Modeling and Cycle-to-Cycle Control of the Angioplasty Balloon Forming Process

Yan Chen

Department of Electrical and Computer Engineering
Centre for Intelligent Machines – Systems and Control Group
Industrial Automation Lab
McGill University, Montreal, Canada,
February 2008

A thesis presented to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical and Computer Engineering

Your file Votre référence
ISBN: 978-0-494-51454-2
Our file Notre référence
ISBN: 978-0-494-51454-2

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis.

I understand that my thesis may be made electronically available to the public.

Abstract

The development of a new angioplasty balloon is a time consuming process. This thesis aims at reducing the amount of time and materials spent on the experimental stage of the development of new angioplasty balloons. This can be achieved by building a nonlinear neural network model of the balloon forming process and implementing an off-line cycle-to-cycle controller. The controller can learn from the previous experiments and provide better input parameters for improving the quality of the next balloons formed in the process. It is shown in the experimental test results that the neural network model can provide accurate estimates of the process outputs. The neural network model combined with a cycle-to-cycle control strategy has the potential to replace the trial-and-error approach to balloon development that is commonly applied today.

Résumé

Le développement d'un nouveau ballon pour l'angioplastie est un processus qui prend beaucoup de temps. Cette thèse a pour but de réduire le temps et la quantité de matière utilisés pour le stade expérimental du développement d'un nouveau ballon d'angioplastie. Ceci peut être fait en construisant un modèle nonlinéaire du procédé de formage du ballon à l'aide d'un réseau neuronique et en implantant une loi de commande cycle à cycle sur le modèle. Le contrôleur peut apprendre à partir des expériences passées et fournir de meilleurs paramètres d'entrée pour améliorer la qualité des ballons qui seront formés subséquemment dans le procédé. On montre par des résultats expérimentaux que le modèle à réseau neuronique peut fournir des estimations précises des sorties du procédé. Le modèle à réseau neuronique combiné à une stratégie de commande cycle à cycle a le potentiel pour remplacer l'approche courante par essai erreur pour le développement de nouveaux ballons d'angioplastie.

Acknowledgements

First and foremost, I would like to express my most sincere appreciation and gratitude to my supervisor, Dr. Benoit Boulet, for his valuable guidance and encouragement during these two years in my master's study and in the preparation of this manuscript. Without his help, I could never complete the work presented in this thesis.

I would also like to thank the CNRC-NRC IMI for allowing me to use their equipment. I am indebted to many IMI employees. I appreciated Dr. Robert Di Raddo's guidance, experience, and sugguestions on how to perform the experiments. A big thank you goes to Christian De Grandpré for always being there in the laboratory to answer my questions. I would also acknowledge Zoe Sarrat-Cave, Kajsa Duke, Nushi Choudhury for their excellent work on angioplasty balloons, which is referenced in this thesis.

I am especially grateful to Dominic Lalli, Alexandre Boyer, Yingxuan Duan, Yingfeng Yu from McGill University for their help in various technical aspects of my research.

Finally, I want to dedicate this work to my parents. Words can never be enough to express my gratitude to their love, support and understanding. Especially, I would like to thank my father for always being there for me and guiding me through.

Table of Contents

List of Figures

List of Equations

List of Tables

Chapter 1
Introduction

Heart is one of the most essential muscles in the human body. It pumps the blood to the lungs and to the rest of the body. Oxygenated blood, which is carried to the heart muscle by coronary arteries, is needed in the circulatory system. Therefore, it is important to maintain the cardiac and coronary artery function. But heart disease has now become one of the leading diseases all over the world. There are several causes of the heart diseases. One of them is due to the reduced amount of blood flow in the circulatory system, which results in coronary artery diseases (CAD). Millions of people from all over the world now suffer from coronary artery diseases due to artery blockage or narrowing, as shown in Figure 1-1 [1]. Much research is conducted to find treatments for CAD. Balloon angioplasty has become a standard and practical method for the treatment of the coronary artery diseases since the early 1980's. Balloon angioplasty, also known as percutaneous transluminal coronary angioplasty (PTCA), is a catheter-based technique used to open clogged arteries [2]. It has been proven to be a safer and more efficient treatment for coronary artery diseases than other methods, such as bypass surgery.

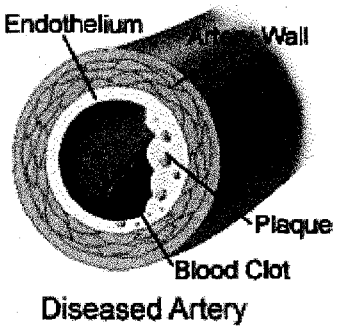


Figure 1-1 The Diseased Coronary Artery (Permission to print granted by Image copyright Texas Heart Institute, www.texasheart.org)

1.1 Thesis Objective

Nowadays, trial-and-error is the method commonly used for the development of an angioplasty balloon both in the experimental stage and in the industrial production of the balloon. With the trial-and-error method, an unpredicted number of experiments needs to be done in order to find the best process input parameters. Even the manufacturing companies have not found a better way to produce the balloons. For the trial-and-error method, tests are conducted by heuristically varying the input parameters and measuring the outputs, which depends greatly on the operators' knowledge and experience of the process. It typically takes several months, or sometimes even years, to develop an angioplasty balloon without defect. It is shown in [3] that, with the aid of an iterative learning control algorithm, the amount of time invested in the experimental development stages can be greatly reduced. However, based on the linear model of the angioplasty balloon forming process [3], iterative learning control can provide satisfactory results for one batch of tubing but fail on another batch. Taking more process input parameters into consideration, a more efficient method including a nonlinear model will be introduced in this thesis, which can help reduce the amount of time spent on the experimental stage and provide a set of input parameters based on three batches of tubing if the desired output parameters are given. The approach in this thesis involves building a nonlinear model of the balloon forming process and implementing a cycle-to-cycle controller which can provide a set of input parameters to start with, and reduce the number of experiments be performed. In addition, the approach in this thesis makes good use of the previous experimental data by implementing an off-line learning algorithm.

Thus, this thesis aims at building a nonlinear model based on the previous experimental results with the input parameters and the output parameters of the balloon forming machine and applying an off-line learning controller to find improved input parameters if the desired output parameters are given.

1.2 Benefits to the Industry

The design of a nonlinear model and the implementation of an off-line controller on the angioplasty balloon forming machine will contribute to the development and production of the angioplasty balloon in many ways. With a trial-an-error method, the operator must vary the setting parameters on the balloon forming machine by experience after each cycle until the produced balloon meets the required specifications such as surface quality, minimal wall-thickness and high burst pressure [3]. This procedure is considered to be experience demanding and time consuming. After the off-line controller is applied, it gives the user the suggested input parameters with the desired output parameters.

1.3 Background Review

1.3.1 Coronary artery disease (CAD)

Heart diseases has become the leading cause of death in the U.S. and also are a major cause of disability. Coronary artery disease affects almost 1.3 million Americans [1], making it the most common type of heart disease. CAD is an illness that occurs when the coronary arteries are narrowed or blocked due to the accumulation of cholesterol and other material, called plaque, on their inner walls [4]. As the plaque deposits grow in size, the coronary arteries becomes narrowed and clogged and not enough blood can flow through the arteries. As a result, the blood flow to the heart muscle is reduced, and the

heart muscle cannot get enough oxygenated blood. At first, this reduced blood flow may not cause obvious symptoms. But as the plaque deposits increase in size, it may result in angina, shortness of breath or heart attack. Angina is chest pain or discomfort that occurs due to the lack of blood [4]. And most heart attacks happen when a blood clot develops at the site of plaque in a coronary artery and suddenly cuts off most or all the hearts' blood supply, which can cause permanent damage to the heart muscle. Over time, CAD can weaken the heart muscle and lead to heart failure or arrhythmias. Heart failure means the heart can't pump blood effectively to the rest of the body while arrhythmias are changes in the normal beating rhythm of the heart and it can be quite serious in some cases [4]. In the Figure 1-2 [5], it shows the difference among the healthy coronary arteries, the blocked coronary arteries and the coronary arteries after the balloon angioplasty.

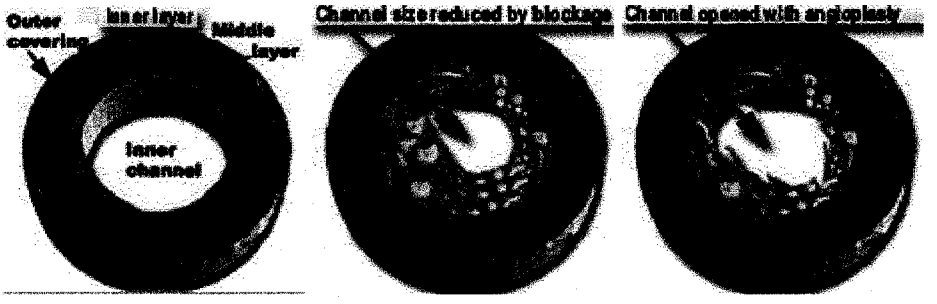


Figure 1-2 Coronary Arteries (Permission to print granted by Landhort and Dr. Abdulla M. Abdulla)

1.3.2 Factors causing CAD and Prevention

It is important to maintain the health of coronary arteries. And it is always necessary to know the main causes to CAD and how to prevent them. There are many things one could do to reduce the risk of getting a CAD.

It has been identified by clinical and statistical studies that several factors could contribute to the increase of the chances of getting CAD and heart attacks. CAD usually begins with damage or injury to the inner layer of a coronary artery. And this damage could be caused by many factors, some of which could be treated or controlled, for instance, by changing the lifestyle. These factors that could be controlled are smoking, high blood pressure, high cholesterol, certain diseases as diabetes, and radiation therapy to the chest, etc. The risk of getting CAD could be reduced by changing lifestyles or taking medicine. For instance, one should maintain a healthy lifestyle by quitting smoking, keeping track of the blood pressure and the cholesterol levels and getting them under control, exercising regularly and getting enough physical activity, eating a healthy diet and keeping a healthy weight, etc. Besides, medicine could also be an additional treatment and relieve the symptoms of CAD, and the commonly used drugs are cholesterol medicine, aspirin, Beta blocker, etc. [4].

However, there are also some major risk factors that are out of control, such as the increase of age, the gender and the heredity (including race). It is shown that over 83% of the people who die of coronary heart disease are 65 or older, men have a greater risk of heart attack than women do, and children are more likely to develop heart diseases if their parents have it. In this situation, more aggressive treatments are needed to restore and

promote the blood flow if the change of lifestyle or medicine does not improve or the symptoms are getting worse. One of the treatments is called coronary artery bypass surgery. During this procedure, the doctor creates a graft to bypass clogged coronary arteries using a vessel from another part of your body, which allows the blood to flow around the clogged coronary artery. Bypass surgery can improve blood flow to the heart, relieve chest pain, and possibly prevent a heart attack [4]. But it requires an open heart surgery since the doctors need to open the patient's chest to reroute the vessels. So it is most often reserved for cases of multiple narrowed coronary arteries. [4] And another commonly used treatment is called balloon angioplasty, which is less invasive.

1.3.3 History of Balloon Angioplasty

Balloon angioplasty, also known as percutaneous transluminal coronary angioplasty (PTCA), is a widely used catheter-based technique for opening clogged arteries [2]. In 1929, Werner Forssmann, a 25 year-old German physician, performed the first operation on himself to enter the heart with a catheter, for which he was awarded the Nobel Prize in 1956. Later on, Dr. Porstmann developed a balloon catheter to open the iliac artery in 1973. And Dr. Andreas Gruentzig had bee n working on a thin and flexible balloon catheter to open a clogged coronary artery throughout 1970s. In 1977, he successfully performed the first balloon angioplasty. Then he brought his technology to the United States in 1980 when he immigrated to the United States, where he taught others his techniques for performing the balloon angioplasty. It has been extensively applied in the U.S. to relieve chest pain and prevent heart attacks for CAD since then [6]. It has been considered to be relatively inexpensive and commonly used in diagnosing and treating

CAD. Around 500,000 patients took the angioplasty treatment per year in the late 1900s [6]. The number has increased dramatically to 2 million worldwide in 2003 [3].

1.3.4 How does Balloon Angioplasty work

During a balloon angioplasty, the Interventional cardiologist uses a catheter to insert into the patient's artery, as shown in Figure 1-3 [8]. A small balloon is located at the tip of the catheter and inflated to pass through the artery and push the plaque back against the artery wall and clear up the blockage. Usually, a stent, as in shown in Figure 1-4 [1], which is a small metal tube-shaped device, is left in the artery when angioplasty is performed. When a stent is placed into the artery, it acts as a support to keep the artery open [1, 2, and 7].

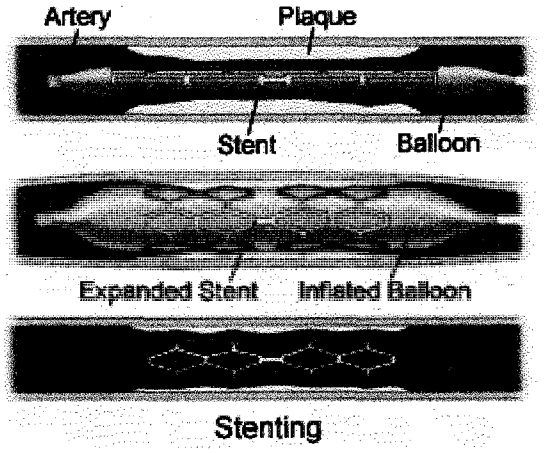


Figure 1-3 Stent (Permission granted by Image copyright Texas Heart Institute, www.texasheart.org)

1.3.5 The future of Balloon Angioplasty

Compared with the bypass surgery, the balloon angioplasty is a relatively more simple and safe procedure. There is, however, some drawback that exists, which is that some 30-50% of patients taking the balloon angioplasty procedure need to repeat it because their arteries clog again. The angioplasty may not succeed in removing the blockage. Even if

relieved, the plague might be built up again. The initial clogging is known as stenosis in the academic terms, and it is called restenosis when it happens after angioplasty. At the beginning of the 21st century, most research into angioplasty focused on the prevention of restenosis. Some angioplasty balloons are equipped with medicine as heparin, to prevent the inner arterial plaque accumulation. Such drugs are routinely given to patients after the procedure, but with coated balloons to be delivered directly to the artery. Some surgeons are also experimenting with a stent, which can be placed in the artery during angioplasty to prevent it narrowing again. Stents are small metal tube-shaped devices that may be either stainless steel or some kind of flexible steel mesh. At preset, the newest angioplasty technology is to combine the balloon with the stent for the best results for both the patient and the surgeon [8].

The goal of balloon angioplasty is to promote the blood to go through the artery by pushing the plaque against the wall of the coronary artery, which are the arteries that supply blood to the heart muscle [2].

Chapter 2
Background Review

Medical balloons are made into a variety of shapes and sizes for different purposes and applications. The size of angioplasty balloons may vary from 1 to 25 millimeters in diameter and the double-wall thickness varies from 0.03 to 0.05 mm [3]. This thesis studies at the forming process of 3 x 15 mm angioplasty balloons. The specifications of the balloon size will be given in the following section. Figure 2-1 shows the length of each part for a 3 x 15 mm angioplasty balloon.

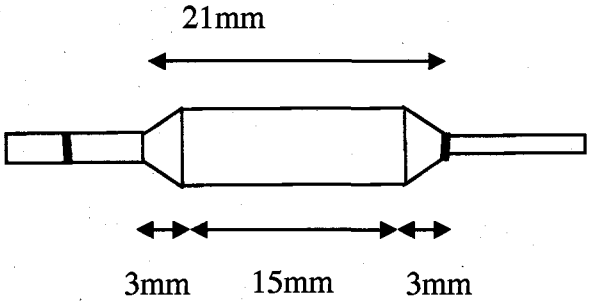


Figure 2-1 The Size of Each Part of a 3 x 15 mm Angioplasty Balloon

2.1 Angioplasty Balloon Description

Basically, an angioplasty balloon is made of five parts consisting of the proximal neck, the proximal cone, the body, the distal cone and the distal neck, as shown in Figure 2-2 [9]. Key characteristics for the balloons are the balloon diameter, the balloon length, the burst pressure, the balloon profile and the balloon compliance.

The extreme ends of the balloon are called the necks, which are usually different in size. And the proximal neck is the one with relatively larger diameter while the distal neck is the end with smaller diameter. The body is the smooth cylindrical part of the main body

of the balloon. The cones are the conical parts of the balloons with the proximal cone connecting the proximal neck to the body while the distal cone connecting the distal neck to the body [3].

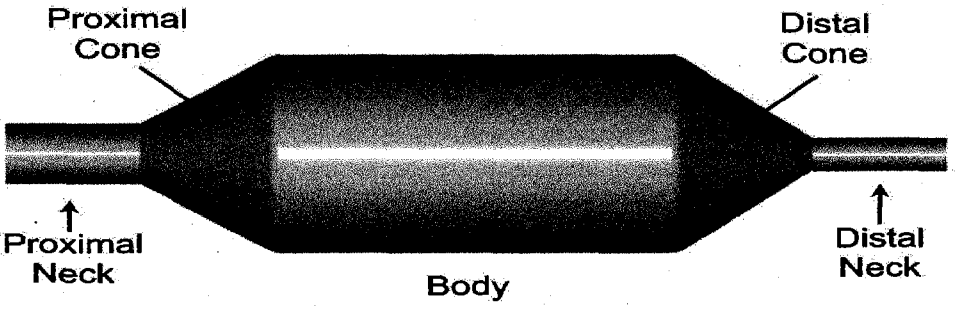


Figure 2-2 A Typical Angioplasty Balloon (Permission to print granted by Advanced Polymers, Inc)

The angioplasty balloon with size 3 x 15 mm means that the body of the balloon is 3 mm in diameter and 15 mm in length, which is shown in Figure 2-1.

2.2 Different types of Medical Balloons

There are two basic types of balloons used in the medical industry and the two share few similarities [9].

2.2.1 High-pressure medical balloons

The first type of balloon is the high-pressure, non-elastic balloon used for applying pressure and angioplasty balloons fall into this category. High-pressure balloons are made from non-compliant or low-compliant materials. Compliance describes how much a balloon can expand compared with its original size when pressure is applied. A balloon that can expand several times its original size is said to be more compliant than the one that barely changes. High-pressure balloons maintain their original size and shape even

under high pressure. An important characteristic of angioplasty balloons is to be able to retain their original size and shape with high pressures without much stretching of the balloon walls.

2.2.2 Low pressure medical balloons

The other medical balloon type is the low-pressure, elastomeric balloon made of resilient materials like latex or silicon and mainly used in fixation and occlusion. Different from the high-pressure balloons, low-pressure balloons are more compliant because they usually expand to several times their original size. The low-pressure balloons usually go back to their original size once the pressure is reduced.

2.3 Raw materials for angioplasty balloons

The raw materials used in the manufacture of angioplasty balloon tubing are essential to balloon development because strength and flexibility are two of the key requirements. Several plastic materials have been used to satisfy these requirements [8, 9]. In the 1970s, the first angioplasty balloons used by Dr. Gruentzig were made of flexible polyvinyl chloride (PVC). But it cannot be used in industry because the balloons made from PVC have thick in the walls and have relatively low rated burst pressure. Later on, cross-linked polyethylene was used instead of PVC in the balloon development. Then, in the late 1980's, nylon became the material of choice for making angioplasty balloons. Polyurethane balloons came out in the 1990's. Now the materials commonly used are PET, Pebax®, or nylon. PET is somewhat stronger than nylon while nylon is more flexible. The material used in this project is Pebax® tubing of grade 7233. Pebax® has been selected because it is more kink resistant during the folding and assembly of the

balloons, has less recoil during sterilization and can be ordered in a range of hardness [9, 10].

Chapter 3
ANGIOPLASTY BALLOON MANUFACTURING PROCESS

3.1 Angioplasty balloon manufacturing Process Overview

Generally speaking, the angioplasty balloon manufacturing process is a complicated one, which mainly consists of an extrusion process, a double end stretching process, and finally a balloon forming process. First, the raw material for making angioplasty balloons needs to be extruded into a tube shape. Then the tube will be double stretched at each end and formed into a balloon.

3.1.1 Extrusion

In order to produce angioplasty balloons for use in application, the first step is to extrude high-quality and highly concentric tubing that meets specific requirements, such as uniform wall thickness, and excellent concentricity.

At first, the raw material is put into a heated barrel. The raw material is mixed into a homogeneous blend by a rotating screw. Then, a gear pump is used to maintain a consistent output volume of the melt plastic, which is pumped through an extruder. The melt plastic comes out of the extruder as a long tube, which is pulled though a cooling bath by a mechanical puller to freeze the tubing dimensions. Finally, a cutter is used to cut the tube to a specified length [6].

The balloons used in this thesis are developed with a polymer compound, Pebax®, which is produced by the Technical Polymers Group of ATOFINA Chemicals Inc. Pebax® could provide the best compromise of a high-rated burst pressure and an average

compliance. This thesis aims at producing the 3 x 15 mm angioplasty balloons. The tubing used to make 3 x 15 mm balloons comes from Innovative Extrusions in California. The tubing specification is shown in the following table 3-1 [3]:

Table 3-1 Tubing Specifications

	Tubing for 3 mm Balloons
COMPANY NAME	Innovative Extrusions
MATERIALS	Pebax®
DUROMETER	72D
DIMENSIONS	0.036'' x 0,00195'' x 6'
QUANTITY	304.8 m
LOT NUMBER	FP030904-001
PURCHASE ORDER	10396

The durometer index value is a specification of the material hardness with the value 'A' meaning softer and value 'D' meaning harder [3].

3.1.2 Double end stretching

The next step after extrusion is double end stretching, which is done on the computerized double end stretcher (DES). An overview of the DES is shown in Figure 3-1 [3, 11]. Figure 3-2 [3, 11] offers a closer view of the DES.

