

Thesis for Master's Degree

Backpropagation- and LMI-based Neuro-Adaptive  
Observer for a Class of Uncertain Nonlinear Systems

Yeongho Jeong

School of Mechanical Engineering

Gwangju Institute of Science and Technology

2025

석사학위논문

불확실한 비선형 시스템을 위한 역전파 및 선형  
행렬 부등식 기반 적응 신경망 관측기

정영호

기계공학부

광주과학기술원

2025

# Backpropagation- and LMI-based Neuro-Adaptive Observer for a Class of Uncertain Nonlinear Systems

Advisor: Pilwon Hur

by

Yeongho Jeong

School of Mechanical Engineering

Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science and Technology in the School of Mechanical Engineering

Gwangju, Republic of Korea

June 19, 2025

Approved by



---

Professor Pilwon Hur

Committee Chair

# Backpropagation- and LMI-based Neuro-Adaptive Observer for a Class of Uncertain Nonlinear Systems

Yeongho Jeong

Accepted in partial fulfillment of the requirements for  
the degree of Master of Science and Technology

June 19, 2025

Committee Chair



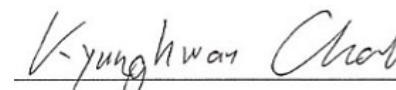
Prof. Pilwon Hur.

Committee Member



Prof. Pyojin Kim.

Committee Member



Prof. Kyunghwan Choi.

Dedicated to my family.

MS/ME Yeongho Jeong. Backpropagation- and LMI-based Neuro-Adaptive Observer for a Class of Uncertain Nonlinear Systems. School of Mechanical Engineering. 2025. 45p. Advisor: Prof. Pilwon Hur.

## Abstract

This thesis proposes a design method for backpropagation (BP)- and linear matrix inequality (LMI)-based neuro-adaptive observer for uncertain nonlinear systems in discrete-time domain. The proposed scheme employs a neural network with a single hidden layer to approximate the unknown uncertainties. To avoid divergence risk associated with discretization, the observer is directly formulated and analyzed in the discrete-time domain. A Lyapunov function is constructed to guarantee the stability of both the linear observer and the weight updates of the neural network. The observer gain is determined by solving the LMI conditions, and the design is simplified by minimizing the number of tuning parameters, using a common gain structure for all vertices. Furthermore, designing an  $H_\infty$  observer can reduce the effect of neural network approximation error and the measurement noise. In conclusion, the proposed method minimizes the number of tuning parameters, accurately estimates the states and uncertainties, ensures LMI-based stability with backpropagation-based updates, suppresses disturbance effects through  $H_\infty$  design, and is directly applicable in the

discrete-time domain. Simulation results indicate that the proposed method successfully tracks the actual states and the lumped nonlinear term and reduce the effects of neural network approximation error and the measurement noise with comparison of the root mean square error (RMSE) values.

©2025

Yeongho Jeong

ALL RIGHTS RESERVED

## 국 문 요 약

본 논문은 이산 시간 도메인에서의 불확실한 비선형 시스템에 대해 역전파 및 선형 행렬 부등식 기반의 신경망 적응 관측기 설계 방법을 제안한다. 제안된 기법은 하나의 은 닉총을 가진 신경망을 이용하여 미지의 불확실성을 근사하며, 이산화 과정에서 발생할 수 있는 발산 위험을 방지하기 위해 관측기를 이산 시간 영역에서 직접 설계 및 해석 한다. 선형 관측기와 신경망의 가중치 업데이트 모두에 대해 안정성을 보장하기 위해 Lyapunov 함수를 구성하였다. 튜닝 파라미터의 수를 최소화하여 설계를 단순화하고, 관측기 이득은 선형 행렬 부등식 조건을 만족하도록 설계되며, 이는 모든 꼭짓점에 대해 공통된 이득 구조를 사용한다. 또한,  $H_{\infty}$  기반의 관측기 설계를 도입함으로써 신경망 근사 오차와 측정 잡음의 영향을 줄일 수 있다. 요약하면, 제안된 방법은 튜닝 파라미터 수를 최소화하고, 상태 및 불확실성을 정확하게 추정하며, 역전파 기반 가중치 업데이트 와 LMI 기반 안정성 조건을 동시에 만족시킨다. 또한,  $H_{\infty}$  설계를 통해 외란의 영향을 억제하고, 이산 시간 영역에서 직접 적용 가능한 구조를 갖는다. 시뮬레이션 결과를 통해, 제안된 방법이 실제 상태와 불확실성을 정확하게 추종하며, 신경망 근사 오차 및 측정 잡음의 영향을 효과적으로 감소시키는 것을 확인하였다. 이는 제곱평균근 오차 값을 기준으로 한 기존 방법과의 비교를 통해 입증된다.

©2025

정영호

ALL RIGHTS RESERVED

# Contents

<b>Abstract (English)</b>	<b>i</b>
<b>Abstract (Korean)</b>	<b>iii</b>
<b>List of Contents</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Conventional State Observer . . . . .	2
1.3 Limitations of Conventional Methods . . . . .	3
1.4 Neuro-Adaptive Observer . . . . .	4
1.5 Conventional Neuro-adaptive Observers . . . . .	7
1.6 Previous Research . . . . .	9
1.6.1 LMI-based Neuro-adaptive Observer [12] . . . . .	9
1.6.2 Nonlinear in Parameters Neural Network [13] . . . . .	10
1.7 Research Objectives . . . . .	11
<b>2 Neuro-Adaptive Observer</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 System Description . . . . .	12
2.3 Neuro-Adaptive Observer . . . . .	16
2.3.1 Observer Design . . . . .	17
2.3.2 Estimation Error Dynamics . . . . .	17
2.3.3 Weight Update Law . . . . .	18
2.4 Stability Analysis . . . . .	19
<b>3 <math>H_\infty</math> Neuro-Adaptive Observer</b>	<b>23</b>
3.1 Introduction . . . . .	23
3.2 $H_\infty$ Neuro-Adaptive Design . . . . .	24

<b>4 Validation</b>	<b>27</b>
4.1 Simulation Setup . . . . .	27
4.2 Speed and Load Torque Estimation . . . . .	28
4.2.1 Tuning Parameter Correlation and Visualization . . . . .	28
4.2.2 Estimation Results with Optimized Tuning Parameters . . . . .	29
4.3 Flux Linkage and VSI Nonlinearity Estimation . . . . .	32
4.3.1 Correlation, Visualization and Estimation Results . . . . .	34
4.4 Discussion . . . . .	38
<b>5 Conclusion and Future Work</b>	<b>39</b>
<b>References</b>	<b>41</b>
<b>Acknowledgements</b>	<b>46</b>

## List of Tables

1.1	Comparison between two existing neural network observer methods . . . . .	11
4.1	Specifications of the PMSM Compressor . . . . .	28
4.2	Tuning parameter ranges used in the simulation . . . . .	29
4.3	Specifications of the PMSM Drive . . . . .	34
4.4	Tuning parameter ranges used in the simulation . . . . .	34

## List of Figures

1.1	Single hidden layer neural network structure.	4
1.2	Various kinds of active function	6
1.3	Discretization can shift eigenvalues outside the unit circle, resulting in instability despite continuous-time stability	8
2.1	Comparison of $\tanh(x)$ and $\tanh(0.25x)$ : Normalization prevents early saturation	16
4.1	SIMULINK implementation for estimation	27
4.2	RMSE performance changes of the tuning parameters in different methods: (a) State estimation, (b) Uncertainty estimation	30
4.3	Results of Rotor Speed and Uncertainties Estimation: (a) rotor speed estimation result, (b) uncertainty estimation result	31
4.4	Current conduction paths for two switching states in a VSI: $S = 1$ (top) and $S = 0$ (bottom), showing the corresponding current flow direction.	33
4.5	Comparison of ideal and actual VSI output voltages under both positive and negative current conditions, illustrating the effects of dead time ( $T_d$ ), switching delays ( $T_{on}, T_{off}$ ), and voltage drops ( $V_D, V_P$ )	33
4.6	RMSE performance changes of the tuning parameters in different methods: (a) $d$ -axis flux linkage estimation result, (b) $q$ -axis flux linkage estimation result	35

4.7 RMSE performance changes of the tuning parameters in different methods: (a) $d$ -axis VSI nonlinearity estimation result, (b) $q$ -axis VSI nonlinearity estimation result . . . . .	36
4.8 Results of Flux Linkage Estimation: (a) $d$ -axis flux linkage estimation result, (b) $q$ -axis flux linkage estimation result . . . . .	37
4.9 Results of VSI Nonlinearity Estimation: (a) $d$ -axis VSI nonlinearity estimation result, (b) $q$ -axis VSI nonlinearity estimation result . . . . .	37

# Chapter 1

## Introduction

### 1.1 Background

In modern control systems, accurate knowledge of the system states is essential for implementing effective feedback control strategies. However, in many practical applications, not all system states can be directly measured due to sensor limitations, high costs, or safety concerns. Instead, output measurements are used to infer the internal dynamics of the system, giving rise to the problem of observer design, which aims to reconstruct the unmeasured states from available inputs and outputs.

In particular, the challenge of simultaneously estimating the system states and the unknown nonlinearities in the discrete-time domain has received considerable attention over the past decades [1], [2], [3]. To address discretization-induced instability, sampled-data observers have been developed that maintain robust performance under fast sampling rates [4]. However, this task becomes significantly more difficult in the presence of modeling uncertainties, external disturbances, and measurement noise—inevitable elements in practical engineering systems such as electric drives, unmanned aerial vehicles, and flexible robotic manipulators. These effects introduce unknown dynamics that cannot be fully captured by nominal mathematical models.

The observer plays a critical role in such contexts by acting as an auxiliary system

that estimates the hidden state variables in real-time, thereby enabling reliable control decisions.

## 1.2 Conventional State Observer

Traditional observer designs—such as the Luenberger observer, the Sliding Mode Observer, or the Kalman filter—typically rely on linearized system models and assume accurate knowledge of system dynamics. However, these assumptions are difficult to satisfy in nonlinear systems where the dynamics may vary significantly with the operating point or environmental conditions. Uncertainties such as friction, actuator saturation, load torque variations, and model parameter mismatches frequently arise in practical systems and often degrade the performance of linear observers. To overcome this, interval observer have been proposed to estimate upper and lower bounds of system states without requiring precise modeling [5].

As a result, the design of observers that are robust to system uncertainty has become an active area of research. To address the challenges, a variety of observer design techniques have been proposed. These include Extended Kalman Filters (EKF) [6], [7], [8], which linearize the nonlinear system around operating points; Sliding Mode Observers (SMO) [9], [10], [11], which offer robustness against matched uncertainties through discontinuous control. In addition, observer designs based on the  $H_\infty$  framework have been proposed for nonlinear systems with time-varying delays and uncertainties [15].

### 1.3 Limitations of Conventional Methods

Despite the long-standing success of conventional observer designs in control theory, including linear structures such as the Luenberger observer, the Extended Kalman Filter, and the Sliding Mode Observer, their applicability becomes limited when dealing with complex nonlinear systems or uncertain environments. These methods often rely on accurate mathematical models and linear system assumptions, making them inadequate for real-world systems characterized by modeling errors, time-varying dynamics, external disturbances, and measurement noise.

Furthermore, many traditional observer schemes are developed in the continuous-time domain without considering the effects of discretization, which can lead to unexpected degradation in performance when implemented digitally. In particular, the mismatch between continuous-time and discrete-time stability conditions may cause estimation divergence, especially under high sampling rates or delay-sensitive applications.

In addition, conventional observers often face challenges in accurately estimating nonlinear components or unmodeled dynamics, as they are typically designed for structured or nominal models. These limitations motivate the development of more robust and adaptive observer frameworks—such as those incorporating learning-based techniques—to ensure reliable performance under uncertain and nonlinear conditions.

