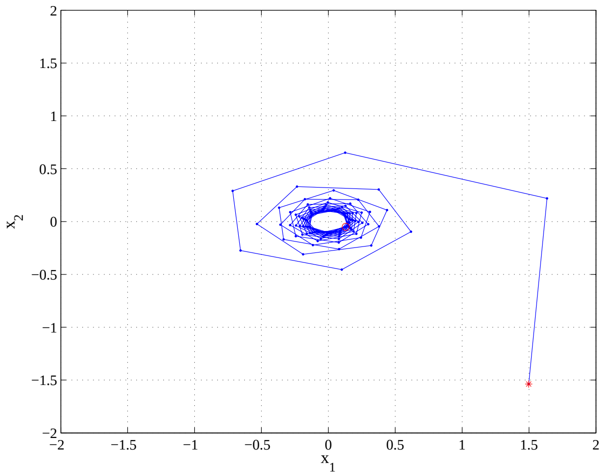

Figure 4.3: Trajectory for test problem 15

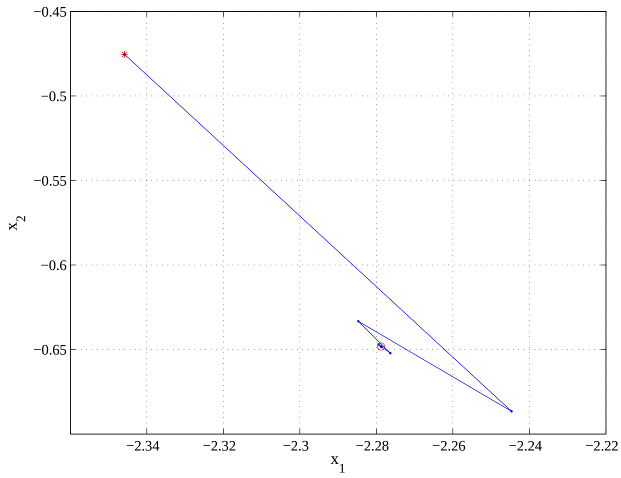

Figure 4.4: Trajectory for test problem 22

TP.	N
13	30
14	300
15	3500
16	250
17	90
18	20
19	600
20	40
21	10
22	5

Table 4.4: Time steps N to convergence

In summary, for test problems in the form of (4.3), the oscillation of system trajectories, increases in the sizes of system matrices, and slower convergence times tend to increase the difficulty of determining stability. Barabanov's LMI criterion is less conservative than our criterion and Liu's criterion on test problems 14 and 15 where the neural network is a very special case of (4.3).

4.6 An example of stability analysis of LDDNs

In this section, we analyze stability of test problem 23, which is an LDDN. We found $p_2 = 2.6051$. Therefore, this proves that the equilibrium point is GAS.

Since the function f_{10} is linear, $\alpha_{10} = \beta_{10} = 1$. This doesn't satisfy Liu's condition $u_i > q_i \geq 0$. So we cannot use Liu's criterion. Since we cannot represent the LDDN in the form of (4.3), Barabanov's LMI criterion cannot be applied to analyze stability of this network. Therefore, neither Liu's criterion nor Barabanov's LMI criterion can be applied to check stability of the equilibrium point.

4.7 Conclusions

The second DB criterion, Theorem 3.4, is as tight as both Barabanov's LMI criterion and Liu's criterion for the first 12 test problems, but fails to prove stability of equilibria for test problems 13 to 22. Liu's criterion also fails to prove stability of these GAS equilibrium points. Barabanov's LMI criterion can prove stability of the equilibrium points for test problems 14 and 15 where $\mathbf{W} = \mathbf{I}$ and $\mathbf{s}_1 = \mathbf{0}$. In general, when a system has more oscillatory responses, larger system matrices and slower convergence, all of the methods described in this paper have more difficulty in determining stability.

To analyze the stability of LDDNs using either Liu's criterion or Barabanov's LMI criterion, we need to have state transformations which can convert the standard form (3.7) into their corresponding models: (4.2) and (4.3). This is not always possible, as in test problem 23. There do exist LDDNs that cannot be analyzed with either Liu's or Barabanov's methods. The DB methods developed in this paper can be applied to any LDDN network. Our DB criteria can be applied to analyze the stability of equilibrium points for neural networks of forms (1) in [4, p. 292], (3.7), and (4.3).

Another advantage of the DB criteria when compared with Barabanov's method for neural networks in the form of (1) in [4, p. 292] is that the dimensionality of the matrices involved in the DB method are generally smaller. This is because of the use of the state space extension method in [4], which requires that the number of states be increased substantially in these cases.

The dissipativity approach has not been used before for the stability analysis of recurrent neural networks. The results shown in this chapter demonstrate the promise of this approach. In addition, with the dissipativity method there is the potential for additional stability criteria to be developed. This is because of the flexibility in choosing the supply rate function.

CHAPTER 5

TRAINING RECURRENT NETWORKS FOR STABILITY

In this chapter, we will apply the novel stability analysis methods presented in the previous chapter to the problem of training recurrent neural networks while maintaining stability. It has been shown [17] that the error surfaces of recurrent neural networks can have spurious valleys that can cause training difficulties. These valleys are caused by network instabilities. If we can maintain network stability during training, we could avoid the spurious valleys.

In this chapter, we describe a new training method for maintaining network stability. The first step is to define a new performance index, which combines mean square error with the maximum eigenvalue of the matrix $-\mathbf{P}_2$ from (3.16). If this maximum eigenvalue is less than zero, the network is guaranteed to be stable. By minimizing this maximum eigenvalue, we have the best chance of maintaining stability and avoiding the spurious valleys.

The next section describes the modified performance index, and the following section describes how the gradient of this performance index can be computed. The gradient is needed for the training algorithm, which finds the network weights and biases that minimize the performance index.

5.1 Modified performance index

Let's consider an LDDN network, as in (3.5) and (3.6). If we represent the network in standard form, we have the system (3.7). Assume that a set of training data is provided

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, ..., \{\mathbf{p}_q, \mathbf{t}_q\}, \tag{5.1}$$

where $\mathbf{p}_i$ is an input to the network and the corresponding target response is $\mathbf{t}_i$. Now we define a modified performance index as follows

$$J = \frac{1}{q}\sum_{i=1}^{q}(\mathbf{t}_i - \mathbf{a}^M(i))^T(\mathbf{t}_i - \mathbf{a}^M(i)) + \sigma\lambda \tag{5.2}$$

where $\mathbf{a}^M$ is the network output, σ is a constant and λ is the maximum eigenvalue of the matrix $-\mathbf{P}_2$ (3.16). Next we will compute the gradient of the performance index J with respect to weights and biases of the network.

5.2 Gradient computation

In this work, we use the scaled conjugate gradient algorithm [28] to update weights and biases. Thus, the command 'trainscg' in the Neural Network Toolbox of Matlab [29] will be modified to train LDDNs with the modified performance index (5.2).

Let

$$mse = \frac{1}{q}\sum_{i=1}^{i=q}(\mathbf{t}_i - \mathbf{a}^M(i))^T(\mathbf{t}_i - \mathbf{a}^M(i)) \tag{5.3}$$

Then

$$\frac{\partial J}{\partial \mathbf{x}} = \frac{\partial (mse)}{\partial \mathbf{x}} + \sigma\frac{\partial \lambda}{\partial \mathbf{x}} \tag{5.4}$$

where $\mathbf{x}$ is a vector of weights and biases of the network.

To calculate the first derivative of J with respect to weights and biases, we separately compute the first derivative of the mean square error (mse) and the first derivative of λ. We can use the standard backpropagation algorithm [9] to compute $\frac{\partial (mse)}{\partial \mathbf{x}}$. However, to compute the derivative of λ with respect to $\mathbf{x}$, we need a novel development. In the next section, we will demonstrate how to find the derivative of an eigenvalue with respect to an element of the matrix. Then, we show how to find the derivative of the eigenvalue of $-\mathbf{P}_2$ with respect to network weights.

5.2.1 The derivative of eigenvalue

In this section, a method for computation of eigenvalue derivatives is reviewed [30]. Let $\mathbf{K}(p) \in C^{n\times n}$ be an non-defective matrix [30], where p is a scalar variable and C is the set of complex numbers. Let $\Lambda(p) \in C^{n\times n}$ be the eigenvalue matrix of $\mathbf{K}(p)$ and $\mathbf{X}(p) \in C^{n\times n}$ be a corresponding eigenvector matrix of $\mathbf{K}(p)$. Then

$$\mathbf{K}(p)\mathbf{X}(p) = \mathbf{X}(p)\Lambda(p) \tag{5.5}$$

Assume $\mathbf{K}(p)$, $\Lambda(p)$ and $\mathbf{X}(p)$ are differentiable in a neighborhood of $p = p_0$. Taking the first derivative both sides of the equation (5.5), we obtain

$$\mathbf{K}^{'}(p)\mathbf{X}(p) + \mathbf{K}(p)\mathbf{X}^{'}(p) = \mathbf{X}^{'}(p)\Lambda(p) + \mathbf{X}(p)\Lambda^{'}(p) \tag{5.6}$$

Left multiplying both sides of (5.6) by $\mathbf{X}^{-1}(p)$ results in

$$\mathbf{X}^{-1}(p)\mathbf{K}^{'}(p)\mathbf{X}(p) + \mathbf{X}^{-1}(p)\mathbf{K}(p)\mathbf{X}^{'}(p) = \mathbf{X}^{-1}(p)\mathbf{X}^{'}(p)\Lambda(p) + \mathbf{X}^{-1}(p)\mathbf{X}(p)\Lambda^{'}(p) \tag{5.7}$$

Since **K** is non-defective, the eigenvectors are independent. Thus, there exists a matrix **C** such that

$$\mathbf{X}^{'}(p) = \mathbf{X}(p)\mathbf{C} \tag{5.8}$$

Plugging (5.8) into (5.7), we get

$$\mathbf{X}^{-1}(p)\mathbf{K}^{'}(p)\mathbf{X}(p) - \Lambda^{'}(p) = -\Lambda(p)\mathbf{C} + \mathbf{C}\Lambda(p) \tag{5.9}$$

Let $\Lambda^{'}(p) = diag(\lambda_k)$ for $k = 1, 2, ...n$, $\mathbf{X}^{-1}(p) = [\mathbf{y}_1 \ \ \mathbf{y}_2 \ \ ... \ \ \mathbf{y}_n]^T$ and $\mathbf{X}(p) = [\mathbf{x}_1 \ \mathbf{x}_2 \ ... \ \mathbf{x}_n]$. Then, for k = 1, 2,..., n

$$\lambda_k^{'} = \mathbf{y}_k^T\mathbf{K}^{'}(p)\mathbf{x}_k. \tag{5.10}$$

Now we want to take the first derivative of the maximum eigenvalue of the matrix $\mathbf{K}(p)$ with respect to p. Let λ_m be the maximum eigenvalue of $\mathbf{K}(p)$. Then $\lambda_m = max(\lambda_i)$ for $i = 1,\ 2,\ ...\ n$. Thus

$$\lambda_m^{'} = \mathbf{y}_m^T\mathbf{K}^{'}(p)\mathbf{x}_m. \tag{5.11}$$

where vectors $\mathbf{y}_m$ and $\mathbf{x}_m$ are associated with the eigenvalue λ_m. Therefore,

$$\lambda_m'|_{p=p_0} = \mathbf{y}_m^T \mathbf{K}'(p)|_{p=p_0} \mathbf{x}_m. \tag{5.12}$$

We will use this method to compute the first derivative of the maximum eigenvalue λ with respect to weights in the next section.

5.2.2 The derivative of maximum eigenvalue

In this section, we compute the first derivative of the maximum eigenvalue of the matrix $-\mathbf{P}_2$ with respect to weights. From (3.16), we define

$$\Xi = -\mathbf{P}_2 = \begin{bmatrix} \Xi_{11} & \Xi_{12} \\ \Xi_{21} & \Xi_{22} \end{bmatrix} \tag{5.13}$$

where

$$\begin{aligned}
\Xi_{11} =& -\mathbf{T} + \frac{1}{2}(\mathbf{W}^2)^T(\mathbf{B}+\mathbf{A})\mathbf{T} + \frac{1}{2}\mathbf{T}(\mathbf{B}+\mathbf{A})\mathbf{W}^2 - (\mathbf{W}^2)^T\mathbf{BTAW}^2 \\
&+ \mathbf{Q} + \frac{1}{2}((\mathbf{W}^2)^T\Lambda + \Lambda\mathbf{W}^2), \\
\Xi_{12} =& \frac{1}{2}\mathbf{T}(\mathbf{A}+\mathbf{B})\mathbf{W}^1 - (\mathbf{W}^2)^T\mathbf{BTAW}^1 + \frac{1}{2}\Lambda\mathbf{W}^1, \\
\Xi_{21} =& \Xi_{12}^T, \\
\Xi_{22} =& -\mathbf{Q} - (\mathbf{W}^1)^T\mathbf{BTAW}^1,
\end{aligned}$$

$\mathbf{W}_{i,j}^1 = [w_{i,j}^1]_{S\times S}$ and $\mathbf{W}_{i,j}^2 = [w_{i,j}^2]_{S\times S}$. Let λ_m be the maximum eigenvalue of Ξ. In this case, Ξ is a real, symmetric matrix. So λ_m is a real number, depending on the weights.