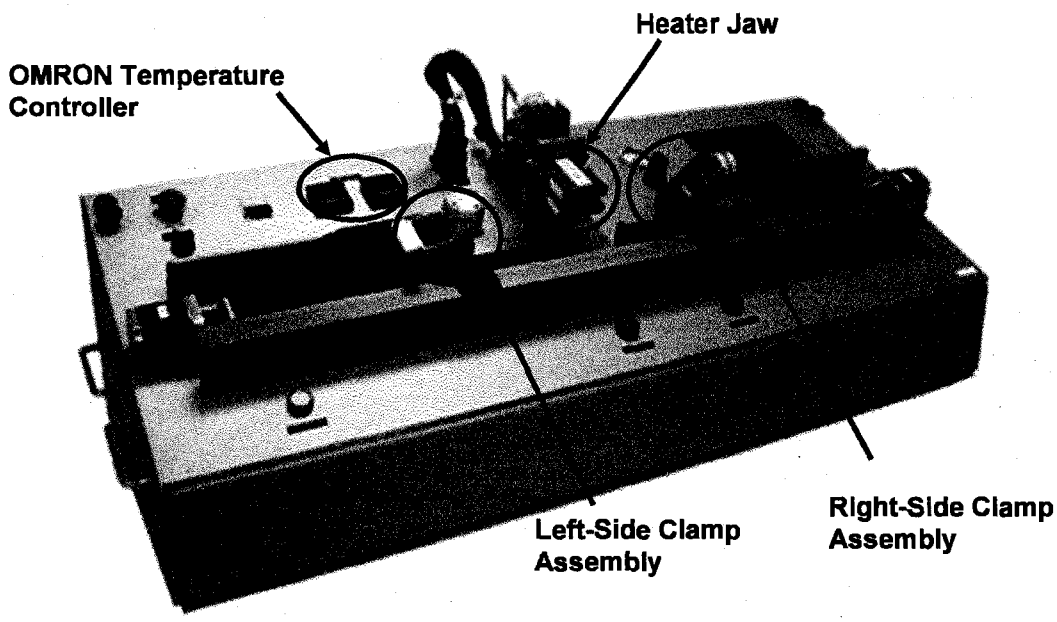


Figure 3-1 The Double end stretcher (Permission to print granted by Interface Associates)

This step is designed to stretch tubing and prepare it for the next balloon forming process. Because the angioplasty balloon requires a very small neck diameter which may be smaller than the raw tubing, diameter reduction is done by stretching the tubing at each end. The DES stretches both ends of the tube and leaves an unstretched length in the middle, called a parison [6] shown in Figure 3-3 [3, 12]. The DES is mainly composed of a heater jaw, a temperature controller, a left-side clamp assembly, a right-side clamp assembly and air cooling vents. It offers a variety of selection of tubing stretching parameters by a touch screen display to input parameters such as stretch speed, temperature, distance and timing.

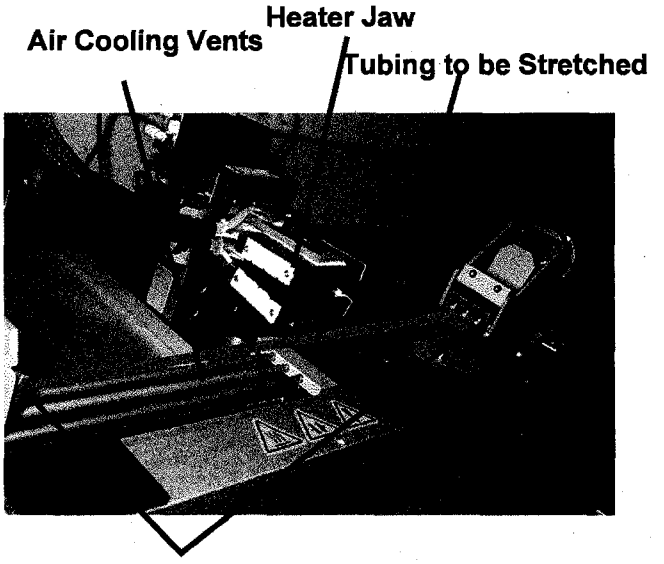


Figure 3-2 DES Components (Permission to print granted by Interface Associates)

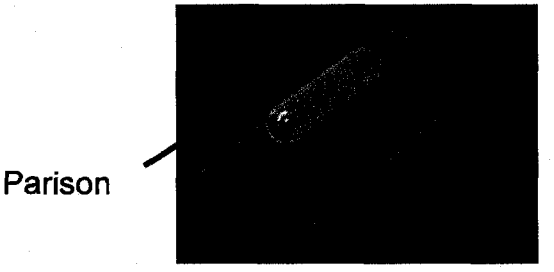


Figure 3-3 Parison

To start with, the machine is initialized to the start position by pressing SAFETY and RESET buttons simultaneously. Then, the tubing is loaded manually and clamped into place. The left-side air-powered clamp is activated automatically. Once this is done, the operator starts the sequence by pressing the SAFETY and START buttons simultaneously. The general stretch sequence is to heat the left side, to stretch the left side, to heat the right side, to stretch the right side and to complete the sequence. The

diagram in Figure 3-4, provided by Zoe Sarrat-Cave, illustrates the double stretch process [3, 13]

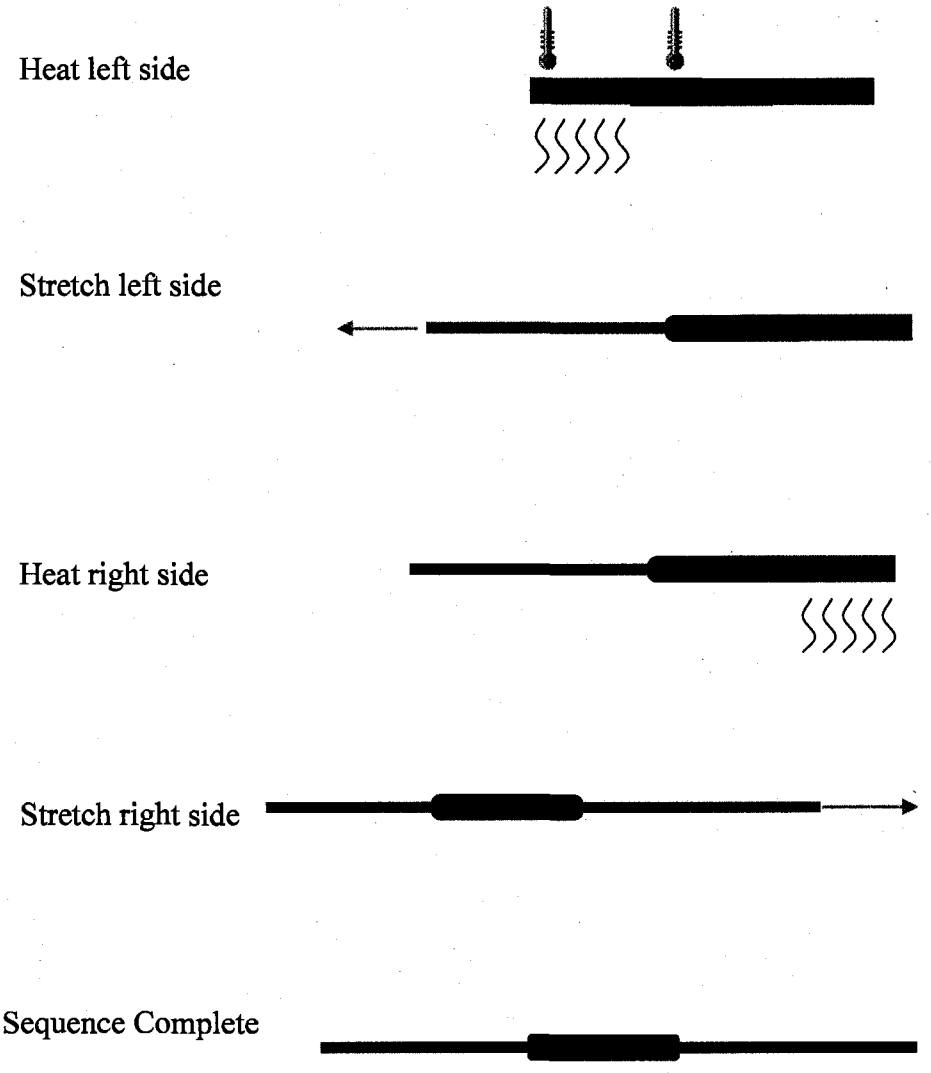


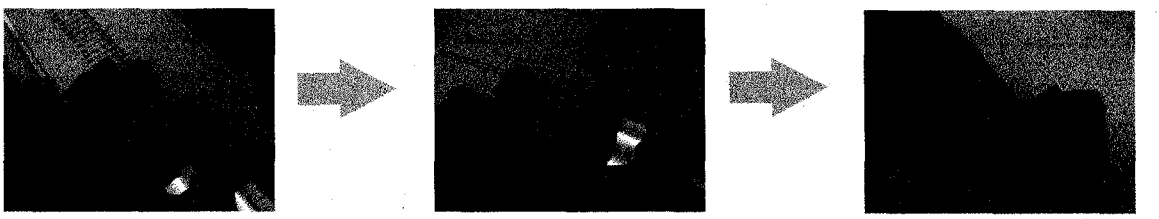
Figure 3-4 A general double end stretch operation sequence

After the machine is initialized and moved to the home position, the right-side clamp assembly moves to the right to feed twice the heater width of tubing (146mm) plus the UNSTRETCHED LENGTH specified by the input parameter setting. Then the left-side clamp is activated and the sequence continues according to the left-side parameter

settings. Heater assembly moves in and its jaw closes. Heating lasts at the set temperature for the amount of time specified by the input parameters. The heating process is controlled by the OMRON temperature controllers installed on the double-end stretcher. The jaw heater opens and the heater assembly retracts away from the tubing. Then, the right-side clamp assembly moves to the left side at the rate of the STRETCH SPEED and STRETCH DISTANCE as specified in the parameter setting, which creates the proximal side of the parison. The left-side assembly moves to the right to the distance according to the RELAX BEFORE COOL setting. Both the left-side and the right-side clamp assembly move synchronously left to position the parison for heating on the right (distal) side. At this time, air is blown across the parison for the specified COOLING TIME. Then, the left-side clamp assembly moves right and the right-side clamp assembly repeats what the left-side has done previously. Finally, when the cooling time has elapsed, the left air clamp releases and the sequences are complete. The operator manually opens the right-side clamp and takes the tubing out of the machine [3, 13].

3.1.3 Preparation for balloon forming process

After creating the parison with DES, the balloon tubing is prepared for the balloon forming. Three steps would be done before the pre-formed tube goes to the balloon forming machine. First, centre the parison on "zero" line of the graduated cutting board with the proximal end on the left as shown in Figure 3-5. Secondly, trim the proximal end to 155 mm. Thirdly, roughen the surface of the proximal end with sand-paper to guarantee the good grip in chuck, which is shown in Figure 3-5 [3].



Centre parison

Cut proximal (left) side to 155mm

Roughen proximal end (~1 inch)

Figure 3-5 Preparations

3.1.4 Balloon forming process

Finally, the balloon is formed through a specialized kind of blow molding process. The final step, the balloon forming process, is accomplished on the balloon forming machine (BFM). The machine used in this project is the computerized hot-mould balloon forming machine 9810H produced by Interface Associates, as shown in Figure 3-6 [3, 11] and a close overview is shown in Figure 3-7 [3, 11]. Figure 3-8 [3, 11] shows the center mold and end plugs for the BFM. Figure 3-9 shows what an angioplasty balloon looks like.

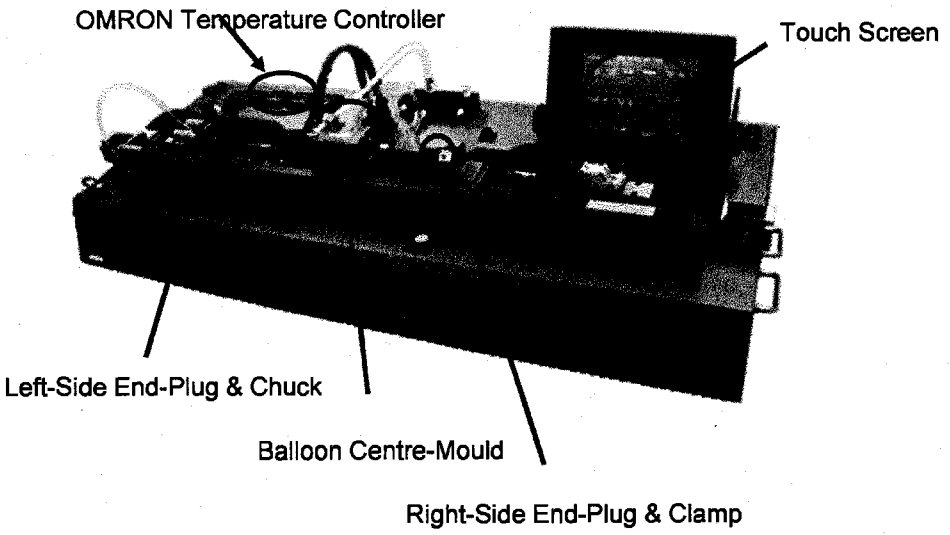


Figure 3-6 Balloon Forming Machine 9810H [12] (Permission to print granted by Interface Associates)

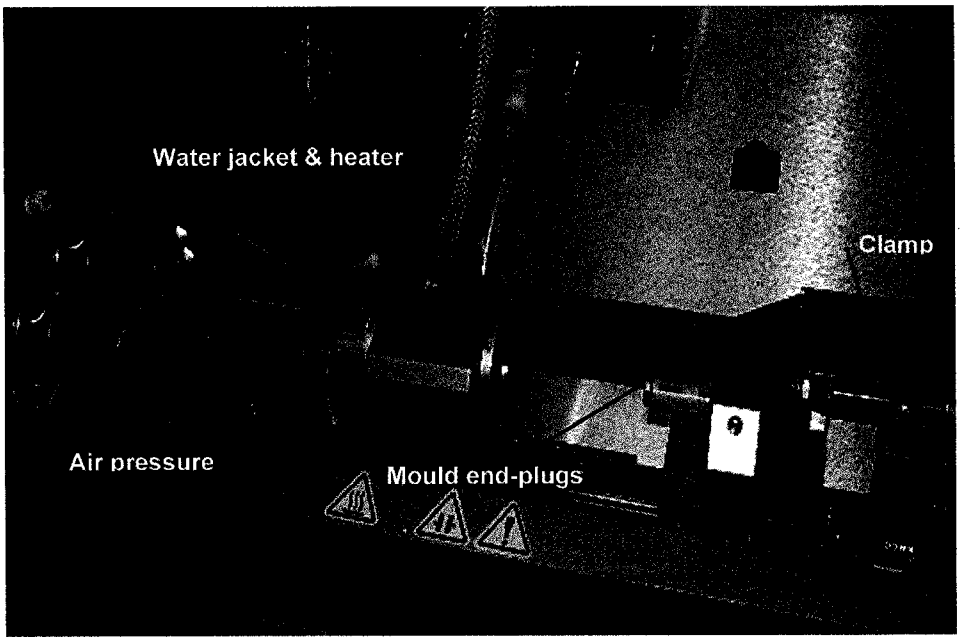


Figure 3-7 A Close View of BFM (Permission to print granted by Interface Associates)

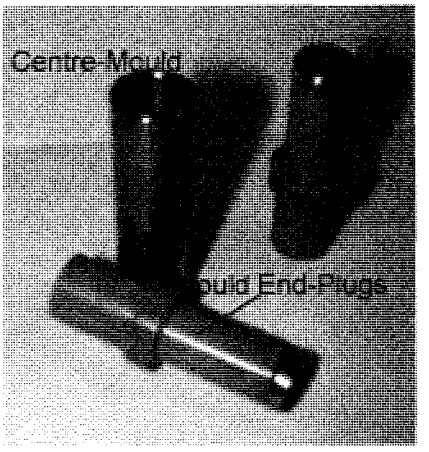


Figure 3-8 Moulds (Permission to print granted by Interface Associates)

Figure 3-9 Angioplasty Balloon

The general forming process, provided by Zoe Sarrat-Cave, is shown in Figure 3-10 [3, 12].

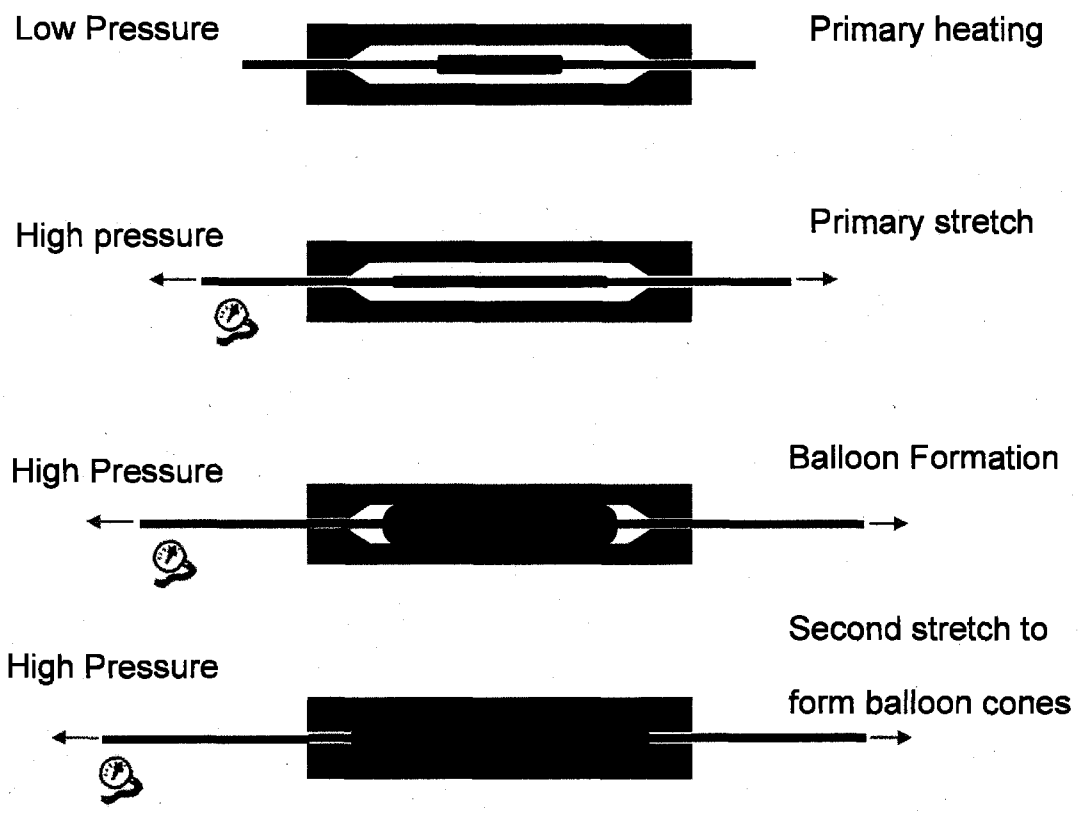


Figure 3-10 General balloon forming process

		STEPS ->	01	02	03	04	05	06	07	08	09	10	11	12
SET VALUES	Heat (C)	Left												
		Center				90	90	90	141	134				
		Right												
	Cool (on/off)	Center									X			
		Plugs									X			
	Purge (on/off)	Center										X		
		Plugs										X		
	Pressure (Bar)	Set To				35						0		
		Fill Rate				10						31		
		Dump										X	X	
		Vacuum												
	Stretch (mm)	Left		15	0	1		12		16				60
		Right		0	0	1		12		16				100
	Speed (mm/s)	Left		80	80	20		130		100				40
		Right		80	80	20		130		100				60
	Force (N)	Left		53.4	53.4	53.4		53.4		100				5
		Right		53.4	53.4	53.4		53.4		100				5
	Clamps Down	Left			X									
		Right			X									
	Mold Closed			X	X	X	X	X	X	X	X	X	X	
STEP EXIT	Heat (C)	Left												
		Center				>80			>120	>130	<35			
		Right												
	Force (N)	Left												
		Right												
	Pressure (Bar)												<2.5	
	Position (mm)	Left												
		Right												
	Time (s)						0.2	2				7		10

Figure 3-11 Overview BFM Inputs (Permission to print granted by Interface Associates)

Figure 3-11 [3, 11] shows the balloon is formed by a computer controlled balloon forming machine though 12 steps. Before a detailed description of the whole balloon forming process is given, a guide to the symbols is shown in Figure 3-12 [3, 12] and the BFM steps are shown in Figure 3-13 [3, 12]. The detailed process [3, 6, 12] is described as follows: the operator actuates the machine and loads the pre-form balloon tube into the centre-mould, which is called a glassform. The glassforms may vary according to the different requirement of the balloons to be produced. The proximal end of the tube is inserted into the chuck and the chuck is manually actuated. This end is connected to a supply of compressed air. The distal end of the tube is fed though the end-plugs and the mould-centre and welded up. Now, the tube is fully loaded in the balloon forming

machine and the forming process can be started. The first step is pre-forming. During this period, the compressed air at the proximal end is turned on, which keeps the pre-form at a constant inner pressure. The heated jaws warm the piece for a defined time (warm-up time) to prepare the plastic for the next step. After the warm-up, the computer controlled system switches to a high pressure with a certain temperature mode that will again be held for a specified period (forming time). And both ends of the tube are axially pre-stretched. This step is called primary heating with high pressure and pre-stretch. Usually, high pressure with the lowest possible temperature will yield good balloons. The next step is the primary stretch. Both ends of the heated, pressurized tube are axially stretched to thin out the material at a specified speed to a specified stretch length. During this time, the heated parison expands due to the pressure and forms the balloon body in the centre-mould. Generally speaking, more stretch and higher stretch speed will yield better balloons with thinner wall-thickness. A secondary stretch usually comes after the primary stretch, which also applies an axial stretch to both ends of the tube to a specified distance at a certain speed. The cones of the balloon are formed by pulling material into end-plugs. Finally, the centre-mould is cooled down by water and the water is purged from the mould with compressed air flow. Now the balloon is ready to be carefully removed from the mould and prepared for quality inspection and evaluation.

