2008

NONLINEAR IDENTIFICATION AND CONTROL: A PRACTICAL SOLUTION AND ITS APPLICATION

Xiaodong Na
University of Kentucky, xna2@uky.edu

ABSTRACT OF DISSERTATION

Xiaodong Na

The Graduate School

University of Kentucky

2008

NONLINEAR IDENTIFICATION AND CONTROL: A PRACTICAL SOLUTION AND ITS APPLICATION

ABSTRACT OF DISSERTATION

A dissertation submitted in partial fulfillment of the

requirements for the degree of Doctor of Philosophy in the

College of Engineering

at the University of Kentucky

By

Xiaodong Na

Lexington, Kentucky

Director: Dr. Bruce Walcott, and Dr. YuMing Zhang, Professor of Electrical Engineering

Lexington, Kentucky

2008

ABSTRACT OF DISSERTATION

NONLINEAR IDENTIFICATION AND CONTROL: A PRACTICAL SOLUTION AND ITS APPLICATION

It is well known that typical welding processes such as laser welding are nonlinear although mostly they are treated as linear system. For the purpose of automatic control, Identification of nonlinear system, especially welding processes is a necessary and fundamental problem. The purpose of this research is to develop a simple and practical identification and control for welding processes.

Many investigations have shown the possibility to represent physical processes by nonlinear models, such as Hammerstein structure, consisting of a nonlinearity and linear dynamics in series with each other. Motivated by the fact that typical welding processes do not have non-zeroes, a novel two-step nonlinear Hammerstein identification method is proposed for laser welding processes. The method can be realized both in continuous and discrete case.

To study the relation among parameters influencing laser processing, a standard diode laser processing system is built as system prototype. Based on experimental study, a SISO and 2ISO nonlinear Hammerstein model structure are developed to approximate the diode laser welding process. Specific persistent excitation signals such as PRTS (Pseudo-random-ternary-series) to Step signal are used for identification. The model takes welding speed as input and the top surface molten weld pool width as output. A vision based sensor implemented with a Pulse-controlled-CCD camera is proposed and applied to acquire the images and the geometric data of the weld pool. The estimated model is then verified by comparing the simulation and experimental measurement. The verification shows that the model is reasonably correct and can be use to model the nonlinear process for further study. The two-step nonlinear identification method is proved valid and applicable to traditional welding processes and similar manufacturing processes. Based on the identified model, nonlinear control algorithms are also studied. Algorithms include simple linearization and backstepping based robust adaptive control algorithm are proposed and simulated.

KEYWORDS: Hammerstein, Laser processing, Nonlinear identification, adaptive control, manufacturing

Xiaodong Na

January 19, 2008

NONLINEAR IDENTIFICATION AND CONTROL: A PRACTICAL SOLUTION AND ITS APPLICATION

By

Xiaodong Na

Bruce Walcott

Co Director of Dissertation

Yuming Zhang

Co Director of Dissertation

Yuming Zhang

Director of Graduate Studies

January 19, 2008

RULES FOR THE USE OF DISSERTATIONS

Unpublished dissertations submitted for the Doctor's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.

Name	Date

DISSERTATION

Xiaodong Na

The Graduate School
University of Kentucky
2008

NONLINEAR IDENTIFICATION AND CONTROL: A PRACTICAL SOLUTION AND ITS APPLICATION

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Xiaodong Na
Lexington, Kentucky
Director: Dr. Bruce Walcott, and Dr. YuMing Zhang, Professor of Electrical Engineering
Lexington, Kentucky
2008

ACKNOWLEDGEMENTS

This work is supported by the U. S National Science Foundation under grant CMMI-0527889, the Natural Science Foundation of China under grant 60540420641 and Center for Manufacturing at University of Kentucky. At first, I would like to thank my advisor Dr. Bruce Walcott and Dr. Yuming Zhang and committee including Dr. Larry Holloway, and Dr. Fuqian Yang for their advice and support as always. I also want to thank Professor Yusheng Liu for the help in my research. Also, thank all of my colleagues and friends at the Welding Research Laboratory for their help throughout my research and experiments. Finally I want to thank my wife, Zheng Xu, who has been a wonderful lady always standing by my side and helping me through difficulties and also my family including my parents, my sisters for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FILES

CHAPTER 1

INTRODUCTION

1.1 Background

Laser Welding is a fusion process accomplished with various lasers applying to materials. As a non-contact process, laser welding finishes the welding work through laser beam. With laser beam, energy is concentrated and used directly on the small welding area. As a result, the welding zone is very narrow and hardly distorted due to little heat influence. Compared to traditional processes, Laser Welding is of potential. Its non-contact, localized, and narrow heat zone can create high quality result. Common re-working and after-work procedure are no more required, which saves cost and labor. Right now, Laser processing has been widely applied in various fields including automotive, microelectronics, aerospace, etc.

Common types of laser welding includes CO2 gas laser, Solid state laser (YAG type), and Diode laser welding. CO2 laser uses a mixture of high purity carbon dioxide with helium and nitrogen as the medium, infrared of 10.6 micro-meters. Argon or helium is used to prevent oxidation. YAG laser uses a solid bar of yttrium aluminum garnet doped with neodymium as the medium, whose infrared is only 1.06 micro-meters. Diode laser is based on the conversion between high electrical to optical powers [1]-[5].

In many cases, the concern with health and safety forced engineers to design an advanced automatic welding process. However, the difficulty to ensure quality and lacking of knowing knowledge made automatic welding a challenging task. One of the critical factors along with welding process is the measurement system that provides information of welding pool. The importance of the measurement on the welding pool is obvious. First of all, not an exact model has ever been developed to describe the welding process. Even though it is possible to model the welding process, the model is too complicate for control purposes. Secondly, too many unknown factors influence the

dynamic and changeable process. Third, successful intelligent welding system requires appropriate and real time measurement working with specific developed control algorithm so that the process is robust and adaptive. Fourth, known factors such as strong welding light, reflective welding pool surface, dynamic and fast changing, noise, physical deformation, and so on makes measurement hardly accomplish. Last, very limited useful sensors are applicable for welding process measurement.

Various Researches has been done related to welding pool measurement based on the vision sensing tool such as monochrome camera and optic sensors. Through the sensors, welding related information is achieved online or offline so that control loop can adjust the welding process parameters. Considering the role of penetration to welding process, many study started with penetration measurement and control. With aid of pool oscillation [6]-[7], infrared sensor [8]-[9], ultrasound [10]-[11], and related, reliable and applicable measurement and control of welding pool was possible. The disadvantage of these methods is its limit while differentiate from partial penetration and full penetration [8] [12]-[13]. Besides, the system is very easily influenced by instable noise. Accordingly, successful two-dimensional measurement can to some extent improve the control progress [14]-[16]. The most potential method is 3D welding surface measurement and control because it is able to provide enough geometric process information. Very limited researches have been done about three-dimensional surface of welding pool. In the past years, University of Kentucky, novel and pioneering research has been done in this field. Special CCD cameras were added into the process, working with various algorithms, extracting three-dimensional model information related the welding pool [16]-[23]. Right now, the ongoing research has been extended from arc welding to laser welding. The goal is to design a suitable sensor associated with regular welding process.

Beyond the difficulty of measurement, model identification is another critical procedure for successful automatic welding system. It is well known that most real life dynamic processes are inherently nonlinear. Laser welding is of no exception. It is dynamic, nonlinear, and of uncertainty although in many cases laser welding processes have been treated as linear system. Unlike linear processes approximating the processes around given operating points, nonlinear model is advantageous to describe the global

behavior of the processes on the entire range. However, due to the nonlinear nature and limited knowledge of welding processes, it is challenging to improve control quality and even automated processes. Moreover, the difficulty for welding process control is the model architecture because most of the information and relationship are from the experiments. There has hardly been an appropriate model applied to the welding process. Accordingly, developing suitable identification procedure specifically for laser welding process is very necessary. Because it is not realistic to develop identification techniques for general nonlinear processes, our study is focused on potential application of laser welding processes. We consider especially the processes that consist of two parts: a static nonlinearity and a linear dynamic subsystem. These model structures are also called Hammerstein model and have been successfully applied to nonlinear processes in many areas such as heat exchangers, chemical processes, biological processes, tank reactor, distillation columns, and so on. We believe Hammerstein model can also possibly represent laser welding process. We will further consider not only discrete system model but also continuous model for the sake of control performance.

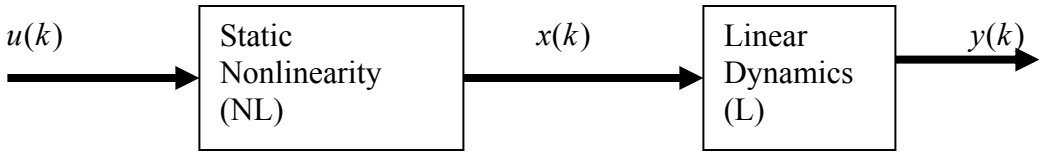


Figure 1.1 Nonlinear Hammerstein Structure

1.2 Research Objective

This thesis is intended to develop a simple method to identification and control for laser welding processes that can be described by a Hammerstein model with a static nonlinearity and a linear dynamics subsystem in series with each other. The identification is of both discrete-time and continuous-time case. In particular, the nonlinearity is very slowly time-varying and determines the system static characteristic and the linear dynamic determines the transient characteristics.

The objectives include:

1. Develop a laser welding system as an experimental platform

2. Characterizing the laser welding process in terms of inputs (speed, laser intensity) and output (weld pool geometry) and investigate the relations
3. Develop a practical method to identify the laser welding processes under working conditions
4. Use experimental data to test and validate the identification method

1.3 Basic Idea

A standard diode laser system is roughly implemented for research purposes in the welding research lab, Center for Manufacturing, University of Kentucky. The sensor system is implemented with hardware including a specially made digital camera, with built in band pass filter, Frame Grabber (National Instrument made), and vision based software (LabView/vision/IMAQ). The digital camera is synchronized with a high speed pulse laser. The sensor system acquires the images and extracts the geometric information of the molten weld pool. The working piece is 1mm mild steel sheet whose length is about 400mm and width is about 100mm. To study how the parameters such as welding speed and laser intensity influence the weld pool, series open loop experiments under different inputs are made. Through the experimental data, the identification is then developed based on Hammerstein model structure in both continuous and discrete cases. Based on the model, control algorithm is developed and simulation is created. Experiments are then made to verify the correctness of the model identification.

1.4 Organization

This dissertation is organized as follows. In Chapter 2, existing sensing and control for laser welding are introduced. In Chapter 3, a review of Hammerstein identification and control algorithms are described. In Chapter 4, vision based sensor and related image processing algorithms are presented. In Chapter 5, the proposed Hammerstein Identification is briefly introduced and proved the applicability. In Chapter 6, experimental setup of the vision based sensor and results of image processing are demonstrated. In chapter 7, the proposed Hammerstein Identification procedure is described in detail and in Chapter 8 the results of Identification including the experimental set up are given. A SISO nonlinear model is also estimated and verified

with experimental data. The identification is also expanded to MIMO case. In Chapter 9, possible nonlinear control algorithm is proposed and investigated by simulation. Conclusion is summarized and future work is estimated in Chapter 10.

CHAPTER 2

REVIEW OF SENSING AND CONTROL FOR LASER WELDING

Laser welding has drawn a lot of attention in fields like shipyards, automobiles, aircrafts, and so on. Compared to traditional techniques, laser welding is advantageous for its high speed, high ratio of depth to width, small heat affected zone, low distortion, and potentially deep penetration [1]. However the inefficient quality monitoring and environment concerns [4] [5] to some extent hinder the development of the automated laser welding applications for industries. Accordingly developing an automated laser welding process has been a critical field in research and industrial study. Till now, various studies have been done to monitor the laser welding process. Some focused on the emission signals such as acoustic, infrared, ultraviolet, plasma, and so on [25]. Others aimed to the weld pool images acquired with CCD cameras [17] [25] [29]. Some concentrated on the heat distribution with numerical analysis [24].

2.1 Keyhole Sensing and Control

During the process of laser welding, high energy of laser beam is focused onto a single location and a keyhole is created as shown in Figure 2.1. To ensure successful welding and avoid effects like burnt-through, keyhole depth should be controlled not too much beyond the height of the material. The advantage of Keyhole control is the possible small heated zone, which results better after work quality. Around the keyhole depth control, much work has been studied. In [8] a COMS camera is installed into the system, corresponding to the optic mirror and monitoring online the welding process. Relation between penetration depth and laser power was analyzed based on the experiments. Although the system is only used for ND-YAG and CO2 Laser Beams, its possible potential might be applicable to other systems. In [11], penetration is studied on CO2 laser welding of thin sheet steel under experimental conditions. Statistical based analysis and various sensor systems are also studied. The difficulty of the keyhole control is how

to decide the penetration in real time, without sacrificing the advantage of laser welding process. In [2]-[3] [13], a sensor system based on emission detects and moreover the penetration was studied and developed.

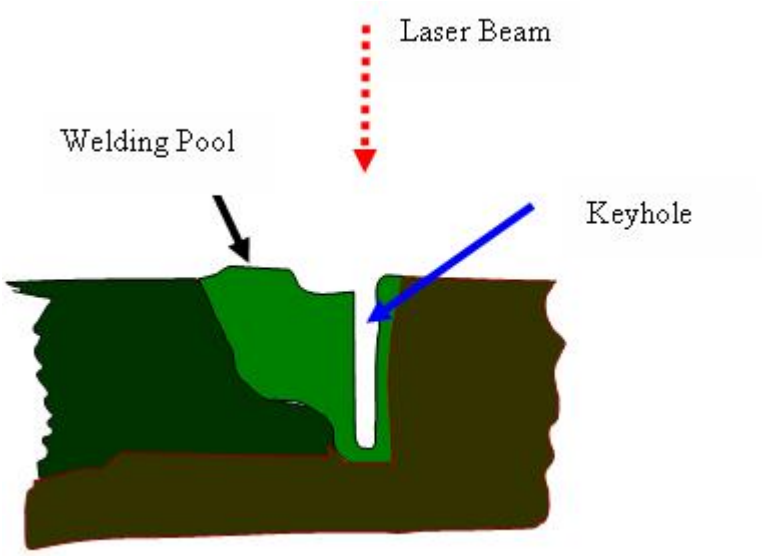


Figure 2.1 Keyhole in laser welding

2.2 Emission or Radiation

A challenge for laser welding is how to control the process online without compromising the welding quality. Because it is hard to find the parameters and relation related to the process, suitable measurement is always influential. Approximate analysis on signals will be of great help on the process. Various signals such as optical, acoustical, infrared, and so on [3] [6] [11] [12] [23] are studied based on spectrum and statistical analysis as shown in Figure 2.2.

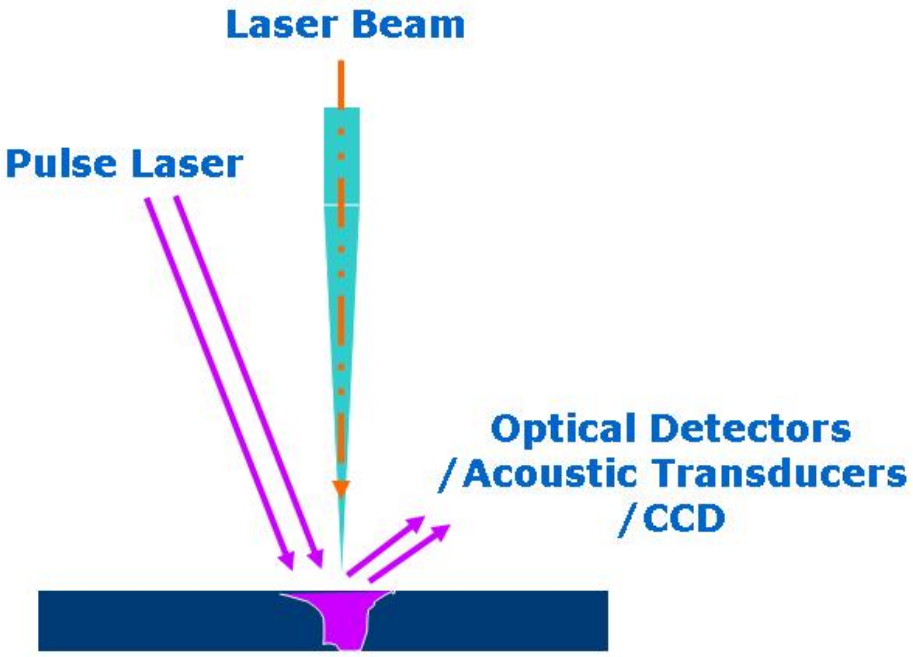


Figure 2.2 Signal emission and sensors in laser welding

2.3 Focus Control

In [25] [26], the focus length of the laser is adjustable by the control system. With specific system design, the laser power can play to its full potential. The disadvantage is its instable noise, off-line monitoring, and difficulty to detect full penetration and partly penetration. The design will have to take into consideration of sensing system integration.

2.4 Power and Speed Control

It is a logic sense to think of controlling power and speed during the laser welding process and since laser welding is using the heat energy to melt the material and finish the welding, critical parameters that influence the energy should be taken care of at first. Power and travel speed are the influential factors related to laser welding. However, the difficulty is how to implement a suitable relation to evaluate the process. With power and travel speed as input, the welding quality parameter as output, laser welding process could be achieved as a tracking control [27]-[28].

2.5 Vision Based Sensing and Control

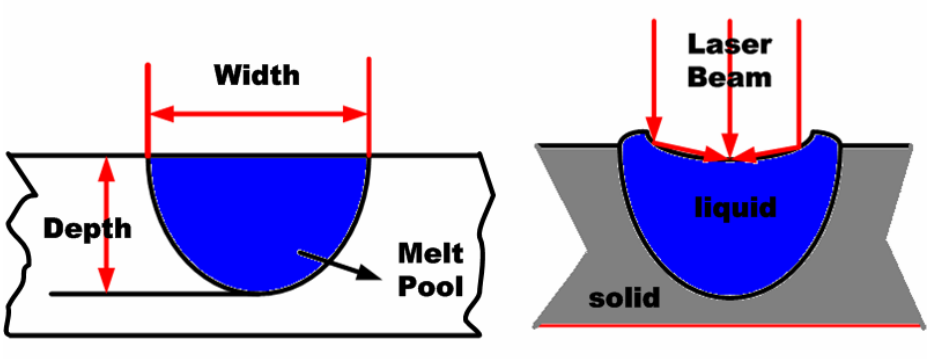


Figure 2.3 Weld pool (1a) Laser Welding (1b)

As shown in Figure 2.3, the full knowledge of the weld pool geometry includes the length, the width, and the depth. Weld pool images were acquired with CCD/CMOS cameras and studied to investigate the relations among various parameters along with laser welding process [10] [13] [29]. An on-going research to study laser welding process based on Vision based image acquisition and processing is being done in the Welding Research Lab, Center for Manufacturing, University of Kentucky.

CHAPTER 3

REVIEW OF IDENTIFICATION AND CONTROL

In this chapter, we briefly review the literature of nonlinear identification and control. As introduced, we are intending to develop an algorithm to identify both the linear dynamics and the nonlinearity in series with each other, which is similar to Hammerstein-type systems. The consideration of Hammerstein representation is motivated by the potential application to welding process, i.e. laser welding process.

3.1 System Identification [30]-[34]

One of the benefits from control application is the operational performance, without which many systems would exhibit an unacceptable natural behavior. Examples of such physical systems are chemical processes, vehicles, aircraft, consumer electronics, power plants, space structures, industrial robots, and so on. While designing a system, engineers always expect a balance with higher performance and lower throughput. However, the enhancement of some performance often implies the reduction of uncertainties and disturbances. Although with such improvements as feedback compensators, system performances will be realized as expected, a mature model built on the enough understanding of the system can never been overestimated.

There are generally two ways to build models. One is through differential equations in accordance with natural laws, such as physics, chemistry, geometry, and so on. The advantage lies at its simplification and detail. The disadvantage is the requirement of understanding of the system. The other method is system identification, which is to develop a model from observed or collected data based on time series analysis and statistical methods. The advantage of system identification is its easiness without the need of knowing too much detail of the system. The disadvantage is its too much calculation and less detail model, which probably cause unstable system.

3.1.1 ARMAM: Autoregressive Moving Average Model [30] –[31]

ARMAM is one of the most widely used statistical based analyses. It can be applied to areas like economics, finance, aerospace, automobile, chemistry, and so on. Fundamentally, the model represents the relation between input and output signals on the basis of previous time series data.

A general ARMAM is:

$$r_t = \phi_0 + \sum_{i=1}^{p} \phi_i \bullet r_{t-i} + a_t + \sum_{i=1}^{q} \theta_i \bullet a_{t-i} \qquad (3.1.1.a)$$

, which is also called as ARMA (p, q) model, with p, q as the order.

To identify the model, the order and the coefficients will have to be estimated with methods such as least square estimation and optimization.

3.1.2 Hammerstein Nonlinear System Identification

It has been noted that many systems can be approximated by a static nonlinearity and a linear dynamics in series with each other. These models are called Hammerstein block cascade models and used to model several classes of nonlinear systems such as chemical processes, heat exchangers, and distillation. Because it is difficult to identify nonlinear parameters, Hammerstein is of no exception. The good about Hammerstein model structure is sometimes we can choose the nonlinear element based on the available input and linear transfer function.

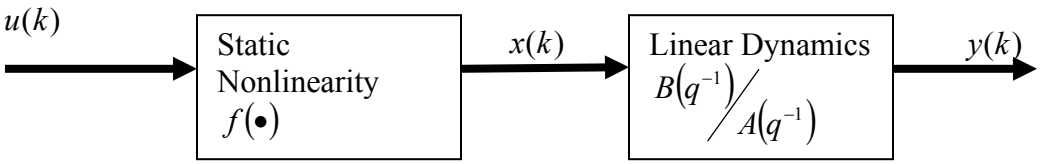


Figure 3.1 Hammerstein Nonlinear Model Structure

The Hammerstein Model can be described by a memoryless nonlinear element followed by a linear dynamic system as shown in Figure 3.1, where $x(k) = f(u(k))$; $u(k), y(k)$ are the input and output of the system respectively and $x(k)$ is the nonlinear function of the

input. $x(k)$ can not be measured but can be eliminated by estimating the nonlinear function.. The static nonlinearity nonlinearizes the input and then the virtual input is sent into the linear system, which is also modeled by a linear transfer function, until the output is sent out. One of the characteristics of Hammerstein model is that the effect of nonlinearity is independent of the linear dynamics. In other words, the slope of the nonlinearity at a certain operating point can be the instantaneous system gain, determining the static response of the system.

If we introduce the shifting operator q that is defined by:

$$y(k-1) = q^{-1}y(k), \tag{3.1.2.a}$$

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \tag{3.1.2.b}$$

$$B(q^{-1}) = b_1 + b_2 q^{-1} + \cdots + b_m q^{-m} \tag{3.1.2.c}$$

Then the system can be represented by the structure of:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})} f((u(k))) \tag{3.1.2.d}$$

If the nonlinear function can be assumed as a polynomial of order l, then the system can be modeled by:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})} \sum_{i=1}^{l} \gamma_i u^i(k) \tag{3.1.2.e}$$

Hammerstein identification has been studied in many years. In 1966, iterative search method based Hammerstein structure was proposed and applied [35]. The model was composed of a memoryless polynomial gain and a linear discrete time system transfer function. Mean least Square error criterion between measured output and predicted output were applied to adjust the parameters of both linear dynamics and the nonlinearity gain. Similar techniques were also used to identify nonlinear system with various format transfer function. A non-iterative technique was used to identify the Hammerstein model in a general case where the transfer function has zeroes [36]. Zhu [6] proposed a aA relaxation iteration scheme was proposed with a model structure in which the error is bilinear in the parameters [37]. The order of the linear dynamics and the nonlinearity are determined by looking at an output error related criterion which is control relevant. A

multivariable Hammerstein identification for state-space and linear-FIR models was proposed. White noise persistent excitation series was taken into consideration [38]. Moreover, they have shown that the linear-FIR case with white noise input, the standard iterative procedure is guaranteed to converge and thus provides an asymptotically optimal estimator. In [39], a frequency weighted least squares based formulation method was presented to identify the parameters of Hammerstein nonlinear continuous time system based on input and noise contaminated output data observed over a finite time interval. Cross-correlation was added into an identification algorithm for the system having the structure of a Hammerstein model was studied and procedures to decouple the identification of the linear dynamics from the characterization of the nonlinear element were also proposed [40]. GA algorithm was used for identification to proximate the piecewise linear dynamics and memoryless nonlinear characterization of the Hammerstein models [41] [42]. To identify the Hammerstein structure, especially the nonlinearity, Genetic programming was applied to the system [43]. A novel approach based on a point-slope parameterization of the static nonlinearity and optimal computing was proposed to simultaneously approximate the linear dynamic and static nonlinear blocks of the nonlinear feedback model [44]. A practical method to identify heat exchange process with Hammerstein model was presented [45]. The model was based on experimental study. Eskinat and Johnson [3] used the identification methods for simulated distillation columns and to an experimental heat exchanger process. To fully investigate the potential of Hammerstein identification, a separable nonlinear least squares methods for on-line estimation of neural nets Hammerstein models was proposed [46]. An algorithm which transforms the nonlinear model into a model which is linear in parameters and a pseudo-inverse technique leading to a consistent estimator or the initial realization as well the model of the noise was studied [47]. A new iterative procedure to identify Hammerstein models and minimizes the infinity norm of the deviation between the true model and identified model [48]. To compare the performance between iterative and non-iterative algorithm, a non-iterative algorithm was studied and the algorithm was applied for simultaneous identification of the linear and nonlinear parts of the multivariate Hammerstein structures. The proposed algorithm is based on LS and SVD [49]. To conquer the over-sampling problem during the identification of Hammerstein

model, a new least squares type of identification algorithm was proposed [50]. Dynamic separation techniques were also studied for a class of Hammerstein models [51]-[53]. Optimal design and recursive techniques were greatly applied to the nonlinear processes with the assumptions. A relaxation based technique and its consistency was proposed and studied for nonlinear processes [54]. Neural network was proved to be another possible method to identify the nonlinearity. A recursive identification algorithm has been derived to update the weights and the parameters of the ARMA [55].

3.2 Nonlinear Control

Nonlinear control is one of the biggest challenges in modern control theory. It has various applications such as Aerospace, Automobile, UAV, Welding, and so on. While linear control system theory has been well developed, trying to control the system beyond the limit of equation point is still a pursuit for control engineers.

There has been many ways related to nonlinear control. Given the fact that linear control theory has been studied so many years, it is very reasonable to think of a way to linearize the nonlinear system model at first, which is, introducing factors to counteract the effect of nonlinear elements. Feedback linearization can be categorized into the field [56]-[57]. Adaptive Control [58]-[59] certainly benefited the progress. However, due the limit of system complexity, especially when the system order is higher and system has unmodeled dynamics, Feedback linearization probably will not have the good result, which forced new method like back-stepping [57], constructive methods [55], and Robust Adaptive method [59]-[61].

3.2.1 Feedback Linearization [56]

The central idea of Feedback Linearization is to cancel the nonlinear part of the system model through controller and convert the system to be linear. Depending on the system model, feedback linearization falls into two categories. One is output feedback linearization and the other state feedback linearization.

Let:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \qquad (3.2.1.a)$$

$$u = \alpha(x) + \beta(x)\upsilon(t) \qquad (3.2.1.b)$$

$$\begin{aligned} \dot{z} &= Az(t) + B\upsilon(t) \\ y &= Cz(t) \end{aligned} \qquad (3.2.1.c)$$

The implementation is to find a feedback transformation as equation 3.2.1.b, so that, equation 3.2.1.c under the conditions that (A, B) is controllable and (A, C) is observable

Something worth mentioning is when the system is full state available, the output and control law will be the focus to decide the successful linearization; when system is not full state available, further consideration should be taken such effect as zero dynamics.

For full state feedback linearization, the task is to find a feedback transformation so that

$$\dot{z} = Az(t) + B\upsilon(t) \qquad (3.2.1.d)$$

Let the system be:

$$y^n = f(x) + b(x)u \qquad (3.2.1.e)$$

, where $b(x) \neq 0$ and x the state vector $\left[y, y^1, y^2, ...y^n\right]^T$

For a tracking system, let reference input $r^n(t)$ exists and bounded, design a stable polynomial $a(s) = (s-\lambda_1)(s-\lambda_2)...(s-\lambda_n) = s^{(n)} + a_{n-1}s^{(n-1)} + ... + a_0$. Then define the tracking error as:

$$\begin{aligned} e &= y - r \\ q &= \left[e, e^{(1)}, ...e^{(n-1)}\right] \end{aligned} \qquad (3.2.1.f)$$

Then choose the control law as

$$u = \frac{r^{(n)} - a^T q - f(x)}{b(x)} \qquad (3.2.1.g)$$

3.2.2 H_∞ Nonlinear Control [56] [62]

Started since late 1980s by McFarlane and Glover, H-infinity control has been one of most popular methods in nonlinear control fields. Fundamentally H-infinity is similar to optimal control as shown in Figure 3.2. It has been successfully developed for different models and proven to be effective for robust systems for substantial uncertainty.

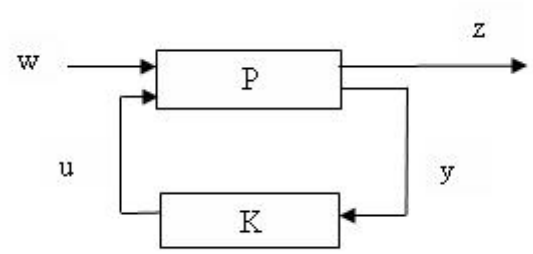


Figure 3.2 H-infinity control

Let the system be:

$$\begin{aligned} \dot{x} &= A(x) + B_1(x)w + B_2(x)u \\ z &= C_1(x) + D_1(x)u \\ y &= C_2(x) + D_2(x)w \end{aligned} \qquad (3.2.2.a),$$

where x is the state, z is the performance objective, y is the measured output and u the controller

If we rewrite the system as:

$$\begin{aligned} \begin{bmatrix} z \\ y \end{bmatrix} &= P\begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}\begin{bmatrix} w \\ u \end{bmatrix} \\ u &= Ky \end{aligned} \qquad (3.2.2.b)$$

, then the performance objective z might be represented with

$$z = \left(P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\right)w \qquad (3.2.2.c)$$

Define $F(P,K)=\left(P_{11}+P_{12}K\left(I-P_{22}K\right)^{-1}P_{21}\right)$ (3.2.2.d)

Then the task of H infinity is to find a controller K such that

$F=\left(P_{11}+P_{12}K\left(I-P_{22}K\right)^{-1}P_{21}\right)$ (3.2.2.e)

Equation 3.2.2.d is minimized following the norm

$\|F\|_{\infty}=\sup_{\omega}\overline{\sigma}(F(j\omega))$, where $\overline{\sigma}$ is the singular value of the matrix F

(3.2.2.f)

3.2.3 Backstepping [56]-[57]

Backstepping has been studied since late 1980s and 1990s due to its potential to dealing with uncertainty. It basically is a recursive method, designed to derive an error equation and construct a control law and parameter adjustment law so that the state of the error equation settle down to zero.

As shown in Figure 3.3., let the system be

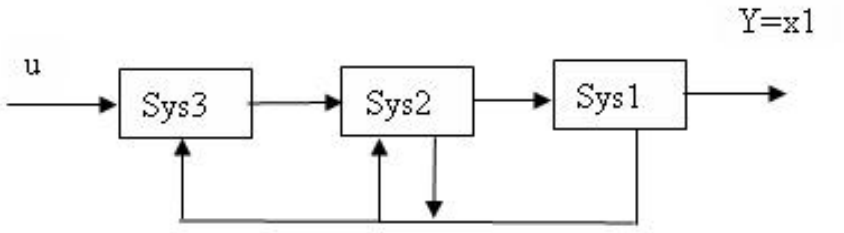


Figure 3.3 Backstepping system diagram

$$\begin{aligned}
&\frac{dx_1}{dt} = f_1(x_1) + x_2 \\
&\frac{dx_2}{dt} = f_2(x_1, x_2) + x_3 \\
&\vdots \\
&\frac{dx_i}{dt} = f_i(x_1, x_2, ...x_i) + x_{i+1} \\
&\vdots \\
&\frac{dx_n}{dt} = f_n(x_1, x_2, ..., x_n) + u \\
&y = x_1
\end{aligned} \tag{3.2.3}$$

(3.2.3.a)

The task is to design the transformation with canceling the nonlinear effects.

First, let:

$$\begin{aligned}
&x_2 = \alpha(x_1) = -f_1(x_1) - \lambda_1 x_1 \\
&z_1 = x_1
\end{aligned} \tag{3.2.3.b}$$

Assume $\lambda_1 > 0$

Define the Lyapunov function as

$$V_1(x_1) = \frac{1}{2} x_1^2 \tag{3.2.3.c}$$

Then:

$$\dot{V}_1 = x_1 \frac{dx_1}{dt} = -\lambda_1 x_1^2 + x_1(x_2 - \alpha_1(x_1)) \tag{3.2.3.d}$$

Let $z_2 = x_2 - \alpha_1(x_1)$ (3.2.3.e)

Then

$$\begin{aligned}
&\frac{dx_1}{dt} = -\lambda_1 x_1 + z_2 \\
&\frac{dz_2}{dt} = f_2(x_1, x_2) - \frac{\partial \alpha_1}{\partial x_1}(f_1(x_1) + x_2) + x_3
\end{aligned} \tag{3.2.3.f}$$

Define

$$\bar{f}_2 = f_2(x_1, x_2) - \frac{\partial \alpha_1}{\partial x_1}\left(f_1(x_1) + x_2\right) \tag{3.2.3.g}$$

Then the Lynapunov candidate is

$$V_2(x_1, z_2) = V_1(x_1) + \frac{1}{2} {z_2}^2 \tag{3.2.3.h}$$

Then

$$\dot{V}_2 = -\lambda_1 {x_1}^2 + x_1 z_2 + z_2\left(\bar{f}_2(x_1, z_2) + x_3\right) \tag{3.2.3.i}$$

So, for x_3, let

$$\alpha_2(x_1, z_2) = -\bar{f}_2(x_1, x_2) - x_1 - \lambda_2 z_2 \tag{3.2.3.j}$$

$$\begin{aligned} &\dot{V}_2 = -\lambda_1 {x_1}^2 - \lambda_2 {z_2}^2 + z_2 z_3 \\ &z_3 = x_3 - \alpha_2 \end{aligned} \tag{3.2.3.k}$$

So, we have

$$\begin{aligned} &\frac{dx_1}{dt} = -\lambda_1 x_1 + z_2 \\ &\frac{dz_2}{dt} = -x_1 - \lambda_2 z_2 + z_3 \\ &\frac{dz_3}{dt} = f_3 - \left(\frac{\partial \alpha_2}{\partial x_1}\dot{x}_1 + \frac{\partial \alpha_2}{\partial z_1}\dot{z}_2\right) + x_4 \end{aligned} \tag{3.2.3.l}$$

Similarly, let:

$$\bar{f}_3 = f_3 - \left(\frac{\partial \alpha_2}{\partial x_1}\dot{x}_1 + \frac{\partial \alpha_2}{\partial z_1}\dot{z}_2\right) \tag{3.2.3.m}$$

Define the lyapunov function:

$$V_3(x_1, z_2, z_3) = V_2(x_1, z_2) + \frac{1}{2} {z_3}^2$$

(3.2.3.n)

So the first derivative of equation 3.2.3.n will be as:

$$\dot{V}_3(x_1, z_2, z_3) = -\lambda_1 x_1^{\,2} - \lambda_2 z_2 + z_2 z_3 + z_3\left(\bar{f}_3 + x_4\right) \tag{3.2.3.o}$$

$$\alpha_3 = -\bar{f}_3 - z_2 - \lambda_3 z_3 \tag{3.2.3.p}$$

Similarly, let

$$\begin{aligned} \frac{dx_1}{dt} &= -\lambda_1 x_1 + z_2 \\ \frac{dz_2}{dt} &= -x_1 - \lambda_2 z_2 + z_3 \\ &\vdots \\ \frac{dz_n}{dt} &= f_n - \left(\frac{\partial \alpha_{n-1}}{\partial x_1}\dot{x}_1 + \frac{\partial \alpha_{n-1}}{\partial z_1}\dot{z}_2 \ldots\right) + u \end{aligned} \tag{3.2.3.q}$$

$$\bar{f}_n = f_n - \left(\frac{\partial \alpha_2}{\partial x_1}\dot{x}_1 + \frac{\partial \alpha_2}{\partial z_1}\dot{z}_2 + \ldots\right) \tag{3.2.3.r}$$

$$V_n(x_1, z_2, \ldots z_n) = V_1(x_1, x_2, \ldots, x_{n-1}) + \frac{1}{2} z_n^{\,2} \tag{3.2.3.s}$$

So:

$$\dot{V}_n(x_1, z_2, z_3) = -\lambda_1 x_1^{\,2} - \sum_{k=2}^{n-1} \lambda_k z_k^2 + z_{n-1} z_n + z_n\left(\bar{f}_n + u\right)$$

$$u = -\bar{f}_n - z_{n-1} - z_{n-1} - \lambda_n z_n \tag{3.2.3.t}$$

Moreover, the closed loop system will be

$$\begin{aligned} \frac{dx_1}{dt} &= -\lambda_1 x_1 + z_2 \\ \frac{dz_2}{dt} &= -x_1 - \lambda_2 z_2 + z_3 \\ &\vdots \\ \frac{dz_n}{dt} &= -z_{n-1} - \lambda_n z_n \end{aligned} \tag{3.2.3.u}$$

The Lyapunov function of equation 3.2.3.u will be:

$$\dot{V}_n(x_1, z_2, z_3) = -\lambda_1 x_1^{\,2} - \sum_{k=2}^{n} \lambda_k z_k^2$$

(3.2.3.v)

3.2.4 Model Based Predictive Control (MPC) [63]-[64]

MPC usually includes the process model which is often a linear discrete system model obtained from experiment, a predication equation which runs forward for a fixed number of time steps to predict the likely process behavior, a known future reference trajectory, and a cost function which usually is quadratic and costs future process output errors. The way how MPC works is basically to minimize the variance of the output from the system at the k th step ahead in the future, where k is the estimated plant dead time.

Let the process model be:

$y(k) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})} u(k-1) + \frac{C(q^{-1})}{D(q^{-1})} \Delta(k)$, where $A(q)$, $B(q)$, $C(q)$, $D(q)$ are polynomials in the backward shift operator q^{-1}, Δ is the Gaussian white noise sequence with zero mean.

Denote t as current time and the design objective is to choose a control $U(t)$:

$$U(t) = Min(E\{y_{t+k} \,|_t^2\}) \qquad (3.2.4.1)$$

And the cost function can be defined as:

$$J_t = E\{[y_{t+k}^2 + \lambda u_t^{\,2}]\} \qquad (3.2.4.2),$$

where λ is a weighting factor.

If we write the cost function:

$$J_t = \sum_{j=0}^{N} [y_{t+j+1}^2 + \lambda_j u_{t+j}^{\,2}] \qquad (3.2.4.3)$$

, then the predictive term can be extended over a range of future time.

3.2.5 Fuzzy Control [65] [65]

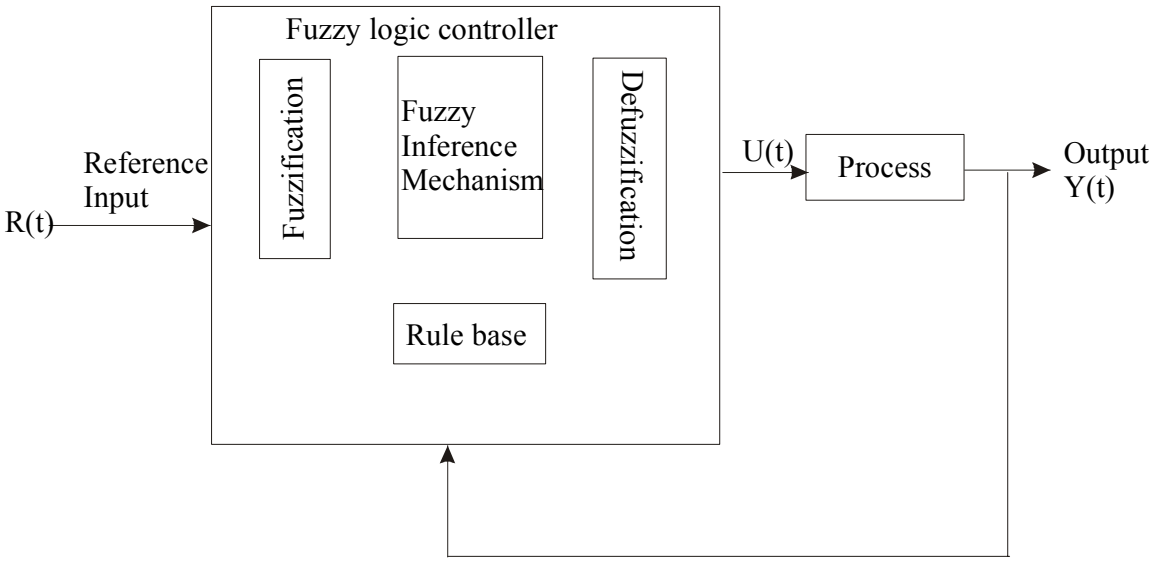


Figure 3.4 Fuzzy Control

A fuzzy controller as shown in Figure 3.4 is designed to roughly emulate the human deductive process as shown in Figure 3.3 and the typical rule base is a set of "if-then" rules that are quantified via fuzzy logic to represent the knowledge that human experts may have about how to solve the problem in their domain of expertise. The fuzzy inference mechanism successively decides what rules are most related to the current situation and applies the action indicated by these rules. The fuzzification interface converts numeric inputs into a form that the fuzzy inference mechanism can use to determine which knowledge in the rule base is most relevant at the current time and generate the fuzzy set of output. The defuzzification interface combines the conclusion reached by the fuzzy inference mechanism and provides the numeric output. The fuzzy logic control provides a formal methodology for representing, manipulating, and implementing the human's heuristic knowledge about how to control a system. Both inputs and outputs are real variables and need to be fuzzified and defuzzified. The important component of a fuzzy control system is the rule base and inference mechanism. The rule base is often implemented based on the linguistic expression of expert knowledge. The inference mechanism can be implemented mathematically.

$$u_i(u_1,u_2,...,u_n) = uA_1^j(u_1) * uA_2^k(u_2) * ... * uA_n^l(u_n)$$

(3.2.5.1)

where $u_i(u_1,u_2,...,u_n)$ is the certainty that the antecedent of rule i match the input information $(u_1,u_2,...,u_n)$, $uA_1^j(u_1)$ is the membership function associated with fuzzy set A_1^j and input u_1.

For i th rule, the computed fuzzy set with membership function is

$$u\hat{B}_q^i(y_q) = u_i(u_1,u_2,...,u_n) * u_{B_q^i}(y_q)$$

(3.2.5.2)

where $uB_q^p(y_q)$ specifies the certainty level of output y_q in fuzzy set B_q^p. The overall implied fuzzy set with membership function:

$$u\hat{B}_q(y_q) = u\hat{B}_q^1(y_q) \oplus u\hat{B}_q^2(y_q) \oplus ... \oplus u\hat{B}_q^R(y_q) \quad (3.2.5.3)$$

, where $\oplus$ defines $x \oplus y = \max\{x, y\}$.

The control system needs to defuzzify the fuzzy set computed from the input to output the numerical value to the plant. There are a few criteria including the Max, center of area (COA), center average, center of gravity (COG), etc. A very common method is COG, in which a crisp output is chosen using the center of the area and area of each implied fuzzy set and is given as follows:

$$y_q^{crisp} = \frac{\sum_{i=1}^{R} c_q^i \int_{yq} u\hat{B}_q^i(y_q) dy_q}{\sum_{i=1}^{R} \int_{yq} u\hat{B}_q^i(y_q) dy_q} \quad (3.2.5.4)$$

, where c_q^i is the center of area of the membership function of B_q^p associated with the implied fuzzy set $\hat{B}_q^i$ for the i th rule, the fuzzy system is defined such that

$$\sum_{i=1}^{R} \int_{yq} u\hat{B}_q^i(y_q) dy_q \neq 0 \text{ for all } u_i. \quad (3.2.5.5)$$

3.2.6 Robust Adaptive Control [59]-[61]

Consider a SISO nonlinear system represented by:

$$y^{(n)} = f(y, \dot{y}, \cdots, y^{(n-1)}, u, \dot{u}, \cdots, u^{(m-1)}) + \theta_u u^{(m)} + \Delta(y, \dot{y}, \cdots, y^{(n-1)}, u, \dot{u}, \cdots, u^{(m-1)}, \omega) + d(t) \tag{3.2.6.1}$$

, where y is output, u is control, y^i is ith derivative, $d(t)$ is unknown bounded disturbance; $\Delta(\ )$ is uncertain nonlinearity and unmodelled dynamics, θ is unknown parameters with known sign.

Let:

$$\begin{aligned} x_1 &= y, x_2 = y^{(1)}, \cdots, x_n = y^{(n-1)} \\ z_1 &= u, z_2 = u^{(1)}, \cdots, z_m = u^{(m-1)} \end{aligned} \tag{3.2.6.2}$$

$$\dot{x}_n = f(x,z) + \theta_u v + \Delta(x,z,\omega) + d(t) \tag{3.2.6.3}$$

$$\dot{x}_i = x_{i+1}, \text{ where } 1 \le i \le n-1 \tag{3.2.6.4}$$

$$\dot{z}_i = z_{i+1}, \text{ where } 1 \le i \le m-1 \tag{3.2.6.5}$$

$$\dot{z}_m = v \tag{3.2.6.6}$$

$$v = u^{(m)} \tag{3.2.6.7}$$

The unmodelled dynamics is described by:

$$\dot{\omega} = q(\omega, x, z) \tag{3.2.6.8}$$

We assume that Δ, q are unknown nonlinear functions which are continuous and Lipschitz and satisfy

$$\Delta(0,z,0) = 0$$

$$\left|\Delta(x,z,\omega)\right| \le c_1\|x\| + c_2\|z\| + c_3\|\omega\| \tag{3.2.6.9}$$

Then the control scheme is:

$$\begin{aligned} e_1 &= x_1 - y_r \\ e_2 &= x_2 - \dot{y}_r \\ &\cdots \\ e_n &= x_n - y_r^{(n-1)} \end{aligned} \tag{3.2.6.10}$$

$$e = [e_1, e_2, \cdots, e_n]^{\mathrm{T}} \tag{3.2.6.11}$$

$$\begin{aligned} \dot{e} &= Ae + b[f(e + \bar{y}_r, z) + \theta_u v - y_r^{(n)} + \Delta(e + \bar{y}_r, z, \omega) + d(t)] \\ \dot{z} &= \bar{A}z + \bar{b}v, \\ \dot{\omega} &= q(\omega, e + \bar{y}_r, z), \end{aligned} \tag{3.2.6.12}$$

$$\dot{e} = A_m e + b[Ke + f(e + \bar{y}_r, z) - y_r^{(n)} + \theta_u v + \Delta(e + \bar{y}_r, z, \omega) + d(t)] \tag{3.2.6.13}$$

We introduce a dynamic signal described by:

$$\dot{r} = -\bar{c}_0 r + r_m, \text{ where } r(0) = r^0 > 0 \text{ and } \bar{c}_0 \in (0, c_0) \tag{3.2.6.14}$$

$r_m = \|e + \bar{y}_r\|^2 \gamma_0 \|e + \bar{y}_r\|^2 + d_0 \overset{\Delta}{=} r_m(e, \bar{y}_r)$ that satisfies with Lyapunov condition

$V_\omega(\omega(t)) \le r(t) + D(t)$, where $D(t)$ is a defined function

The robust adaptive controller is then designed by:

$$\begin{aligned} v &= -\beta\ e^T Pb\{[\bar{f}(e + \bar{y}_r, z)]^2 + \|e + \bar{y}_r\|^2 + \|z\|^2 + [\alpha_1^{-1}(2r)]^2 + (Ke)^2 + 1\} \\ &= v(e, z, r, \bar{y}_r, \beta) \end{aligned} \tag{3.2.6.15}$$

, where β is the adaptive parameter of the controller and α_1^{-1} is the inverse function and is a function of class K_∞.

P is Lyapunov matrix under the condition of:

$PA_m + A_m^T = -Q$, where $Q = Q^T > 0$

$$\dot{\beta} = \beta_m(e, z, r, y_r) - \Gamma\sigma\beta$$

$$\begin{aligned} \beta_m &= \Gamma(e^T Pb)^2 \left\{[f(e + y_r, z)]^2 + \|e + y_r\|^2 + \|z\|^2 + [\alpha^{-1}(2r)]^2 + (Ke)^2 + 1\right\} \\ &= \beta_m(e, r, \bar{y}_r) \end{aligned} \tag{3.2.6.16}$$

For the design constants we assume they are known and satisfy the condition of $T > 0, \sigma > 0$.

In the case that not all the states are available, output feedback control is implemented based on the high gain state observers.

$$\dot{\hat{e}}_1 = \hat{e}_{i+1} + \left(\frac{\sigma_i}{\varepsilon^i}\right)(e_1 - \hat{e}_i) \quad , 1 \le i \le n-1$$
$$\hat{e}_n = \left(\frac{\sigma_n}{\varepsilon^n}\right)(e_1 - \hat{e}_1) \qquad (3.2.6.17)$$

, where $\varepsilon > 0$ is a small constant, $\sigma_i > 0, i = 1,2,...,n$ are chosen so that $A_n = A - K_\sigma C$ is a Hurwitz matrix, and $K_\sigma = [\sigma_1 \quad , \quad \cdots \quad , \quad \sigma_n]$, $C = [1 \quad , \quad \cdots \quad , \quad 0]$.

To eliminate peaking in the implementation of the observer, we define

$$\hat{e}_i = \frac{q_i}{\varepsilon^{i-1}} \text{ ,where } 1 \le i \le n \qquad (3.2.6.18)$$

Thus, the observer can be designed by:

$$\varepsilon\dot{q}_i = q_{i+1} + \sigma_i(e_1 - q_1)$$
$$\varepsilon\dot{q}_n = \sigma_n(e_1 - q_1) \quad \text{, where } 1 \le i \le n-1$$

In order to prevent the peaking from entering the control system, we saturate the control signal and adaptive controller outside of their domains of interests.

And for $v(e, r, \bar{y}_r, \beta), \beta_m(e, r, \bar{y}_r)$, we can denote the following equations:

$$v^S(e, r, \bar{y}_r, \beta) = M_v sat\left(\frac{v(e, r, \bar{y}_r, \beta)}{M_v}\right) \qquad (3.2.6.19)$$

$$\beta_m{}^S(e, r, \bar{y}_r) = M_\beta sat\left(\frac{\beta_m(e, r, \bar{y}_r)}{M_\beta}\right) \qquad (3.2.6.20)$$

$sat(\ )$ represents the saturation function.

Thus the robust adaptive output controller can be obtained by replacing $v(e, r, \bar{y}_r, \beta)$ and $\beta_m(e, r, \bar{y}_r)$ with $v^S(e, r, \bar{y}_r, \beta)$ and $\beta_m{}^S(e, r, \bar{y}_r, \beta)$

CHAPTER 4

VISION BASED SENSOR: WELD POOL IMAGE PROCESSING

The geometry of the molten weld pool is critical to welding process. By equipping a high speed pulse laser with CCD digital camera in our system, grayscale images of weld pool can be acquired. However these images must be smoothed and segmented from the background in order to extract the feature of interest due to the inevitable effects of noises. Accuracy of the smoothing and segmentation has great influence on the subsequent analysis

Image smoothing has been studied in many years. Fundamentally it is to remove noises and enhance the interested regions. Various filters based on convolution masks have been proposed for this purpose [67]-[70]. The side effect of image smoothing is sometimes the resultant image is undesirably blurry, especially at sharp edges.