For convenience, let's define $\mathbf{Z}_{k,l} = [z_{i,j}]_{S\times S}$, where

$$z_{i,j} = \begin{cases} 1, \ if\ i = k\ and\ j = l \\ 0, \ if\ others \end{cases} \tag{5.14}$$

For example, if $i = 2,\ j = 3$ and $S = 3$ then

$$\mathbf{Z}_{2,3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

First, we need to compute the first derivative of Ξ with respect to a weight w, where w is either $w^1_{i,j}$ or $w^2_{i,j}$. Keep in mind that $w^2_{i,j} = 0$ when $i \leq j$, so we only consider the case $w^2_{i,j}$ with $i < j$. We have

$$\frac{\partial \Xi}{\partial w} = \begin{bmatrix} \frac{\partial \Xi_{11}}{\partial w} & \frac{\partial \Xi_{12}}{\partial w} \\ \frac{\partial \Xi_{21}}{\partial w} & \frac{\partial \Xi_{22}}{\partial w} \end{bmatrix} \tag{5.15}$$

For $w = w^1_{i,j}$,

$$\begin{aligned}
\frac{\partial \Xi_{11}}{\partial w^1_{i,j}} &= \mathbf{0} \\
\frac{\partial \Xi_{12}}{\partial w^1_{i,j}} &= \frac{1}{2}\mathbf{T}(\mathbf{A}+\mathbf{B})\mathbf{Z}_{i,j} - (\mathbf{W}^2)^T\mathbf{BTAZ}_{i,j} + \frac{1}{2}\Lambda\mathbf{Z}_{i,j} \\
\frac{\partial \Xi_{21}}{\partial w^1_{i,j}} &= (\frac{\partial \Xi_{12}}{\partial w^1_{i,j}})^T \\
\frac{\partial \Xi_{22}}{\partial w^1_{i,j}} &= -(\mathbf{Z}_{i,j})^T\mathbf{BTAW}^1 - (\mathbf{W}^1)^T\mathbf{BTAZ}_{i,j}
\end{aligned}$$

For $w = w^2_{i,j}$,

$$\begin{aligned}
\frac{\partial \Xi_{11}}{\partial w^2_{i,j}} &= \frac{1}{2}(\mathbf{Z}_{i,j})^T(\mathbf{B}+\mathbf{A})\mathbf{T} + \frac{1}{2}\mathbf{T}(\mathbf{B}+\mathbf{A})\mathbf{Z}_{i,j} - (\mathbf{Z}_{i,j})^T\mathbf{BTAW}^2 \\
&\quad - (\mathbf{W}^2)^T\mathbf{BTAZ}_{i,j} + \frac{1}{2}((\mathbf{Z}_{i,j})^T\Lambda + \Lambda\mathbf{Z}_{i,j}), \\
\frac{\partial \Xi_{12}}{\partial w^2_{i,j}} &= -(\mathbf{Z}_{i,j})^T\mathbf{BTAW}^1 \\
\frac{\partial \Xi_{21}}{\partial w^2_{i,j}} &= (\frac{\partial \Xi_{12}}{\partial w^1_{i,j}})^T \\
\frac{\partial \Xi_{22}}{\partial w^2_{i,j}} &= \mathbf{0}
\end{aligned}$$

Then, let's assume that we want to take the first derivative of λ_m at a certain point $\mathbf{W}^1 = \mathbf{W}^1_0$ and $\mathbf{W}^2 = \mathbf{W}^2_0$. Let $\mathbf{K} = \Xi(\mathbf{W}^1_0, \mathbf{W}^2_0)$. Let $\Lambda = diag(\lambda_k)$ for $k = 1, 2, ...n$ be the eigenvalues and $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_n]$ be the corresponding eigenvectors of the matrix $\mathbf{K}$, where $n = 2 * S$. Assume $\mathbf{X}^{-1} = [\mathbf{y}_1\ \mathbf{y}_2\ ...\ \mathbf{y}_n]^T$. Let λ_m be the maximum eigenvalue, with associated $\mathbf{x}_m$ and $\mathbf{y}_m$. From (5.12) and (5.15), we get

$$\frac{\partial \lambda_m}{\partial w^1_{i,j}} = \mathbf{y}^T_m \frac{\partial \Xi}{\partial w^1_{i,j}} \mathbf{x}_m \tag{5.16}$$

$$\frac{\partial \lambda_m}{\partial w_{i,j}^2} = \mathbf{y}_m^T \frac{\partial \Xi}{\partial w_{i,j}^2} \mathbf{x}_m \tag{5.17}$$

Finally, we can calculate the first derivative of the maximum eigenvalue with respect to any weight of the LDDN. Steps for the calculation of the derivative of the maximum eigenvalue are as follows.

- Given $\mathbf{W}^1 = \mathbf{W}_0^1$ and $\mathbf{W}^2 = \mathbf{W}_0^2$. Compute $\mathbf{K} = \Xi(\mathbf{W}_0^1, \mathbf{W}_0^2)$.
- Find eigenvalues and eigenvectors of the matrix $\mathbf{K}$: $\Lambda = diag(\lambda_k)$ and $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_n]$.
- Find $\mathbf{X}^{-1} = [\mathbf{y}_1\ \mathbf{y}_2\ ...\ \mathbf{y}_n]^T$.
- Find $\lambda_m = max(\lambda_k)$.
- Compute $\frac{\partial \lambda_m}{\partial w_{i,j}^1}|_{\mathbf{W}_0^1, \mathbf{W}_0^2} = \mathbf{y}_m^T \frac{\partial \Xi}{\partial w_{i,j}^1}|_{\mathbf{W}_0^1, \mathbf{W}_0^2} \mathbf{x}_m$.
- Compute $\frac{\partial \lambda_m}{\partial w_{i,j}^2}|_{\mathbf{W}_0^1, \mathbf{W}_0^2} = \mathbf{y}_m^T \frac{\partial \Xi}{\partial w_{i,j}^2}|_{\mathbf{W}_0^1, \mathbf{W}_0^2} \mathbf{x}_m$ if $i < j$

We will modify the command 'trainscg' in the Neural Network Toolbox to train the LDDN with the modified performance index. The regular trainscg already computes the derivative of mse with respect to the weights and biases. Now we will use the steps above to compute the derivative of the maximum eigenvalue with respect to weights. We will add this part to the regular trainscg with a penalty parameter σ. The modified trainscg will be used to train controller networks in next chapter.

CHAPTER 6

SIMULATIONS AND TEST RESULTS FOR STABLE TRAINING

In the previous chapter, we proposed a modified recurrent neural network training algorithm for maintaining network stability. In this chapter, two examples will be used to demonstrate the method. These examples are both model reference control (MRC) systems. The controllers of these systems are LDDNs. We will use the modified algorithm to train the controller networks while maintaining system stability. In the next section, a brief introduction to MRC systems is given. In the following section, a controller network for a linear MRC system will be trained. In the final section, we will train a controller network for a nonlinear MRC system.

6.1 Model reference control (MRC) using recurrent neural networks

In this section, we provide a brief description of neural network-based MRC (NN-based MRC) systems. An MRC system has the general structure shown in Fig. 6.1. In this figure, the plant model is chosen such that the plant model output is as close to the plant output as possible. The reference model represents the desired response of the closed-loop system. The controller is designed so that the plant output closely tracks the output of the reference model.

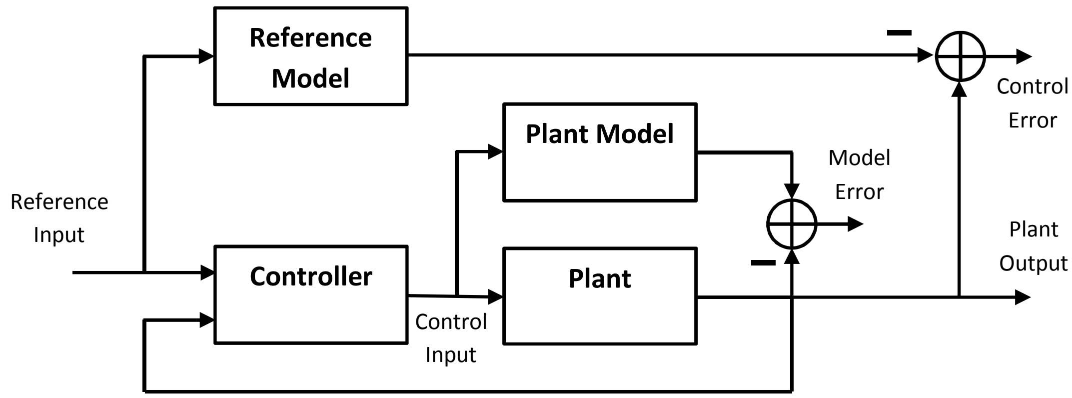

Figure 6.1: Model reference control system

Based on this idea, NN-based MRC systems were introduced in [31]. The NN-based MRC structure is shown in Fig. 6.2, where the plant model and the controller are neural networks. In order to design the NN-based MRC controller, first we have to train the NN plant model using the data observed from the input and the output of the plant. Then we have to choose a reference model whose response represents the desired behaviour of the plant. We will collect a training data set from the reference model. Then, we train the NN controller so that the control error is small enough while the MRC system remains stable. In the next section, we will show how to do the plant training.

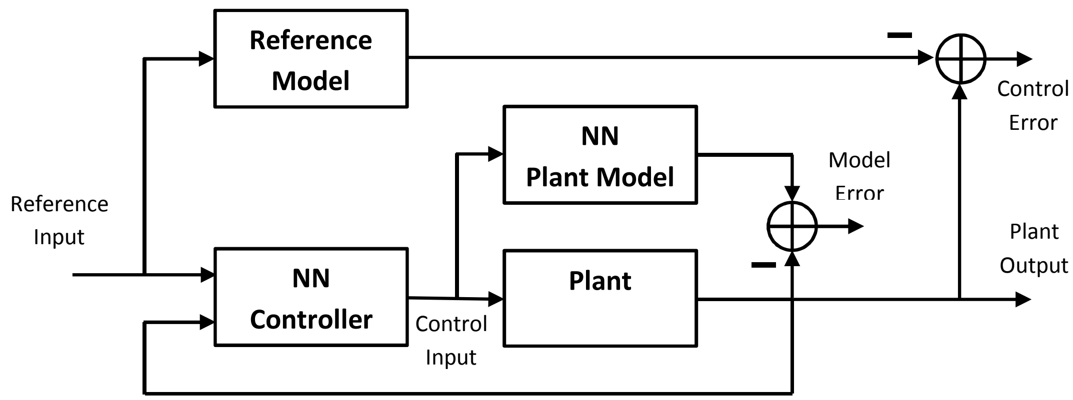

Figure 6.2: NN-based model reference control system

6.1.1 Plant training

The plant training process includes two stages. The first stage is to train the open-loop network (one-step-ahead training) and the final stage is to train the closed loop network.

First, we create a training data set. An input signal P, a series of step functions with a random magnitude and random width, will be generated. This input signal is applied to the plant. At the same time, we sample the plant output T. The sequences P and T will be used as data for plant training.

The NN plant model is shown in Fig. 6.3. This network consists of two layers with $tanh$ transfer function in the first layer and linear transfer function in the second layer. The number of neurons in the output layer depends on the number of outputs of the plant. In our examples, the plant is single input and single output. So there is one neuron in the second layer and one input to the network.

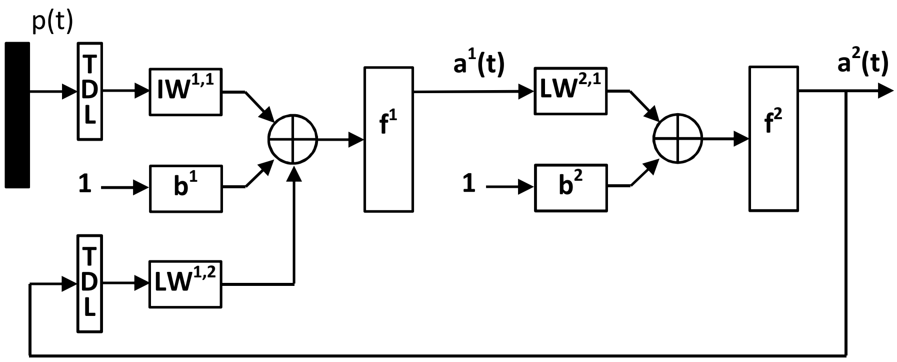

Figure 6.3: Neural network plant model

Next, we perform one-step-ahead training for the NN plant. To do this, we cut the feedback loop and use the network output as the second input to the network. The open loop network is shown in Fig. 6.4. The sequence P is applied to the first input and the sequence T is applied to the second input. The corresponding network output $\mathbf{a}^2$ will be compared with the target T. This model error will be used to update weights and biases.

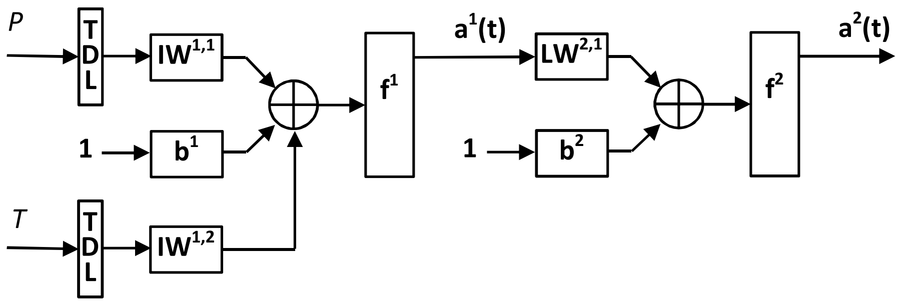

Figure 6.4: Network architecture for one-step-ahead plant training

After training the open loop network, we close the network by connecting the network output to the second input. The closed loop network is the original network shown in Fig. 6.3. We use the trained weights and biases from the one step-ahead training as initial weights and biases for the two-step-ahead training.

Now we need to prepare the data for training. Assume that *d* is the maximum number of delays in the network. The sequences P and T will be divided into subsequences with a length of $d + 2$. Each subsequence of P will be applied to the input of the closed loop network, and the corresponding network output will be compared to the corresponding T subsequence. The model error will be used to update the weights and biases.

We will do the same thing for k-step-head training with $k \geq 3$, but each preceeding subsequence has $d + k$ data points, and the initial weights and biases are taken from the preceeding $k - 1$ step-ahead training. This training process will be ended when the subsequence has the same length as the original P sequence. The weights and biases from the final training will be used to do the controller training in the next section.

6.1.2 Controller training

In this section, we will show how to train the controller network of NN-based MRC systems. First, an NN controller is created. The controller network and the plant network will be combined as in Fig. 6.5. In this figure, the NN plant includes layers 3 and 4. Its weights and biases are taken from the plant training, and they are not adjusted during controller training. Layers 1 and 2 make up the NN controller. In our examples, the controller network has two inputs: the first input is the reference input and the other input is the NN plant output.

Since we will use the modified performance index in (5.2) to train the controller network, which involves the matrix $\mathbf{P}_2$ of (3.16), we have to find matrices $\mathbf{Q}$, $\mathbf{T}$ and Λ. Initial weights and biases of the controller network will be chosen as small random numbers or zeros for the second layer. This will increase the chance of getting stable weights and biases. From these initial weights and biases of the controller network, with the trained weights and biases of the NN plant, we will compute the initial $\mathbf{Q}$, $\mathbf{T}$ and Λ matrices. These matrices will be recalculated after each k-step-ahead training segment.