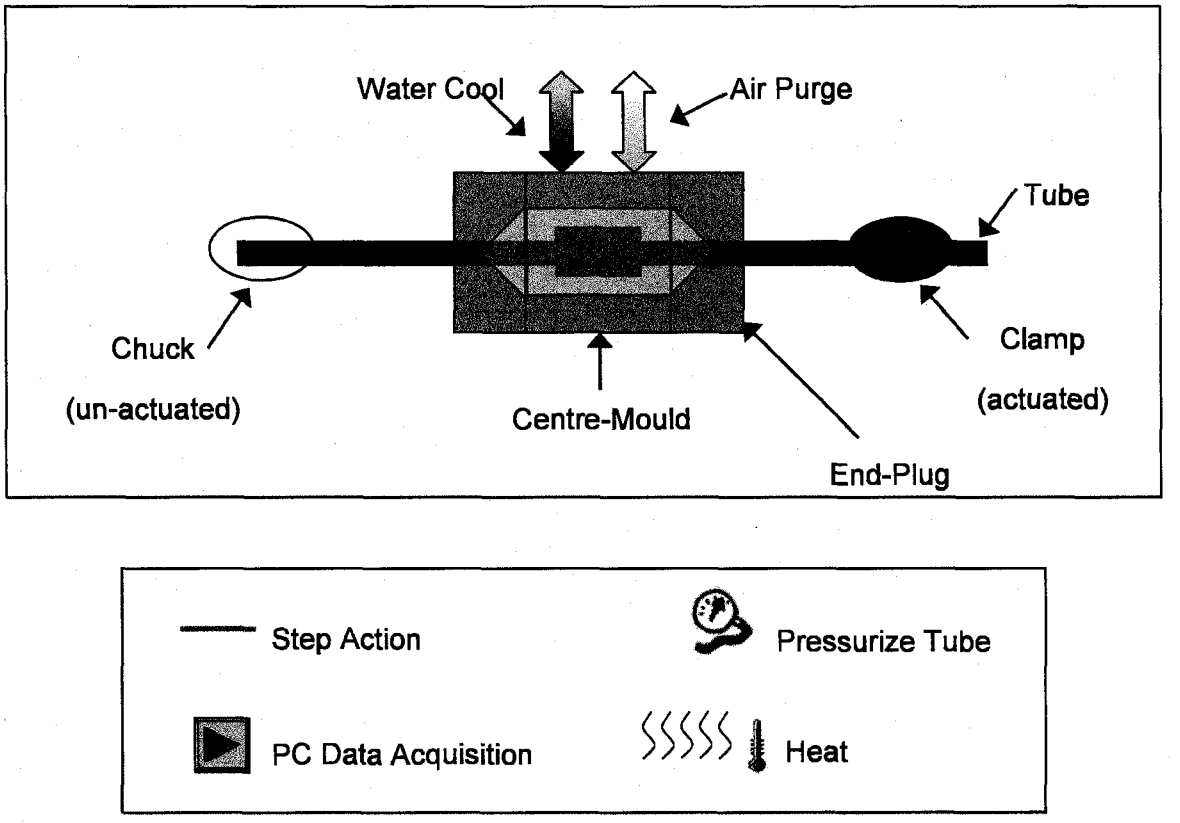
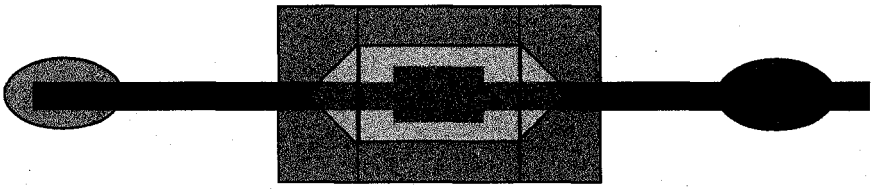
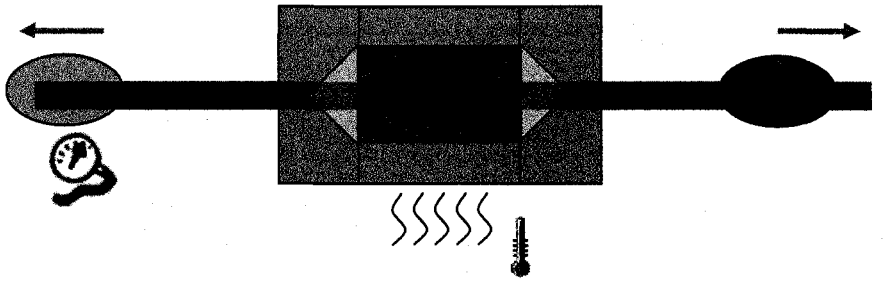


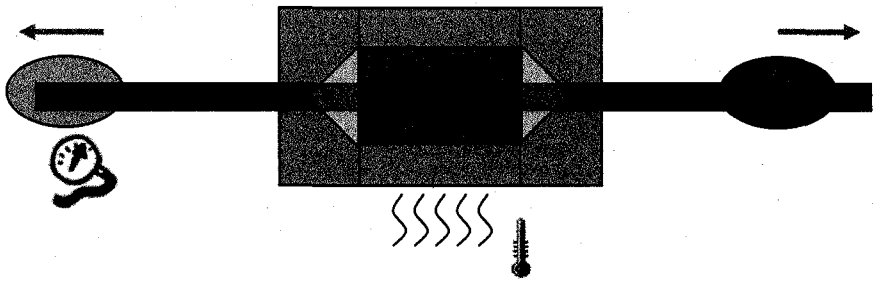
Figure 3-12 Guide to Symbols



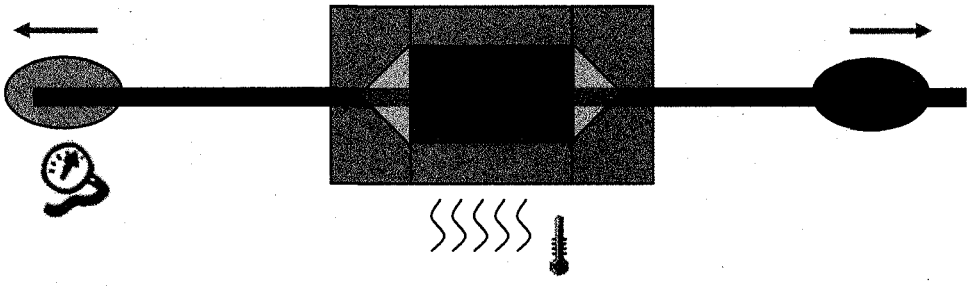
Step 1: tube loaded



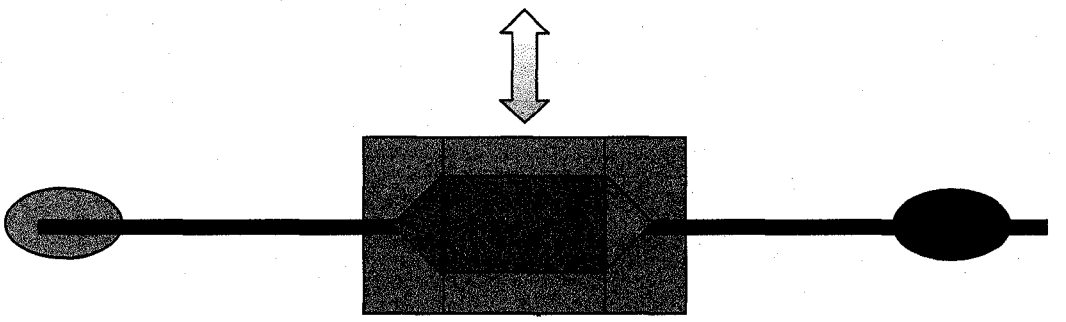
Step 2 & 3: Pre-form, Primary heating with high pressure and pre-stretch



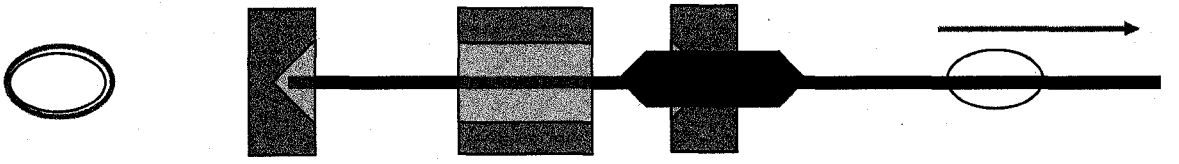
Step 4 & 5: Primary Stretch



Step 5: Secondary Stretch



Step 6: Cooling



Step 7: Balloon Removal

Figure 3-13 Balloon Forming Procedures

3.2 Angioplasty balloon manufacture quality control

Quality control is essential in medical devices. Angioplasty balloons are manufactured one at a time and typically each balloon is evaluated once it is manufactured. In this project, the evaluation of an angioplasty balloon is followed by three steps and four properties are to be inspected in these three steps. The first one is qualitative analysis that is to scale the balloon from score 0 (the best quality) and score 4 (the worst quality) by the physical appearance [3]. The score standard is as given in Table 3-2 [3]. Figure 3-14 below shows the two common defects of an angioplasty balloon.

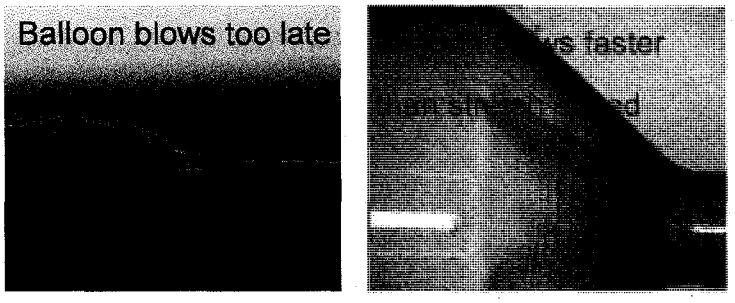


Figure 3-14 The Angioplasty Balloon Common Defects

Table 3-2 Angioplasty Balloon Score Standard

Score	Criteria	Sample
0	**The balloon has no visible defects.**	
1	**The balloon contains minor defects**: crow feet may be visible but do not surpass more that 50% of the length of the cone, no axial lines are visible.	
2	**The balloon contains some major defects**: crow feet surpass more than 50% of the length of the cones, no axial lines are visilbe.	
3	**The balloon contains many major defects**: crow feet are visible in the body of the balloon, and/or one or more axial lines are present.	
4	**No balloon was formed.** The balloon forming sequence did not produce a balloon.	

The wall thickness is an important parameter to decide whether the balloon is acceptable or not. A proper wall thickness can achieve good folding behavior and a small cross section. A thicker wall will lead to a different folded balloon profile, which is not acceptable. The balloon wall thickness is to be measured by micrometer as shown in

Figure 3-15 [3]. The measurements are taken at three different locations of the balloon: the proximal end, the middle and the distal end.

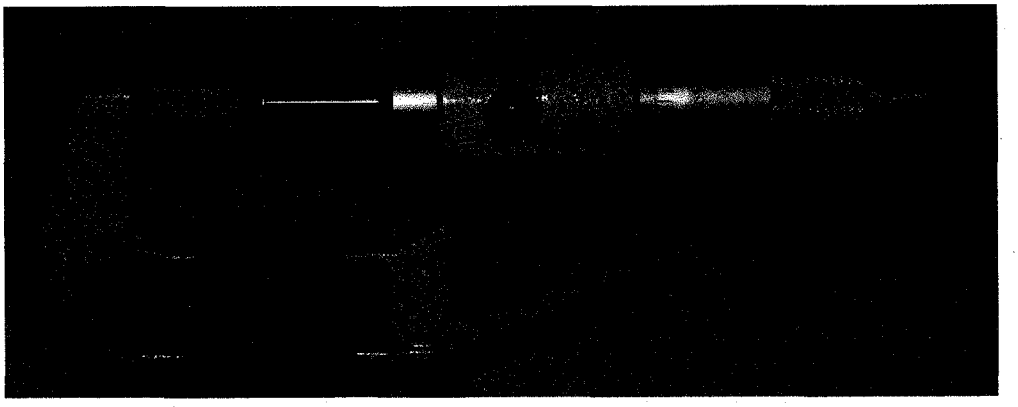


Figure 3-15 The micrometer for measuring the wall thickness

The last two characteristics, the burst pressure and the burst diameter also called the compliance, are measured by a deployment tester.

Chapter 4

OBJECTIVE

4.1 Problem description

The balloon fabrication process is controlled by a complicated combination of manipulated variables, controlled variables, and uncontrolled variables. Balloon forming is considered to be a long trial and error process to find the right parameters to form a balloon that meets the performance specifications without defects. It is believed that most balloon manufacturing companies take months or even years before attaining a balloon suitable for PTCA surgery [3]. When a new balloon is to be developed, the process parameters are determined through a mixture of numerous trial-and-error experiments and experience. Among the requirements [3] for an angioplasty balloon are:

- Balloon is free of visible defects;
- The thinnest possible wall thickness, so as to make a minimum deflated balloon profile enabling the medical device to pass through the clogged coronary artery;
- Burst pressure of more than 15 bar to dilate hard and heavily calcified stenoses;
- Defined burst diameter (balloon compliance)

However, the former experiments and investigations showed that one cannot achieve all the desired balloon properties by optimizing the balloon forming process, because a very slight difference, such as the inner diameter, in the balloon tubes may result in a significant change on the parameters of the forming process. Therefore, the trial and error method poses a problem when a new batch of tubing replaces the old batch. The operator

may need to repeat many experiments to find the new setting parameters for the balloon forming machine.

4.2 The objective

Based on the problem described above, if given a desired balloon characteristics and performance, this thesis provides an approach to find out the balloon forming parameters based on a set of data from the experiments that were previously done. To reduce the material and time invested in the research and development projects, a model of the angioplasty balloon fabrication process is also developed.

4.3 Process parameters of balloon fabrication process

It is known that a variety of parameters will affect the performance of an angioplasty balloon, some of which are uncontrollable. In this thesis, only the controllable parameters are taken into consideration. These include parison length, primary heat temperature, heat time, heat forming pressure, primary stretch speed and distance. To evaluate the effect of the process parameters on the ultimate properties of the balloons, experiments were done in which the main process parameters were varied. Through the control of the process parameters, balloons with various wall thickness and different compliance characteristics can be produced. For instance, the wall thickness is mainly a function of the forming pressure, the heat temperature, and the heat time. And usually, forming with a high pressure at low temperatures results in a thick-walled balloon with high burst pressure and low compliance.

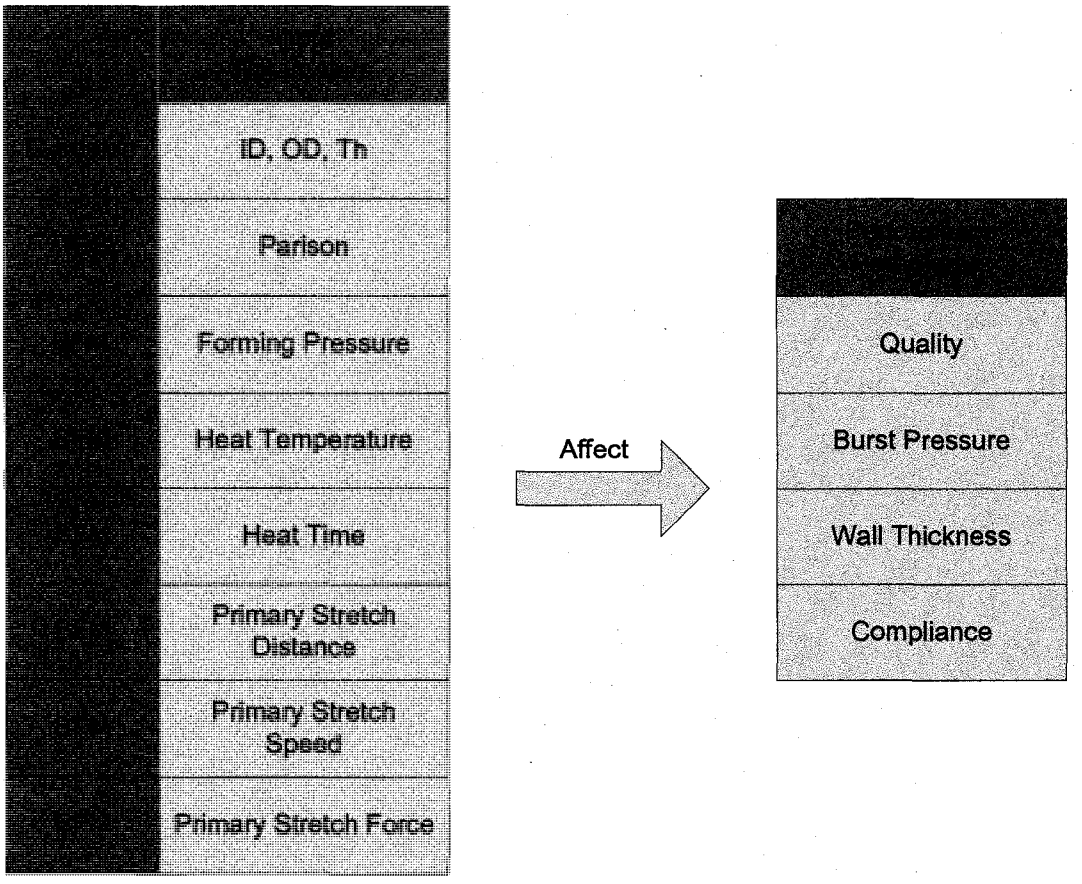


Figure 4-1 Input Parameters that Affect the Balloon Performance Parameters

Figure 4-1 above shows the main input parameters that affect the balloon performance parameters. Let us start with the extrusion process. Since the balloon tubes are to be ordered from some company, the tube characteristics are considered to be uncontrollable, that is to say we cannot change parameters such as tube inner diameter, outer diameter and the original thickness. It is common that there is some slight difference between the tubes even if they are ordered from the same company with the same specifications. In this thesis, the tube characteristics rather than the process parameters for extrusion are going to be taken into consideration. Then, we take a look into the double-end stretching process, which can make different parison lengths. The double-end stretching process parameters include the stretch speed, the stretch distance, the cooling time, the

temperature, the heating time, the unstretched length, etc. In this thesis, the process parameters in the extrusion and double end stretching process are not included in the process modeling directly. Instead, the result from the double-end stretcher, i.e. the parison length, is to be considered as a controllable parameter. Finally, let us look at the balloon forming process parameters, which are the forming pressure, the heat temperature, the heating time, the primary stretch speed, the primary stretch distance, the primary stretch force, as shown in Figure 4-2.

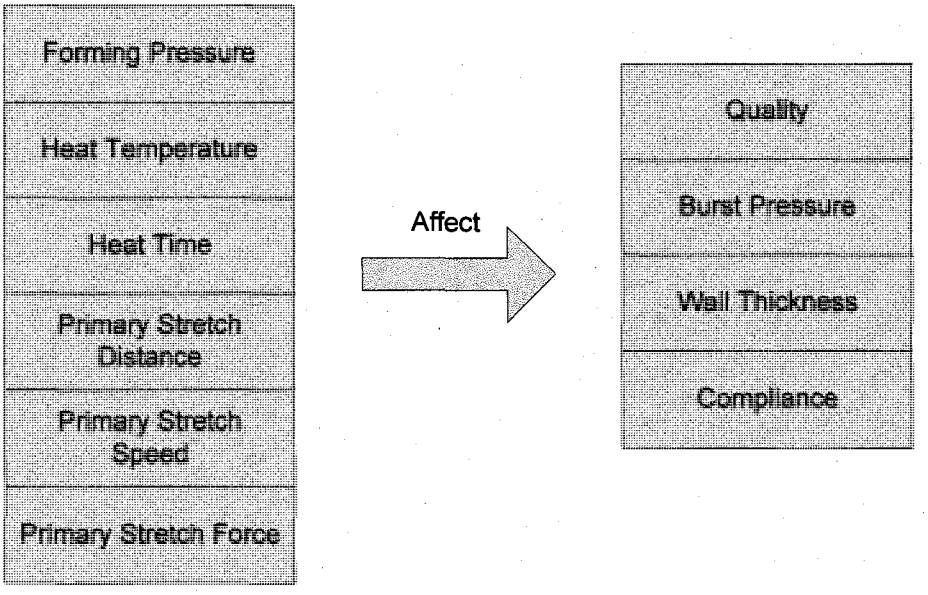


Figure 4-2 BFM Input Parameters and Output Parameters

In the following chapter, we will determine how the BFM process parameters affect the characteristic performance.

Chapter 5

Experimental Data Preprocessing

This chapter aims at reducing the time and number of experiments spent on research and development of angioplasty balloon production when a new batch of tubing has some slight differences from the previous batch. A few new experiments may be required in order to measure the charateristics of the new batch of tubing. Experimental data used for both modeling and process control is from the old experiment database and the new experiments done on the new tubing. The previous experiments were done with tubing batch #1 and #2 ordered before. The purpose is to find the suitable input parameters for the 3rd balloon tube batch if the desired balloon performance output parameters are given.

5.1 Process Inputs and Outputs

The quality, the wall-thickness, the burst pressure and the compliance are the most important quantities to be regulated and they are the outputs of the process. The parison length, the primary heat temperature, the heat time, the forming pressure, the balloon forming temperature, the primary stretch distance, the primary stretch speed, the tubing inner diameter, the tubing outer diameter and the tubing wall thickness can affect the outputs most directly, which is why they are considered as good candidates for control input parameters. Other variables, such as the room temperature, the room humidity, and material properties will be considered as disturbances. Therefore, the process can be modeled as an 8-input 3-output process with disturbance. Let us name the input-output variables as follows:

The controllable input parameters are:

x_1 : the primary heat temperature;

x_2 : the heat time;

x_3 : the forming pressure;

x_4 : the balloon forming temperature;

x_5 : the primary stretch distance;

x_6 : the parison length;

The uncontrollable input parameters are:

x_7 : the tube inner diameter;

x_8 : the tube outer diameter;

x_9 : the tube wall thickness;

Table 5-1 below shows the differences among the three original batches of tubing in inner diameters, outer diameters and wall thicknesses.

Table 5-1 Tube Specifications

	Tube Batch #3	**Tube Batch #2**	**Tube Batch #1**
Inner Diameter (mm)	0.508	0.508	0.508
Outer Diameter (mm)	0.86	0.9144	0.9144
Wall Thickness (mm)	0.176	0.2032	0.20955

The output parameters are:

y_1 : the quality;

y_2 : the exit wall thickness;

y_3 : the burst pressure;

y_4 : the compliance.

In addition, there are also some disturbance variables such as the room temperature, the room humidity, the time the tube is exposed to the air, etc.

The input variables and the output variables in the process are shown in Figure 5-1:

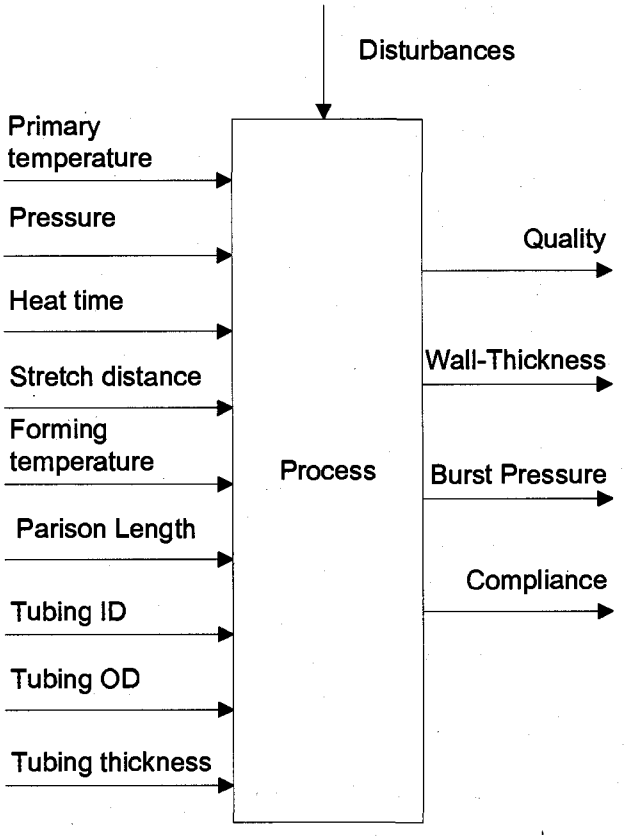


Figure 5-1 Original Input Variables and the Output Variables

5.2 Chosen Inputs and Outputs

Six input parameters and four output parameters will be analyzed in the following chapters and used to make the nonlinear model. The selected inputs are the primary heat temperature, the heat time, the forming pressure, the balloon forming temperature, the primary stretch distance, and the parison length. The four output parameters are the quality, the wall thickness, the burst pressure, the compliance. It is illustrated in Figure 5-2.