To highlight these challenges in detail, this section discusses the key limitations of both classical linear observers and conventional neuro-adaptive observer designs.

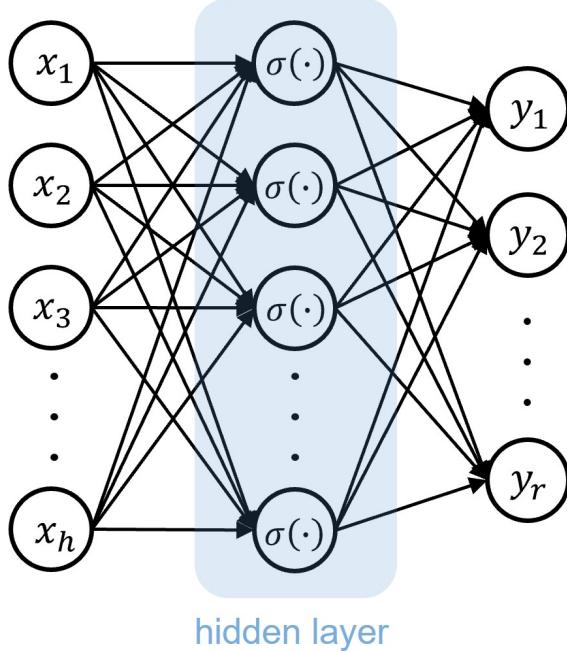


Figure 1.1: Single hidden layer neural network structure.

#### 1.4 Neuro-Adaptive Observer

Neuro-adaptive observers have shown great promise due to their universal function approximation capability and adaptability to unmodeled dynamics. These observer integrate neural network structures with adaptive update laws, allowing them to estimate both system states and unknown uncertainties in real time. A typical neural network structure used in observer design consists of an input layer, one or more hidden layers, and an output layer. In many applications, a single hidden layer is sufficient due to the universal approximation property of neural networks, as illustrated in Fig. 1.1.

Neural networks are known as universal function approximators, meaning they can approximate any continuous function to arbitrary accuracy given sufficient neurons and appropriate activation functions [16], [17]. This property was rigorously established

in early works on neural network-based identification and control [18]. This enables them to capture complex, highly nonlinear relationships that are difficult to model explicitly. For example, dynamic neural networks have been employed to construct robust observers that adapt to online uncertainties [19]. Recent designs also incorporate Lyapunov-based guarantees in neural observers to ensure stability while handling modeling uncertainty [20]. When applied to observer design, this capability allows neural networks to learn and compensate for unknown system dynamics, nonlinearities, and disturbances directly from data, without requiring an accurate mathematical model. This includes neural adaptive interval observers that provide guaranteed bounds on state estimates under uncertainty [21]. This data-driven approach is closely related to adaptive neural control structures for nonlinear systems are well described in [22], where online learning techniques are employed for feedback control.

Furthermore, neural networks can be trained online, making them suitable for adaptive observer structures where the system behavior evolves over time or where precise modeling is infeasible. For example, recurrent neural network observers have been used to track dynamic states under nonlinear uncertainty [23].

The activation functions used in the hidden layer, such as sigmoid, hyperbolic tangent (tanh), Rectified Linear Unit (ReLU), and radial basis function as shown in Fig. 1.2, enable the network to model nonlinear relationships, allowing it to represent nonlinear mappings from inputs (e.g., system states and inputs) to outputs (e.g., estimated disturbances or model uncertainties). When properly integrated with a state observer structure, the neural network acts as a nonlinear estimator that enhances es-

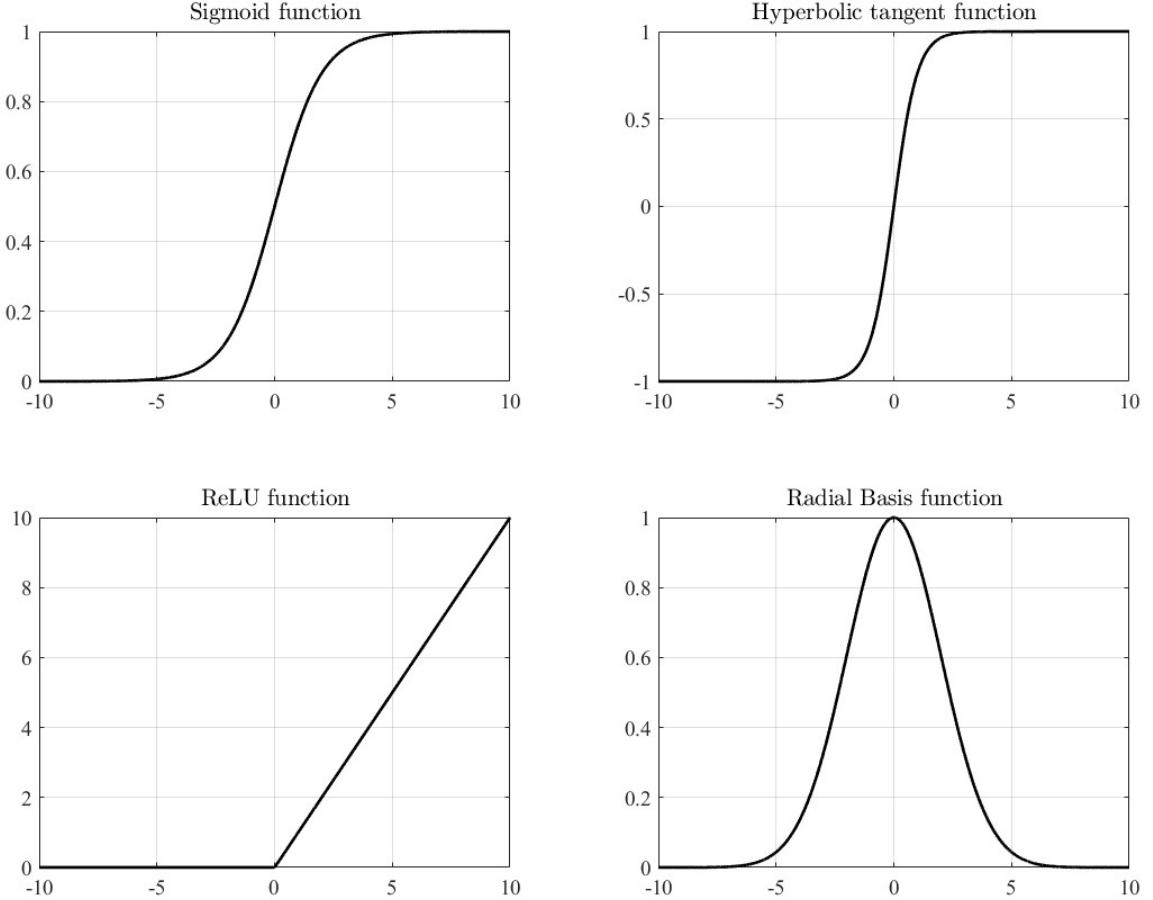


Figure 1.2: Various kinds of active function

timation accuracy and robustness. Prescribed-performance neural observers have also been proposed to ensure bounded estimation error within predefined limits [24].

Due to these advantages, neuro-adaptive observers have shown promising results in a wide range of applications, including robotic manipulators [25], [26], [27], power electronic converters [28], [29], and electric drives [30], [31], [32]. Their flexibility and learning ability make them particularly attractive in scenarios where the system is subject to time-varying dynamics, uncertainties, and measurement noise—conditions

under which traditional observers often struggle. This motivates the integration of neural networks into observer frameworks for nonlinear and uncertain systems, as pursued in this thesis.

### 1.5 Conventional Neuro-adaptive Observers

To overcome these challenges, some neuro-adaptive observer designs have been proposed [12], [13], [14]. Conventional neuro-adaptive observer design methodologies, particularly those applied to nonlinear systems, often exhibit several limitations that hinder their practical implementation. First, these methods typically involve a large number of tuning parameters such as learning rates, gain matrices, error bounds, and stability margins. As a result, the tuning process becomes time-consuming and lacks a systematic guideline, often relying on trial-and-error procedures. This not only increases the development cost but also raises concerns regarding repeatability and reliability in implementation.

Moreover, conventional observer designs are generally developed in the continuous-time domain. When applied to digital or discrete-time systems, such as those implemented on microcontrollers, the discretization process introduces additional challenges. As illustrated in Fig. 1.3, even if a continuous-time system is stable (i.e., all eigenvalues in the left half-plane), its discrete-time counterpart may become unstable if the eigenvalues are shifted outside the unit circle during discretization. This phenomenon can significantly affect observer performance, especially in high-speed systems or those with time delays. In particular, there exists a non-negligible risk of divergence due

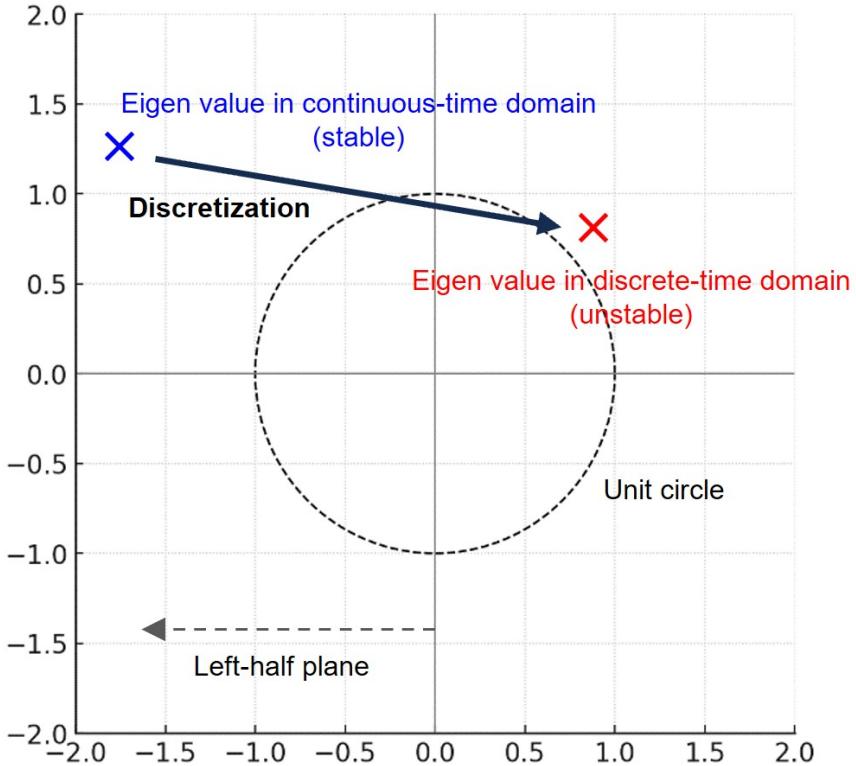


Figure 1.3: Discretization can shift eigenvalues outside the unit circle, resulting in instability despite continuous-time stability

to the mismatch between continuous-time stability conditions and their discrete-time counterparts. This risk is especially significant in systems with high sampling rates or non-negligible time delays.

Another important limitation arises from the nature of estimation error convergence. Many existing observers can only guarantee that the estimation error remains within a bounded region, rather than converging to zero asymptotically. The size of this region depends on the system nonlinearity, observer gains, and disturbance levels. Consequently, under varying operating conditions or unknown disturbances, the estimation accuracy can degrade significantly.

## 1.6 Previous Research

In this section, two major existing approaches to neuro-adaptive observer design are analyzed. The structural formulation, update mechanisms, and theoretical foundations of each method are reviewed. Based on this analysis, the advantages and disadvantages of each approach are discussed, providing the motivation and rationale for the methodology proposed in this thesis.

### 1.6.1 LMI-based Neuro-adaptive Observer [12]

First, the LMI-based neuro-adaptive observer which incorporates gain design via convex optimization under uncertainty is reviewed. An optimal gain set must satisfy a matrix inequality condition derived from Lyapunov stability analysis, typically formulated as a linear matrix inequality (LMI) involving system matrices and tuning parameters. Once this condition is satisfied for all vertices of a polytopic uncertainty set, the observer gain  $L_z = P^{-1}R$  can be computed using standard LMI solvers.