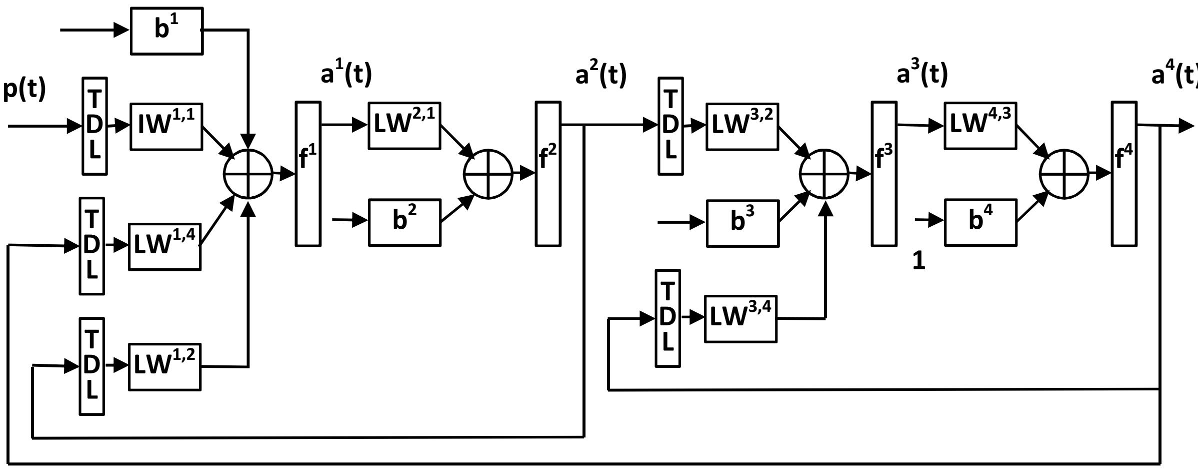

Figure 6.5: Neural network plant model and neural network controller

Next, we will do one step-ahead training. To do this, two output feedback loops will be opened. Thus the output becomes the second input of the network, as in Fig. 6.6. Therefore, $LW^{1,4}$ becomes $IW^{1,2}$, $LW^{3,4}$ becomes $IW^{3,2}$. The training data will be generated from the reference model. When we train this open loop network, the weights and biases in layers 3 and 4 are kept unchanged, and only the weights and biases in the first two layers are updated.

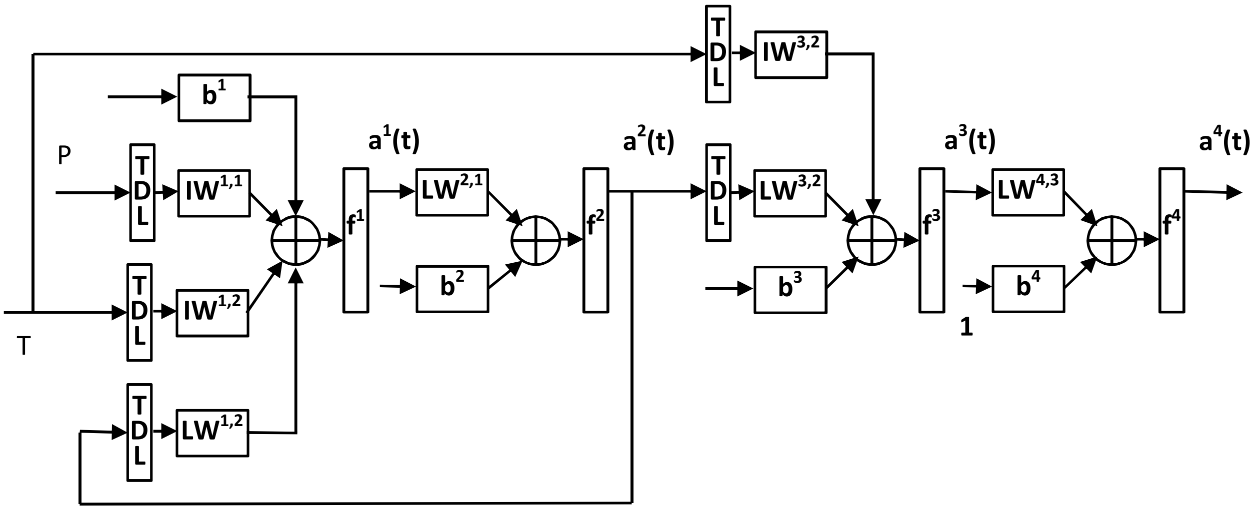

Figure 6.6: Open loop network for controller training

After one step-ahead training, we will do k step-ahead training as we did for the plant training. We will use the network in Fig. 6.5 for k step-ahead training. Thus, $IW^{1,2}$ becomes $LW^{1,4}$ and $IW^{3,2}$ becomes $LW^{3,4}$. When we train the controller network, the weights and biases of the plant network are kept constant.

6.2 Design the MRC for a linear system

In this section, we design an NN controller for a linear plant using the modified trainscg. Our object here is to illustrate and to verify the proposed method.

6.2.1 Plant model

The linear plant that we have chosen to demonstate the modified algorithm with is

$$G(z) = \frac{z^{-1}}{1 - z^{-1} + 0.25z^{-2}} \tag{6.1}$$

The NN representation for this plant is

$$n^1(k) = IW^{1,1}u(k-1) + LW^{1,1}[a^1(k-1)\ a^1(k-2)]^T$$

$$a^1(k) = n^1(k) \tag{6.2}$$

where $u(k)$ is the input to the NN plant, $IW^{1,1} = 1$, $LW^{1,1} = [1 \ -0.25]$ and a^1 is the NN plant output. The plant network is shown in Fig. 6.7. It has one neuron and the activation function is linear.

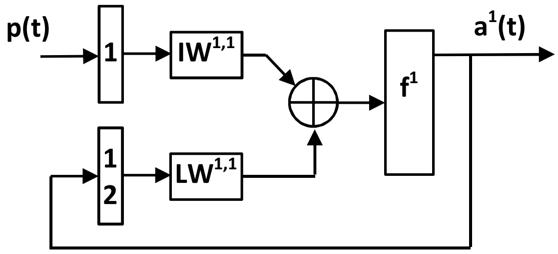

Figure 6.7: The plant network

In this example, we don't need to train the NN plant because it is known. The next step is to choose a reference model.

6.2.2 Model reference

We choose the following continuous-time reference model:

$$G(s) = \frac{144}{s^2 + 24s + 144} \tag{6.3}$$

We sample this model every 0.01 sec to generate a training data set. The reference input and the output are shown in Fig. 6.8. The next step is to create an NN controller and train it.

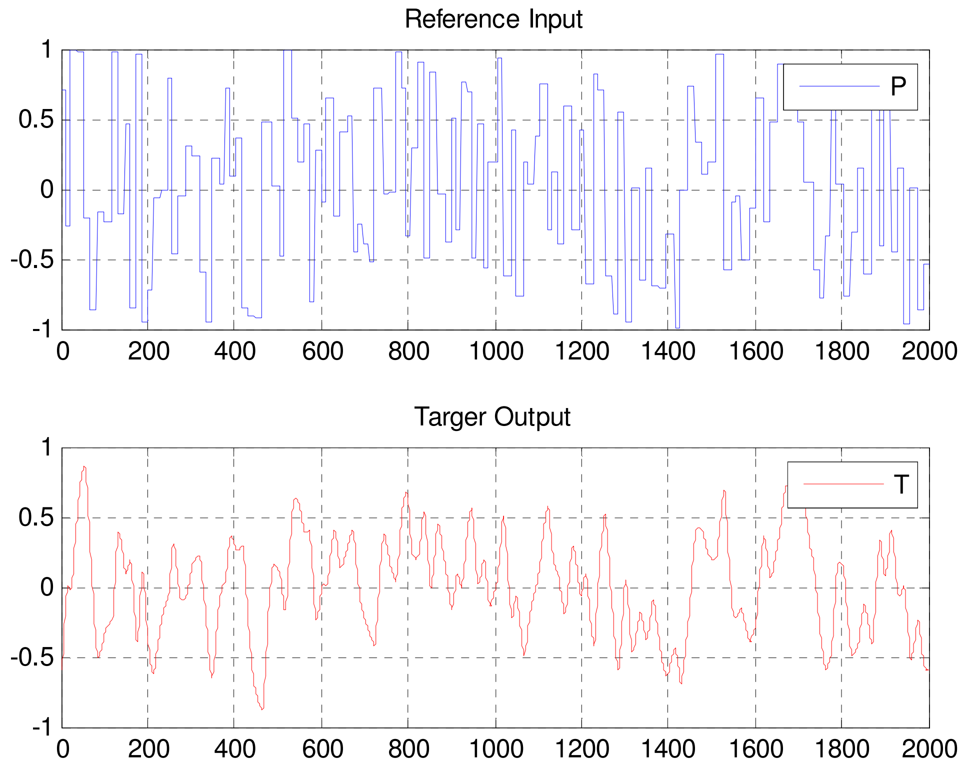

Figure 6.8: The reference input and the target

6.2.3 Controller training

In this section, we will train a controller network using the modified trainscg. The NN controller has one neuron with a linear activation function, delays 1 and 2 from the neuron output, delay 1 from the input, and delays 1 and 2 from the network output. The NN-based MRC system is shown in Fig. 6.9.

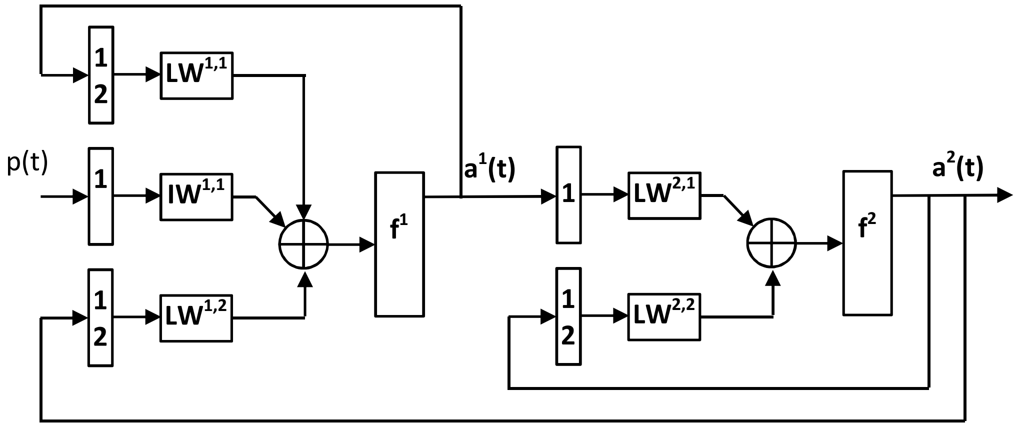

Figure 6.9: NN-based MRC system for the linear plant

The NN controller will be trained with different values for the penalty term coefficient σ. By increasing σ, we can increase the weight on λ and force the system to be stable. We use random weights in the stable area as initial weights of the controller network. First, small random weights are chosen. Then we check the stability of the network. If it is unstable, we multiply all these weights by a number less than 1 and check stability again. We keep doing this until the system is stable. From these initial stable weights and the weights of the NN plant, we find the matrices $\mathbf{Q}$, $\mathbf{T}$ and Λ for one-step-ahead training. After each k step-ahead training stage, we recalculate these matrices from the current weights of the network. Table 6.1 shows the maximum closed loop pole magnitude (MPM), the maximum eigenvalue (λ_m) of $-\mathbf{P}_2$, the mean square error (MSE) and the maximum absolute error (MAE) after 1998 step-ahead training for different values of σ.

It can be seen that when σ increases, the maximum pole magnitude goes down, which indicates greater stability margin, and the error as well as the mean square error goes up. For all cases, $MPM < 1$ and $\lambda_m < 0$, which means that the system is stable.

In conclusion, the modified trainscg works well for the linear NN-based MRC system. In the next section, we will demonstrate the algorithm for a nonlinear physical system.

σ	MPM	MAE	MSE	λ_m
10^{-5}	0.8489	0.1247	$4.1 * 10^{-5}$	$-3.6 * 10^{-5}$
10^{-4}	0.8428	0.1247	$5.5 * 10^{-5}$	$-2.4 * 10^{-5}$
10^{-3}	0.8279	0.1247	$9.3 * 10^{-5}$	$-2.8 * 10^{-5}$
10^{-2}	0.3560	0.2640	$7.1 * 10^{-4}$	$-6.9 * 10^{-5}$
10^{-1}	0.3751	0.3980	$1.5 * 10^{-3}$	$-2.7 * 10^{-5}$

Table 6.1: MPM, MAE, MSE and λ_m after 1998 step ahead training

6.3 Design the MRC for a magnetic levitation system

In this section, we design an NN controller for a magnetic levitation system using the modified trainscg. First, we introduce a magnetic levitation system. Then we train the plant using the Levenberg-Marquardt algorithm (trainlm in [29]). Finally, we train the NN controller network using the modified trainscg.

6.3.1 Magnetic levitation system

The magnetic levitation system is shown in Fig. 6.10. This is a simplified version of the MAGLEV train system [32].

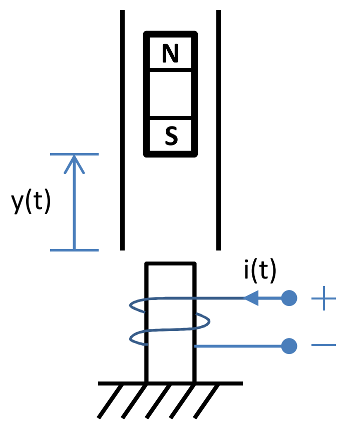

Figure 6.10: The magnetic levitation system

In this system, the magnet is placed above the electromagnet. It can only move in the vertical direction. $y(t)$ is the distance of the magnet from the electromagnet. $i(t)$ is the current flowing in the electromagnet. The equation of motion can be represented as follows [29]

$$\frac{d^2y(t)}{dt} = -g + \frac{\alpha}{M}\frac{i^2(t)}{dt} - \frac{\beta}{M}\frac{dy(t)}{dt} \tag{6.4}$$

where M is the mass of the magnet, g is the gravitational constant, β is a viscous friction coefficient and α is a field strength constant. This is a nonlinear system with one input and one output. The input is the current and the output is the position of the magnet. Our goal is to control the position of the magnet such that it tracks a target.

6.3.2 Plant training

We use Equation (6.4) to generate a training data set for plant training. The parameters are $M = 3$, $g = 9.8$, $\beta = 12$ and $\alpha = 15$. The data training includes 4000 data points. It is shown in Fig. 6.11, where P is the control input, which is applied to the plant and the target T is the corresponding output of the plant.