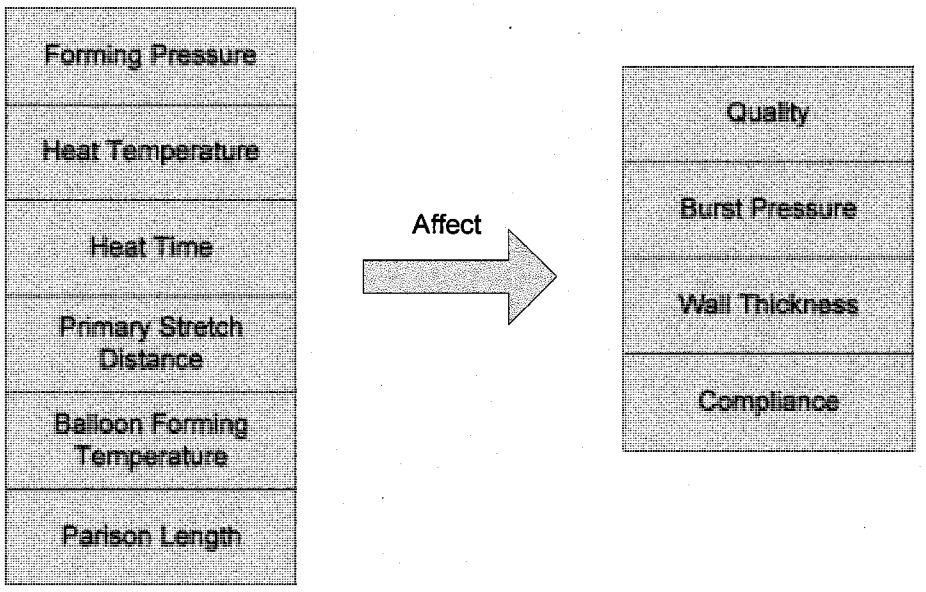


Figure 5-2 Chosen Inputs and Outputs

5.3 Data Pre-processing

5.3.1 Removing the outliers

External disturbances always exist and these disturbances may cause significant errors in the output measurements. Sometimes, these disturbances come from certain extreme machine operating conditions or operation mistakes. The output measurement errors sometimes cause significant problem in modeling. Therefore, the experimental data with

outliers should be removed before modeling. The method used in this thesis to find the outliers in the experimental database is the leave-one-out method [25]. The leave-one-out method is one of the simplest and commonly used method of cross-validation.

Assume that the experimental dataset is of size N (N=81). In the experiment, the leave-one-out method is performed N times. The whole dataset is classified into two groups: the training data and the test data. The leave-one-out method consists of using N-1 experimental data points as the training data to build the process model with a radial basis function (RBF) neural network and the rest of the data for testing whether the error between the model output and the actual output is within tolerance. Finally, the experimental data points with the largest test errors are considered as outliers and are discarded.

After repeating N times, n (n=17) experimental data points are considered to be outliers because their test error is very large.

5.3.2 Normalization of the input matrix and the output matrix

Given a set of input data and related output data, 'normalization' should be performed before modeling because each value of the inputs and the outputs are measured with different units and they differ greatly. Some numerical values may not be desirable because they can create a matrix that is nearly rank deficient. In order to allow both the input matrix and the output matrix to have a better condition number, they are pre-multiplied by scaling matrices before training. The scaling matrices aim to put a limit on each value of the input matrix and the output matrix of 1.

Now, before moving to the next step of modeling, the input matrix $\mathbf{X} \in \mathbb{R}^{6\times N}$ and the output matrix $\mathbf{Y} \in \mathbb{R}^{4\times N}$ are obtained, where N is the number of experiments after removing the outliers, which is N=64 in this case.

Chapter 6
System Modeling Based on RBF Neural Network

This chapter aims at building a nonlinear relationship between the process input parameters and output parameters based on RBF neural network with the data collected and pre-processed from the tests described in the last chapter.

6.1 System Identification

Before implementing a controller, the process model should be identified. In the previous chapter, the six controllable input parameters selected for system modeling are the primary heat temperature $T_p(°C)$, the heat time $t(\mathrm{s})$, the forming pressure $P(atm)$, the balloon forming temperature $T_f(°C)$, the primary stretch distance $d(mm)$, and the parison length $l(mm)$. The three measured output parameters are also identified as the wall thickness $WTh(mm)$, the burst pressure $P_{burst}(atm)$, the compliance $D_{\max}(mm)$. Another one output parameter is to be observed, which is the balloon quality Q with a range from 0 to 4. The vector of input variables is

$$\mathbf{x}_i := \begin{bmatrix} T_p \\ t \\ P \\ T_f \\ d \\ l \end{bmatrix}$$

Equation 6-1 Vector of Input Signals

The vector of output signals is

$$\mathbf{y}_i := \begin{bmatrix} Q \\ WTh \\ P_{burst} \\ D_{\max} \end{bmatrix}$$

Equation 6-2 Vector of Output Signals

Assume that the number of experiments is N, the objective is to make a model and find the relationship $\mathbf{Y} \approx \mathrm{F}(\mathbf{X})$, between the input matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N], \mathbf{X} \in \mathbb{R}^{6\times N}$ and the output matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N], \mathbf{Y} \in \mathbb{R}^{4\times N}$ based on the experimental data previous collected.

The purpose of this chapter is to make a model of the balloon forming process based on process input matrix $\mathbf{X} \in \mathbb{R}^{6\times N}$ and output matrix $\mathbf{Y} \in \mathbb{R}^{4\times N}$. Because the balloon fabrication process is a complicated process that is nonlinear, a radial basis function (RBF) neural network is utilized here to build the relationship between the inputs and the outputs.

6.2 Introduction to neural network

Neural networks, as nonlinear models of real world systems, have been successfully applied in many research and development areas such as signal processing, pattern recognition, system identification, prediction and estimation, and process monitoring and control [16]. In the past few decades, neural networks have seen a fast development due to both a number of break throughs in research and some distinct advances in the power of computer hardware that can implement complicated mathematical algorithms [17]. In

both academia and industry, neural networks have become an invaluable tool to represent nonlinearity, input-output mapping, adaptivity, VLSI implementability, fault tolerance, etc. [16]

6.2.1 Neural Network Principles

6.2.1.1 What is a Neural Network

Generally speaking, a neural network consists of a number of node elements, called neurons (or processing elements), which are connected together to construct either a single layer or multiple layers. The weighted connections between each layer are decided by a weight updating procedure, which is generally a learning process called training. The weight updating process is carried out by passing a set of training data through the model and adjusting the weights to minimize the error between the desired output and the model output [16]. In some neural networks, a supervisor is applied to supervise the learning process, which is referred to as 'supervised learning'. In contrast, a neural network that can realize self-organizing learning procedure can learn in an unsupervised way [17].

6.2.1.2 Models of a Neuron

The neuron is the most fundamental process element (also called as an information-processing unit) to the operation of a neural network [18]. The block diagram in Figure 6-1 [18] shows how a neuron forms the basic foundation of a neural network. There are three basic elements of a neuron [18]:

- a set of connections characterized by a set of weights;

- a summation that performs a linear operation and sums the weighted input signals;
- an activation function used for squashing the amplitude range of the output signal usually to a closed unit interval [0,1] or [-1,1].

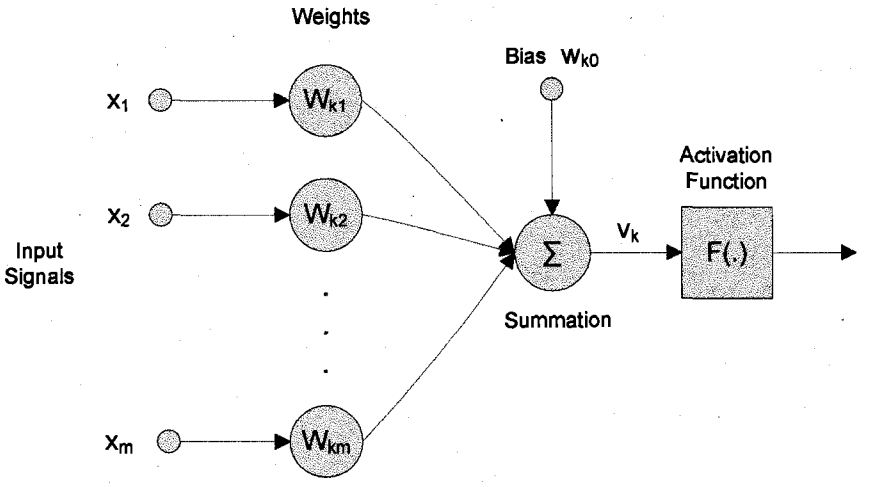


Figure 6-1 Basic Neuron Model

From Figure 6-1, a neuron k in the above diagram can be described mathematically as the following equations:

$$v_k = \sum_{i=0}^{m} \omega_{ki} x_i ,$$

Equation 6-3 Linear Combination Output

and

$$y_k = \mathrm{F}(v_k) ,$$

Equation 6-4 A Neuron Output

where F(.) is the activation function.

Here are some types of activation functions that are commonly applied [18].

- Threshold Function:

$$F(v)=\begin{cases}1 & \text{if } v\geq 0\\ 0 & \text{if } v<0\end{cases}$$

Equation 6-5 Activation Function: Threshold Function

- Piecewise-Linear Function:

$$F(v)=\begin{cases}1 & \text{if } v\geq 0.5\\ v+0.5 & \text{if } -0.5<v<0.5\\ 0 & \text{if } v\leq -0.5\end{cases}$$

Equation 6-6 Activation Function: Piecewise-Linear Function

- Sigmoid Function:

$$F(v)=\frac{1}{1+e^{-a\cdot v}}$$

Equation 6-7 Activation Function: Sigmoid Function

6.2.1.3 The Learning Process of Neural Networks

The most significant property of neural networks is their learning capability, which enables them to learn from the environment and improve their performance [18]. The definition of learning from Mendel and McClaren (1970) is:

Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place.

From the above definition, a learning algorithm is to be adopted in the learning process. There are five basic learning algorithms that are commonly adapted during the learning process: error-correction learning, memory-based learning, Hebbian learning, competitive learning and Boltzmann learning. Some of the algorithms require supervised learning such as error-correction learning and Boltzmann learning, while others require self-organized learning rules such as Hebbian learning and competitive learning [18].

6.2.2 Neural Network Architectures

There are several types of neural network architectures and the most commonly used structures for system identification and control include the single layer perceptron (SLP), multilayer perceptron (MLP), radial basis function neural networks, and recurrent neural networks.

6.2.2.1 Single Layer Perceptrons

The simplest form of layered neural networks is the single layer perceptron network. The single-layer perceptron network is a type of feedforward networks with an input layer of source signals projecting onto the output layer of neurons. Figure 6-2 shows the structure of a single layer perceptron.

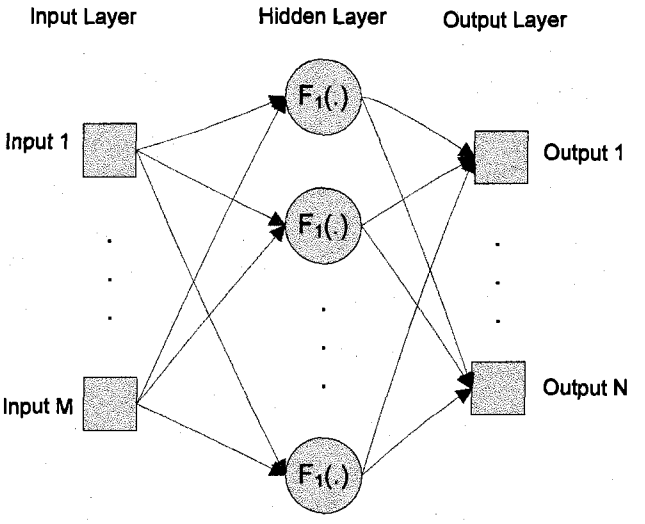


Figure 6-2 Signal Layer Perceptron Structure

6.2.2.2 Multi-layer Perceptrons (MLP)

Another commonly used neural network but with a more complicated structure than the single layer perceptrons is the multilayer perceptron. MLPs assume that many neuron layers exist in the networks, but no connection exists between neurons in a particular layer. The architecture of MLP is a multiple-layer network, which is composed of one input layer, one output layer with several intermediate or hidden layers between them. The input layer requires a set of data or inputs and the output layer is fed with weighted inputs from the layer connected to it and produces an output that is summed by nonlinear functions [18]. Figure 6-3 [18] shows a two-hidden-layer MLP network.

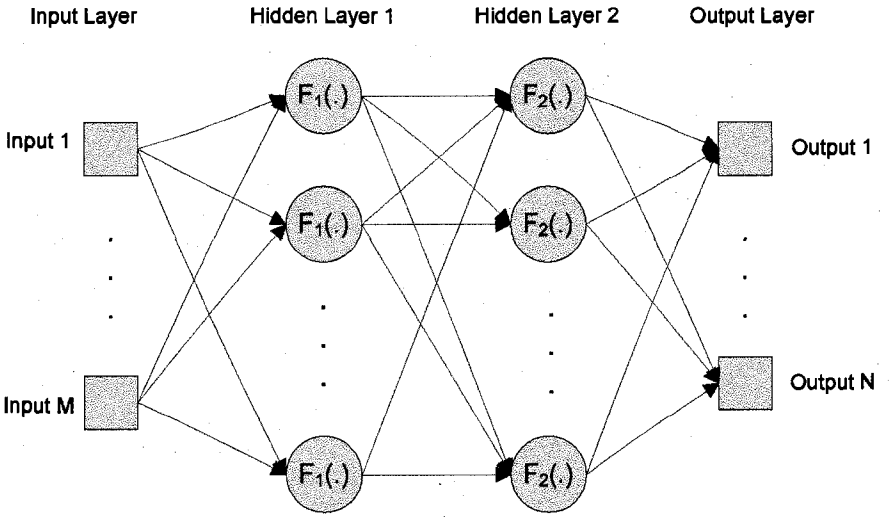


Figure 6-3 Multilayer Perceptron Network Structure

6.2.2.3 Recurrent Neural Networks (RNN)

The two types of neural network structures mentioned above belong to feedforward networks. In this section, a different architecture will be introduced. Recurrent neural network is a network of neurons with feedback connections. Unlike the feedforward neural networks, the structure of RNNs incorporates feedbacks. In general, the recurrent network is built as an MLP network architecture augmented with feedback loop. Usually, the output of every neuron is fed back with varying weights to the inputs of all neurons [17]. It is a good approach for modeling and analyzing dynamic systems. Elman network and Hopfield network are two examples of the RNNs. Figure 6-4 [19] shows a general recurrent neural network.

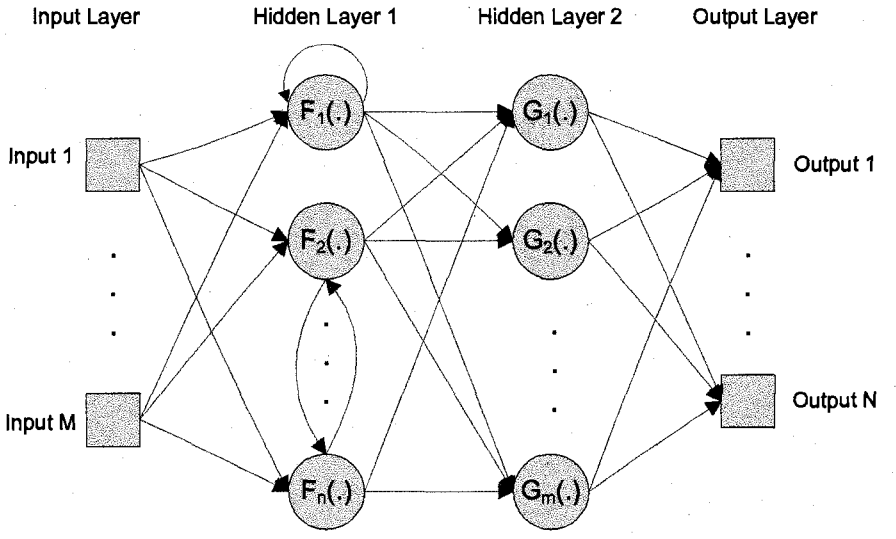


Figure 6-4 Recurrent Neural Network Structure

The above figure is a two hidden-layer RNN example with the output of the hidden neurons fed back as inputs to the network.

6.2.2.4 Radial Basis Function Networks (RBFN)

MLPs and ANNs, have been found, in practice, to perform poorly in several ways such as slow convergence of weights during learning, difficulty in modeling differential responses [17]. Therefore, radial basis function networks have become popular in practice. Radial basis function neural networks are a kind of relatively simple networks with only one hidden layer, the output layer of which is merely a linear combination of the hidden layer signals; RBF networks have the ability to model any nonlinear function in a relatively straightforward way [18]. Figure 6-5 [18] shows the structure of a RBF neural network. As described above, RBF networks have three different layers [18]:

- the input layer made of input neurons;

- the hidden layer with proper number of neurons and each neuron performs a nonlinear radial basis function on the inputs;
- the output layer, which is a linear combination of the outputs of hidden neurons.

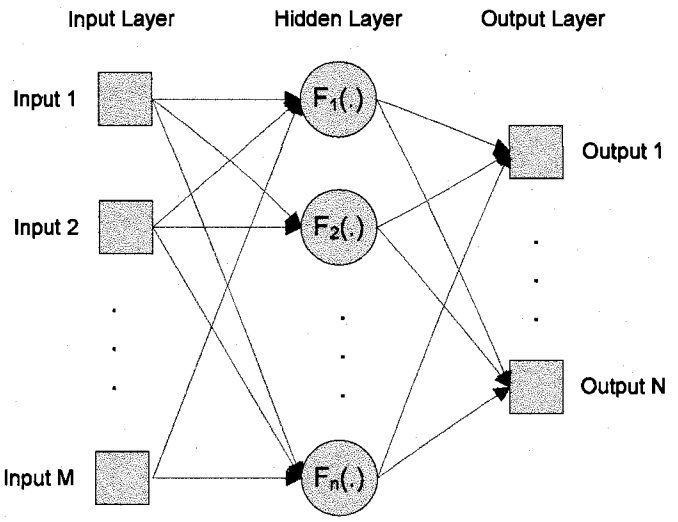


Figure 6-5 Radial Basis Function Network Structure

Because the output layer is merely a linear combination of the hidden layer signals, the weight updating procedure is much simpler for RBF networks [18].

6.2.3 RBF Neural Network and its Applications

6.2.3.1 RBF Neural Networks

In this thesis, the radial basis function network is chosen to find a relationship between the inputs signals and the output signals of the balloon forming process because of its features of relatively fast convergence and capability of modeling a nonlinear multivariable system.

6.2.3.2 RBF Neural Network Architecture

RBF neural networks have only one hidden layer and each neuron of the hidden layer performs a radial activation function while the outputs perform a weighted summation of the hidden layer outputs. Figure 6-6 shows a general diagram of an RBF neural network with one output.

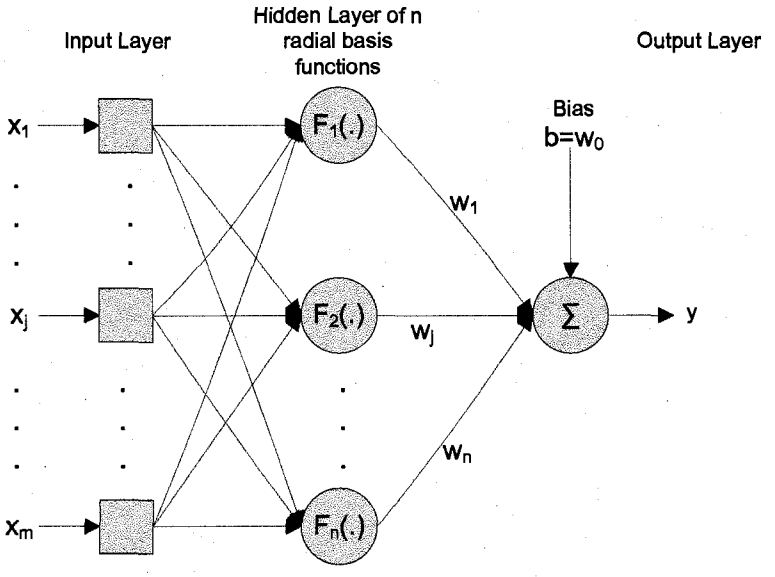


Figure 6-6 A radial basis function network

In mathematical terms, an RBF neural network can be described as follows:

Assume the network consists of an input vector $\mathbf{x}$ with m input signals and a set of weight $\omega_i \quad (i = 1, \cdots, n)$ with a bias term $b = \omega_0$, and n is the number of hidden layer neurons. The neural network output can be calculated by:

$$y = \sum_{i=0}^{n} \omega_i \mathrm{F}_i(\mathbf{x})$$

Equation 6-8 The Neural Network Output

The $F_i(.)$'s are the activation functions with the form of

$$F_i(\mathbf{x}) = \Phi(\|\mathbf{x} - \mathbf{c}_i\|),$$

Equation 6-9 Activation Functions for RBF Networks

where $\mathbf{c}_i$ $(i = 1, \cdots, n)$ are considered to be the radial basis function centers. $\Phi(\cdot)$ is the radial basis function and there are many choices for this function. Here, some of the most commonly used radial basis functions [17] are given as follows:

- Gaussian:

$$\Phi(r) = e^{-\frac{r^2}{2\sigma^2}}, \quad \sigma > 0, r \in \mathbb{R}$$

Equation 6-10 Radial Basis Function: Gaussian

- Piecewise linear approximation:

$$\Phi(r) = r, \quad r \in \mathbb{R}$$

Equation 6-11 Radial Basis Function: Piecewise Linear Approximation

- Cubic approximation:

$$\Phi(r) = r^3, \quad r \in \mathbb{R}$$

Equation 6-12 Radial Basis Function: Cubic Approximation

- Thin plate spline

$$\Phi(r) = r^2 \log(r), \quad r > 0$$

Equation 6-13 Radial Basis Function: Thin Plate Spline

- Multiquadrics:

$$\Phi(r)=\left(r^2+c^2\right)^{1/2},\ c>0, r\in\mathbb{R}$$

Equation 6-14 Radial Basis Function: Multiquadrics

- Inverse multiquadrics:

$$\Phi(r)=\frac{1}{\left(r^2+c^2\right)^{1/2}},\ c>0, r\in\mathbb{R}$$

Equation 6-15 Radial Basis Function: Inverse Multiquadrics

where r is the Euclidean distance between the input $\mathbf{x}$ and the radial basis function center $\mathbf{c}_i$ so that

$$r=\left\|\mathbf{x}-\mathbf{c}_i\right\|.$$

Equation 6-16 Euclidean Distance

Among all the radial basis functions listed above, the Gaussian function is the most intuitive one. And in practice, the thin plate spline function also works very well [17].