Segmentation is to distinguish regions that shares common properties in an image. The properties vary with applications of interest and can be such as pixel grayscale, color, etc. Several segmentation methods have been researched and applied in the past years, among which thresholding and edge detection are the mostly mentioned [70]-[77].

The fundamental idea is, if we define $\{P(x,y)\}$ as the local property and $\{S_i(x,y) \mid i = 1,2,...,n\}$ as the regions, then the segmentation of image X should satisfy at least:

$X = \bigcup_{i=1}^{n} S_i$, where $S_i \neq \Phi$ and Φ is empty.

$S_i \bigcap S_j = \Phi, \forall i \neq j$

$\{P(x,y) \mid (x,y) \in S_i\} = 1$

$\{P(x,y) \mid (x,y) \in (S_i \bigcup S_j)\} = 0$

In our research, only monochrome images are considered although all the methods and principles related can be applied to color images as well.

4.1 Image smoothing: Pre-conditioning

Image acquired with devices such as CCD or CMOS digital camera inevitably generates noises and blurring. The blurring can be caused by the movement during the capturing process, out-of-focus, short exposure time, or light distortion, etc. To get better results in later processing, it is better to smooth these raw images. The fundamental process can be either image deblur or noise removal or both [77]. The methods can be implemented with various filters. Depending on the situation, it might need to repeat debluring several times to get satisfactory results.

4.1.1 Deblur

Mathematically, a blurred image can be represented by a convolution between original image and distortion operator, called PSF, point spread function [70] [71].

$g = H * f + n$, where g is the blurred image, H is the distortion, and n is the noise

$$(H * f)[m,n] = \sum_{i=1}^{m}\sum_{j=1}^{n} h(i,j) f(m-i, n-j)$$

The PSF convolution kernel can be Gaussian Shaped, with the form of

$$PSF(x,y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.1.1}$$

Several algorithms can be used to deblur the raw images. The algorithms can be either linear or non-linear, either iterative or non-iterative.

4.1.1.1 Lucy-Richardson Algorithm [77 [78]

This algorithm requires no priori information about the original image. It performs better when the noise is not too strong.

$g_i = \sum_{j=1}^{n} h_{ij} f_j$, where h_{ij} is the PSF at position i and f_j is the pixel grayscale at location j

$$f_i^{t+1} = f_i^t \sum_{j=1}^{n} \frac{h_{ji} g_j}{\sum_{i=1}^{m} h_{ji} f_i}, \text{ or } f_i^{t+1} = f_i^t H * \left(\frac{g}{Hf^t}\right) \tag{4.1.2}$$

The solution can be iteratively obtained.

4.1.1.2 Maximum Likelihood Estimator [78]-[79]

This algorithm computes the maximum likelihood estimate of the intensity of a Poison process. Let f be the unblurred image, the expected value at the kth pixel in the blurred image is $\sum_{j=1}^{n} h_{kj} f_j$. The actual kth pixel value g_k is viewed as simply one realization of a Poisson distribution with mean $\sum_{j=1}^{n} h_{kj} f_j$.

Accordingly,

$$p(g_k \mid f) = \frac{e^{-\sum_{j=1}^{n} h_{kj} f_j} \left(\sum_{j=1}^{n} h_{kj} f_j \right) g_k}{g_k!} \tag{4.1.3}$$

$$p(g \mid f) = \prod_{k=1}^{m} \frac{e^{-\sum_{j=1}^{n} h_{kj} f_j} \left(\sum_{j=1}^{n} h_{kj} f_j \right) g_k}{g_k!} \tag{4.1.4}$$

The maximum likelihood solution satisfies $f_i = f_i \sum_{k=1}^{n} \frac{h_{ki} g_k}{\sum_{j=1}^{m} h_{kj} f_j}$ (4.1.5)

, similar to that of Lucy Richardson algorithm

4.1.1.3 Wiener Filter [69] [70] [77]

Wiener filter is a linear filter. The filter assumes that the image is distorted by Gaussian distribution noise. The filter tries to minimize the mean square error between the acquired image and restored image. It operates in the frequency domain.

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \tag{4.1.6}$$

Hence, in regular case,

$\hat{F}(u,v) = G(u,v)T(u,v)$, where $T(u,v) = H^{-1}(u,v)$

$$\hat{F}(u,v) = G(u,v)H^{-1}(u,v) + N(u,v)H^{-1}(u,v) \tag{4.1.7}$$

This way is very easily influenced by the noise because the inverse filter typically has very high gain at high frequencies so that the noise term completely dominates the restored image. In this case, future attempt should be taken to reduce the noise amplification.

The Wiener filter minimizes the mean-square-error:

$$e^2 = E\left[\left(f(x,y) - \hat{f}(x,y)\right)^2\right] \tag{4.1.8}$$

$$\hat{F}(u,v) = G(u,v)\frac{W(u,v)}{H(u,v)}, \text{ where } W(u,v) = \frac{P_H(u,v)}{P_H(u,v) + P_N(u,v)} \tag{4.1.9}$$

$$T(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + S\eta(u,v) \Big/ S_f(u,v)} \tag{4.1.10}$$

, where $S_\eta(u,v), S_f(u,v)$ are the power spectra of the noise and the original image.

4.1.2 Denoise

Digital images are prone to variety of types of noise, as the result of error in the process of image acquisition. When image is acquired directly in a digital format, the mechanism for gathering the data such as CCD detector can easily introduce noise. Several methods are available for noise removal with suitable filters.

4.1.2.1 Linear Filter: Statistical Average [69] [70] [78]

The output image is based on a local averaging of input filter. The idea is to replace each pixel value with the average of its neighbors including itself, defined by the filter window.

$$h[j,k] = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{4.1.11}$$

The filter is normalized so that $\sum h[j,k]=1$. This is to guarantee the signal energy preserved before and after the convolution.

4.1.2.2 Gaussian Filter [69] [70]

Image is convolved with Gaussian mask whose parameters are determined by Gaussian function:

$$h(x,y)=\frac{1}{2\pi\sigma}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)=h_x \bullet h_y \tag{4.1.12}$$

4.1.2.3 Median Filter [69] [70] [78]

Median filter replaces the pixel value with the median value of the neighboring pixel and itself.

$$\begin{bmatrix} ... & ... & ... & ... & ... \\ ... & 124 & 126 & 127 & ... \\ ... & 120 & 150 & 125 & ... \\ ... & 115 & 119 & 123 & ... \\ ... & ... & ... & ... & ... \end{bmatrix} \tag{4.1.13}$$

, where the neighboring values are $\{115,119,120,123,124,125,126,127,150\}$ and the man value is 124

4.2 Thresholding

Thresholding is the simplest and most fundamental segmentation method. It has been studied for many years and various algorithms have been proposed [69]-[721]. The idea of thresholding is to select a level of grayscale and partition the image into two sub-classes that include pixels with intensities below and above the threshold level. Accordingly, thresholding can be classified into two categories: global algorithm and local adaptive algorithm based on how the threshold level is selected. In the global algorithm, one threshold is selected across the whole image. In the local adaptive

algorithm, the image is partitioned into sub-images and threshold level is selected in according with the local property.

Define $\{f, g \mid (x, y)\}$ as the grayscale function of the image before and after the threshold and (x, y) is the location of the pixel and T is the threshold level. Then after thresholding, the image will be:

$$g(x,y) = \begin{cases} 0 & f(x,y) < T \\ 1 & f(x,y) \geq T \end{cases} \tag{4.2.1}$$

It is obvious that the threshold value is the key parameter in thresholding. There are many existing methods to selecting threshold. Common methods include statistical mean and histogram based [70] [80], iterative methods [81]-[82], optimal procedure [84], entropy based [85], fuzzy [86] and neural network [87].

4.2.1 Global Threshold: Mean Value [70]

$$T = mean(f(x,y)). \tag{4.2.2}$$

This method is very simple, but inevitably influenced by the noises. It performs well when the regions distinct very sharply, especially when the histogram only has one or two peaks.

4.2.2 Histogram [71]

The valley between the two peaks is one good point to start. For the grayscale $0 \leq z \leq L-1$, the valley threshold value z^* should satisfy $h(z^*) < h(z^* - 1)$ and $h(z^*) < h(z^* + 1)$, where $h(z)$ is the histogram.

4.2.3 Iterative Threshold [72]

At first, select an initial estimate of threshold level at T_0, for example, the average intensity of the all the pixels.

Then partition the image into two sub-classes $\{C_1, C_2\}$, such that $\{C_1 < T_0, C_2 > T_0\}$.

Calculate the mean intensity values $\{\mu_1, \mu_2\}$ of the two sub-classes.

Select a new threshold at $T = \frac{1}{2}(\mu_1 + \mu_2)$.

Repeat procedures until T converges.

4.2.4 Optimal Threshold (Otsu's method) [74]

Otsu's method chooses the threshold to minimize the weighted between-class variance of the pixels during the thresholding.

Assume the grayscale level of the image is $i = \{1,2,\cdots,m\}$, the weighted between-class variance is represented by:

$\sigma_w{}^2(t) = q_1(t)\sigma_1{}^2(t) + q_2(T_k)\sigma_2{}^2(t)$, where the class probabilities are estimated as:

$q_1 = \sum_{i=1}^{t} P(i)$ and $q_2 = \sum_{i=t+1}^{m} P(i)$.

The means of the classes are given by:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \tag{4.2.3}$$

$$\mu_2(t) = \sum_{i=t+1}^{m} \frac{iP(i)}{q_2(t)} \tag{4.2.4}$$

, where $P(i) = \frac{n_i}{N}$, N is the total pixel number and n_i is the number of pixels in level i

The individual variances of the classes are represented by:

$$\sigma_1{}^2(t) = \sum_{i=1}^{t} (i - \mu_1(t))^2 \frac{P(i)}{q_1(t)} \tag{4.2.5}$$

$$\sigma_2{}^2(t) = \sum_{i=t+1}^{m} (i - \mu_2(t))^2 \frac{P(i)}{q_2(t)}. \tag{4.2.6}$$

The algorithm scans the full range of t within $[1,256]$ until it maximizes $\sigma_w(t)$.

4.2.5 Adaptive Threshold [69]

This method is realized in two steps:

At first, the image is partitioned into sub-images $\{S_1, S_2, ..., S_m\}$. The size of the sub-images can be determined by the histogram. In our case, we apply the size at 50×50. For every sub-image $\{S_i, i = 1,2,..., m\}$, the threshold is selected in accordance with the statistical mean $T_i = mean(S_i), i = 1,2,..., m$. The images are then thresholded according to the local threshold values $T = \{T_1, T_2, ..., T_m\}$.

4.2.6 Least Mean Error [80]

Assume the proportion of the particle pixel corresponding to the whole image is θ and the background is $1-\theta$, then the probability of the particle can be represented by:

$$p(z) = \theta p_1(z) + (1-\theta) p_i(z) \tag{4.2.7}$$

At any level T, the probability to detect target as background is:

$$E_1(T) = \int_T^{\infty} p_1(z) dz \tag{4.2.8}$$

And the probably to detect the background as target:

$$E_2(T) = \int_{-\infty}^{T} p_2(z) dz \tag{4.2.9}$$

So, total probability is:

$$E(T) = \theta E_1(T) + (1-\theta) E_2(T) \tag{4.2.10}$$

Let:

$$\frac{\partial E(T)}{\partial T} = 0 \tag{4.2.11}$$

Then:

$$-\theta p_1(T) + (1-\theta) p_2(T) = 0. \tag{4.2.12}$$

$$\ln \frac{\theta \sigma_2}{(1-\theta)\sigma_1} - \frac{(T-\mu_1)^2}{2\sigma_1^2} = \frac{-(T-\mu_2)^2}{2\sigma_2^2} \tag{4.2.12}$$

Suppose the proportion satisfies Gaussian distribution:

$$p_1(z) \sim N(\mu_1, \sigma_1) \tag{4.2.14}$$

$$p_2(z) \sim N(\mu_2, \sigma_2) \tag{4.2.15}$$

Then, if

$$\sigma_1^2 = \sigma_2^2 = \sigma^2 \tag{4.2.16}$$

$$T = \frac{\mu + \mu_2}{2} + \frac{\sigma^2}{\mu_2 - \mu} \ln \frac{\theta}{1-\theta}. \tag{4.2.17}$$

4.2.7 Center of Mass [70]

The threshold level is determined in accordance with the location of the center of mass, which is:

$$Center\_X = \frac{\sum_{i=1}^{m} xf(x,y)}{\sum_{i=1}^{m} f(x,y)} \tag{4.2.18}$$

$$Center\_Y = \frac{\sum_{i=1}^{m} yf(x,y)}{\sum_{i=1}^{m} f(x,y)} \tag{4.2.19}$$

4.2.8 Fuzzy C-Mean [86]

Let C be an integer between $[1,n]$ and $\{x_1, x_2, ..., x_n\}$ denote a set of column vectors in R^p, where p represents the number of features in each vector. Given X, we say that C fuzzy subsets $\{X \to [0,1]\}$ are a fuzzy C partition of X if the following conditions on the membership value u_{ik} for the cluster i and the feature vector x_k are satisfied:

$u_{ik} = u_i(x_k) \mid (1 \le i \le C, 1 \le k \le n)$, where $0 \le u_{ik} \le 1, \forall i, k$

$$\sum_{i=1}^{C} u_{ik} = 1, \forall k \tag{4.2.20}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n, \forall i$$

Grouping pixels blocks according to their membership by regions yields $U = [u_{ik}], 1 \le i \le c, 1 \le k \le n$, where U denotes the fuzzy membership matrix of pixel block k in region i and c denotes the number of regions. The matrix thus becomes:

$$\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{bmatrix} \tag{4.2.21}$$

Each row of matrix U represents the membership grade of all blocks belonging to region R_i

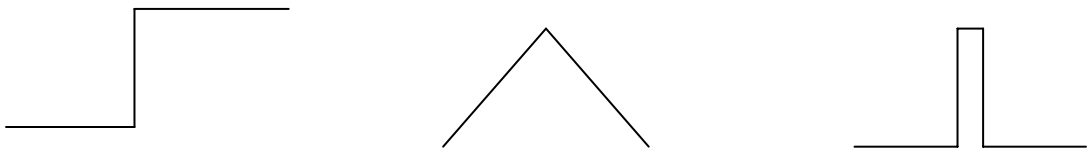
The method minimizes the membership function:

$$J(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m(x_k - v_i) \tag{4.2.22}$$

, where $V = [v_1, v_2, ..., v_c]$ holds the vectors representing the center of each region

4.3 Edge Detection: the Goal

Edge detection is another common method for image segmentation [49]-[53]. It relies on discontinuities in the image data to locate boundaries of the segments and is the fundamental feature of the image. Because edge functions like high frequency signals, it can be recognized by mask filters. Considering the tendency of the pixel grayscale, there are three different edges. The first is Step-edge, in which grayscale jumps from one level to a higher one. The second is Roof-edge, in which grayscale gradually increases to some extent and then decreases. The third is Line-Edge, in which grayscale changes from one level to the other and then back forth. Step edge is mostly used and studied.



Step Edge Roof Edge Line Edge

Figure 4.1 Edge Curve

Various edge detection methods have been studied. Common methods vary from gradient and zero-cross of second order derivatives to statistics. The detection has been

improved from open to close loop. Since edge can be described as the discontinuity of the signals, and especially given the relationship between the maximum point or spot and the signal first differential or second differential, it is reasonable to determine the edge with finding the signal maximums. On the other hand, considering the fact that derivatives can be approximated by numerical methods, if derivatives happen right at every pixel, then the derivatives can be represented similarly like a discrete equation.

According Taylor series expansion, if expanding $f(x+\Delta x)$ about x, then

$$f(x+\Delta x)=f(x)+\Delta x f'(x)+\frac{(\Delta x)^2}{2!}f''(x)+... \tag{4.3.1}$$

Then:

$$f'(x)=\frac{f(x+\Delta x)-f(x)}{\Delta x}+o\left((\Delta x)^2\right) \tag{4.3.2}$$

Or:

$$f'(x)=\frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x}+o(\Delta x) \tag{4.3.3}$$

Let $f(x,y)$ be the image signal distribution related to attributes such as intensity, lamination, and grayscale, with x, y the pixel location in the image.

Then the gradient of the signal can be calculated through the gradient:

$$\nabla f(x,y)=\left(\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right) \tag{4.3.4}$$

Similarly, we can get

$$\|\nabla f\|=\sqrt{\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2} \tag{4.3.5}$$

$$\theta=\tan^{-1}\left(\frac{\partial f / \partial y}{\partial f / \partial x}\right) \tag{4.3.6}$$

For approximation, the gradient can be simplified with:

$$\nabla f(x,y) = \left(f(x,y) - f(x-1,y), f(x,y) - f(x,y-1)\right) \tag{4.3.7}$$

, given Δx as the pixel distance

Or,

$$\begin{aligned} \nabla f_x(x,y) &= f(x,y) - f(x-1,y) \\ \nabla f_y(x,y) &= f(x,y) - f(x,y-1) \end{aligned} \tag{4.3.8}$$

The existent issue of above approximation is that the gradient approximation is actually at the point $(x-1/2, y-1/2)$, not at (x,y). As a result, the edge location would be shifted by one half of a pixel. So, a better choice for the approximation might be:

$$\nabla f(x,y) = \left(f(x+1,y) - f(x-1,y), f(x,y+1) - f(x,y-1)\right) \tag{4.3.9}$$

Or,

$$\begin{aligned} \nabla f_x(x,y) &= f(x+1,y) - f(x-1,y) \\ \nabla f_y(x,y) &= f(x,y+1) - f(x,y-1) \end{aligned} \tag{4.3.10}$$

Whatever the operator is used to approximate the gradient, the resultant vector will contain the information related to the edge. The magnitude of the vector represents the strength of the edge. Its angular direction shows the path of the edge change.

Simple methods implemented horizontally or vertically such as Prewitt or Sobel [70] gradient operator based on convolution masks are to calculate gradient magnitude and direction to locate the edge. Canny [68] method aims to optimally find the edge points with hysterisis tracking and two level thresholds. Laplacian [90] of Gaussian zero-crossing measures the second spatial derivative of the image function to locate the peak of intensity. Standard techniques also include wavelet analysis [91]-[93] and Hough Transformation [48]

4.3.1 Multi-Scale Edge Detection via Local Normal Maxima [91]-[93]

Define two wavelet functions along x and y :

$$\phi^x(x,y)=\frac{\partial\theta(x,y)}{\partial x}\text{, and }\phi^y(x,y)=\frac{\partial\theta(x,y)}{\partial y} \tag{4.3.11}$$

, where $\theta(x,y)$ is a smoothing function.

The signal has to satisfy the conditions:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi^x(x,y)dxdy=0 \tag{4.3.12}$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi^y(x,y)dxdy=0 \tag{4.3.13}$$

If the smoothing function is Gaussian distribution:

$$\theta(x,y)=\frac{1}{2\pi\sigma}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.3.14}$$

Then we have:

$$\phi^x(x,y)=\frac{1}{2\pi\sigma^4}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.3.15}$$

$$\phi^y(x,y)=\frac{1}{2\pi\sigma^4}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.3.16}$$

Suppose the image function:

$f(x,y\in L^2(R))$ satisfies with the condition:

$$|f(x,y)-f(x_0,y_0)|\le K\left(|x-x_0|^2+|y-y_0|\right)^{\alpha/2} \tag{4.3.17}$$

, where K,α are constants and L means Lipschitz.

Choose multi-scale $\left(x/2^i,y/2^i\right)$ along x and y, then

$$\begin{bmatrix}W_{2^i}{}^x f(x,y)\\ W_{2^i}{}^y f(x,y)\end{bmatrix}=2^i\begin{bmatrix}f*\phi_{2^i}{}^x\\ f*\phi_{2^i}{}^y\end{bmatrix}=2^i\begin{bmatrix}\frac{d}{dx}\left(f*\theta_{2^i}\right)\\ \frac{d}{dy}\left(f*\theta_{2^i}\right)\end{bmatrix}=2^i\nabla\left(f*\theta_{2^i}\right) \tag{4.3.18}$$

The norm of the gradient is represented by:

$$M_{2^i}f(x,y)=\sqrt{\left|W_{2^i}{}^x\right|^2+\left|W_{2^i}{}^y\right|^2} \tag{4.3.19}$$

The edge point can be determined by searching the local maxima of the gradient.

4.3.2 Roberts [69]-[71]

L. G. Robert started to study on edge detection since 1960s. He introduced the method of Robert Operator.

Robert Operator is to calculate the maximum of all the gradient vectors and then filter by threshold, which is,

$$G(i,j)=\sqrt{\left(f(i,j)-f(i+1,j+1)\right)^2+\left(f(i,j+1)-f(i+1,j)\right)^2} \tag{4.3.20}$$

The implementation is to use two 2×2 filter masks to convolve the signal matrix and the masks can be written as:

$$G_x=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} \tag{4.3.21}$$

$$G_y=\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix} \tag{4.3.22}$$

Let $f(x,y)$ be the grayscale function of the image. The edge can be located by gradient:

$$\nabla f=\left(G_x,G_y\right)=\left(\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right). \tag{4.3.23}$$

The norm of the gradient G is calculated by:

$$\|\nabla f\|=\sqrt{|G_x|^2+|G_y|^2} \tag{4.3.24}$$

$$\|\nabla f\|=\sqrt{|f(i,j)-f(i+1,j+1)|^2+|f(i,j+1)-f(i+1,j)|^2} \tag{4.3.25}$$

4.3.3 Laplacian [69] –[71]

The idea is from the zero cross of second order derivatives, where the signal curve has the maximum or minimum.

Define the function $f(x,y)$, then the Laplacian transform will be

$$\nabla^2 f=\left(\frac{\partial}{\partial x^2}+\frac{\partial}{\partial y^2}\right)f(x,y)=\frac{\partial f(x,y)}{\partial x^2}+\frac{\partial f(x,y)}{\partial y^2} \tag{4.3.26}$$

The common mask is

$$L = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{4.3.27}$$

And

$$L = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{4.3.28}$$

4.3.4 Prewitt [69]-[71]

Prewitt studied the convolution effect of 3×3 mask in 1970s. The idea is similar to Robert Operator

$$P_H = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.3.29}$$

$$P_V = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.3.30}$$

4.3.5 Sobel [69]-[71]

A 3×3 mask is used by convolution to approximate the partial derivatives and determine the local maximum points.

$$S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$
$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{4.3.31}$$

A simple analysis on Prewitt and Sobel operators:

Assume the grayscale distribution satisfies the formula like

$M_{x,y} = \alpha x + \beta y + \gamma$, with gradient as (α, β)

Then the pixel will be like

$$\begin{pmatrix} -\alpha-\beta+\gamma & -\alpha+\gamma & -\alpha+\beta+\gamma \\ -\beta+\gamma & \gamma & \beta+\gamma \\ \alpha-\beta+\gamma & \alpha+\gamma & \alpha+\beta+\gamma \end{pmatrix} \tag{4.3.32}$$

If we define the operators as

$$\begin{pmatrix} -a & -b & -a \\ 0 & 0 & 0 \\ a & b & a \end{pmatrix} \text{and} \begin{pmatrix} -a & 0 & a \\ -b & 0 & b \\ a & 0 & a \end{pmatrix} \tag{4.3.33}$$

It is easy to see the result after the convolution

$$\begin{aligned} g_x &= 2\beta(2a+b) \\ g_y &= 2\alpha(2a+b) \end{aligned} \tag{4.3.34}$$

Then the gradient will be

$$G = \sqrt{g_x{}^2 + g_y{}^2} = 2(2a+b)\sqrt{\alpha^2+\beta^2} \tag{4.3.35}$$

So, $2(2a+b)=1$

Different values of a and b will determine the Prewitt and Sobel Operators

In Prewitt, a=b=1/6. in Sobel, a=1/8 and b=1/4

How to determine the best case to apply Prewitt or Sobel operators depends on the image signal noises, although there are actually no strict regulations on its applications.

4.3.6 Canny [89]

This is one of the most nominal methods till present. It was introduced by John Canny in 1986. It follows Canny criteria including Good Detection, Localization, and single response. Canny method has been studied and developed after its origin. To realize good detection requires max SNR (signal noise ratio), which is

$$SNR = \frac{H_G}{H_n} = \frac{\left|\int_{-\pi}^{\pi} G(-x)f(x)dx\right|}{n_0\left|\int_{-\pi}^{\pi} f^2(x)dx\right|^{1/2}} \tag{4.3.36}$$

, where n_0 is the square mean root of the unit length noise amplitude.

Let $f(x)$ be the impulse response of the filter, with $x \in [-W, W]$ and $G(x)$ the edge function, $n(x)$ the noise,

Good localization requires the expectation

$$E\left(x_0^{\,2}\right)=\frac{E\left[H_n^{''}(x_0)^2\right]}{\left(H_G^{''}(0)\right)^2}=\frac{n_0^{\,2}\int_{-\pi}^{\pi}f^{'2}(x)dx}{\left[\int_{-\pi}^{\pi}G^{'}(-x)f^{'}(x)dx\right]} \tag{4.3.37}$$

$$Loc(f)=\frac{\left|\int_{-\pi}^{\pi}G^{'}(-x)f(x)dx\right|}{n_0\left|\int_{-\pi}^{\pi}f^{'2}(x)dx\right|} \tag{4.3.38}$$

The object will be to find a function

$f(x)$, with the performance index:

$$J=\frac{\left|\int_{-\pi}^{\pi}G(-x)f(x)dx\right|\left|\int_{-\pi}^{\pi}G^{'}(-x)f^{'}(x)dx\right|}{n_0\left|\int_{-\pi}^{\pi}f^{2}(x)dx\right|^{1/2}n_0\int_{-\pi}^{\pi}f^{'2}(x)dx} \tag{4.3.39}$$

The criteria of the single response mean that the implementation should eliminate multiple possible responses to an edge and determine the max points of edge.

One of the advantages of Canny method is its dual thresholds technique. Threshold level can be achieved by signal histogram. Matlab uses 07-0.8. The method greatest avoids the effects of noise and error. When implementing, a higher threshold level at first is calculated by the histogram and then the lower threshold level will be scaled down by 0.4.

The difficulty to the edge detection is the width of the filter mask. The wider it is, the better the performance is. The process is multi-stage.

At first, a 2D Gaussian filter mask is convoluted to get rid of noise, for example

$$\frac{1}{115}\begin{bmatrix}2&4&5&4&2\\4&9&12&9&4\\5&12&15&12&5\\4&9&12&9&4\\2&4&5&4&2\end{bmatrix} \tag{4.3.40}$$

Then Prewitt, or Sobel Operators are used to approximate the directional gradient

$|G|=\sqrt{G_x^2+G_y^2}$, with the angle $\theta=\arctan\left(\frac{G_x}{G_y}\right)$

As a result, the edge direction will be determined. With aid of specific procedures such as non-maximum suppression, streak elimination, and threshold filtering, the edge will be achieved in the end.

CHAPTER 5

HAMMERSTEIN NONLINEAR IDENTIFICATION: PROPOSED

5.1 Theoretical Background

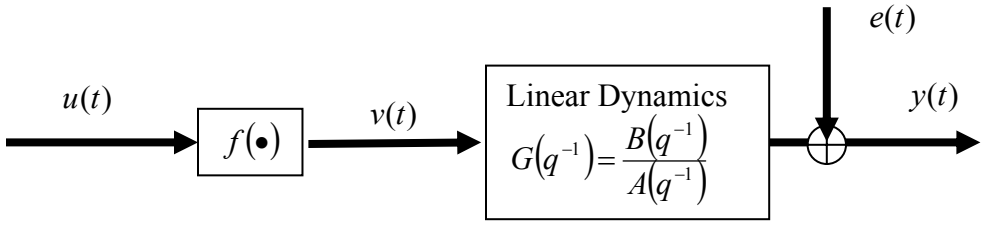


Figure 5.1 Nonlinear Hammerstein model structure

We consider a discrete time SISO (Single-input-single-output) Hammerstein system as shown above, where $u(k), y(k)$ are the system input and output respectively. The intermediate signal $v(k)$ is not measurable but can be eliminated by the nonlinear function $f(\bullet)$; $G(q)$ is the linear dynamic transfer function of the system represented by $G(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})}$ and q is the delay operator for the difference equation and satisfies with:

$$q^{-1}y(k) = y(k-1) \tag{5.1}$$

Without loss of generality, we assume the nonlinearity is a polynomial represented by:

$$f(\bullet) = \sum_{l=0}^{p} c_l u^l \tag{5.2}$$

And the intermediate stat v can be written by:

$$v(t) = f(u(t)) \tag{5.3}$$

Accordingly, the Hammerstein model can be written by a difference equation:

$$A(q^{-1})y(k) = B(q^{-1})f(u(k)) + e(k), \text{ where} \tag{5.4}$$

$$A(q^{-1}) = \sum_{i=0}^{n} a_i q^{-i} = a_0 + a_1 q^{-1} + \cdots + a_n q^{-n} \tag{5.5}$$

$$B(q^{-1}) = \sum_{i=0}^{m} b_i q^{-i} = b_0 + b_1 q^{-1} + \cdots + b_m q^{-m} \tag{5.6}$$

$$a_0 = 1 \tag{5.7}$$

$$f(u) = \sum_{l=0}^{p} c_l u^l = c_0 + c_1 u + \cdots + c_p u^p \tag{5.8}$$

Accordingly, the linear dynamic can be written by the transfer function:

$G(z^{-1}) = \dfrac{B(z^{-1})}{A(z^{-1})} = \dfrac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}$, where z is the operator of Z transform

With simple substitution, the Hammerstein model is rewritten by: (5.9)

$$y(k) = -\sum_{i=1}^{n} a_i y(k-i) + \sum_{j=0}^{m} b_j v(k-j) + e(k) \tag{5.10}$$

Or

$$y(k) = -\sum_{i=1}^{n} a_i y(k-i) + \sum_{j=0}^{m} b_j \sum_{l=0}^{p} c_l u^p (k-j) + e(k) \tag{5.11}$$

If we convert it into matrix form, then we have:

$$y(k) = \phi^T \theta \tag{5.12}$$

$$\phi^T = \begin{bmatrix} -y(k-1) & \cdots & -y(k-n) & u(k) & \cdots & u(k-m) & u^p(k) & \cdots & u^p(k-m) \end{bmatrix} \tag{5.13}$$

$$\theta^T = \begin{bmatrix} a_1 & \cdots & a_n & b_0 c_0 & \cdots & b_m c_0 & b_0 c_p & \cdots & b_m c_p \end{bmatrix} \tag{5.14}$$

Let

$$w = bc \tag{5.15}$$

And rewrite the parameter

$$\theta^T = \begin{bmatrix} a_1 & \cdots & a_n & w_{00} & \cdots & w_{m0} & w_{0p} & \cdots & w_{mp} \end{bmatrix} \tag{5.16}$$

Accordingly the Hammerstein model identification becomes to find the parameter θ. This can be obtained using Least Square method based on the data pairs of input and output. It is easy to see that, if the static gain of the linear dynamics is unity, the system has the property of:

$$G\left(z^{-1}\right)|_{t\to\infty} = G|_{z=1} = \frac{b_0 + b_1 + \cdots + b_m}{1 + a_1 + \cdots + a_n} \tag{5.17}$$

This is true especially when the system static response is mostly determined by the nonlinearity function.

Let,

$$\sum_{j=0}^{m} w_{jl} = \left(b_0 + b_1 + \cdots + b_m\right)c_l \text{, where } l = 1,2,\cdots,p \tag{5.18}$$

Then,

$$c_l = \frac{\sum_{j=0}^{m} w_{jl}}{b_0 + b_1 + \cdots + b_m} \tag{5.19}$$

Or,

$$c_l = \frac{\sum_{j=0}^{m} w_{jl}}{1 + a_0 + a_1 + \cdots + a_n} \text{, where } l = 1,2,\cdots,p \tag{5.20}$$

And

$$b_j = \frac{w_{jl}}{c_l} \text{, where } j = 1,2,\cdots,m \tag{5.21}$$

So, the parameters of the nonlinearity and linear dynamics can be also obtained.

5.2 Conditions for the Method

Just like it is very challenging to identify any nonlinear system, Hammerstein model is of no exception. Obviously, the higher the order of the linear dynamics, which means the more the zeros and the poles, the more complex the system model will be. Also, sometimes, it is very difficult to find a suitable format for the nonlinearity function to complete the Hammerstein structure. For the control purpose in the future study, we certainly expect that the model of low enough model and simpler nonlinearity. To this end, we further make following assumptions

Assumption 1: Zero of noise

For simplicity of identification procedure, we assume the noise if zero.

Assumption 2: Excitation signal input

The excitation signal $u(k)$ of the system is a stationary normally distributed sequence over the entire range of the input signal. The input signal is of persistent excitation and randomly distributed multilevel amplitudes. As a result, the model can be estimated by the data set of:

$\{u(k), y(k)\}, k = 1,2,\cdots, N$

Assumption 3: no zero in the linear dynamic

Apparently the more the amount of the zeroes in the system, the more difficult the identification procedure will be. And if the model is too difficult and complex, it will be no use for further control design. It is known that the dynamic characteristics of a linear system are dominated mainly by the poles. And the poles are actually the roots of the characteristic equation, which is the denominator of the linear dynamics in the Hammerstein model. Moreover, typical laser welding process contains no unstable zeros. Accordingly, it is reasonable and feasible to simplify the linear dynamics as:

$$\frac{y(z)}{x(z)} = \frac{b}{a_0 z^n + a_1 z^{n-1} + \cdots + a_n} \tag{5.22}$$

And for the condition of unity static gain, b might satisfy with:

$$1 = \frac{b}{1 + a_1 + \cdots + a_n}, \tag{5.23}$$

Or

$$b = 1 + a_1 + a_2 + \cdots + a_n \tag{5.24}$$

One thing to note here is that sometimes this relation among coefficients might not be necessarily satisfied because of calculation error.

Assumption 3 is also true under the assumption of 2 when the input signal is a random series with bounded amplitudes during every time interval.

5.3 Simulation

To show the feasibility of approximating the linear dynamics with non zero model, we give following simulation.

We assume the linear dynamic subsystem can be represented by:

$y(k) - 1.0669 y(k-1) + 0.2275 y(k-2) = 0.1u(k) + 0.06u(k-1) + 0.0006u(k-2)$

Or,

$$G(z^{-1}) = \frac{0.1 + 0.06z^{-1} + 0.0006z^{-2}}{1 + 1.0669z^{-1} - 0.2275z^{-2}} \tag{5.25}$$

According to the analysis above, we simply the model as:

$$y(k) - 1.0669y(k-1) + 0.2275y(k-2) = 0.1606u(k) \tag{5.26}$$

For simplicity, we let the input be the step signal

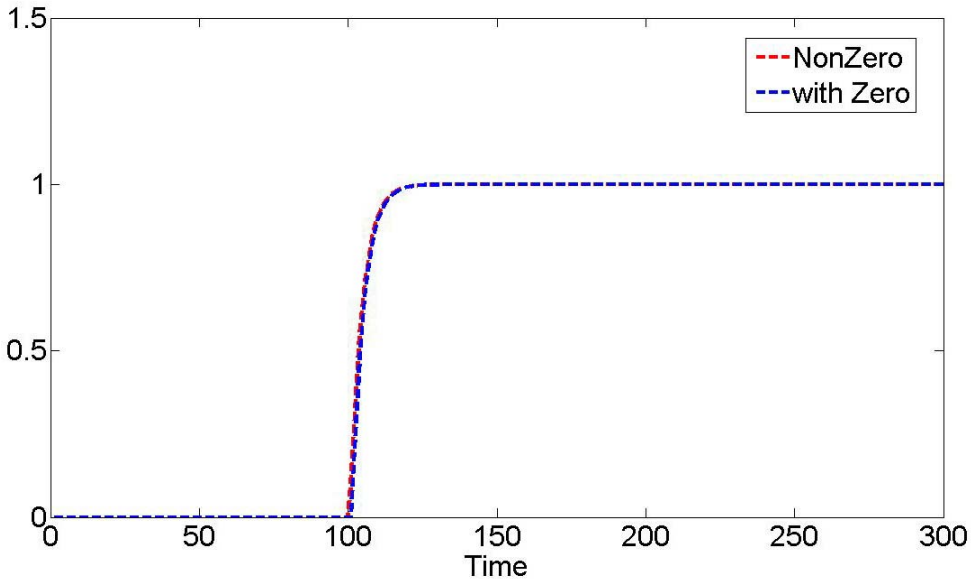


Figure 5.2 Simulation Comparison for step input

To further investigate the difference, we take a PRTS (Pseudo-random-ternary-series) as input signal.

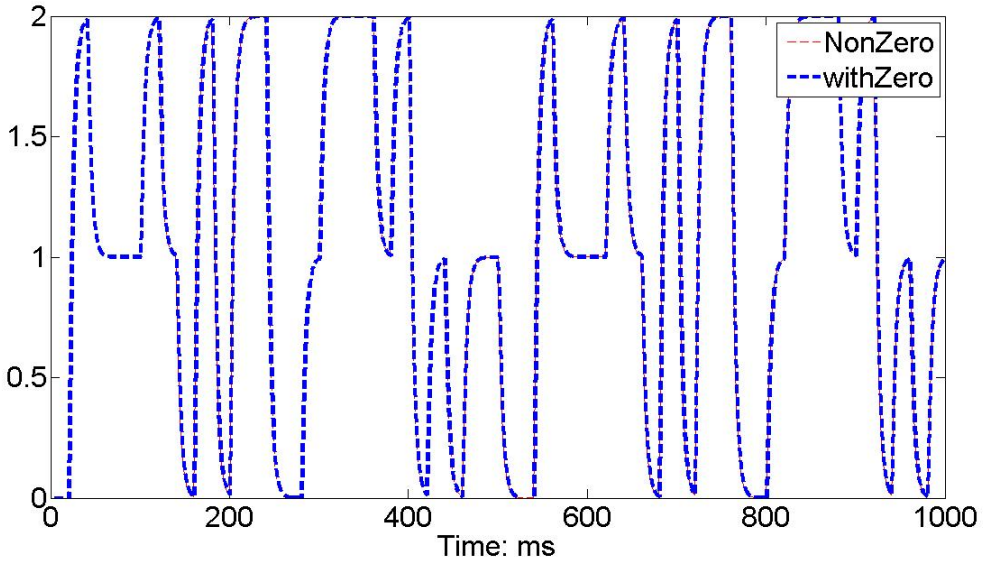


Figure 5.3 Simulation Comparison for PRTS Input

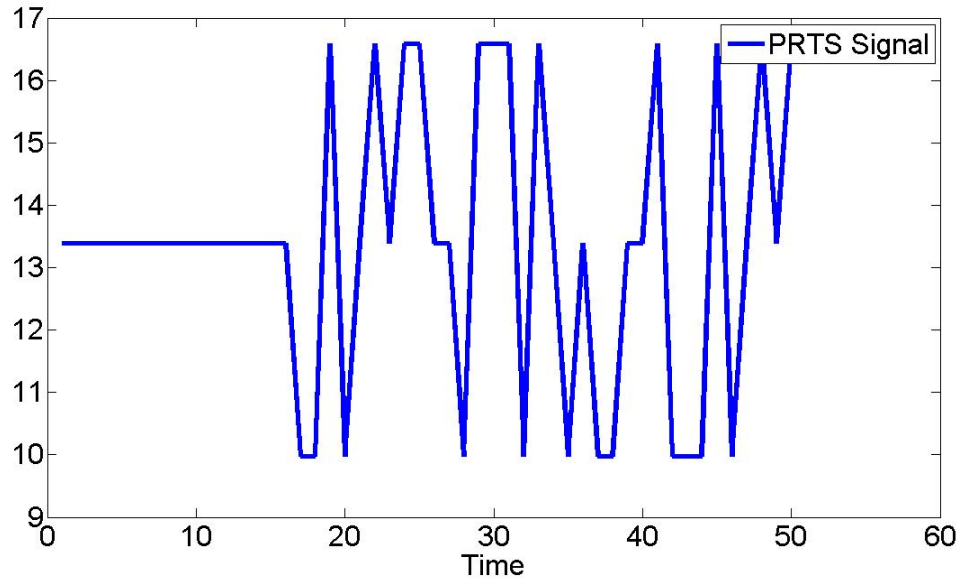


Figure 5.4 PRTS Input

5.4 Summary

We simply run the simulations to compare the responses of the two models under step signal and PRTS signals. It is easy to see that the responses of the two models are strictly identical. This basically shows two possibilities for later Hammerstein identification:

1) Under the conditions, especially of specific test input signals; we can approximate the linear dynamics with non zeros. In other words, we can assume the denominator as a constant b
2) The dynamic characteristics can be obtained even if without the knowledge of all the coefficients of the denominator $B\left(q^{-1}\right)=\sum_{i=0}^{m} b_i q^{-i} = b_0 + b_1 q^{-1} + \cdots + b_m q^{-m}$. This has been shown in the above simulation, although this case is only under the assumption that the input is step based signals. We believe this condition is best for our identification of laser welding processes. However, we also believe that if necessary, the zeros can be added into the linear dynamics and identified following our method with much less known parameters in the system.

Before we go into the detail about our proposed Hammerstein identification, let us take a look what happens if we make another assumption

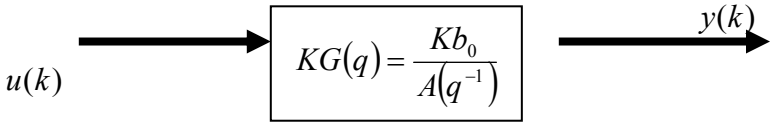


Figure 5.5 Hammerstein Structure Restructure

If we further replace the nonlinearity in the Hammerstein structure by a scaling gain factor $K = f(u)$, then the system can be equivalently considered as a specific linear system, represented by:

$$y(t) = K \frac{b}{A\left(q^{-1}\right)} u(t) \tag{5.27}$$

Or

$$y(k) = \sum_{i=1}^{n} a_i y(k-i) + b_0 f(u) \tag{5.28}$$

Accordingly, the identification is to find the nonlinearity function and the parameters of the linear dynamics.

To see how the algorithm could work, we can rewrite the model as:

$$y(k) = \phi^T \theta$$

$$\phi^T = \begin{bmatrix} -y(k-1) & \cdots & -y(k-n) & 1 \end{bmatrix} \tag{5.29}$$

$$\theta^T = \begin{bmatrix} a_1 & \cdots & a_n & Kb_0 \end{bmatrix} \tag{5.30}$$

Accordingly, the identification can be accomplished by estimating the parameters with methods such as Least Square, and so on

CHAPTER 6

VISION BASED SENSOR: EXPERIMENTAL SET UP

In our experiment, a CCD digital camera as shown in Figure 6.1 equipped with a pulsed lighting laser is installed in the system to capture the images of weld pool (as shown in the following figure). Image processing is implemented with Matlab image processing toolbox [94] and LabView IMAQ Vision builder [95].

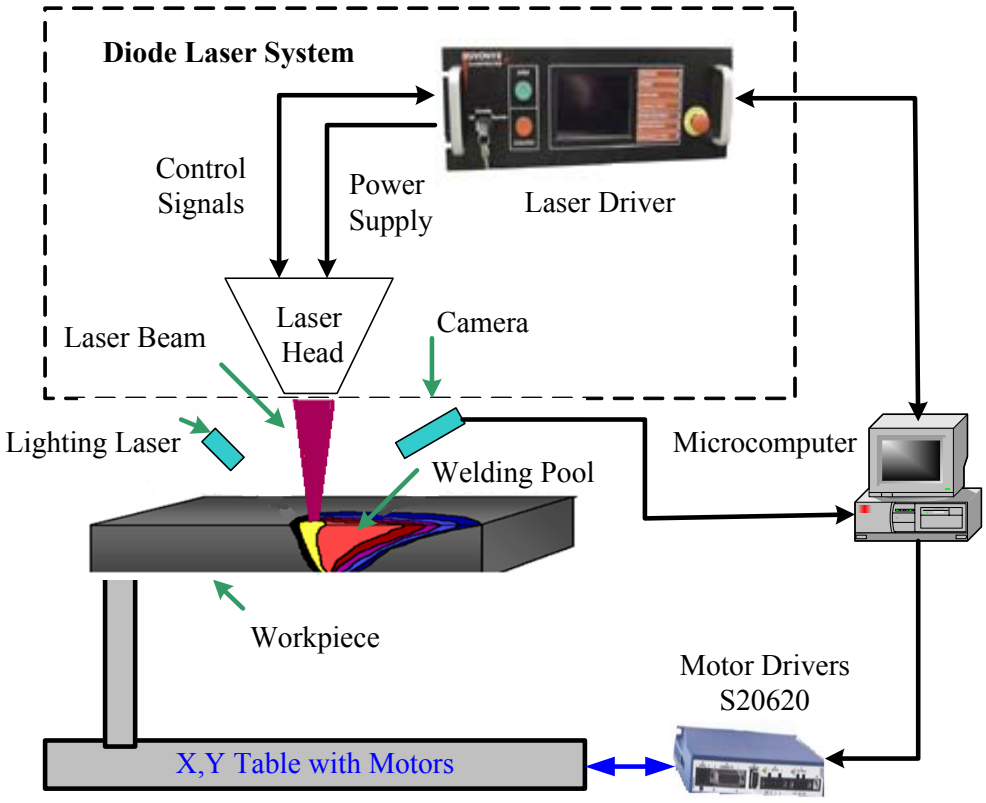


Figure 6.1 System set up

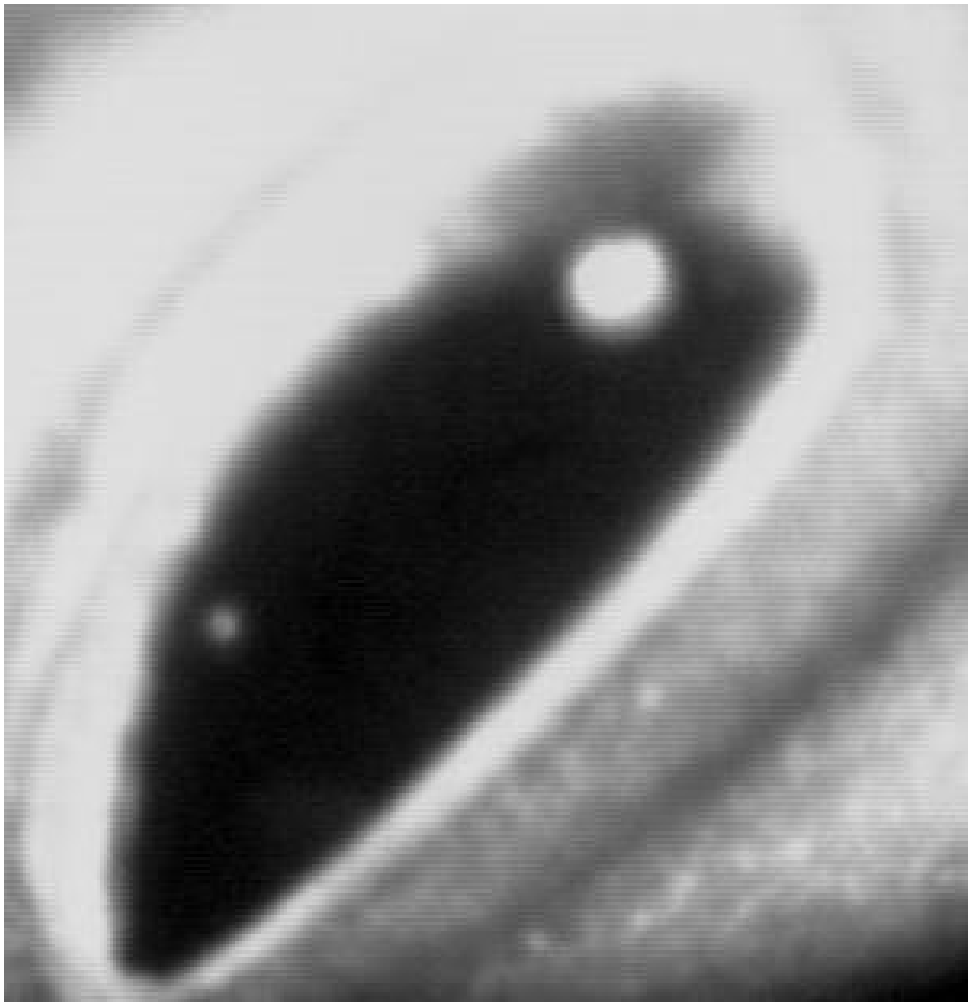


Figure 6.2.a Raw Images after ROI cropped

Figure 6.2.a is the raw image acquired from the vision sensor. Apparently, it needs pre-processing before further feature extraction. As introduced in Chapter 5, there are different methods. To get better results, we try different methods and compare the performance based on system running time and processed results.

6.1 Results from Deblur

Results of debluring using above methods are shown in Figure 6.2. The leftmost is the raw images. Among them the Maximum likelihood performs the worst, which basically means the Poisson distribution assumption might be controversial for our application. Other two are both applicable to our system.

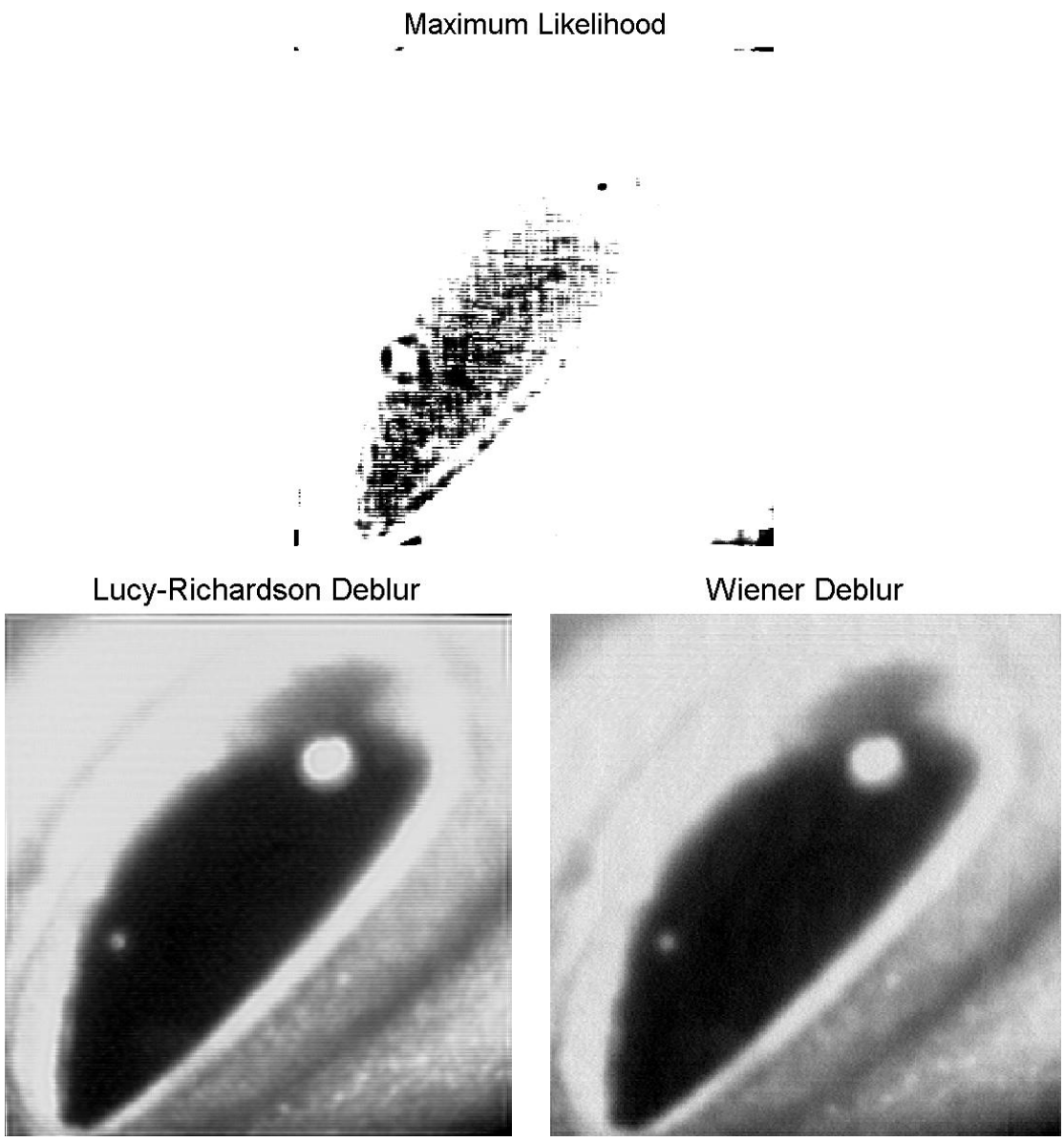


Figure 6.2.b Results of Deblur

6.2 Results from Denoise

Results of filtering are shown as follows. Among them, it seems Average filter performs worse and the result is a little dark. Others work better.