However, one of the main challenges in implementing such observers lies in the presence of numerous tuning parameters—including  $\alpha, \kappa, W_{\max}, a, b, q$ —which significantly complicate the calibration process. The high dimensionality of the parameter space often leads to trial-and-error based tuning, lacking systematic guidelines and reducing practical applicability.

Furthermore, the simultaneous gain design for both state and weight estimation introduces additional complexity. In particular, the coupling between the state observer and learning dynamics of the neural network makes it difficult to achieve accurate

nonlinear approximation while maintaining overall system stability. This trade-off between estimation accuracy and robust convergence must be carefully addressed in the observer design process. Table. 1.1 summarizes the key features of the two existing neuro-adaptive observer methods discussed earlier, providing a concise comparison of their structures, stability conditions, and implementation complexity.

### 1.6.2 Nonlinear in Parameters Neural Network [13]

This subsection reviews a stable neural network-based observer for general multi-variable nonlinear systems in continuous-time domain. The learning rule for the neural network is based on the modified backpropagation (BP) method which includes the  $\epsilon$ -modification term to guarantee robustness simply. Lyapunov-based stability analysis shows that the observer is stable. However, for the derivative of the Lyapunov function to remain negative semi-definite, the estimation error must stay within a certain bounded region (i.e.,  $\|\tilde{x}\| > b$ ). The radius  $b$  of the error ball depends on several factors including the Lyapunov matrix norm  $\|P\|$ , the minimum eigenvalue of  $Q$ , the neural network error  $\bar{w}$ , the learning rates  $\eta_1, \eta_2$ , the damping factor  $\rho_1, \rho_2$ , and several other constants. A larger value of  $b$  implies that the system requires a larger error region before stability can be enforced, which may lead to degraded performance or instability, especially when operating points vary or external disturbances increase. Furthermore, this method is analyzed in continuous-time domain, which can lead to divergence in the discretization process.

Table 1.1: Comparison between two existing neural network observer methods

Feature	LMI-based [12]	BP-based [13]
Weight update	$\dot{\hat{w}} = -KC\tilde{x}$	$\dot{\hat{W}} = -\eta \frac{\partial J}{\partial \hat{W}} - \rho \ \tilde{y}\  \hat{W}$
Stability method	Lyapunov + LMI conditions	Lyapunov
Error condition	Global $\tilde{z}$ boundedness via LMIs	$\ \tilde{x}\  > b$ for $\dot{V} < 0$
Observer gain	$L_z = P^{-1}R$ from LMI	Fixed $L$ , tuned empirically
Weight update	Gain on state error	Modified backpropagation
Tuning complexity	High (many parameters: $\alpha, \kappa, W_{\max}$ , etc.)	Relatively simple

## 1.7 Research Objectives

To overcome the aforementioned challenges, this study proposes a neuro-adaptive observer with backpropagation-based weight update law and LMI-based robust gain selection. The primary objective of existing methods while enhancing practical applicability. First, the proposed method aims to simplify the tuning process by minimizing the number of design parameters. This reduces the reliance on manual calibration and improves ease of implementation in diverse operating conditions. Second, accurate estimation of unknown uncertainties is pursued by adaptively adjusting the learning rate and convergence speed. This ensures the observer can maintain high accuracy even under rapidly changing system dynamics. Third, the observer is designed to guarantee stability and robustness against disturbance, model uncertainty, and measurement noise. This is achieved through an LMI-based observer gain design framework. Finally, the entire observer structure is formulated directly in the discrete-time domain, eliminating the instability risks associated with discretizing continuous-time designs. This makes the proposed method inherently suitable for digital control systems and real-time implementation.

# Chapter 2

## Neuro-Adaptive Observer

### 2.1 Introduction

In this chapter, a neuro-adaptive observer designed in discrete-time nonlinear systems with unknown disturbances or dynamics uncertainties is proposed. This approach aims to estimate both the system states and the nonlinear term by incorporating a neural network into a conventional observer structure. The neural network serves as a nonlinear estimator that adaptively learns the unknown component from input-output data.

### 2.2 System Description

Consider a nonlinear system in discrete-time domain described by the following form:

$$\begin{aligned}x_{k+1} &= f_k(x_k, u_k), \\y_k &= Cx_k + n_k,\end{aligned}\tag{2.1}$$

where  $x \in \mathbb{R}^n$  is the state vector,  $u \in \mathbb{R}^m$  is the known input vector,  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is the nonlinear function that describes the system dynamics, depending on the current state and input. The output vector is given by  $y \in \mathbb{R}^p$ , and  $n \in \mathbb{R}^p$  denotes the measurement noise vector. The matrix  $C \in \mathbb{R}^{p \times n}$  is the known output matrix, and the

index  $k$  indicates the discrete-time step.

The equation 2.1 can be rewritten in the form

$$x_{k+1} = Ax_k + Bu_k + Fg_k(x_k, u_k), \quad (2.2)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$  and  $F \in \mathbb{R}^{n \times r}$  are the known constant matrices, and  $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  captures the unknown or nonlinear part of the system dynamics. The matrix  $A$  is a Schur matrix (i.e., all eigenvalues of  $A$  satisfy  $\|\lambda_i\| < 1$  for all  $i$ ).

To approximate unknown nonlinearities in the system dynamics, a single-hidden-layer neural network structure is employed. The output of the neural network is defined as:

$$g_k(x_k, u_k) = W_k \sigma(z_k) + \varepsilon_k(z_k), \quad (2.3)$$

where  $W \in \mathbb{R}^{r \times h}$  is the output weight matrix,  $\sigma(\cdot)$  is a nonlinear activation function applied element-wise,  $z \in \mathbb{R}^h$  is the input to the hidden layer, and the neural network approximation error  $\varepsilon \in \mathbb{R}^r$ . The neural network input vector  $z_k$  is typically constructed as a function of the current state and input, i.e.,  $z_k = h(x_k, u_k)$  where  $h(\cdot)$  is a user-defined deterministic mapping. A common choice for  $z_k$  includes concatenated terms such as  $z_k = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix}^T$ .

Assuming that the activation function  $\sigma(\cdot)$  is Lipschitz continuous and bounded

such that

$$\|\sigma(z_1) - \sigma(z_2)\| \leq L_\sigma \|z_1 - z_2\|, \quad \forall z_1, z_2 \in \mathbb{R}^h \quad (2.4a)$$

$$-\infty < \sigma_{min} \leq \sigma(\cdot) \leq \sigma_{max} < \infty. \quad (2.4b)$$

with a constant  $L_\sigma > 0$ . This structure allows the network to approximate a wide range of continuous nonlinear functions due to the universal approximation property of  $\sigma(\cdot)$ .

However, in practice, when the magnitude of  $z$  becomes excessively large, the activation function  $\sigma(z)$  tends to saturate—i.e., its output approaches a constant value regardless of further increases in  $z$ . In this saturated region, the neural network becomes insensitive to input variations, reducing its ability to approximate nonlinearities accurately.

To address this limitation, a normalization matrix  $N \in \mathbb{R}^{h \times h}$  is introduced to scale the input vector  $z$ , resulting in the modified structure:

$$g_k(x_k, u_k) = W_k \sigma(N z_k) + \varepsilon_k(z_k). \quad (2.5)$$

The matrix  $N$  is typically selected as a diagonal matrix with positive entries less than or equal to one, i.e.,  $N = \text{diag}(N_1, N_2, \dots, N_h)$ , where  $0 < N_i \leq 1$ ,  $i \in \{1, 2, \dots, h\}$ . This normalization ensures that the effective input to the activation function remains within a bounded region where  $\sigma(\cdot)$  is most responsive—typically around the origin for functions such as tanh or sigmoid. As shown in Fig. 2.1, when the input to the activation function is not normalized,  $\tanh(x)$  rapidly saturates outside

the narrow region near the origin, resulting in near-constant output values and vanishing gradients. On the other hand, scaling the input using a normalization factor (e.g.,  $\tanh(0.25x)$ ) expands the effective linear region of the activation function, keeping it sensitive to input variations and thereby enhancing the approximation and learning capability of the neural network. As a result, the activation outputs avoid saturation and the network maintains its learning capability and nonlinear approximation power.

Even with normalization, if a portion of the activation output  $\sigma(Nz_k)$  becomes zero, the neural network's ability to approximate nonlinearities may become momentarily unstable. In particular, when certain elements of  $\sigma(Nz_k)$  vanish due to the shape of the activation function, the corresponding directions in the approximation space become inactive. As a result, the estimated nonlinear term may lose sensitivity to changes in the input, leading to a temporary degradation in the accuracy of nonlinearity estimation.

This phenomenon does not necessarily violate the stability condition, but it can increase the residual approximation error, which may reduce the effectiveness of compensation in the observer dynamics. Therefore, care must be taken to ensure that the inputs to the activation function are not driven into regions that result in zero or near-zero activation outputs.

To overcome this problem, various methods are used to solve the problem of terms becoming zero, such as adding a positive bias to shift the activation function away from zero.

These mechanisms improve both the numerical stability of the observer and the accuracy of the nonlinear function approximation, especially in online or adaptive learning

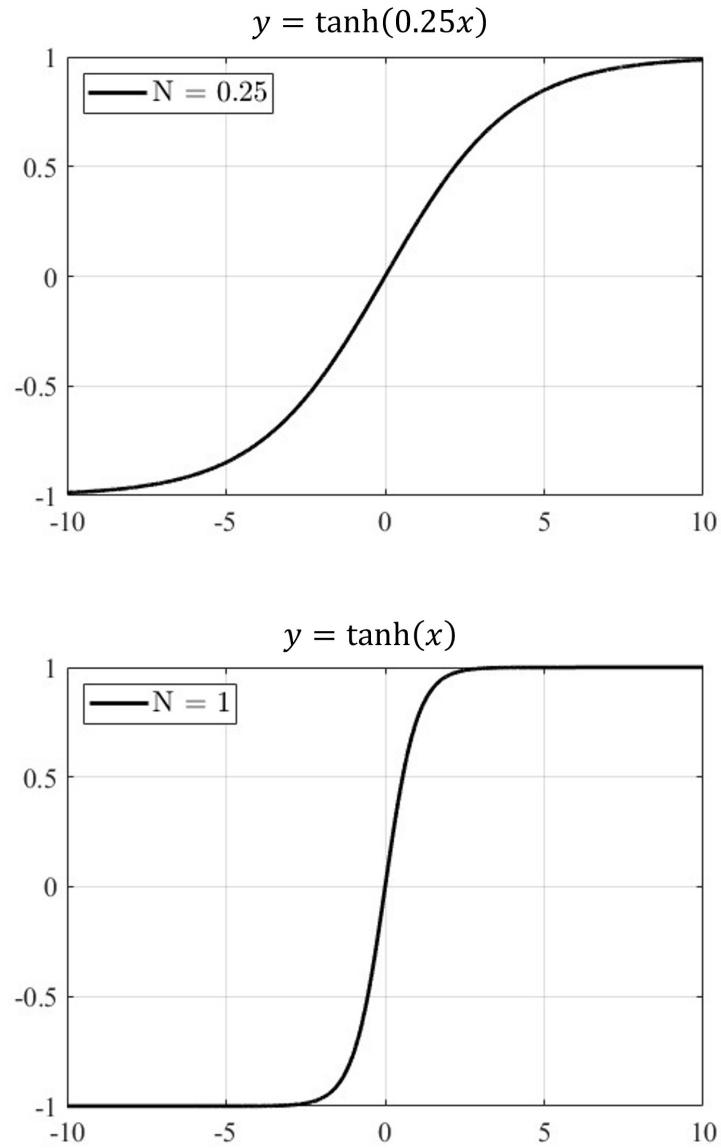


Figure 2.1: Comparison of  $\tanh(x)$  and  $\tanh(0.25x)$ : Normalization prevents early saturation.

scenarios where input magnitudes may vary significantly over time.