In the above Figure 6-6, only one output is shown, but it is possible for RBF neural networks to model multi-input and multi-output systems. To build a RBF neural network, the radial basis function centers are first selected. Then, the network weights can be updated and adjusted to minimize the errors between the desired outputs and the network outputs.

6.2.3.3 RBF Neural Network Application

Being one of the primary fields of research in numerical analysis, radial basis functions were first introduced in the field of the solutions to the real multivariate interpolation problem [16]. Due to their nonlinear approximation features, RBF neural networks have

been widely applied in many fields such as nonlinear system identification, pattern recognition, interpolation, data fusion, image processing, control, etc.

6.3 System Identification Based on Neural Networks

6.3.1 Objective

Based on the experimental data, a system model describing the relationship between the inputs and the outputs of the balloon forming process is to be established, that is, we want to find a mapping $F(\cdot)$ such that

$$\mathbf{Y} \approx F(\mathbf{X}).$$

Equation 6-17 Relationship between Inputs X and Outputs Y

Based on experience and previous experimental results, the following input parameters have been found to have significant effects on the outputs and are chosen as the input signals of the RBF neural network.

- Primary heat temperature;
- Heat time;
- Form pressure;
- Secondary heat temperature;
- Primary stretch distance;
- Parison length;

And the output parameters for the RBF neural network are:

- Quality;
- Wall Thickness;
- Burst pressure;
- Compliance.

The experiments were done with the three batches of tubing ordered from the same company but having slight differences. The training data set is taken as 3/4 of the whole data set after preprocessing and removing the outliers; and the testing data set takes the rest 1/4 of the whole data set after preprocessing and removing the outliers.

6.3.2 Approaches for System Modeling

6.3.2.1 Approach I: the RBF neural network

To build and train the RBF neural network, ¾ of the whole experimental data are taken as the training data and the rest ¼ as test data. A Gaussian kernel is selected here as the radial basis function. In this thesis, a Matlab® function *newrb* is utilized to initialize, build and train the RBF neural network model. The *newrb* function provides an efficient way to build up a RBF neural network by adding one neuron in the hidden layer at a time until the sum-square error satisfies the requirement or a maximum number of neurons have been achieved [21]. The Figure 6-7 [21] is the architecture of a radial basis neuron.

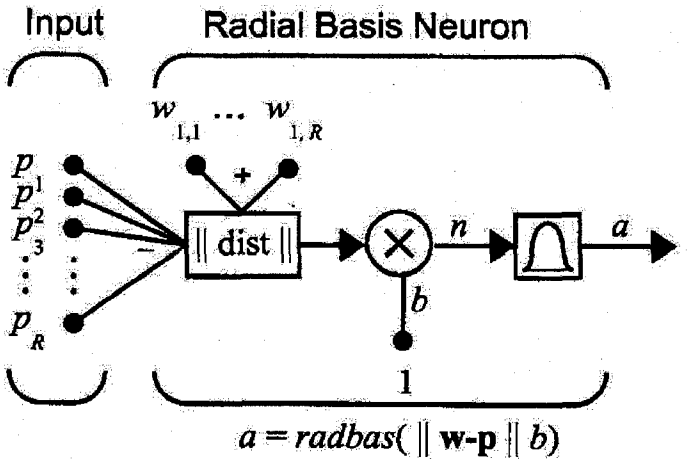


Figure 6-7 The Architecture of a Radial Basis Neuron

And the Radial basis function is the Gaussian function shown in Figure 6-8 [21],

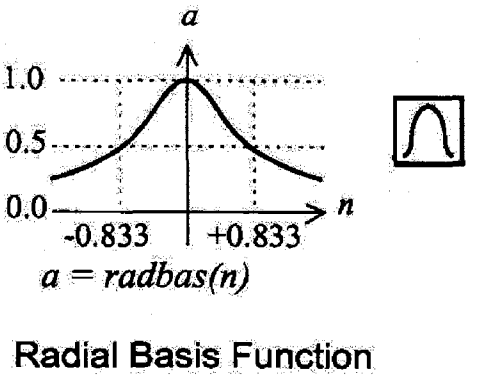


Figure 6-8 Gaussian Function as the RFB Kernel Function

The radial basis network architecture is as shown in Figure 6-9 [21]:

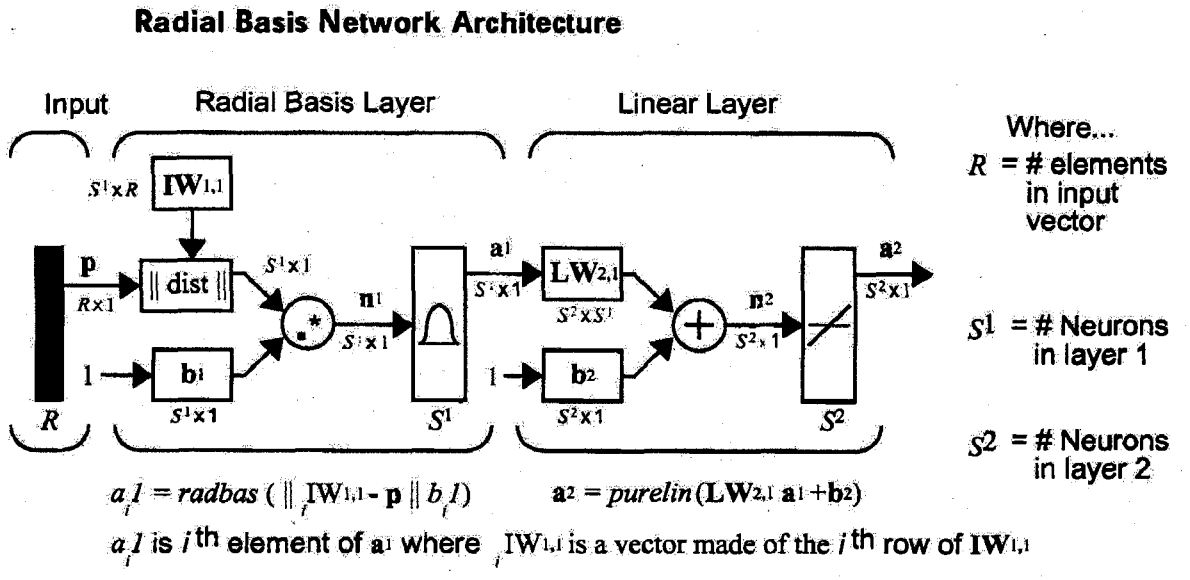


Figure 6-9 Radial Basis Function Architecture in Matlab®

From the above diagram in figure 6-9, the following mathematical equations can be written:

$$\mathbf{a}^1 = radbas(netprod(dist(\mathbf{IW}_{1,1},\mathbf{p}),\mathbf{b}^1)) = \exp\left\{-\left(\left\|\mathbf{IW}_{1,1}-\mathbf{p}\right\|\cdot\mathbf{b}^1\right)^2\right\}$$

Equation 6-18 RBF Neural Network Neuron Output by Matlab®

$$\mathbf{a}^2 = purelin\left(\mathbf{LW}_{2,1}\cdot\mathbf{a}^1+\mathbf{b}^2\right) = \mathbf{LW}_{2,1}\cdot\mathbf{a}^1+\mathbf{b}^2 = \mathbf{LW}_{2,1}\cdot\exp\left\{-\left(\left\|\mathbf{IW}_{1,1}-\mathbf{p}\right\|\cdot\mathbf{b}^1\right)^2\right\}+\mathbf{b}^2$$

Equation 6-19 RBF Neural Network Output by Matlab®

With the function *newrb* provided by Matlab®, a system model is built based on the training data set. With a maximum sum-square error set to be 0.1, the RBF neural network training process is as shown in Figure 6-10:

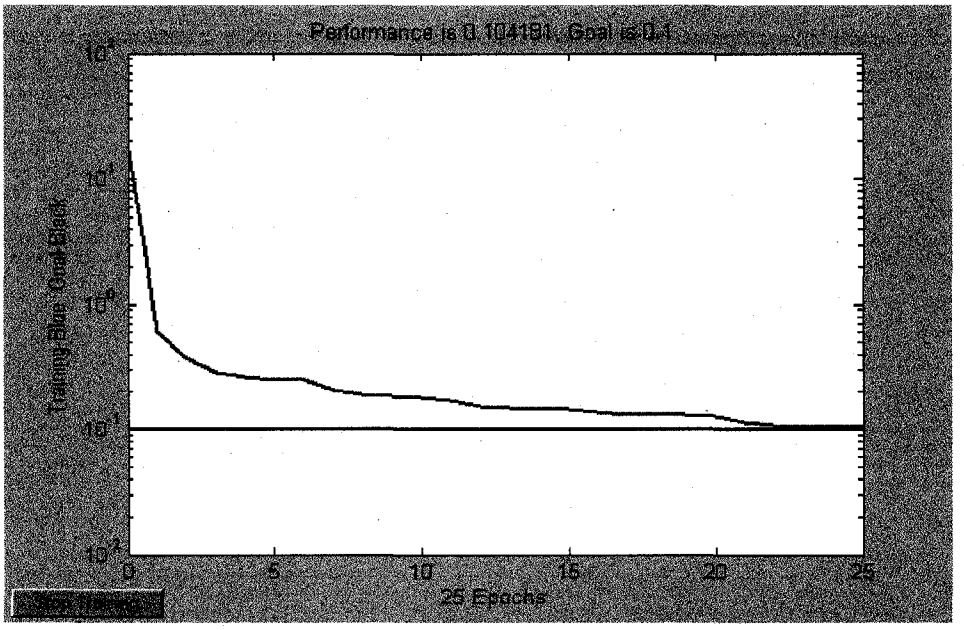


Figure 6-10 RBF Neural Network Training

And the test error distribution is shown in Figure 6-11

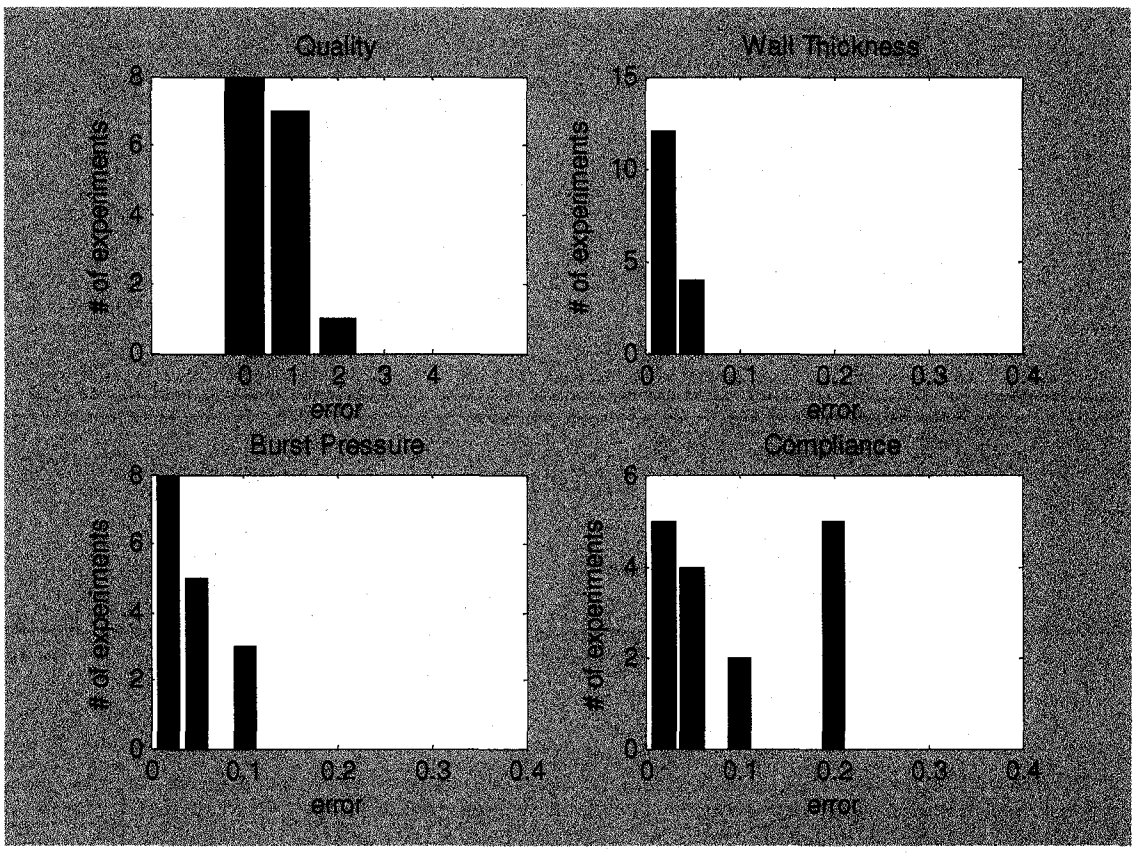


Figure 6-11 the test error distribution of the RBF neural network model

6.3.2.2 Approach II: the Single Layer Feedforward network with Backpropagation

In this part, a single-layer neural network is introduced to model the balloon forming process. The Network Structure is shown in Figure 6-12.

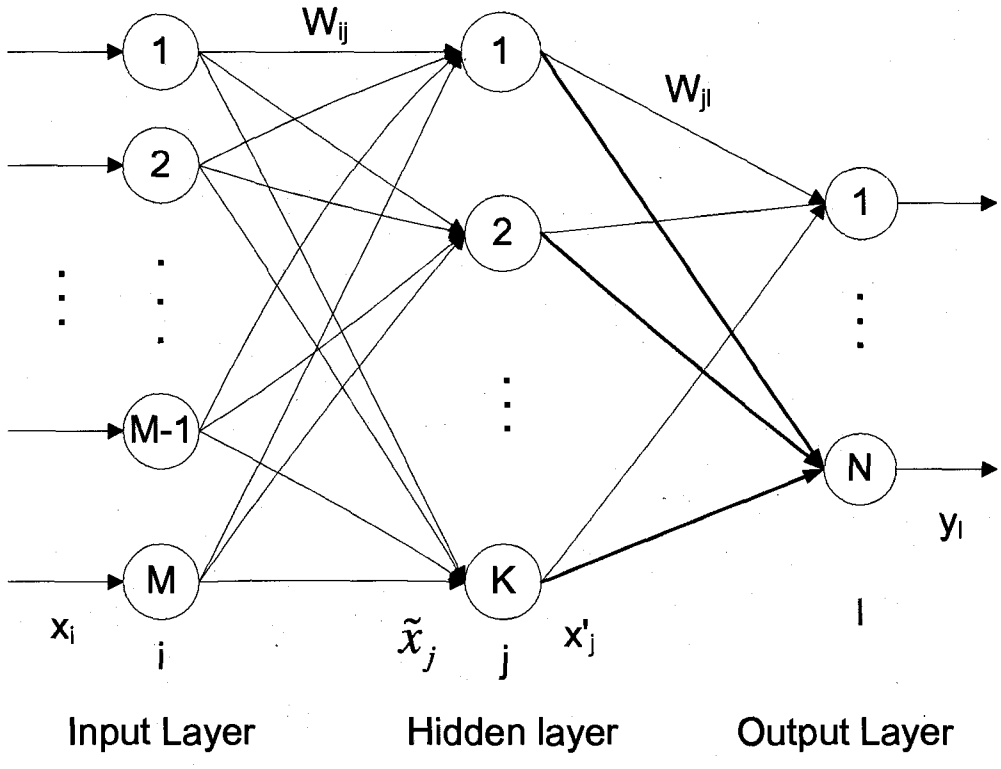


Figure 6-12 A Single Layer Feedforward Neural Network

The backpropagation learning algorithm [16], which is a popular learning algorithm used in neural networks, is applied here in the single layer feedforward neural network. The algorithm diagram is shown in Figure 6-13.

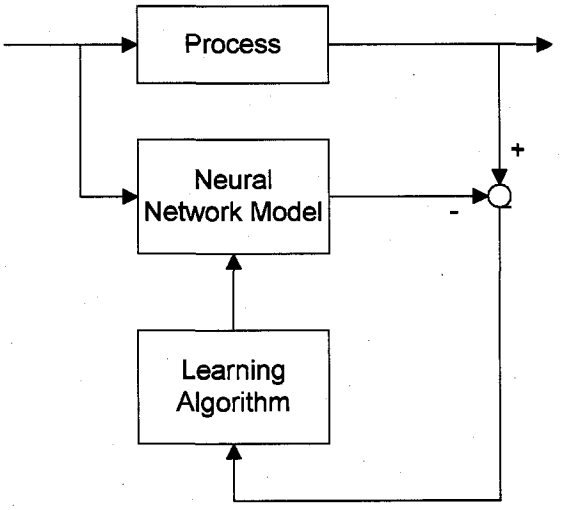


Figure 6-13 Backpropagation Learning Algorithm Diagram

In the above figure, the network structure has M inputs, K hidden neurons, N outputs. The algorithm [16, 22, 23 and 24] is described as follows:

(1) Forward: to compute the network output

The input of the hidden layer is $\tilde{x}_j$

$$\tilde{x}_j = \sum_i \omega_{ij} x_i .$$

Equation 6-20 Hidden Layer Input

The output of the hidden layer is

$$x_j' = f\left(\tilde{x}_j\right) = \frac{1}{1+e^{-\tilde{x}_j}}.$$

Equation 6-21 Hidden Layer Output

The output of the network is

$$y_l = \sum_j \omega_{jl} x_j' .$$

Equation 6-22 BP Neural Network Output

The error between the l^{th} actual output and desired output is

$$e_l = y_l^o - y_l .$$

Equation 6-23 Error between the Actual Output and the Desired Output

The cost function for the p^{th} experimental data is

$$E_p = \frac{1}{2}\sum_{l=1}^{N} e_l^2 .$$

Equation 6-24 Cost Function

(2) Backward: for weight update using backpropagation

The algorithm for weight update between the hidden layer and the output layer is

$$\Delta\omega_{jl} = -\eta\frac{\partial E_p}{\partial \omega_{jl}} = \eta e_l \frac{\partial x_l}{\partial \omega_{jl}} = \eta e_l x_j$$

Equation 6-25 Weight Update Increment between the Hidden and the Output Layer

where $\eta \in [0,1]$ is the learning rate.

The $k+1^{th}$ weight update is

$$\omega_{jl}(k+1) = \omega_{jl}(k) + \Delta\omega_{jl}$$

Equation 6-26 Weight Update

The algorithm for weight update between the input layer and the hidden layer is

$$\Delta\omega_{ij} = -\eta\frac{\partial E_p}{\partial \omega_{ij}} = \eta\sum_{l=1}^{N} e_l \frac{\partial x_l}{\partial \omega_{ij}}$$

Equation 6-27 Weight Update Increment between the Hidden and the Input Layer

The $k+1^{th}$ weight update is

$$\omega_{ij}(k+1) = \omega_{ij}(k) + \Delta\omega_{ij}$$

Equation 6-28 Weight Update between the Hidden and the Input Layer

Here, the momentum factor $\alpha \in [0,1]$ is introduced and the weight update becomes

$$\omega_{jl}(k+1) = \omega_{jl}(k) + \Delta\omega_{jl} + \alpha\left(\omega_{jl}(k) - \omega_{jl}(k-1)\right)$$

Equation 6-29 Weight Update with Momentum between the Hidden and the Output Layer

$$\omega_{ij}(k+1) = \omega_{ij}(k) + \Delta\omega_{ij} + \alpha\left(\omega_{ij}(k) - \omega_{ij}(k-1)\right)$$

Equation 6-30 Weight Update with Momentum between the Hidden and the Input Layer

6.3.3 Comparing the Test Results from the RBF and BP neural networks

After training the RBF neural network and the single layer BP neural network, to compare which one works better, certain tests are designed. The original models are built through a set of fixed training data from ¾ of the whole experimental data set. And test is based on the remaining ¼ of the experimental data.