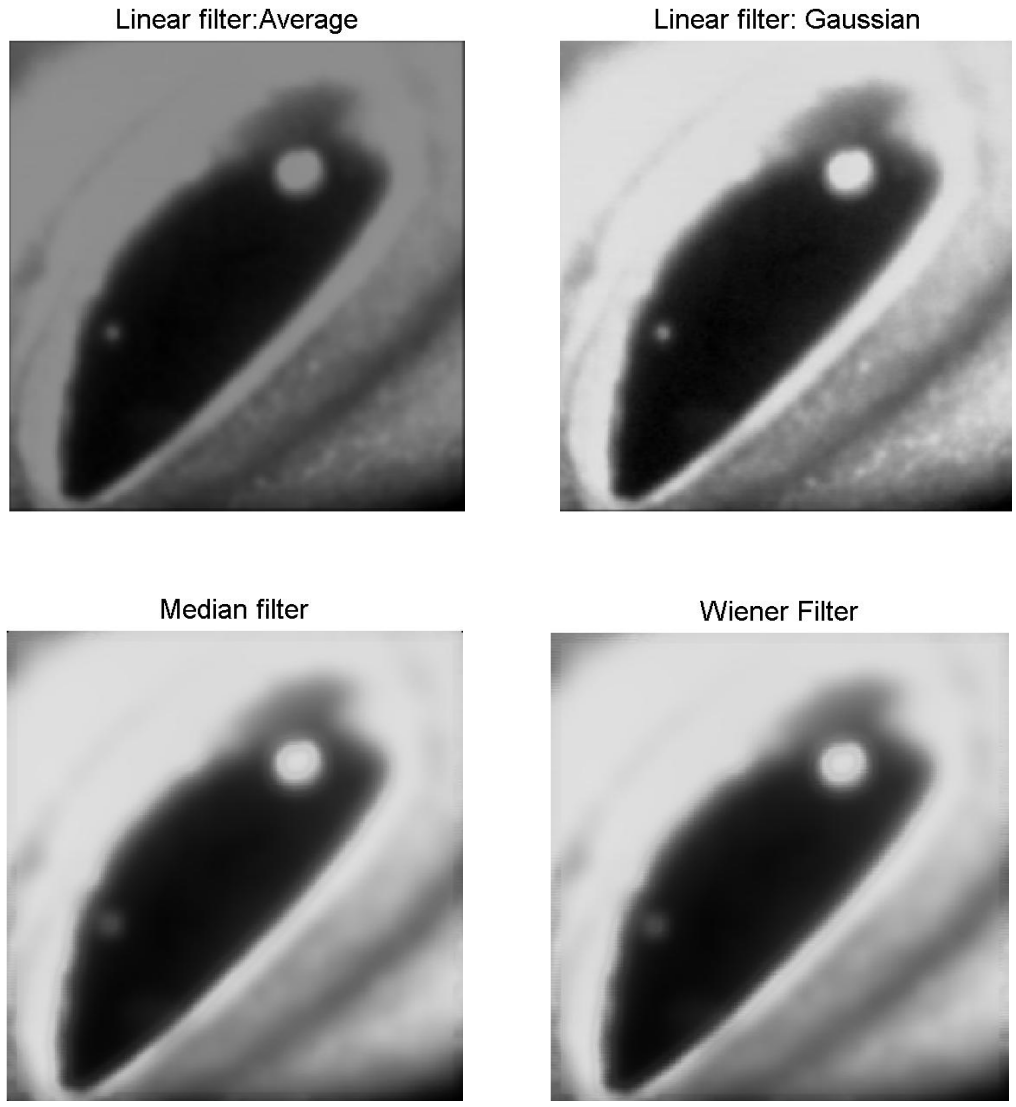


Figure 6.3 Results of Denoise

Table 6.1 Pre-processing Performance Comparison

Method	Running time (ms)	Results (pixels%)
Maximum Likelihood	78	55
Lucy-Richardson	82	95
Wiener Deblur	18	94
Linear Average denoise	18	93
Linear Gaussian	22	92
Median	18	90
Wiener	22	90

6.3 Deblur and Filter Together

To better show the performance of image deblurring and noise removal, we run filter and deblur together. Results are as follows. Without loss of generality, we choose median filter for noise removal and winer filter for deblur. For our application, image deblur actually makes the image brighter and clearer. Filter sharpens the regions in the images, through which edge detection is easier to implement.

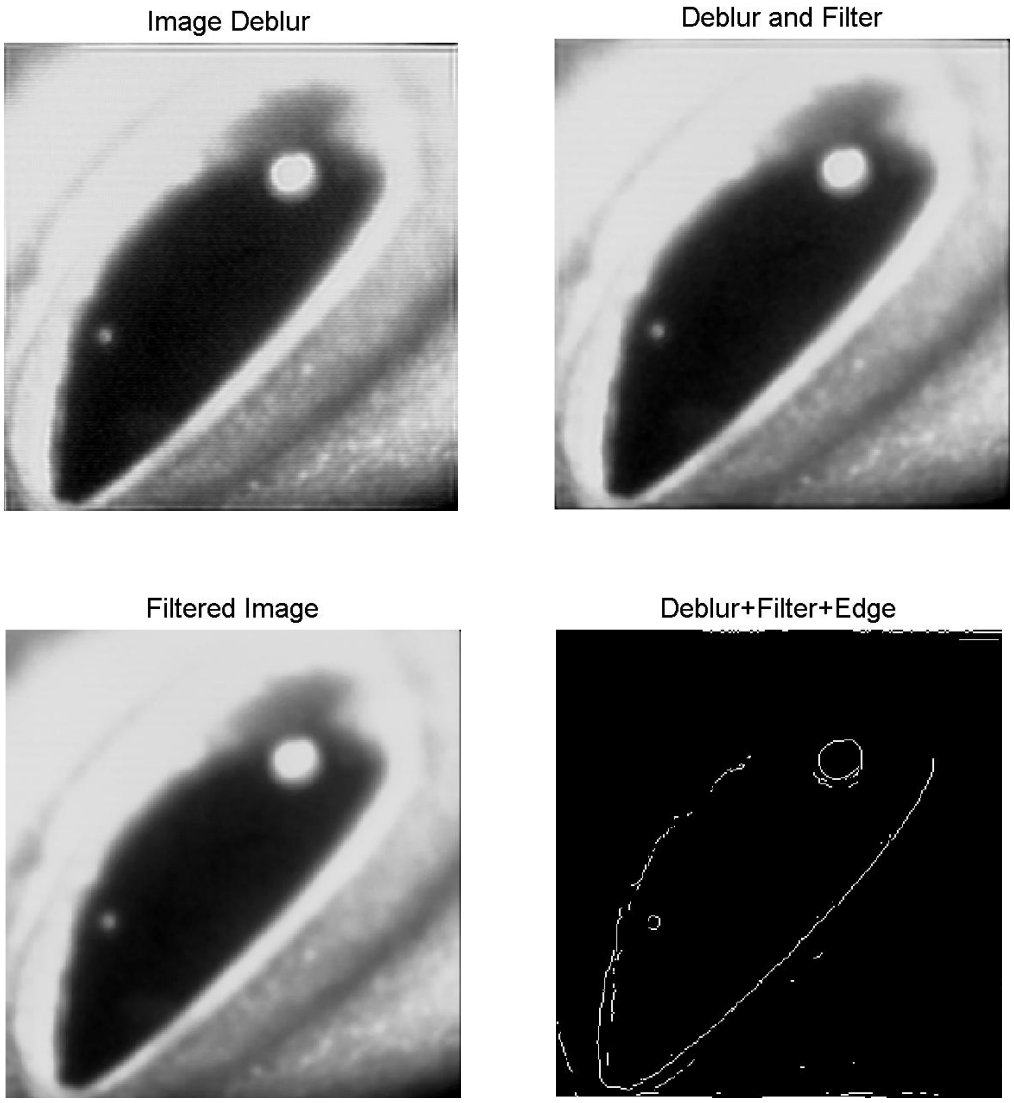


Figure 6.4 Results of Deblur and Denoise

6.4 Results of Thresholding

The results of thresholding with the threshold selection methods introduced above are shown in the following for comparison. In the raw image, the black section in the middle is the molten weld pool whose feature is what we are interested in.

Among them, Global thresolding based on histogram performs worse and loses too much information because of the threshold level selected. Adaptive threshold seems to work better. But besides the weld pool, it gives too much extra information which might be complicating later processing. The one with the statistical mean, Center of Mass, and Least mean error performs relatively comparable to each other. That is because

theoretically these three share common grounds. FCM also performs better except it is a little bit complex to realize and not suitable for real time image processing. OTSU and iterative also generates better results and easily to realize in the system.

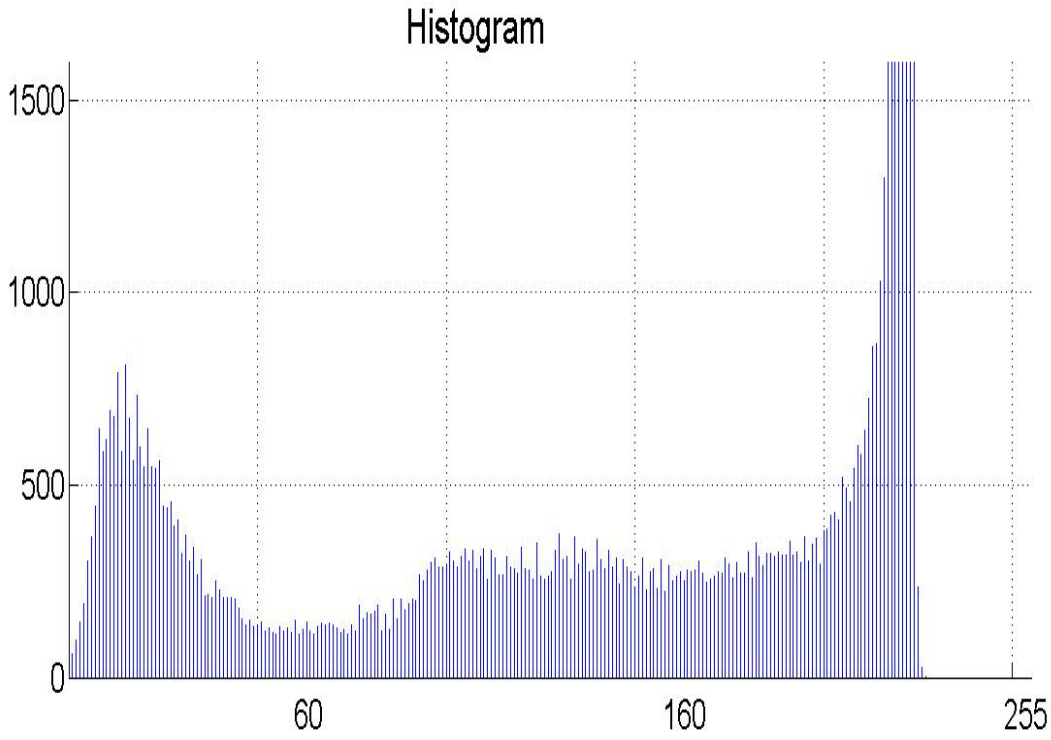
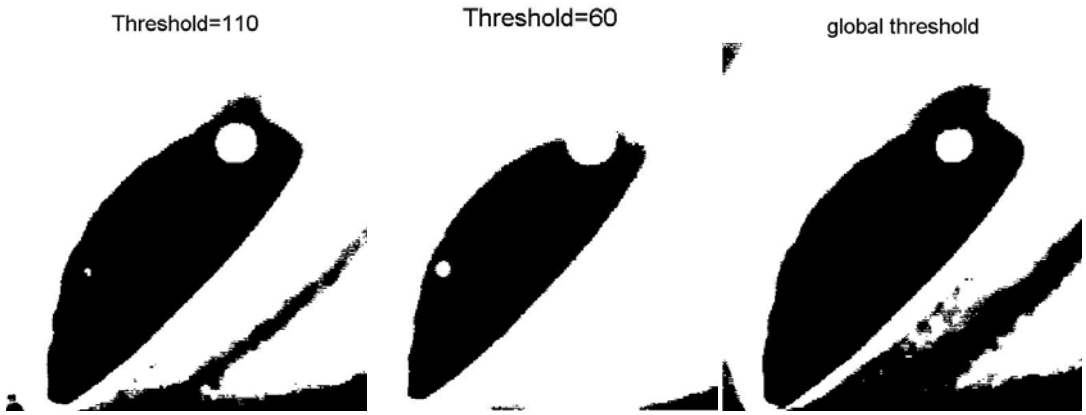


Figure 6.5 Histogram of Original Image



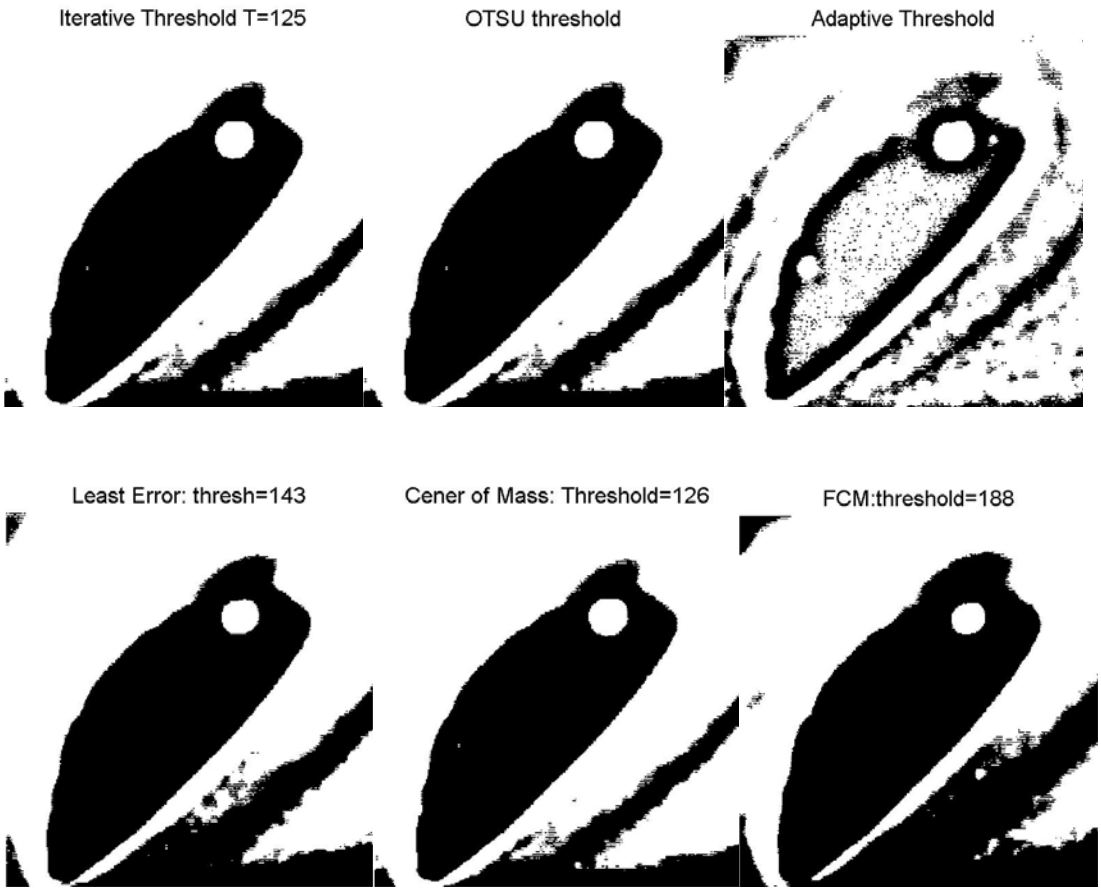


Figure 6.6 Results of Thresholding

Table 6.2 Thresholding Performance Comparison

Threshold Method	Running time (ms)	Results (pixels%)
Global T=110	22ms	85
Global T=60	22	75
Global Average	21	90
Iterative T=125	78	92
OTSU	24	93
Adaptive	107	82
Least Error	184	90
Center of Mass	56	89
FCM	210	86

6.5 Results of Edge Detection

Edge detection results are shown in the following with methods introduced above. Apparently, Canny gives deeper results, which also has too many edge points for later process, same as Laplacian operator. Prewitt and Robert perform a little bit better, but it seems losing more edge information interested. Wavelet demonstrate the best and the edge interested is very sharp and distinct. Sobel also performs reasonably better, except it has a gap on the top corner which might need edge growing or linking.

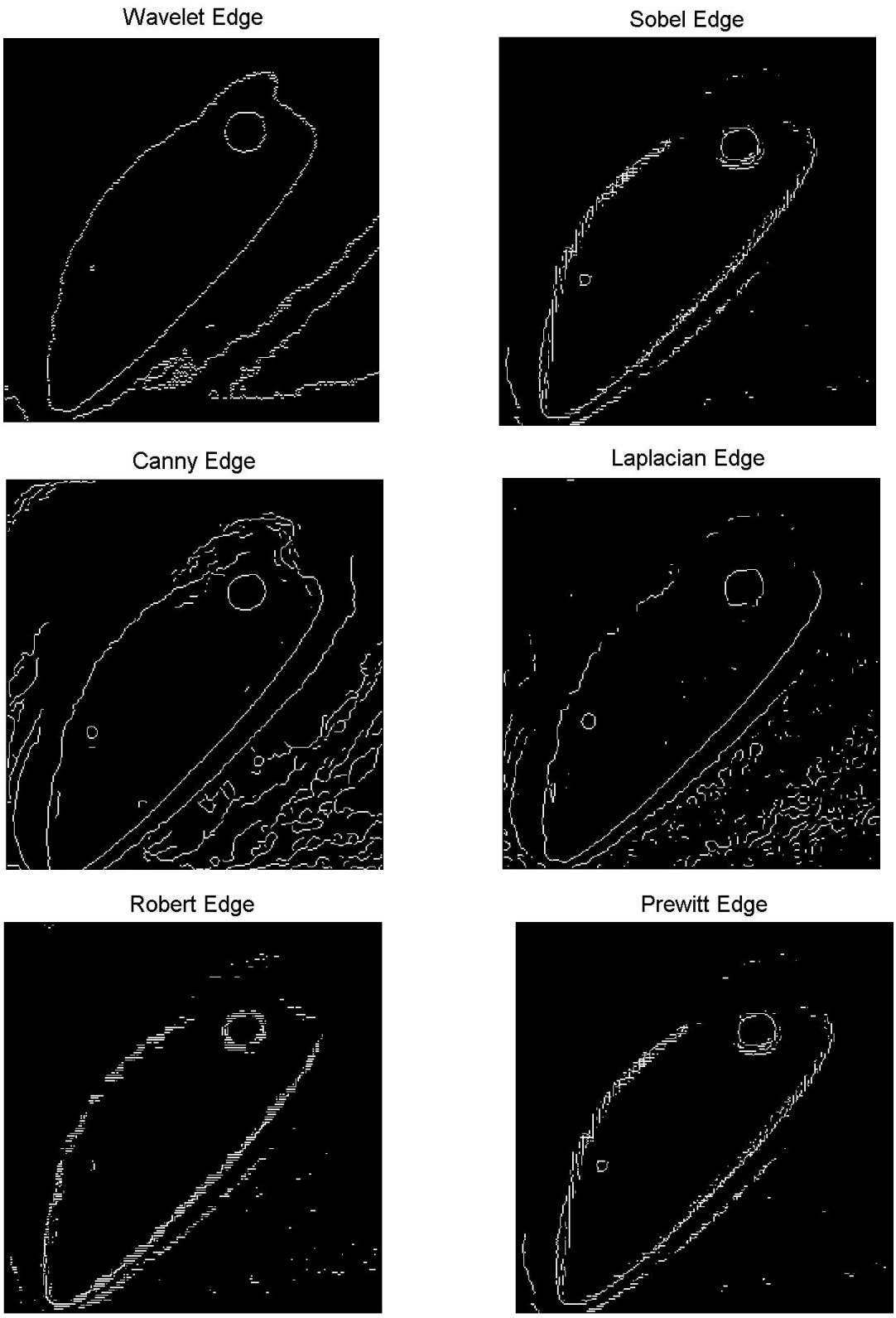


Figure 6.7 Results of Edge Detection

Note: system is running Window XP Profession, 1.6Hz Intel Pentium CPU, and 512 RAM

Table 6.3 Image Processing Method Performance Comparison

Method	Running time (ms)	Results (pixels%)
Wavelet	111ms	95
Sobel	33	88
Canny	45	89
Laplacian	35	74
Robert	36	88
Prewitt	32	89

6.6 Calibrated Width: Maximum Edge

The calibration is to seek a map function between the real world and the digital image world as shown in Figure. 6.8. In the experiments, we rotate the image 30 degree counterwise to offset the angle of the laser line and use the calibration settings:

$$dx = 10mm \Rightarrow 150\,pixels$$
$$dy = 10mm \Rightarrow 124\,pixels$$
$$\theta = 60^{\circ}$$

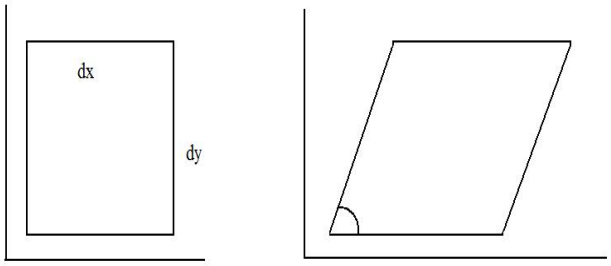


Figure 6.8 Calibration

6.7 Architecture of Vision Sensor

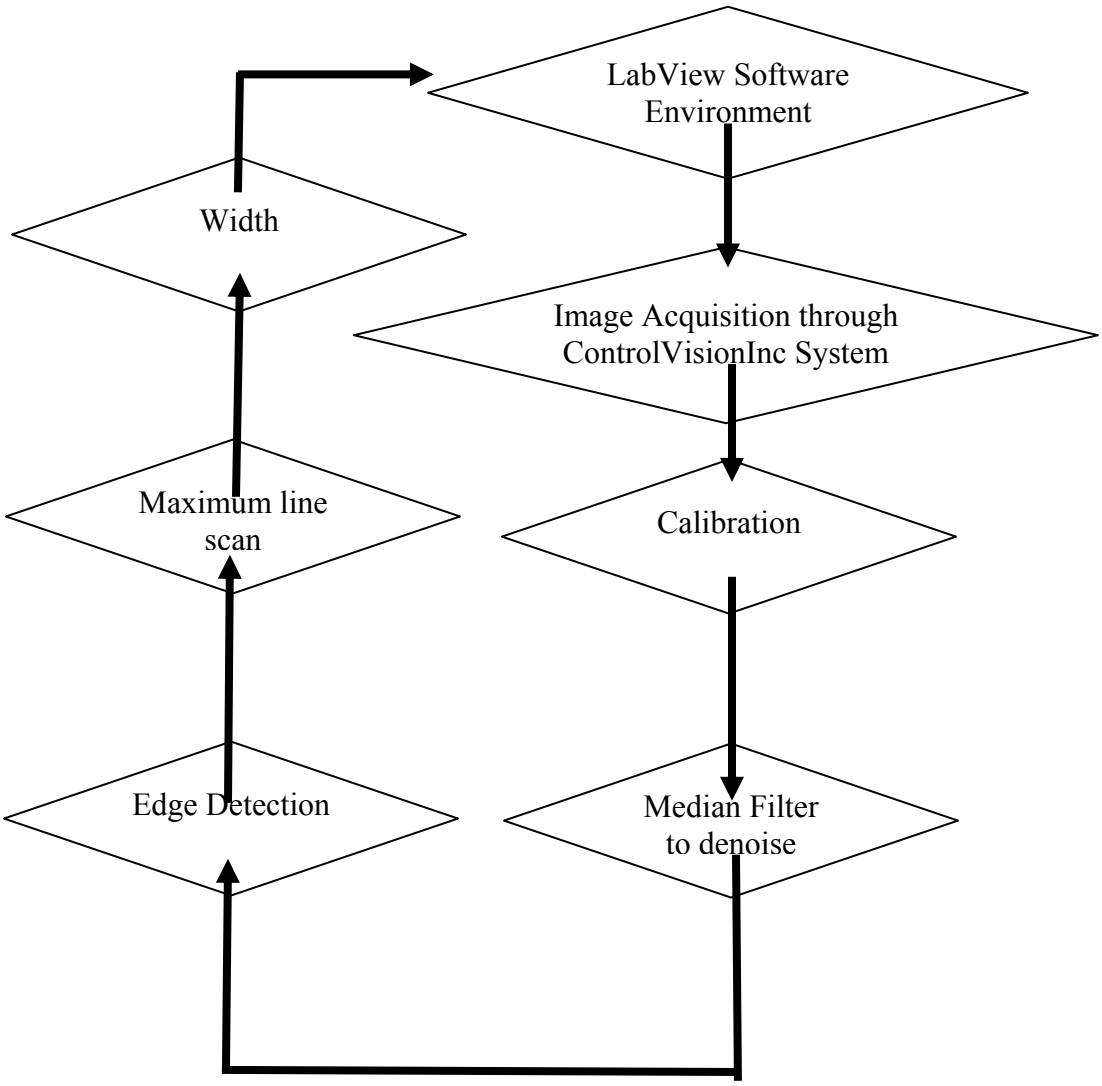


Figure 6.9.a Feature extraction process

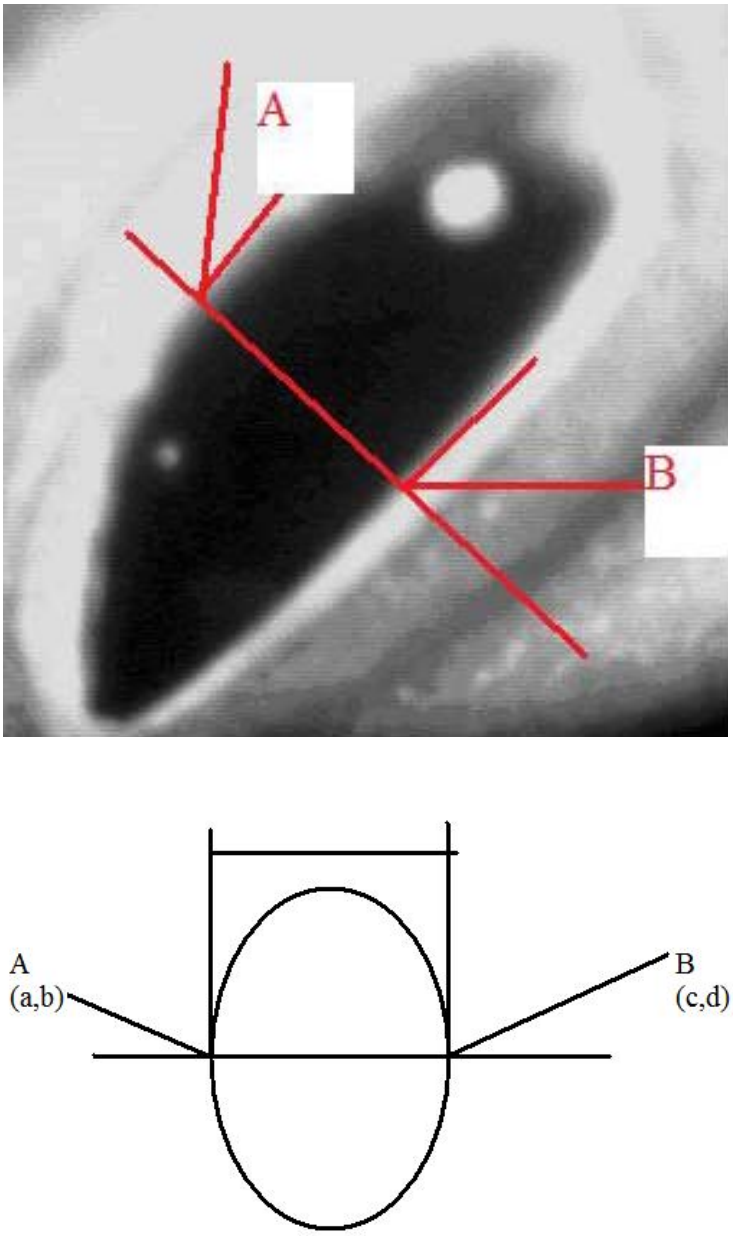


Figure 6. 9.b Weld pool width diagram

In the experiments, because of the unseen noise, we apply median filter at first. The mask template is:

$$G(x,y)=\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \tag{6.1}$$

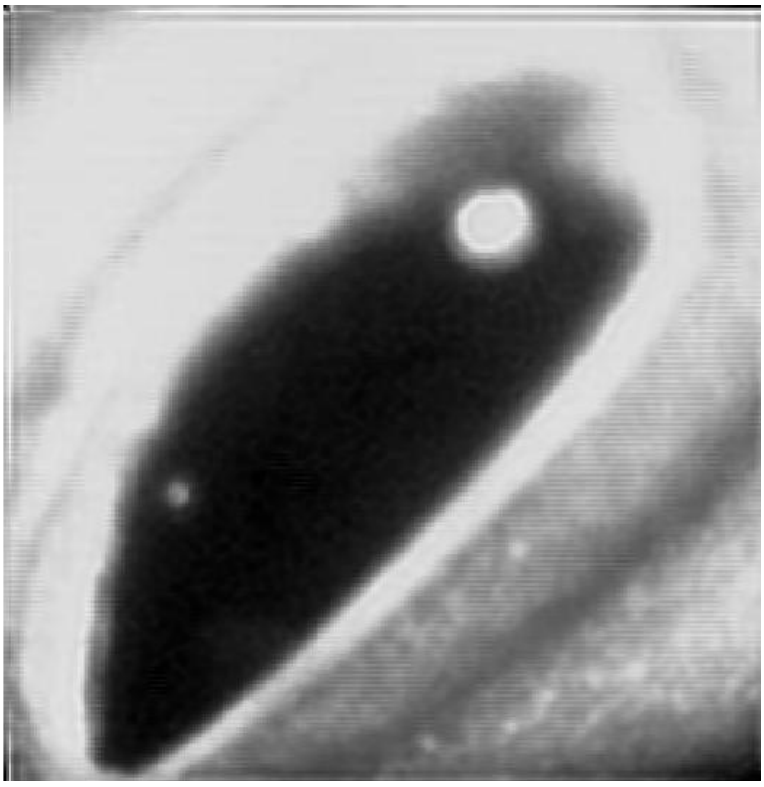


Figure 6. 10.a Processed with Median Filter

To further enhance the image, Laplace operator is applied to the image:

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{6.2}$$

Or,

$$\frac{\nabla^2 f(x,y)}{\partial x^2} = f(i+1,j) + f(i-1,j) - 2f(i,j) \tag{6.3}$$

$$\frac{\nabla^2 f(x,y)}{\partial y^2} = f(i,j+1) + f(i,j-1) - 2f(i,j) \tag{6.4}$$

With substitution, the operator will be:

$$\nabla^2 f = f(i-1,j) + f(i+1,j) + f(i,j+1) + f(i,j-1) - 4f(i,j) \tag{6.5}$$

Also, depending on the need, sometimes, we multiply the image itself for enhancement:

$$G(x,y)=\frac{f(x,y)f(x,y)}{255} \tag{6.6}$$

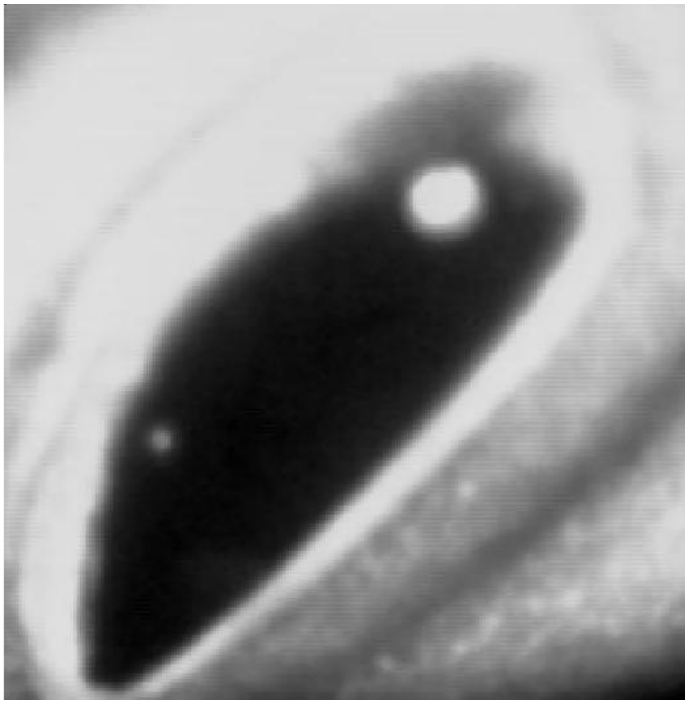


Figure 6. 10.b Processed with Enhancement

To get better thresholding, we set up two ways. One is the range of the threshold level, i.e. [110 135]. The other way is similar to adaptive thresholding. The procedure is:
At first, calculate the probability and the expectation of pixels:

$$G(x,y)=\frac{m}{N} \tag{6.7}$$

$$E(x,y)=\sum_{k=0}^{255} kG(x,y) \tag{6.8}$$

Then set the threshold

$$T=\gamma E(x,y) \tag{6.9}$$

Or

$$T = \gamma \sum_{k=0}^{255} kG(x, y) \tag{6.10}$$

$$\gamma \geq 1 \tag{6.11}$$

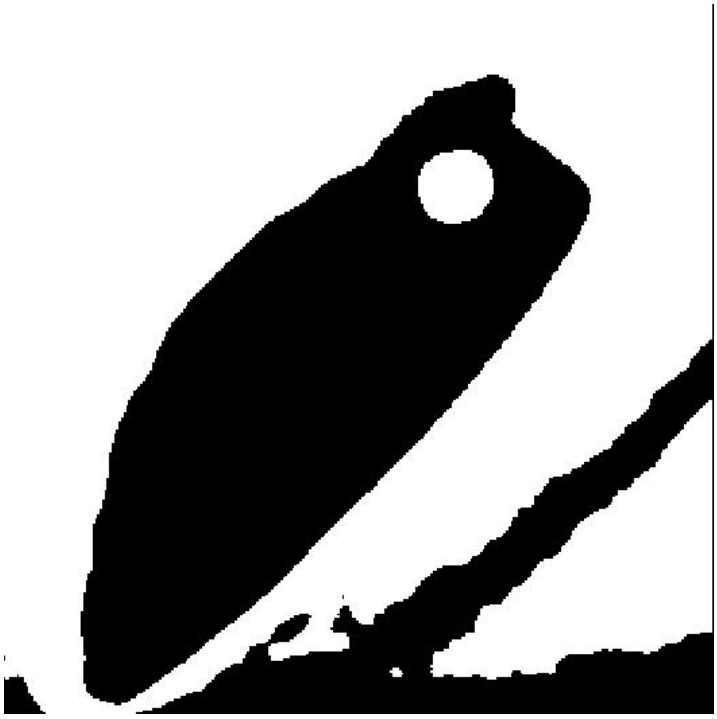


Figure 6. 10.c Processed with adaptive thresholding

For edge detection, as introduced above, we simply choose Sobel operator:

$$S_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{6.12}$$

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{6.13}$$

The detected and highlighted edge is shown as following:

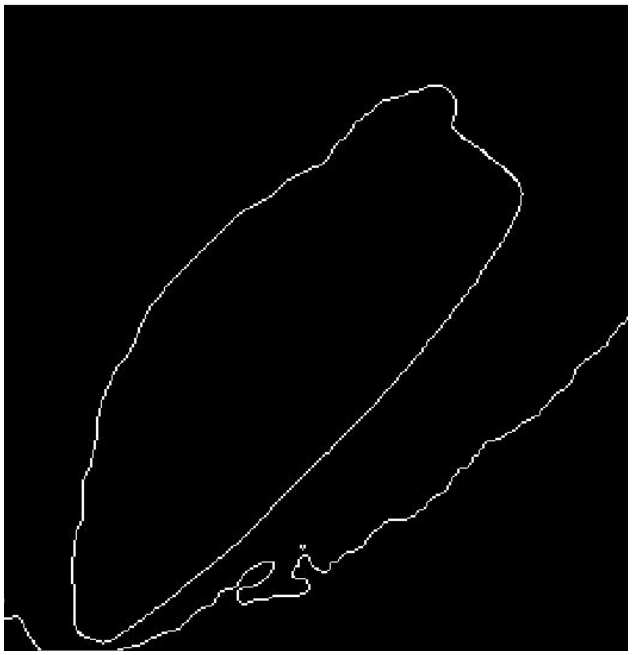


Figure 6. 10.d Edge Detection with Sobel Operator

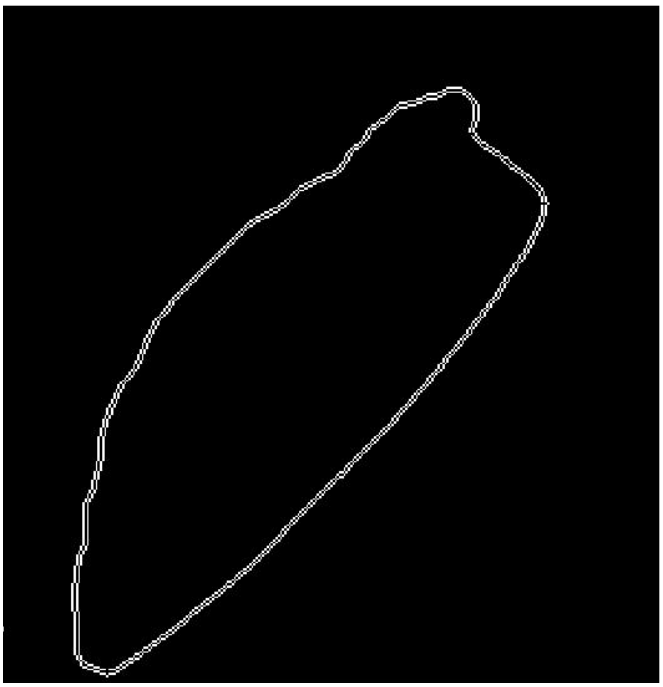


Figure 6.10.e Highlighted Edge

CHAPTER 7

NONLINEAR HAMMERSTEIN IDENTIFICATION: DETAIL

7.1 Continuous Identification

As introduced previously, system identification can be classified into two categories, i.e., continuous and discrete [96]-[104]. The advantage of discrete identification is its easiness [104]-[105]. The disadvantage is that its performance may be compromised by the signal frequency and data completeness. Because dynamic systems are naturally continuous, building a continuous model is relatively beneficial for control purposes.

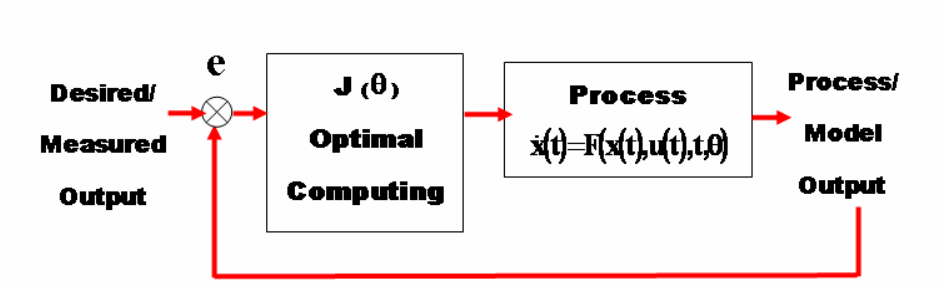


Figure 7.1 Optimal System Identification

Consider a standard state-space representation system:

$$\dot{x} = F(x(t), u(t), t, \theta) \tag{7.1.1}$$

, where $\{x,\ u,\ \theta\}$ are the vectors of state, input or control, and associated parameters, $\dot{x}$ is the first-order derivative of the state vector and F is the system function. Then the identification shown in Figure 7.1 optimally computes the unknown parameters with minimizing the cost function:

$$J(\theta) = \int_0^t g(e) d\tau \tag{7.1.2}$$

, where e is the predictive error between the estimation and the measurement. g is the objective function which is, for simplicity, represented by[54] [106]:

ISE, integral of square error, $J(\theta)=\int_0^t e^2 d\tau$

IAE, integral of absolute error $J(\theta)=\int_0^t |e| d\tau$ (7.1.3)

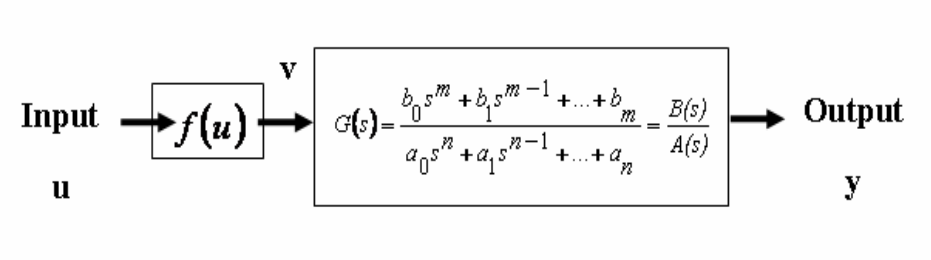


Figure 7.2 Hammerstein Structure

Figure 7.2 is the traditional Hammerstein structure in which the nonlinearity directly feeds into the linear dynamics, where the linear transfer function $G(s)$ is generally represented by:

$$G(s)=\frac{b_0 s^m+b_1 s^{m-1}+\cdots+b_m}{a_0 s^n+a_1 s^{n-1}+\cdots+a_n}=\frac{B(s)}{A(s)} \quad (7.1.4);$$

$f(u(t))$ is the nonlinear function of the system input and the input of the linear dynamics; $\{a_i\}, i=0,1,\cdots n$ and $\{b_i\}, i=0,1,\cdots m$ are parameters of the system model; u and y are the input and the output.

Denote

$$v(t)=f(u(t)) \quad (7.1.5)$$

Then in the frequency domain:

$$y(s)=G(s)L(f(u(t))) \quad (7.1.6)$$

, where L is the operator of Laplace transform.

Without loss of generality, denote:

$$a_n=1 \quad (7.1.7)$$

$$K(u(t)) = L^{-1}(B(s)L(f(u(t)))) \tag{7.1.8}$$

, where L^{-1} is the operator of inverse Laplace transform.

Then with substitution, equation 6.2.6 can be rewritten as:

$$a_0 y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots y(t) = K(u(t)) \tag{7.1.9}$$

Rewrite it into matrix form:

$$y(t) = \phi^T(t)\theta \tag{7.1.10}$$

$$\phi^T(t) = \begin{bmatrix} -y^{(n)}(t) & \cdots & -y^{(1)}(t) & 1 \end{bmatrix} \tag{7.1.11}$$

$$\theta^T = \begin{bmatrix} a_0 & \cdots & a_{n-1} & K(u(t)) \end{bmatrix} \tag{7.1.12}$$

Accordingly this Nonlinear Hammerstein identification becomes to estimate unknown parameters in equation 7.1.10-12. Among these parameters only $K(u(t))$ varies with inputs.

Suppose $f(u(t))$ is a polynomial:

$$f(u(t)) = C_0 + C_1 u(t) + \cdots + C_p u^p(t) \tag{7.1.13}$$

Now consider step input $u(t) = U$. In this case, $f(u(u))$ is also a step signal and its Laplace transform is $L(f(u(t))) = f(U)/s$; Further if we suppose equation 7.1.4 takes the following form:

$$G(s) = \frac{b_m}{a_0 s^n + a_1 s^{n-1} + \cdots + a_n} \tag{7.1.14}$$

Denote

$$K_1(u) = L^{-1}(b_m f(U)/s) = b_m f(U) \tag{7.1.15}$$

Then with similar substitution, equation 7.1.9 can be written as:

$$a_0 y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots y(t) = K_1(u(t)) \tag{7.1.16}$$

, where $K_1(u(t)) = K(u(t))$ and $B(s) = b_m$ is a constant.

As a result, the proposed identification for a specific Hammerstein structure with equation 7.1.14 as the linear dynamics and equation 7.1.13 as the nonlinear static function in general can be achieved in two steps:

Step 1: Identify A(s) under different step input $\{U_j\}, j = 1, \cdots, M$ and because of the use of a step input U_j,

$$K_1(u) = K_1(U_j) = b_m f(U_j) \tag{7.1.17}$$

is now a constant and can be treated as a parameter to be identified such that equation 7.1.12 can be written as

$$\theta^T(U_j) = \begin{bmatrix} a_0 & \cdots & a_{n-1} & K(U_j) \end{bmatrix} \tag{7.1.18}$$

Hence, A(s) and the value of $b_m f(U_j)$ can be identified.

Step 2: Determine the nonlinear function $f(u)$ and b_m from the steady-state responses $Y(U_j)$ under step input $\{U_j\}, j = 1, \cdots, M$.

In this case:

$$K(U_j) = \phi^T(U_j)\theta \tag{7.1.19}$$

With

$$\phi^T(U_j) = \begin{bmatrix} 1 & U_j & U_j^2 & \cdots & U_j^p \end{bmatrix} \tag{7.1.20}$$

And

$$\theta^T = \begin{bmatrix} b_m C_0 & b_m C_1 & b_m C_2 & \cdots & b_m C_p \end{bmatrix} \tag{7.1.21}$$

Then the parameters in $b_m f(U_j)$ can be determined using a linear least square algorithm:

$$\theta = \left(\Phi^T \Phi\right)^{-1} \Phi^T \left[Y(U_1) \quad \cdots \quad Y(U_M) \right]^T \tag{7.1.22}$$

$$\Phi^T = \left[\phi^T(U_1) \quad \cdots \quad \phi^T(U_M) \right]^T \tag{7.12.23}$$

7.2 Continuous Identification: Error based

Above the authors propose and apply a Hammerstein based identification on the diode laser welding process. This method is fundamentally applicable to the nonlinear processes whose dynamics can be approximated by $A(s)$ only. However, to ensure

accurate results, the method requires the dynamic step response experiments to be conducted at as more different inputs signals as possible. Sometimes it is very time consuming. For this reason, the authors develop an error based nonlinear identification to reduce the need for the number of step responses experiments. The method consists of two steps and step 1 is the same as that in the Hammerstein identification.

Now assume $u(t)=U$ is the step signal applied to generate the only dynamic step response experiment and

$$a_0 y^{(n)}(t)+a_1 y^{(n-1)}(t)+\ldots+y(t)=K(U) \tag{7.2.1}$$

is identified by $\theta^T=\begin{bmatrix}a_0 & a_1 & a_{n-1} & \cdots & K(U)\end{bmatrix}$ (7.2.2)

In the identification of the laser system, take one of the estimated models during a specific input signal as the welding speed and obtain:

$$a_0 y^{(n)}(t)+a_1 y^{(n-1)}(t)+\ldots+y(t)=K(U_i) \tag{7.2.3}$$

The linear dynamic parameters of equation 7.2.3 above are now used as the estimates of those in the system despite variations in the inputs.

Now assume a number of steady-state responses $\{y_j(\infty)\}$ have been obtained under $\{U_j\}$ such that $\{y_j(\infty) \quad U_j\}, j=1,2,\cdots M$ are available.

The objective is to identify the structure and the parameters for the nonlinear function $K(U)$ through $\{K(U_j) \quad U_j\}$.

To this end,

$a_0 y^{(n)}(t)+a_1 y^{(n-1)}(t)+\ldots+y(t)=1$ is simulated. and the simulated steady-state responses $\hat{y}_j(\infty)$ is used to compute

$$K(U_j)=y_j(\infty)/\hat{y}_j(\infty)\big|_{K(U)=1} \tag{7.2.4}$$

Theoretically $\hat{y}_j(\infty)=1$ according to finalization thermo of Laplace transform:

$$y(s)|_{s=0}=\frac{1}{a_0 s^n+a_1 s^{n-1}+\cdots+1}|_{s=0}=1$$

Then easy to see that

$$K(U_j)=y_j(\infty)/\hat{y}_j(\infty)\big|_{K(U)=1}=y_j(\infty)$$

As a result, a set of $\{K(U_j),\ U_j\}$ will be easily obtained through a single model of linear system using

$$a_0 y^{(n)}(t) + a_1 y^{(n-1)}(t) + \ldots + y(t) = 1 \qquad (7.2.5)$$

and the static response experimental results as given above.

Then for any proposed structure of $K(U)$, its parameters can be identified using as linear least square algorithm.

7.3 Discrete Identification: Another Perspective

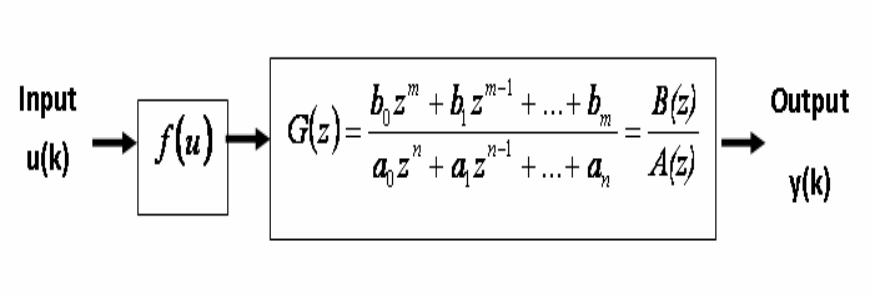


Figure 7.3 Discrete Hammerstein Structure

Figure 7.3 is the discrete Hammerstein structure in which the nonlinearity directly feeds into the linear dynamics, where the linear transfer function $G(z)$ is generally represented by:

$$G(z) = \frac{b_0 z^m + b_1 z^{m-1} + \cdots + b_m}{a_0 z^n + a_1 z^{-1} + \cdots + a_n} = \frac{B(z)}{A(z)} \qquad (7.3.1)$$

$f(u(k))$ is the nonlinear function of the system input and the input of the linear dynamics; $\{a_i\}, i = 0,1,\cdots n$ and $\{b_i\}, i = 0,1,\cdots m$ are parameters of the system model; $u(k)$and $y(k)$are the input and the output.

Denote $v(k) = f(u(k))$ (7.3.2)

Then in the Z domain: $y(z) = G(z)Z(f(u(k)))$ (7.3.3)

, where Z is the operator of Laplace transform.

Without loss of generality, denote $a_0 = 1$ (7.3.4)

$$S(u(k)) = Z^{-1}(B(z)Z(f(u(k)))) \qquad (7.3.5)$$

, where Z^{-1} is the operator of inverse Laplace transform.

Then with substitution, 7.3.1 can be rewritten as:

$$a_0 y(k) + a_1 y(k-1) + \cdots y(k-n) = S(u(k)) \tag{7.3.6}$$

Rewrite 6.3.6 into matrix form:

$$\begin{aligned} y(k) &= \phi^T(k)\theta \\ \phi^T(k) &= \begin{bmatrix} -y(k-1) & \cdots & -y(k-n) & 1 \end{bmatrix} \\ \theta^T &= \begin{bmatrix} a_1 & \cdots & a_n & S(u(k)) \end{bmatrix} \end{aligned} \tag{7.3.7}$$

Accordingly this Nonlinear Hammerstein identification becomes to estimate unknown parameters in equation 7.3.7 and among these parameters only $S(u(k))$ will vary with the inputs.