### 2.3 Neuro-Adaptive Observer

The objective is to design an observer that can estimate both the system state  $x_k$  and the nonlinear uncertainties  $g_k(x_k, u_k)$  in system 2.2 and approximated NN struc-

ture 2.5 using input and output of the system. To address this, a neural network is embedded into the observer structure to estimate  $g(x_k, u_k)$  adaptively. The neural network is trained online to capture the nonlinear behavior.

### 2.3.1 Observer Design

The proposed neuro-adaptive observer is designed with the following form:

$$\begin{aligned}\hat{x}_{k+1} &= A\hat{x}_k + Bu_k + F\hat{g}_k(\hat{x}_k, u_k) + L(y_k - \hat{y}_k), \\ \hat{y}_k &= C\hat{x}_k, \\ \hat{g}_k(\hat{x}_k, u_k) &= \hat{W}_k\sigma(N\hat{z}_k),\end{aligned}\tag{2.6}$$

with the estimated state vector  $\hat{x}$ , estimated output vector  $\hat{y}$ , the observer gain  $L \in \mathbb{R}^{n \times p}$  and the uncertainty term  $\hat{g}(\hat{x}_k, u_k)$  where  $\hat{W}$  is updated weight matrix by an adaptive law to approximate the ideal weight matrix  $W$ . The input vector of the neural network, denoted as  $\hat{z}_k \in \mathbb{R}^h$  is constructed from the estimated state and other related elements as  $\hat{z}_k = h(\hat{x}_k, u_k)$  where  $h(\cdot)$  is a deterministic mapping. The specific choice of  $\hat{z}_k$  influences the approximation quality of the neural network.

### 2.3.2 Estimation Error Dynamics

The estimation error dynamics is obtained by subtracting 2.2 and 2.5 by 2.6 as follows:

$$\begin{aligned}\tilde{x}_{k+1} &= (A - LC)\tilde{x}_k + F\tilde{W}_k\sigma(N\hat{z}_k) + \phi_k, \\ \tilde{y}_k &= C\tilde{x}_k + n_k,\end{aligned}\tag{2.7}$$

where the estimation error for state, output, and weight  $\tilde{x}_k = x_k - \hat{x}_k$ ,  $\tilde{y}_k = y_k - \hat{y}_k$ ,  $\tilde{W}_k = W_k - \hat{W}_k$  and  $\phi_k = F\hat{W}_k(\sigma(Nz_k) - \sigma(N\hat{z}_k)) + F\varepsilon_k - Ln_k$ . To facilitate stability analysis, the error term  $F\tilde{W}_k\sigma(N\hat{z}_k)$  in the error dynamics (2.7) is reformulated from matrix to vector form as  $F\hat{\Sigma}_k\tilde{w}_k$  with

$$\begin{aligned}\hat{\Sigma}_k &= I_r \otimes \sigma(N\hat{z}_k)^T, \\ \tilde{w}_k &= \begin{bmatrix} \tilde{w}_1^T \\ \tilde{w}_2^T \\ \vdots \\ \tilde{w}_r^T \end{bmatrix}_k,\end{aligned}\tag{2.8}$$

where  $\otimes$  denotes the Kronecker product,  $I_r$  is a  $r \times r$  identity matrix,  $\tilde{w}_i$  is the row vector for  $i$ th row in the weight matrix  $\tilde{W}$ .

### 2.3.3 Weight Update Law

The estimation error  $\tilde{x}_k$  converges to zero with the adaptive law that makes the term  $F\hat{\Sigma}_k\tilde{w}_k$  to zero. Such adaptive law is proposed based on the backpropagation as follows:

$$\hat{w}_{k+1} = \hat{w}_k - T_s \eta \frac{\partial J_k}{\partial \hat{w}_k},\tag{2.9}$$

with loss function  $J_k = \frac{1}{2}\tilde{y}_k^T\tilde{y}_k$ , the sampling time  $T_s$ , and the learning rate  $\eta$ . Using the chain rule to solve (2.9), the adaptive law is represented as follows:

$$\hat{w}_{k+1} = \hat{w}_k + T_s \eta \hat{\Sigma}_k^T F^T (I_n - A)^{-T} C^T \tilde{y}_k.\tag{2.10}$$

To represent the error dynamics for weight vector  $w$ , the static approximation method (i.e.,  $x_{k+1} = x_k$ ) is used. This assumption simplifies the estimation error dynamics and enables the Lyapunov-based stability conditions. Although the neural network weights are updated online, the static approximation is introduced to facilitate Lyapunov-based stability analysis. This assumption is reasonable when the learning rate is sufficiently small, making the change in weights negligible over each time step. The actual difference between  $w_{k+1}$  and  $w_k$  can be treated as a bounded perturbation in the error dynamics, allowing the use of robust analysis techniques such as  $H_\infty$  observer design.

Finally, the error dynamics of the weight vector is represented as follows:

$$\tilde{w}_{k+1} = \tilde{w}_k - T_s \eta \hat{\Sigma}_k^T F^T (I_n - A)^{-T} C^T C \tilde{x}_k + \varphi_k, \quad (2.11)$$

with  $\varphi_k = -T_s \eta \hat{\Sigma}_k^T F^T (I_n - A)^{-T} C^T n_k$ .

## 2.4 Stability Analysis

The error dynamics (2.7) and (2.11) is reformulated the augmented state as follows:

$$\underbrace{\begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{w}_{k+1} \end{bmatrix}}_{\xi_{k+1}} = \underbrace{\begin{bmatrix} A - LC & F \hat{\Sigma}_k \\ -T_s \eta \hat{\Sigma}_k^T F^T (I_n - A)^{-T} C^T C & I_h \end{bmatrix}}_{\mathcal{A}(\hat{\Sigma}_k)} \underbrace{\begin{bmatrix} \tilde{x}_k \\ \tilde{w}_k \end{bmatrix}}_{\xi_k} + \underbrace{\begin{bmatrix} \phi_k \\ \varphi_k \end{bmatrix}}_{\delta_k}, \quad (2.12)$$

with  $I_h$  is the  $h \times h$  identity matrix.

The time-varying matrix  $\mathcal{A}(\hat{\Sigma}_k)$  can be represented with affine parameter dependent

model as follows:

$$\mathcal{A}(\hat{\Sigma}_k) = \mathcal{A}_0 + \mathcal{A}_1 \Omega_k + \Omega_k^T \mathcal{A}_2, \quad (2.13)$$

where  $\Omega_k = \begin{bmatrix} 0 & 0 \\ 0 & \hat{\Sigma}_k \end{bmatrix}$ ,  $\mathcal{A}_0$  is the matrix with constant components of  $\mathcal{A}(\hat{\Sigma}_k)$ ,  $\mathcal{A}_1$  and  $\mathcal{A}_2$  are the matrices where all elements are zero except for the matrix corresponding to  $\hat{\Sigma}_k$  or  $\hat{\Sigma}_k^T$ . By the boundedness of  $\sigma(\cdot)$  defined in (2.4b), the matrix  $\hat{\Sigma}_k$  is element-wise bounded as well. Since  $\mathcal{A}(\cdot)$  is affine with respect to  $\hat{\Sigma}_k$ , stability of the system can be guaranteed over the entire bounded set if the stability condition is satisfied at all vertices of the uncertainty set [33]. Using this approach, the observer gain  $L$  can be designed to satisfy  $2^h$  LMI conditions corresponding to the vertices of the uncertainty set.

Theorem 1 presents a condition that guarantees the stability of the error dynamics (2.12).

*Theorem 1:* The estimation error dynamics (2.12) is stable if there exist matrices  $P = P^T = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} > 0$  and  $R$  of appropriate dimensions such that the following inequality is feasible:

$$\begin{aligned}
& \begin{bmatrix} \Gamma + (2\alpha - 1)P & \mathcal{A}(\hat{\Sigma}_k)^T P & \Pi^T \\ P\mathcal{A}(\hat{\Sigma}_k) & P & 0 \\ \Pi & 0 & -P_1 \end{bmatrix} \leq 0, \\
\Gamma = & \begin{bmatrix} A^T P_1 A - A^T R C - C^T R^T A + \mathcal{A}_{21}^T P_2 \mathcal{A}_{21} \\ \mathcal{A}_{12}^T P_1 A - \mathcal{A}_{12}^T R C + P_2 \mathcal{A}_{21} \\ A^T P_1 \mathcal{A}_{12} - C^T R^T \mathcal{A}_{12} + \mathcal{A}_{21}^T P_2 \\ \mathcal{A}_{12}^T P_1 \mathcal{A}_{12} + P_2 \end{bmatrix}, \quad (2.14) \\
\Pi = & \begin{bmatrix} RC & 0 \end{bmatrix}, \\
\mathcal{A}_{12} = & F \hat{\Sigma}_k, \\
\mathcal{A}_{21} = & -T_s \eta \hat{\Sigma}_k^T F^T (I_n - A)^{-T} C^T C.
\end{aligned}$$

When the inequality (2.14) is feasible, the linear observer gain  $L$  is given by  $L = P_1^{-1}R$  and ensures that  $\lim_{k \rightarrow \infty} \hat{x}_k = x_k$  exponentially.

*Proof:* To verify the stability of the proposed observer, the Lyapunov function is defined as follows:

$$V_k = \xi_k^T P \xi_k. \quad (2.15)$$

Now, it remains to prove that  $\Delta V = V_{k+1} - V_k \leq -2\alpha V_k$  for all  $\xi_k \neq 0$ :

$$\Delta V = \xi_k^T (\mathcal{A}^T P \mathcal{A} - P) \xi_k + \xi_k^T \mathcal{A}^T P \delta_k + \delta_k^T P \mathcal{A} \xi_k + \delta_k^T P \delta_k. \quad (2.16)$$

In order to satisfy the condition  $\Delta V \leq -2\alpha V_k$ , the following inequality should be

satisfied:

$$\begin{bmatrix} \xi_k \\ \delta_k \end{bmatrix}^T \mathcal{M} \begin{bmatrix} \xi_k \\ \delta_k \end{bmatrix} \leq 0, \quad (2.17)$$

where  $\mathcal{M} = \begin{bmatrix} \mathcal{A}^T P \mathcal{A} + (2\alpha - 1)P & \mathcal{A}^T P \\ P \mathcal{A} & P \end{bmatrix}$  and  $\alpha$  is a fixed positive scalar that determines the convergence speed of the system (2.12).

Then, using the Schur complement, the notation  $R = P_1 L$  and  $P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}$ , the inequality (2.14) can be derived from the condition  $\mathcal{M} \leq 0$ . Therefore,  $\Delta V \leq -2\alpha V_k$  for all  $\xi_k \neq 0$ .  $\square$

Although the effects of noise and modeling errors have been partially considered in the design of the neuro-adaptive observer, its performance may still degrade under severe disturbances or large approximation errors. Therefore, to enhance robustness, the next chapter introduces an  $H_\infty$ -based observer design that explicitly minimizes the worst-case estimation error.

# Chapter 3

## $H_\infty$ Neuro-Adaptive Observer

### 3.1 Introduction

In this chapter, a robust  $H_\infty$  neuro-adaptive observer is proposed to reduce the effects of residual terms  $\delta_k$ , which include the neural network approximation error  $\varepsilon_k$  and the measurement noise  $n_k$ . These disturbances can adversely affect the estimation performance; hence, reducing their impact is essential for improving observer accuracy. To address such residual uncertainties, various robust observer design methods have been investigated, including Sliding Mode Observers, Extended Kalman Filters, and  $H_\infty$  observers. Among these, the  $H_\infty$ -based design approach is adopted in this chapter due to its ability to minimize the worst-case estimation error in the presence of bounded disturbances.