With fixed training data and test data (¾ of the whole as training and ¼ as test), the test results are shown as follows:

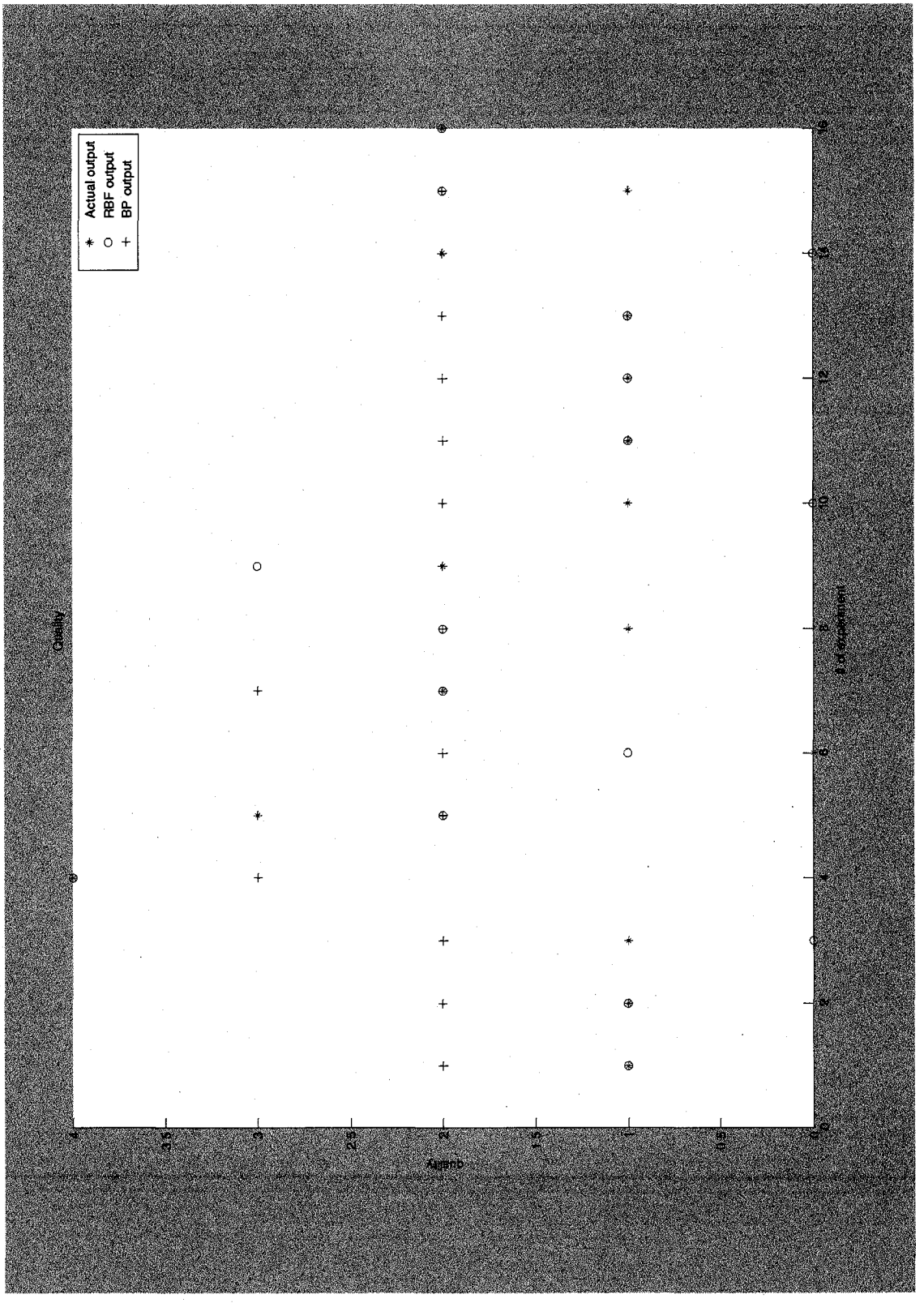


Figure 6-14 Quality from the RBF Model and BP Model

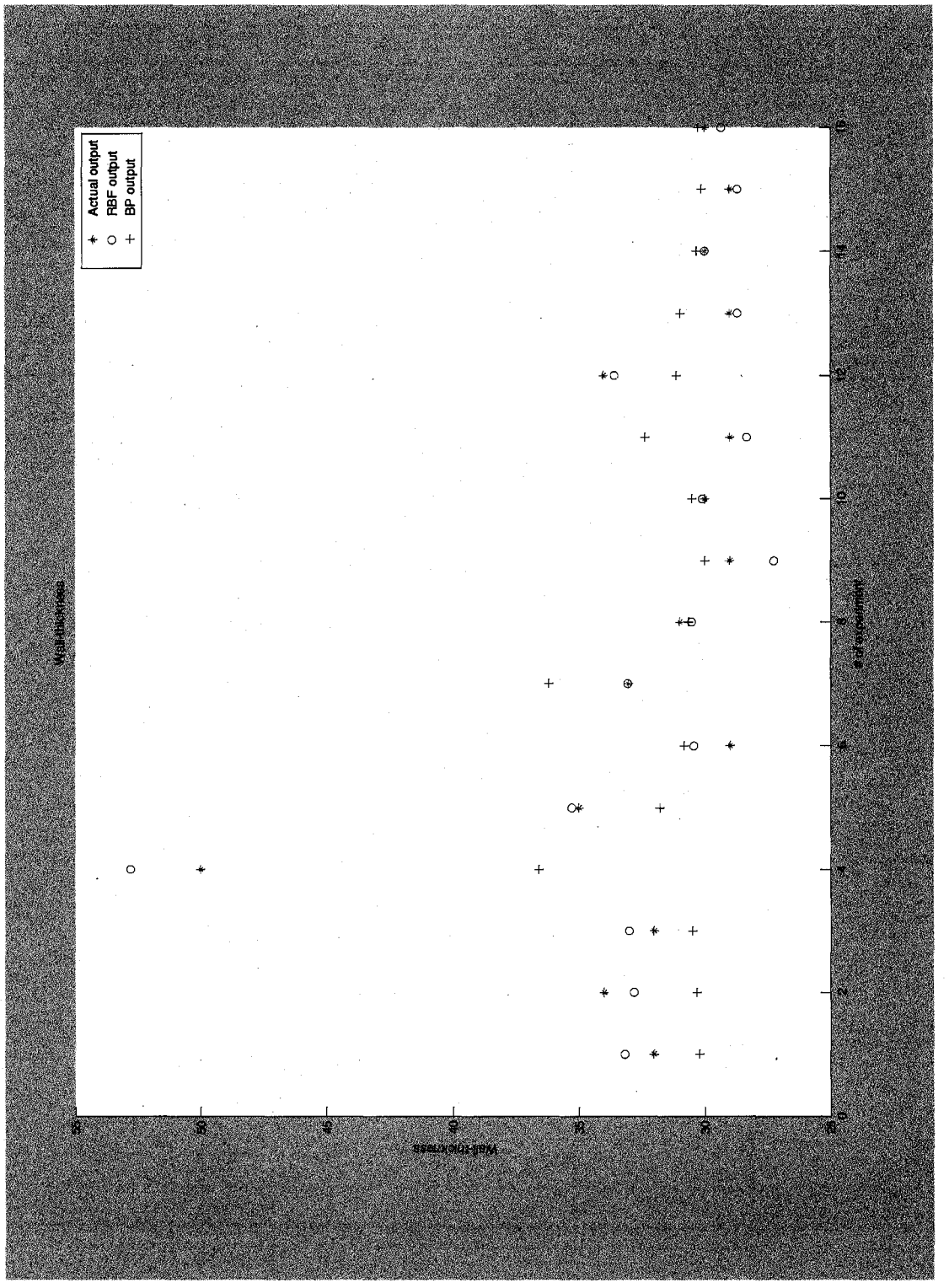


Figure 6-15 Wall Thickness from the RBF Model and BP Model

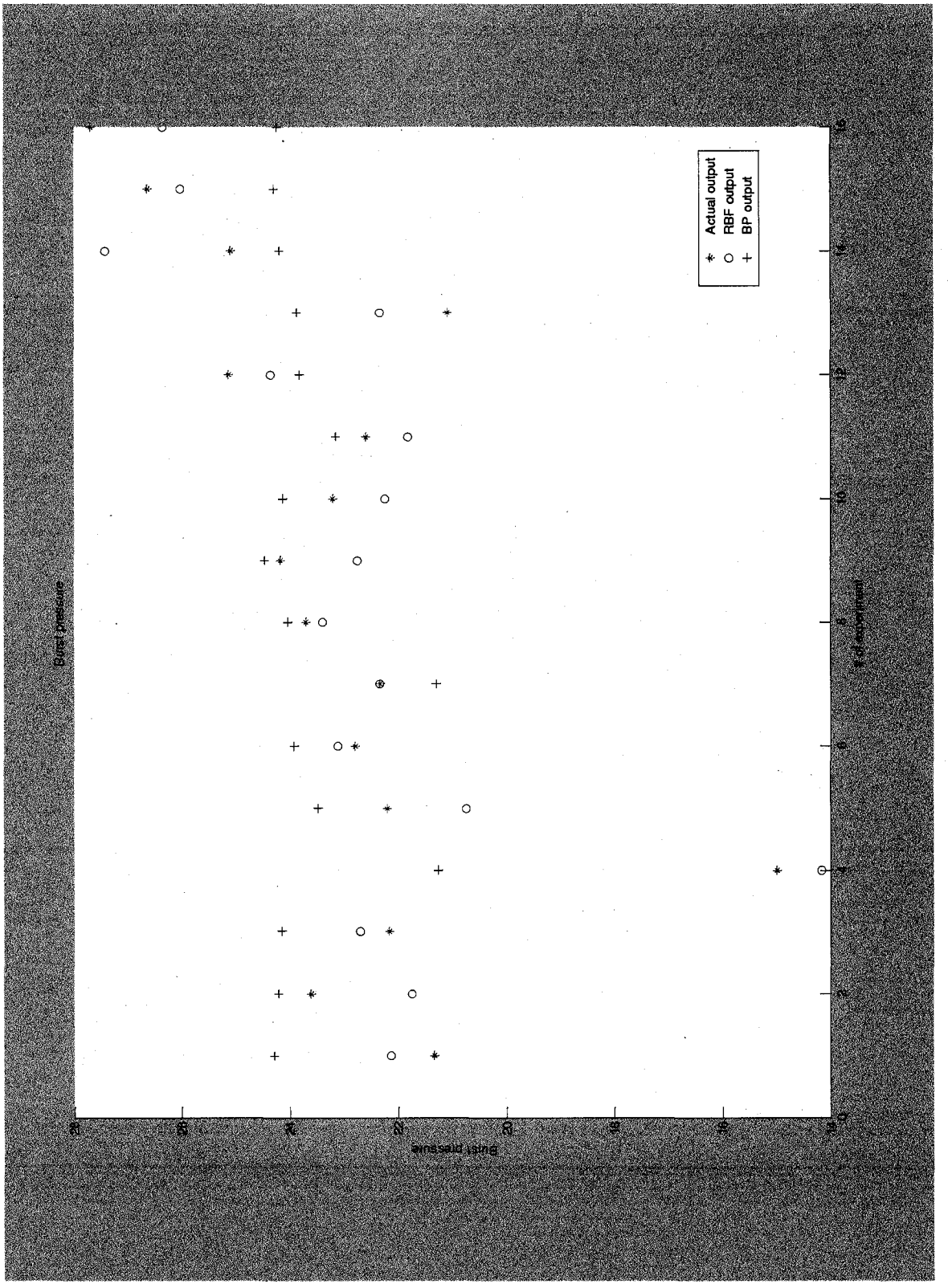


Figure 6-16 Burst Pressure from the RBF Model and BP Model

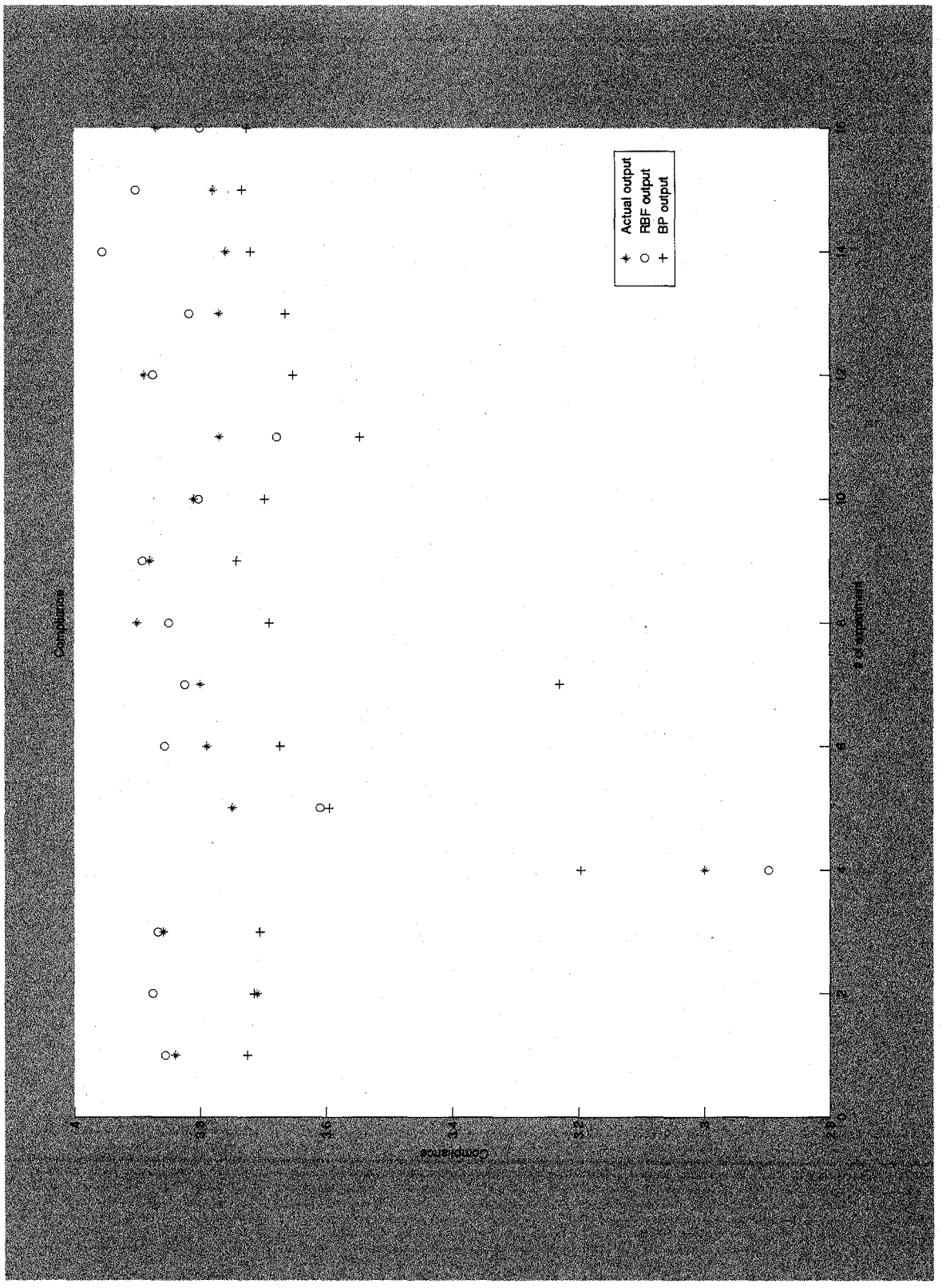


Figure 6-17 Compliance from the RBF Model and BP model

Figure 6-18 shows the comparison of the test results from the RBF neural network model and the BP neural network model.

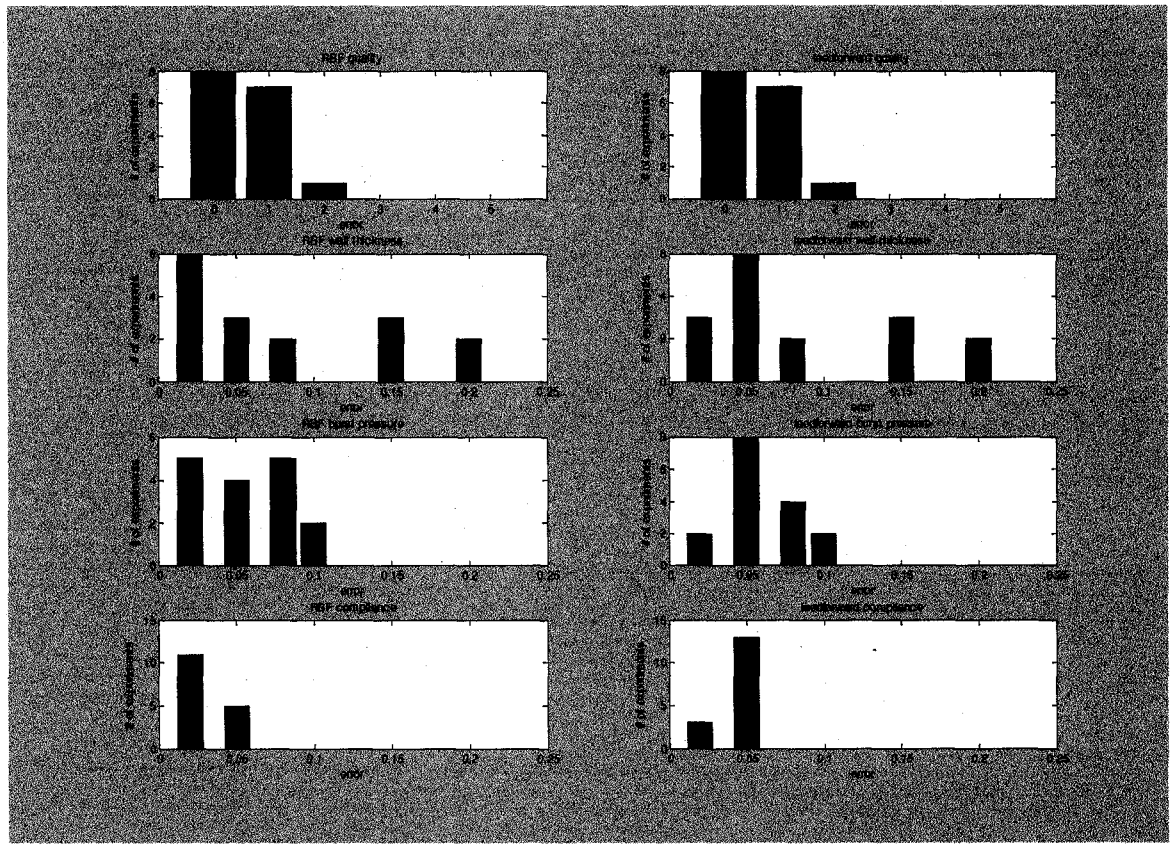


Figure 6-18 Test Result Comparison between RBF Model and BP Model

From the above test result comparisons, the system model built with the RBF neural network gives more accurate results. Thus, it is utilized in this thesis for modeling the balloon forming process.

Chapter 7

Cycle-to-Cycle Off-line Control of BFM

7.1 System Identification

Before implementing an off-line controller on the BFM, the first step is the identification of the system in use. In the previous chapter, the six input signals are given as the primary heat temperature $T_p\,(^\circ C)$, the heat time $t(\text{sec})$, the forming pressure $P(atm)$, the balloon forming temperature $T_f\,(^\circ C)$, the primary stretch distance $d(mm)$, and the parison length $l(mm)$. The three measured output parameters are also identified as the wall thickness $Wth(mm)$, the burst pressure $P_{burst}\,(atm)$, the compliance $D_{\max}\,(mm)$. One output parameter is to be observed, which is the balloon quality Q with a range from 0 to 4.

With Matlab®, a neural network object *net* is obtained. The architecture of RBF neural network *net* after training consists of one input weight matrix *net*.**IW**, one layer weight matrix *net*.**LW** and two bias vectors *net*.$\mathbf{b}_1$ and *net*.$\mathbf{b}_2$

The relationship between input signals **x** and output **y** is

$$\mathbf{y} = net.\mathbf{LW} \cdot \exp\left\{-\left(\left\|net.\mathbf{IW} - \mathbf{x}\right\| \cdot net.\mathbf{b}_1\right)^2\right\} + net.\mathbf{b}_2$$

Equation 7-1 System Model: Relationship between Model Input x and Output y

And the weights and bias values are as follows:

$$net.\mathbf{IW} = \begin{bmatrix}
0.8000 & 0.7500 & 0.4000 & 0.1300 & 0.1500 & 0.1300 \\
0.7000 & 0.7000 & 0.4200 & 0.1400 & 0.1050 & 0.1300 \\
0.7000 & 0.7000 & 0.3900 & 0.1340 & 0.1050 & 0.1300 \\
0.6000 & 0.6500 & 0.3500 & 0.1300 & 0.1350 & 0.1350 \\
0.5000 & 0.6000 & 0.3500 & 0.1300 & 0.1350 & 0.1350 \\
0.6000 & 0.6000 & 0.4100 & 0.1300 & 0.1350 & 0.1350 \\
0.6000 & 0.6000 & 0.3500 & 0.1300 & 0.0850 & 0.1350 \\
0.6000 & 0.6000 & 0.1000 & 0.1300 & 0.1350 & 0.1350 \\
0.8000 & 0.6000 & 0.3500 & 0.1300 & 0.1350 & 0.1350 \\
0.6000 & 0.8000 & 0.3500 & 0.1300 & 0.1350 & 0.1350 \\
0.8000 & 0.7000 & 0.3800 & 0.1300 & 0.1500 & 0.1300 \\
0.7600 & 0.7000 & 0.4000 & 0.1300 & 0.1500 & 0.1300 \\
0.8000 & 0.6750 & 0.4000 & 0.1300 & 0.1500 & 0.1300 \\
0.6000 & 0.6000 & 0.3500 & 0.1300 & 0.1750 & 0.1350 \\
0.3500 & 0.6000 & 0.3500 & 0.1300 & 0.1350 & 0.1350 \\
0.6200 & 0.5500 & 0.3600 & 0.1400 & 0.1050 & 0.1300 \\
0.7800 & 0.7000 & 0.4000 & 0.1300 & 0.1500 & 0.1300 \\
0.6000 & 0.6000 & 0.2000 & 0.1300 & 0.1350 & 0.1350 \\
0.8000 & 0.7000 & 0.3900 & 0.1300 & 0.1500 & 0.1300
\end{bmatrix}$$

Equation 7-2 System Model -- Input Weight Matrix: net.IW

$$net.\mathbf{LW} = 10^{11} \cdot \begin{bmatrix} -0.2202 & -0.0214 & -0.0166 & 0.0070 \\ -0.0009 & 0.0000 & -0.0000 & 0.0000 \\ 0.0009 & -0.0000 & 0.0000 & -0.0000 \\ -0.1025 & -0.0096 & -0.0068 & 0.0035 \\ -0.1578 & -0.0778 & 0.0806 & 0.0400 \\ 0.1588 & 0.0559 & -0.0453 & -0.0269 \\ -0.0100 & -0.0018 & -0.0023 & -0.0001 \\ -0.0644 & -0.0223 & 0.0166 & 0.0103 \\ -0.0397 & -0.0194 & 0.0203 & 0.0101 \\ 0.0258 & 0.0024 & 0.0017 & -0.0009 \\ -2.1609 & -0.7807 & 0.7310 & 0.4080 \\ 1.4940 & 0.7169 & -0.7810 & -0.3869 \\ -0.4423 & -0.0427 & -0.0332 & 0.0140 \\ -0.0119 & -0.0023 & -0.0029 & -0.0002 \\ 0.0311 & 0.0156 & -0.0161 & -0.0079 \\ 0.0006 & -0.0000 & 0.0000 & -0.0000 \\ -2.9895 & -1.4337 & 1.5619 & 0.7739 \\ 0.1707 & 0.0594 & -0.0458 & -0.0279 \\ 4.3183 & 1.5616 & -1.4621 & -0.8159 \end{bmatrix}^T$$

Equation 7-3 System Model -- Layer Weight Matrix: net.LW

$$net.\mathbf{b}_1 = \begin{bmatrix} 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \end{bmatrix}$$

Equation 7-4 System Model -- Vector Basis 1: net.$\mathbf{b}_1$

$$net.\mathbf{b}_2 = 10^4 \cdot \begin{bmatrix} 9.7894 \\ 1.8294 \\ 2.2056 \\ 0.1119 \end{bmatrix}$$

Equation 7-5 System Model -- Vector Basis 2: net.$\mathbf{b}_2$

7.2 Off-line Control

7.2.1 Controller Design

The main objective here is, given the desired outputs, to find out the closest inputs to start with based on the experiments previously done. A simple flow diagram of the off-line control algorithm based on the RBF neural network is illustrated in Figure 7-1.