Suppose $f(u(k))$ is a polynomial:

$$f(u(k)) = C_0 + C_1 u(t) + \cdots + C_p u^p(t) \tag{7.3.8}$$

Now consider step input $u(k) = U$ in Figure.7.3. In this case, $f(u(k))$ is also a step signal and its Z transform is $Z(f(u(t))) = f(U)z/(z-1)$; Further if we suppose 6.3.1 takes the following form:

$$G(z) = \frac{b_m}{a_0 z^n + a_1 z^{n-1} + \cdots + a_n} \tag{7.3.9}$$

Denote:

$$K_1(u) = L^{-1}(b_m f(U) z/(z-1)) = b_m f(U) \tag{7.3.10}$$

Thenthen with similar substitution, equation 7.3.1 can be written as:

$$a_0 y(k) + a_1 y(k-1) + \cdots y(k-n) = S_1(u(k)) \tag{7.3. 11}$$

, where $S_1(u(t)) = S(u(t))$ and $B(s) = b_m$ is a constant.

As a result, the proposed identification for a specific Hammerstein structure with equation 7.3.9 as the linear dynamics and equation 7.3.10 as the nonlinear static function in general can be achieved in two steps:

Step 1: Identify A(s) under different step input $\{U_j\}, j = 1, \cdots, M$

Because of the use of a step input U_j $S(u) = S(U_j) = b_m f(U_j)$ is now a constant and can be treated as a parameter to be identified such that equation 7.3.7 can be written as

$$\theta^T(U_j) = \begin{bmatrix} a_0 & \cdots & a_{n-1} & S(U_j) \end{bmatrix}$$

Accordingly, the model structure $A(z)$ and the value of $b_m f(U_j)$ can be identified.

Step 2: Determine the nonlinear function $f(u)$ and b_m from the steady-state responses $Y(U_j)$ under step input $\{U_j\}, j = 1, \cdots, M$.

In this case,

$$S(U_j) = \phi^T(U_j)\theta \tag{7.3.12}$$

$$\phi^T(U_j) = \begin{bmatrix} 1 & U_j & U_j^2 & \cdots & U_j^p \end{bmatrix} \tag{7.3.13}$$

$$\theta^T = \begin{bmatrix} b_m C_0 & b_m C_1 & b_m C_2 & \cdots & b_m C^p \end{bmatrix} \tag{7.3.14}$$

Then the parameters in $b_m f(U_j)$ can be determined using a linear least square algorithm as:

$$\theta = \left(\Phi^T \Phi\right)^{-1} \Phi^T \left(a_0 + a_1 + \cdots + a_n\right) \begin{bmatrix} Y(U_1) & \cdots & Y(U_M) \end{bmatrix}^T ,$$

where $\Phi^T = \begin{bmatrix} \phi^T(U_1) & \cdots & \phi^T(U_M) \end{bmatrix}^T$

7.4 Extend to 2ISO model

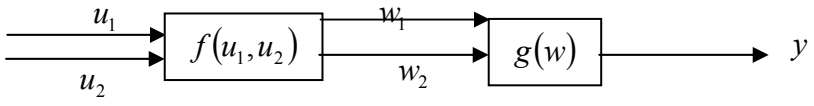


Figure 7.4 Two-input-Single-output Hammerstein System structure

The two steps model identification can expand to 2ISO case, with specific modification. As shown in Figure 7.4, u is the input, y is the output, $g(w)$ is the linear dynamics, w is the intermediate input and $f(u_1, u_2)$ is the nonlinearity. For now, as SISO identification above, we assume the linear system is time-invariant and the inputs u_1, u_2 are independent signals.

The linear dynamic, in Laplace domain can be represented by:

$$G(w)=\frac{B(s)}{A(s)}=\frac{b_0+b_1 s+\cdots+b_m s^m}{a_0+a_1 s+\cdots+a_n s^n} \tag{7.4.1}$$

Denote

$$w(t)=f(u_1(t),u_2(t)) \tag{7.4.2}$$

Then in the frequency domain:

$y(s)=G(s)L(f(u_1(t),u_2(t)))$, where L is the operator of Laplace transform.

Without loss of generality, denote:

$a_0=1$

$K(u_1(t),u_2(t))=L^{-1}(B(s)L(f(u(t))))$, where L^{-1} is the operator of inverse Laplace transform.

Then with substitution, the system structure can be rewritten as:

$$a_0 y^{(n)}(t)+a_1 y^{(n-1)}(t)+\cdots y(t)=K(u_1(t),u_1(t))$$

Rewrite into matrix form:

$$y(t)=\phi^T(t)\theta$$

$$\phi^T(t)=\begin{bmatrix}-y^{(n)}(t) & \cdots & -y^{(1)}(t) & 1\end{bmatrix}$$

$$\theta^T=\begin{bmatrix}a_0 & \cdots & a_{n-1} & K(u_1(t),u_2(t))\end{bmatrix} \tag{7. 4.3}$$

Accordingly this Nonlinear Hammerstein identification becomes to estimate unknown parameters and among these parameters only $K(u_1(t),u_2(t))$ will vary with the inputs.

For simplicity, if we assume the nonlinearity can be represented by a power series, thus

$$f(u_1,u_2)=\sum_{i=0}^{m}\sum_{0}^{n}c_{i,j}u_1^{\,i}u_2^{\,j} \tag{7.4.4}$$

Then the input to the linear dynamics is:

$$w=f(u_1,u_2\mid t)=\sum_{i=0}^{m}\sum_{j=0}^{n}c_{i,j}u_1^{\,i}(t)u_2^{\,j}(t) \tag{7.4.5}$$

Now consider step input $u_1(t)=U_1$ and $u_2(t)=U_2$ In this case, $w=f(u_1(t),u_2(t))$ is also a step signal and its Laplace transform is $L(f(w(t)))=f(U_1,U_2)/s$; further if the linear dynamics takes the following form: $G(s)=\dfrac{b_0}{a_0 s^n+a_1 s^{n-1}+\cdots+a_n}$

And denote $K_1(u) = L^{-1}(b_0 f(U_1, U_2)/s) = b_0 f(U_1, U_2)$ then with similar substitution, the linear dynamics can be written as:

$a_0 y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots y(t) = K_1(u(t))$, where $K_1(u_1(t), u_2(t)) = K(u_1(t), u_2(t))$ and $B(s) = b_0$ is a constant, where $K_1(u(t)) = K(u(t))$ and $B(s) = b_m$ is a constant.

As a result, the proposed identification for a specific Hammerstein structure as the linear dynamics and as the nonlinear static function in general can be achieved in two steps:

Step 1: Identify A(s) under different step input $\{U_{1i}, U_{2j}\}, i = 1, \cdots, m; j = 1,2, \cdots, n$

Because of the use of a step input $\{U_{1i}, U_{2j}\}$, the static nonlinearity $K_1(u_1, u_2) = K_1(U_{1i}, U_{2j}) = b_0 f(U_{1i}, U_{2j})$ is now a constant and can be treated as a parameter to be identified such that the parameters can be written as

$$\theta^T(U_{1i}, U_{2j}) = \begin{bmatrix} a_0 & \cdots & a_{n-1} & K(U_{1i}, U_{2j}) \end{bmatrix} \tag{7.4.6}$$

Hence, A(s) and the value of $b_0 f(U_{1j}, U_{2j})$ can be identified.

Step 2: Determine the nonlinear function $f(u_1, u_2)$ and b_0 from the steady-state responses $Y(U_{1i}, U_{2j})$ under step input $\{U_{1i}, U_{2j}\}, i = 1, \cdots, m; j = 1,2, \cdots, n$.

In this case,

$K(U_{1i}, U_{2j}) = \phi^T(U_{1i}, U_{2j})\theta$

with $\phi^T(U_{1j}, U_{2j}) = \begin{bmatrix} 1 & U_2^1 & U2_0^2 & \cdots & U_1^m U_2^n \end{bmatrix}$

and $\theta^T = \begin{bmatrix} b_0 c_0 & b_0 c_{12} & b_0 c_{13} & \cdots & b_0 c_{mn} \end{bmatrix}$ (7.4.7)

Then the parameters in $b_m f(U_j)$ can be determined using a linear least square algorithm as

$\theta = (\Phi^T \Phi)^{-1} \Phi^T [Y(U_{11}, U_{21}) \quad \cdots \quad Y(U_{1m}, U_{2n})]^T$

where $\Phi^T = [\phi^T(U_{11}, U_{21}) \quad \cdots \quad \phi^T(U_{1m}, U_{2n})]^T$

The proposed identification method obtains the parameters by estimating the linear dynamics and the static nonlinearity separately. And when the parameters of the linear

dynamics are available, the static nonlinearity is obtained based on the static signals acquired when the system is steady.

7.5 Extends to MIMO model

For simplification, we start with a 2I2O model

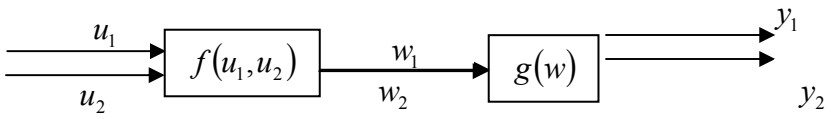


Figure 7.5 2I2O model

Assume the model structure can be represented by:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} + \begin{bmatrix} p_{11} & 0 \\ 0 & p_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$, where y_1 and y_2 are the output.

$$w_1 = f(u_1, u_2 \mid t) = \sum_{i=1}^{m} \sum_{j}^{n} c_{i,j} {u_1}^i(t) {u_2}^j(t) \tag{7.5.1}$$

$$w_2 = f(u_1, u_2 \mid t) = \sum_{i=1}^{m} \sum_{j}^{n} c'_{i,j} {u_1}^i(t) {u_2}^j(t) \tag{7.5.2}$$

Assume the predicted Hammerstein model is:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} + \begin{bmatrix} e_{11} & 0 \\ 0 & e_{22} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{7.5.3}$$

The parameters for the linear dynamics will be similar to single output system structure and for the nonlinearity, we can solve it by minimizing the following objective function:

$$J(c_{i,j}) = \int_0^t (y_1 - \hat{y}_1)^2 \, d\tau \tag{7.5.4}$$

$$J(d_{i,j}) = \int_0^t (y_1 - \hat{y}_1)^2 \, d\tau \tag{7.5.5}$$

By differentiating the objective function above with respect to

$$c_{i,j} \mid i = 1,2,...,m; j = 1,2,...,n \tag{7.5.6}$$

$$\frac{\partial J}{\partial c_{ij}} = 2\int_0^t (y_1 - \hat{y}_1) e_{11} \frac{\partial w_1}{\partial c_{ij}} \, d\tau \tag{7.5.7}$$

$$\frac{\partial w_1}{\partial c_{ij}} = u_1^{\,i} u_2^{\,j} \tag{7.5.8}$$

Accordingly, we will have:

$$\left[\{c(i,j) \mid i = 1,2,...,m; j = 1,2,...,n\}\right]^T = A_1^{-1}\left[\left\{\int_0^t (y_1 - a_{11}\dot{y}_1 - a_{12}\dot{y}_2 - b_{11}\ddot{y}_1 - b_{12}\ddot{y}_2) c_{11} u_1^{\,i} u_2^{\,j}\right\}\right]^T \tag{7.5.8}$$

, where $A = \left[\{A(i,j)\}\right]$, and $A(i,j) = e_{11} u_1^{\,i} u_2^{\,j} \int_0^t e_{11} u_1^{\,i} u_2^{\,j} \, d\tau$ (7.5.9)

Parameters for y_2 can be similarly obtained.

CHAPTER 8

HAMMERSTEIN IDENTIFICATION: EXPERIMENTAL STUDY

8.1 Experimental Setup

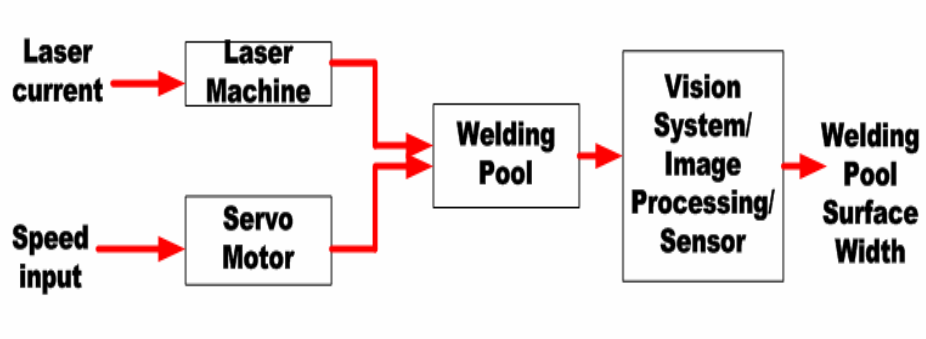


Figure. 8.1 Speed Input Open Loop Laser Welding System

The experimental system was set up as shown in Figure 8.1, using a laser driver to power the laser head which generates the laser to heat the work-piece. The laser energy is proportional to the current of the laser driver. The computer controls the welding process by adjusting the laser energy and the welding speed. A digital camera with a high speed shutter is equipped to capture the weld pool pictures at the speed of 30 frames per second. The welding material is 1mm mild steel sheet. The laser power is $1\,\mathrm{k}W$. Four parameters, i.e., laser energy, laser focus distance, incident laser angle, and welding speed influence the experimental process. When conducting the experiments, we optimally fix the focus distance and the incident angle at 89mm and 42 degree respectively. For better measurements, the laser energy is operated between 38mA and 58mA and the welding speed is controlled between 9.5mm/s and 13.5mm/s. In each experiment, only one of the parameters, i.e., either laser energy or speed is changed. The experiment data is observed until steady state is reached.

8.2 Experimental Data Analysis

An example of the raw images on the weld pool is shown in Figure 8.2.a. The processed image is shown in Figure. 8.2.b. The black field in the middle of the image is the liquid weld pool. Because all the acquired images are monochrome, i.e., black and white, the authors study the pixel intensity distribution of the images. Given the fact that the histogram counts the total number of pixels at each grayscale value and displays the data in graph, it is possible to see and isolate the distinct regions of interest containing certain grayscale intensities. The histogram result is shown in Fig8.2.c. According to the experimental study, there might be some relations between the intensity distribution of the pixels and the penetration of the welding process. In average, for the intensities of the pixels, the more they fall in the range [0 50], the more possibly full penetration takes place; the more they fall in the range [200 250], the more possibly there is no penetration; the more they fall in range [50 200], the more possibly part penetration happens. Although grayscale analysis is relatively easier to perform, it is very sensible to noises and can only make rough prediction on the weld penetration. For quality monitoring purposes, the topside weld pool width needs to be analyzed and controlled.

The open loop static responses under different laser currents and speed inputs are shown as in Figure 8.3.a and Figure 8.3.b. Based on the open loop responses shown in Fig. 8.3 and the observation on the width of the backside weld bead, the we find that (1) increasing the laser current increases the topside surface width of the weld pool and the penetration which is measured by the width of the backside weld bead; (2) increasing the traveling speed decreases the topside surface width of the weld pool and the penetration. However, sometimes the resultant effects of various factors can be compromised. For example, when laser current increases, the width of the welding pool increases, but for the same increase of the laser current, the resultant increase of the surface width of the welding pool will also depend on the welding speed, i.e., if the welding speed is relatively low, the increase is bigger; and if the welding speed is relatively high, the increase is smaller.

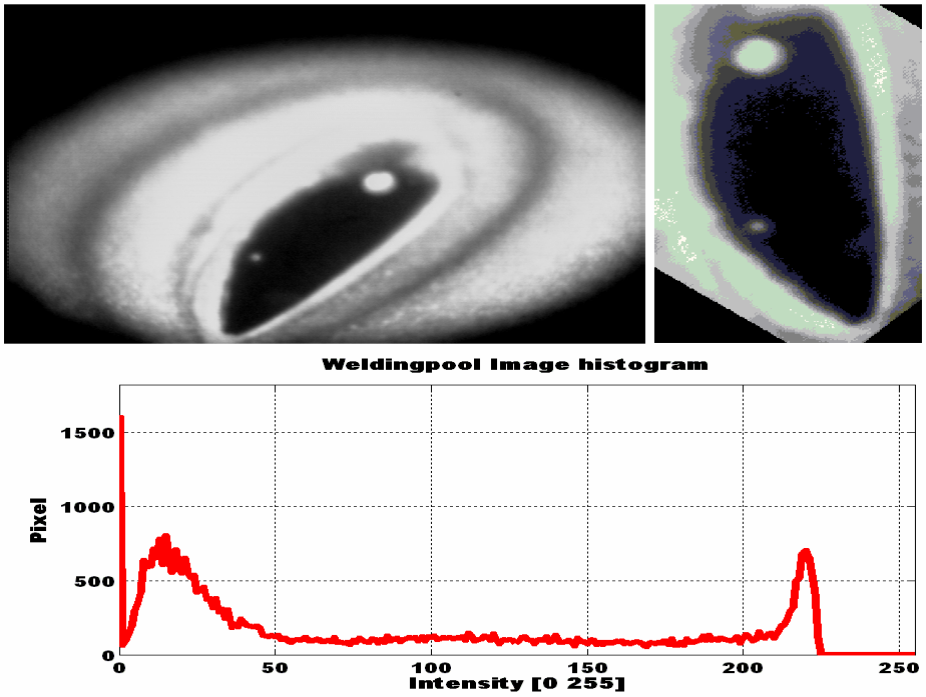


Figure. 8.2.a Raw image; Figure. 8.2.b Processed image

Figure. 8.2.c the histogram of the weld pool image

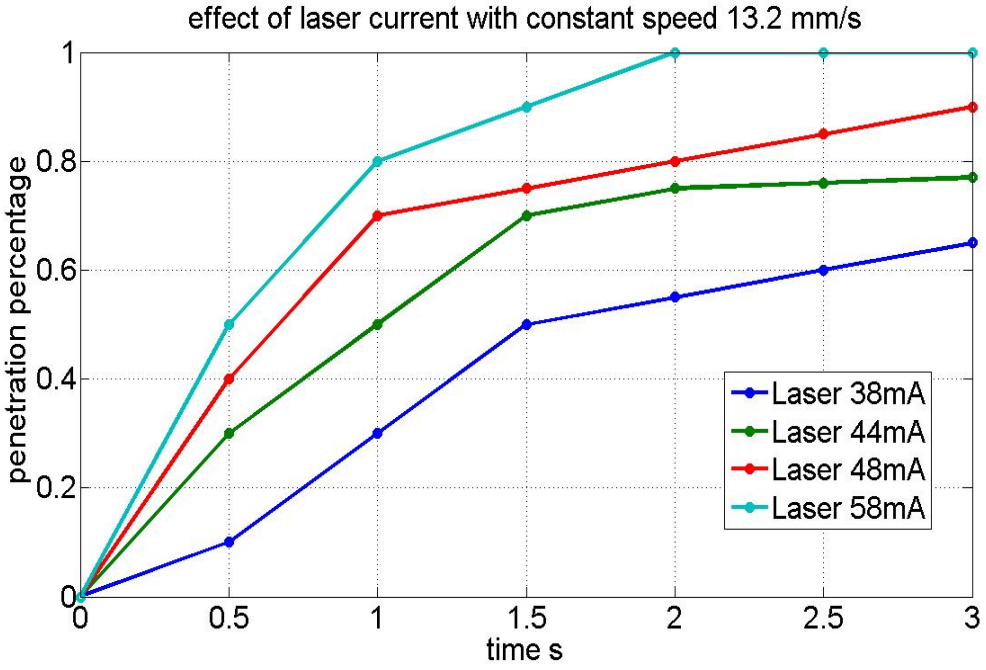


Figure 8.3.a Effects of the laser current on the penetration

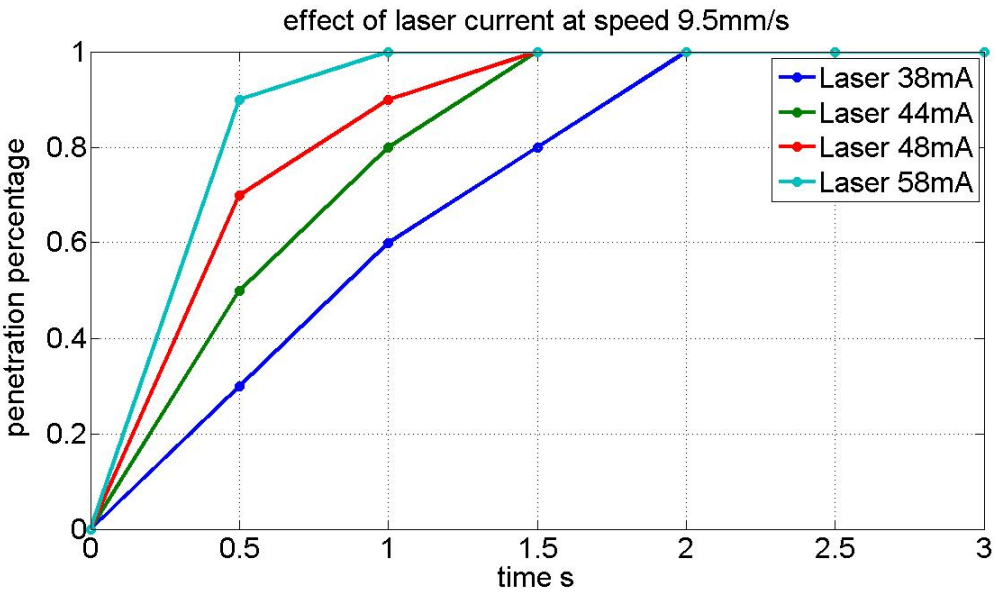


Figure 8.3.b Effects of the laser current on the penetration

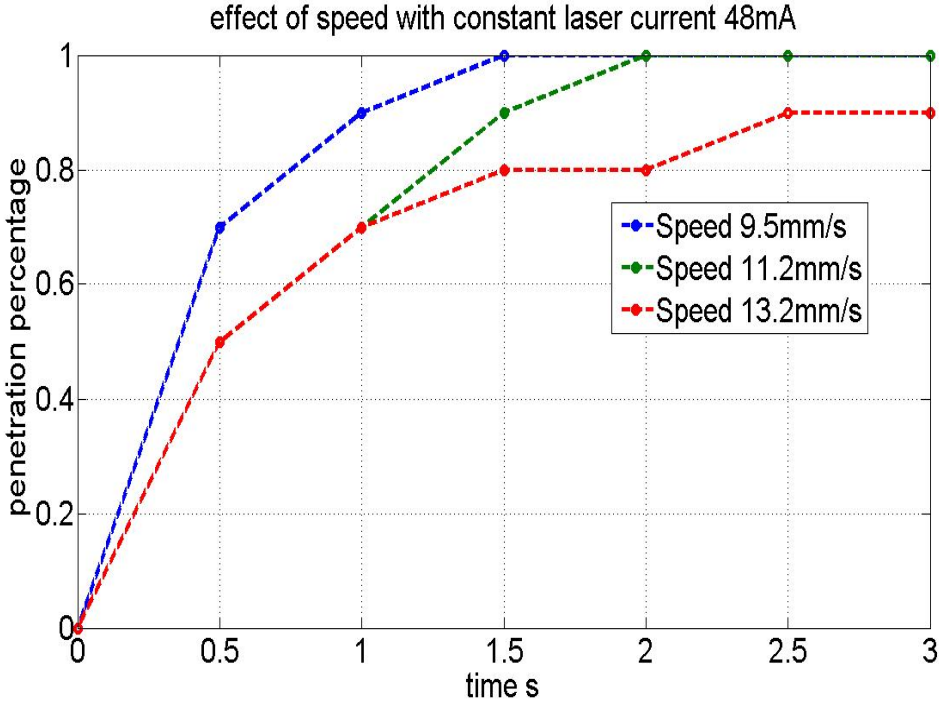


Figure 8.3.c Effects of the welding speed on penetration

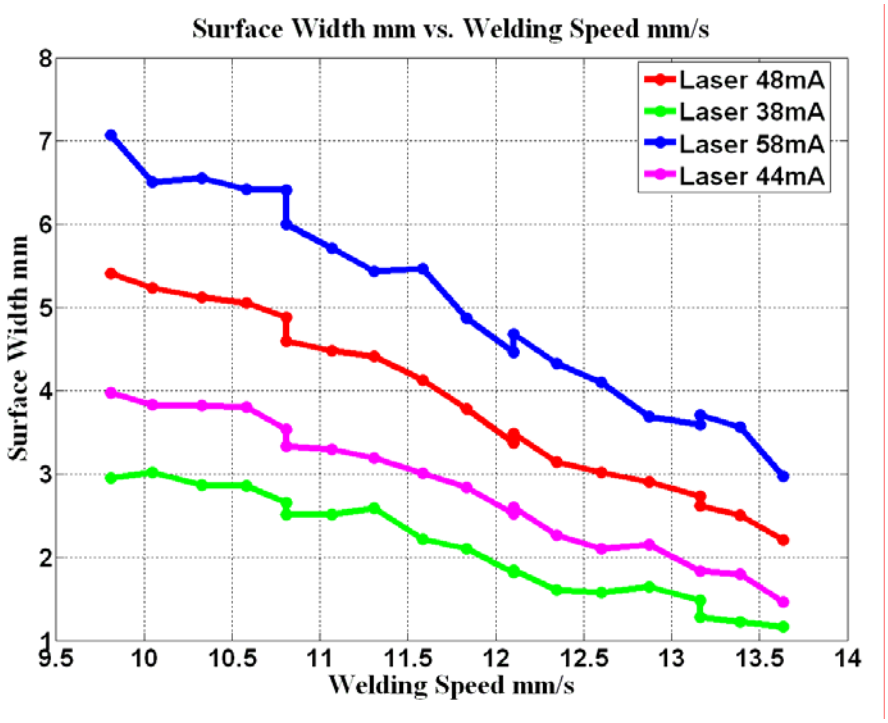


Figure 8.3.d Effects of the welding speed on surface width

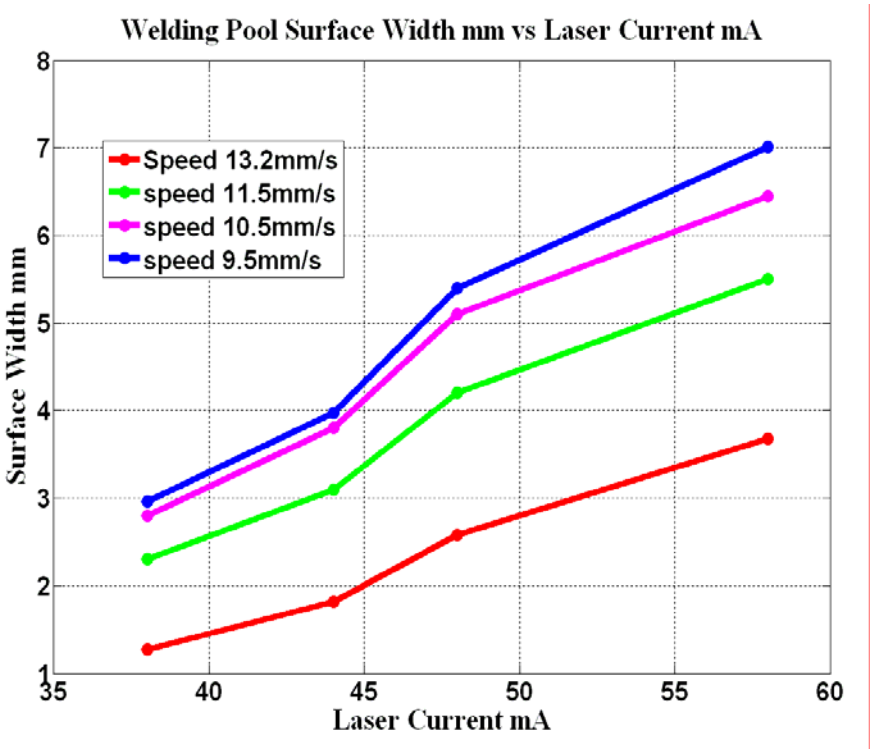


Figure 8.3.e Effects of the Laser energy on the surface width

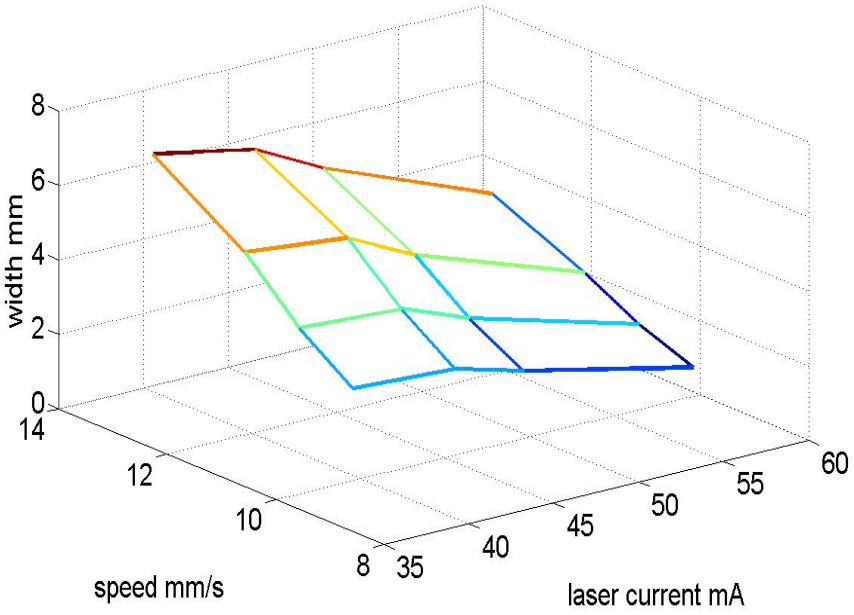


Figure 8.3.f Mesh graph for the correlation of parameters

Through experimental study, we noticed that the system step response shown in Figure. 8.4 with the welding speed as the input exhibits a standard second order system. This discovery is very beneficial especially for developing a suitable system identification procedure. Given the fact that Hammerstein structure consists of an input nonlinearity directly feeding into a linear dynamics, it might be possible to start with identifying a standard second order linear system and then the nonlinearity after conditional revision.

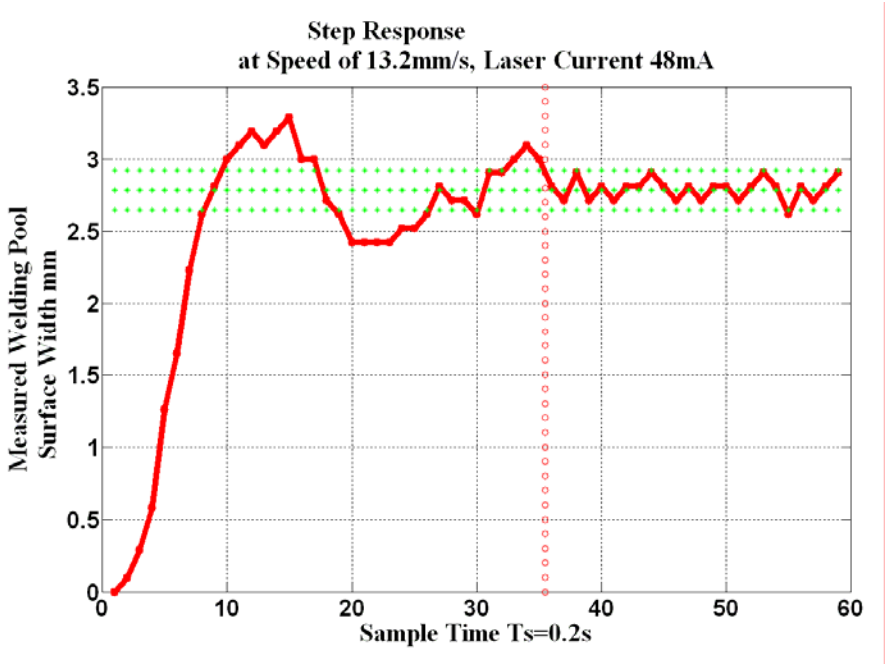


Figure 8.4 Step Response at speed of 13.2mm/s (laser current 48mA)

8.3 Experimental Identification on the Diode Laser System

For now, the authors only identify a SISO model in which $y(t)$ is the topside surface width of the weld pool and $u(t)$ is the reciprocal of the welding speed. Because the step response shown in Fig. 6 exhibits a second order linear model, hence the authors start with the structure:

$$T^2\ddot{y}(t)+2\xi T\dot{y}(t)+y(t)=K(u) \tag{8.3.1}$$

$$K(u)=b_m f(U_j) \tag{8.3.2}$$

Because only the position signals are directly measured in the experiments, the authors implement the optimal computing method shown in Figure 8.5a to search the parameters

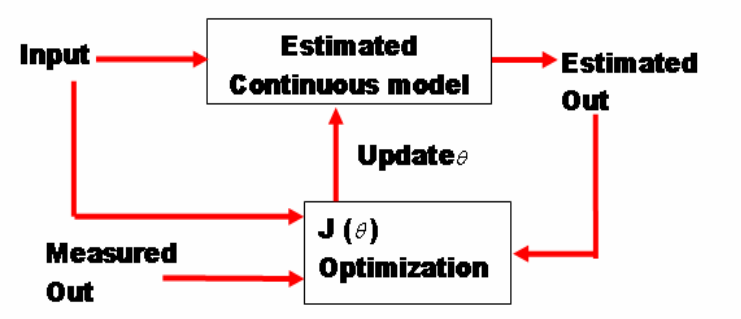


Figure 8.5.a Continuous Identification Process

For the optimization procedure, NEILDER-MEAD algorithm [107] shown in Fig. 8.5.b is applied:

Step 1:

Let the initial three estimated solution points be a , b and c , where $f(a) < f(b) < f(c)$and f is the function.

Step 2:

If the three points or their function values are sufficiently close to each other, then declare a as the minimum point and terminate the procedure.

Step 3:

Otherwise, choose new point e with $\begin{matrix} e = m + 2(m - c) \\ m = (a + b)/2 \end{matrix}$;

And if $f(e) < f(b)$, then take e as the new c;

Otherwise, define $r = (m + e)/2$;

And if $f(r) < f(c)$, then take r as the new c; if $f(r) \geq f(b)$, then define $s_1 = (c + m)/2$;

And if $f(s_1) < f(c)$, then choose s_1 as the new c;

Otherwise, define $c_1 = (a + c)/2$ and choose m and c_1 as the new points: b and c.

Step 4:

Go back to Step 1.

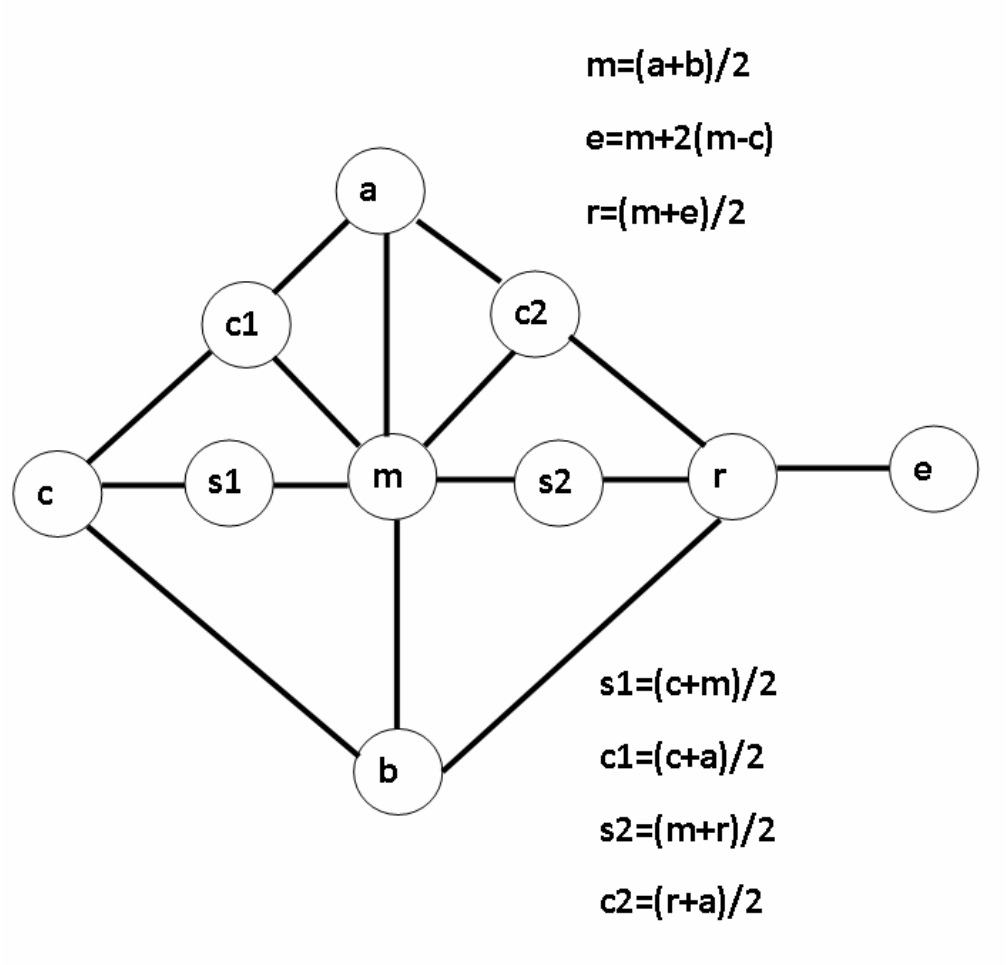


Figure. 8.5.b Optimal Search Process

8.3.1 Experimental Based Identification for Diode Laser Processing

Table 8.1

Continuous Linear Identification

laser (mA)	speed (mm/s)	$b_m \times f(U_j)$	Damp	Time Constant	IAE
48	9.81	5.4070	0.4367	0.32282	3.4108
48	10.05	5.2326	0.398	0.32282	4.2326
48	10.33	5.1163	0.4321	0.32282	3.6046
48	10.58	5.0581	0.3962	0.34253	3.8871
48	10.60	4.8457	0.4679	0.32727	2.1837
48	10.81	4.5930	0.4894	0.32282	3.9826
48	11.07	4.4767	0.4192	0.32282	4.0503
48	11.20	4.3995	0.39625	0.34253	2.1678
48	11.31	4.4186	0.4266	0.32282	4.1726
48	11.58	4.1279	0.3967	0.32282	3.7857
48	11.83	3.7791	0.3636	0.32282	3.309
48	12.10	3.3721	0.4266	0.32282	3.4977
48	12.35	3.1395	0.3925	0.32282	3.3981
48	12.60	3.0233	0.4186	0.32282	4.2736
48	12.87	2.9070	0.4284	0.32282	3.8998
48	13.20	2.7168	0.4667	0.32287	1.1111
48	13.39	2.5000	0.4544	0.32282	3.3386
48	13.64	2.2093	0.4478	0.32282	4.4505

As introduced, the SISO model to be identified is given by equation 8.3.1-2. For the identification, the authors apply the identification on the step signal based experimental data under inputs $\{U_j\}, j = 1, \cdots M$. These inputs correspond to those in Figure 8.3. The identification results are summarized in Table 8.1. It is easy to see that the static function $b_m f(u)$ varies with the inputs conditions, which verifies the existence of nonlinearity with the established diode laser welding process.

8.3.2 PRTS (Pseudo-Random Ternary signal) [108]

Besides of the step signals, the authors take PRTS shown in Figure 8.6 as the input and perform the experiments to study on the nonlinearity. The output is still the topside surface width of the weld pool. The sequence length of the PRTS signal is 26 and the sample time is 2 units per cycle. Due to the system capacity, only 52 units are sampled and used for the identification and model validation.

sample time is 2 units per cycle. Due to the system capacity, only 52 units are sampled and used for the identification and model validation.

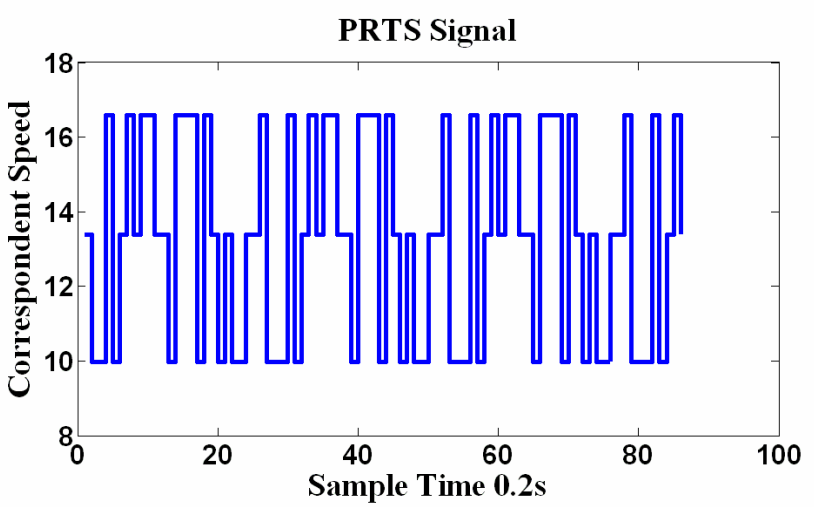
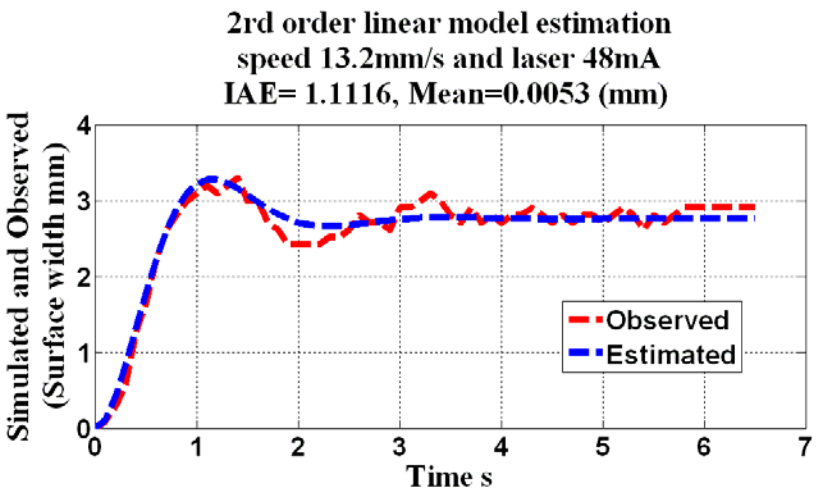


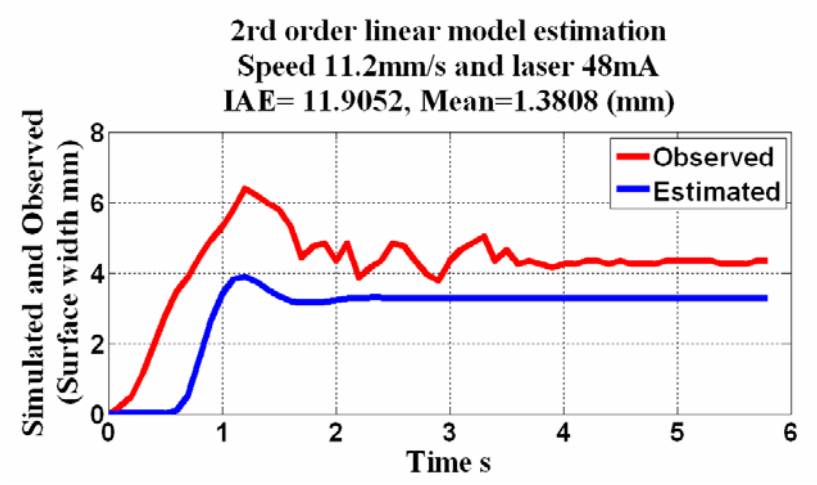
Figure 8.6 PRTS Signal

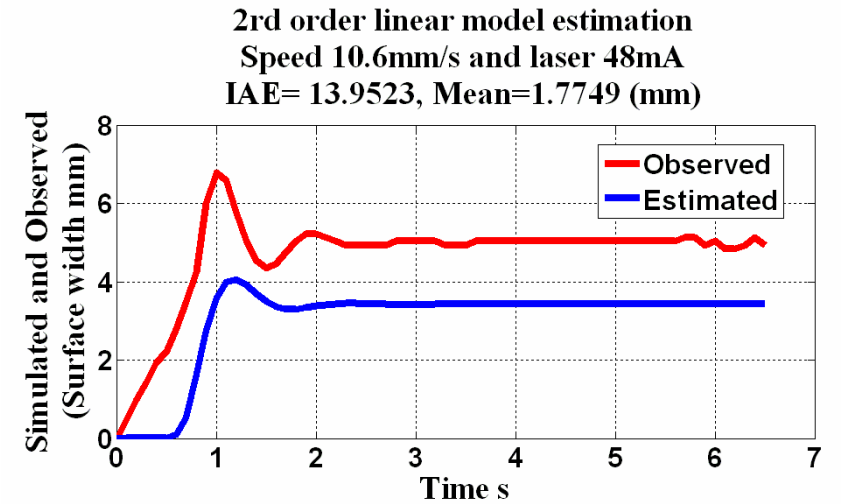
8.3.3 Nonlinearity Identification

Before identifying the nonlinearity, the authors study how it influences the linear dynamics. At first the authors take one of the linear models and simulate it under various inputs. Then the authors compare the simulations with the experiments. If the model exhibits consistent data agreements, then the linear dynamics is certainly enough to describe the system. Otherwise, further step is needed.

Without loss of generality, the authors chose the one under the input of 13.2mm/s (with the lowest IAE in Table 8.1) and examined the step responses under the inputs of 13.2mm/s, 11.2mm/s, and 10.6mm/s (IAE relatively smaller) and PRTS signal. The results are shown in Figure 8.7. It is easy to see that these results demonstrate different data agreements, which further verifies nonlinearity does exists in the diode laser welding process similarly as has been observed from another welding process [83]. As a result, step 2 is demanded for a complete nonlinear identification of the laser welding process.