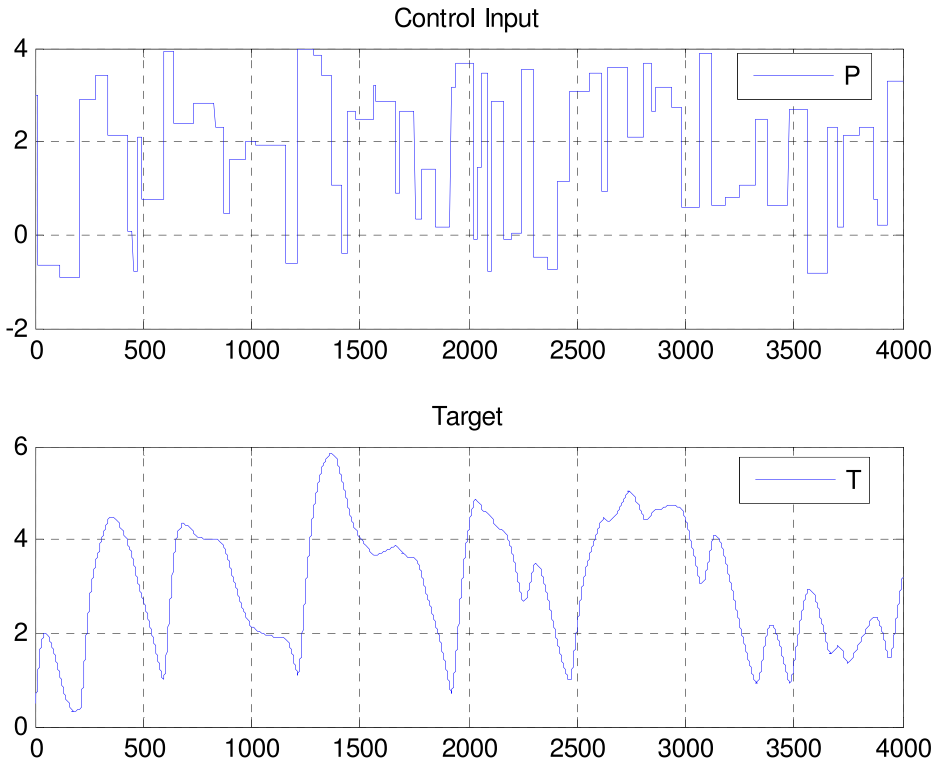

Figure 6.11: Control input and target

The NN plant model is shown in Fig. 6.3. It includes 10 neurons in the first layer, three delays in the input, and two delays in the feedback output. We use *trainlm* [29] to consecutively do from one-step-ahead training up to 3997-step-ahead training. After the final training, we have the performance index as in Fig. 6.12, the network output and the error as in Fig. 6.13.

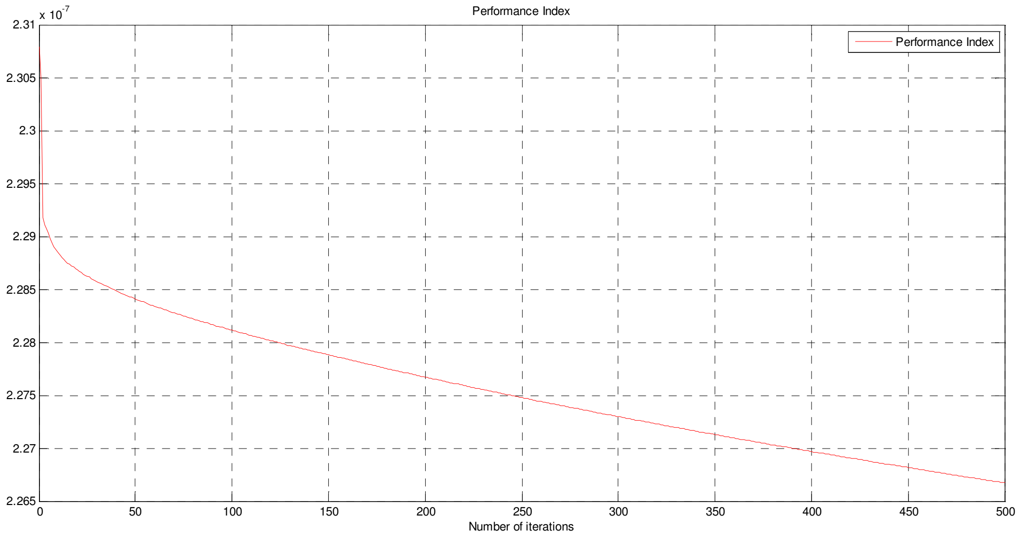

Figure 6.12: Performance Index

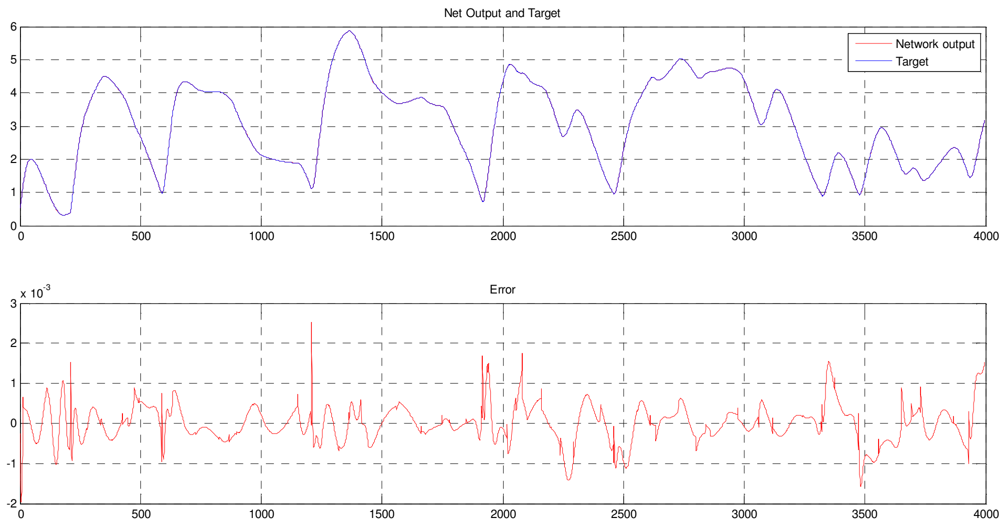

Figure 6.13: The network output, the target and the error

In summary, the model error is very small. The maximum error is less than $3 * 10^{-3}$, so the trained NN plant is accurate enough. It will be used during training of the NN controller in the next section. During the controller training, the weights and biases of the plant network are kept constant.

6.3.3 Controller training

First, we have to choose a reference model. In this example, the reference model is chosen as follows

$$G(s) = \frac{9}{s^2 + 6s + 9} \tag{6.5}$$

An input sequence of step functions with random magnitude and random width is generated. This input P is applied to the input of the reference model (6.5). Then we sample the system output with a sampling time $T_s = 0.01$. The sampled output T is used as the target. The reference input and the target are shown in Fig. 6.14.

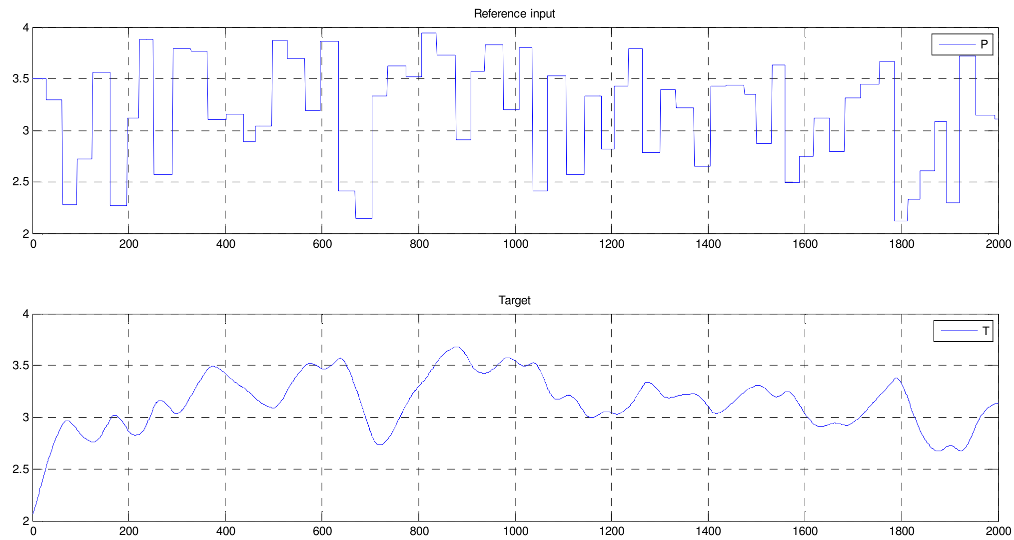

Figure 6.14: The reference model input and output (target)

This training data P and T will be used to train an NN controller. In this example, the NN controller is chosen as in Fig. 6.5. We choose 10 neurons and $tanh$ as the activation function in the first layer, with two delays from the network output, two delays from the second layer and two delays in the input. The second layer has one neuron with linear activation function.

We use modified trainscg to train the controller network, with different values for the penalty parameter σ. Small initial random weights and biases are used for the NN controller for one-step-ahead training. Using these initial weights and biases, and the trained weights and biases of the plant network, we compute the matrices **Q**, **T** and Λ. After each k-step-ahead training stage, these matrices will be updated.

The maximum eigenvalue is shown in Table 6.2, and the maximum absolute error is shown in Table 6.3, after k-step-ahead training with different σ.

σ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
100 step-ahead	0.0559	0.1006	0.3188	0.5182	0.3313
200 step-ahead	0.0494	0.1412	0.6069	0.7375	0.4175
300 step-ahead	0.0413	0.1245	0.7575	0.8497	0.4645
400 step-ahead	0.0413	0.1424	0.8380	0.9247	0.5063

Table 6.2: The maximum eigenvalue with different σ

σ	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}
100 step-ahead	0.1116	0.0790	0.0421	0.0403	0.0399
200 step-ahead	0.0645	0.0465	0.0334	0.0353	0.0244
300 step-ahead	0.0769	0.0396	0.0286	0.0286	0.0281
400 step-ahead	0.0476	0.0202	0.0165	0.0174	0.0141

Table 6.3: The maximum absolute error with different σ

We would expect that the error would increase as σ increases, because more weight is being placed on the maximum eigenvalue, and therefore relatively less weight is being placed on the error. This is clearly shown in Table 6.3. We would also expect that the maximum eigen value of $-\mathbf{P}_2$ would decrease as σ increases. This general trend is seen in Table 6.2. This pattern is not as clear, because the maximum eigenvalues in each entry in Table 6.2 are not exactly comparable. In each case, the weights and biases were different, and so the **Q**, **T** and Λ matrices were also different.

The following figures demonstrate that the modified trainscg algorithm works effectively. Figure 6.15 shows a typical plot of mean square error versus iteration. Figure 6.16

shows the maximum eigenvalue versus iteration, and Figure 6.17 shows the combined performance index versus iteration. Although MSE and maximum eigenvalue may sometimes increase, the combined performance index always decreased in all of our test cases.

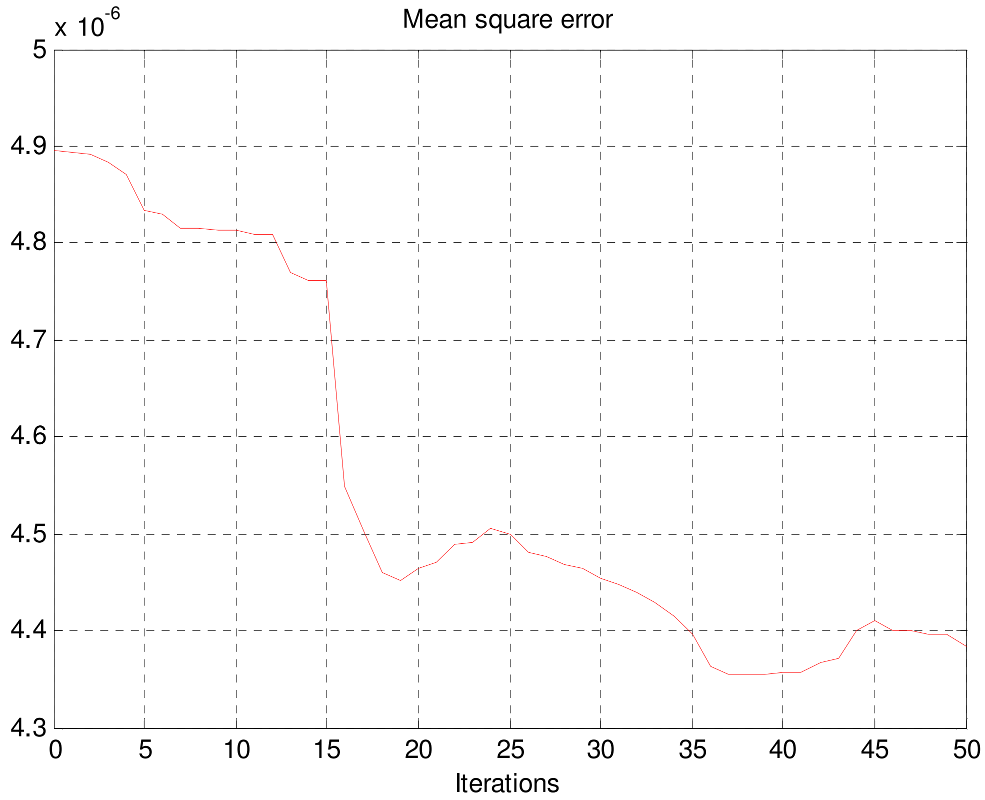

Figure 6.15: The mean square error with $\sigma = 10^{-6}$ after 10-step-ahead training