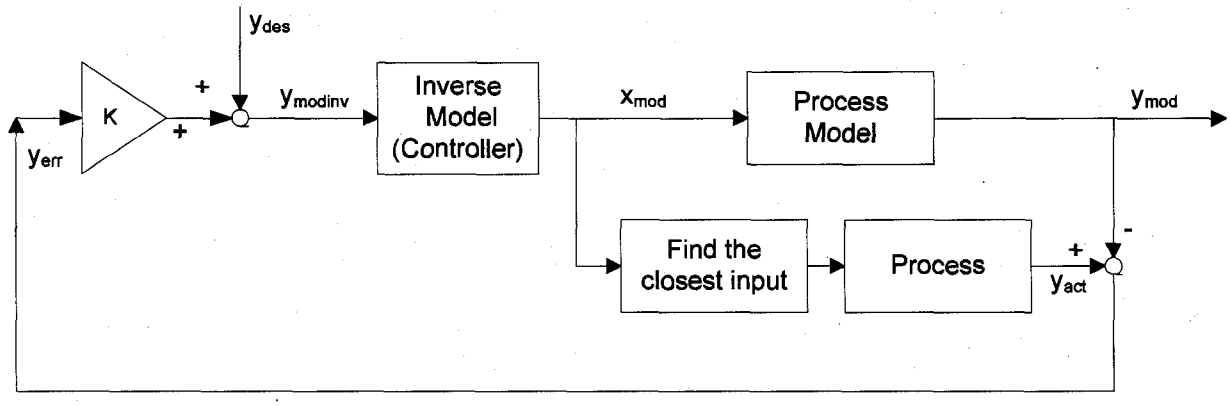


Figure 7-1 Offline Cycle-to-Cycle Control Diagram

In order to make good use of the previous experimental data and save time redoing new experiment, an off-line control algorithm is applied in this thesis. After the inverse model calculates the estimate inputs $\mathbf{x}_{\text{mod}}$, a look-up table is used to find the closest inputs to the inputs given by the inverse model and take the related output parameters as the actual outputs $\mathbf{y}_{act}$ in the algorithm.

The controller applied in Figure 7-1 is a controller based on an RBF neural network, which is the inverse model of the process built by another RBF neural network. The inverse model is built based on the same training data as the system model. The neural network object *netinv* with one input weight matrix *netinv*.**IW**, one layer weight matrix

netinv.**LW** and two bias vectors *netinv*.$\mathbf{b}_1$ and *netinv*.$\mathbf{b}_2$. The relationship between output **y** and input signals **x** is

$$\mathbf{x} = netinv.\mathbf{LW} \cdot \exp\left\{-\left(\left\|netinv.\mathbf{IW} - \mathbf{y}\right\| \cdot netinv.\mathbf{b}_1\right)^2\right\} + netinv.\mathbf{b}_2$$

Equation 7-6 Syem Inverse Model

And the weights and bias values are as follows:

$$
netinv.\mathbf{IW} = \begin{bmatrix}
0.2000 & 0.2800 & 0.2619 & 0.3950 \\
0.2000 & 0.3300 & 0.2117 & 0.3600 \\
0.2000 & 0.3000 & 0.2417 & 0.3920 \\
0.2000 & 0.3200 & 0.2217 & 0.3840 \\
0.2000 & 0.3300 & 0.2127 & 0.3820 \\
0.2000 & 0.3100 & 0.2659 & 0.3750 \\
0.2000 & 0.3200 & 0.2344 & 0.3810 \\
0.2000 & 0.3300 & 0.2110 & 0.3890 \\
0.2000 & 0.3000 & 0.2410 & 0.3900 \\
0.2000 & 0.3300 & 0.2518 & 0.3950 \\
0.2000 & 0.2800 & 0.2447 & 0.3890 \\
0.2000 & 0.3500 & 0.2511 & 0.3850 \\
0.2000 & 0.3500 & 0.2415 & 0.3870 \\
0 & 0.3000 & 0.2270 & 0.3800 \\
0 & 0.3000 & 0.2320 & 0.3840 \\
0 & 0.3200 & 0.2150 & 0.3870 \\
0.1000 & 0.2800 & 0.2315 & 0.3940 \\
0.1000 & 0.2800 & 0.2099 & 0.3760 \\
0.1000 & 0.2800 & 0.2686 & 0.3830 \\
0.1000 & 0.2800 & 0.2315 & 0.3870 \\
0.3000 & 0.4100 & 0.1815 & 0.3460 \\
0.1000 & 0.2900 & 0.2774 & 0.3900 \\
0 & 0.3400 & 0.2205 & 0.3900 \\
0.4000 & 0.5000 & 0.1500 & 0.3000 \\
0.4000 & 0.5000 & 0.1500 & 0.3000 \\
0.1000 & 0.3300 & 0.2415 & 0.3900 \\
0.1000 & 0.3200 & 0.2414 & 0.3850 \\
0.1000 & 0.3100 & 0.2400 & 0.3800 \\
0.1000 & 0.2900 & 0.2210 & 0.3710 \\
0.3000 & 0.3700 & 0.1909 & 0.3480 \\
0.1000 & 0.3100 & 0.2440 & 0.3870 \\
0.1000 & 0.3000 & 0.2250 & 0.3770 \\
0 & 0.3000 & 0.2340 & 0.3810
\end{bmatrix}
$$

to be continued

$$netinv.\mathbf{IW} = \begin{bmatrix} 0.1000 & 0.3100 & 0.2420 & 0.3890 \\ 0.1000 & 0.3100 & 0.2305 & 0.3860 \\ 0.1000 & 0.3200 & 0.2918 & 0.3960 \\ 0.1000 & 0.2900 & 0.2214 & 0.3740 \\ 0 & 0.3500 & 0.2436 & 0.3880 \\ 0.3000 & 0.3300 & 0.2201 & 0.3830 \\ 0.1000 & 0.3300 & 0.2134 & 0.3960 \\ 0.1000 & 0.2900 & 0.2118 & 0.3790 \\ 0.3000 & 0.3000 & 0.2576 & 0.3792 \\ 0.1000 & 0.3300 & 0.2326 & 0.3790 \\ 0.3000 & 0.3100 & 0.2617 & 0.3710 \\ 0.3000 & 0.2900 & 0.2519 & 0.3670 \\ 0.1000 & 0.3200 & 0.2250 & 0.3830 \\ 0.1000 & 0.3400 & 0.2230 & 0.3780 \\ 0.3000 & 0.3000 & 0.2571 & 0.3760 \end{bmatrix}$$

Equation 7-7 System Inverse Model -- Input Weight Matrix: netinv.IW

$$netinv.\mathbf{LW} = 10^7 * \begin{bmatrix} 0.3541 & 0.2785 & 0.4137 & 0.0906 & -0.5972 & -0.0486 \\ -2.0115 & 0.9122 & -0.1695 & 0.0222 & -0.5286 & 0.0113 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7067 & -1.1526 & -0.4944 & -0.1645 & 1.2772 & 0.0710 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.6783 & 0.3931 & 0.2929 & 0.0923 & -0.4942 & -0.0471 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7395 & -1.1730 & -0.7923 & -0.2080 & 1.6423 & 0.0936 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.6160 & 1.2174 & 0.8533 & 0.2233 & -1.7281 & -0.1023 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4066 & -0.0940 & 0.0598 & 0.0117 & 0.0037 & -0.0051 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5632 & -0.2463 & -0.2917 & -0.0756 & 0.4181 & 0.0409 \\ 0.5740 & -0.2440 & -0.0248 & -0.0171 & 0.2315 & 0.0008 \\ -0.3835 & 0.6155 & 0.2641 & 0.0843 & -0.6676 & -0.0397 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2074 & -0.3707 & -0.3498 & -0.0878 & 0.5808 & 0.0451 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1.1121 & 0.4798 & 0.0749 & 0.0324 & -0.4631 & -0.0073 \\ -0.0476 & 0.0744 & 0.0208 & 0.0092 & -0.0703 & -0.0036 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1.2217 & -0.1802 & 0.3991 & 0.0612 & -0.2214 & -0.0452 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

to be continued

$$
\textit{netinv}.\mathbf{LW} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-0.4524 & 0.1025 & -0.0668 & -0.0061 & -0.0347 & 0.0097 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0.7011 & -0.2996 & -0.1231 & -0.0348 & 0.3879 & 0.0088 \\
-0.9352 & 0.0655 & -0.1459 & -0.0326 & 0.0400 & 0.0233 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-0.0072 & 0.4576 & 0.1302 & 0.0567 & -0.3838 & -0.0292 \\
0.0717 & -0.2784 & -0.1109 & -0.0347 & 0.2683 & 0.0206 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0.8824 & -0.5576 & 0.0602 & -0.0228 & 0.3391 & 0.0029 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Equation 7-8 System Inverse Model -- Layer Weight Matrix: netinv.LW

$$netinv.\mathbf{b}_1 = \begin{bmatrix} 0.1041 \\ 0.1041 \end{bmatrix}$$

to be continued

$$net.\mathbf{b}_1 = \begin{bmatrix} 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \\ 0.1041 \end{bmatrix}$$

Equation 7-9 System Inverse Model -- Basis Vector 1: netinv.$\mathbf{b}_1$

$$netinv.\mathbf{b}_2 = 10^5 * \begin{bmatrix} 1.4494 \\ -1.2436 \\ 1.4494 \\ 0.1055 \\ -0.7041 \\ -0.0890 \end{bmatrix}$$

Equation 7-10 System Inverse Model -- Basis Vector 2: netinv.$\mathbf{b}_2$

7.2.2 Controller Test

Both the process model and the inverse model are built based on the same training data trX & trY. Let us see how well the original data can be recovered though the system inverse model and the system model.

Test steps:

(1) Given the process model 'net' and the inverse model 'netinv', we can get the process model output as y_model = sim(net, teX)

(2) Then using the processs model output y_model to test the inverse model, x_invmodel = sim (netinv, y_model)

(3) Compare the difference between teX and x_invmodel.

(4) Calculate the error distribution for each input parameter: <2%, <5%, <8%, <10%, >10%, shown in the following figure.

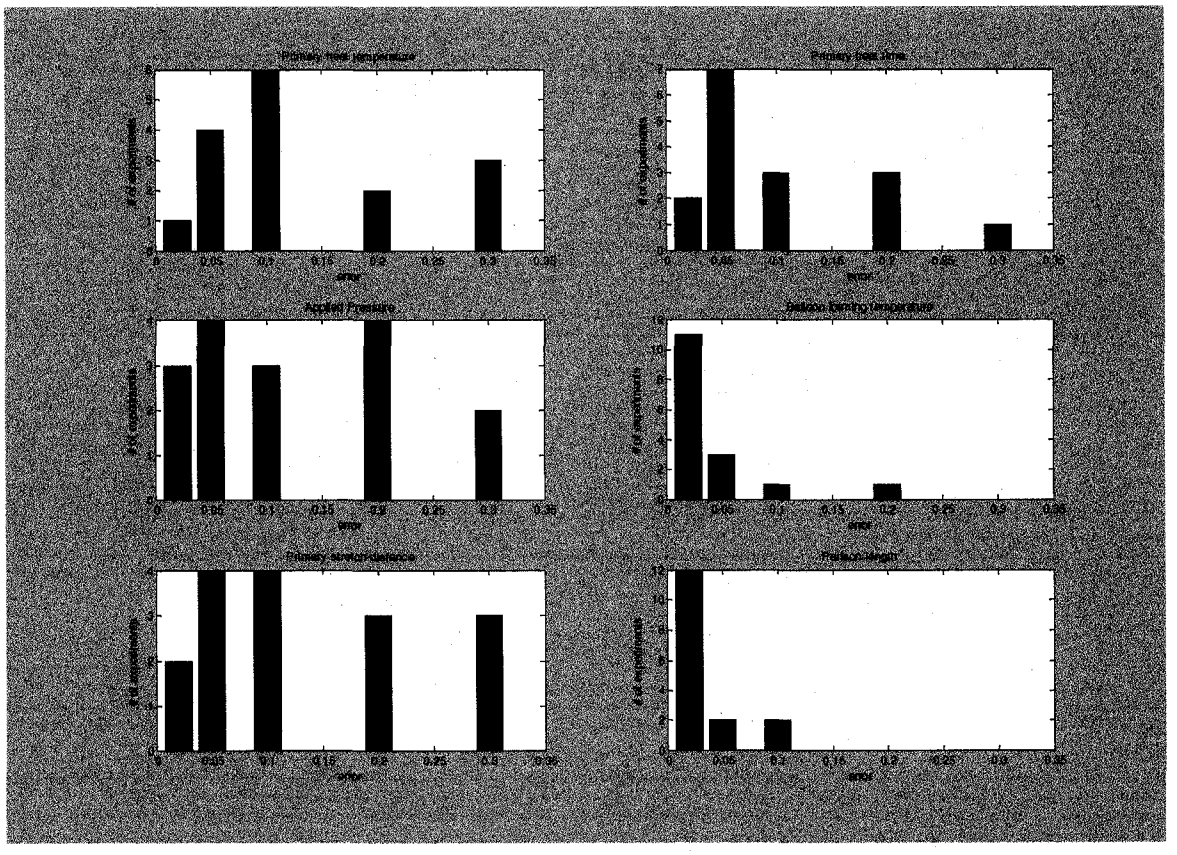


Figure 7-2 Model Recover Error

From the above figure, with the process model and the inverse model, the original data can be recovered with only a small error in most cases. For the balloon forming

temperature and parison length, the errors between the original data and the outputs from the process model and the inverse mode almost fall within 10%.

7.3 Predicting the Input Parameters by the Inverse Model

The inverse model may be used directly to predict the closest input parameters to the given output parameters. Figure 7-3 shows the error distributions between the actual values and the input parameters given by the inverse model. And the test results are as follows:

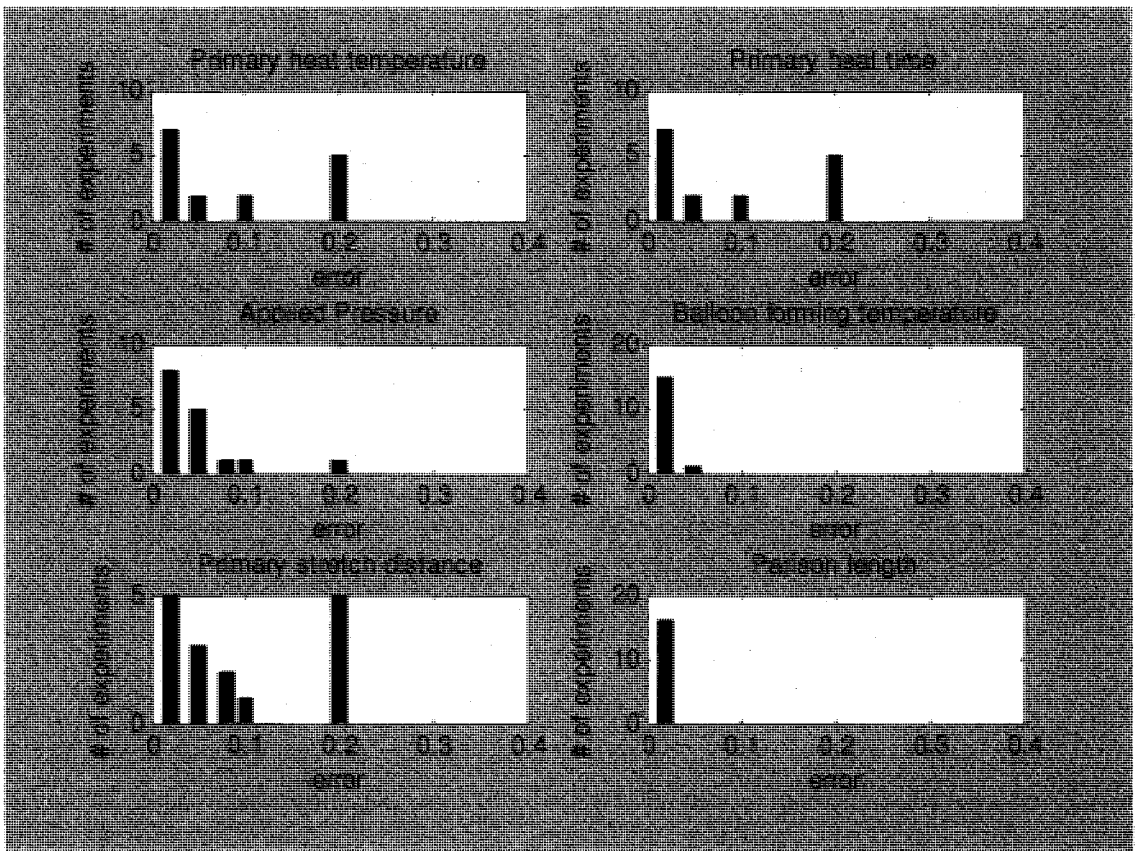


Figure 7-3 Error Distribution for the Inverse Model

In the above figure, it can be seen that the process inverse model works well in most cases and most of the test errors are within 10%.

7.4 Implementing a Cycle-to-cycle Controller based on Offline Learning

Assume that

$\widetilde{\mathbf{P}}$: Process Model

$\widetilde{\mathbf{P}}^{-1}$: Inverse Model

$\mathbf{P}$: Process

$\mathbf{y}_{des}$: the desired output

$\mathbf{y}_{mod}$: the process model output

$\mathbf{y}_{modinv}$: the inverse model input

$\mathbf{y}_{act}$: the actual process output

$\mathbf{y}_{err}$: the difference between the actual process output and the process model output

$\mathbf{x}_{mod}$: the process model input

net: the RBF neural network process model

netinv: the RBF neural network process inverse model

The purpose of implementing an offline controller is to increase the prediction accuracy while saving time to find out a set of input parameters to start with by making good use of the previous experimental data. The offline cycle-to-cycle control algorithm implemented in this thesis is as follows:

(1) use ¾ of the whole experimental data as training data and ¼ as test data;

(2) train both process model $\widetilde{\mathbf{P}}$ and inverse model $\widetilde{\mathbf{P}}^{-1}$ based on the training data set;

(3) Given a desired output $\mathbf{y}_{des}$, put it into the inverse model $\tilde{\mathbf{P}}^{-1}$ and get an input $\mathbf{x}_{mod}$:

$$\mathbf{x}_{mod} = \tilde{\mathbf{P}}^{-1}\mathbf{y}_{des}$$

Equation 7-11 Predicting the Input for a given output

(4) Put $\mathbf{x}_{mod}$ into the process model $\tilde{\mathbf{P}}$ and calculate the process model output $\mathbf{y}_{mod}$:

$$\mathbf{y}_{mod} = \tilde{\mathbf{P}}\mathbf{x}_{mod}$$

Equation 7-12 Predicting the output for a given input

(5) Put $\mathbf{x}_{mod}$ into the training dataset of actual process $\mathbf{P}$ and find out the closest input to $\mathbf{x}_{mod}$ and the related output $\mathbf{y}_{act}$

(6) Calculate the difference between the closest actual output $\mathbf{y}_{act}$ and the process model output $\mathbf{y}_{mod}$:

$$\mathbf{y}_{err} = \mathbf{y}_{act} - \mathbf{y}_{mod}$$

Equation 7-13 Error between the Actual Closest Output and the Model Output

(7) Add the difference $\mathbf{y}_{err}$ to the desired output $\mathbf{y}_{des}$ for the next iteration until this difference is minimized. The optimization process is

Repeat

$$\mathbf{y}_{modinv} = \mathbf{y}_{des} + k * \mathbf{y}_{err}$$

$$\mathbf{x}_{mod} = \tilde{\mathbf{P}}^{-1}\mathbf{y}_{modinv}$$

$$\mathbf{y}_{mod} = \tilde{\mathbf{P}}\mathbf{x}_{mod}$$

$$\mathbf{y}_{err} = \mathbf{y}_{act} - \mathbf{y}_{mod}$$

$$\left(\begin{array}{l} \mathbf{y}_{err} = \mathbf{y}_{act} - \tilde{\mathbf{P}}\mathbf{x}_{mod} = \mathbf{y}_{act} - \tilde{\mathbf{P}}\tilde{\mathbf{P}}^{-1}\mathbf{y}_{modinv} = \mathbf{y}_{act} - \tilde{\mathbf{P}}\tilde{\mathbf{P}}^{-1}\left(\mathbf{y}_{des} + k * \mathbf{y}_{err}\right) \\ \\ \Rightarrow \mathbf{y}_{err} = \dfrac{\mathbf{y}_{act} - \tilde{\mathbf{P}}\tilde{\mathbf{P}}^{-1}\mathbf{y}_{des}}{1 + \tilde{\mathbf{P}}\tilde{\mathbf{P}}^{-1}k} \end{array} \right)$$

Until $\mathbf{y}_{err}$ is minimized.

Equation 7-14 Algorithm for Minimizing the Error

(8) Then, the process model input $\mathbf{x}_{mod}$ is the input we want to find out.

Please note that in this method, the gain k is different from the gains that are commonly used in the conventional control methods. It should not be more than 1 because the each values of the input to the inverse neural network model $\mathbf{y}_{modinv} = \mathbf{y}_{des} + k * \mathbf{y}_{err}$ should be less than one, so if K*err is made to be more than 1, then the NN model does not work well. Therefore, in this thesis, in order to restrict the input values to the model in a unit circle, that is less than 1, k is chosen to be 0.01 by experiments.

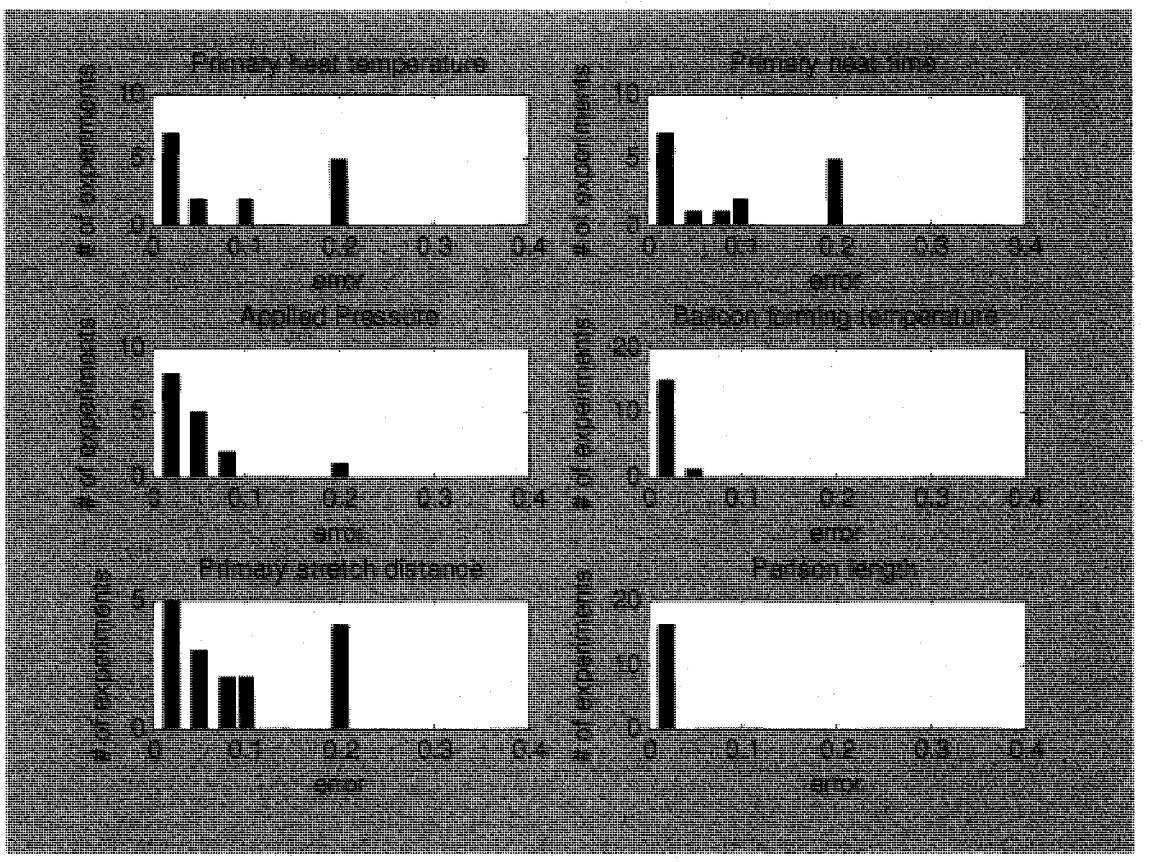


Figure 7-4 Input Prediction Error Distribution after Implementing a Learning Algorithm

In the above figure, it can be seen that most of the test errors fall within 10% due to the iterative offline cycle-to-cycle control algorithm performed on the controller. Comparing Figure 7-4 (with the iterative learning algorithm) with Figure 7-3 (without the iterative learning algorithm), it is clear that the prediction accuracy is increased after implementing an iterative offline learning algorithm and some of them are kept even within 5%.