2rd order linear model estimation
speed 13.2mm/s and laser 48mA
IAE= 1.1116, Mean=0.0053 (mm)
Simulated and Observed
(Surface width mm)
Time s
Observed
Estimated


2rd order linear model estimation
Speed 11.2mm/s and laser 48mA
IAE= 11.9052, Mean=1.3808 (mm)
Simulated and Observed
(Surface width mm)
Time s
Observed
Estimated


2rd order linear model estimation
Speed 10.6mm/s and laser 48mA
IAE= 13.9523, Mean=1.7749 (mm)
Simulated and Observed
(Surface width mm)
Time s
Observed
Estimated

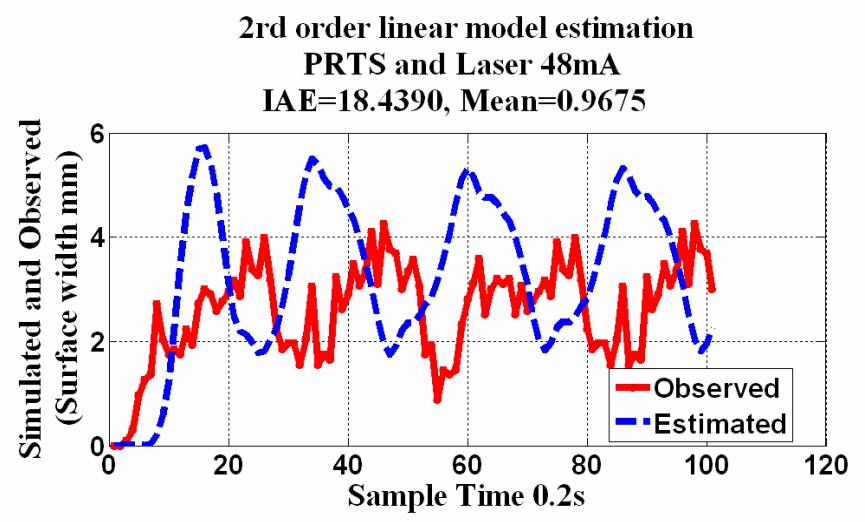


Figure 8.7 Simulated and observed comparison

For simplicity, the authors assume the nonlinearity in equation 8.3.2 is a polynomial. Based on the results from the linear dynamics identification, the authors develop two structures below and the identification results are shown in Figure 8.8:

$$K(u) = -0.11856u^3 + 2.9457u^2 - 23.0018u + 59.3253 \quad (8.3.3)$$

$$K(u) = 0.0048078u^4 - 0.28697u^3 + 5.1487u^2 - 35.7543u + 86.8896 \quad (8.3.4)$$

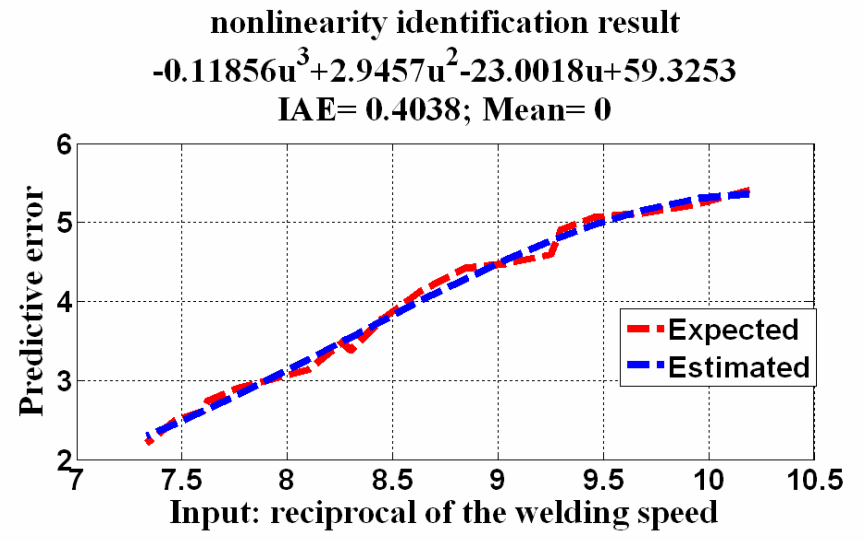


Figure 8.8.a Nonlinearity Identification 3rd order

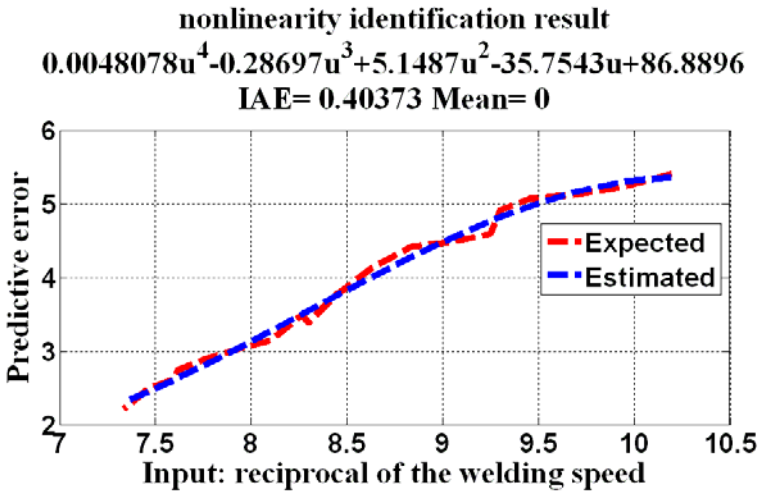


Figure 8.8.b Nonlinearity Identification 4th order

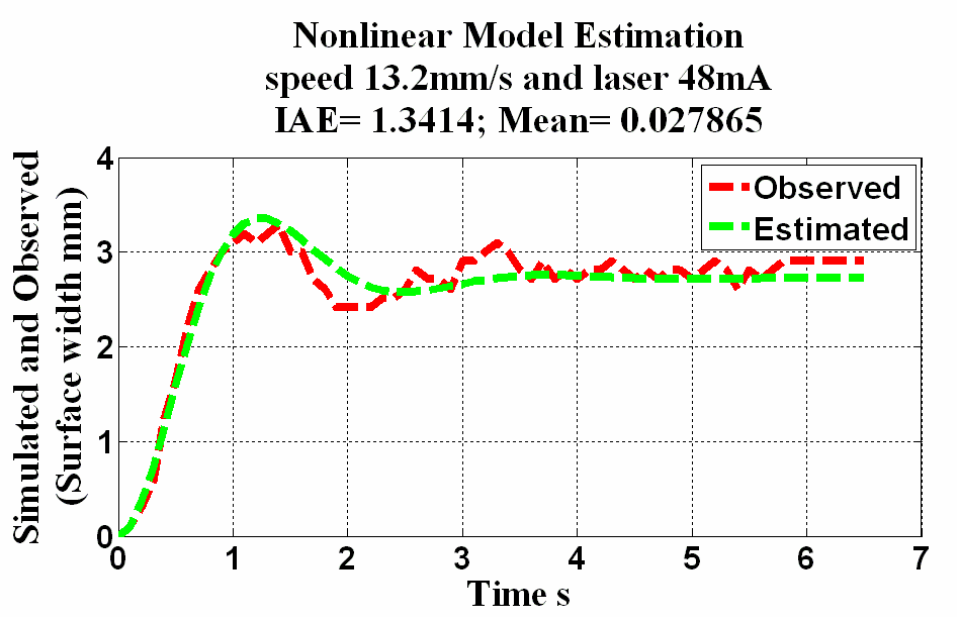
8.3.4 Order Determination and Model Validation

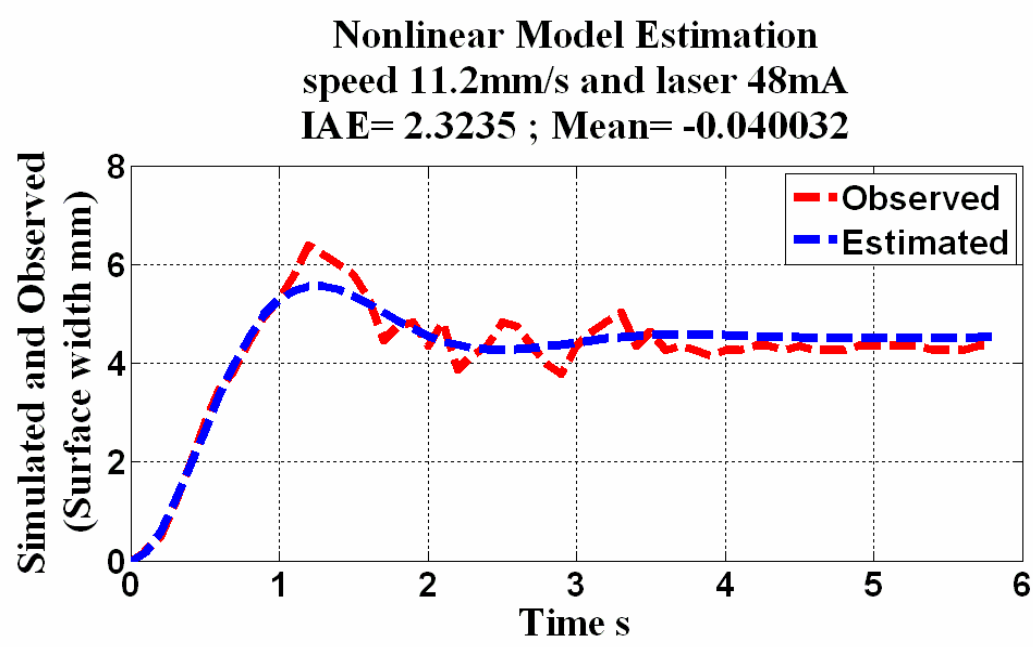
With equation 8.3.3 and 8.3.4, the complete models are:

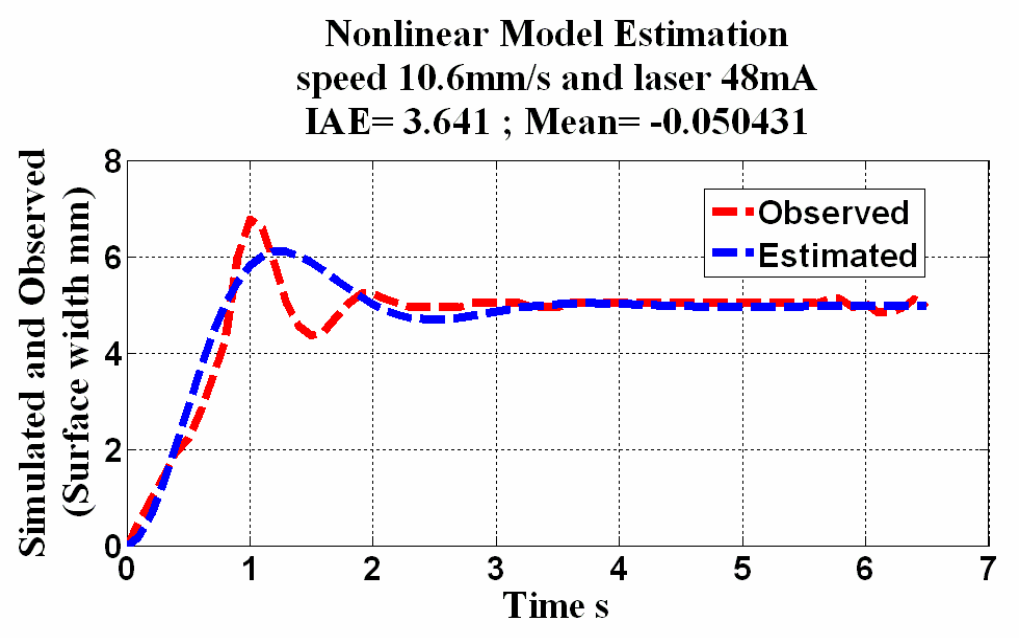
$$\begin{aligned} &0.1286\ddot{y}(t)+0.3015\dot{y}(t)+y(t) \\ &=-0.11856u^3(t) \\ &+2.9457u^2(t)-23.0018u(t)+59.3253 \end{aligned} \tag{8.3.5}$$

$$\begin{aligned} &0.1286\ddot{y}(t)+0.3015\dot{y}(t)+y(t) \\ &=0.0048078u^4(t)-0.28697u^3(t) \\ &+5.1487u^2(t)-35.7543u(t)+86.8896 \end{aligned} \tag{8.3.6}$$

To choose suitable order and validate the model, the authors run simulations on both equation under inputs of step signals including 13.2mm/s, 11.2mm/s and 10.6mm/s and PRTS signals. Then the authors examine the data agreements between the simulations and experiments. Based on the performance results in Table 8.2, the authors choose 8.3.6.


Nonlinear Model Estimation
speed 13.2mm/s and laser 48mA
IAE= 1.3414; Mean= 0.027865
Simulated and Observed
(Surface width mm)
Time s
Observed
Estimated
0
1
2
3
4
5
6
7


Nonlinear Model Estimation
speed 11.2mm/s and laser 48mA
IAE= 2.3235 ; Mean= -0.040032
Simulated and Observed
(Surface width mm)
Time s
Observed
Estimated
0
2
4
6
8
1
3
5



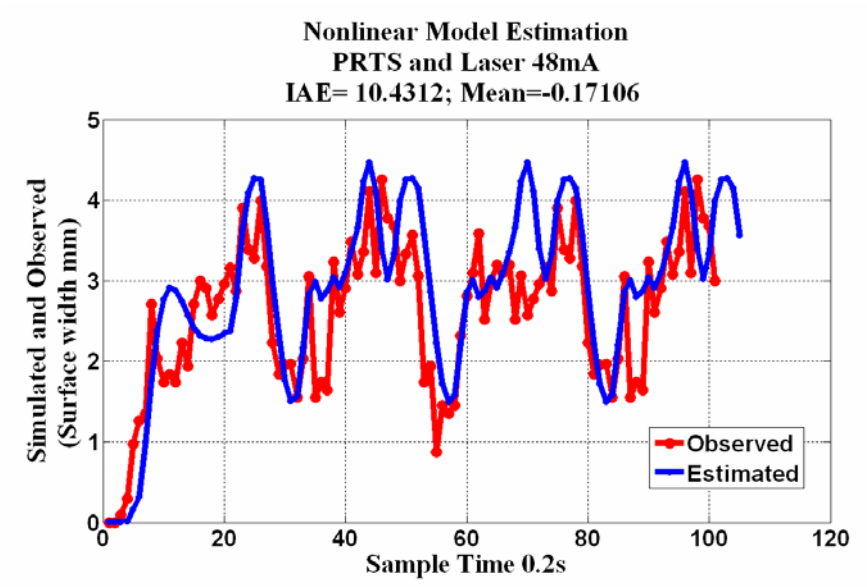


Figure 8.9 Model Evaluation

The validation results of equation 8.3.5-6 are shown in Figure 8.9. Based on the results, the authors conclude that the identification succeeds the validation and the identification model given by the 4th order nonlinearity equation 8.3.6 and the linear dynamics with $T = 0.32282$ and $\xi = 0.4667$ reasonably approximate the laser welding system. The authors nevertheless noticed that the IAE is higher under PRTS input. This deviation is

expected due to the effects of from the measurements errors, servo motor sensitivity, sampling completeness, signal noise, and so on.

Table 8.2

Continuous Identification Comparison of 3RD and 4TH Order Model

Laser (mA)		IAE of nonlinear model		Mean of Predictive Error	
		3rd order	4th order	3rd order	4th order
48	13.2	2.0724	1.3414	0.1852	0.02785
48	11.2	4.1654	2.3235	0.44419	0.040032
48	10.6	5.8421	3.6413	0.53468	0.050431
48	PRTS	10.4538	10.4312	0.48708	0.17106

8.4 Continuous identification: Error based

In the identification of the laser system, the authors take 13.2mm/s as the welding speed and obtain:

$$T^2 \times \ddot{y}(t) + 2 \times \xi \times T \times \dot{y}(t) + y(t) = K(U) \tag{8.4.1}$$

$$\begin{aligned} K(U) &= 2.7168 \\ T &= 0.32282 \\ \xi &= 0.4670 \end{aligned} \tag{8.4.2}$$

The linear dynamic parameters of equation above are now used as the estimates of those in the system despite variations in the inputs.

Now assume a number of steady-state responses $\{y_j(\infty)\}$ have been obtained under $\{U_j\}$ such that $\{y_j(\infty) \quad U_j\}, j = 1,2,\cdots M$ are available.

The objective is to identify the structure and the parameters for the nonlinear function $K(U)$ through $\{K(U_j) \quad U_j\}$.

To this end,

$T^2 \times \ddot{y}(t) + 2 \times \xi \times T \times \dot{y}(t) + y(t) = 1$ is simulated and the simulated steady-state responses $\hat{y}_j(\infty)$ is used to compute

$$K(U_j) = y_j(\infty) / \hat{y}_j(\infty)\big|_{K(U)=1} \tag{8.4.3}$$

As a result, a set of $\{K(U_j), \quad U_j\}$ will be easily obtained through a single simulation of linear system using

$$T^2 \times \ddot{y}(t) + 2 \times \xi \times T \times \dot{y}(t) + y(t) = 1 \tag{8.4.4}$$

and the static response experimental results as given above.

Then for any proposed structure of $K(U)$, its parameters can be identified using as linear least square algorithm.

8.4.1 Nonlinearity Result

The authors similarly identify two models to compare for selection. The results are shown in Figure 8.9

$$\begin{aligned} f &= -0.11711u^3(t) + 2.9097u^2(t) \\ &- 22.7048u(t) + 58.5147 \end{aligned} \tag{8.4.5}$$

$$\begin{aligned} f &= 0.00593u^4 - 0.32845u^3(t) \\ &+ 5.6269u^2(t) - 38.4342u(t) + 92.5132 \end{aligned} \tag{8.4.6}$$

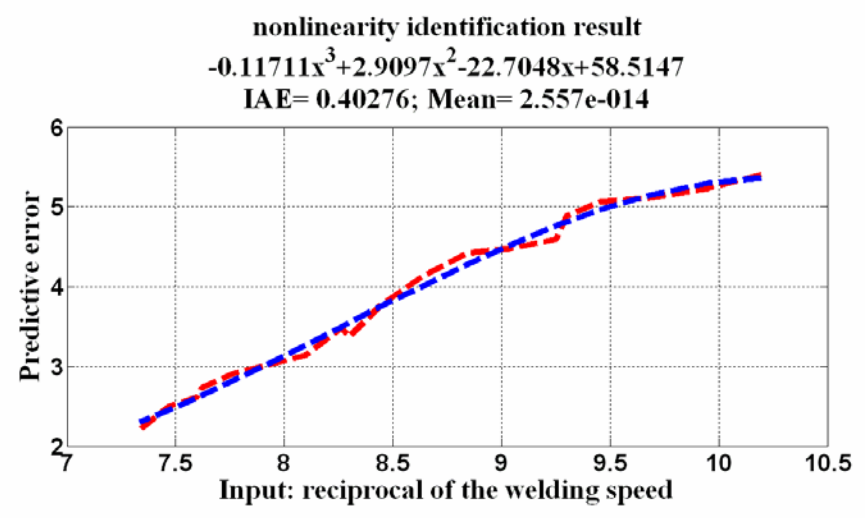


Figure 8.10 Nonlinearity identification

8.4.2 Complete Nonlinear Model Structure

With equation 8.4.5-6, the authors establish the complete model:

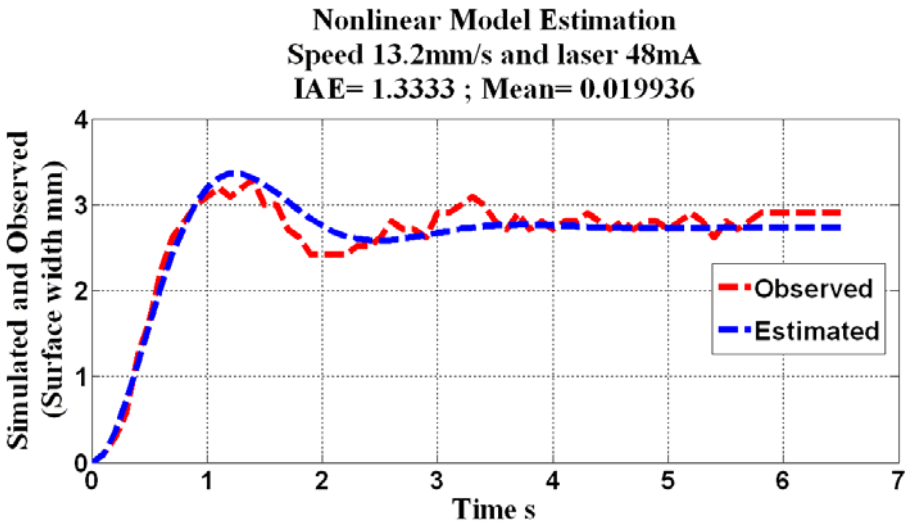
$$0.1286\ddot{y}(t)+0.3015\dot{y}(t)+y(t)=-0.11718u^3(t) + 2.9097u^2(t)-22.7048u(t)+58.5147 \tag{8.4.7}$$

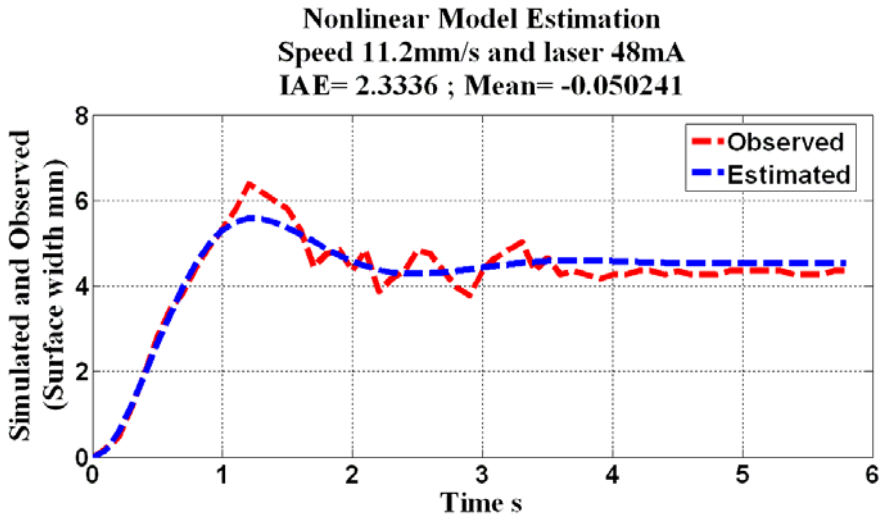
$$0.1286\ddot{y}(t)+0.3015\dot{y}(t)+y(t)=0.00593u^4(t) - 0.32845u^3(t)+5.6269u^2(t) - 38.4342u(t)+92.5132 \tag{8.4.8}$$

The validation results of both models under inputs of step signals including 13.2mm/s, 11.2mm/s and 10.6mm/s and PRTS signals are summarized in Table 8.3. The authors fundamentally examine the data agreements between the simulations results of the model and the experimental data. Based on the results comparison, the authors simply choose equation 8.4.8 because it performs relatively better in various inputs conditions. The validations under various inputs are also shown in Figure 8.10. Generally speaking, this identified nonlinear model succeeds the validation and is appropriates enough to describe our system.

Table 8.3 Error based Continuous Nonlinear Identification

Laser (mA)		IAE of nonlinear model		Mean of Predictive Error	
		3rd order	4th order	3rd order	4th order
48	13.02	1.5999	1.3333	0.11098	0.019936
48	11.2	2.5224	2.3336	0.118	0.050241
48	10.6	3.7654	3.658	0.1446	0.062992
48	PRTS	10.5548	10.5038	0.23223	0.17355


Nonlinear Model Estimation
Speed 13.2mm/s and laser 48mA
IAE= 1.3333 ; Mean= 0.019936
Simulated and Observed
(Surface width mm)
Time s
Observed
Estimated
0
1
2
3
4
5
6
7


Nonlinear Model Estimation
Speed 11.2mm/s and laser 48mA
IAE= 2.3336 ; Mean= -0.050241
Simulated and Observed
(Surface width mm)
Time s
Observed
Estimated
0
2
4
6
8
1
3
5

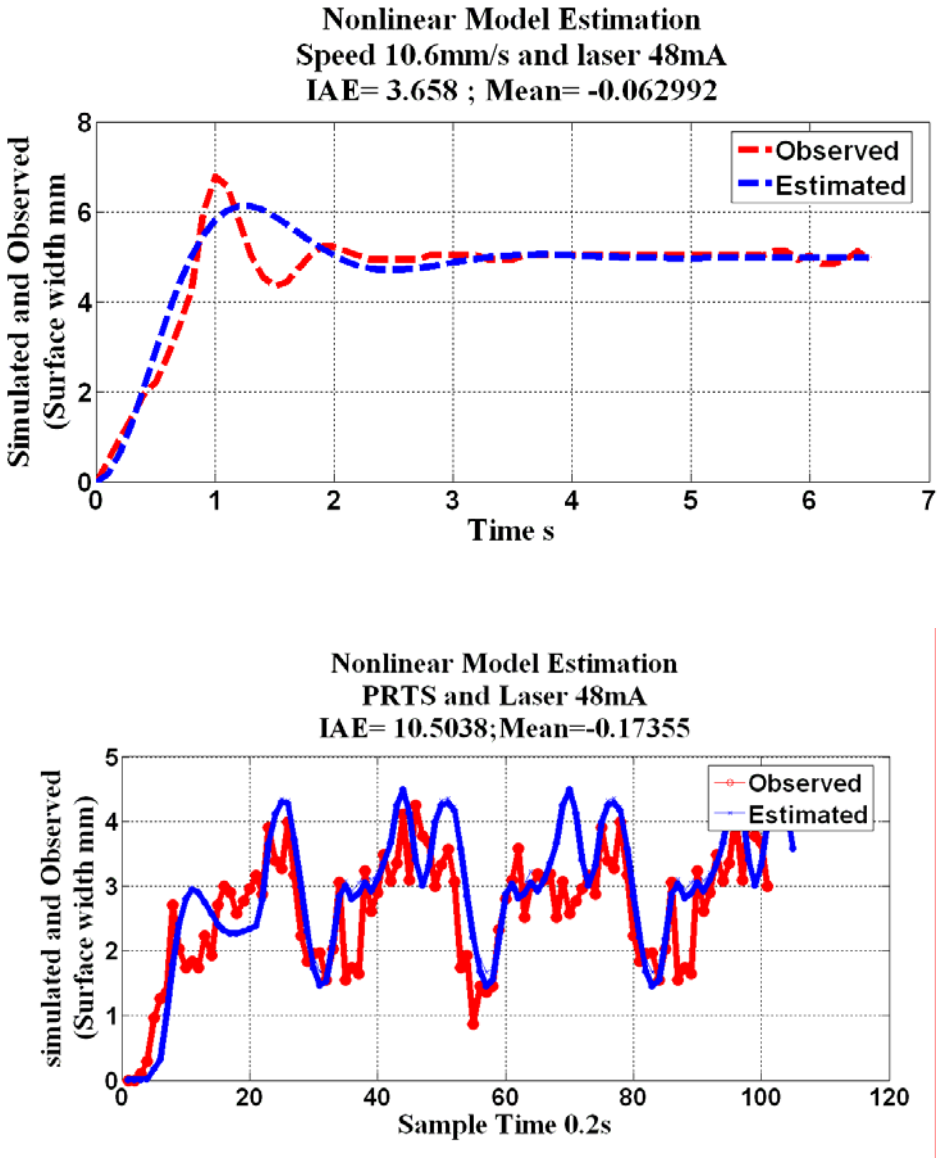


Figure 8.11 Error based Model Validation

8.4.3 Validation Results: Review

TABLE 8.4

CONTINUOUS NONLINEAR IDENTIFICAITON

Laser (mA)	Speed (mm/s)	IAE of linear model	Error based nonlinear model		Hammerstein nonlinear model	
			IAE	Mean of predictive error	IAE	Mean of predictive error
48	13.02	1.11	1.3333	0.019936	1.3414	0.027865
48	11.2	11.905	2.3336	0.050241	2.3225	0.040032
48	10.6	13.952	3.658	0.062992	3.641	0.050431
48	PRTS	18.438	10.5038	0.17355	10.4312	0.17106

For comparison purposes, the authors summarize the validations results of using both identification methods in Table 8.4. Based on the comparison, it is of no double that the nonlinear model demonstrates much better than the linear dynamics when approximating the diode laser welding system. From the perspective of nonlinear identification, both nonlinear methods work well on the laser welding system. The Hammerstein based identification performs slightly better than the Error Based Method because the Mean of Predictive error and IAE are slightly smaller under most working conditions. In fact, when the non-linearity is significant, its effect on the system behavior would be much more significant than that of the linear dynamics. For such a system dominated by non-linearity, it may be advantageous to use the much less time-consuming Error Based Method which does not need dynamic responses under different step responses and is thus may be more affordable.

8.5 Discrete Identification: Another Perspective

Similar as in Continuous identification, the authors only identify a SISO model in which $y(t)$ is the topside surface width of the weld pool and $u(t)$ is the reciprocal of the welding speed. Because the continuous identification already estimated a second order

model and the experimental step response also exhibits a second order model, hence the authors start with the structure:

$$T^2 y(k) + 2\xi T y(k-1) + y(k-2) = S(u) \tag{8.5.1}$$

$$S(u) = b_m f(U_j) \tag{8.5.2}$$

Then linear Least Square method is used to obtain the unknown parameters $T^2, 2\xi T$ and $K(U_j)$ from a step response under the step input U_j

8.5.1 Linear Dynamics Identification

The SISO discrete model to be identified is given by equation 8.5.1-2. For identification, the authors apply the step signal based experimental data under inputs $\{U_j\}, j = 1, \cdots M$. These inputs are still similar to those in the continuous identification. The identification results are summarized in Table 8.5. It is easy to see that the static function $b_m f(u)$ varies with the inputs conditions, which verifies the existence of nonlinearity

TABLE 8.5

DISCRETE LINEAR IDENTIFICAITON

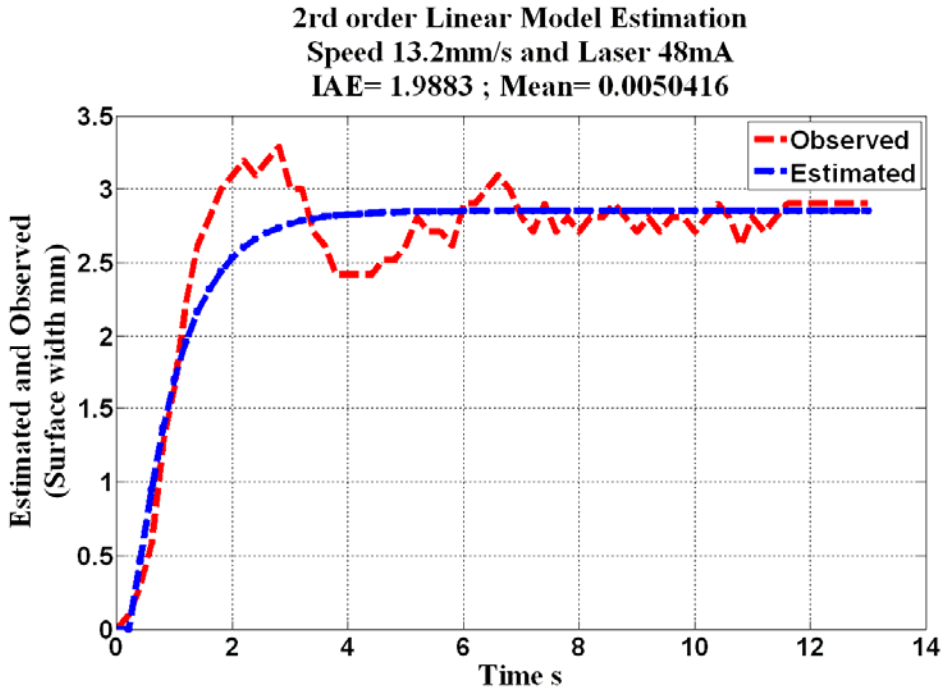
laser (mA)	speed (mm/s)	$b_m \times f(U_j)$	a1(y(k-1))	a2(y(k-2))	IAE
48	9.81	0.8684	1.0716	-0.2275	3.2721
48	10.05	0.8404	1.0081	-0.2275	4.7386
48	10.33	0.8217	1.2906	-0.2275	4.1594
48	10.58	0.8123	1.2968	-0.2275	4.0997
48	10.60	0.7782	1.0562	-0.2275	1.6401
48	10.81	0.7376	1.0716	-0.2275	3.2899
48	11.07	0.7190	1.1573	-0.2275	4.7061
48	11.20	0.7066	1.0286	-0.2275	2.3632
48	11.31	0.7096	0.9961	-0.2275	3.7019
48	11.58	0.6629	1.0124	-0.2275	4.0265
48	11.83	0.6069	1.2508	-0.2275	3.8036
48	12.10	0.5416	1.2178	-0.2275	3.1519
48	12.35	0.5042	1.1872	-0.2275	3.4798
48	12.60	0.4855	0.4186	-0.2275	3.2466
48	12.87	0.4669	0.4284	-0.2275	3.3678
48	13.20	0.4363	1.0669	-0.2275	1.0436
48	13.39	0.4015	1.2178	-0.2275	4.3215
48	13.64	0.3548	1.1715	-0.2275	3.7586

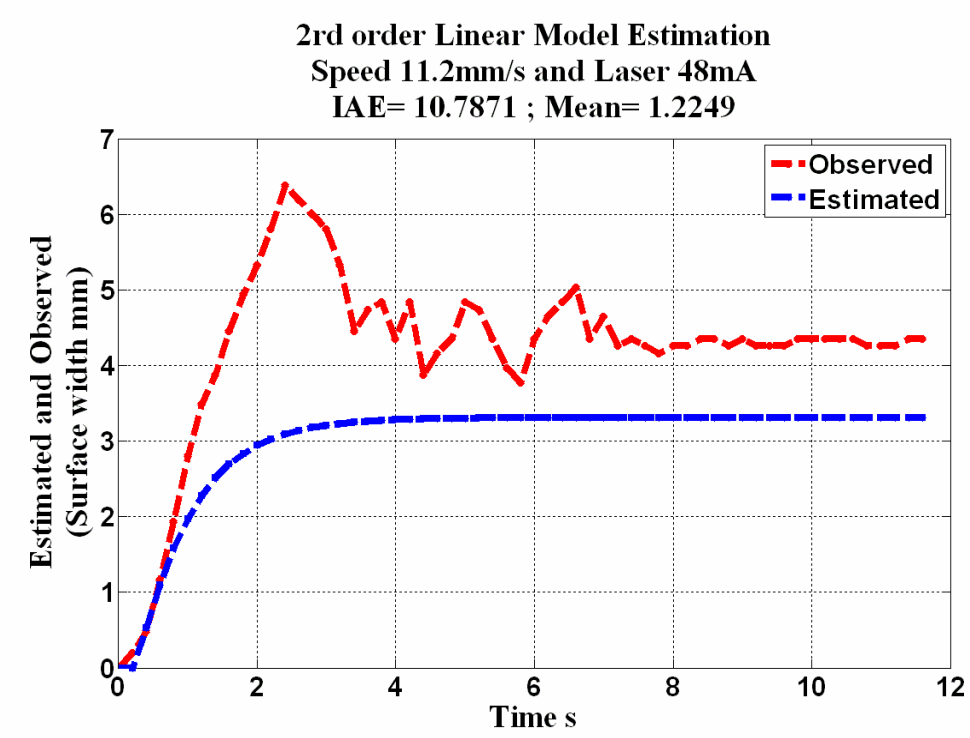
8.5.2 Influence of the Nonlinearity

Before identifying the nonlinearity, the authors study how it influences the linear dynamics. At first the authors take one of the linear models and simulate it under various inputs. Then the authors compare the simulations with the experiments. If the model exhibits consistent data agreements, then the linear dynamics is certainly enough to describe the system. Otherwise, further step is needed.

Without loss of generality, the authors chose the one under the input of 13.2mm/s (with the lowest IAE in Table 8.5) and examined the step responses under the inputs of 13.2mm/s, 11.2mm/s, and 10.6mm/s (IAE relatively smaller) and PRTS signal. The results are shown in Fig. 8.12. It is easy to see that these results demonstrate different data agreements, which further verifies nonlinearity does exists in the diode laser welding

process similarly as the continuous identification. As a result, step 2 is required for a complete nonlinear identification of the laser welding process.





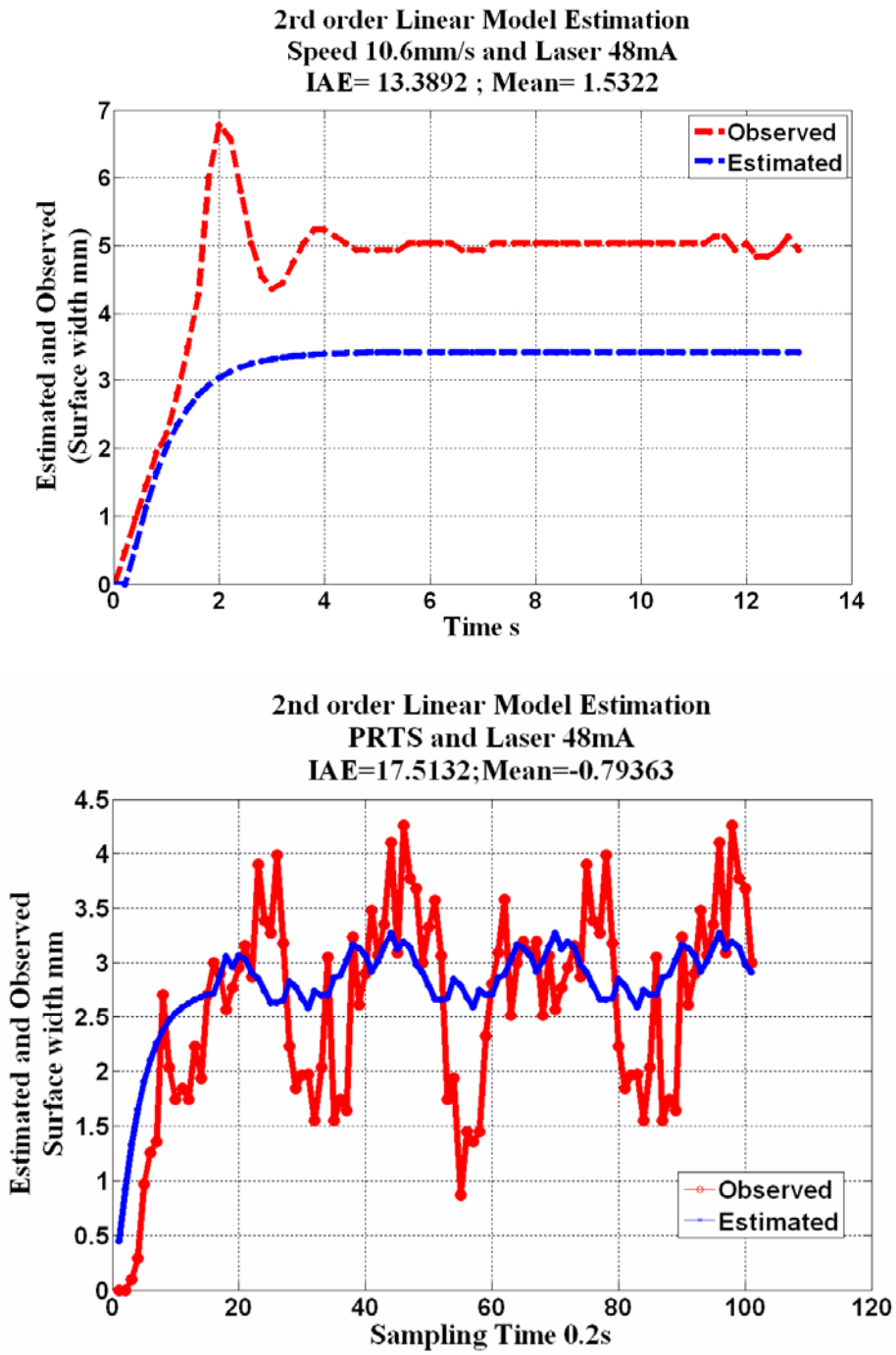


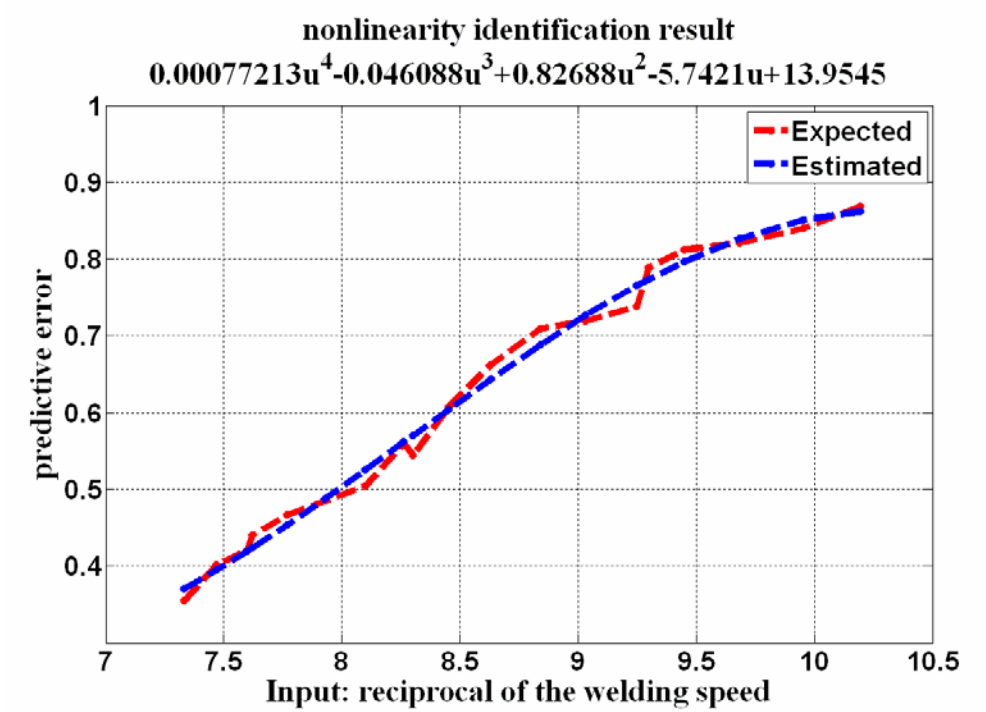
Figure 8.12 Simulated and observed comparison

8.5.3 Nonlinearity Identification

For simplicity, the authors assume the nonlinearity is a polynomial. Based on the results from the linear dynamics identification, the authors develop two structures below and the identification results are shown in Figure 8.13

$$K(u) = -0.01904u^3 + 0.47308u^2 - 3.6941u + 9.5276 \tag{8.5.3}$$

$$K(u) = 0.00077213u^4 - 0.046088u^3 + 0.82688u^2 - 5.7421u + 13.9545 \tag{8.5.4}$$



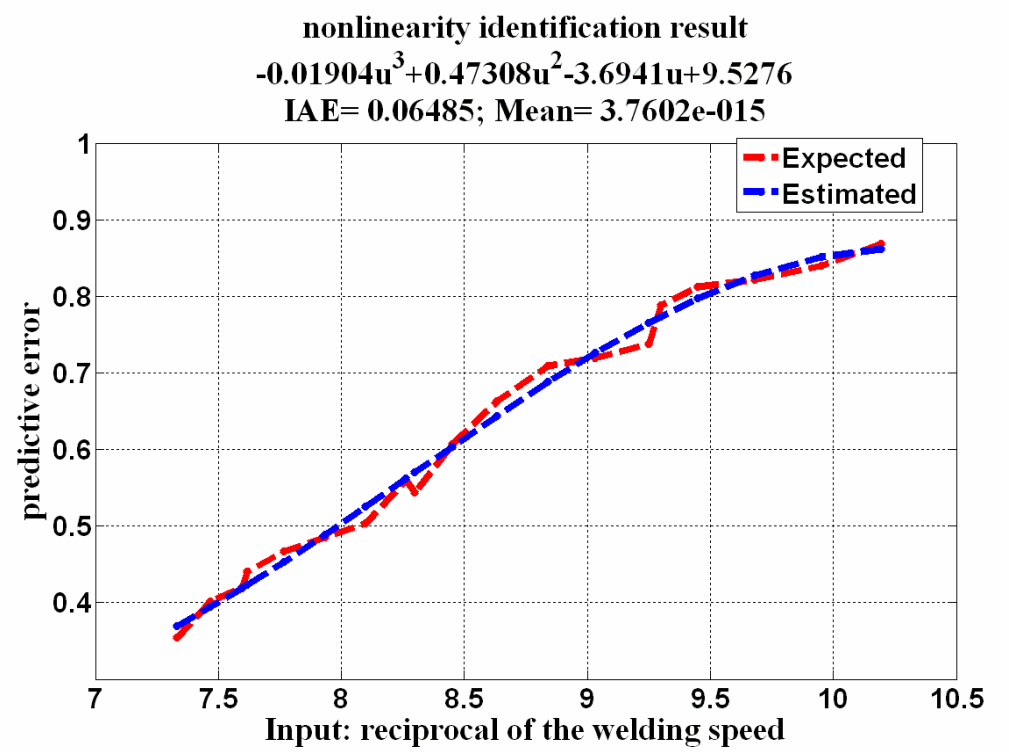


Figure 8.13 Nonlinearity Identification

8.5.4 Order Determination and Model Validation

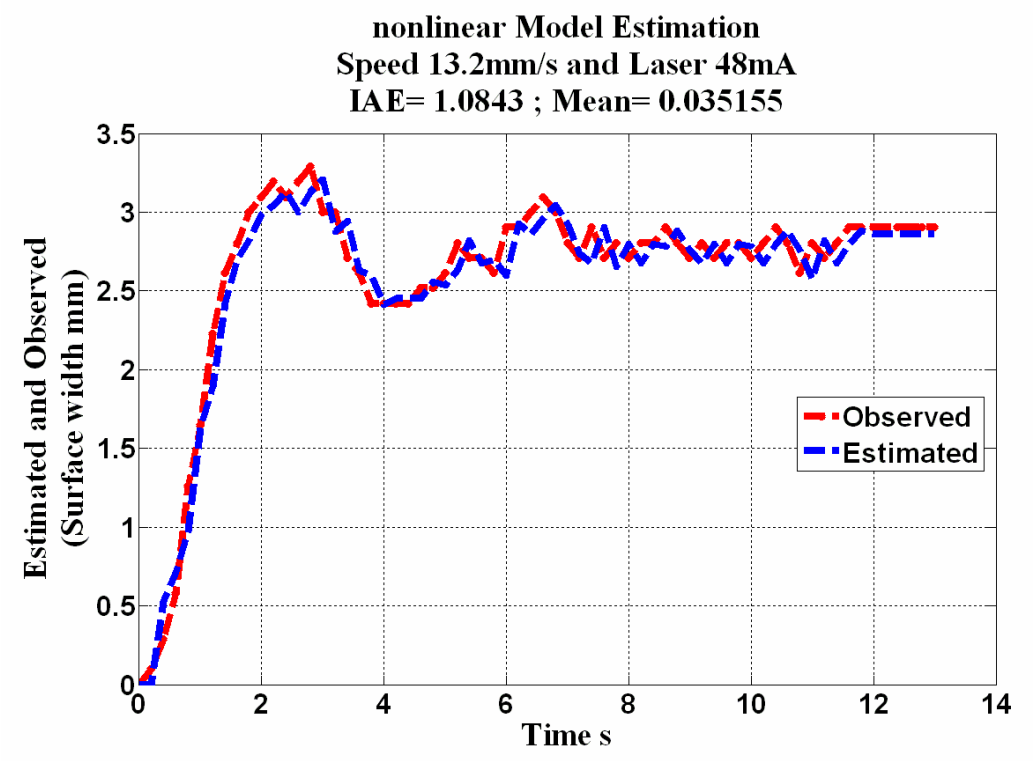
The complete models are:

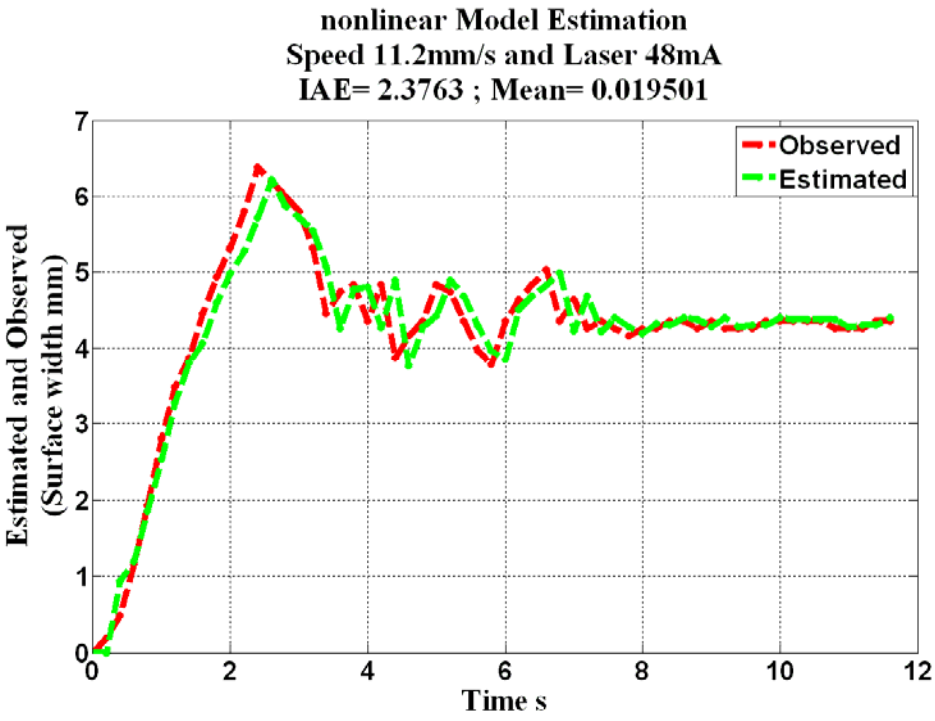
$$\begin{aligned} &y(k)-1.0669y(k-1)+0.2275y(k-2)\\ &=-0.019046u^3(k)\\ &+0.47308u^2(k)-3.6941u(k)+9.5276 \end{aligned} \tag{8.5.5}$$

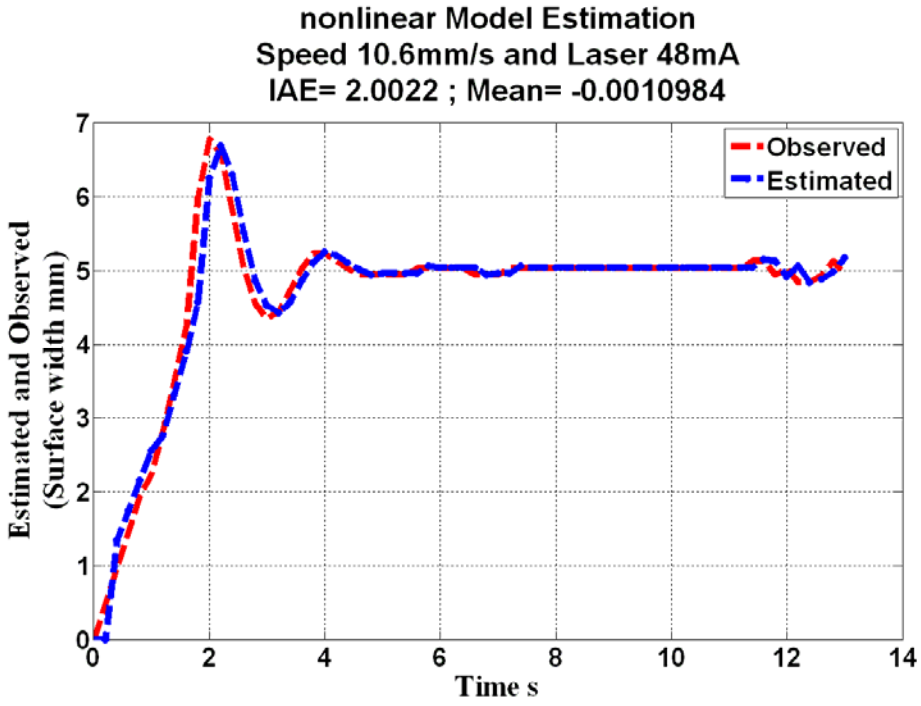
$$\begin{aligned} &y(k)-1.0669y(k-1)+0.2275y(k-2)\\ &=0.00077213u^4(k)-0.046088u^3(k)\\ &+0.82688u^2(k)-5.7421u(k)+13.9545 \end{aligned} \tag{8.5.6}$$

Table 8.6 Discrete Identification Results Comparison of 3RD and 4TH Order

Laser (mA)		IAE of nonlinear model		Mean of Predictive Error	
		3rd order	4th order	3rd order	4th order
48	13.2	2.0724	1.3414	0.1852	0.02785
48	11.2	4.1654	2.3235	0.44419	0.040032
48	10.6	5.8421	3.6413	0.53468	0.050431
48	PRTS	10.4538	10.4312	0.48708	0.17106




nonlinear Model Estimation
Speed 11.2mm/s and Laser 48mA
IAE= 2.3763 ; Mean= 0.019501
Observed
Estimated
Estimated and Observed
(Surface width mm)
Time s
0
1
2
3
4
5
6
7
0
2
4
6
8
10
12


nonlinear Model Estimation
Speed 10.6mm/s and Laser 48mA
IAE= 2.0022 ; Mean= -0.0010984
Observed
Estimated
Estimated and Observed
(Surface width mm)
Time s
0
1
2
3
4
5
6
7
0
2
4
6
8
10
12
14

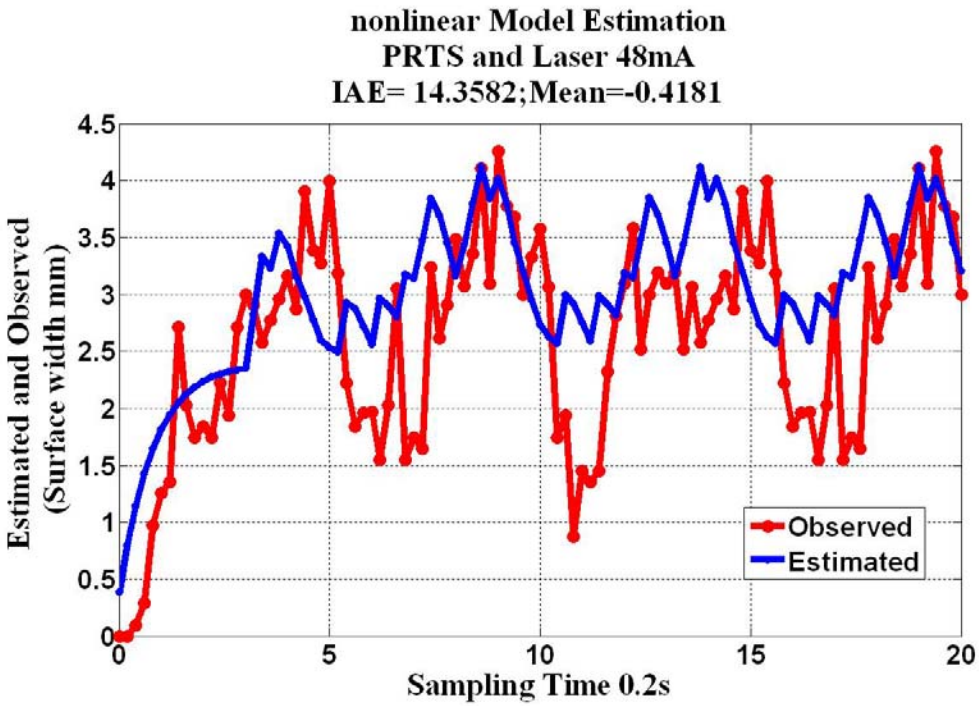


Figure 8.14 Model Validation

To choose suitable order and validate the model, the authors run simulations on both modes under inputs of step signals including 13.2mm/s, 11.2mm/s and 10.6mm/s and PRTS signals. Then the authors examine the data agreements between the simulations and experiments as shown in Table 8.6. Simply the authors choose 8.5.6

The validation results are shown in Figure 8.14 under different inputs signals. Based on the results, the authors conclude that the identification succeeds the validation and the identification model given by the 4^{th} order nonlinearity and the linear dynamics with $T = 0.32282$ and $\xi = 0.4667$ reasonably approximate the laser welding system. The authors nevertheless noticed that the IAE is higher under PRTS input. This deviation is expected due to the effects of from the measurements errors, servo motor sensitivity, sampling completeness, signal noise, and so on

8.6 Summary

The step response of the established diode laser welding process demonstrates a second order linear system. However, the gain of the linear dynamics varies with the input.

Hence, the authors proposed a two steps Hammerstein identification procedure both in continuous and discrete case: (1) detect the linear dynamics and the static non-linear function of the input from step responses with different step levels; (2) identify the parameters in the static nonlinear function. Besides of the Hammerstein identification, the authors also propose an Error Based identification and specifically apply the method on our system. The difference from the Hammerstein identification is the second step, which is based on the error of the linear dynamics and the static states under various inputs. A SISO nonlinear continuous and discrete model under using both procedures respectively is then estimated based on the experimental study. Both models take the reciprocal of the welding speed as the input and the topside surface width of the weld pool as the output. How to determine the suitable order for the identification is also introduced. To validate the identification, the authors examine the responses of the identified model under different inputs signals ranging from step signal to PRTS series. Validation results demonstrate better data agreements compared to the linear model structure. Both methods work adequately on our system but Hammerstein is more generally applicable for other similar process. It seems the discrete model approaches the experimental data somewhat roughly better than the continuous model. It is nevertheless noticed that the PRTS validation still generates bigger residuals. This is expected because of the influence from servo motor, image processing, and sampling completeness.