Unlike Extended Kalman Filters, which rely heavily on precise system models and may diverge under severe nonlinearities or noise, or Sliding Mode Observers, which may suffer from chattering phenomena due to their discontinuous structure, the  $H_\infty$  neuro-adaptive observer provides guaranteed performance by explicitly optimizing for the worst-case disturbance scenario. This ensures strong robustness even under significant model uncertainties or external noise.

### 3.2 $H_\infty$ Neuro-Adaptive Design

The observer gain with  $H_\infty$  gain  $\gamma$  is determined as follows:

$$\sup_{0 < \|\delta_k\|_2 < \infty} \frac{\|\xi_k\|_2}{\|\delta_k\|_2} \leq \gamma, \quad (3.1)$$

where  $\gamma > 0$  is an upper bound of  $H_\infty$  performance. A smaller  $\gamma$  implies better attenuation of the effect of undesirable terms. The sufficient condition for satisfying the inequality is provided in Theorem 2.

*Theorem 2:* For a positive scalar  $\gamma$ , the estimation error (2.12) converges if there exist matrix  $P = P^T = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} > 0$  and  $R$  of appropriate dimensions such that the following inequality is feasible:

$$\begin{bmatrix} \Gamma + (2\alpha - 1)P + I & \mathcal{A}(\hat{\Sigma}_k)^T P & \Pi^T \\ P\mathcal{A}(\hat{\Sigma}_k) & P - \gamma^2 I & 0 \\ \Pi & 0 & -P_1 \end{bmatrix} \leq 0. \quad (3.2)$$

When the inequality (3.2) is feasible, the linear observer gain  $L$  is given by  $L = P_1^{-1}R$  and ensures that  $\lim_{k \rightarrow \infty} \hat{x}_k = x_k$  exponentially and the  $H_\infty$  observer is designed to suppress the worst-case energy gain from the residual term  $\delta_k$  to the estimation error  $\xi_k$ , ensuring robustness under bounded uncertainties.

*Proof:* The  $H_\infty$  performance condition (3.1) is satisfied if the following inequality holds.

$$V_H = \Delta V + \|\xi_k\|^2 - \gamma^2 \|\delta_k\|^2 \leq 0. \quad (3.3)$$

The inequality (3.3) becomes

$$\xi_k^T (\mathcal{A}^T P \mathcal{A} - P + I) \xi_k + \xi_k^T \mathcal{A}^T P \delta_k + \delta_k^T P \mathcal{A} \xi_k + \delta_k^T (P - \gamma^2 I) \delta_k \leq 0. \quad (3.4)$$

The condition  $V_H \leq -2\alpha V_k$  can be represented in matrix form:

$$\begin{bmatrix} \xi_k \\ \delta_k \end{bmatrix}^T \bar{\mathcal{M}} \begin{bmatrix} \xi_k \\ \delta_k \end{bmatrix} \leq 0, \quad (3.5)$$

$$\text{where } \bar{\mathcal{M}} = \begin{bmatrix} \mathcal{A}^T P \mathcal{A} + (2\alpha - 1)P + I & \mathcal{A}^T P \\ P \mathcal{A} & P - \gamma^2 I \end{bmatrix}.$$

Then, using the Schur complement, the notation  $R = P_1 L$  and  $P = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}$ , the inequality (3.2) can be derived from the condition  $\bar{\mathcal{M}} \leq 0$ . Therefore,  $V_H \leq -2\alpha V_k$  for all  $\xi_k \neq 0$ .  $\square$

Given the LMI condition in Theorem 2, the design of observer gain  $L$  can be achieved by convex optimization. In order to minimize the  $H_\infty$  gain in inequality (3.1), the following optimization problem using the LMIs,

$$\min_{P,R} \quad \bar{\gamma} \quad (3.6)$$

subject to  $P > 0, \bar{\gamma} > 0$ , and (3.2),

where  $\bar{\gamma} = \gamma^2$ .

Unlike empirical tuning methods, this approach provides a systematic framework

for selecting observer gains that ensure both stability and guaranteed worst-case performance. Since the  $H_\infty$  gain  $\gamma$  reflects the maximum amplification from disturbance to estimation error, minimizing it enhances robustness especially in high-noise or highly uncertain environments.

# Chapter 4

## Validation

### 4.1 Simulation Setup

This chapter presents the validation results of the proposed neuro-adaptive observer through two case studies. The first section focuses on the estimation performance of rotor speed and nonlinearity which includes load torque and parameter uncertainty in a PMSM compressor system. The second section demonstrates the effectiveness of the observer in estimating flux linkage and voltage source inverter (VSI) nonlinearity, highlighting its applicability to practical motor drive systems. To evaluate the performance of the proposed method, it will be compared with the methods presented in Section 1.6, [12] and [13]. The proposed neuro-adaptive observer was validated in MATLAB/SIMULINK simulation as shown in Fig. 4.1.

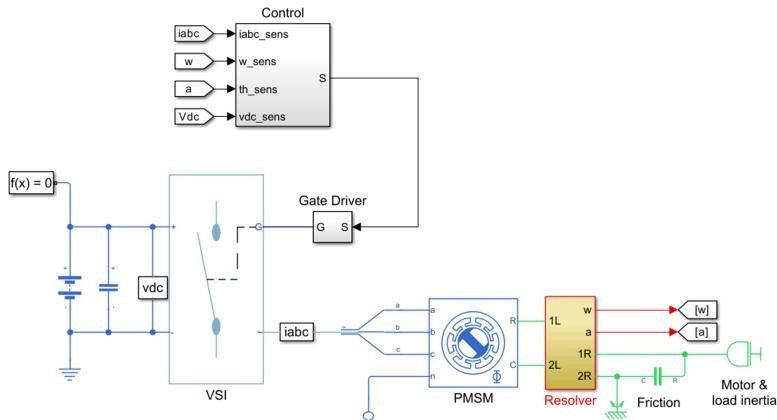


Figure 4.1: SIMULINK implementation for estimation

## 4.2 Speed and Load Torque Estimation

This section presents validation results focusing on the estimation of rotor speed and system uncertainties. The estimation process is based on the well-known mechanical dynamics equation of the PMSM as follows:

$$\dot{\omega}_m(t) = -\frac{B_m}{J_m}\omega_m(t) + \frac{1}{J_m}T_e(t) - \frac{1}{J_m}T_L(t), \quad (4.1)$$

where  $\omega_m$  is the rotational speed of the PMSM,  $T_e$  is the torque of the PMSM, and  $T_L$  is the load torque of the PMSM, respectively. The PMSM parameters used in the simulation are listed in Table 4.1 and the measurement noise is  $n_k \sim \mathcal{N}(0, 0.01^2)$ . The rotor speed is controlled to maintain 1800 RPM, while the load torque is applied in proportion to the rotor angle.

Table 4.1: Specifications of the PMSM Compressor

DC-link voltage ( $V_{dc}$ )	150 V
Sampling time ( $T_s$ )	100 $\mu$ s
Number of pole pairs ( $P$ )	4
Stator resistance ( $R_s$ )	1.1 $\Omega$
d-axis Inductance ( $L_d$ )	8 mH
q-axis Inductance ( $L_q$ )	8 mH
Inertia ( $J_m$ )	$4.44 \times 10^{-4}$ kg·m <sup>2</sup>
Friction ( $B_m$ )	0.005 Nm·s/rad
Known inertia ( $\hat{J}_m$ )	$10^{-4}$ kg·m <sup>2</sup>
Known friction ( $\hat{B}_m$ )	0.008 Nm·s/rad

### 4.2.1 Tuning Parameter Correlation and Visualization

To evaluate the sensitivity of the observer performance with respect to the tuning parameters, a correlation analysis is performed between the estimation error and each

parameter. The objective is to identify the parameters that have a significant impact on the estimation accuracy. Based on the correlation results, two most influential parameters are selected for each observer design method. These parameters are visualized using contour plots, as shown in Fig. 4.2, which display the distribution of the root mean square error (RMSE) over the parameter space.

The tuning parameters are selected as shown in Table 4.2.

Table 4.2: Tuning parameter ranges used in the simulation

Method	Parameter	Minimum value	Maximum value
Proposed method	$\eta$	$5 \times 10^{-1}$	3.3
	$\alpha$	$10^{-2}$	$1.5 \times 10^{-1}$
LMI-based [12]	$\kappa$	$3 \times 10^{-1}$	$5.8 \times 10^{-1}$
	$\alpha$	$5 \times 10^1$	$4 \times 10^2$
BP-based [13]	$\eta_1$	$10^5$	$1.2 \times 10^7$
	$\eta_2$	$10^2$	$7.8 \times 10^6$

As shown in Fig. 4.2, subplots (a) and (b) represent the state and uncertainty estimation error respectively for each parameter. This visualization aims to identify parameter regions yielding low estimation error, offering insights into the robustness of each method to parameter variations. For each method, the selected pair of tuning parameters are varied within a specified range, and the resulting RMSE is visualized using color maps.

#### 4.2.2 Estimation Results with Optimized Tuning Parameters

To validate the effectiveness of the selected tuning parameters, simulation results are presented using the parameter sets that yielded the lowest RMSE in the previous correlation analysis as shown in Fig. 4.3.

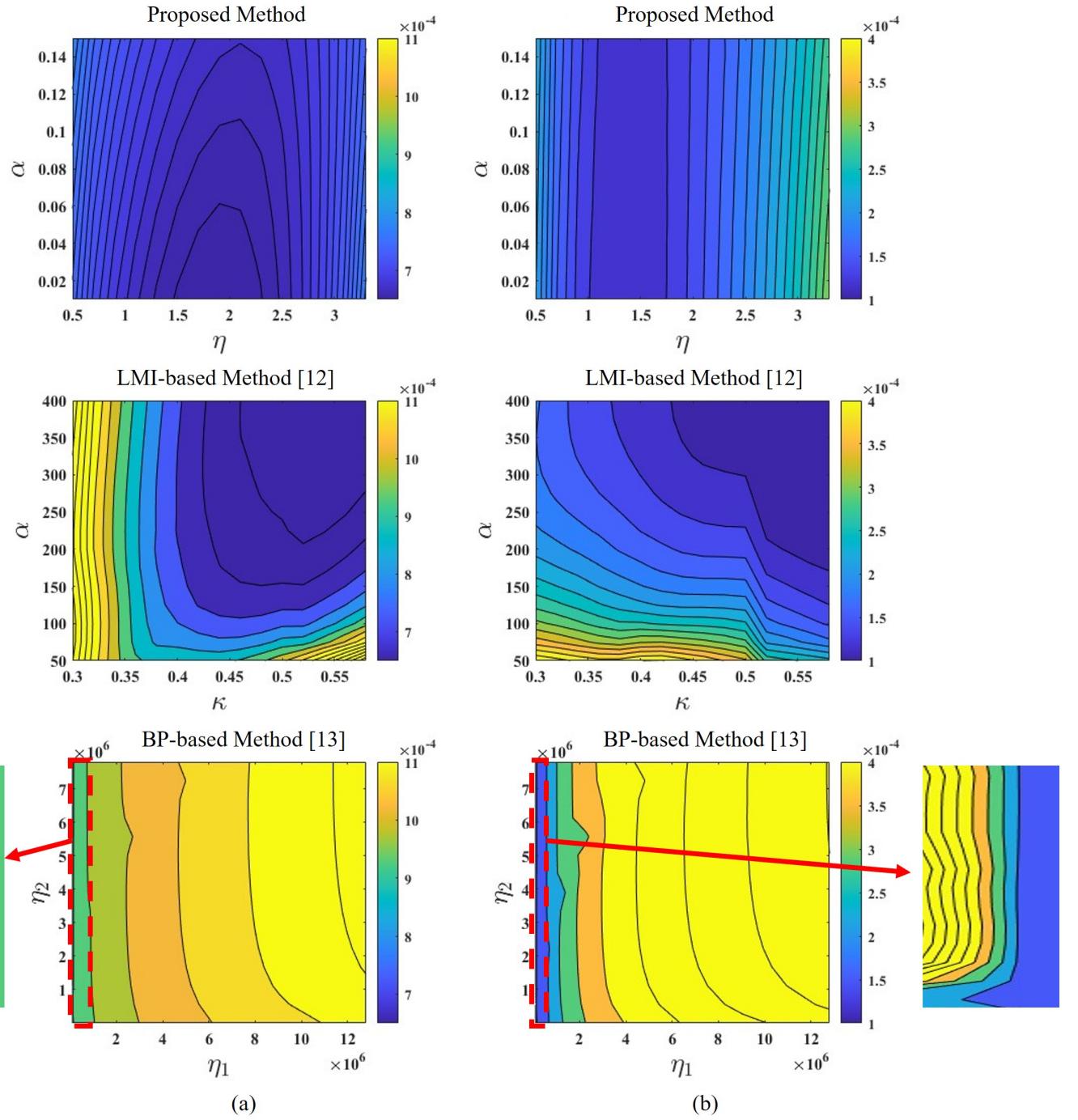


Figure 4.2: RMSE performance changes of the tuning parameters in different methods:  
(a) State estimation, (b) Uncertainty estimation