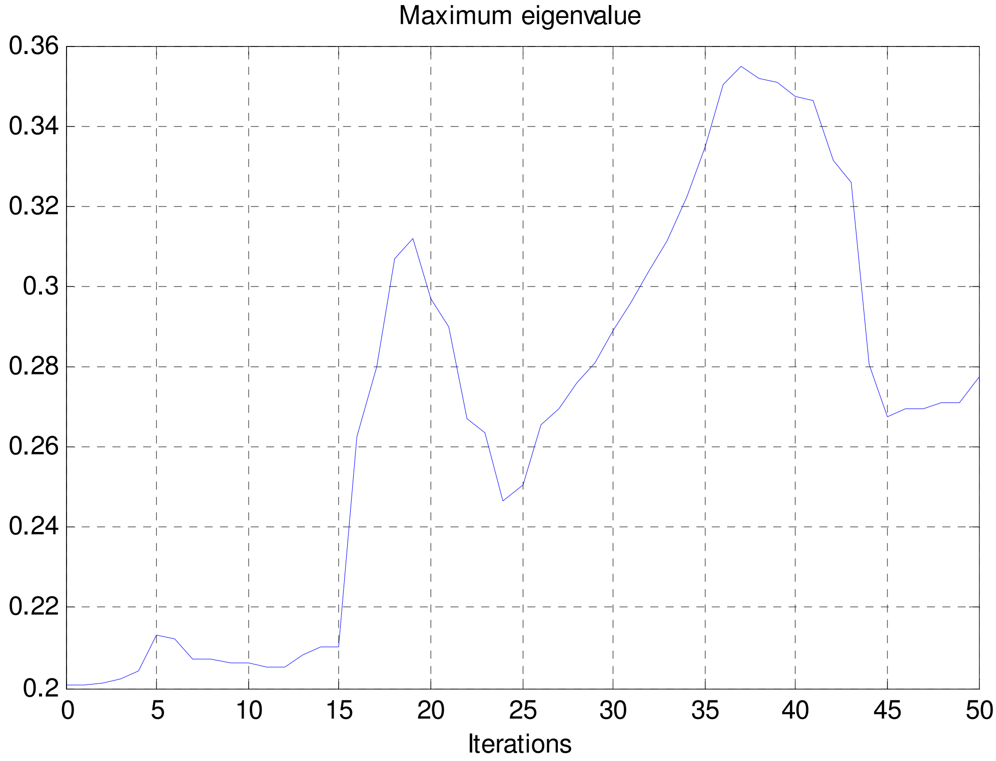

Figure 6.16: The maximum eigenvalue with $\sigma = 10^{-6}$ after 10-step-ahead training

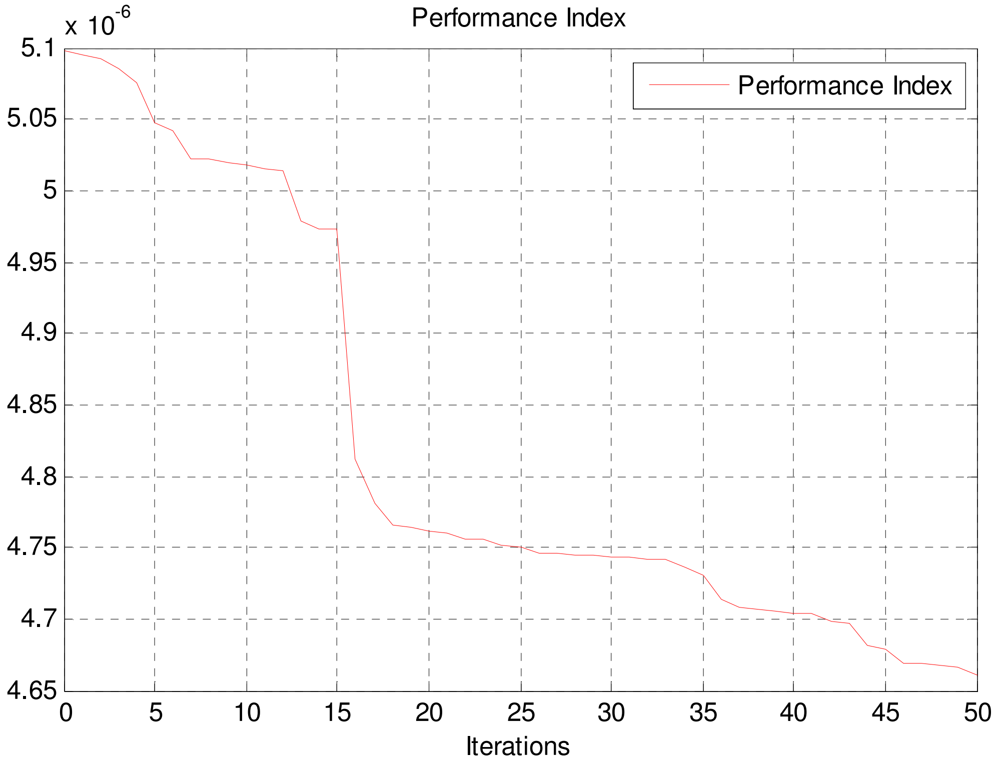

Figure 6.17: The combined performance index with $\sigma = 10^{-6}$ after 10-step-ahead training

6.4 Conclusions

We proposed a novel training algorithm for maintaining system stability. It is demonstrated through two examples: the linear plant and the magnetic levitation system. The results show that it is possible to train recurrent neural networks for stability using the new training algorithm. This has been only a demonstration of the potential use of our novel dissipativity criteria for stable training of RNNs. Future work will be needed to refine the algorithm to achieve consistent stable training.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In this work, we have used dissipativity theory to analyze the stability of a general class of discrete-time dynamic neural networks, called Layered Digital Dynamic Networks (LDDNs). To our knowledge, this is the first time that dissipativity theory has been applied to the analysis of stability in discrete-time neural networks. The application of dissipativity theory requires the selection of supply rate functions. The flexibility in choosing the supply rate allows the development of a variety of stability criteria. In this report, we have developed three different sets of stability criteria, based on three different choices for the supply rate function. We do not claim that these sets are the best that can be obtained. However, the use of dissipativity theory for the analysis of LDDNs opens up the possibility for additional criteria to be developed.

We have tested our dissipativity based (DB) criteria on a wide variety of recurrent neural networks and have compared the results with two other state-of-the-art methods. We have analyzed the performance of the various criteria on cases where they perform well and also on cases where they fail to perform. All of the methods tend to perform worse as the network responses oscillate more, have larger system matrices and take longer to converge. Our DB criterion performed at least as well as Liu's criterion [6, p. 1382] on all of the networks that we tested. In two of 22 cases, Barabanov's method [18, p. 4554] was able to determine stability when the DB criterion was not able to. These two cases represented systems that were in a form upon which the Barabanov method was designed.

The DB methods described in this work were derived for a general recurrent network structure - the LDDN. There are LDDN architectures to which the Liu and Barabanov methods cannot be applied (test problem 23, for example). In these cases, only our DB methods are appropriate (of the three methods analyzed for this report). However, the same can also be said of the DDSNNM architecture of Liu. There are certain DDSNNM structures that cannot be represented in the LDDN format or in the recurrent network structure used by Barabonov [18, p. 4553]. Each method is best suited to the architecture for which it was designed.

We have proposed a new training method using the DB criterion to train recurrent neural networks for stability. The standard performance index is modified with an additional term consisting of the maximum eigenvalue of the matrix $-\mathbf{P}_2$ multiplied by a constant σ. The important thing is to compute the first derivative of the modified performance index with respect to weights and biases. We use the standard backpropagation algorithm to compute the gradient of the mean square error. Then, we show how to compute the gradient of the maximum eigenvalue with respect to the network weights. By combining these two results, we have the gradient of the modified performance index with respect to the network weights. The weights can be updated by using any gradient-based learning algorithm. In this work, we use the scaled conjugate gradient algorithm, which is already implemented in the Neural Network Toolbox. The modified algorithm was tested on two examples of NN-based MRC systems. The tests demonstrated the potential of the modified algorithm to produce stable training.

7.2 Future Work

One area where the stability analysis of recurrent networks is very important is neural network control. After a neural network controller has been designed, it is important to verify that the closed loop control system is stable. Also, it would be desirable to maintain the stability of the closed loop system throughout the training process. This is because there exist spurious valleys in the error surfaces of recurrent networks in regions where the network is unstable. We can avoid these valleys by maintaining stability during training. So, our future work will focus on improving the proposed training method for nonlinear systems and developing new DB-based stability criteria, which are less conservative. We will also use other gradient-based learning algorithms to implement the stable training method.

CHAPTER 8

APPENDIX (Test Problems)

8.1 Test Problem 01

Consider the one layer network given in [4, p. 300], which can be put in standard form with

$$\mathbf{W}^1 = 0.5\begin{bmatrix} 0.0333 & -0.0355 \\ 1.1882 & -2.2687 \end{bmatrix}, \mathbf{W}^2 = 0.5\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$\mathbf{b} = 0.5[-1.0092\ 3.5970]^T$ and $\mathbf{f} = [tanh\ tanh]^T$.

The DB criterion: $\mathbf{A} = diag(0.76323, 0.12656)$, $\mathbf{B} = diag(0.93757, 0.87661)$, $\mathbf{T} = diag(1000, 122.55)$, $\Lambda = diag(100, 0)$ and

$$\mathbf{Q} = \begin{bmatrix} 933.72 & -16.855 \\ -16.855 & 51.898 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.76323, 0.12656)$, $\mathbf{U} = diag(0.93757, 0.87661)$, $\Lambda = diag(1000, 0)$, $\mathbf{T} = diag(1000, 154.6)$,

$$\mathbf{P} = \begin{bmatrix} 41.699 & -67.822 \\ -67.822 & 130.97 \end{bmatrix} \ and\ \mathbf{G} = \begin{bmatrix} 57.374 & 0.36672 \\ 0.36672 & 0.58042 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.76323, 0.12656)$, $\mathbf{N} = diag(0.93757, 0.87661)$, $\Gamma = diag(1000, 207.55)$,

$$\mathbf{H} = \begin{bmatrix} 190.56 & -43.735 \\ -43.735 & 87.827 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 1000 & -10^{-10} \\ -10^{-10} & 4.2315 * 10^{-10} \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 4.2315 \\ 4.2315 & 1 \end{bmatrix}.$$

8.2 Test Problem 02

Consider the network in [6, p. 1388]. This network can be put into the standard form with

$$\mathbf{W}^1 = \begin{bmatrix} 0 & 0 & -0.5 & -1 \\ 0 & 0 & -0.01 & -0.5 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.8 \end{bmatrix}, \mathbf{W}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0.1 & 0 & 0 \\ 0.1 & 1 & 0 & 0 \end{bmatrix}$$

$\mathbf{b} = [-7 \ \ 7 \ \ 0 \ \ 0]^T$ and $\mathbf{f} = [\text{tanh tanh id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0, 0, 1, 1)$, $\mathbf{B} = diag(1, 1, 1, 1)$,

$\mathbf{T} = diag(109.78, 941.44, 979.18, 1000)$, $\Lambda = diag(0, 0, 10^{-6}, 10^{-6})$ and

$$\mathbf{Q} = \begin{bmatrix} 18.554 & -4.6636 & -4.2217 & -0.43615 \\ -4.6636 & 441.78 & -38.317 & -15.767 \\ -4.2217 & -38.317 & 42.684 & 17.189 \\ -0.43615 & -15.767 & 17.189 & 224.46 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0, 0)$, $\mathbf{U} = diag(1, 1)$, $\Lambda = diag(3.5715 * 10^{-8}, 228.76)$,

$\mathbf{T} = diag(504.19, 1000)$,

$$\mathbf{P} = \begin{bmatrix} 302.39 & 185.76 \\ 185.76 & 1000 \end{bmatrix} \ and\ \mathbf{G} = \begin{bmatrix} 147.31 & -0.023467 \\ -0.023467 & 149.78 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0, 0)$, $\mathbf{N} = diag(1, 1)$, $\Gamma = diag(333.07, 1000)$,

$$\mathbf{H} = \begin{bmatrix} 201.44 & 73.981 \\ 73.981 & 724.64 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 162.79 & -10^{-10} \\ -10^{-10} & 552.05 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 3.1009 * 10^9 \\ 3.1009 * 10^9 & 1 \end{bmatrix}.$$

8.3 Test Problem 03

Consider the two layer network given in [4, p. 301]. The standard form weight matrices are

$$\mathbf{W}^1 = 0.5\begin{bmatrix} -1.3482 & -1.8825 & 1.5 & 0.5 \\ -0.7464 & -0.5695 & 1.2 & -0.1 \\ 0 & 0 & 0.4904 & -0.7599 \\ 0 & 0 & -1.4697 & -1.4608 \end{bmatrix}$$

,

$$\mathbf{W}^2 = 0.5\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0.2 & 0 & 0 \\ -0.5 & 1.40 & 0 & 0 \end{bmatrix}, \ \mathbf{b} = 0.5\begin{bmatrix} 0.3 \\ -0.5 \\ -1 \\ 1 \end{bmatrix}$$

and $\mathbf{f} = [\text{tanh tanh tanh tanh}]^T$.