8.7 2ISO Experimental Based Study

In our earlier study, we have developed a SISO Hammerstein model with welding speed as input and the width on the top surface of the weld pool, in which the linear dynamics was second order and the static nonlinearity was a standard power series. The model is as follows:

$$0.1286\ddot{y}(t)+0.3015\dot{y}(t)+y(t)=0.0048078u^4(t)-0.28697u^3(t) \\ +5.1487u^2(t)-35.7543u(t)+86.8896 \tag{8.7.1}$$

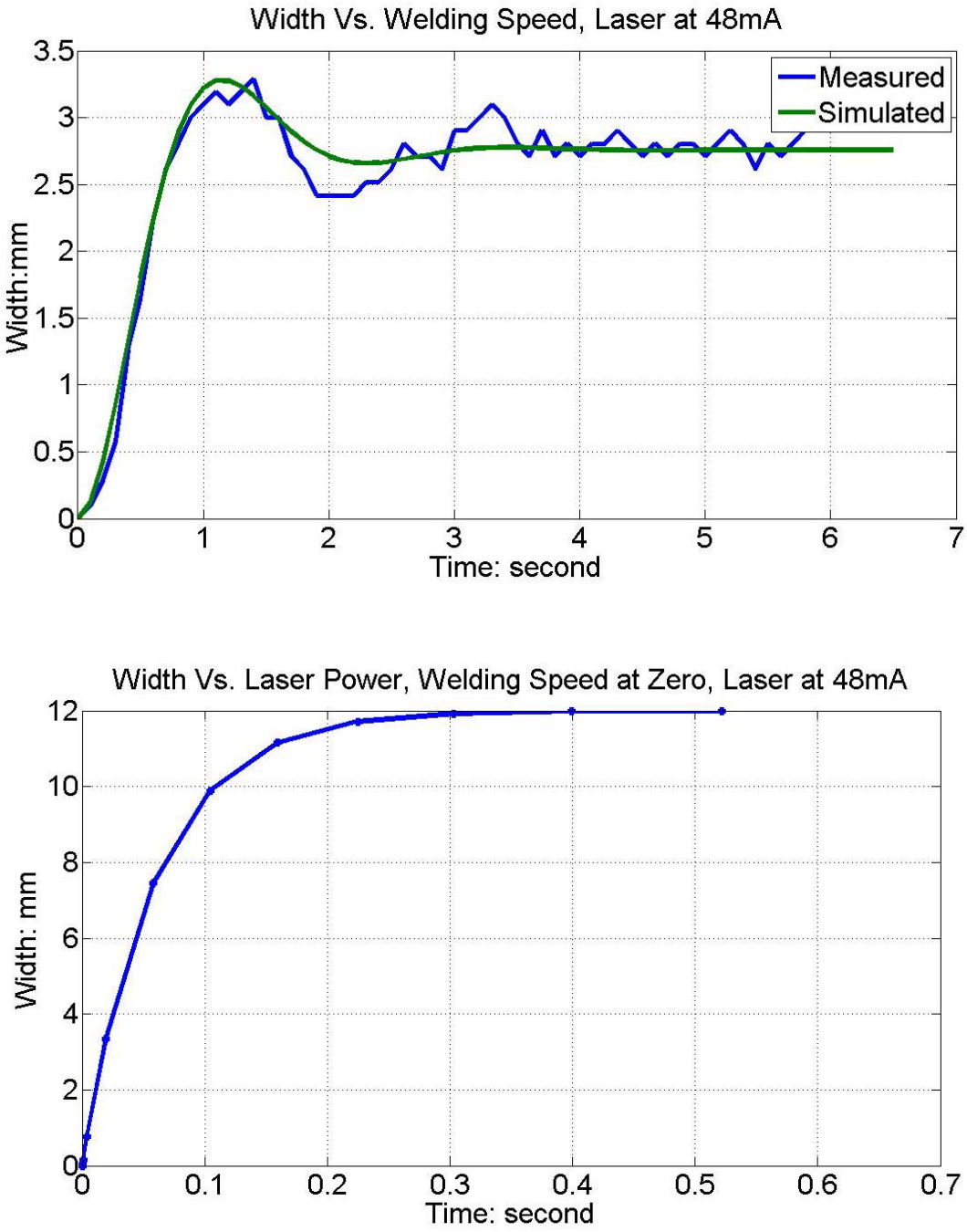


Figure 8.15 Experimental results

The SISO above was founded under the step inputs of welding speed and laser power. So, if we take the laser power as the input and the weld pool width as the output, the model structure will be exactly same except the gain on the right side. Further research also

showed that when the welding speed is zero, the step response with laser power as input and the width as output exhibits a first order model linearly.

Based on the analysis above, we start with a third order linear dynamics

Suppose the third order Hammerstein nonlinear structure can be represented by:

$$c_1\dddot{y}(t)+c_2\ddot{y}(t)+c_3\dot{y}(t)+y(t)=f(u_1(t),u_2(t))$$

(8.7.2)

The system structure is obtained as:

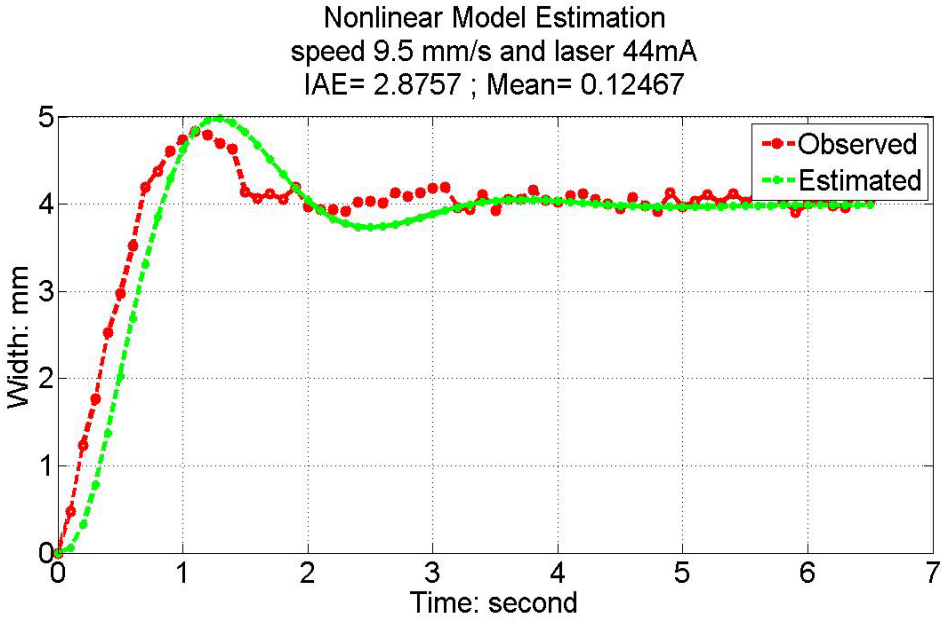
$$0.0071\dddot{y}(t)+0.142165\ddot{y}(t)+0.34\dot{y}(t)+y(t)=f(u_1(t),u_2(t))$$

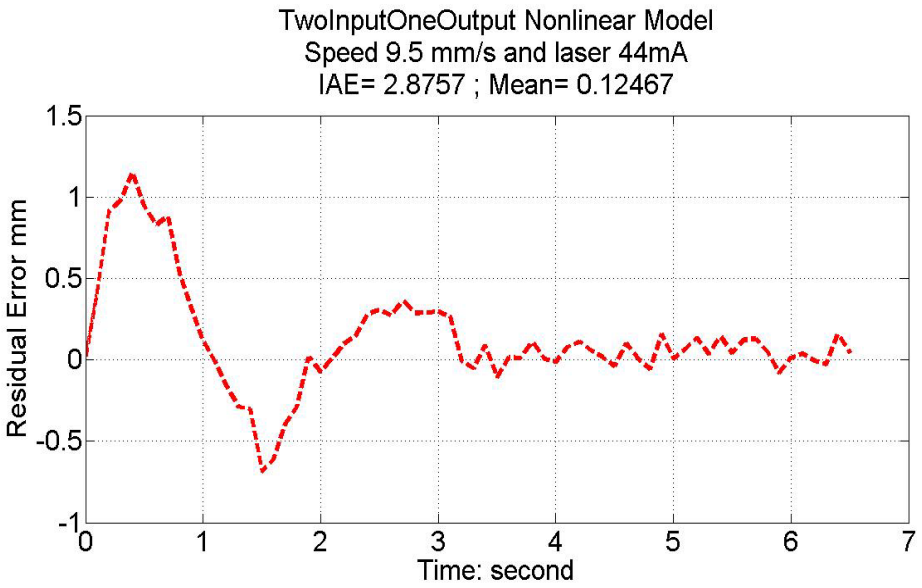
(8.7.3)

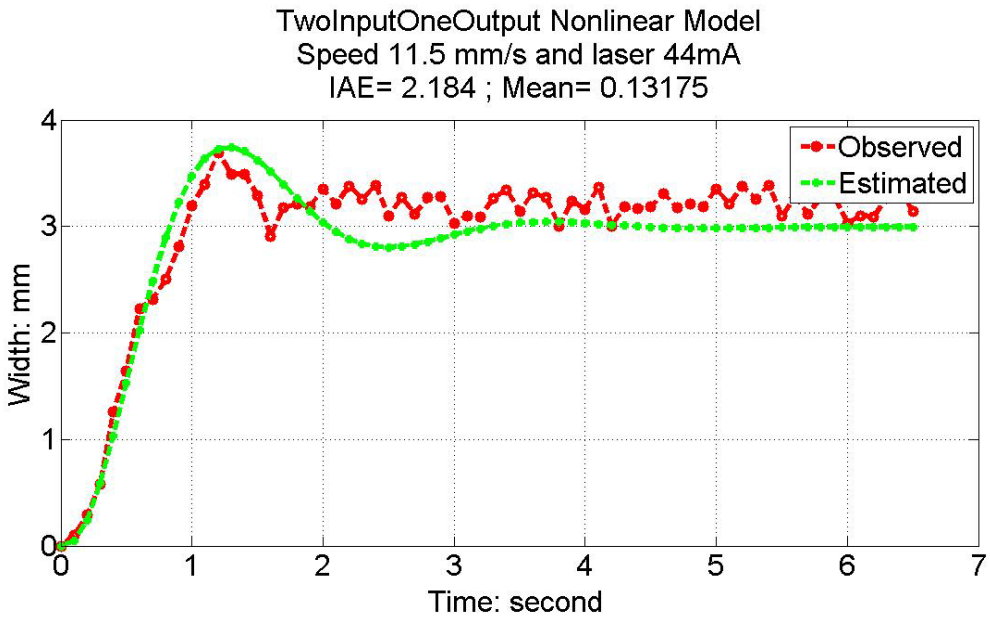
$$f(u_1(t),u_2(t))=-0.0086{u_1}^4(t)+0.1310{u_1}^3(t)-0.2520{u_1}^2(t)-2.0846u_1(t)$$
$$-0.2514u_2(t)u_1(t)+0.0153u_2(t)u_1(t)^2+1.1719u_2(t)-2.4705$$

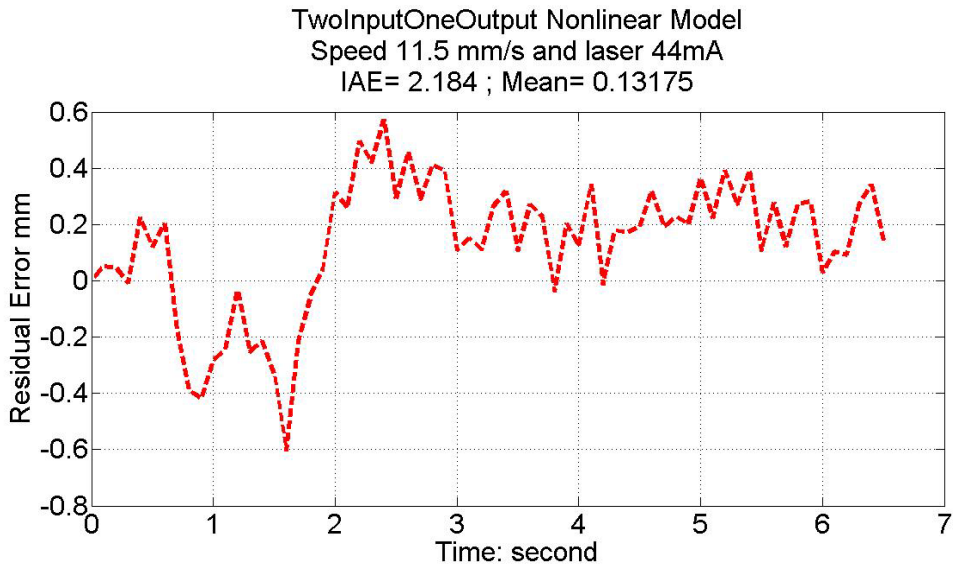
(8.7.4)

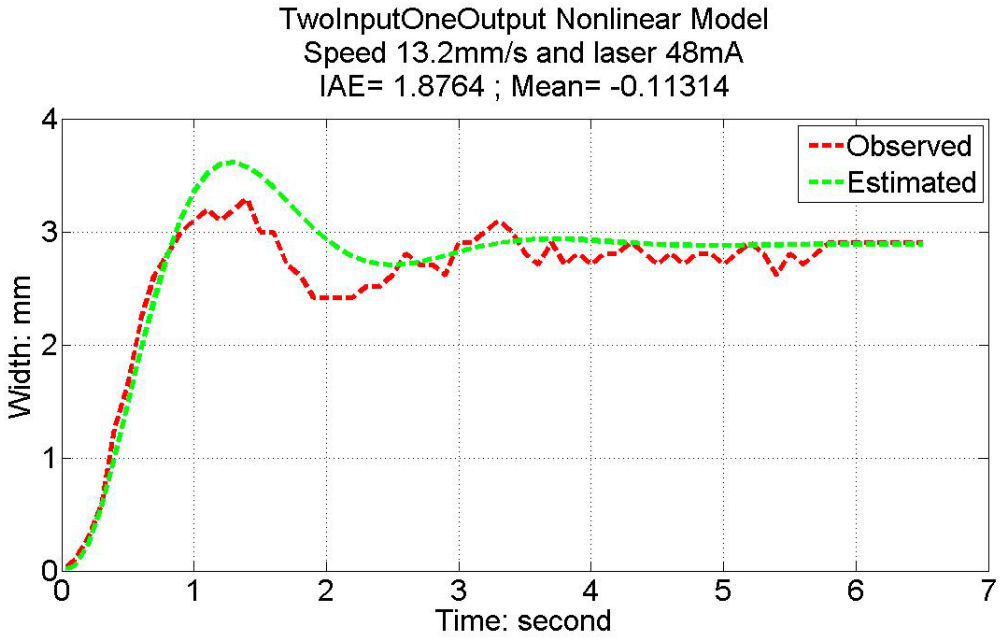
The model is validated with similar inputs as those used in SISO case above. The results are shown in Figure 8.16

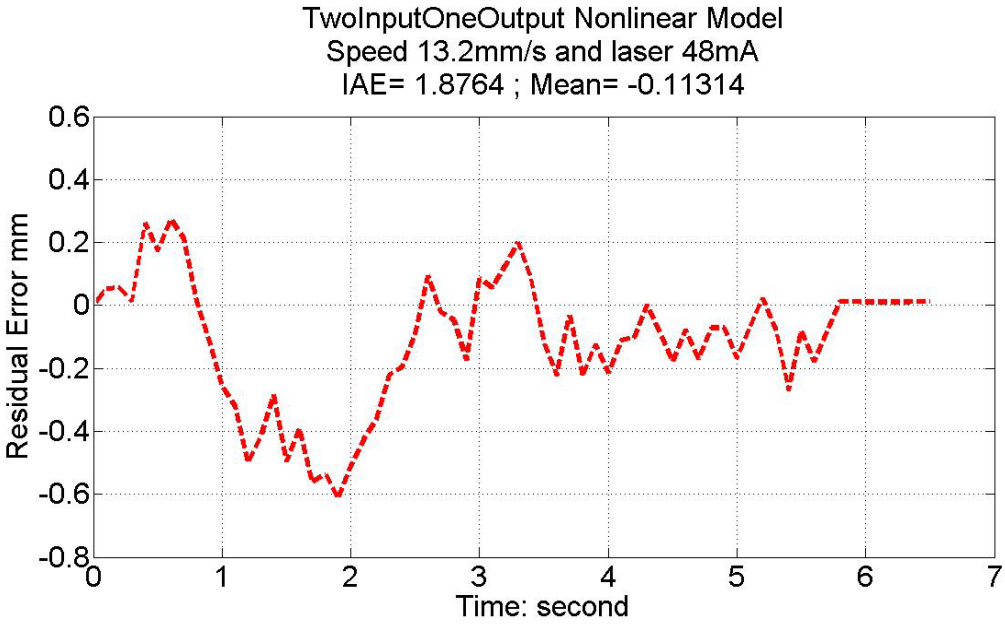


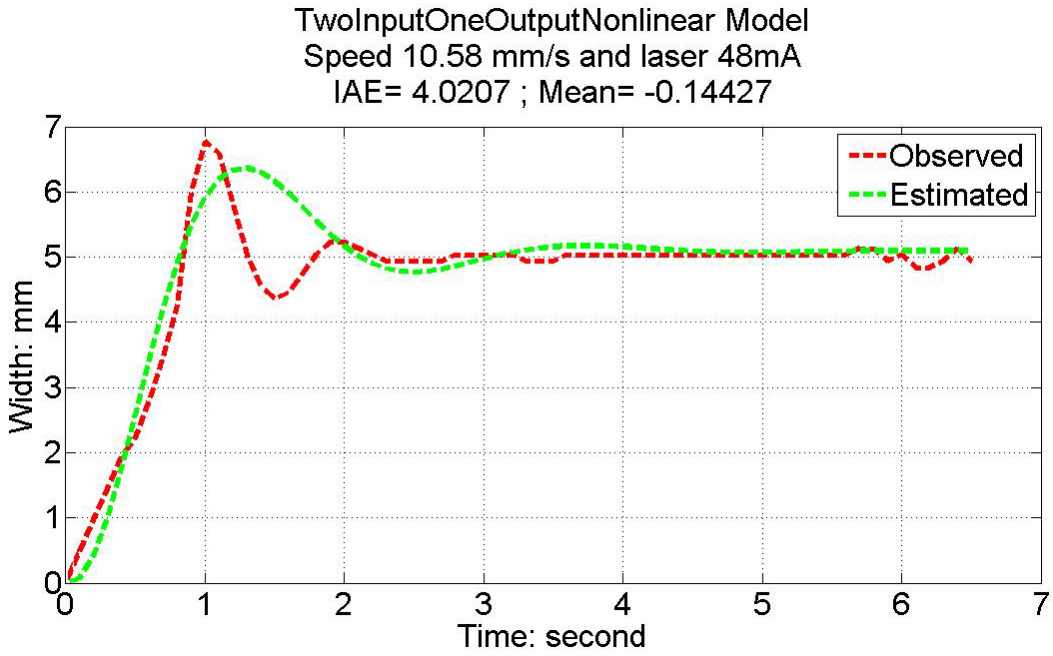

TwoInputOneOutput Nonlinear Model
Speed 9.5 mm/s and laser 44mA
IAE= 2.8757 ; Mean= 0.12467
Residual Error mm
Time: second

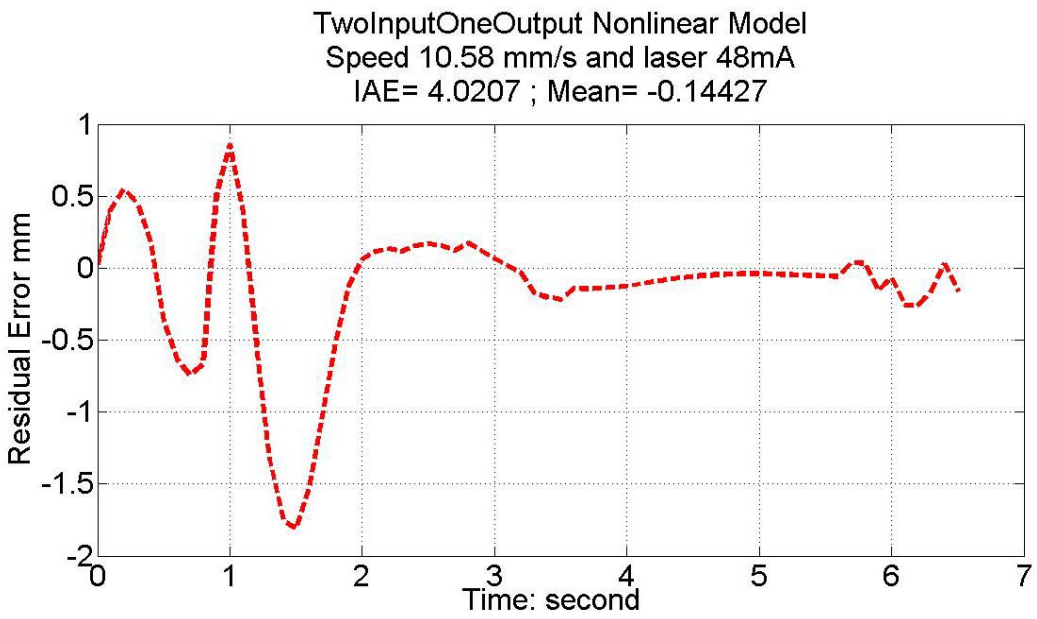

TwoInputOneOutput Nonlinear Model
Speed 11.5 mm/s and laser 44mA
IAE= 2.184 ; Mean= 0.13175
Observed
Estimated
Width: mm
Time: second

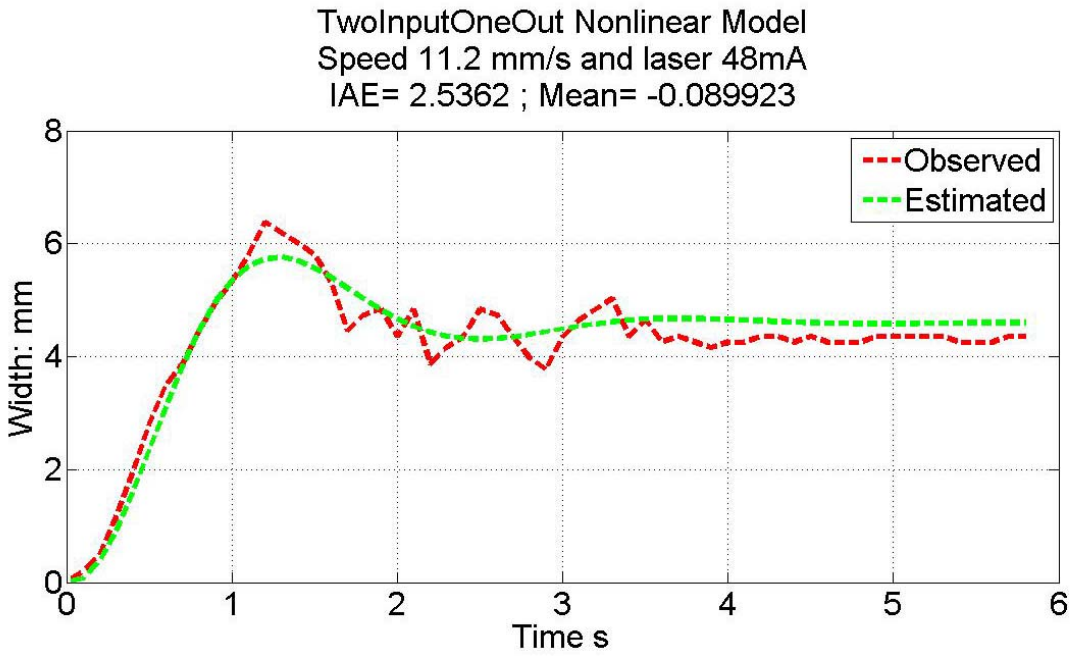

TwoInputOneOutput Nonlinear Model
Speed 11.5 mm/s and laser 44mA
IAE= 2.184 ; Mean= 0.13175
Residual Error mm
Time: second

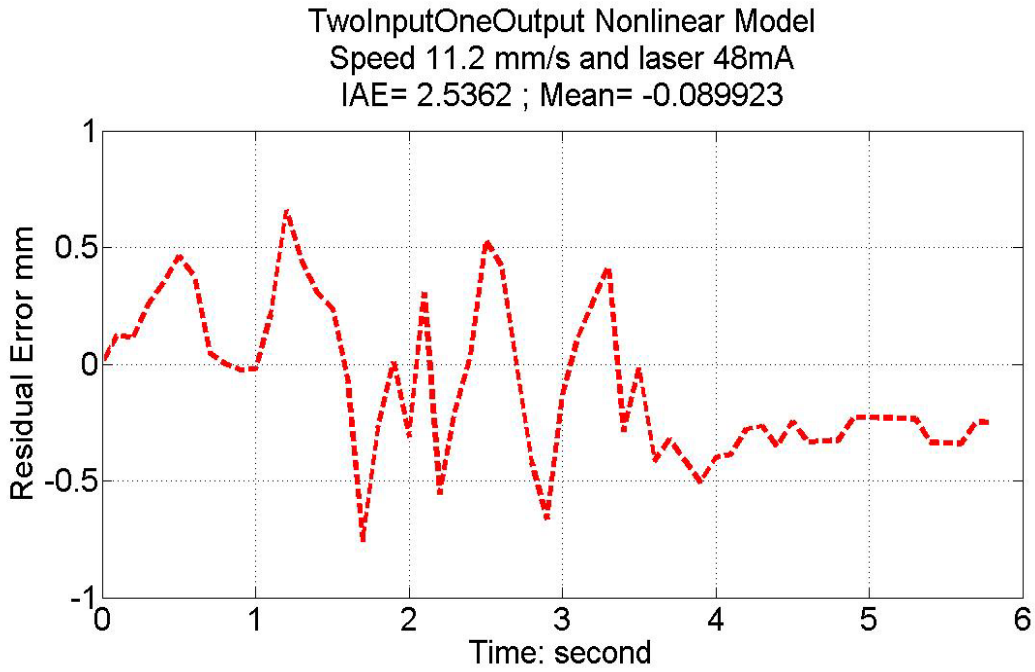

TwoInputOneOutput Nonlinear Model
Speed 13.2mm/s and laser 48mA
IAE= 1.8764 ; Mean= -0.11314
Observed
Estimated
Width: mm
Time: second

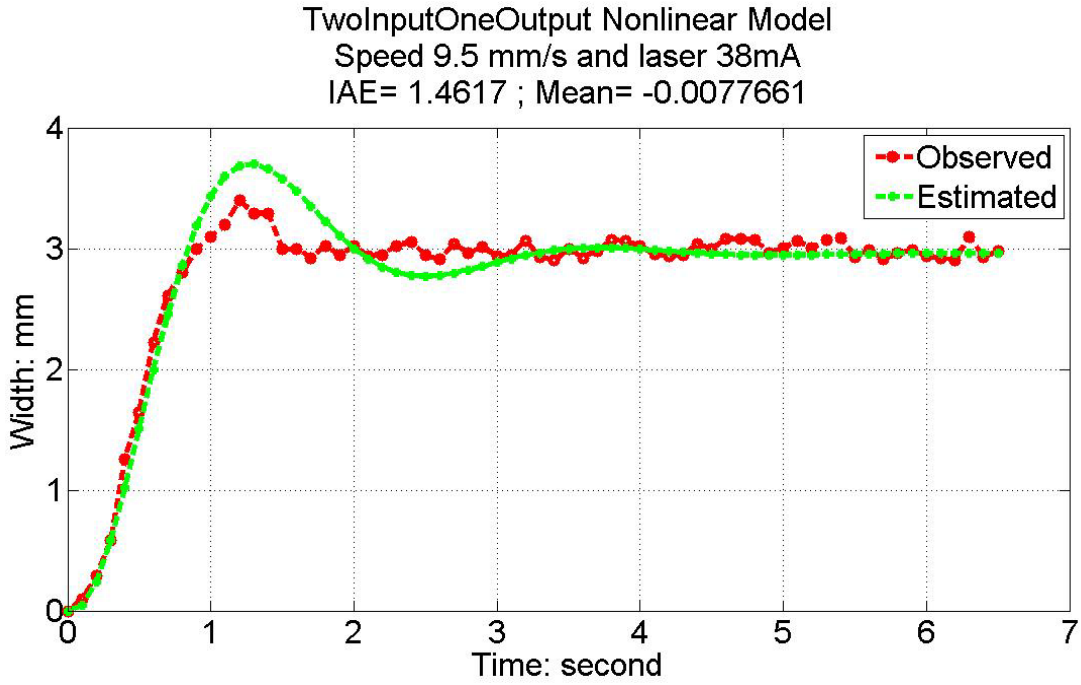

TwoInputOneOutput Nonlinear Model
Speed 13.2mm/s and laser 48mA
IAE= 1.8764 ; Mean= -0.11314
Residual Error mm
Time: second

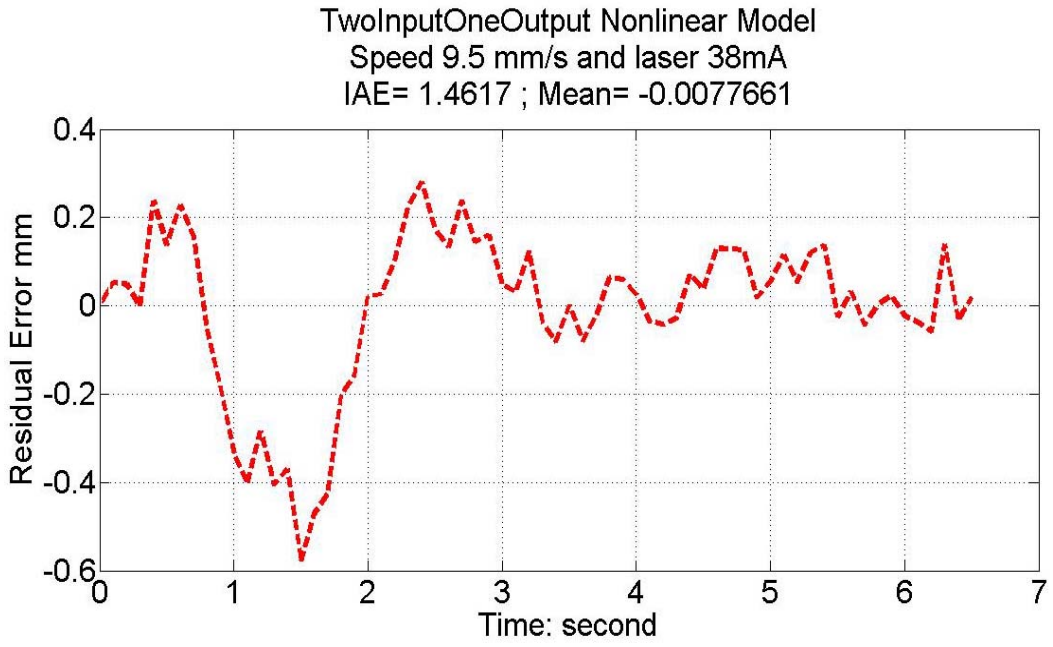

TwoInputOneOutputNonlinear Model
Speed 10.58 mm/s and laser 48mA
IAE= 4.0207 ; Mean= -0.14427
Observed
Estimated
Width: mm
Time: second

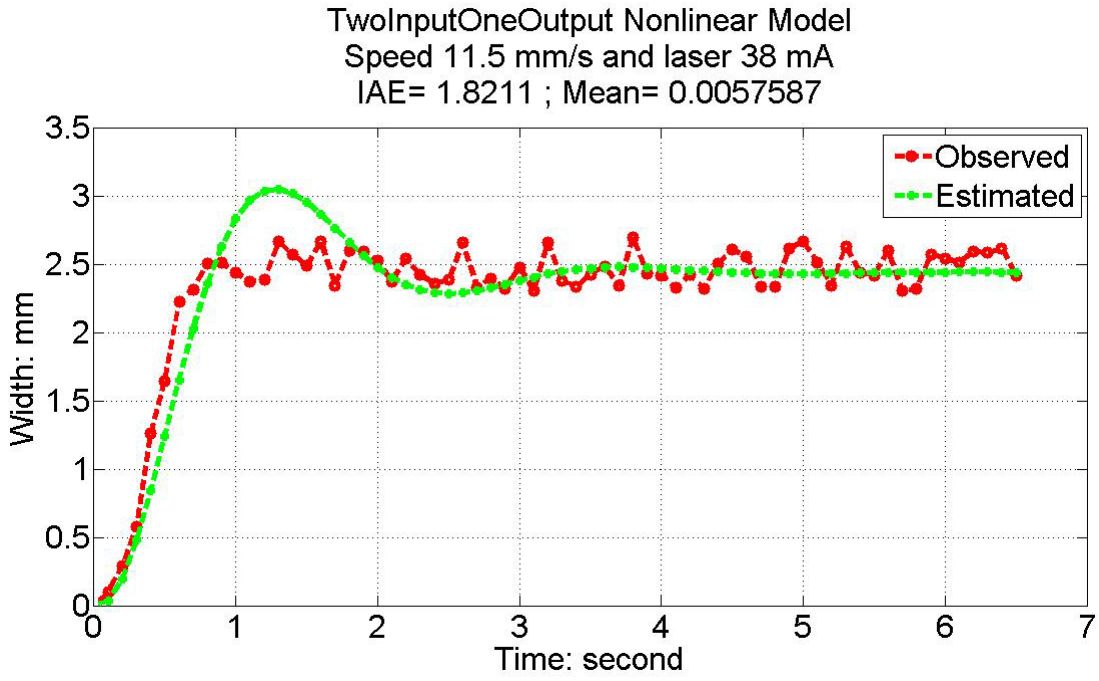

TwoInputOneOutput Nonlinear Model
Speed 10.58 mm/s and laser 48mA
IAE= 4.0207 ; Mean= -0.14427
Residual Error mm
Time: second

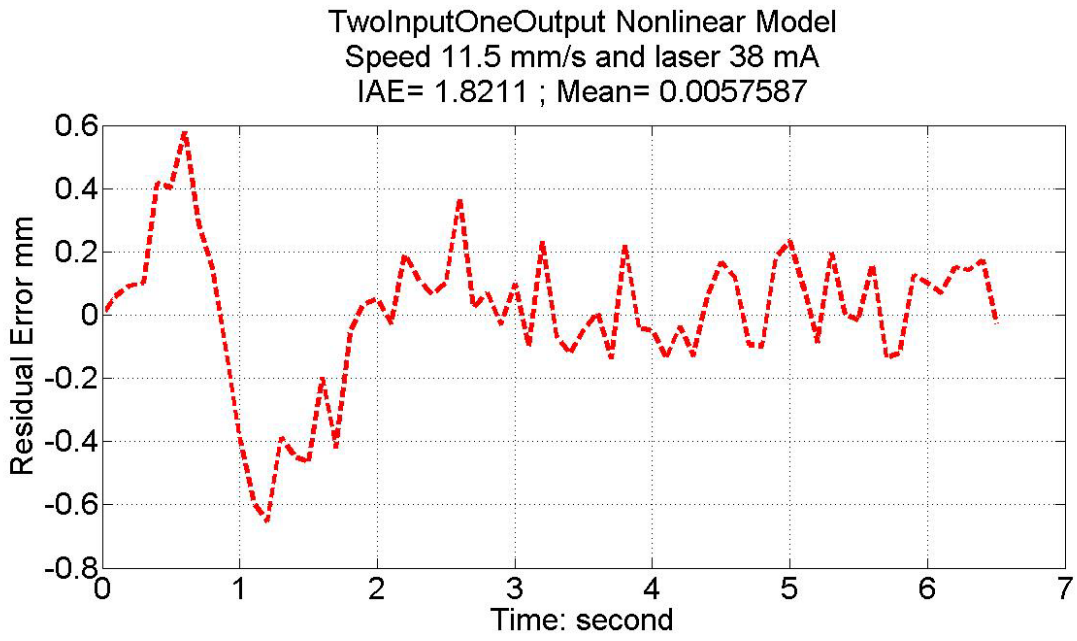

TwoInputOneOut Nonlinear Model
Speed 11.2 mm/s and laser 48mA
IAE= 2.5362 ; Mean= -0.089923
Observed
Estimated
Width: mm
Time s

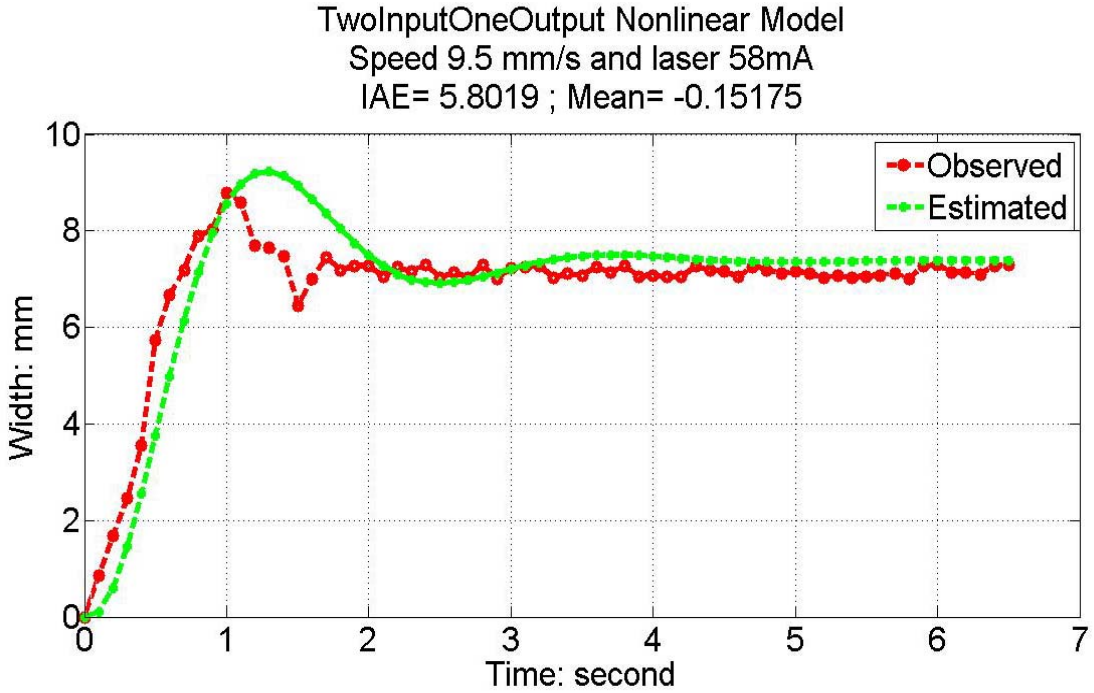

TwoInputOneOutput Nonlinear Model
Speed 11.2 mm/s and laser 48mA
IAE= 2.5362 ; Mean= -0.089923
Residual Error mm
Time: second
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6

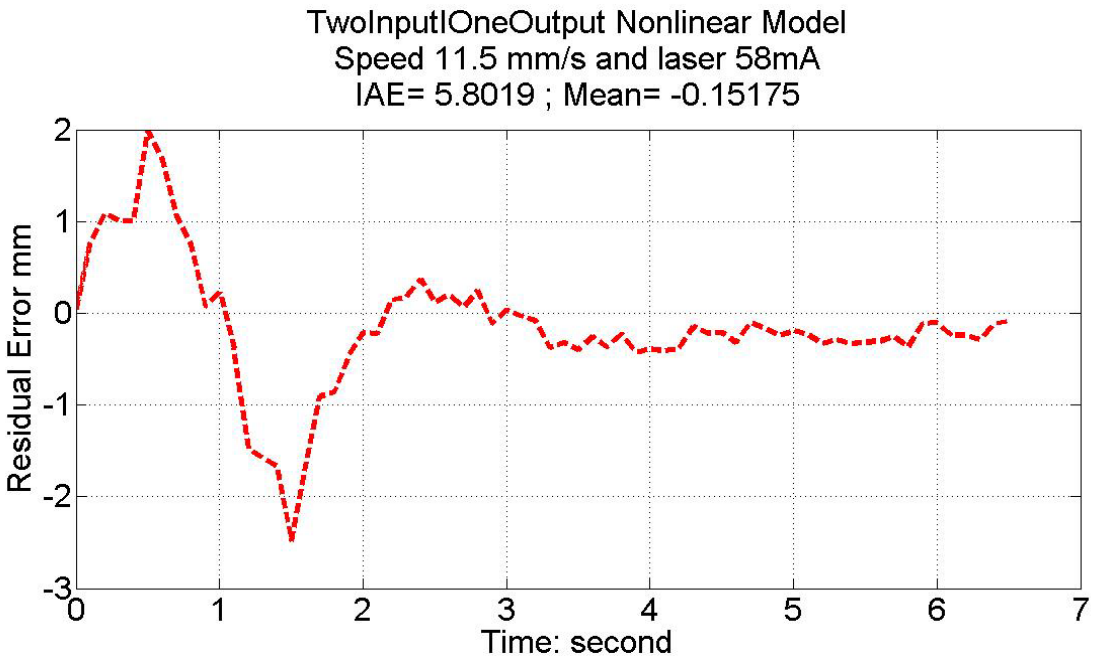

TwoInputOneOutput Nonlinear Model
Speed 9.5 mm/s and laser 38mA
IAE= 1.4617 ; Mean= -0.0077661
Width: mm
Time: second
Observed
Estimated
4
3
2
1
0
0
1
2
3
4
5
6
7

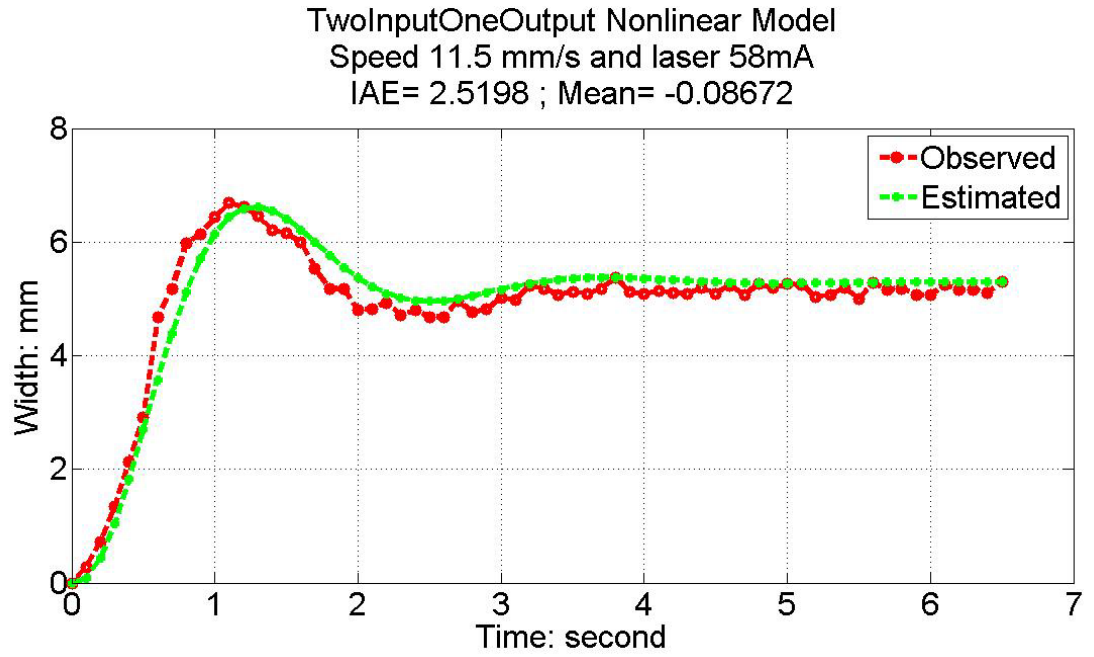

TwoInputOneOutput Nonlinear Model
Speed 9.5 mm/s and laser 38mA
IAE= 1.4617 ; Mean= -0.0077661
Residual Error mm
Time: second

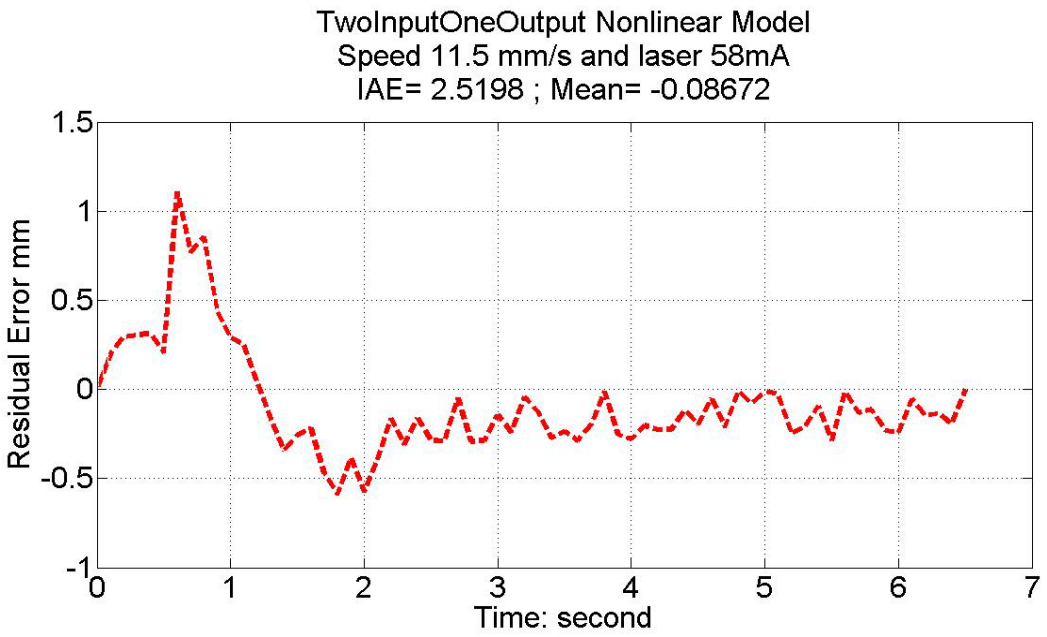

TwoInputOneOutput Nonlinear Model
Speed 11.5 mm/s and laser 38 mA
IAE= 1.8211 ; Mean= 0.0057587
Observed
Estimated
Width: mm
Time: second

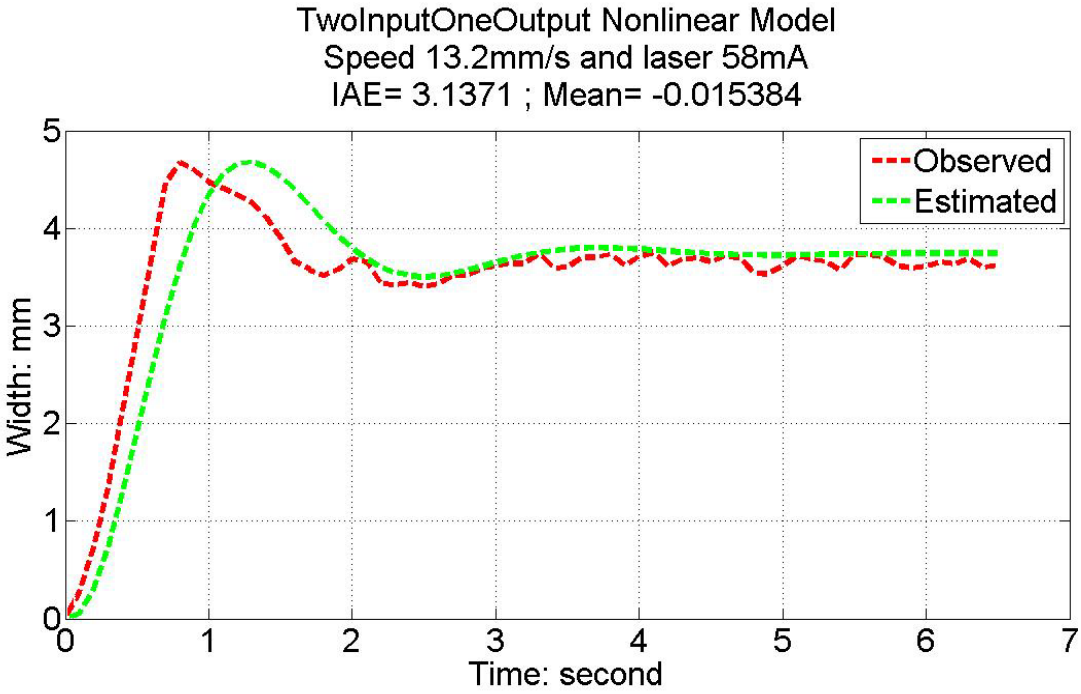

TwoInputOneOutput Nonlinear Model
Speed 11.5 mm/s and laser 38 mA
IAE= 1.8211 ; Mean= 0.0057587
Residual Error mm
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
Time: second


TwoInputOneOutput Nonlinear Model
Speed 9.5 mm/s and laser 58mA
IAE= 5.8019 ; Mean= -0.15175
Width: mm
10
8
6
4
2
0
0
1
2
3
4
5
6
7
Time: second
Observed
Estimated


TwoInputIOneOutput Nonlinear Model
Speed 11.5 mm/s and laser 58mA
IAE= 5.8019 ; Mean= -0.15175
Residual Error mm
Time: second


TwoInputOneOutput Nonlinear Model
Speed 11.5 mm/s and laser 58mA
IAE= 2.5198 ; Mean= -0.08672
Observed
Estimated
Width: mm
Time: second


TwoInputOneOutput Nonlinear Model
Speed 11.5 mm/s and laser 58mA
IAE= 2.5198 ; Mean= -0.08672
Residual Error mm
Time: second


TwoInputOneOutput Nonlinear Model
Speed 13.2mm/s and laser 58mA
IAE= 3.1371 ; Mean= -0.015384
Observed
Estimated
Width: mm
Time: second

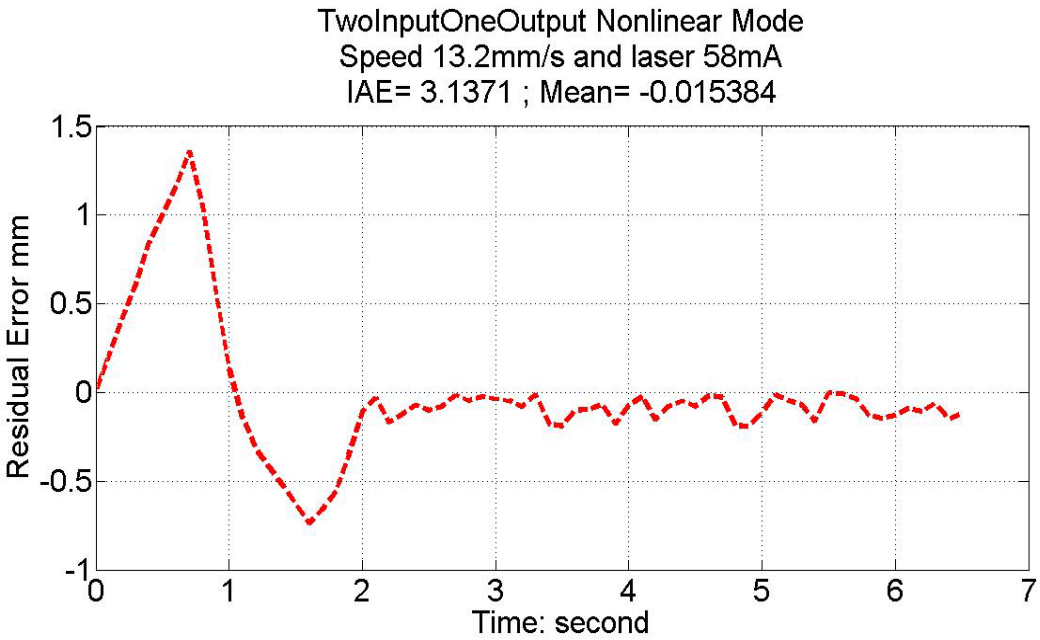


Figure 8.16 2ISO model validation

CHAPTER 9

NONLINEAR CONTROL DESIGN

9.1 Known Model system: Linearization Design

Consider a standard nonlinear Hammerstein model represented by state format:

$$\begin{aligned} y(s) &= G(s)z(s) \\ z(s) &= L(f(u(t))) \\ z(t) &= f(u(t)) \end{aligned} \tag{9.1.1}$$

, where y, u are the output and input respectively, and z is the intermediate input feeding into the linear dynamics $G(s)$; L is the Laplace transform operator

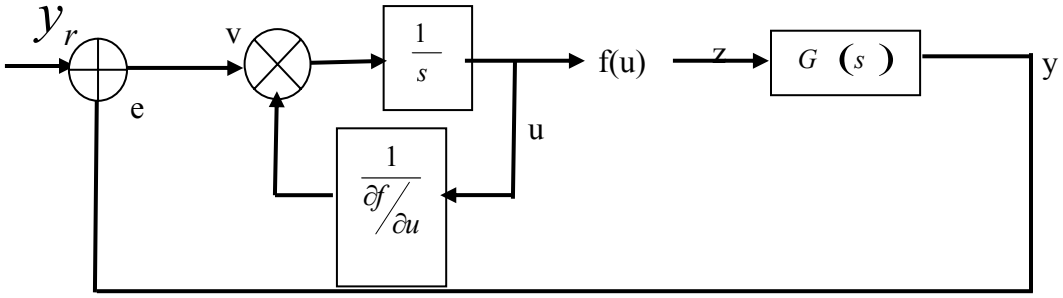


Figure 9.1.a Hammerstein control design based on linearization

Because the goal is to cancel the effect of nonlinearity in the equation 9.1.1 so that the system can be treated as linear system, therefore we take the first derivative of z at first. Then we have:

$$\dot{z} = \frac{\partial f(u)}{\partial u}\frac{du}{dt} \tag{9.1.2}$$

Define the virtual control state v, i.e.

$$v = \dot{z}\,. \tag{9.1.3}$$

Or in Laplace transform,

$$v(s) = \frac{z(s)}{s} \tag{9.1.4}$$

With simple substitution, equation (9.2) can be written by:

$$v = \frac{\partial f(u)}{\partial u}\dot{u} \tag{9.1.5}$$

Or in Laplace transform,

$$v(s) = L\left(\frac{\partial f(u)}{\partial u}\right)\frac{u(s)}{s} \tag{9.1.6}$$

Substituting equation (9.3) into (9.1), we have,

$$y(s) = \frac{1}{s}v(s)G(s)\,. \tag{9.1.7}$$

If we assume the structure of the nonlinearity is known, then the system can be treated as linear system as shown in Figure 9.1

Figure 9.1 is the result of simulation tracking step signal following the linearization and control design from equation 9.1-4.