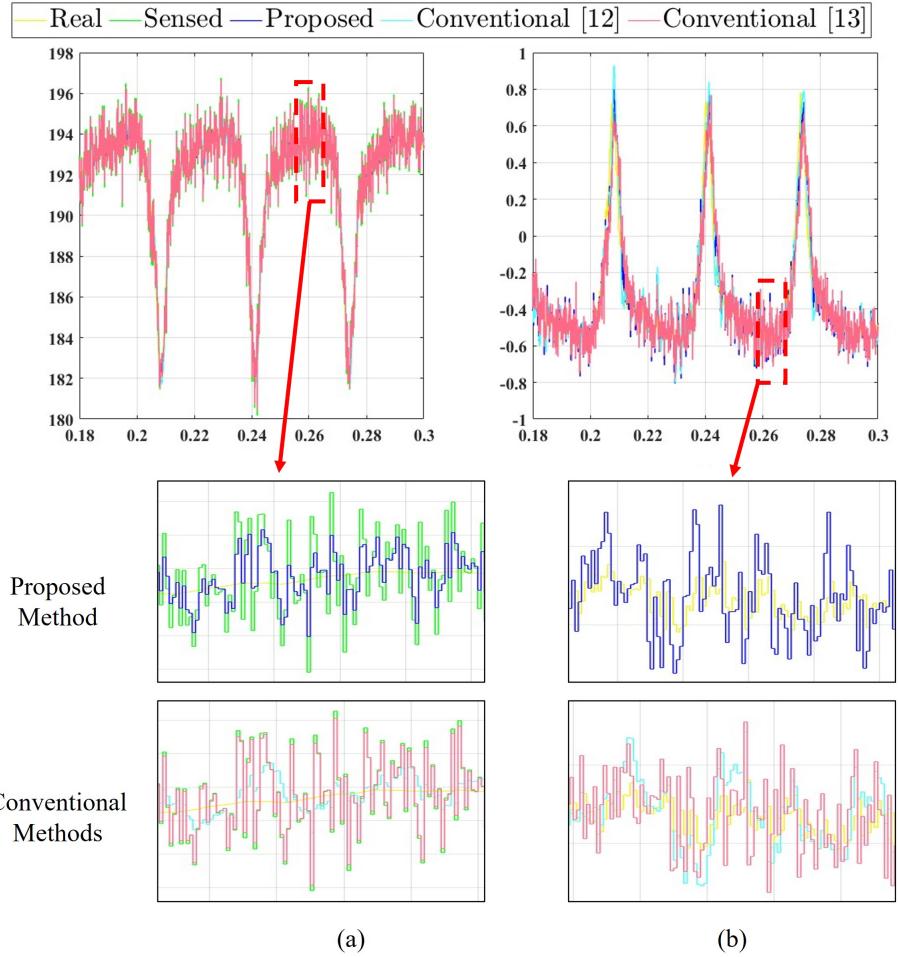


Figure 4.3: Results of Rotor Speed and Uncertainties Estimation: (a) rotor speed estimation result, (b) uncertainty estimation result

Fig. 4.3 illustrates the performance of the proposed observer in estimating rotor speed and unknown uncertainties, compared with two existing methods [12], [13]. In Fig. 4.3-(a), the rotor speed estimation result is presented. The proposed method closely follows the true rotor speed even in the presence of rapidly changing uncertainty and measurement noise. It is also evident that the proposed method yields significantly less estimation noise than the conventional methods, demonstrating enhanced robustness to measurement noise and modeling uncertainty.

Fig. 4.3-(b) shows the estimation results of the uncertainty. The enlarged views

highlight a region where rapid changes occur. All methods successfully track the variations of the unmodeled dynamics with higher resolution and lower delay. Especially, the proposed method achieves fast and delay-free estimation, successfully tracking sudden changes in the system uncertainty in real time.

### 4.3 Flux Linkage and VSI Nonlinearity Estimation

In this section, we demonstrate the effectiveness of the proposed observer in estimating the VSI nonlinearities. These nonlinearities are caused by turn on/off switching delays ( $T_{on}, T_{off}$ ), dead time ( $T_d$ ), voltage drops of the diodes and power devices ( $V_D, V_P$ ), and saturation effects. These phenomena cause a mismatch between the reference voltage ( $v^*$ ) and the actual applied voltage ( $v$ ), which depends on both the direction and magnitude of the current. This behavior cannot be captured by simple linear models and must be treated as a nonlinear disturbance ( $\Delta v = v - v^*$ ), which varies with load dynamics and switching state. These nonlinearity lead to degraded performance in conventional control schemes if not properly estimated or compensated as shown in Fig 4.4 and Fig 4.5.

The following equations describe the  $d$ - and  $q$ -axis flux linkages of the PMSM:

$$\begin{aligned}\dot{\lambda}_d(t) &= \omega_r \lambda_q(t) - R_s i_d(t) + v_d^*(t) + \Delta v_d(t), \\ \dot{\lambda}_q(t) &= -\omega_r \lambda_d(t) - R_s i_q(t) + v_q^*(t) + \Delta v_q(t),\end{aligned}\tag{4.2}$$

where  $\lambda$  is stator flux linkage,  $\omega_r$  is electrical rotor speed,  $R_s$  is stator winding resistance,  $i$  is stator current, and subscripts  $d$  and  $q$  denote the direct and quadrature

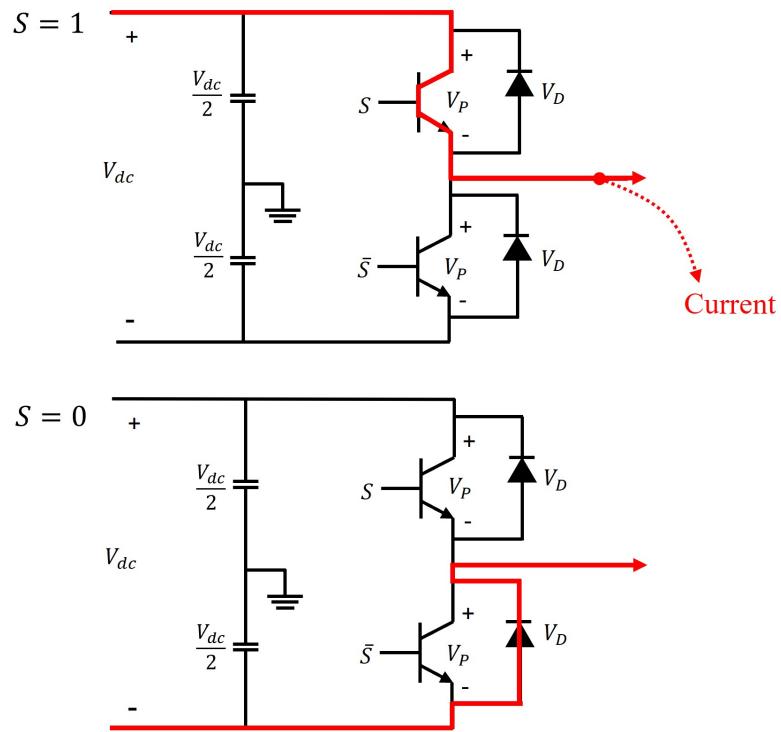


Figure 4.4: Current conduction paths for two switching states in a VSI:  $S = 1$  (top) and  $S = 0$  (bottom), showing the corresponding current flow direction.

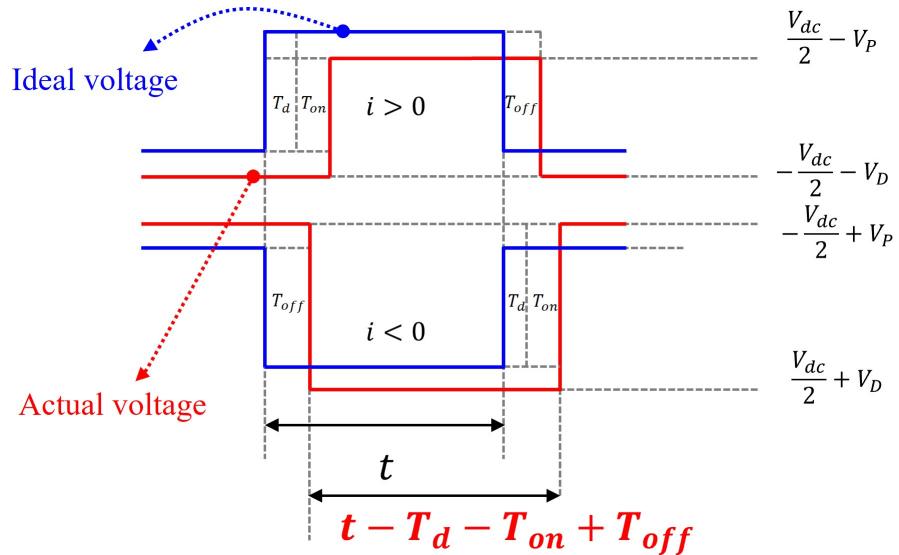


Figure 4.5: Comparison of ideal and actual VSI output voltages under both positive and negative current conditions, illustrating the effects of dead time ( $T_d$ ), switching delays ( $T_{on}, T_{off}$ ), and voltage drops ( $V_D, V_P$ ).

Table 4.3: Specifications of the PMSM Drive

Base speed	250 RPM
Maximum torque	180 Nm
DC-link voltage ( $V_{dc}$ )	325 V
Sampling time ( $T_s$ )	25 $\mu$ s
Number of pole pairs ( $P$ )	8
Flux linkage of $d$ -axis	0.24 mH
Flux linkage of $q$ -axis	0.29 mH
Stator resistance ( $R_s$ )	10.9 m $\Omega$
Dead time ( $T_d$ )	1 $\mu$ s
Voltage drop of the diodes ( $V_D$ )	0.8 V
Voltage drop of the power devices ( $V_P$ )	0.5 V

axes. The parameters used in the simulation are summarized in Table 4.3.

Following the same procedure as the previous rotor speed and uncertainty estimation simulations in section 4.2, the tuning parameters are visualized and the set of parameters that minimizes the estimation error is selected and validated.

#### 4.3.1 Correlation, Visualization and Estimation Results

The tuning parameters are selected as shown in Table 4.4. Fig. 4.6-(a) shows  $d$ -axis flux linkage, Fig. 4.6-(b) shows  $q$ -axis flux linkage estimation error and Fig. 4.7-(a) and Fig. 4.7-(b) represent  $d, q$ -axis nonlinearity estimation error respectively for each parameter.