The DB criterion: $\mathbf{A} = diag(0.32231, 0.41816, 0.25498, 0.2636)$, $\mathbf{B} = diag(0.99439, 0.93639, 0.90631, 0.9752)$, $\mathbf{T} = diag(290.19, 1000, 418.16, 380.73)$, $\Lambda = diag(0, 0, 10^{-6}, 0.00039036)$ and

$$\mathbf{Q} = \begin{bmatrix} 134.2 & 3.3473 & -100 & 24.895 \\ 3.3473 & 363.83 & -100 & -9.6501 \\ -100 & -100 & 171.86 & 34.766 \\ 24.895 & -9.6501 & 34.766 & 120.92 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.32231, 0.41816, 0.25498, 0.2636)$, $\mathbf{U} = diag(0.99439, 0.93639, 0.90631, 0.9752)$, $\Lambda = diag(0, 0, 0, 0)$, $\mathbf{T} = diag(142, 564.36, 211.17, 189.88)$,

$$\mathbf{P} = \begin{bmatrix} 108.4 & 59.729 & -99.995 & 19.375 \\ 59.729 & 220.9 & -100 & 2.8786 \\ -99.995 & -100 & 170.43 & 35.055 \\ 19.375 & 2.8786 & 35.055 & 119.51 \end{bmatrix},$$

and

$$\mathbf{G} = \begin{bmatrix} 20.459 & -29.246 & -8.954 & 4.4233 \\ -29.246 & 106.92 & -24.559 & 2.0016 \\ -8.954 & -24.559 & 45.595 & -11.396 \\ 4.4233 & 2.0016 & -11.396 & 8.211 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.32231, 0.41816, 0.25498, 0.2636)$,

$\mathbf{N} = diag(0.99439, 0.93639, 0.90631, 0.9752)$, $\Gamma = diag(636.85, 1000, 1000, 165.85)$,

$\mathbf{H} =$

$$\begin{bmatrix} 363.39 & -98.097 & -16.095 & 43.689 & -18.712 & -16.793 & 105.52 & -106.22 \\ -98.097 & 951.22 & 42.764 & -63.609 & -1.0098 & -72.852 & -229.65 & 25.213 \\ -16.095 & 42.764 & 374.06 & 65.911 & -421.84 & 86.45 & -38.807 & -26.445 \\ 43.689 & -63.609 & 65.911 & 758.61 & -208.24 & -57.617 & 34.07 & -71.827 \\ -18.712 & -1.0098 & -421.84 & -208.24 & 861.48 & -55.45 & 45.91 & 17.913 \\ -16.793 & -72.852 & 86.45 & -57.617 & -55.45 & 287.95 & 11.896 & -4.9001 \\ 105.52 & -229.65 & -38.807 & 34.07 & 45.91 & 11.896 & 309.59 & 26.527 \\ -106.22 & 25.213 & -26.445 & -71.827 & 17.913 & -4.9001 & 26.527 & 269.63 \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} 103.55 & -33.567 & -1.3562 & -16.466 \\ -33.567 & 628.26 & -1.7904 & -135.58 \\ -1.3562 & -1.7904 & 208.28 & -1e-010 \\ -16.466 & -135.58 & -1e-010 & 381.65 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 2.2319 & 2.7202 & 1.2334 \\ 2.2319 & 1 & 1.2188 & 2.5446 \\ 2.7202 & 1.2188 & 1 & 3.1013 \\ 1.2334 & 2.5446 & 3.1013 & 1 \end{bmatrix}.$$

8.4 Test Problem 04

Consider the example given in [33, p. 1778]. Representing this system in standard form we get

$$\mathbf{W}^1 = \begin{bmatrix} 0.2753 & -0.0306 & 0.2967 & -0.2277 \\ 0.1844 & -0.3387 & 0.1676 & -0.0663 \\ 0 & 0 & 0.6428 & 0.2309 \\ 0 & 0 & -0.1106 & 0.5839 \end{bmatrix}, \mathbf{W}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.3064 & -0.0631 & 0 & 0 \\ 0.2937 & 0.2769 & 0 & 0 \end{bmatrix},$$

and $\mathbf{f} = [\tanh \tanh \tanh \tanh]^T$.

The DB criterion: $\mathbf{A} = diag(0.81975, 0.84453, 0.6808, 0.67383)$,
$\mathbf{B} = diag(1, 1, 1, 1)$, $\mathbf{T} = diag(1000, 756.64, 1000, 1000)$,
$\Lambda = diag(8.4132 * 10^{-9}, 1.3655 * 10^{-7}, 3.9446 * 10^{-8})$ and

$$\mathbf{Q} = \begin{bmatrix} 274.24 & 91.103 & -68.509 & -68.402 \\ 91.103 & 251.03 & -99.277 & 15.481 \\ -68.509 & -99.277 & 280.07 & 43.601 \\ -68.402 & 15.481 & 43.601 & 205.53 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.81975, 0.84453, 0.6808, 0.67383)$,
$\mathbf{U} = diag(1, 1, 1, 1)$, $\mathbf{T} = diag(1000, 770.69, 1000, 1000)$, $\Lambda = diag(0, 0, 0, 0)$,

$$\mathbf{P} = \begin{bmatrix} 265.47 & -28.351 & 21.634 & -100 \\ -28.351 & 403.35 & -80.933 & 46.384 \\ 21.634 & -80.933 & 434.49 & 75.265 \\ -100 & 46.384 & 75.265 & 362.42 \end{bmatrix},$$

and

$$\mathbf{G} = \begin{bmatrix} 167.45 & -70.26 & -55.17 & 14.998 \\ -70.26 & 280.63 & 21.786 & -95.94 \\ -55.17 & 21.786 & 72.125 & -5.1197 \\ 14.998 & -95.94 & -5.1197 & 96.704 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.81975, 0.84453, 0.6808, 0.67383)$,

$\mathbf{N} = diag(1, 1, 1, 1)$, $\Gamma = diag(1000, 954.94, 1000, 998.52)$,

$\mathbf{H} =$

$$\begin{bmatrix} 1000 & -190.37 & -1.7249 & -125.4 & 51.001 & -65.832 & -53.517 & 173.12 \\ -190.37 & 805.55 & 143.72 & -44.463 & -60.447 & -33.642 & -90.638 & 193.26 \\ -1.7249 & 143.72 & 686.94 & -208.69 & -198.63 & 85.043 & 22.886 & -102.38 \\ -125.4 & -44.463 & -208.69 & 482.96 & 51.316 & -88.32 & 93.608 & -59.962 \\ 51.001 & -60.447 & -198.63 & 51.316 & 908.23 & -108.71 & -50.821 & 44.838 \\ -65.832 & -33.642 & 85.043 & -88.32 & -108.71 & 884.85 & 27.148 & 24.569 \\ -53.517 & -90.638 & 22.886 & 93.608 & -50.821 & 27.148 & 623.88 & -74.082 \\ 173.12 & 193.26 & -102.38 & -59.962 & 44.838 & 24.569 & -74.082 & 763.18 \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} 1000 & -10^{-6} & -4.2819 & -10^{-6} \\ -10^{-6} & 625.22 & -35.158 & -12.784 \\ -4.2819 & -35.158 & 1000 & -10^{-6} \\ -10^{-6} & -12.784 & -10^{-6} & 810.79 \end{bmatrix}, \ and\ \beta = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

8.5 Test Problem 05

$$\mathbf{E} = \begin{bmatrix} 0.4498 & -1.3460 \\ 0.6169 & 0.3715 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} -1.1407 & -0.4336 \\ -1.0933 & -0.1685 \end{bmatrix},$$

$\mathbf{s}_1 = [-0.2185\ 0.5413]^T$ and $\mathbf{f} = [\text{tanh tanh id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.21589, 0.073411, 1, 1)$, $\mathbf{B} = diag(0.91974, 0.68839, 1, 1)$, $\mathbf{T} = diag(198.75, 614.03, 1000, 1000)$, $\Lambda = diag(0, 0, 10^{-6}, 10^{-6})$ and

$$\mathbf{Q} = \begin{bmatrix} 104.97 & -51.538 & -52.084 & -95.512 \\ -51.538 & 632.6 & 394.96 & -99.792 \\ -52.084 & 394.96 & 400.31 & -10.052 \\ -95.512 & -99.792 & -10.052 & 175.29 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.21589, 0.073411)$, $\mathbf{U} = diag(0.91974, 0.68839)$, $\Lambda = diag(1.2472 * 10^{-6}, 0)$, $\mathbf{T} = diag(1000, 770.69, 1000, 1000)$,

$$\mathbf{P} = \begin{bmatrix} 580.65 & 99.456 \\ 99.456 & 167.37 \end{bmatrix} \; and \; \mathbf{G} = \begin{bmatrix} 66.297 & -1.183 \\ -1.183 & 61.627 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.21589, 0.073411)$, $\mathbf{N} = diag(0.91974, 0.68839)$, $\Gamma = diag(388.59, 1000)$,

$$\mathbf{H} = \begin{bmatrix} 310.09 & 42.138 \\ 42.138 & 68.108 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 4.803 * 10^{-6} & -10^{-6} \\ -10^{-6} & 0.00093555 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 4.803 \\ 4.803 & 1 \end{bmatrix}.$$

8.6 Test Problem 06

$$\mathbf{E} = \begin{bmatrix} -2.1382 & -1.0618 \\ 0.0389 & -0.2408 \\ -0.2108 & -1.0018 \\ 0.7125 & -0.3998 \end{bmatrix}, \mathbf{W}^T = \begin{bmatrix} -0.0544 & -2.7524 \\ -0.0989 & -1.2134 \\ -0.1434 & 1.0171 \\ 0.2366 & -0.4306 \end{bmatrix},$$

$\mathbf{s}_1 = [0.6501 \; - 0.4852]^T$ and $\mathbf{f} = [\text{tanh tanh id id id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.33079, 2.3982 * 10^{-6}, 1, 1, 1, 1)$,

$\mathbf{B} = diag(0.81069, 0.25584, 1, 1, 1, 1)$,

$\mathbf{T} = diag(99.97, 125.78, 1000, 956.55, 998.82, 961.05)$,

$\Lambda = diag(0.00034784, 0, 10^{-6}, 10^{-6}, 10^{-6}, 0.00077597)$ and

$$\mathbf{Q} = \begin{bmatrix} 2461.1 & 1109.6 & 1011.2 & 29.791 & 14.364 & 144.84 \\ 1109.6 & 2117.3 & 743.8 & 231.45 & 926.58 & 609.29 \\ 1011.2 & 743.8 & 567.01 & 47.012 & 29.205 & 328.99 \\ 29.791 & 231.45 & 47.012 & 265.26 & 121.42 & -35.132 \\ 14.364 & 926.58 & 29.205 & 121.42 & 812.93 & -43.762 \\ 144.84 & 609.29 & 328.99 & -35.132 & -43.762 & 695.36 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.33079, 2.3982 * 10^{-6})$, $\mathbf{U} = diag(0.81069, 0.25584)$, $\Lambda = diag(0.36978, 0)$, $\mathbf{T} = diag(1000, 164.49)$,

$$\mathbf{P} = \begin{bmatrix} 155.88 & -20.026 & -91.828 & -95.885 \\ -20.026 & 128.2 & 67.815 & 32.503 \\ -91.828 & 67.815 & 113.02 & 71.445 \\ -95.885 & 32.503 & 71.445 & 236.97 \end{bmatrix} \; and \; \mathbf{G} = \begin{bmatrix} 20.837 & -0.55989 \\ -0.55989 & 7.5994 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.33079, 2.3982*10^{-6})$, $\mathbf{N} = diag(0.81069, 0.25584)$, $\Gamma = diag(1000, 339.64)$,

$$\mathbf{H} = \begin{bmatrix} 317.03 & 133.95 & -381.52 & -54.939 \\ 133.95 & 957.13 & -299.75 & -241.94 \\ -381.52 & -299.75 & 674.12 & -27.545 \\ -54.939 & -241.94 & -27.545 & 714.63 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 1000 & -10^{-6} \\ -10^{-6} & 0.017865 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 17865 \\ 17865 & 1 \end{bmatrix}.$$

8.7 Test Problem 07

$$\mathbf{E} = \begin{bmatrix} -0.5367 & -0.8914 \\ 1.1566 & 1.0866 \\ 1.1402 & -0.1332 \end{bmatrix}, \mathbf{W}^T = \begin{bmatrix} 0.1399 & -1.3558 \\ -0.2022 & -0.6691 \\ 1.3142 & 1.0448 \end{bmatrix},$$

$\mathbf{s}_1 = [-0.7165 \ -0.8795]^T$ and $\mathbf{f} = [\text{tanh tanh id id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.08737, 0.007409, 1, 1, 1)$,
$\mathbf{B} = diag(0.76153, 0.55182, 1, 1, 1)$, $\mathbf{T} = diag(210.36, 99.142, 987.86, 959.92, 1000)$,
$\Lambda = diag(0, 0.06456, 10^{-6}, 10^{-6}, 10^{-6}, 0.00077597)$ and

$$\mathbf{Q} = \begin{bmatrix} 429.74 & 600.34 & 587.49 & -33.662 & -65.315 \\ 600.34 & 886.98 & 824.87 & -79.6 & -60.587 \\ 587.49 & 824.87 & 873.36 & -22.622 & -74.159 \\ -33.662 & -79.6 & -22.622 & 41.952 & -31.658 \\ -65.315 & -60.587 & -74.159 & -31.658 & 127.32 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.08737, 0.007409)$, $\mathbf{U} = diag(0.76153, 0.55182)$, $\Lambda = diag(0, 3.7044 * 10^{-7})$, $\mathbf{T} = diag(1000, 443.53)$,

$$\mathbf{P} = \begin{bmatrix} 994.17 & 288.8 & 242.55 \\ 288.8 & 154.79 & -99.992 \\ 242.55 & -99.992 & 1000 \end{bmatrix} \ and\ \mathbf{G} = \begin{bmatrix} 35.827 & -9.5597 \\ -9.5597 & 102.88 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.08737, 0.007409)$, $\mathbf{N} = diag(0.76153, 0.55182)$, $\Gamma = diag(1000, 461.85)$,

$$\mathbf{H} = \begin{bmatrix} 797.85 & 408.86 & -4.1171 \\ 408.86 & 320.72 & -171.01 \\ -4.1171 & -171.01 & 592.92 \end{bmatrix}, \mathbf{G} = 10^{-6} \begin{bmatrix} 6.4083 & -1.0002 \\ -1.0002 & 6.4146 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 6.4073 \\ 6.4073 & 1 \end{bmatrix}.$$

8.8 Test Problem 08

$$\mathbf{E} = \begin{bmatrix} 0.4498 & -1.3460 \\ 0.6169 & 0.3715 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0.2692 & 0.7177 \\ -0.0074 & 0.8009 \end{bmatrix},$$

$\mathbf{s}_1 = [-0.2586\ 0.0537]^T$ and $\mathbf{f} = [\text{tanh tanh id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.45842, 0.74239, 1, 1)$, $\mathbf{B} = diag(0.94416, 0.98544, 1, 1)$, $\mathbf{T} = diag(1000, 1000, 994.81, 1000)$, $\Lambda = diag(0, 0, 10^{-6}, 10^{-6})$ and

$$\mathbf{Q} = \begin{bmatrix} 45.541 & -83.784 & -65.824 & 2.8525 \\ -83.784 & 819.24 & 585.23 & -100 \\ -65.824 & 585.23 & 518.02 & 29.31 \\ 2.8525 & -100 & 29.31 & 411.68 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.45842, 0.74239)$, $\mathbf{U} = diag(0.94416, 0.98544)$, $\Lambda = diag(0, 0)$, $\mathbf{T} = diag(1000, 1000)$,

$$\mathbf{P} = \begin{bmatrix} 210.56 & 160.82 \\ 160.82 & 1000 \end{bmatrix} \; and \; \mathbf{G} = \begin{bmatrix} 299.36 & -78.169 \\ -78.169 & 173.97 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.45842, 0.74239)$, $\mathbf{M} = diag(0.94416, 0.98544)$, $\Gamma = diag(1000, 1000)$,

$$\mathbf{H} = \begin{bmatrix} 262.69 & 125.53 \\ 125.53 & 1000 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 800.06 & -10^{-6} \\ -10^{-6} & 1000 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 1.4003 \\ 1.4003 & 1 \end{bmatrix}.$$