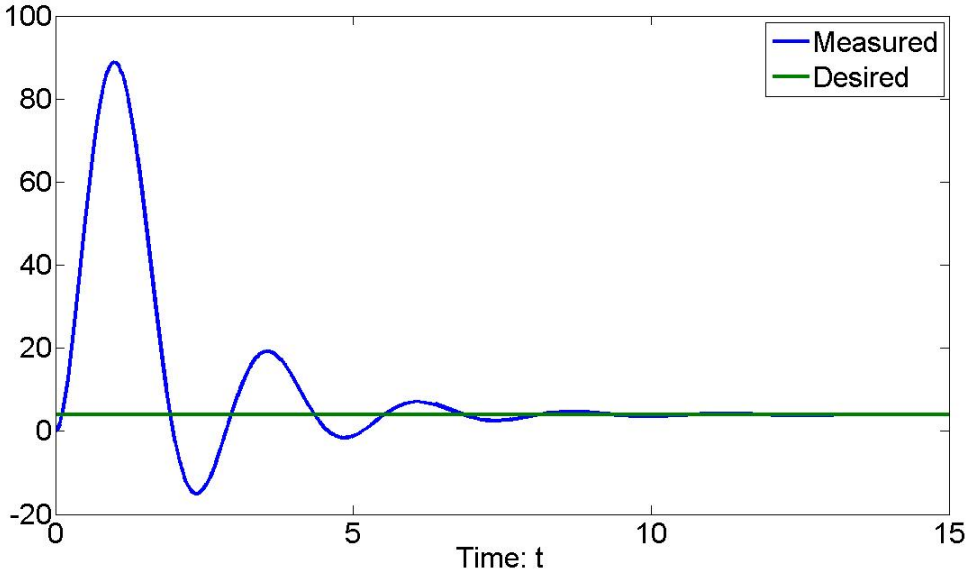


Figure 9.1.b Hammerstein control design based on linearization

9.2 Known Model System: Constructive Design ($\dot{x} = f(x,u)$)

Assumption 1:For $x(t_0) = x_0 \in U \subset R^n$ there exists $\varepsilon > 0$ and $T(\varepsilon, x_0)$ such that

$\|x(t)\| < \varepsilon$ for $t \geq t_0 + T$

Assumption 2: a function $\alpha(\bullet)$ is class K if it is continuous and strictly increases

Assumption 3: system is input-to-state partially stable if there exists functions such that

$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \forall x \in R^n$

$\frac{\partial V(x)}{\partial x} f(x,u) \leq -\alpha_3(\|x\|) + \alpha_4(\|u\|) + d$

$x(t) \leq \beta(\|x(0)\|, t) + \gamma(\|u\|_\infty) + d$

$d > 0$

Before we design the control law for nonlinear Hammerstein model, consider a nonlinear system described by:

$$y^n(t) = f\left(y, \dot{y}, y^{n-1}\right) + u \tag{9.2.1}$$

, we can recursively construct the control design with defining Lyapunov candidates. For example, assume the model as

$$\ddot{y}(t) = f(y, \dot{y}) + u$$
$$f = \frac{0.3015}{0.1286}\dot{y} + \frac{1}{0.1286}y \tag{9.2.2}$$

Let:

$$x_1 = y$$
$$x_2 = \dot{y} \tag{9.2.3}$$

Then we have:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = f + u$$
$$f = 2.34x_2 + 7.77x_1 \tag{9.2.4}$$

Let

$e_i = x_i - x_{id}$, where i=1,2

And $\dot{e}_i = \dot{x}_i - \dot{x}_{id}$ (9.2.5)

At first, define the Lyapunov candidate function:

$$V_1 = \frac{1}{2}e_1^{\,2}. \tag{9.2.6}$$

Then the derivative of V is:

$$\dot{V}_1 = e_1\dot{e}_1$$
$$= e_1(\dot{x}_1 - \dot{x}_{1d})$$
$$= e_1(x_2 - \dot{x}_{1d}) \tag{9.2.7}$$

Because the goal is to design the control in order that the tracking error converge to zero, we can consider x_2 as the virtual control to the system and when $x_2 \Rightarrow x_{2d}$.

Accordingly, define

$x_{2d} = -k_1e_1 + \dot{x}_{1d}$, where k_1 is a positive constant (9.2.8)

Then,

$$\dot{V}_1 \Rightarrow -k_1e_1^{\,2} \le 0 \tag{9.2.9}$$

Define

$$V_2 = V_1 + \frac{1}{2} e_2^{\,2} \tag{9.2.10}$$

Take the derivative of V_2, we have

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + e_2 \dot{e}_2 \\ &= -k_1 e_1^{\,2} + e_2 \left(f + u - \dot{x}_{2d} \right) \end{aligned} \tag{9.2.11}$$

Design the control u as,

$u = -f + \dot{x}_{nd} - k_2 e_2$, where k_2 is a positive constant (9.2.12)

Then,

$$\dot{V}_2 = -k_1 e_1^{\,2} - k_2 e_2^{\,2} \le 0 \tag{9.2.13}$$

Accordingly, for the system, the designed virtual control can guarantee that the first derivative of defined Lyapunov candidate negative semidefinite as shown in equation (3). Namely, the system is asymptotically stable and bounded.

9.3 Proof:

Consider the system:

$$y^n(t) = f\left(y, y, y^{n-1}\right) + u \tag{9.3.1}$$

, where y, u are the output and the control respectively; $y^i, i = 1,2,\cdots n$ is the ith derivative of y

Let

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{y} \\ &\cdots\cdots \\ x_{n-1} &= y^{n-2} \\ x_n &= y^{n-1} \end{aligned} \tag{9.3.2}$$

Then we have:

$$\begin{aligned}
&\dot{x}_1 = x_2 \\
&\dot{x}_2 = x_3 \\
&\cdots\cdots \\
&\dot{x}_i = x_{i+1} \\
&\cdots \\
&\dot{x}_{n-1} = x_n \\
&\dot{x}_n = f\left(x_1, x_2, \cdots, x_n\right) + u
\end{aligned} \tag{9.3.3}$$

Or,

$$\begin{aligned}
&\dot{x}_i = x_{i+1} \\
&\dot{x}_n = f\left(x\right) + u \\
&1 \le i \le n-1
\end{aligned} \tag{9.3.4}$$

, where $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$ (9.3.5)

Assume the desired signal is x_d and x_d is nth derivable. Then the tracking error can be written by:

$$e = x - x_d \tag{9.3.6}$$

Or,

$$
\begin{aligned}
e_1 &= x_1 - x_{1d} \\
e_2 &= x_2 - x_{2d} \\
&\cdots\cdots \\
e_i &= x_i - x_{id} \\
&\cdots \\
e_{n-1} &= x_{n-1} - x_{(n-1)d} \\
e_n &= x_n - x_{nd}
\end{aligned}
\tag{9.3.7}
$$

Or,

$$
\begin{aligned}
\dot{e}_1 &= \dot{x}_1 - \dot{x}_{1d} \\
\dot{e}_2 &= \dot{x}_2 - \dot{x}_{2d} \\
&\cdots\cdots \\
\dot{e}_i &= \dot{x}_i - \dot{x}_{id} \\
&\cdots \\
\dot{e}_{n-1} &= \dot{x}_{n-1} - \dot{x}_{(n-1)d} \\
e_n &= \dot{x}_n - \dot{x}_{nd}
\end{aligned}
\tag{9.3.8}
$$

With simple substitution, we have:

$$
\begin{aligned}
\dot{e}_1 &= x_2 - \dot{x}_{1d} \\
\dot{e}_2 &= x_3 - \dot{x}_{2d} \\
&\cdots\cdots \\
\dot{e}_i &= x_{i+1} - \dot{x}_{id} \\
&\cdots \\
\dot{e}_{n-1} &= x_n - \dot{x}_{(n-1)d} \\
e_n &= f + u - \dot{x}_{nd}
\end{aligned}
\tag{9.3.9}
$$

It is easy to see that the system is strict feedback system. The design procedure is to define the control law with constructing Lyapunov candidate for the system. For every subsystem, we can try to define a virtual control.

At first, define the Lyapunov candidate as:

$$V_1 = \frac{1}{2} e_1^{\,2} \tag{9.2.10}$$

Then x_2 can be considered as a virtual control for the first equation and the task is to seek the control law so that $x_2 = x_{2d}\left(x_1, x_{1d}\right)$ and the derivative of V is semidefinite or negative.

The derivative is represented by

$$\dot{V}_1 = e_1 \dot{e}_1 \tag{9.3.11}$$

Or,

$$\dot{V}_1 = e_1\left(x_2 - \dot{x}_{1d}\right) \tag{9.3.12}$$

Or,

$$\dot{V}_1 = e_1\left(e_2 + x_{2d} - \dot{x}_{1d}\right) \tag{9.3.13}$$

If we can define

$$x_{2d} = -k_1 e_1 + \dot{x}_{1d} \tag{9.3.14}$$

Then, we have,

$$\begin{aligned} \dot{V}_1 &= -k_1 e_1^{\,2} + e_1 e_2 \\ &\le -k e_1^{\,2} + M_1 \end{aligned} \quad \text{, where } \left|e_1 e\right|_2 = M_1 \tag{9.3.15}$$

Define

$$V_2 = V_1 + \frac{1}{2} e_2^{\,2} \tag{9.3.16}$$

By taking the derivative, we have,

$$\dot{V}_2 = \dot{V}_1 + e_2 \dot{e}_2 \tag{9.3.17}$$

Or,

$$\begin{aligned}\dot{V}_2 &= -k_1 e_1^{\,2} + e_1 e_2 + e_2\left(x_3 - \dot{x}_{2d}\right) \\ &= -k_1 e_1^{\,2} + e_2\left(e_1 + \left(x_3 - \dot{x}_{2d}\right)\right) \\ &= -k_1 e_1^{\,2} + e_2\left(e_1 + e_3 + x_{3d} - \dot{x}_{2d}\right)\end{aligned} \tag{9.3.18}$$

If we choose

$$x_{3d} = -k_2 e_2 + \dot{x}_{2d} \tag{9.3.19}$$

Then,

$$\begin{aligned}\dot{V}_2 &= -k_1 e_1^{\,2} - k_2 e_2^{\,2} + e_2 e_1 + e_2 e_3 \\ &\le -k_1 e_1^{\,2} - k_2 e_2^{\,2} + M_2\end{aligned} \quad \text{, where } M_2 = \left|e_1 e_2\right| + M_1 \tag{9.3.20}$$

If we continue similar procedure until

$$V_n = V_{n-1} + \frac{1}{2} e_n^{\,2} \tag{9.3.21}$$

$$\begin{aligned}\dot{V}_n &= \dot{V}_{n-1} + e_n \dot{e}_n \\ &= -k_1 e_1^{\,2} - k_2 e_2^{\,2} - \cdots - k_{n-1} e_{n-1}^{\,2} + e_n\left(e_{n-1} + f + u - \dot{x}_{nd}\right)\end{aligned} \tag{9.3.22}$$

Choose the control:

$$u = -k_n e_n - f + \dot{x}_{nd} \tag{9.3.23}$$

Then,

$$\begin{aligned}\dot{V}_n &= -k_1 e_1^{\,2} - k_2 e_2^{\,2} - \cdots - k_{n-1} e_{n-1}^{\,2} - k_n e_n^{\,2} + e_n e_{n-1} \\ &\le -\sum_{i=1}^{n} k_i e_i^{\,2} + M_n\end{aligned} \tag{9.3.24}$$

$$M_n = \left| e_{n-1} e_n \right| + M_{n-1}$$

(9.3.25)

Accordingly, $V(e)$ is bounded under the conditions of equations (9.2.5-7) and decreases monotonically, which means that the system is asymptotically stable.

To verify the control design, we simply run a simulation based on the identified Hammerstein model represented by:

$$\begin{aligned} \ddot{y} &= f(y, \dot{y}) + u \\ f &= 2\dot{y}(t) + y(t) \end{aligned} \qquad (9.3.26)$$

The tracking signals are both sin(t) and step signal with amplitude of 1.

For simplicity, take k=1. The simulation tracking step and sin(2π t) signals are as shown in Figure 9.2.

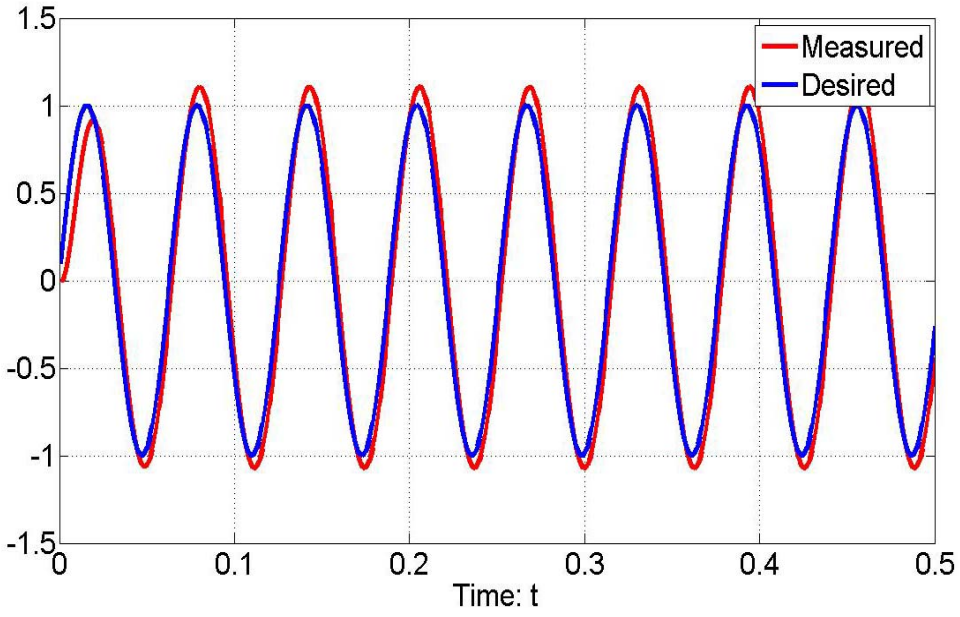


Figure 9.2.a Tracking sinusoidal signal

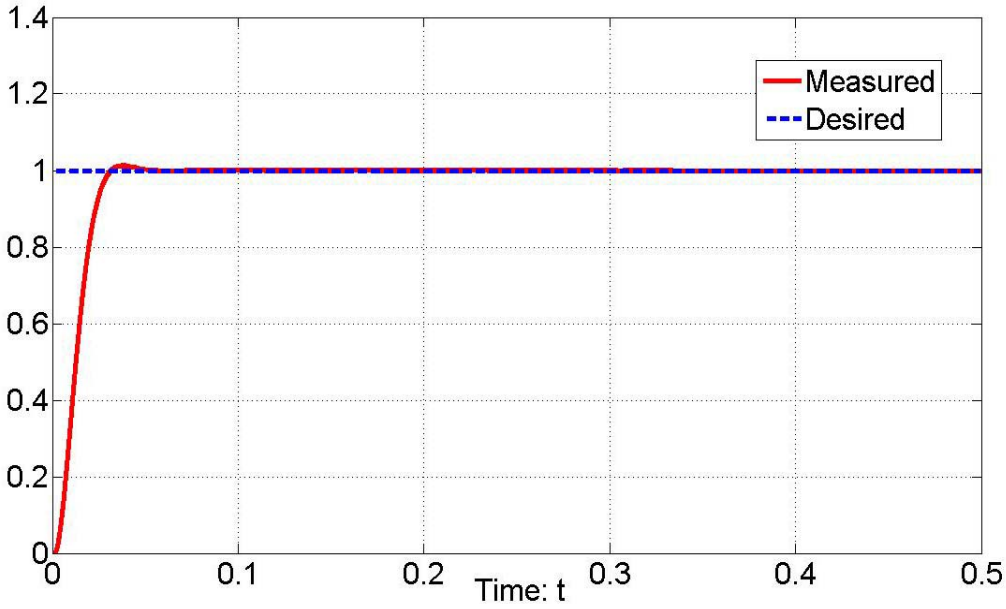


Figure 9.2.b Tracking step signal

For further study, consider a system represented by:

$$y^n(t) = \theta f\left(y, \dot{y}, y^{n-1}\right) + u$$

, where θ is unknown (9.4.1)

Then the procedure will have to consider the unknown parameter.

For simplicity, assume the system is second order and $\ddot{y}(t) = \theta f(y) + u$

(9.4.2)

$$z_1 = x_1$$
$$z_2 = x_2 - q(x, \theta)$$
$$q(x_1, \theta) = -cx_1 - \theta f(x_1)$$

(9.4.3)

Take the derivatives, then

$$\dot{z}_1 = \dot{x}_1$$
$$\dot{z}_2 = \dot{x}_2$$

(9.4.4)

Or,

$$\dot{z}_1 = -cz_1 + z_2 - \widehat{\theta}$$
$$\dot{z}_2 = u + \widehat{\theta} f \frac{\partial q}{\partial x_1} - \frac{\partial q}{\partial x_1}\left(z_2 - cz_1\right) + \dot{\theta} f \quad (9.4.5)$$

, where θ is the estimation of the parameter

Define Lyapunov function:

$$V = \frac{1}{2}\left(z_1^{\,2} + z_2^{\,2} + \widehat{\theta}^{\,2}\right)$$

(9.4.6)

The first derivative of V can be written by:

$$\dot{V} = z_1\dot{z}_1 + z_2\dot{z}_2 + \hat{\theta}\dot{\hat{\theta}}$$

(9.4.7)

Or,

$$\begin{aligned}
\dot{V} &= -cz_1^2 - \hat{\theta}\left(fz_1 - f\frac{\partial q}{\partial x_1}z_2 \right) \\
&+ z_2\left(u + z_1 - \frac{\partial q}{\partial x_1}(z_2 - cz_1) + \dot{\theta}f \right) + \hat{\theta}\dot{\hat{\theta}} \\
&= -cz_1^2 - \hat{\theta}\left(fz_1 - f\frac{\partial q}{\partial x_1}z_2 + \dot{\hat{\theta}} \right) \\
&+ z_2\left(u + z_1 - \frac{\partial q}{\partial x_1}(z_2 - cz_1) + \dot{\theta}f \right)
\end{aligned}$$

(9.4.8)

If we choose:

$$\begin{aligned}
u &= -kz_2 - z_1 + \frac{\partial q}{\partial x_1}(z_2 - cz_1) - \dot{\theta}f \\
\dot{\hat{\theta}} &= fz_1 - f\frac{\partial q}{\partial x_1}z_2
\end{aligned}$$

(9.4.9)

Then we have,

$$\dot{V} = -cz_1^2 - kz_2^2 \le 0$$

(9.4.10)

Accordingly, the first derivative of Lyapunov function is negative semidefinite. Namely, z_1, z_2, θ are bounded and accordingly, the system is asymptotically stable.

9.4 Hammerstein With $\dot{u}$ Introduced

Consider the Hammerstein system model represented by

$$y^n(t) = f\left(y, \dot{y}, y^{n-1}\right) + g(u) \tag{9.4.11}$$

Apparently, there is no zero points in the linear dynamics. However, under conditions we might be required to add $\dot{u}$ into the model. Then the system is represented by:

$$y^n(t) = f\left(y, \dot{y}, y^{n-1}\right) + \theta\dot{u} + g(u) \tag{9.4.12}$$

As verified in the previous chapter, typical welding processes do not have nonstable zero points in the system. In our current system, when we identify the Hammerstein model, we apply step or random-level step signals as the persistent test signal. With step signals as the input, it is impossible to estimate the extra parameter related to $\dot{u}$. However, we can still add it into the denominator functioning as a very small signal in the linear dynamics because the characteristics of the linear dynamics are determined by the poles in the denominator.

To test the feasibility, we change the parameter and compare the simulation results under sinusoidal signals and step signals. Example comparison results are shown in Table 9.1 and Figure 9.2.a-b under the condition that $\theta = 0.001$ and $\theta = 0.0.1$

Table 9.1 Model comparison between with $\dot{u}$ and without $\dot{u}$

θ	Error between outputs $\sum(y_u - y_{udot})$	
	Sin(t)	Step signal
0.1	86.0453	0
0.01	8.6045	0
0.005	4.3023	0
0.002	1.7269	0
0.0015	1.7209	0
0.001	0.8605	0

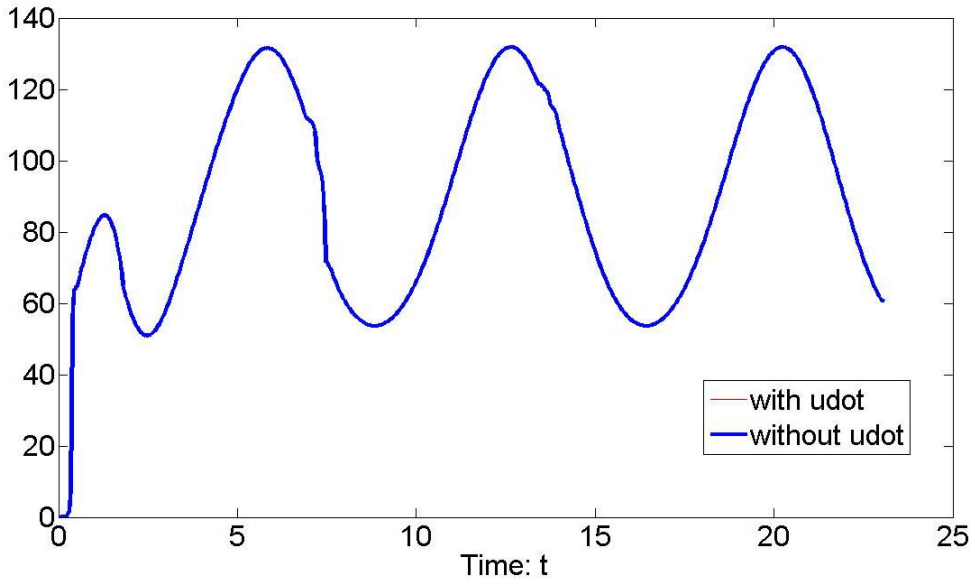


Figure 9.3.a With $\dot{u}$ and θ=0.001 in the model

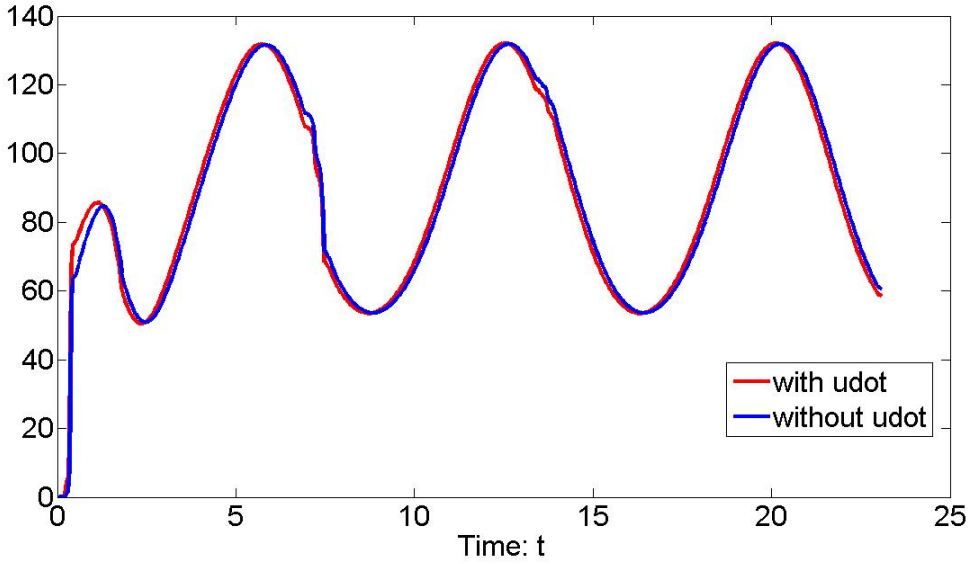


Figure 9.3.b With $\dot{u}$ and θ=0.01 in the model

Apparently, if we forcefully add $\dot{u}$ whose coefficient is under the condition of $\theta \leq 0.001$, then the system response will not be influenced too much under the inputs

of such as sinusoidal or step signal. We already verified in the previous chapter that the existence of $\dot{u}$ does not have any influence on the system performance.

In this case, we at first consider the nonlinear Hammerstein model represented by:

$$y^n(t) = f\left(y, \dot{y}, y^{n-1}\right) + \theta\dot{u} + g(u) \tag{9.4.13}$$

, where y, u are the output and the control respectively; $y^i, i = 1,2,\cdots n; \dot{u}$ is the ith derivative of y, u

Let

$$\begin{aligned}
&x_1 = y \\
&x_2 = \dot{y} \\
&\cdots\cdots \\
&x_{n-1} = y^{n-2} \\
&x_n = y^{n-1} \\
&z = u \\
&\dot{z} = v
\end{aligned} \tag{9.4.14}$$

, where v is the virtual control input for the system

Then we have:

$$\begin{aligned}
&\dot{x}_1 = x_2 \\
&\dot{x}_2 = x_3 \\
&\cdots\cdots \\
&\dot{x}_i = x_{i+1} \\
&\cdots \\
&\dot{x}_{n-1} = x_n \\
&\dot{x}_n = f\left(x_1, x_2, \cdots, x_n\right) + \theta v \\
&z = u \\
&\dot{z} = v
\end{aligned} \tag{9.4.15}$$

Or,

$$\begin{aligned} &\dot{x}_i = x_{i+1} \\ &\dot{x}_n = f(x) + u \\ &1 \le i \le n-1 \end{aligned} \tag{9.4.16}$$

, where $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$

During the identification, $g(u)$ is approximated by a polynomial. In fact, it can be other complex formats. However, we assume that $g(u)$ is continuous and bounded

We further assume all the states of the system are available for feedback and define:

$$\begin{aligned} &e_1 = x_1 - x_d \\ &e_2 = x_2 - \dot{x}_d \\ &\cdots\cdots \\ &e_i = x_i - x_d^{\,i-1} \\ &\cdots \\ &e_{n-1} = x_n - x_d^{\,n-2} \\ &e_n = x_n - x_d^{\,n-1} \end{aligned} \tag{9.4.17}$$

Then we have,

$$\begin{aligned} &\dot{e}_1 = x_2 - x_d \\ &\dot{e}_2 = x_3 - \ddot{x}_d \\ &\cdots\cdots \\ &\dot{e}_{n-1} = x_n - x^n{}_d \\ &\dot{e}_n = \dot{x}_n - x_d^{\,n} \end{aligned} \tag{9.4.18}$$

With simple substitution, we can have:

$$\dot{e} = Ae + b\left(f\left(e + \bar{x}_d, z\right) + \theta v - x_d^{\,n} + g\left(e + \bar{x}_d\right)\right)$$
$$\dot{z} = v \tag{9.4.19}$$

, where $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$ (9.4.20)

Define

$$A_m = A - bK \tag{9.4.21}$$

, where K is chosen so that the matrix $A_m = A - bK$ is Hurwitz.

Then,

$$\dot{e} = A_m e + b\left(f\left(e + \bar{x}_d, z\right) + \theta v - x_d^{\,n} + g\left(e + \bar{x}_d\right)\right)$$
$$\dot{z} = v \tag{9.4.22}$$

, where $\bar{x}_d = \begin{bmatrix} x_d & \cdots & x_d^{\,n-1} & x_d \end{bmatrix}$ (9.4.23)

Further, we define a dynamic signal described by:

$$\dot{r} = -c_0 r + r_m\left(e, \bar{x}_d\right)$$
$$r(0) > 0$$
$$c_0 > 0 \tag{9.4.25}$$
$$r_m = \left\|e + \bar{x}_d\right\|^2 \gamma_0 \left\|e + \bar{x}_d\right\| + d_0$$

Then the robust adaptive controller can be designed by:

$$v = -\beta e^T P b\left(\bar{f}\left(e + \bar{x}_d, z\right) + \left\|e + \bar{x}_d\right\|^2 + \left\|z\right\|^2 + \left(\alpha^{-1}(2r)\right)^2 + (Ke)^2 + 1\right)$$
$$\dot{\beta} = \beta_m\left(e, z, r, \bar{x}_d\right) - \Gamma\sigma\beta$$
$$\beta_m = \Gamma\left(e^T P b\right)^2\left(\bar{f}\left(e + \bar{x}_d, z\right) + \left\|e + \bar{x}_d\right\|^2 + \left\|z\right\|^2 + \left(\alpha^{-1}(2r)\right)^2 + (Ke)^2 + 1\right)$$
$$\Gamma > 0$$
$$\sigma > 0$$

(9.4.26)

$\alpha^{-1}(\bullet)$ is the inverse of function of $\alpha(\bullet)$

P is the solution of $\begin{matrix} PA_m + A_m{}^T = -Q \\ Q = Q^T > 0 \end{matrix}$ (9.4.27)

9.5 Related to the Diode Laser Processing System With $\dot{u}$

For controller design, we chose Robust Adaptive Control as introduced in Chapter 3. As shown in the identification, our laser welding system can be represented by:

$$\begin{aligned} &\ddot{y}(t) = \theta_1 y + \theta_2 \dot{y}(t) + \theta_3 + \theta_4 u(t) + \theta_5 u^2(t) \\ &+ \theta_6 u^3(t) + \theta_7 u^4(t) + \theta \dot{u} \end{aligned} \tag{9.4.28}$$

Define the state function as:

$$\begin{aligned} &x_1(t) = y(t) \\ &x_2(t) = \dot{y}(t) \\ &z(t) = u(t) \\ &\dot{z}(t) = v \end{aligned} \tag{9.4.29}$$

Then the system can be represented by:

$$\begin{aligned} &\dot{x}_1(t) = x_2(t) \\ &\dot{x}_2(t) = \theta_1 x_1(t) + \theta_2 x_2(t) + \theta_3 + \theta_4 z(t) + \theta_5 z^2(t) + \theta_6 z(t)^3 + \theta_7 z(t)^{\backslash 4} + \theta_8 v \end{aligned} \tag{9.4.30}$$

For simplicity, we let the nonlinear function be:

$$f(\cdot) = \theta_1 x_1(t) + \theta_2 x_2(t) + \theta_4 z(t) + \theta_5 z^2(t) + \theta_6 z(t)^3 + \theta_7 z(t)^4 \tag{9.4.31}$$

Accordingly, the amplitude limit function can be written as:

$$f(\cdot) \le \left\| \theta_1 x_1(t) + \theta_2 x_2(t) + \theta_4 z(t) + \theta_5 z^2(t) + \theta_6 z(t)^3 + \theta_7 z(t)^4 \right\| \tag{9.4.32}$$

$$f(\cdot) \le \theta\left(\sqrt{x_1^{\,2}} + \sqrt{x_2^{\,2}} + \sqrt{z^2} + z^2 + \sqrt{z^6}\right) \tag{9.4.33}$$

, where θ can be unknown

Thus:

$$\bar{f}(\cdot) = \sqrt{x_1^{\,2}} + \sqrt{x_2^{\,2}} + \sqrt{z^2} + z^2 + \sqrt{z^6} \tag{9.4.34}$$

Equation 9.4.34 gives the boundary of the nonlinearity function.

Let the tracking signal $y_r = \sin(t)$, the standard sinusoidal signal with amplitude 1

Then the error signal can be written by:

$$\begin{aligned} e_1(t) &= x_1(t) - y_r(t) \\ e_2(t) &= x_2(t) - \dot{y}_r(t) \end{aligned} \tag{9.4.35}$$

Or,

$$\begin{aligned} \dot{e}_1(t) &= e_2(t) \\ \dot{e}_2(t) &= \theta_1 + \theta_8 v + \theta_4 z + \theta_5 z^2 + \theta_6 z^3 + \theta_7 z^4 - \ddot{y}_r(t) \\ \dot{z}(t) &= v \end{aligned} \tag{9.4.36}$$

Or,

$$\dot{e}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left[f(e + y_r, z) - \ddot{y}_r(t) + \theta_5 v + \theta_3 \right] \tag{9.4.37}$$

Let A=$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ (9.4.38)

Then equation 9.3 can be rewritten by:

$$\dot{e}(t) = Ae + b\left[f(e + y_r, z) - \ddot{y}_r(t) + \theta_5 v + \theta_3 \right] \tag{9.4.39}$$

With simpler substitution, the error matrix is:

$$\dot{e}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left[f(e + y_r, z) - \ddot{y}_r(t) + \theta_5 v + \theta_3 \right] \tag{9.4.40}$$

To design the controller, we reformat the equation as:

$$\dot{e}(t) = A_m e + b\left[Ke + f(e + y_r, z) - \ddot{y}_r(k-3) + \theta_5 v + \theta_3 \right]$$

, where $A_m = A - bK$ and K is chosen so that A_m is Hurwitz.

(9.4.41)

$$\dot{e}(t)=\begin{bmatrix}0 & 1\\ -3 & -16\end{bmatrix}e+\begin{bmatrix}0\\ 1\end{bmatrix}\left[\begin{bmatrix}3 & 16\end{bmatrix}\begin{bmatrix}e_1\\ e_2\end{bmatrix}+f\left(e+y_r,z\right)-\ddot{y}_r(t)+\theta_5 v+\theta_3\right]$$

(9.4.42)

For matrix:

$$A_m=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}-\begin{bmatrix}0\\ 1\end{bmatrix}\begin{bmatrix}k_1 & k_2\end{bmatrix} \tag{9.4.43}$$

$$A_m=\begin{bmatrix}0 & 1\\ -k_1 & -k_2\end{bmatrix} \tag{9.4.44}$$

The constant matrix $K=\begin{bmatrix}k_1 & k_2\end{bmatrix}$ is chosen so that the roots of the characteristic equation have negative real parts.

We then can design the robust adaptive controller

$$v=-\beta e^T Pb\left\{\left[f\left(e+y_r,z\right)\right]^2+\left\|e+y_r\right\|^2+\left\|z\right\|^2+\left[\alpha^{-1}(2r)\right]^2+(Ke)^2+1\right\}$$

(9.4.45)

, where β is the adaptive parameter of the controller and $\alpha_1{}^{-1}$ is the inverse function and is a function of class K_∞. For now, we assume $\alpha_1(\cdot)=\|\cdot\|^2$

P is Lyapunov matrix under the condition of:

$PA_m+A_m{}^T=-Q$, where $Q=Q^T>0$ (9.4.46)

$$\dot{\beta}=\beta_m\left(e,z,r,y_r\right)-\Gamma\sigma\beta \tag{9.4.47}$$

$$\beta_m=\Gamma\left(e^T Pb\right)^2\left\{\left[f\left(e+y_r,z\right)\right]^2+\left\|e+y_r\right\|^2+\left\|z\right\|^2+\left[\alpha^{-1}(2r)\right]^2+(Ke)^2+1\right\}$$

(9.4.48)

For the design constants we assume they are known and satisfy the condition of

$T>0,\sigma>0$. (9.4.49)

Because in our experiments, only the position signal is detected, we further implement high gain observer for the purpose of output feedback.

Let the error signal be:

$$\dot{\hat{e}}_1 = \hat{e}_2 + \left(\frac{\sigma_1}{\varepsilon}\right)\left(e_1 - \hat{e}_1\right)$$
$$\hat{e}_2 = \left(\frac{\sigma_2}{\varepsilon^2}\right)\left(e_1 - \hat{e}_1\right) \tag{9.4.50}$$

, where $\varepsilon > 0$ is a small constant, $\sigma_i > 0, i = 1,2$ are chosen so that $A_n = A - K_\sigma C$ is a Hurwitz matrix, and $K_\sigma = \begin{bmatrix}\sigma_1 & \sigma_2\end{bmatrix}$, $C = \begin{bmatrix}1 & 0\end{bmatrix}$.

Accordingly for the matrix:

$$A_n = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} - \begin{bmatrix}\sigma_1 & \sigma_2\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} \tag{9.4.51}$$

$$A_n = \begin{bmatrix}-\sigma_1 & 1\\ -\sigma_2 & 0\end{bmatrix} \tag{9.4.52}$$

The positive constants matrix $K_\sigma = \begin{bmatrix}\sigma_1 & \sigma_2\end{bmatrix}$ is chosen so that the characteristic equation $s^2 + \sigma_1 s + \sigma_2 = 0$ has the roots with negative real parts.

(9.4.53)

To eliminate peaking in the implementation of the observer, we define

$$\hat{e}_1 = \frac{q_1}{\varepsilon}$$
$$\hat{e}_2 = q_2 \tag{9.4.54}$$

Thus:

$$\varepsilon\dot{q}_1 = q_2 + \sigma_1\left(e_1 - q_1\right)$$
$$\varepsilon\hat{q}_2 = \sigma_2\left(e_1 - q_1\right) \tag{9.4.55}$$

In order to prevent the peaking from entering the control system, we saturate the control signal and adaptive controller outside of their domains of interests. In our experiments,

$$M_v = 50$$
$$M_\beta = 30$$

(9.4.56)

With these constants, which are larger than or equal to the upper bound of those signals $v(e,r,\bar{y}_r,\beta), \beta_m(e,r,\bar{y}_r)$, we can denote the following equations:

$$v^S(e,r,\bar{y}_r,\beta) = M_v sat\left(\frac{v(e,r,\bar{y}_r,\beta)}{M_v}\right). \tag{9.4.57}$$

$$\beta_m{}^S(e,r,\bar{y}_r) = M_\beta sat\left(\frac{\beta_m(e,r,\bar{y}_r)}{M_\beta}\right) \tag{9.4.58}$$

$sat(\ )$ represents the saturation function. (9.4.59)

Thus the robust adaptive output controller can be obtained by replacing $v(e,r,\bar{y}_r,\beta)$ and $\beta_m(e,r,\bar{y}_r)$ with $v^S(e,r,\bar{y}_r,\beta)$ and $\beta_m{}^S(e,r,\bar{y}_r,\beta)$

Simulation are made under both step and sin(t) as tracking signals. In both cases, 0.5sin(t) signal is added as the disturbance.

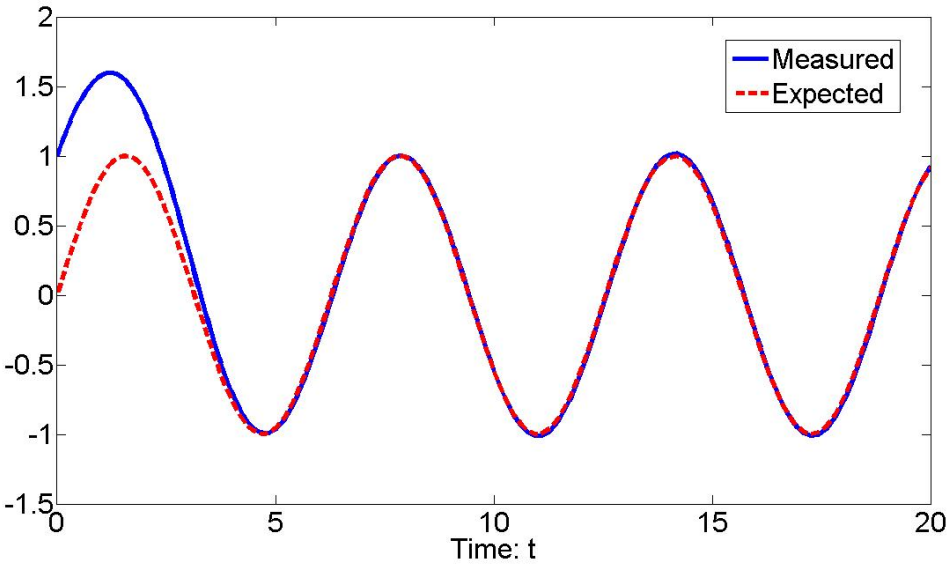


Figure 9.4.a Tracking sin(t) with $\dot{u}(t)$ in the model

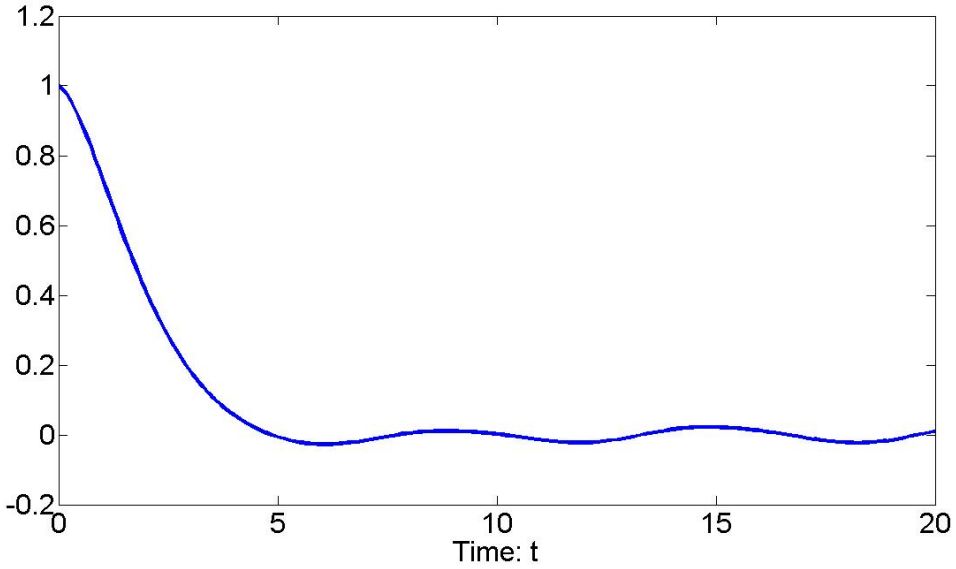


Figure 9.4.b Tracking error of sin(t) with $\dot{u}(t)$ in the model

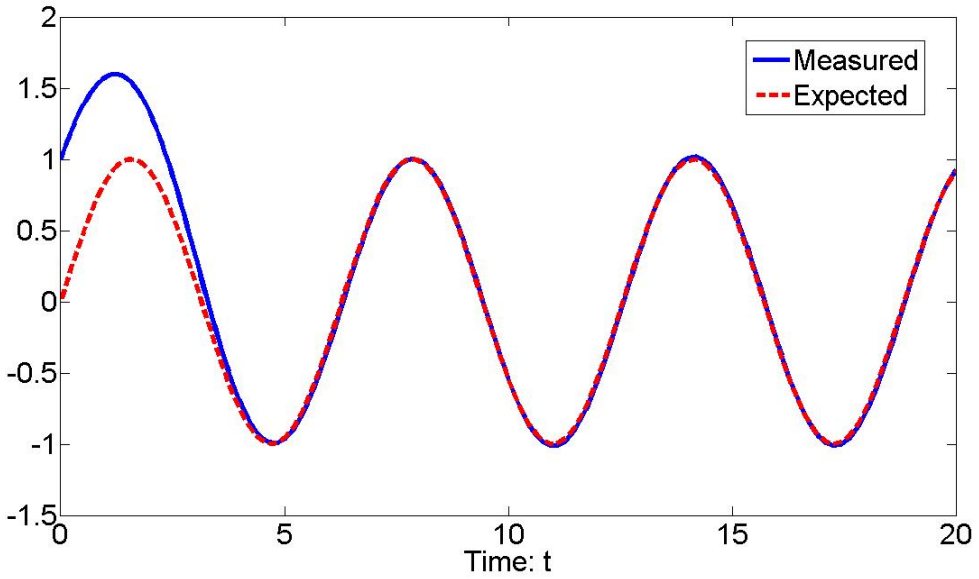


Figure 9.5.a Observer based Tracking sin(t) with $\dot{u}(t)$ in the model

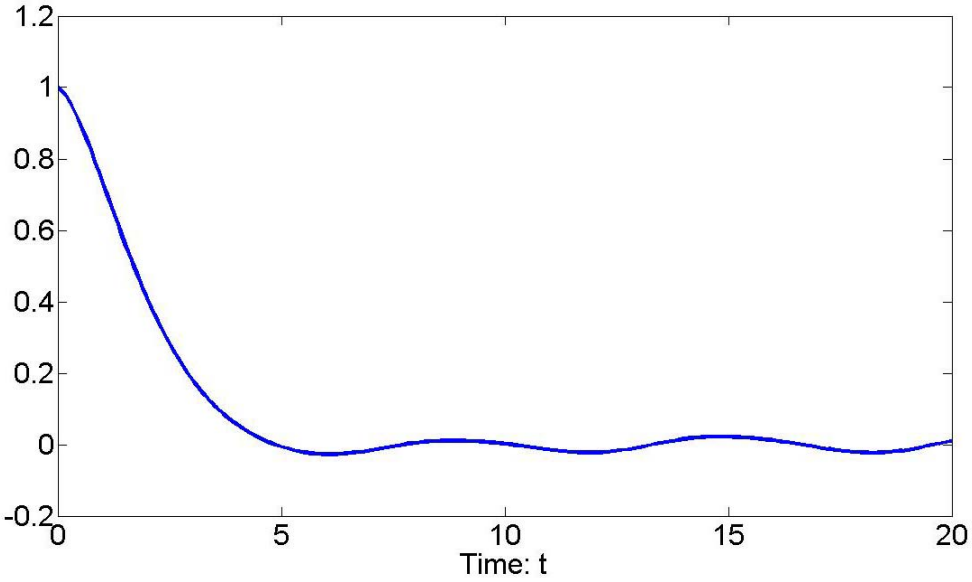


Figure 9.5.b Observer based Tracking sin(t) with $\dot{u}(t)$ in the model

9.6 Proof:

We consider the system that can be represented by:

$$y^n(y) = f\left(y, \dot{y}, \cdots, y^{n-1}\right) + \theta u + g(u) \tag{9.6.1}$$

, where y, u are the output and the control respectively; $y^i, i = 1,2,\cdots n$ is the ith derivative of y.

Assume

$$f\left(y, \dot{y}, \cdots, y^{n-1}\right) \le \theta \bar{f}\left(y, \dot{y}, \cdots, y^{n-1}\right), \text{ where } \theta > 0 \tag{9.6.2}$$

$$g(u) \le c\|g(u)\|, \text{ where } c > 0$$

Let

$$
\begin{aligned}
&x_1 = y \\
&x_2 = \dot{y} \\
&\cdots\cdots \\
&x_{n-1} = y^{n-2} \\
&x_n = y^{n-1} \quad \text{, where } v \text{ is the virtual control input for the system} \\
&z = u \\
&z = v
\end{aligned}
\tag{9.6.3}
$$

Then we have:

$$
\begin{aligned}
&\dot{x}_1 = x_2 \\
&\dot{x}_2 = x_3 \\
&\cdots\cdots \\
&\dot{x}_i = x_{i+1} \\
&\cdots \\
&\dot{x}_{n-1} = x_n \\
&\dot{x}_n = f(x_1, x_2, \cdots, x_n) + \theta v \\
&z = u \\
&z = v
\end{aligned}
\tag{9.6.4}
$$

Or,

$$
\begin{aligned}
&\dot{x}_i = x_{i+1} \\
&\dot{x}_n = f(x) + u \\
&1 \le i \le n-1
\end{aligned}
\tag{9.6.5}
$$

, where $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$

During the identification, $g(u)$ is approximated by a polynomial. In fact, it can be other complex formats. However, we assume that $g(u)$ is continuous and bounded

We further assume all the states of the system are available for feedback and define:

$$
\begin{aligned}
e_1 &= x_1 - x_d \\
e_2 &= x_2 - \dot{x}_d \\
&\cdots\cdots \\
e_i &= x_i - x_d^{\,i-1} \\
&\cdots \\
e_{n-1} &= x_n - x_d^{\,n-2} \\
e_n &= x_n - x_d^{\,n-1}
\end{aligned}
\tag{9.6.6}
$$

Then we have,

$$
\begin{aligned}
\dot{e}_1 &= x_2 - x_d \\
\dot{e}_2 &= x_3 - \ddot{x}_d \\
&\cdots\cdots \\
\dot{e}_{n-1} &= x_n - x^n{}_d \\
\dot{e}_n &= \dot{x}_n - x_d^{\,n}
\end{aligned}
\tag{9.6.7}
$$

With simple substitution, we can have:

$$
\begin{aligned}
\dot{e} &= Ae + b\left(f\left(e + \bar{x}_d, z\right) + \theta v - x_d^{\,n} + g\left(e + \bar{x}_d\right)\right) \\
z &= v
\end{aligned}
\tag{9.6.8}
$$

, where $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$ (9.6.9)

Define

$$
A_m = A - bK \tag{9.6.10}
$$

, where K is chosen so that the matrix $A_m = A - bK$ is Hurwitz.