Table 4.4: Tuning parameter ranges used in the simulation

Method	Parameter	Minimum value	Maximum value
Proposed method	$\eta$	$10^9$	$5.3 \times 10^{10}$
	$\alpha$	$3.75 \times 10^{-2}$	$7.25 \times 10^{-2}$
LMI-based	$\kappa$	$10^{-2}$	$1.5 \times 10^{-1}$
	$\alpha$	$1.65 \times 10^3$	$2.35 \times 10^3$
BP-based	$\eta_1$	$4 \times 10^4$	$6 \times 10^5$
	$\eta_2$	$10^4$	$1.5 \times 10^5$

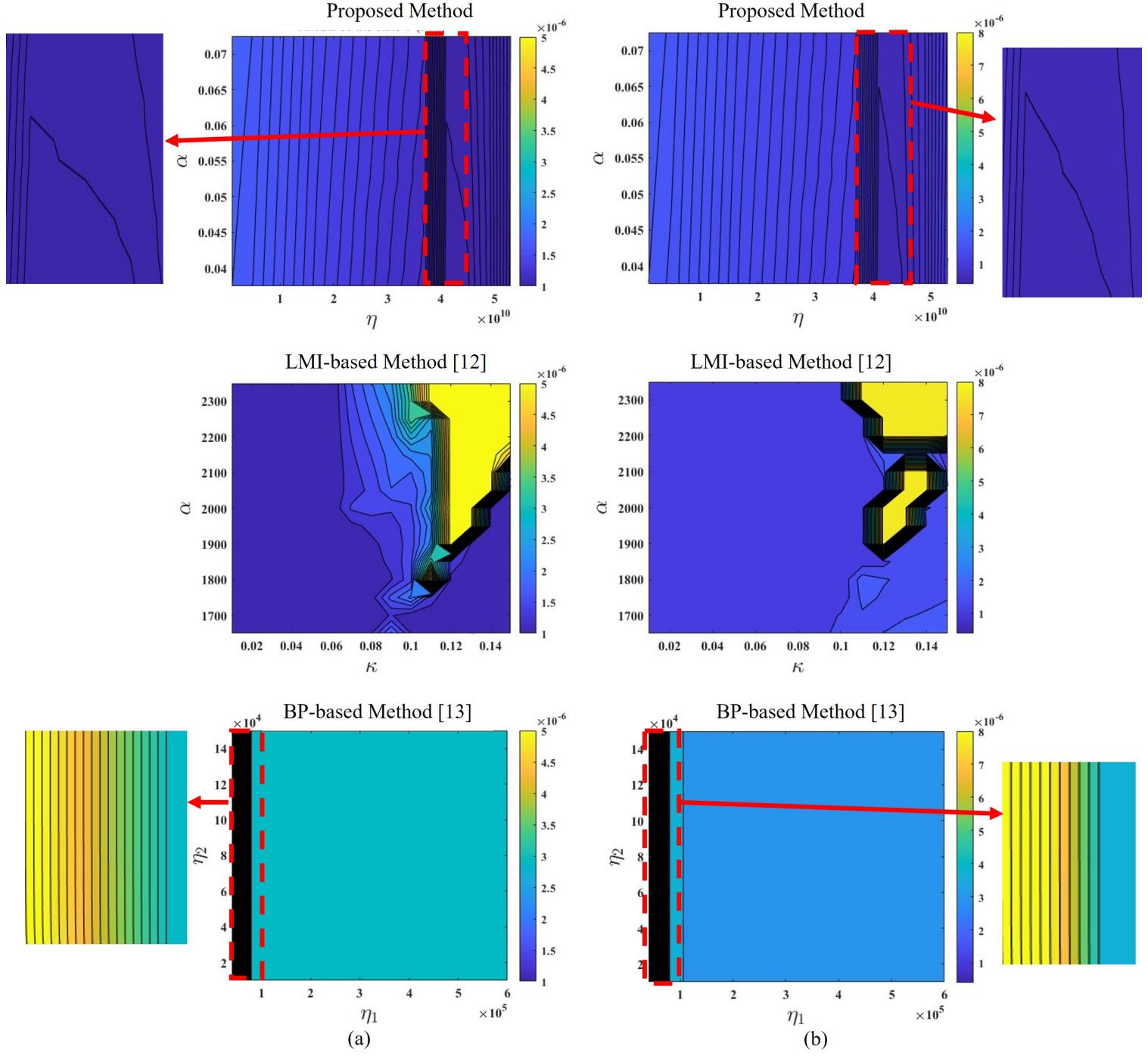


Figure 4.6: RMSE performance changes of the tuning parameters in different methods:  
 (a)  $d$ -axis flux linkage estimation result, (b)  $q$ -axis flux linkage estimation result

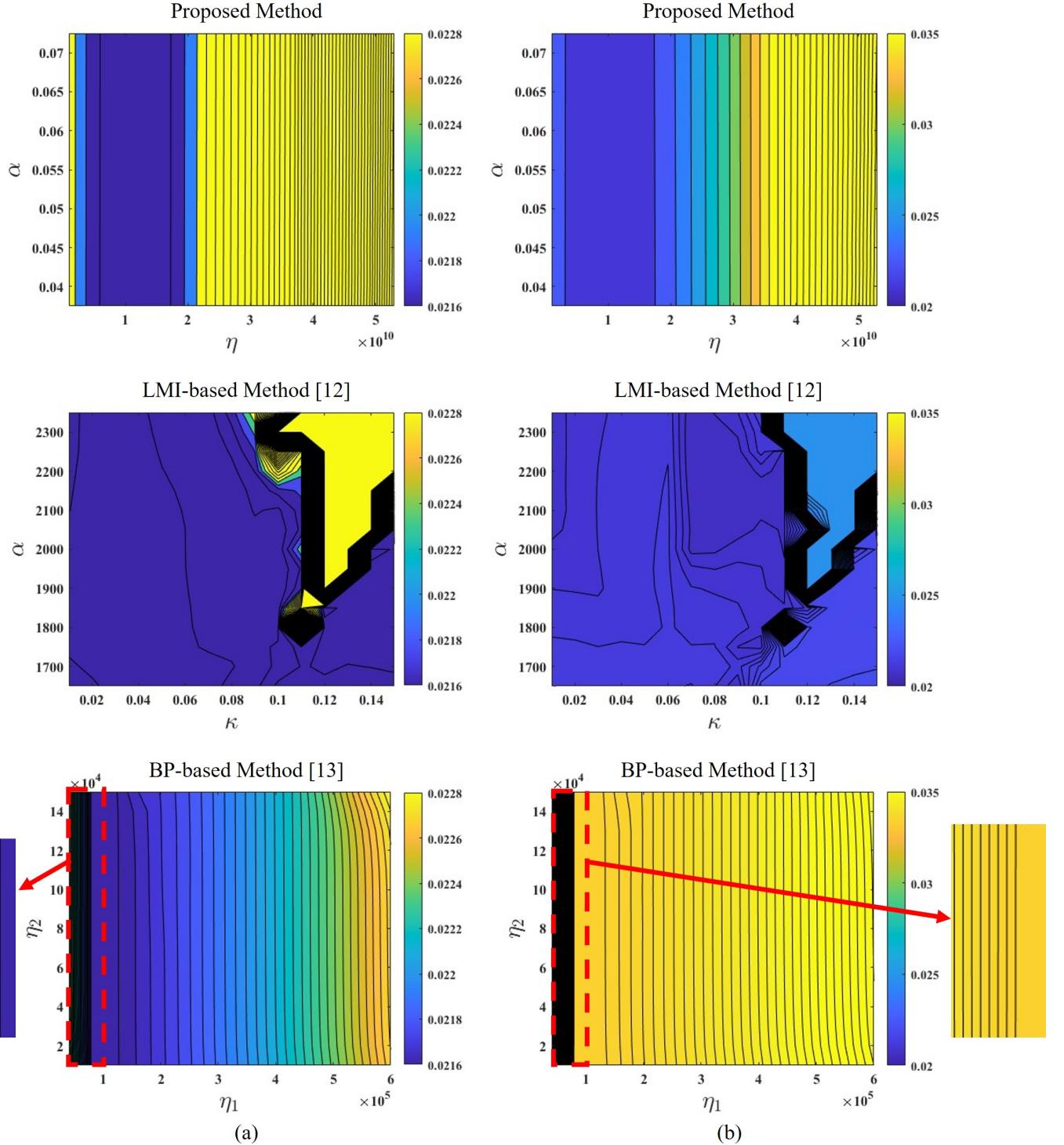


Figure 4.7: RMSE performance changes of the tuning parameters in different methods: (a)  $d$ -axis VSI nonlinearity estimation result, (b)  $q$ -axis VSI nonlinearity estimation result

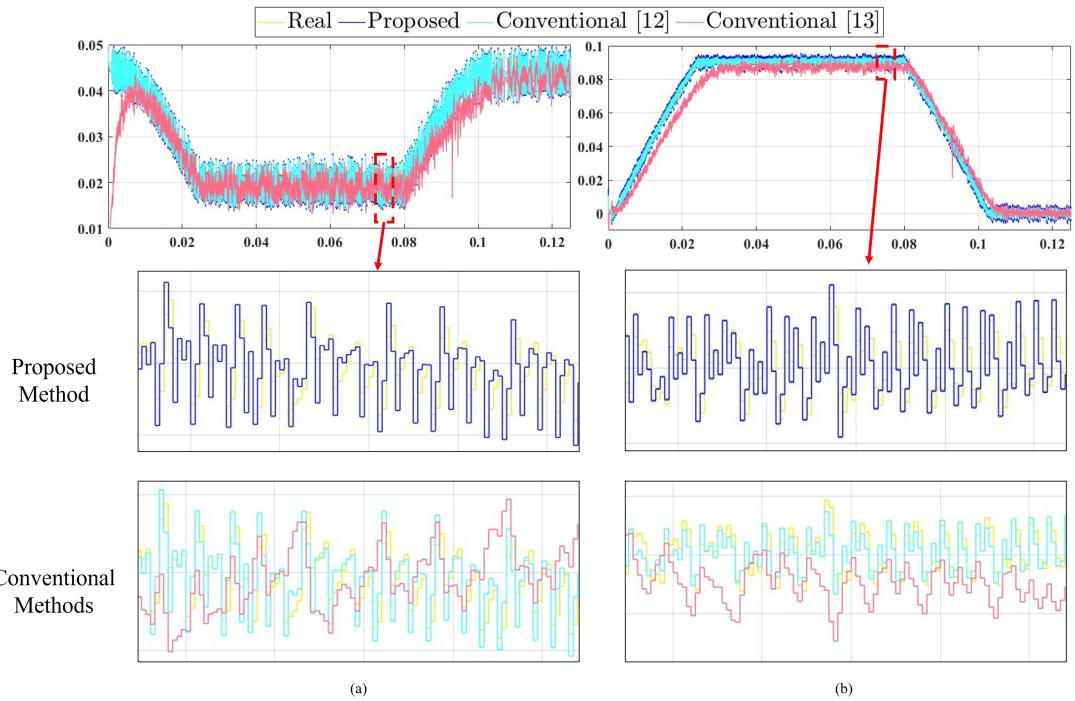


Figure 4.8: Results of Flux Linkage Estimation: (a)  $d$ -axis flux linkage estimation result, (b)  $q$ -axis flux linkage estimation result