8.9 Test Problem 09

$$\mathbf{E} = \begin{bmatrix} -0.4376 & 0.6711 \\ 0.2333 & -0.3892 \\ -0.1496 & 0.3565 \\ 0.8825 & 0.0271 \end{bmatrix}, \mathbf{W}^T = \begin{bmatrix} 0.8775 & 0.2470 \\ 0.3328 & 0.7495 \\ 0.2635 & 0.7885 \\ 0.2241 & 0.1221 \end{bmatrix},$$

$\mathbf{s}_1 = \mathbf{0}$ and $\mathbf{f} = [\text{tanh tanh id id id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.5986, 0.67847, 1, 1, 1, 1)$, $\mathbf{B} = diag(1, 1, 1, 1, 1, 1)$, $\mathbf{T} = diag(949.76, 1000, 1000, 1000, 1000, 1000)$, $\Lambda = diag(0, 0, 10^{-6}, 100, 10^{-6}, 10^{-6})$ and

$$\mathbf{Q} = \begin{bmatrix} 111.28 & 13.247 & 25.708 & 21.974 & -28.664 & -59.065 \\ 13.247 & 188.85 & -36.728 & 163.94 & -70.283 & -95.483 \\ 25.708 & -36.728 & 325.53 & 70.29 & 48.782 & 50.854 \\ 21.974 & 163.94 & 70.29 & 620.19 & 161.65 & -92.585 \\ -28.664 & -70.283 & 48.782 & 161.65 & 435.83 & 114.96 \\ -59.065 & -95.483 & 50.854 & -92.585 & 114.96 & 195.98 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.5986, 0.67847)$, $\mathbf{U} = diag(1, 1)$, $\Lambda = diag(4.9935 * 10^{-5}, 0)$, $\mathbf{T} = diag(1000, 1000)$,

$$\mathbf{P} = \begin{bmatrix} 978.57 & 335.7 & 167.82 & 171.9 \\ 335.7 & 999.77 & 677.6 & 122.11 \\ 167.82 & 677.6 & 1000 & 105.3 \\ 171.9 & 122.11 & 105.3 & 289.55 \end{bmatrix} \; and \; \mathbf{G} = \begin{bmatrix} 248.26 & -6.0496 \\ -6.0496 & 247.97 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.5986, 0.67847)$, $\mathbf{M} = diag(1, 1)$, $\Gamma = diag(1000, 991.51)$,

$$\mathbf{H} = \begin{bmatrix} 1000 & 241.22 & 194.23 & 155.82 \\ 241.22 & 970.92 & 577.69 & 97.549 \\ 194.23 & 577.69 & 991.47 & 91.865 \\ 155.82 & 97.549 & 91.865 & 437.93 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 1000 & -240.56 \\ -240.56 & 912.39 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

8.10 Test Problem 10

$$\mathbf{E} = \begin{bmatrix} 0.1820 & -1.5142 \\ -0.0995 & 0.0070 \\ -0.3681 & 1.2971 \\ -0.1194 & 1.1440 \end{bmatrix}, \mathbf{W}^T = \begin{bmatrix} 0.2361 & 0.5642 \\ 0.6413 & 0.5412 \\ 0.2559 & 0.7715 \\ 0.1664 & 0.2860 \end{bmatrix},$$

$\mathbf{s}_1 = \mathbf{0}$ and $\mathbf{f} = [\text{tanh tanh id id id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.726, 0.3722, 1, 1, 1, 1)$, $\mathbf{B} = diag(1, 1, 1, 1, 1, 1)$, $\mathbf{T} = diag(1000, 1000, 994.43, 1000, 1000, 1000)$, $\Lambda = diag(0, 0, 10^{-6}, 100, 10^{-6}, 10^{-6})$ and

$$\mathbf{Q} = \begin{bmatrix} 273.56 & -73.734 & -29.437 & 93.543 & -58.562 & 48.331 \\ -73.734 & 1766.5 & 1130.1 & 36.435 & -69.631 & -83.295 \\ -29.437 & 1130.1 & 897.18 & 105.85 & 97.635 & 61.019 \\ 93.543 & 36.435 & 105.85 & 650.02 & 87.074 & 6.3594 \\ -58.562 & -69.631 & 97.635 & 87.074 & 313.06 & 2.5765 \\ 48.331 & -83.295 & 61.019 & 6.3594 & 2.5765 & 208.29 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.726, 0.3722)$, $\mathbf{U} = diag(1, 1)$, $\Lambda = diag(0.036388, 0)$, $\mathbf{T} = diag(1000, 1000)$,

$$\mathbf{P} = \begin{bmatrix} 652.64 & 286.48 & 620.53 & 339.77 \\ 286.48 & 951.01 & 455.72 & 103.38 \\ 620.53 & 455.72 & 999.99 & 292.86 \\ 339.77 & 103.38 & 292.86 & 403.84 \end{bmatrix} \ and\ \mathbf{G} = \begin{bmatrix} 462.6 & -37.436 \\ -37.436 & 123.96 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.726, 0.3722)$, $\mathbf{M} = diag(1, 1)$, $\Gamma = diag(942.82, 1000)$,

$$\mathbf{H} = \begin{bmatrix} 816.33 & 578.16 & 598.65 & 522.47 \\ 578.16 & 825.05 & 476.38 & 489.74 \\ 598.65 & 476.38 & 1000 & 286.54 \\ 522.47 & 489.74 & 286.54 & 685.9 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 415.61 & -363.06 \\ -363.06 & 1000 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

8.11 Test Problem 11

$$\mathbf{E} = \begin{bmatrix} -1.5369 & -1.4479 \\ 2.0182 & 0.6986 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0.5301 & 1.1132 \\ -0.7163 & 0.3181 \end{bmatrix},$$

$\mathbf{s}_1 = [-2.0962 \ -0.3127]^T$ and $\mathbf{f} = [\text{tanh tanh id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.00053451, 0.0053027, 1, 1)$, $\mathbf{B} = diag(0.38817, 0.40828, 1, 1)$, $\mathbf{T} = diag(1000, 350.31, 821.55, 1000)$, $\Lambda = diag(0, 0, 10^{-6}, 10^{-6})$ and

$$\mathbf{Q} = \begin{bmatrix} 1673.5 & 793.22 & 676.56 & -91.493 \\ 793.22 & 1189.6 & 891.01 & 227.33 \\ 676.56 & 891.01 & 737.18 & 221.63 \\ -91.493 & 227.33 & 221.63 & 252.43 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.00053451, 0.0053027)$, $\mathbf{U} = diag(0.38817, 0.40828)$, $\Lambda = diag(0, 0)$, $\mathbf{T} = diag(1000, 457.88)$,

$$\mathbf{P} = \begin{bmatrix} 87.837 & 73.387 \\ 73.387 & 288.62 \end{bmatrix} \ and\ \mathbf{G} = \begin{bmatrix} 113.99 & -59.337 \\ -59.337 & 72.36 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.00053451, 0.0053027)$, $\mathbf{N} = diag(0.38817, 0.40828)$, $\Gamma = diag(1000, 435.69)$,

$$\mathbf{H} = \begin{bmatrix} 229.27 & 258.81 \\ 258.81 & 412.7 \end{bmatrix}, \mathbf{G} = 10^{-6} \begin{bmatrix} 1.575501 & -1 \\ -1 & 1.575501 \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 1.5755 \\ 1.5755 & 1 \end{bmatrix}.$$

8.12 Test Problem 12

$$\mathbf{E} = \begin{bmatrix} -1.7083 & -0.3869 \\ 1.2134 & -1.0272 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0.0337 & 0.3472 \\ -1.2111 & -1.0742 \end{bmatrix},$$

$\mathbf{s}_1 = [0.4811 \; -0.0876]^T$ and $\mathbf{f} = [\text{tanh tanh id id}]^T$.

The DB criterion: $\mathbf{A} = diag(0.58581, 0.026454, 1, 1)$, $\mathbf{B} = diag(0.98733, 0.68489, 1, 1)$, $\mathbf{T} = diag(860.8, 173.26, 997.37, 1000)$, $\Lambda = diag(0, 0, 10^{-6}, 10^{-6})$ and

$$\mathbf{Q} = \begin{bmatrix} 1855.5 & 362.01 & 939.2 & -93.541 \\ 362.01 & 1028.2 & 635.64 & 661.26 \\ 939.2 & 635.64 & 776.97 & 322.28 \\ -93.541 & 661.26 & 322.28 & 527.61 \end{bmatrix}.$$

Liu's criterion: $\mathbf{Q} = diag(0.58581, 0.026454)$, $\mathbf{U} = diag(0.98733, 0.68489)$, $\Lambda = diag(0, 0)$, $\mathbf{T} = diag(995.2, 611.75)$,

$$\mathbf{P} = \begin{bmatrix} 513.77 & 295.37 \\ 295.37 & 262.31 \end{bmatrix} \; and \; \mathbf{G} = \begin{bmatrix} 16.938 & 1.7133 \\ 1.7133 & 13.071 \end{bmatrix}.$$

Barabanov's criterion: $\mathbf{M} = diag(0.58581, 0.026454)$, $\mathbf{N} = diag(0.98733, 0.68489)$, $\Gamma = diag(1000, 563.99)$,

$$\mathbf{H} = \begin{bmatrix} 256.06 & 141.33 \\ 141.33 & 130.33 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0.0049901 & -10^{-6} \\ -10^{-6} & 8.3245 * 10^{-6} \end{bmatrix},$$

and

$$\beta = \begin{bmatrix} 1 & 8.3245 \\ 8.3245 & 1 \end{bmatrix}.$$

8.13 Test problem 13

Consider the network given in [8] $x(k+1) = tanh(\mathbf{W}x(k))$ where

$$\mathbf{W}^1 = \begin{bmatrix} 0.5893 & -0.4047 & 0.3142 & 0.3133 & -0.5308 \\ 1.0074 & -0.7935 & 0.7659 & 0.2278 & 0.0204 \\ -1.0197 & -0.0221 & 0.1484 & 0.1643 & 0.8982 \\ 1.1161 & -0.7743 & 0.4514 & -0.8473 & -0.0883 \\ 0.6870 & -1.0181 & 0.0379 & -0.5418 & -0.6798 \end{bmatrix}$$

$\mathbf{W}^2 = \mathbf{0}$, $\mathbf{b} = \mathbf{0}$ and $\mathbf{f} = [\text{tanh tanh tanh tanh tanh}]^T$.

8.14 Test problem 14

Given a network (4.3) where $\mathbf{W} = diag(1, 1)$, $\mathbf{s}_1 = [0\ 0]^T$ and

$$\mathbf{E} = \begin{bmatrix} -1.0510 & 1.6516 \\ -1.3141 & 1.1566 \end{bmatrix}$$

8.15 Test problem 15

Given a network (4.3) where $\mathbf{W} = diag(1, 1)$, $\mathbf{s}_1 = [0\ 0]^T$ and

$$\mathbf{E} = \begin{bmatrix} 0.4498 & -1.3460 \\ 0.6169 & 0.3715 \end{bmatrix}.$$

8.16 Test problem 16

Given a network (4.3) where

$$\mathbf{E} = \begin{bmatrix} -0.0176 & 1.4660 \\ -1.9825 & -0.3525 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0.7133 & 0.5571 \\ 0.7637 & 0.5651 \end{bmatrix},$$

and $\mathbf{s}_1 = [-0.1768\ 1.5514]^T$.

8.17 Test problem 17

Consider a network (4.3) where

$$\mathbf{E} = \begin{bmatrix} 0.4889 & -0.1699 \\ 1.9743 & 1.4636 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 0.3325 & -1.1325 \\ -0.6563 & 1.9766 \end{bmatrix},$$

and $\mathbf{s}_1 = [-0.6537\ 1.722]^T$.

8.18 Test problem 18

Given a network (4.3) where

$\mathbf{E}\ (Column(1:5)) =$

$$\begin{bmatrix} -0.1442 & -0.1923 & -0.2243 & 0.5341 & 0.2312 \\ 1.2841 & 0.7068 & 0.6862 & -0.8575 & 1.2335 \end{bmatrix},$$

$\mathbf{E}\ (Column(6:10)) =$

$$\begin{bmatrix} 0.8103 & 0.2148 & -0.3047 & 0.7268 & 0.2490 \\ -1.1181 & -0.6498 & -0.8863 & 0.5849 & 1.3329 \end{bmatrix},$$

$\mathbf{W}^T\ (Column(1:5)) =$

$$\begin{bmatrix} -0.0822 & -0.6546 & 0.1519 & -0.6173 & -1.1329 \\ -0.5581 & -0.3620 & -1.4393 & -0.4941 & -0.5955 \end{bmatrix},$$

$\mathbf{W}^T\ (Column(6:10)) =$

$$\begin{bmatrix} 0.51552 & 0.5686 & -2.8238 & -0.066408 & 0.26577 \\ 0.37745 & -0.40056 & -1.1589 & 0.16272 & 0.52101 \end{bmatrix},$$

and $\mathbf{s}_1 = [0.8594\ -0.1774\ -0.6582\ 0.2043\ 1.4457\ -0.4587\ 0.2136\ 1.5275\ 0.1615\ -0.2330]^T$.

8.19 Test problem 19

Given a network (4.3) where

$\mathbf{E} =$

$$\begin{bmatrix} -1.2676 & -0.0239 & -1.1729 & -0.5952 & 0.9547 \\ -0.9055 & -1.7919 & -1.0372 & -1.3526 & 0.3340 \end{bmatrix},$$

$\mathbf{W}^T =$

$$\begin{bmatrix} -0.2504 & -2.6603 & 1.3109 & 0.4479 & 1.1745 \\ -0.0003 & -1.6884 & -0.3583 & 0.4195 & -0.4836 \end{bmatrix},$$

and $\mathbf{s}_1 = [-0.5025\ -1.6517\ 1.0859\ -0.4030\ 0.5661]^T$.

8.20 Test problem 20

Given a network (4.3) where

$$\mathbf{E} = \begin{bmatrix} 0.1526 & 0.9272 \\ -2.0829 & 0.3752 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} 1.8267 & 0.4274 \\ -0.2127 & 0.2953 \end{bmatrix},$$

and $\mathbf{s}_1 = [-0.4759\ 1.1288]^T$.