Then,

$$\begin{aligned} \dot{e} &= A_m e + b\left(f\left(e+\bar{x}_d, z\right) + \theta v - x_d^{\,n} + g\left(e+\bar{x}_d\right)\right) \\ z &= v \end{aligned} \tag{9.6.11}$$

$$\text{, where } \bar{x}_d = \begin{bmatrix} x_d & \cdots & x_d^{\,n-1} & x_d \end{bmatrix} \tag{9.6.12}$$

Further, we define a dynamic signal described by:

$$\begin{aligned} \dot{r} &= -c_0 r + r_m\left(e, \bar{x}_d\right) \\ r(0) &> 0 \\ c_0 &> 0 \\ r_m &= \left\|e+\bar{x}_d\right\|^2 \gamma_0 \left\|e+\bar{x}_d\right\| + d_0 \end{aligned} \tag{9.6.13}$$

From [60], signal r has the property of :

$$V \le r(t) + D(t) \tag{9.6.14}$$

Then the robust adaptive controller can be designed by:

$$\begin{aligned} v &= -\beta e^T P b\left(\bar{f}\left(e+\bar{x}_d, z\right) + \left\|e+\bar{x}_d\right\|^2 + \left\|z\right\|^2 + \left(\alpha^{-1}(2r)\right)^2 + (Ke)^2 + 1\right) \\ \dot{\beta} &= \beta_m\left(e, z, r, \bar{x}_d\right) - \Gamma\sigma\beta \\ \beta_m &= \Gamma\left(e^T P b\right)^2\left(\bar{f}\left(e+\bar{x}_d, z\right) + \left\|e+\bar{x}_d\right\|^2 + \left\|z\right\|^2 + \left(\alpha^{-1}(2r)\right)^2 + (Ke)^2 + 1\right) \\ \Gamma &> 0 \\ \sigma &> 0 \end{aligned} \tag{9.6.15}$$

$$\alpha^{-1}(\bullet) \text{ is the inverse of function of } \alpha(\bullet) \tag{9.6.16}$$

$$P \text{ is the solution of } \begin{aligned} PA_m + A_m^{\,T} &= -Q \\ Q = Q^T &> 0 \end{aligned} \tag{9.6.17}$$

Define the Lyapunov candidate function:

$$V = e^T P e + \theta\Gamma^{-1}\left(\beta - \hat{\beta}\right)^2 \tag{9.6.18}$$

, where $\hat{\beta}$ is a positive constant and the desired value of β

Take the first order derivative

$$
\begin{aligned}
&\dot{V} \\
&= \dot{e}^T Pe + e^T P\dot{e} + 2\theta\Gamma^{-1}\left(\beta - \hat{\beta}\right)\dot{\beta} \\
&= -\left(A_m e + b\left(f\left(e + \bar{x}_d, z\right) + \theta v - x_d^{\,n} + g\left(e + \bar{x}_d\right)\right)\right)^T Pe \\
&+ e^T P\left(A_m e + b\left(f\left(e + \bar{x}_d, z\right) + \theta v - x_d^{\,n} + g\left(e + \bar{x}_d\right)\right)\right) + V_1
\end{aligned}
\tag{9.6.19.a}
$$

, where

$$
\begin{aligned}
&V_1 \\
&= \\
&2\theta\Gamma^{-1}\left(\beta - \hat{\beta}\right)\left(\Gamma\left(e^T Pb\right)^2\left(\bar{f}\left(e + \bar{x}_d, z\right) + \left\|e + \bar{x}_d\right\|^2 + \|z\|^2 + \left(\alpha^{-1}(2r)\right)^2 + (Ke)^2 + 1\right) - \Gamma\sigma\beta\right)
\end{aligned}
\tag{9.6.19.b}
$$

Apparently,

$$
\begin{aligned}
&V \\
&\leq \\
&-e^T Qe \\
&+ 2\left\|e^T Pb\right\|\|Ke\| + 2\left\|e\ Pb\right\|\left(\theta\bar{f}\left(e + \bar{x}_d, z\right) + c_1\left\|e + \bar{x}_d\right\| + c_2(\|z\|) + \left\|x_d^{\,n}\right\| + c_3\left\|g\left(e + \bar{x}_d\right)\right\|\right) \\
&- 2\theta\hat{\beta}\left(e^T Pb\right)^2\left(\left(\bar{f}\left(e + \bar{x}_d, z\right)\right)^2 + \left\|e + \bar{x}_d\right\|^2 + \|z\|^2 + \left(\alpha^{-1}(2r)\right)^2 + (Ke)^2 + 1\right) \\
&- \theta\sigma\beta^2 + \theta\sigma\hat{\beta}^2 - \theta\sigma\left(\beta - \hat{\beta}\right)^2
\end{aligned}
$$

If we introduce a constant:

$$
c_4 = \sup\left(\left\|y_r^{\,n}\right\| + c_3\left\|g\left(e + \bar{x}_d\right)\right\| + \alpha^{-1}(D(t))\right)
\tag{9.6.20}
$$

Then, with substitution, equation (9.6.19) can be rewritten by:

$$\dot{V} \le -e^T Qe + 2\bar{\beta}\theta\left(\left(e^T Pb\right)\left(Ke\right) - \frac{1}{2\hat{\beta}\theta}\right)^2$$
$$- 2\bar{\beta}\theta\left(\left(e^T Pb\right)\bar{f}\left(e+\bar{x}_d, z\right) - \frac{\theta}{2\hat{\beta}\theta}\right)^2$$
$$+ \frac{\theta^2}{2\hat{\beta}\theta} - 2\bar{\beta}\theta\left(\left(e^T Pb\right)\left(e+\bar{x}_d\right) - \frac{c_1}{2\hat{\beta}\theta}\right)^2$$
$$+ \frac{{c_1}^2}{2\hat{\beta}\theta} - 2\bar{\beta}\theta\left(\left(e^T Pb\right)\left(z\right) - \frac{c_2}{2\hat{\beta}\theta}\right)^2$$
$$+ \frac{{c_2}^2}{2\hat{\beta}\theta} - 2\bar{\beta}\theta\left(\left(e^T Pb\right)\left(\alpha^{-1}\left(2r\right)\right) - \frac{c_3}{2\hat{\beta}\theta}\right)^2$$
$$+ \frac{{c_3}^2}{2\hat{\beta}\theta} - 2\bar{\beta}\theta\left(\left(e^T Pb\right) - \frac{c_4}{2\hat{\beta}\theta}\right)^2$$
$$+ \frac{{c_4}^2}{2\hat{\beta}\theta} - \theta\sigma\beta^2 + \theta\sigma\hat{\beta}^2 - \theta\sigma\left(\beta - \hat{\beta}\right)^2 \tag{9.6.21}$$

Accordingly,

$$\dot{V} \le -e^T Qe - \theta\sigma\left(\beta - \hat{\beta}\right)^2 + M$$
$$M = \frac{1}{2\beta\theta}\left({c_1}^2 + {c_2}^2 + {c_3}^2 + {c_4}^2 + \theta^2 + 1\right) + \sigma\theta\hat{\beta}^2 \tag{9.6.22}$$

Therefore, according to [60] V will decrease monotonically, which means that $\left(e, \beta\right)$ are bounded. The system states are also bounded and the closed loop system is bounded and asymptotically stable.

9.7 Related to the Diode Laser Processing System Without $\dot{u}$

As shown in the identification, our laser welding system can be represented by:

$$\ddot{y}\left(t\right) = \theta_1 y + \theta_2 \dot{y}\left(t\right) + \theta_3 + \theta_4 u\left(t\right) + \theta_5 u^2\left(t\right)$$
$$+ \theta_6 u^3\left(t\right) + \theta_7 u^4\left(t\right)$$

Define the state function as:

$$
\begin{aligned}
x_1(t) &= y(t) \\
x_2(t) &= \dot{y}(t) \\
z(t) &= u(t) \\
z(t) &= v
\end{aligned}
\tag{9.7.1}
$$

Then the system can be represented by:

$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= \theta_1 x_1(t) + \theta_2 x_2(t) + \theta_3 + \theta_4 z(t) + \theta_5 z^2(t) + \theta_6 z(t)^3 + \theta_7 z(t)^{\backslash 4} + \theta_8 v
\end{aligned}
\tag{9.7.2}
$$

For simplicity, we let the nonlinear function be:

$$
f(\cdot) = \theta_1 x_1(t) + \theta_2 x_2(t) + \theta_4 z(t) + \theta_5 z^2(t) + \theta_6 z(t)^3 + \theta_7 z(t)^4 \tag{9.7.3}
$$

Accordingly, the amplitude limit function can be written as:

$$
f(\cdot) \le \left\| \theta_1 x_1(t) + \theta_2 x_2(t) + \theta_4 z(t) + \theta_5 z^2(t) + \theta_6 z(t)^3 + \theta_7 z(t)^4 \right\| \tag{9.7.4}
$$

$f(\cdot) \le \theta\left(\sqrt{x_1^{\,2}} + \sqrt{x_2^{\,2}} + \sqrt{z^2} + z^2 + \sqrt{z^6} \right)$, where θ can be unknown

Thus:

$$
\bar{f}(\cdot) = \sqrt{x_1^{\,2}} + \sqrt{x_2^{\,2}} + \sqrt{z^2} + z^2 + \sqrt{z^6} \tag{9.7.5}
$$

Equation 9.2 gives the boundary of the nonlinearity function.

Let the tracking signal $y_r = \sin(t)$, the standard sinusoidal signal with amplitude 1

Then the error signal can be written by:

$$
\begin{aligned}
e_1(t) &= x_1(t) - y_r(t) \\
e_2(t) &= x_2(t) - \dot{y}_r(t)
\end{aligned}
\tag{9.7.6}
$$

Or,

$$
\begin{aligned}
e_1(t) &= e_2(t) \\
\dot{e}_2(t) &= \theta_1 + \theta_8 v + \theta_4 z + \theta_5 z^2 + \theta_6 z^3 + \theta_7 z^4 - \ddot{y}_r(t) \\
z(t) &= v
\end{aligned}
\tag{9.7.7}
$$

Or,

$$
\dot{e}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left[f(e + y_r, z) - \ddot{y}_r(t) + \theta_5 v + \theta_3 \right] \tag{9.7.8}
$$

Let A=$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ (9.7.9)

Then equation 9.3 can be rewritten by:

$$\dot{e}(t) = Ae + b\left[f(e + y_r, z) - \ddot{y}_r(t) + \theta_5 v + \theta_3\right] \tag{9.7.10}$$

With simpler substitution, the error matrix is:

$$\dot{e}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\left[f(e + y_r, z) - \ddot{y}_r(t) + \theta_5 v + \theta_3\right] \tag{9.7.11}$$

To design the controller, we reformat the equation as:

$$\dot{e}(t) = A_m e + b\left[Ke + f(e + y_r, z) - \ddot{y}_r(k-3) + \theta_5 v + \theta_3\right] \tag{9.7.12}$$

, where $A_m = A - bK$ and K is chosen so that A_m is Hurwitz.

$$\dot{e}(t) = \begin{bmatrix} 0 & 1 \\ -2 & -4 \end{bmatrix} e + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\left[\begin{bmatrix} 2 & 4 \end{bmatrix}\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + f(e + y_r, z) - \ddot{y}_r(t) + \theta_5 v + \theta_3\right] \tag{9.7.13}$$

For matrix:

$$A_m = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix}\begin{bmatrix} k_1 & k_2 \end{bmatrix} \tag{9.7.14}$$

$$A_m = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} \tag{9.7.15}$$

The constant matrix $K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$ is chosen so that the roots of the characteristic equation have negative real parts.

We then can design the robust adaptive controller

$$v = -\beta e^T Pb\left\{\left[f(e + y_r, z)\right]^2 + \left\|e + y_r\right\|^2 + \left\|z\right\|^2 + \left[\alpha^{-1}(2r)\right]^2 + (Ke)^2 + 1\right\} \tag{9.7.16}$$

, where β is the adaptive parameter of the controller and $\alpha_1{}^{-1}$ is the inverse function and is a function of class K_∞. For now, we assume $\alpha_1(\cdot) = \|\cdot\|^2$

P is Lyapunov matrix under the condition of:

$$PA_m + A_m{}^T = -Q, \text{ where } Q = Q^T > 0 \tag{9.7.17}$$

$$\dot{\beta} = \beta_m(e, z, r, y_r) - \Gamma\sigma\beta \tag{9.7.18}$$

$$\beta_m = \Gamma\left(e^T Pb\right)^2 \left\{\left[f\left(e+y_r,z\right)\right]^2 + \left\|e+y_r\right\|^2 + \left\|z\right\|^2 + \left[\alpha^{-1}\left(2r\right)\right]^2 + \left(Ke\right)^2 + 1\right\} \tag{9.7.19}$$

For the design constants we assume they are known and satisfy the condition of

$$T > 0, \sigma > 0. \tag{9.7.20}$$

Because in our experiments, only the position signal is detected, we further implement high gain observer for the purpose of output feedback.

Let the error signal be:

$$\begin{aligned} \dot{\hat{e}}_1 &= \hat{e}_2 + \left(\frac{\sigma_1}{\varepsilon}\right)\left(e_1 - \hat{e}_1\right) \\ \hat{e}_2 &= \left(\frac{\sigma_2}{\varepsilon^2}\right)\left(e_1 - \hat{e}_1\right) \end{aligned} \tag{9.7.21}$$

, where $\varepsilon > 0$ is a small constant, $\sigma_i > 0, i = 1,2$ are chosen so that $A_n = A - K_\sigma C$ is a Hurwitz matrix, and $K_\sigma = \begin{bmatrix}\sigma_1 & \sigma_2\end{bmatrix}$, $C = \begin{bmatrix}1 & 0\end{bmatrix}$. (9.7.22)

Accordingly for the matrix:

$$A_n = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} - \begin{bmatrix}\sigma_1 & \sigma_2\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} \tag{9.7.23}$$

$$A_n = \begin{bmatrix}-\sigma_1 & 1\\ -\sigma_2 & 0\end{bmatrix} \tag{9.7.24}$$

The positive constants matrix $K_\sigma = \begin{bmatrix}\sigma_1 & \sigma_2\end{bmatrix}$ is chosen so that the characteristic equation $s^2 + \sigma_1 s + \sigma_2 = 0$ has the roots with negative real parts.

To eliminate peaking in the implementation of the observer, we define

$$\begin{aligned} \hat{e}_1 &= \frac{q_1}{\varepsilon} \\ \hat{e}_2 &= q_2 \end{aligned} \tag{9.7.25}$$

Thus:

$$\begin{aligned} \varepsilon\dot{q}_1 &= q_2 + \sigma_1\left(e_1 - q_1\right) \\ \varepsilon\dot{q}_2 &= \sigma_2\left(e_1 - q_1\right) \end{aligned} \tag{9.7.26}$$

In order to prevent the peaking from entering the control system, we saturate the control signal and adaptive controller outside of their domains of interests. In our experiments,

$$M_v = 50$$
$$M_\beta = 30 \tag{9.7.27}$$

With these constants, which are larger than or equal to the upper bound of those signals $v(e, r, \bar{y}_r, \beta), \beta_m(e, r, \bar{y}_r)$, we can denote the following equations:

$$v^S(e, r, \bar{y}_r, \beta) = M_v sat\left(\frac{v(e, r, \bar{y}_r, \beta)}{M_v}\right) \tag{9.7.28}$$

$$\beta_m{}^S(e, r, \bar{y}_r) = M_\beta sat\left(\frac{\beta_m(e, r, \bar{y}_r)}{M_\beta}\right) \tag{9.7.29}$$

$sat(\ )$ represents the saturation function. (9.7.30)

Thus the robust adaptive output controller can be obtained by replacing $v(e, r, \bar{y}_r, \beta)$ and $\beta_m(e, r, \bar{y}_r)$ with $v^S(e, r, \bar{y}_r, \beta)$ and $\beta_m{}^S(e, r, \bar{y}_r, \beta)$

9.8 Simulation

We choose the constants:

$$\sigma = 0.00001, K = \begin{bmatrix} 2 & 4 \end{bmatrix}, P = \begin{bmatrix} 1.375 & 0.25 \\ 0.25 & 0.1875 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The simulation track two different input signal respectively: Both simulations are tested with adding 0.5sin(t) as disturbance.

Sinusoidal signal input:

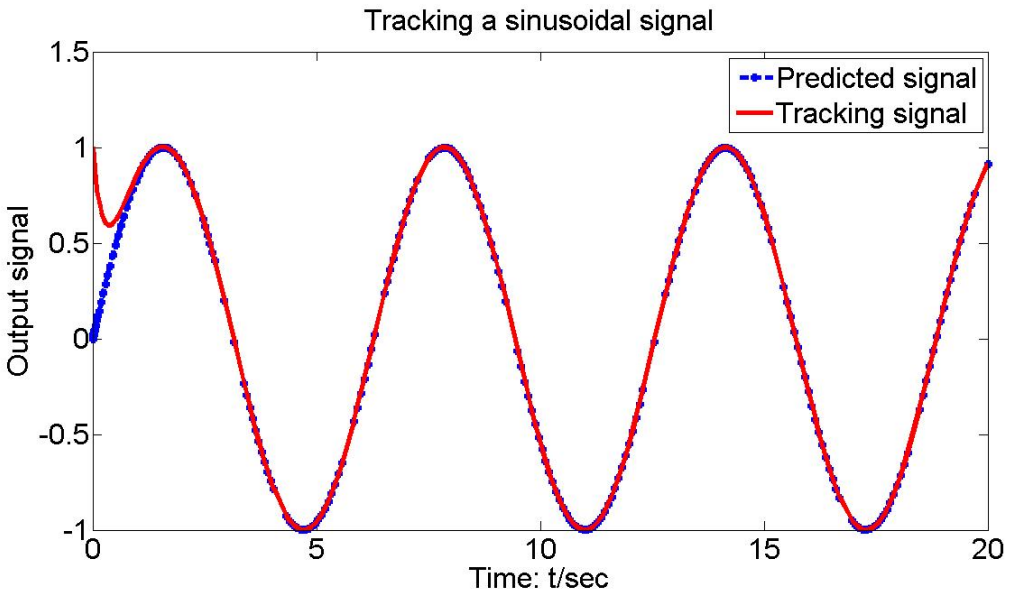


Figure 9.6.a Tracking signal sin(t)

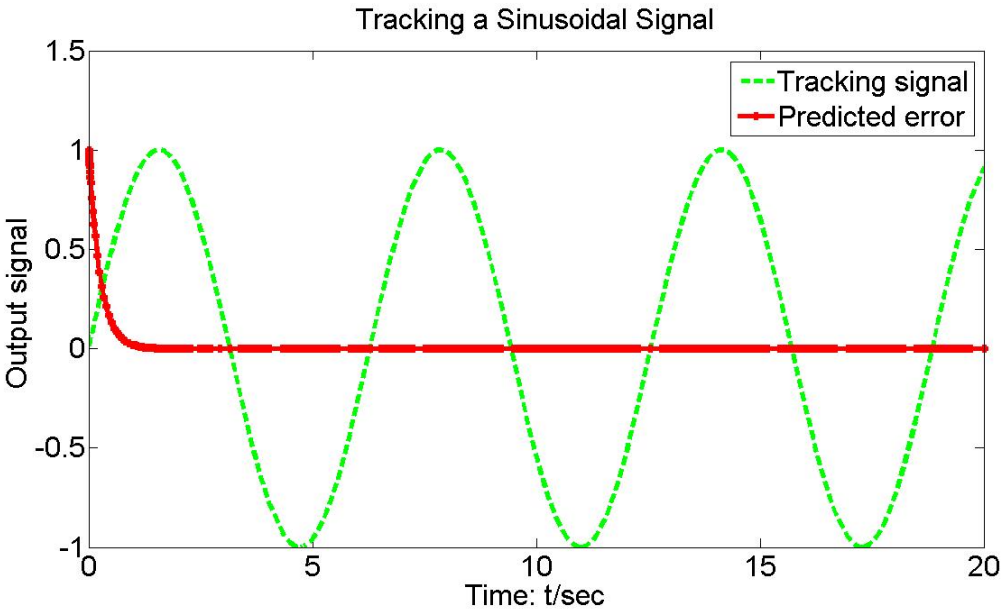


Figure 9.6.b Tracking error: input signal sin(t)

Step signal input

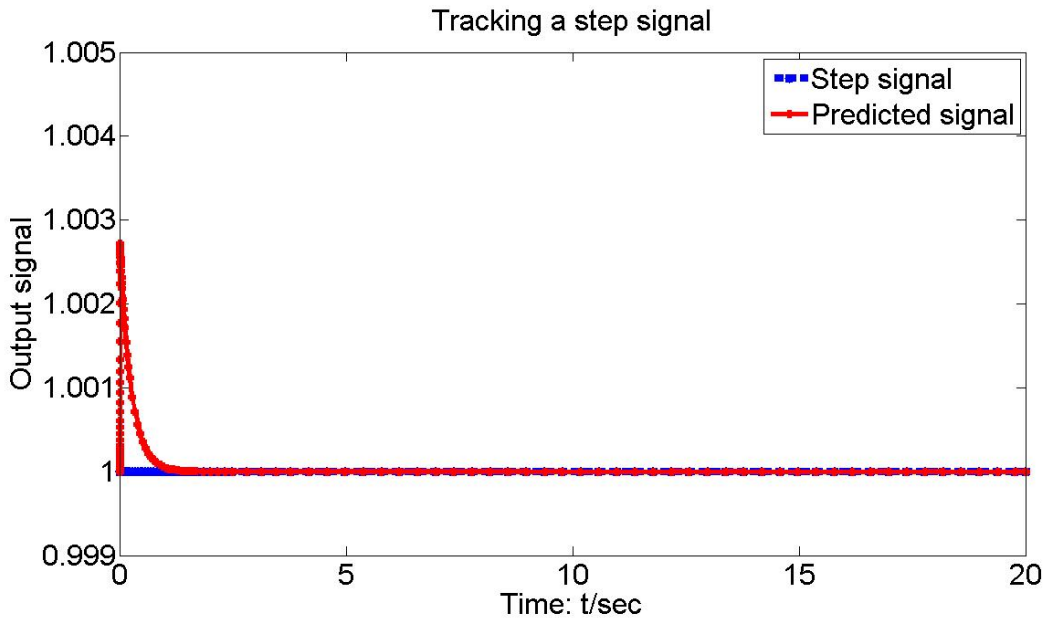


Figure 9.7.a Tracking step signal

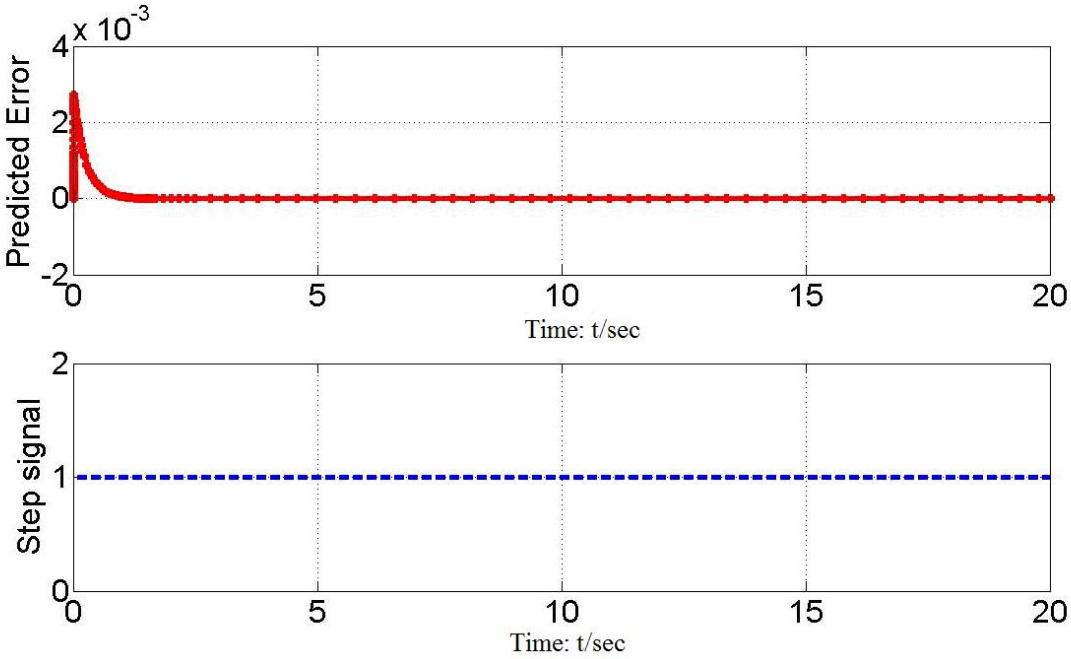


Figure 9.7.b Tracking error: step signal

Output feedback: high-gain observer

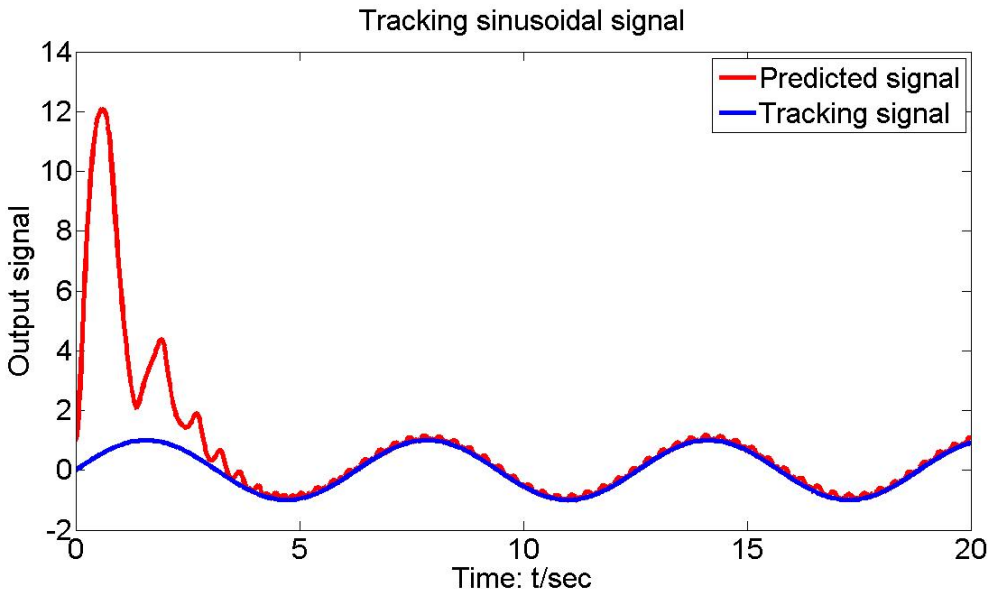


Figure 9.8.a Tracking signal sin(t): Output feedback

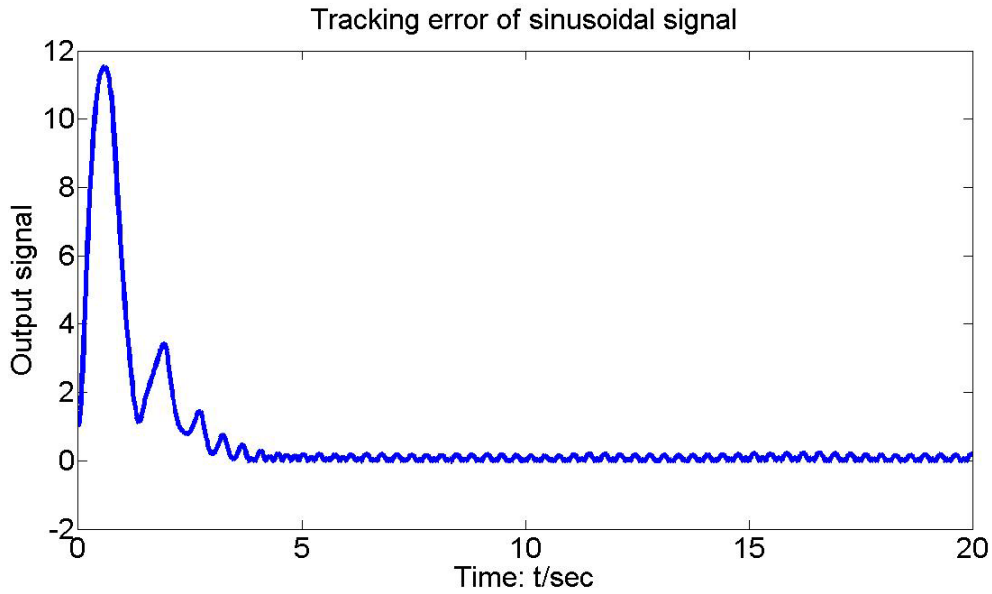


Figure 9.8.b Output feedback tracking error: signal sin(t)

Step Input:

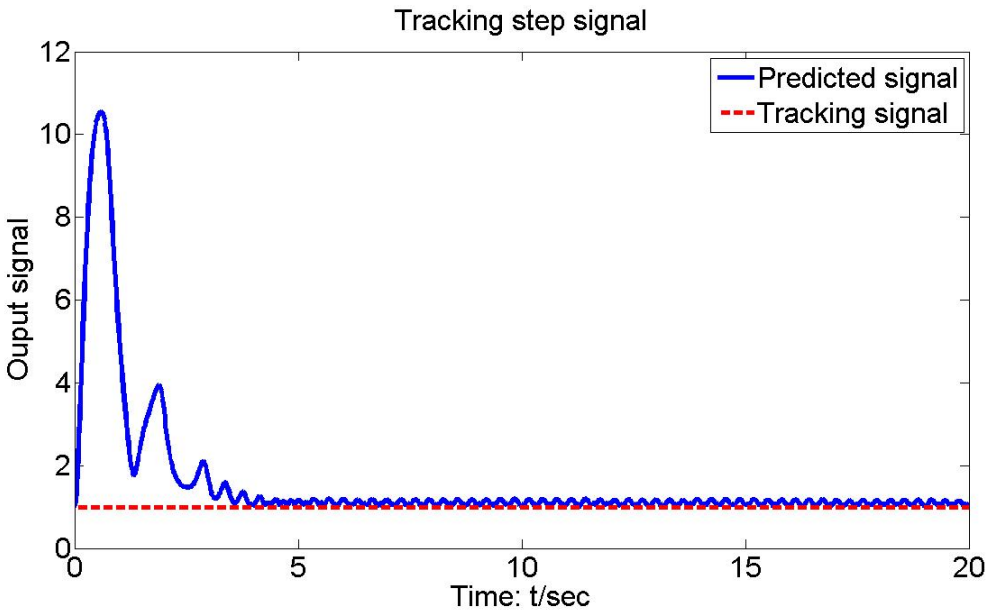


Figure 9.9.a Tracking step signal: Output feedback

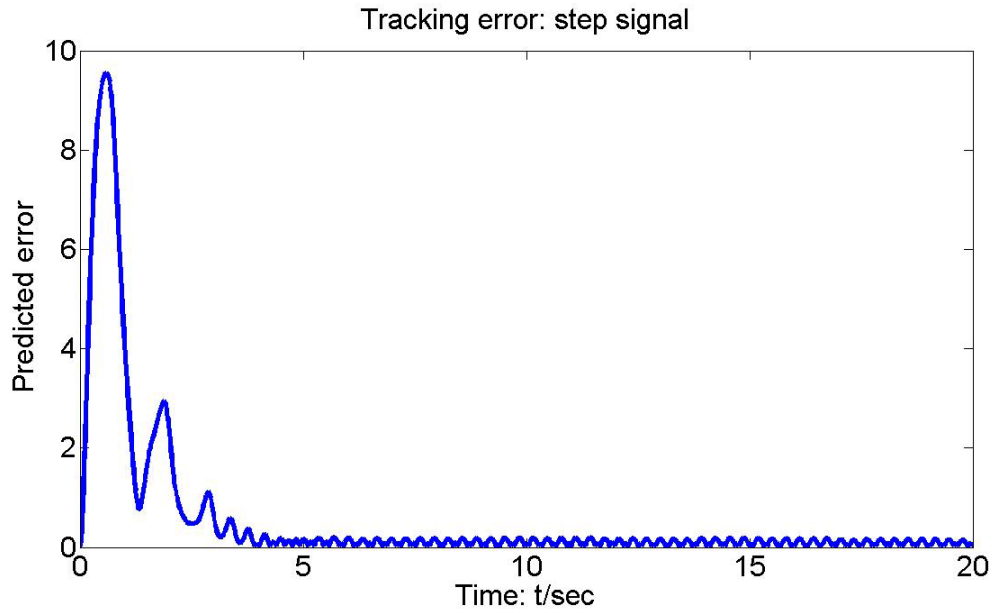


Figure 9.9.b Output feedback tracking error: step signal

9.9 Summary

Apparently the state feedback tracks better with either Sinusoidal or Step signal although he observer design somewhat caused a fluctuation with very small amplitude. This might require further tuning on the constants

CHAPTER 10

CONCLUSION AND FUTURE WORK

In this dissertation, we have proposed a simple and practical nonlinear identification method for manufacturing processes, especially laser welding process. The model is based on nonlinear Hammerstein structure consisting of static nonlinearity and linear dynamics in series with each other. To better serve the study, we built a prototype of diode laser welding system with aid of computer vision. We studied the identification both in continuous and discrete case. For now, a SISO is mainly focused and the model takes speed as the input and the top surface weld pool width as the output. However, the algorithm can be expanded to MIMO with suitable persistent excitation signals.

Through experiments, we have validated the model and proved that the algorithm is capable of identifying the nonlinearity and linear dynamics in certain degree of accuracy. For simplicity, the nonlinearity is approximated by a standard polynomial with selected order and can also be estimated with other format of nonlinearity. The linear dynamic is represented by transfer function in either continuous or discrete case. Although we have identified the model in offline, this algorithm can be capable of real time application. Currently we acquire the experimental data based on vision sensor and image processing. To greatly avoid the influence of measurement error, we choose offline study for identification.

The proposed identification is under conditions, especially non-zero in the linear dynamics because we simplify the denominator with a constant. Although we have proved that under circumstances this configuration is reasonable for laser welding processes, in the future it might still be necessary to identify the model with adding zeroes. Persistent excitation test signal such as multi-level amplitude sinusoidal signal might be of interest for related investigation. Extended MIMO identification is also suggesting for future research.

Current identification is based on open-loop experiments. In the future, experiments applied to more manufacturing system related to close-loop identification and control system will be studied.

REFERENCES

[1] Migliore, L., “Welding with Lasers,” Industrial Laser Review, 1998

[2] Sun, A., Kannatey-Asibu, Jr. E., “Sensor systems for real-time monitoring of laser weld quality,” Journal of Laser Applications 11(4): 153-168, 1999

[3] Sun, A., Kannatey-Asibu, Jr. E., “Monitoring of Laser Weld Penetration Using Sensor Fusion,” Journal of Laser Applications 14(2): 114-121, 2002

[4] Pedrotti, F. L. a. P., L.S., “Introduction to Optics,” New Jersey, Prentice Hall, 1993

[5] Williams, C., “CO Laser Processing-An Overview,” Aircraft Engineering and Aerospace Technology 69(1): 43-52, 1997

[6] Renwick, R. J., and Richardson, R.W., “Experimental investigations of GTA weld pool oscillations,” Welding Journal 62(2): 29-35, 1983

[7] Xiao, Y. H. a. O. G. d., “Weld pool oscillation during GTA welding of mild steel,” Welding Journal 72(8): 428-434, 1993

[8] Nagarajan, S., Bangerjee, P., Chen, W., and Chin, B.A., “Control of the welding process using infrared sensors,” IEEE Transactions on Robotics and Automation 8(1): 86-93, 1992

[9] Beardsley, H. E., Zhang, Y.M., and Kovacevic, R., (1994). "Infrared sensing of full penetration state in gas tungsten arc welding." International Journal of Machine Tool and Manufacturing 34(8): 1079-1090.

[10] Hopko, S. N. a. U., I. C., “Laser generated ultrasound by material ablation using fiber optic delivery,” Ultrasonic 37(1): 1-7, 1999

[11] Farson, D. F. a. A., A., “Laser welding penetration monitoring with multiple emission signal measurements,” Journal of Laser Applications 11(2): 47-53, 1999

[12] Graham, G. M. a. U., I. C., “Automated system for laser ultrasonic sensing of weld penetration,” Mechatronics 7(8): 711-721, 1997

[13] Lankalpall, K. N. T., J. F, and Gartner, M., “A model for Estimating Penetration Depth of Laser Welding Processes,” Journal of Physics: Apply Physics 29(7): 1831-1841, 1996

[14] Becker, S. a. B., V.M.J., “Semi-Automatic 3D Model Extraction from Uncalibrated 2D Camera Views,” SPIE Visual Data Exploration and Analysis II 2410: 447-461, 1995

[15] Beersiek, J., “A CMOS Camera as a Tool for Process Analysis Not Only for Laser Beam,” ICALEO’01 92&93: 1185-1193, 2000

[16] Park, H. a. R. S., “Estimation of weld bead size in CO2 laser welding by using multiple regression and neural network,” Journal of Laser Applications 11(3): 143-150, 1999

[17] Kovacevic, R., Zhang, Y. M., and Ruan, S. “Sensing and control of weld pool geometry for automated GTA welding,” ASME Journal of Engineering for Industry 117(2): 210-222, 1995

[18] Kovacevic, R., Zhang, Y. M., “Sensing free surface of arc weld pool using specular reflection: principle and analysis,” Proceedings of the Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacturing 210(6): 553-564, 1996

[19] Kovacevic, R., Zhang, Y. M., “Real-time image processing for monitoring of free weld pool surface,” ASME Journal of Manufacturing Science and Engineering 119(2): 161-169, 1997

[20] Zhang, Y. M. K., R., “Robust Control of Interval Plants: A time Domain Method." IEEE P-CONTR THEOR AP 144(4): 347-353, 1997

[21] Zhang, Y. M. K., R. (1998), “Neurofuzzy Model-based Predictive Control of Weld Fusion Zone Geometry,” IEEE Transaction on Fuzzy Systems 6(3): 389-401, 1998

[22] Zhang, Y. P., Ioannou, P., “Robustness of Nonlinear Control Systems with Respect to Unmodeled Dynamics,” IEEE Transactions on Automatic Control 44(1), 1999

[23] Morgan, S. A. F., M D T; McLean, M A; Hand, D P; Haran, F M; Su, D; Steen, W M; Jones, J D C, “Real-time process control in CO2 laser welding and direct casting: focus and temperature,” ICALEO '97: Laser Materials Processing. San Diego, California; USA. 83: G290-G299, 1997

[24] C. S. Wu, P. C. Zhao and Y. M. Zhang, “Numerical simulation of transient 3-D surface deformation of full-penetrated GTA weld pool,” Welding Journal, Vol. 83, No. 12, pp. 330-335, 2004.

[25] Haran, F. M., Hand, D. P., Peters, C. and Jones, J. D. C. (1996). "Real-time Focus Control in Laser Welding." Meas. Sci. Technol 7(8): August 1996.

[26] Y. Kawahito and S. Katayama, "In-process monitoring and feedback control during laser microspot lap welding of copper sheets," Journal of Laser Applications, vol. 16, no. 2, pp. 121-127, May 2004.

[27] Bagger. C. and Olsen, Flemming, "Laser welding closed-loop power control," Journal of Laser Applications, 15(1): 19-24, February 2003,

[28] Postma, S, Aarts, R. G. K. M., Meijer, J. and jonker, J. B, "Penetration control in laser welding of sheet metal," Journal of Laser Applications, 14(4): 21-214, November 2002

[29] Duan, P. Y. and Zhang, Y. M., "CMAC-based modelling for HPDDL welding process control," Special Issue on Advanced Sensing, Modelling and Control of Welding Processes, International Journal of Modelling, Identification and Control, vol. 1, no. 2(2), pp 107-114, 2006

[30] Abonyi, J., "Identification and Control of Nonlinear Systems Using Fuzzy Hammerstein Models," Industrial and Engineering Chemistry Research **39**(11): 4302-4314, 2000

[31] Ljung, L., "System Identification – Theory for the User," Upper Saddle River, N. J, Prentice Hall, 1999

[32] Anderson, J., "Nonlinear System Identification Using a Hammerstein Model and a Cumulant-Based Steiglitz-Mcbride Algorithm," Proceedings of ICASSP: pp429-432, 1994

[33] Astrom, K. J., "Adaptive Control," Addison-Wesley, 1995

[34] Ronald, P., "Discrete-Time Dynamic Models," New York, Oxford, Oxford University Press, 1999

[35] Khalil, H., "Nonlinear Systems," Prentice Hall, 2001

[36] Narendra, K. S. and Gallman, P. G., "An iterative method for the identification of nonlinear systems using a Hammerstein model," IEEE Transactions on Automatic Control, 11:546-560, July 1966

[37] Chang F. H. I. and Luus, R., "A noniterative method for identification using Hammerstein model," IEEE Transactions on Automatic control, 5: 464-468, August 1971

[38] Zhu, Y., "Identification of Hammerstein models for control," Proceedings of the 37th IEEE Conference on Decision and Control, 2:219-220, December 1998

[39] Rangan, S, Wolodkin, G and Poolla, K., "New results for Hammerstein system identification," Proceedings of the 34th Conference on Decision and Control, 697-702, December 1995

[40] Daniel-Berhe, S. and Unbehauen, H., "Identification of nonlinear continuous-time Hammerstein model via HMF-method," Proceedings of the 36th Conference on Decision and Control, 2990-2995, December 1997

[41] Billings, S. A. and Fakhouri, S., "Nonlinear system identification using the Hammerstein model," International Journal of System Sciences, 10 (5): 567-578, May 1979

[42] Li, H. X., "Identification of Hammerstein models using Genetic algorithms," IEEE Proceedings, Part D, 146(6): 499-504, November 1999

[43] Kristinsson, K. and Dumont, "System identification and control using genetic algorithms," Systems, Man and Cybernetics, IEEE Transactions, 22(5): 1033-1046, September 1992

[44] Hatanaka, T. and Uosaki, K. "Hammerstein model identification method based on genetic programming," Evolutionary Computation, 2001, Proceedings of the 2001 Congress, 2(2):1430 – 1435, 2001

[45] Tobin, H., Pelt, V. and Bernstein, D. S., "Nonlinear system identification using Hammerstein and nonlinear feedback models with Piecewise Linear Static Maps-part 1 and 2," Proceedings of the American Control of Conference, i:225-229, ii:235-239, June 2000

[46] Ngia, L. S. H., "Separable nonlinear least-squares methods for on-line estimation pf neural nets Hammerstein models," Proceedings of the 2000 IEEE Signal Processing Society Workshop, 1: 65-74, December 2000

[47] Eskinat, E. and Johnson, S. H., "Use of Hammerstein models in identification of nonlinear systems," AICHE Journal, 37(2): 255-268, February 1991

[48] Al-Amer, S. H. and Al-Sunni, F. M., "Identification of Hammersein models to minimize the H_∞ norm of the mismatch error," 4th Asian Control Conference, Singapore, 704-707, September 2002

[49] Voros, J., "Iterative algorithm for parameter identification of Hammerstein sysgtems with two-segment nonlinearities," IEEE Transactions on Automatic Control, 44: 2145-2149, November 1999

[50] Gomez, J. C. and Baeyens, E., "Identification of multivariable Hammerstein systems using rational orthonormal bases, "Proceedings of the 39th IEEE Conference on Decision and Control, 3: 2849-2854, December 2000

[51] Liu, W., Sun, L. and Sano, A., "Least squares identification of Hammerstein model based on over-sampling scheme," UKACC International Conference on Control, 240-245, September 1996

[52] Lu, C. L., Yuan, T. Q. and Liu, W. J., "Identification of Hammerstein model based on dynamical separation technology," Proceedings of the 3rd World Congress on Intelligent Control and Automation, 3: 2124-2128, June 2000

[53] Westwick, D. T. and Kearney, R. E., "Identification of Hammerstein models of the stretch reflex EMG using separable least squares," Proceedings of the 22nd Annual EMBS International Conference, 2: 1901-1905, July 2000

[54] Belforte, B. and Gay, P., "Optimal input design for Hammerstein (FIR) model identification with unknown but bounded errors," Proceedings of the 38th Conference on Decision and Control, 2: 590-591, December 1999

[55] Giarre, L. and Zappa, G., "Conditional center computation in the identification of approximated Hammerstein models," Proceedings of the 40th IEEE Conference on Decision and Control, 3: 3491-3496, December 2001

[56] Karim, M. N. Al-Duwaish, H. and Chandrasekar, V., "Hammerstein model identification by multilayer feedforward neural networks," International Journal of Systems Science, 28(1): 48-54, 1997

[57] Kokotovic, P., "Constructive Nonlinear Control: A Historical Perspective," Automatica 37: 637-662, 2001

[58] Beorge, B. (1994). Time Series Analysis: Forecasting & Control, Prentice Hall

[59] Liu, Y. S. and Li, X. Y., "Decentralized robust adaptive control of nonlinear systems with unmodeled dynamics," IEEE Transactions on Automatic Control 47(5): 848-856, 2002

[60] Liu, Y. S., and Li, X. Y., "Robust adaptive control of nonlinear systems represented by input-output models," IEEE Transactions on Automatic Control 48(6): 1041-1045, 2003

[61] Liu, Y. S., and Li, X. Y., "Robust adaptive control of nonlinear systems with unmodeled dynamis," IEE Proceedings-Control Theory and Applications 151(1): 83-88, 2004

[62] Zhou, K., "Essentials of Robust Control," Englewood Cliffs, NJ, Prentice Hall, 1998

[63] Rossiter, J., "Model-Based Predictive Control: A Practical Approach," CRC, 2003

[64] Zhang, Y. M. and Li, L. "Interval model based robust control of weld joint penetration," ASME Journal of Manufacturing Science and Engineering 121(3): 425-433, 1999

[65] Levine, W., "Control Handbook", CRC press, 1996

[66] Zhang, Y. M. K., R., "Neurofuzzy Model-based Predictive Control of Weld Fusion Zone Geometry," IEEE Transaction on Fuzzy Systems 6(3): 389-401, 1998

[67] Pan, M. C. and Lettington, A. H., "Smoothing images by a probability filter," Intelligence and Systems, in the Proceedings of IEEE International Joint Symposia, 343-346, 1998

[68] Leu, J. G., "Image smoothing based on pixel grouping," International Conference on Image Processing, ICIP'95, 2: 2535-2541, 1995

[69] Gonzales, R. C. and Woods, R. E., "Digital image processing," 2nd Edition, Englewood Cliffs, NJ, Prentice Hall, 2002

[70] Pratt, W, "Digital image processing," 2nd Edition, NY, John Wiley & Sons, 1991

[71] Wang L. and Wang, W, "A comparative performance study of thresholding algorithm for particle images," In Proceedings of ICSP 2000: 2097-2102, 2000

[72] Lievers, W. B. and Pilkey, A. K., "An evaluation of global thresholding techniques for the automatic image segmentation of automotive aluminum sheet alloys," Material Science and Engineering A, 381: 134-142, 2004

[73] Davis, L., "A survey of edge detection techniques," Computer Graphics Image Process, 4: pp. 248-270, 1975

[74] Zhang, Y. J., "A survey on evaluation methods for image segmentation," Pattern Recognition, 29: 1335-1346, 1996

[75] Ziou, D. and Tabbone, S., "Edge detection techniques, an overview," International Journal of Pattern Recognition and Image Analysis, 8: 537-559, 1998

[76] Marr, D. and Hildreth, E., "Theory of edge-detection," In Proceedings of the Royal Society of London: Series B, 207: 187-217, 1980

[77] A. S. Carasso, A. S., "Linear and nonlinear image deblurring: a documented study," SIAM Journal on Numerical Analysis, 36(6): 1659-1680, 1999

[78] Gonzalez, R. C., Woods, R. E. and Eddins, S. L., "Digital image processing using Matlab," Pearson Prentice Hall, Saddle River, NJ, 2004

[79] Williamson M. and Neureuther, A., "Utilizing maximum likelihood deblurring algorithm to recover high frequency components of scanning electron microscopy images," Journal of vacuum Science and Technology B, 22(2): 1071-1023, March 2004

[80] Mehmet, S., "Survey over image thresholding technique and quantitative performance evaluation," Journal of Electronic Imaging, 13(1): 146-165, January 2004

[81] Graham, L., Y. Chen, Y., Kalyan, T., Joie, T. N. H. and Mian, L., "Comparison of some thresholding algorithms for text/background segmentation in difficult document images," In Proceedings of the Seventh International Conference on Document Analysis and Recognition, ICDAR 2003

[82] Nagy, J. G., Palmer, K. and Perrone, L., "Iterative methods for image deblurring: a Matlab object-oriented approach," Numerical Algorithms, 36: 73-93, 2004

[83] Perez, A. and Pavlidis, T., "an Iterative thresholding algorithm for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9: 742-75, 1987

[84] OTSU, N., "A threshold selection method from gray-level histograms," IEEE Transactions on Systems, Man, and Cybernetics, 9(1): 62-66, 1979

[85] Chang, Y. Fu, A. M. N., Yan, H. and Zhao, M., "Efficient two-level image thresholding method based on Bayesian formulation and the maximum entropy principle," Journal of Optical Engineering, 41(10): 2487-2498, October 2002

[86] Cheng, H. D., Chen, J. R. and Sun, Y., "Threshold selection based on fuzzy c-partition entropy approach," Pattern Recognition, 31(7): 857-870, 1998

[87] Cheng, S. C. and Tsai, W. H., "A neural network approach of the moment-preserving technique and its application to thresholding," IEEE Transactions on Communicaiton, 39: 1549-1558, 1991

[88] Basu, M., "Gaussian-based edge-detection methods-a survey,", IEEE transactions on Systems, Man, and Cybernetic, Part C: Application and Reviews, 32(3):390-396 August 2002

[89] Canny, J., "A computational approach to edge-detection," IEEE transactions on Pattern Analysis and Machine Intelligence, 8: 679-698, 1986

[90] Berzins, V., "Accuracy of Laplacian edge detectors," Computer Vision, Graphics and Image Processing, Vol. 27, pp. 195-210, 1984

[91] Mallat, S., "A wavelet tour of signal processing," 2nd Edition, Academic Press, 1999

[92] Daubechies, I., "Ten lectures on Wavelets," Society for Industrial and Applied Mathematics, 1992

[93] Evelyn, B. and Michael, W., "Edge detection using wavelets," ACM SE '06, Melbourne, Florida, USA, March 10-12, 2006

[94] The Mathworks Inc., "Image processing toolbox user's guide," The Mathworks Inc., Natick, MA, 2007

[95] National Instruments Corporation, "IMAQ Vision concept manual," The National Instruments Corporation, TX, 2007

[96] L. Ljung, "System Identification Toolbox User Guide," The Mathworks Inc., 1997.

[97] Kerschen, G., Worden, K., Vakakis, A. F. and Golinval, J. C., "Past, present, and future of nonlinear system identification in structural dynamics," Mechanical Systems and Signal Processing, 20: 505-592, 2006.

[98] Kowalczuk, Z. and Kozlowski, J., "Continuous-time approaches to identification of continuous-time systems", Automatica, 36: 1229-1236, 2000.

[99] Wang, L. and Gawthrop, P., "On the estimation of continuous time transfer functions," International Journal of Control, 74: 889-904, 2001

[100] Young, P. C., "Parameter estimation for continuous time models-a survey," Automatica, 17: 23-28, 1981.

[101] Wlodzimierz, G., "Recursive identification of continuous-time wiener systems," International Journal of Control, 72: 981-989, 1999.

[102] Bryon, M. R., Francis, D. J., Babatunde, O. A. and Ronald, P. K., "Nonlinear model predictive control scheme using second order volterra models," American Control Conference, 3: 3253-3257, June 1994

[103] Rao, G. P. and Unbehauen, H., "Identification of continuous-time systems," Control Theory and Applications, 153: 185-220, March 2006.

[104] Zhang, Y. M. and Liu, Y.C., "Control of dynamic keyhole process," Automatica, 43(5): 876-884, 2007.

[105] Zhang, J and Walcott, B. L., "Adaptive interval model control of arc welding process," IEEE Transactions on Control Systems Technology, 14(6): 1127-1134, 2006.

[106] Dorf, R. C. and Bishop, R. H., "Modern Control Systems," 7th Edition, Addison Wesley, 1995.

[107] Yang, W. Y., Cao, W. W. Chung, T. S. and Morris, J., "Applied numerical methods of MATLAB," pp. 321-323, Wiley, June 2005

[108] Haber, R. and Keviczky, L., "Nonlinear System Identification – Input-Output Approach: Volume 1: Nonlinear System Parameter Identification," First Edition, Springer, 2000.

VITA

Born in 1972 in Anshan, Liaoning , P.R.China

Education:

Harbin Institute of Technology, Bachelor of Engineering, Mechanical Engineering, 1990-1994

Harbin Institute of Technology, Master of Science, Mechatronics, 1997-1999

Eastern Kentucky University, Master of Science, Software Engineering, 2001-2003

Professional Positions:

Angang Group, Anshan, Liaoning, P. R. China, Assistant Engineer, 1994-1997

Xiaodong Na

_____

January 19, 2008

_____