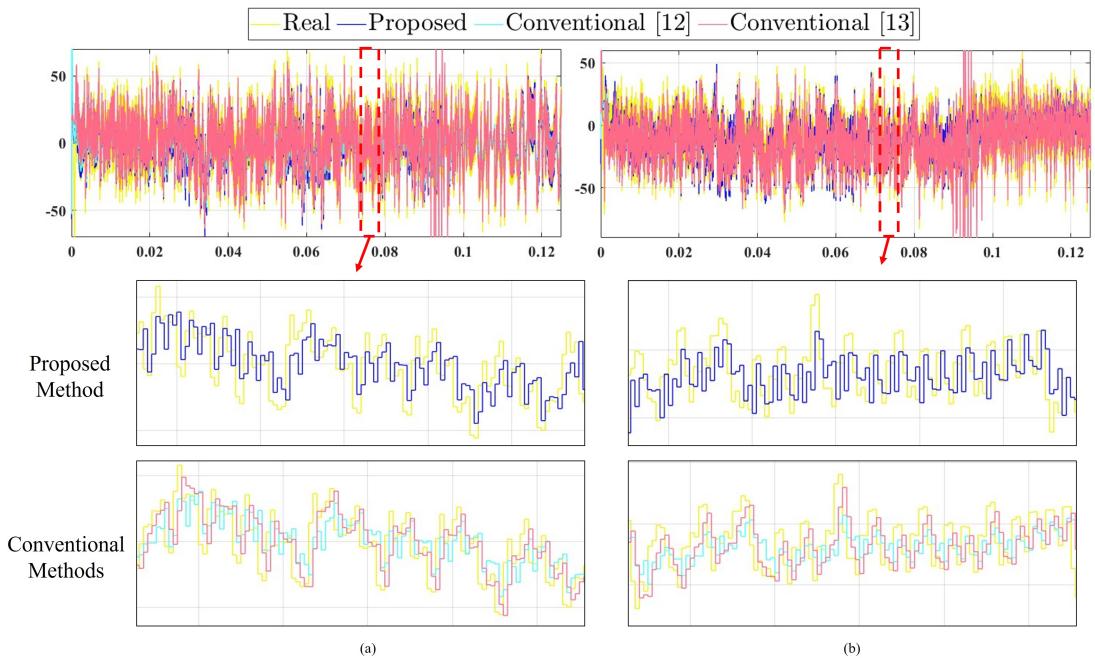


Figure 4.9: Results of VSI Nonlinearity Estimation: (a)  $d$ -axis VSI nonlinearity estimation result, (b)  $q$ -axis VSI nonlinearity estimation result

#### 4.4 Discussion

The two simulation scenarios—(i) speed/load torque estimation and (ii) flux linkage/VSI nonlinearity estimation—collectively highlight the differences in tuning robustness and estimation accuracy among the three observer design approaches. The key characteristic are summarized and analyzed as follows:

- **Proposed Method:** Exhibits consistently low RMSE across a wide tuning parameter range, indicating high robustness to parameter variation. This stems from the integration of backpropagation-based learning and LMI-based gain design, enabling accurate estimation and stable even under uncertainty and measurement noise.
- **LMI-based Method [12]:** Shows good performance near optimal parameters but suffers from narrow stability margins. Its reliance on precise tuning reduces robustness and limits applicability under uncertain or varying conditions.
- **BP-based Method [13]:** Demonstrates high sensitivity to tuning. with performance degrading rapidly outside a small optimal range. Unlike the two methods above, the absence of an LMI-based gain structure results in limited robustness, particularly under uncertainty and measurement noise.

# Chapter 5

## Conclusion and Future Work

In this thesis, backpropagation- and LMI-based neuro-adaptive observer design for an uncertain nonlinear systems in discrete-time domain was proposed. A design method for the weight update rule based on the backpropagation, along with the observer gain derived by solving the LMI problem for all vertices was proposed. Furthermore,  $H_\infty$ -based design minimizes the worst-case estimation error gain via convex LMI optimization, offering guaranteed robustness that is not achievable by backpropagation-based or LMI-based observers alone. The primary goal was to achieve robust and accurate estimation of both system states and nonlinearity terms with reduced sensitivity to measurement noise and other uncertainties. To evaluate the effectiveness of the proposed method, two simulation scenarios were considered: (i) rotor speed and load torque estimation in a PMSM system and (ii) flux linkage and VSI nonlinearity estimation. The results demonstrated that the proposed observer outperformed two representative conventional methods in terms of robustness, tuning simplicity, and estimation accuracy.

The LMI-based method suffers from tuning complexity due to a large number of design parameters and is vulnerable to divergence during discretization. The BP-based method simplifies the weight update law, but it also faces divergence risks after discretization and only guarantees convergence within a bounded error region. The pro-

posed method overcomes these limitations by minimizing the number of tuning parameters, removing the dependence on continuous-time stability, and ensuring robust and accurate estimation without requiring discretization from continuous-time designs. While some divergence risk remains due to the learning rate, simulation results showed that proper tuning of a small number of parameters is sufficient to ensure reliable performance. Furthermore, the method is directly formulated in the discrete-time domain, which makes it naturally compatible with digital control implementations.

In summary, this thesis contributes a robust observer design that (i) enables real-time estimation of nonlinear dynamics using neural networks, (ii) ensures stability through LMI-based gain tuning with integration of backpropagation-based weight updates, (iii) enhances robustness by incorporating  $H_\infty$ -based optimization, and (iv) can be applicable to the discrete-time domain. These features collectively enable accurate state estimation in the presence of measurement noise, and uncertainties.

Future work will focus on improving the practicality and scalability of the proposed observer. One direction is the automation of tuning parameter selection. By convergence speed and optimal learning rates, the manual effort required for parameter tuning can be significantly reduced, enhancing usability in real-world applications. Another direction involves implementing the proposed observer on real-time embedded platforms. Furthermore, the observer could be extended into a full-closed loop controller by incorporating LMI-based neural network compensation with guaranteed stability.

## References

1. A. Zemouche, and M. Boutayeb, “Observer Design for Lipschitz Nonlinear Systems: The Discrete-Time Case,” *IEEE Transactions on Circuits and Systems*, vol. 53, no. 8, pp. 777–781, August 2006.
2. W. Li, L. Liang, W. Liu, and X. Wu, “State of charge estimation of lithium-ion batteries using a discrete-time nonlinear observer,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 11, pp. 8557–8565, Nov. 2017.
3. W. Zhang, H. Su, F. Zhu, and D. Yue, “A note on observer for discrete-time lipschitz nonlinear systems,” *IEEE Transactions on Circuits and Systems*, vol. 59, no. 2, Feb. 2012.
4. Y. Yu, and Y. Shen, “Robust sampled-data observer design for Lipschitz nonlinear systems,” *Kybernetika*, vol. 54, no. 4, 2018.
5. T. Raissi, D. Efimov, and A. Zolghadr, “Interval State Estimation for a Class of Nonlinear Systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 260–265, Jan. 201.
6. M. Boutayeb, and D. Aubry, “A strong tracking extended kalman observer for nonlinear discrete-time systems,” *IEEE Transactions on Automatic Control*, vol. 44, no. 8, pp. 1550–1556, Aug. 1999.
7. A. Barrau, and S. Bonnabel, “The invariant extended kalman filter as a stable observer,” *IEEE Transactions on Automatic Control*, vol. 62, No. 4, Apr. 2017.

8. S. Afshar, F. Germ, and K. Morris, “Extended kalman filter-based observer design for semilinear infinite-dimensional systems,” *IEEE Transactions on Automatic Control*, vol. 69, no. 6, Jun. 2024.
9. M. Ouassaid, M. Maaroufi, and M. Cherkaoui, “Observer-based nonlinear control of power system using sliding mode control strategy,” *Electric Power System Research*, pp. 135–143, 2012.
10. B. Sanchez, C. Cuvas, P. Ordaz, O. Sanchez, and A. Poznyak, “Full-order observer for a class of nonlinear systems with unmatched uncertainties: joint attractive ellipsoid and sliding mode concepts,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, Jul. 2020.
11. R. Lopez, and R. Yescas, “State estimation for nonlinear systems under model uncertainties: a class of sliding-mode observers,” *Journal of Process Control*, pp. 363–370, 2005.
12. W. Jeon, A. Chakrabarty, A. Zemouche, and R. Rajamani, “LMI-based neural observer for state and nonlinear function estimation,” *International Journal of Robust Nonlinear Control*, pp. 6964–6984, 2024.
13. F. Abdollahi, H. Talebi, and R. Patel, “A stable neural network-based observer with application to flexible-joint manipulators,” *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 118–129, Jan. 2006.
14. M. Khoygani, R. Ghasemi, and A. Vali, “Intelligent nonlinear observer design for

- a class of nonlinear discrete-time flexible joint robot,” *Intelligent Service Robotics*, vol. 8, pp. 45–56, 2015.
15. Y. Yang, C. Lin, B. Chen, and X. Zhao, “ $H_\infty$  observer design for uncertain one-sided Lipschitz nonlinear systems with time-varying delay,” *Applied Mathematics and Computation*, vol. 375, pp. 1–12, 2020.
  16. G. Cybenko, “Approximation by superpositions of sigmoidal functions,” *Mathematics of Control, Signals, and Systems*, pp. 303–314, 1989.
  17. K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, pp. 251–257, 1991.
  18. K. Narendra and K. Parthasarathy, “Identification and Control of Dynamical System Using Neural Networks,” *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
  19. S. Chen, S. Cai, T. Chen, C. Xu, and J. Chu, “Dynamic Neural Network-based Robust Observers for Uncertain Nonlinear Systems,” *Neural Networks*, vol. 60, pp. 44–52, 2014.
  20. S. Chen, S. Cai, T. Chen, C. Xu, and J. Chu, “Neural Observer With Lyapunov Stability Guarantee for Uncertain Nonlinear Systems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 11527–11541, 2024.
  21. Z. Yi, W. Xie, L. Liu, and B. Xu, “A Neural Network Adaptive Interval Observer

- Design for Nonlinear Systems,” *IET Control Theory & Applications*, vol. 16, pp. 615–624, 2022.
22. F W Lewis, S. Jagannathan, and A Yesildirak, “Neural Network Control of Robot Manipulators and Nonlinear Systems,” 2020.
  23. Z. Yi, W. Xie, L. Liu, and B. Xu, “Deep Recurrent Neural Network Observer for Uncertain Systems,” *IFAC-PapersOnLine*, vol. 56, pp. 6851–6856, 2023.
  24. H. Anh, and N. Dat, “Online Adaptive Neural Observer for Prescribed Performance Hyper-Chaotic Systems,” *Knowledge-Based Systems*, vol. 299, 2024.
  25. J. Wang, Y. Song, and G. Zhai, “Adaptive Neural Network Tracking Control of Robotic Manipulators Based on Disturbance Observer,” *Processes*, vol. 12, no. 3, pp. 499, 2024.
  26. Y. H. Jang and J. J. Lee, “Neural network-based adaptive controller design of robotic manipulators with an observer,” *Biomed Eng Online*, vol. 6, no. 1, pp. 5, 2007.
  27. S. Long, X. Dang, and J. Huang, “FOESO-Net: A specific neural network for fast sensorless robot manipulator torque estimation,” *Neural Networks*, vol. 168, pp. 14–31, Nov, 2023.
  28. T. L. Paes and A. C. Oliveira, “Neural Network Controller for Power Electronics Circuits,” in *Proc. of the International Conference on Engineering Applications*, 2019.

29. N. M. F. Rashed, M. E. Ahmed, and A. Fathy, “Adaptive neural network control of DC–DC power converter,” *Expert Systems with Applications*, vol. 235, 120195, 2023.
30. P. Dworak and Z. Leonowicz, “Neural Network Observer Design for Sensorless Control of Induction Motor Drive,” *IFAC-PapersOnLine*, vol. 49, no. 27, pp. 240–245, 2016.
31. P. Penczek, R. Orlowska-Kowalska, and M. Kaminski, “Neural Network Applications in Electrical Drives—Trends in Control,” *Energies*, vol. 16, no. 11, pp. 4441, 2023.
32. J. Shim, M. Kim, B. Park, and J. Ha, “Position Estimation and Sensorless Control of PMSM Using Neural Network in Low-Speed Region,” *IEEE Energy Conversion Congress and Exposition(ECCE)*, pp. 6536–6542, 2024.
33. C. Scherer and S. Weiland, “Linear Matrix Inequalities in Control,” University of Stuttgart and Eindhoven University of Technology, 2017.

## Acknowledgements