8.21 Test problem 21

Given a network (4.3) where

$$\mathbf{E} = \begin{bmatrix} 1.1618 & 1.5530 \\ -1.2990 & -0.6542 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} -0.4199 & 0.0067 \\ 1.0924 & -2.5687 \end{bmatrix},$$

and $\mathbf{s}_1 = [0.3535\ 0.5172]^T$.

8.22 Test problem 22

Given a network (4.3) where

$$\mathbf{E} = \begin{bmatrix} 0.0207 & -2.3595 \\ -0.8107 & 0.0597 \end{bmatrix}, \mathbf{W} = \begin{bmatrix} -0.9855 & 0.9763 \\ -0.0560 & -1.2719 \end{bmatrix},$$

and $\mathbf{s}_1 = [-0.2746\ 1.2021]^T$.

8.23 Test problem 23

$\mathbf{W}^1\ (Column(1:5)) =$

$$\begin{bmatrix} 0.1465 & -0.1059 & 0.1594 & 0.2925 & 0.2751 \\ -0.3257 & 0.1052 & 0.1820 & -0.2141 & -0.2609 \\ -0.3717 & 0.1741 & -0.0173 & 0.2439 & -0.2634 \\ 0.0899 & 0.1541 & 0.0439 & 0.3267 & 0.3954 \\ 0.0868 & -0.3327 & -0.3032 & -0.2145 & -0.0482 \\ -0.3874 & -0.0365 & -0.0394 & -0.2085 & -0.1280 \\ -0.3869 & -0.0465 & 0.1727 & -0.3602 & -0.1486 \\ -0.2479 & -0.1174 & 0.3143 & -0.3373 & -0.1079 \\ 0.0695 & -0.2771 & -0.1815 & 0.1127 & -0.0854 \\ -0.3539 & 0.1405 & -0.1962 & -0.2473 & 0.0732 \end{bmatrix}$$

$\mathbf{W}^1$ $(Column(6:10)) =$

$$\begin{bmatrix} -0.3042 & 0.3735 & -0.1231 & -0.0394 & -0.0799 \\ -0.3695 & 0.1319 & -0.2672 & -0.0702 & -0.2410 \\ -0.0331 & 0.2963 & -0.2755 & 0.3213 & 0.1002 \\ 0.2959 & -0.3921 & -0.2471 & -0.3955 & 0.1867 \\ 0.3474 & -0.2904 & -0.0620 & -0.1621 & -0.0993 \\ -0.1884 & 0.2550 & 0.2848 & -0.3607 & -0.3921 \\ -0.2718 & -0.0559 & -0.0078 & 0.1545 & -0.0641 \\ 0.2983 & 0.3123 & 0.2527 & 0.1201 & 0.2029 \\ -0.2097 & 0.1879 & -0.0314 & 0.3864 & 0.2351 \\ 0.1167 & 0.1499 & -0.0341 & 0.0421 & 0.3360 \end{bmatrix}$$

$\mathbf{W}^2$ $(Column(1:5)) =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.2758 & 0 & 0 & 0 & 0 \\ -0.1058 & 0.0966 & 0 & 0 & 0 \\ 0.1850 & -0.2449 & 0.3238 & 0 & 0 \\ 0.0554 & 0.1054 & -0.2125 & 0.0390 & 0 \\ 0.3453 & -0.1318 & 0.1244 & -0.0865 & 0.1019 \\ 0.1593 & -0.0823 & -0.0691 & 0.1242 & 0.2701 \\ -0.0598 & 0.0757 & 0.0526 & 0.1732 & 0.0090 \\ -0.2513 & 0.1605 & 0.3862 & 0.2453 & 0.1629 \\ -0.1077 & -0.2880 & 0.0534 & 0.2584 & 0.1392 \end{bmatrix}$$

$\mathbf{W}^2$ $(Column(6:10)) =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.1027 & 0 & 0 & 0 & 0 \\ 0.2211 & -0.0085 & 0 & 0 & 0 \\ -0.0120 & -0.3083 & 0.1319 & 0 & 0 \\ 0.3996 & 0.3693 & -0.3529 & -0.1118 & 0 \end{bmatrix}$$

$$\mathbf{b} = \mathbf{0}_{(10\times 1)}$$

and $\mathbf{f} = [\text{tanh tanh tanh tanh tanh tanh tanh tanh tanh id}]^T$.

$\mathbf{A} = diag(0,0,0,0,0,0,0,0,0,1)$, $\mathbf{B} = diag(1,1,1,1,1,1,1,1,1,1)$,

$\mathbf{Q}$ $(Column(1:5)) =$

$$\begin{bmatrix} 409.2071 & -60.7473 & 9.4582 & 22.6040 & 104.9155 \\ -60.7473 & 318.8918 & -69.5846 & 146.0832 & -4.6166 \\ 9.4582 & -69.5846 & 478.7037 & -48.1748 & 131.9870 \\ 22.6040 & 146.0832 & -48.1748 & 445.2587 & -5.3633 \\ 104.9155 & -4.6166 & 131.9870 & -5.3633 & 375.6876 \\ 0.7304 & -9.4876 & -66.3357 & 79.2393 & -16.4141 \\ -34.6683 & -62.3295 & 39.5300 & 33.3255 & -38.3926 \\ 66.2225 & 16.2987 & 6.8482 & -21.4796 & -27.1391 \\ -8.1199 & -24.9707 & -150.3311 & -35.8364 & -121.1125 \\ -26.8860 & 94.6296 & -51.7739 & -57.8724 & -16.2736 \end{bmatrix}$$

$\mathbf{Q}\ (Column(6:10)) =$

$$\begin{bmatrix} 0.7304 & -34.6683 & 66.2225 & -8.1199 & -26.8860 \\ -9.4876 & -62.3295 & 16.2987 & -24.9707 & 94.6296 \\ -66.3357 & 39.5300 & 6.8482 & -150.3311 & -51.7739 \\ 79.2393 & 33.3255 & -21.4796 & -35.8364 & -57.8724 \\ -16.4141 & -38.3926 & -27.1391 & -121.1125 & -16.2736 \\ 362.5196 & -43.3246 & -14.7978 & 63.5450 & -86.0946 \\ -43.3246 & 449.1086 & 47.0527 & 125.4525 & -71.5817 \\ -14.7978 & 47.0527 & 346.4056 & -22.5932 & -67.1660 \\ 63.5450 & 125.4525 & -22.5932 & 505.1451 & 70.8772 \\ -86.0946 & -71.5817 & -67.1660 & 70.8772 & 370.0100 \end{bmatrix}$$

$\Lambda = diag(1,1,1,1,1,1,1,1,1,10^{-5})$ and

$\mathbf{T} = 10^3 diag(1, 0.4131, 0.7992, 0.6997, 0.7026, 0.6775, 0.6711, 0.9937, 0.7843, 0.4969)$.

BIBLIOGRAPHY

[1] L. Jin, P. N. Nikiforuk, and M. M. Gupta, "Absolute stability conditions for discrete-time recurrent neural networks," *IEEE Trans. on Neural Networks*, 1994.

[2] K. Tanaka, "An approach to stability criteria of neural network control systems," *IEEE Trans. on Neural Networks*, 2002.

[3] J. A. K. Suykens, J. Vandewalle, and B. L. R. D. Moor, "Nlq theory: Checking and imposing stability of recurrent neural networks for nonlinear modelling," *IEEE Trans. Signal Processing*, 1997.

[4] N. E. Barabanov and D. V. Prokhorov, "Stability analysis of discrete-time recurrent neural networks," *IEEE Trans. on Neural Networks*, 2002.

[5] N. E. Barabanov and D. V. Prokhorov, "A new method for stability analysis of non-linear discrete-time systems," *IEEE Trans. on Automatic Control*, 2003.

[6] M. Liu, "Delayed standard neural network models for control systems," *IEEE Trans. Neural Networks*, 2007.

[7] R. Jafari and M. T. Hagan, "Global stability analysis using the method of reduction of dissipativity domain," *Proceedings of international joint conference on neural networks*, 2011.

[8] K. Kim, E. R. Patron, and R. D. Braatz, "Standard representation and stability analysis of dynamic artificial neural networks: A unified approach," *Proceedings of IEEE symposium computer-aided control system design*, 2011.

[9] O. D. Jesus and M. T. Hagan, "Backpropagation algorithms for a broad class of dynamic networks," *IEEE Trans. Neural Networks*, 2007.

[10] J. A. K. Suykens, J. Vandewalle, and B. L. R. D. Moor, "Lure systems with multilayer perceptron and recurrent neural networks: Absolute stability and dissipativity," *IEEE Trans. Automatic Control*, 1997.

[11] J. C. Williem, "Dissipative dynamical systems part i: General theory," *Arch. Rational Mech. Anal.*, 1972.

[12] J. C. Williem, "Dissipative dynamical systems part ii: Linear systems with quadratic supply rates," *Arch. Rational Mech. Anal.*, 1972.

[13] D. J. Hill and P. J. Moylan, "The stability of nonlinear dissipative systems," *IEEE Trans. Automatic Control*, 1976.

[14] D. J. Hill and P. J. Moylan, "Stability results for nonlinear feedback systems," *Automatica*, 1977.

[15] W. M. Haddad and V. Chellaboina, *Nonlinear dynamical systems and control: A Lyapunov-based approach*. Princeton University Press, 1st ed., 2008.

[16] L. Jin and M. M. Gupta, "Stable dynamic backpropagation learning in recurrent neural networks," *International journal of robust and nonlinear control*, 1999.

[17] J. Horn, O. D. Jesus, and M. T. Hagan, "Spurious valleys in the error surface of recurrent networks - analysis and avoidance," *IEEE Trans. Neural Networks*, 2007.

[18] N. E. Barabanov and D. V. Prokhorov, "Global stability analysis of discrete-time recurrent neural networks," *Proceedings of the American Control Conference*, 2001.

[19] J. Zhao and D. J. Hill, "Dissipativity theory for switched systems," *IEEE Trans. on Automatic Control*, 2008.

[20] W. M. Haddad and Q. Hui, "Dissipativity theory for discontinuous dynamical systems: Basic input, state, and output properties, and finite-time stability of feedback interconnections," *Journal of Nonlinear Analysis: Hybrid systems*, 2009.

[21] V. Chellaboina and W. Haddad, "Stability margins of discrete-time nonlinear nonquadratic optimal regulators," *International Journal of System Science*, 2002.

[22] V. Chellaboina, W. Haddad, and A. Kamath, "A dissipative dynamical system to stability analysis of time delay systems," *International Journal of Robust and Nonlinear Control*, 2004.

[23] C. A. Jacobson, A. M. Stankovic, G. Tadmor, and M. A. Stevens, "Towards a dissipativity framework for power system stabilizer design," *IEEE Trans. on Power Systems*, 1996.

[24] B. Brogliato, R. Lozano, B. Maschke, and O. Egeland, *Dissipative systems analysis and control: Theory and applications*. Springer, 2nd ed., 2007.

[25] S. Hu and J. Wang, "Global stability of a class of discrete-time recurrent neural networks," *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, 2002.

[26] L. Wang and Z. Xu, "Sufficient and necessary conditions for global exponential stability of discrete-time recurrent neural networks," *IEEE Trans. on Circuits and Systems I: Regular Papers*, 2006.

[27] S. Hu and J. Wang, "Global robust stability of a class of discrete-time interval neural networks," *IEEE Trans. Circuits Syst. I: Reg. Papers*, 2006.

[28] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks, vol. 6, pp. 525-533*, 1993.

[29] M. H. Beal, M. T. Hagan, and H. B. Demuth, "Neural network toolbox user's guide," *Matlab*, 2012.

[30] N. P. V. D. AA, H. G. T. Morsche, and R. R. M. Mattheij, “Computation of eigenvalue and eigenvector derivatives for a general complex-valued eigensystem,” *A publication of the International Linear Algebra Society, vol. 16, pp. 300-314*, 2007.

[31] M. T. Hagan, H. B. Demuth, and O. D. Jesus, “An introduction to the use of neural networks in control systems,” *International journal of robust and nonlinear control*, 2002.

[32] R. M. Goodall, “The theory of electromagnetic levitation,” *Physics in Technology*, 1985.

[33] N. E. Barabanov and D. V. Prokhorov, “Two alternative stability criteria for discrete-time rmlp,” *Proceedings of the 41st IEEE conference on Decision and Control*, 2002.

Name: Nam Hoai Nguyen | Date of Degree: December, 2012

Institution: Oklahoma State University | Location: Stillwater, Oklahoma

Title of Study: STABILITY ANALYSIS OF RECURRENT NEURAL NETWORKS USING DISSIPATIVITY

Pages in Study: 95 | Candidate for the Degree of Doctor of Philosophy

Major Field: Electrical Engineering

The purpose of this work is to describe how dissipativity theory can be used for the stability analysis of discrete-time recurrent neural networks and to propose a training algorithm for producing stable networks. Using dissipativity theory, we have found conditions for the globally asymptotic stability of equilibrium points of Layered Digital Dynamic Networks (LDDNs), a very general class of recurrent neural networks. The LDDNs are transformed into a standard interconnected system structure, and a fundamental theorem describing the stability of interconnected dissipative systems is applied. The theorem leads to several new sufficient conditions for the stability of equilibrium points for LDDNs. These conditions are demonstrated on several test problems and compared to previously proposed stability conditions. From these novel stability criteria, we propose a new algorithm to train stable recurrent neural networks. The standard mean square error performance index is modified to include stability criteria. This requires computation of the derivative of the maximum eigenvalue of a matrix with respect to neural network weights. The new training algorithm is tested on two examples of neural network-based model reference control systems, including a magnetic levitation system.

ADVISOR'S APPROVAL: Dr. Martin Hagan

VITA

Nam Hoai Nguyen

Candidate for the Degree of

Doctor of Philosophy

Dissertation: STABILITY ANALYSIS OF RECURRENT NEURAL NETWORKS USING DISSIPATIVITY

Major Field: Electrical Engineering

Biographical:

Education:

Received the B.S. degree from Hanoi University of Science and Technology, Hanoi, Vietnam, 2002, in Electrical Engineering.

Received the M.S. degree from Hanoi University of Science and Technology, Hanoi, Vietnam, 2005, in Electrical Engineering.

Completed the requirements for the degree of Doctor of Philosophy in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2012.

Experience:

Worked for Thai Nguyen University of Technology, Thai Nguyen, Vietnam as a lecturer from 2002 to 2009.