Chapter 8

CONCLUSION

The nonlinear model of the balloon forming process introduced in this thesis works well for building the relationship between the input parameters and output parameters for the three batches of tubing. Moreover, the inverse model has a good performance as a controller in predicting the input parameters if the desired outputs are given. With an offline cycle-to-cycle learning algorithm, the input prediction accuracy for recommending the input parameters is increased. The implementation of this offline cycle-to-cycle controller can help reduce the time spent on the experiments and reduce the number of trials, and as a result, reduce the material consumption.

There is, however, one problem that exists and needs to be solved. If the desired outputs are similar to the ones that are in the database, the algorithm could work well. But if there is a big difference between the desired outputs and the data in the database, then this algorithm may fail to work because both the system model and the controller cannot find a previous experiment in the database that can provide the new tubing's information it requires. Another potential problem is that it may fail to work on a new batch of tubing because the new batch of tubing may have some difference from these three batches used for modeling and the system model does not include much information about the original tube material.

8.1 Recommendations for Future Work

The proposed model and controller based on the RBF neural network can produce satisfactory results when restricted to the three batches of tubing because the model does not consider the parameters about the original materials such as the tube inner diameter, tube outer diameter, etc. In order to improve this, many experiments need to be done to test the significance of these parameters affecting the outputs results. The material parameters with significant effects should be considered and included in the model of the balloon forming process.

Appendix A

The following pages contain the MATLAB files used for angioplasty balloon modeling and control based on RBF neural networks.

```
% Yan Chen
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% This program builds the process model based on RBF neural
% network and BP network
% Calculate the training error and test error separately
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%


close all
clear all
load seedsfile
X = [
70  70  39  134 10.5    130 18  90  14  0.508   0.2032;
70  55  39  140 10.5    130 18  90  14  0.508   0.2032;
70  55  39  128 10.5    130 18  90  14  0.508   0.2032;
55  55  39  140 10.5    130 18  90  14  0.508   0.2032;
62  55  42  140 10.5    130 18  90  14  0.508   0.2032;
62  55  42  128 10.5    130 18  90  14  0.508   0.2032;
62  55  36  140 10.5    130 18  90  14  0.508   0.2032;
62  55  36  128 10.5    130 18  90  14  0.508   0.2032;
62  55  39  134 10.5    130 18  90  14  0.508   0.2032;
62  55  39  134 10.5    130 18  90  14  0.508   0.2032;
62  55  39  134 10.5    130 18  90  14  0.508   0.2032;
70  70  42  140 10.5    130 18  90  14  0.508   0.2032;
60  60  10  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  15  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  20  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  25  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  27  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  29  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  31  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  33  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  37  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  39  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  41  130 13.5    135 18.5    100 15  0.4953  0.20955;
35  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
40  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
45  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
50  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
55  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
65  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
70  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
75  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
80  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
```

```
85  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  35  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  40  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  45  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  50  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  55  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  65  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  70  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  75  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  80  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 8.5     135 18.5    100 15  0.4953  0.20955;
60  60  35  130 9.5     135 18.5    100 15  0.4953  0.20955;
60  60  35  130 10.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 11.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 12.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 14.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 15.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 16.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 17.5    135 18.5    100 15  0.4953  0.20955;
80  70  40  130 15      130 16  90  18  0.4953  0.20955;
74  70  40  130 15      130 16  90  18  0.4953  0.20955;
76  70  40  130 15      130 16  90  18  0.4953  0.20955;
78  70  40  130 15      130 16  90  18  0.4953  0.20955;
82  70  40  130 15      130 16  90  18  0.4953  0.20955;
80  65  40  130 15      130 16  90  18  0.4953  0.20955;
80  67.5    40  130 15  130 16  90  18  0.4953  0.20955;
80  72.5    40  130 15  130 16  90  18  0.4953  0.20955;
80  75  40  130 15  130 16  90  18  0.4953  0.20955;
80  70  37  130 15  130 16  90  18  0.4953  0.20955;
80  70  38  130 15  130 16  90  18  0.4953  0.20955;
80  70  39  130 15  130 16  90  18  0.4953  0.20955;
];
Y = [
2   32  23.44   3.81;
1   32  21.34   3.84;
1   33  23.26   3.79;
1   34  22.3    3.78;
1   33  21.34   3.96;
1   34  23.61   3.71;
2   33  21.27   3.82;
0   35  24.36   3.88;
0   32  21.5    3.87;
1   32  22.17   3.86;
0   34  22.05   3.9;
2   33  21.1    3.89;
4   50  15  3;
```

```
4   50  15  3;
4   50  15  3;
3   41  18.15   3.46;
3   37  19.09   3.48;
3   35  22.2    3.75;
2   33  21.17   3.6;
1   31  24.2    3.89;
0   30  23.2    3.84;
0   29  22.8    3.79;
1   28  23.15   3.94;
1   28  23.15   3.87;
3   33  22.01   3.83;
2   33  22.34   3.8;
2   32  22.17   3.84;
1   32  22.5    3.83;
1   31  23.05   3.86;
1   31  23.7    3.9;
2   30  24.17   3.92;
2   30  24.1    3.9;
2   28  24.47   3.89;
2   29  24.17   3.88;
2   33  25.18   3.95;
1   32  24.14   3.85;
1   31  24  3.8;
1   30  23.21   3.81;
0   30  23.4    3.81;
1   30  22.5    3.77;
1   29  22.14   3.74;
1   29  22.6    3.77;
1   29  22.1    3.71;
2   35  25.11   3.85;
2   35  24.15   3.87;
1   34  25.14   3.89;
1   33  24.15   3.9;
1   31  24.4    3.87;
0   30  22.7    3.8;
1   29  21.1    3.77;
1   29  21.18   3.79;
1   28  20.99   3.76;
3   30  25.76   3.792;
2   30  25.10   3.76;
1   28  26.86   3.83;
1   32  29.18   3.96;
3   29  25.19   3.67;
1   29  26.64   3.78;
2   28  26.19   3.95;
```

```
2   31  26.59   3.75;
3   30  25.71   3.76;
2   30  27.70   3.87;
3   31  26.17   3.71;
1   29  27.74   3.90;
];
X = X(:,1:6);

% Preprocess the data
X(:,1) = X(:,1)/10;
X(:,2) = X(:,2)/10;
X(:,3) = X(:,3)/10;
X(:,4) = X(:,4)/100;
X(:,5) = X(:,5)/10;
X(:,6) = X(:,6)/100;
Y(:,1) = Y(:,1)/10;
Y(:,2) = Y(:,2)/100;
Y(:,3) = Y(:,3)/100;
Y(:,4) = Y(:,4)/10;
[rowx,colx] = size(X);

% Generate the training data -- 3/4 of the whole
% Generate the test data -- 1/4 of the whole

N = rowx;
trN = N*3;
teN = N;
k = 0;

m = rowx/4;
m1 = m*3;
m2 = m;

for i = 0:m-1
    trX(i*3+1,:) = X(i*4+4,:);
    trY(i*3+1,:) = Y(i*4+4,:);
    trX(i*3+2,:) = X(i*4+3,:);
    trY(i*3+2,:) = Y(i*4+3,:);
    trX(i*3+3,:) = X(i*4+1,:);
    trY(i*3+3,:) = Y(i*4+1,:);
end;

for i = 1:m
    teX(i,:) = X(i*4-2,:);
    teY(i,:) = Y(i*4-2,:);
end;
```

```
[NS, INPUT] = size(trX);
[NS, OUT] = size(trY);
%
% RBF network
%
P = trX'
T = trY'
SPREAD = 1;
GOAL = 0.1;
net = newrb(P,T,GOAL,SPREAD)

%
% Feedforward network
%
alfa = 0.8;
xite = 0.2;

% define the number of neurons in the hidden layer
HN = 6;

% weight connection between the hidden layer and the output
w2 = sw2;
w2_1 = w2;
w2_2 = w2_1;

% weight connection between the input and the hidden layer
w1 = sw1;
w1_1 = w1;
w1_2 = w1;
dw1 = 0*w1;

% initialize the input and output values for the hidden
layer
I = zeros(HN,1);
Iout = zeros(HN,1);
FI = zeros(HN,1);

% initialize other parameters
k = 0;
E = 1;
accuary = 0.1;
while (E>=accuary)
    k = k+1;
    times(k) = k;
```

```
for s = 1:1:NS
    xs = trX;
    ys = trY;
    x = xs(s,:);
    for j = 1:1:HN
        I(j) = x*w1(:,j);
        Iout(j) = 1/(1+exp(-I(j)));
    end

    yl = w2'*Iout;
    yl = yl';

    el = 0;
    y = ys(s,:);
    for l = 1:1:OUT
        el = el + 0.5*(y(l)-yl(l))^2;
    end

    E = 0;
    if s==NS
        for s = 1:1:NS
            E = E+el;
        end
    end
    el = y -yl;

    w2 = w2_1+xite*Iout*el+alfa*(w2_1-w2_2);

    for j = 1:1:HN
        S = 1/(1+exp(-I(j)));
        FI(j) = S*(1-S);
    end

    for i = 1:1:INPUT
        for j = 1:1:HN
            tmp = 0;
            for n = 1:1:OUT
                tmp = tmp + el(n)*w2(j,n);
            end
            dw1(i,j) = xite*FI(j)*x(i)*tmp;
        end
    end
    w1 = w1_1+dw1+alfa*(w1_1-w1_2);

    w1_2 = w1_1;
    w1_1 = w1;
```

```
        w2_2 = w2_1;
        w2_1 = w2;
    end
    Ek(k) = E
end
figure(2);
plot(times,Ek,'r');
xlabel('times - k');
ylabel('Error - E');


[NS, INPUT] = size(teX);
[NS, OUT] = size(teY);
%
% RBF test
%
P = teX';
T = teY';
Yd = sim(net,P);
error = Yd-T;
error_RBF = mse(error)
%
% feedforward test
% define the number of neurons in the hidden layer
%
HN = 6;
for i = 1:1:NS
    for j = 1:1:HN
        I(i,j) = teX(i,:)*w1(:,j);
        Iout(i,j) = 1/(1+exp(-I(i,j)));
    end
end
y = w2'* Iout';
y = y';
error = abs(y - teY);
error_PB = mse(error)

k = 0;
Yd = Yd';
for i = 1:NS
    k = k+1;
    times(k) = k;
end
figure(3);
hold on
```

```
plot(times, teY(:,1)*10, 'r*', times, round(Yd(:,1)*10),
'bo', times, round(y(:,1)*10), 'k+')
xlabel('# of experiment');
title('Quality');
ylabel('quality');
legend('Actual output','RBF output','BP output');

figure(4);
hold on
plot(times, teY(:,2)*100, 'r*', times, Yd(:,2)*100, 'bo',
times, y(:,2)*100, 'k+')
title('Wall-thickness');
xlabel('# of experiment');
ylabel('Wall-thickness');
legend('Actual output','RBF output','BP output');

figure(5);
hold on
plot(times, teY(:,3)*100, 'r*', times, Yd(:,3)*100, 'bo',
times, y(:,3)*100, 'k+')
title('Burst pressure');
xlabel('# of experiment');
ylabel('Burst pressure');
legend('Actual output','RBF output','BP output');

figure(6);
hold on
plot(times, teY(:,4)*10, 'r*', times, Yd(:,4)*10, 'bo',
times, y(:,4)*10, 'k+')
title('Compliance');
xlabel('# of experiment');
ylabel('Compliance');
legend('Actual output','RBF output','BP output');

for i = 1:1:NS
    for j = 1:1:OUT
        errorRBF_quality(i) = abs(round(Yd(i,1)*10)-
teY(i,1)*10);
        errorRBF_WT(i) = 100*abs(Yd(i,2)-teY(i,2))/Yd(i,1);
        errorRBF_BP(i) = 100*abs(Yd(i,3)-teY(i,3))/Yd(i,3);
        errorRBF_CP(i) = 100*abs(Yd(i,4)-teY(i,4))/Yd(i,4);
        errorBP_quality(i) = abs(round(y(i,1)*10)-
teY(i,1)*10);
        errorBP_WT(i) = 100*abs(y(i,2)-teY(i,2))/y(i,1);
        errorBP_BP(i) = 100*abs(y(i,3)-teY(i,3))/y(i,3);
        errorBP_CP(i) = 100*abs(y(i,4)-teY(i,4))/y(i,4);
```

```
    end

end

errorRBF(1,:) = errorRBF_quality;
errorRBF(2,:) = errorRBF_WT;
errorRBF(3,:) = errorRBF_BP;
errorRBF(4,:) = errorRBF_CP;
errorBP(1,:) = errorBP_quality;
errorBP(2,:) = errorBP_WT;
errorBP(3,:) = errorBP_BP;
errorBP(4,:) = errorBP_CP;

EN = 6;
count_RBF = zeros(OUT,EN);
count_BP = zeros(OUT,EN);
for i = 1:1:NS
    for j = 2:1:OUT
        if errorRBF(j,i) < 3
            count_RBF(j,1) = count_RBF(j,1)+1;
        elseif errorRBF(j,i) < 5
            count_RBF(j,2) = count_RBF(j,2)+1;
        elseif errorRBF(j,i) < 8
            count_RBF(j,3) = count_RBF(j,3)+1;
        elseif (errorRBF(j,i) < 10)
            count_RBF(j,4) = count_RBF(j,4)+1;
        elseif (errorRBF(j,i) < 15)
            count_RBF(j,5) = count_RBF(j,5)+1;
        else
            count_RBF(j,6) = count_RBF(j,6)+1;
        end;
        if (errorBP(j,i) < 2)
            count_BP(j,1) = count_BP(j,1)+1;
        elseif (errorRBF(j,i) < 5)
            count_BP(j,2) = count_BP(j,2)+1;
        elseif (errorRBF(j,i) < 8)
            count_BP(j,3) = count_BP(j,3)+1;
        elseif (errorRBF(j,i) < 10)
            count_BP(j,4) = count_BP(j,4)+1;
        elseif (errorRBF(j,i) < 15)
            count_BP(j,5) = count_BP(j,5)+1;
        else
            count_BP(j,6) = count_BP(j,6)+1;
        end;
    end
end
```

```
for i = 1:1:NS
    if errorRBF_quality(i) == 0
        count_RBF(1,1) = count_RBF(1,1)+1;
        count_BP(1,1) = count_BP(1,1)+1;
    elseif errorRBF_quality(i) == 1
        count_RBF(1,2) = count_RBF(1,2)+1;
        count_BP(1,2) = count_BP(1,2)+1;
    elseif errorRBF_quality(i) == 2
        count_RBF(1,3) = count_RBF(1,3)+1;
        count_BP(1,3) = count_BP(1,3)+1;
    else
        count_RBF(1,4) = count_RBF(1,4)+1;
        count_BP(1,4) = count_BP(1,4)+1;
    end;
end

errorquality(1,:) = count_RBF(1,:);
errorquality(2,:) = count_BP(1,:);
errorWT(1,:) = count_RBF(2,:);
errorWT(2,:) = count_BP(2,:);
errorBPr(1,:) = count_RBF(3,:);
errorBPr(2,:) = count_BP(3,:);
errorCP(1,:) = count_RBF(4,:);
errorCP(2,:) = count_BP(4,:);

xaxis = [0.02 0.05 0.08 0.10 0.15 0.20]
figure(7);
subplot(2,2,1)
bar(errorquality,'group');
title('quality');
subplot(2,2,2)
bar(errorWT,'group');
title('Wall Thickness');
subplot(2,2,3)
bar(errorBPr,'group');
title('Burst Pressure');
subplot(2,2,4)
bar(errorCP,'group');
title('Compliance');

figure(8);
subplot(4,2,1);
bar(0:5, count_RBF(1,:));
title('RBF quality');
xlabel('error');
ylabel('# of experiments');
```

```
subplot(4,2,2);
bar(0:5, count_BP(1,:));
title('feedforward quality');
xlabel('error');
ylabel('# of experiments');


subplot(4,2,3);
bar(xaxis, count_RBF(2,:));
title('RBF wall thickness');
xlabel('error');
ylabel('# of experiments');

subplot(4,2,4);
bar(xaxis, count_BP(2,:));
title('feedforward wall thickness');
xlabel('error');
ylabel('# of experiments');

subplot(4,2,5);
bar(xaxis, count_RBF(3,:));
title('RBF burst pressure');
xlabel('error');
ylabel('# of experiments');

subplot(4,2,6);
bar(xaxis, count_BP(3,:));
title('feedforward burst pressure');
xlabel('error');
ylabel('# of experiments');

subplot(4,2,7);
bar(xaxis, count_RBF(4,:));
title('RBF compliance');
xlabel('error');
ylabel('# of experiments');

subplot(4,2,8);
bar(xaxis, count_BP(4,:));
title('feedforward compliance');
xlabel('error');
ylabel('# of experiments');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% This program calculates the input parameters if given desired
% outputs based on the cycle-to-cycle control
% Calculate the estimate errors
% Calculate the inverse model estimate errors
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all
clear all

% System indentification based on RBF NN
%% Experimental Data Preprocessing
X = [
70  70  39  134 10.5    130 18  90  14  0.508   0.2032;
70  55  39  140 10.5    130 18  90  14  0.508   0.2032;
70  55  39  128 10.5    130 18  90  14  0.508   0.2032;
55  55  39  140 10.5    130 18  90  14  0.508   0.2032;
62  55  42  140 10.5    130 18  90  14  0.508   0.2032;
62  55  42  128 10.5    130 18  90  14  0.508   0.2032;
62  55  36  140 10.5    130 18  90  14  0.508   0.2032;
62  55  36  128 10.5    130 18  90  14  0.508   0.2032;
62  55  39  134 10.5    130 18  90  14  0.508   0.2032;
62  55  39  134 10.5    130 18  90  14  0.508   0.2032;
62  55  39  134 10.5    130 18  90  14  0.508   0.2032;
70  70  42  140 10.5    130 18  90  14  0.508   0.2032;
60  60  10  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  15  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  20  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  25  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  27  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  29  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  31  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  33  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  37  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  39  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  60  41  130 13.5    135 18.5    100 15  0.4953  0.20955;
35  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
40  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
45  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
50  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
55  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
65  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
70  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
75  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
80  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
85  60  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  35  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  40  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  45  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  50  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  55  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  65  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  70  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  75  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
60  80  35  130 13.5    135 18.5    100 15  0.4953  0.20955;
```

```
60  60  35  130 8.5     135 18.5    100 15  0.4953  0.20955;
60  60  35  130 9.5     135 18.5    100 15  0.4953  0.20955;
60  60  35  130 10.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 11.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 12.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 14.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 15.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 16.5    135 18.5    100 15  0.4953  0.20955;
60  60  35  130 17.5    135 18.5    100 15  0.4953  0.20955;
80  70  40  130 15      130 16  90  18  0.4953  0.20955;
74  70  40  130 15      130 16  90  18  0.4953  0.20955;
76  70  40  130 15      130 16  90  18  0.4953  0.20955;
78  70  40  130 15      130 16  90  18  0.4953  0.20955;
82  70  40  130 15      130 16  90  18  0.4953  0.20955;
80  65  40  130 15      130 16  90  18  0.4953  0.20955;
80  67.5    40  130 15  130 16  90  18  0.4953  0.20955;
80  72.5    40  130 15  130 16  90  18  0.4953  0.20955;
80  75  40  130 15      130 16  90  18  0.4953  0.20955;
80  70  37  130 15      130 16  90  18  0.4953  0.20955;
80  70  38  130 15      130 16  90  18  0.4953  0.20955;
80  70  39  130 15      130 16  90  18  0.4953  0.20955;
];
Y = [
2   32  23.44   3.81;
1   32  21.34   3.84;
1   33  23.26   3.79;
1   34  22.3    3.78;
1   33  21.34   3.96;
1   34  23.61   3.71;
2   33  21.27   3.82;
0   35  24.36   3.88;
0   32  21.5    3.87;
1   32  22.17   3.86;
0   34  22.05   3.9;
2   33  21.1    3.89;
4   50  15      3;
4   50  15      3;
4   50  15      3;
3   41  18.15   3.46;
3   37  19.09   3.48;
3   35  22.2    3.75;
2   33  21.17   3.6;
1   31  24.2    3.89;
0   30  23.2    3.84;
0   29  22.8    3.79;
1   28  23.15   3.94;
1   28  23.15   3.87;
3   33  22.01   3.83;
2   33  22.34   3.8;
2   32  22.17   3.84;
1   32  22.5    3.83;
1   31  23.05   3.86;
1   31  23.7    3.9;
2   30  24.17   3.92;
2   30  24.1    3.9;
```

```
2   28  24.47   3.89;
2   29  24.17   3.88;
2   33  25.18   3.95;
1   32  24.14   3.85;
1   31  24       3.8;
1   30  23.21   3.81;
0   30  23.4    3.81;
1   30  22.5    3.77;
1   29  22.14   3.74;
1   29  22.6    3.77;
1   29  22.1    3.71;
2   35  25.11   3.85;
2   35  24.15   3.87;
1   34  25.14   3.89;
1   33  24.15   3.9;
1   31  24.4    3.87;
0   30  22.7    3.8;
1   29  21.1    3.77;
1   29  21.18   3.79;
1   28  20.99   3.76;
3   30  25.76   3.792;
2   30  25.10   3.76;
1   28  26.86   3.83;
1   32  29.18   3.96;
3   29  25.19   3.67;
1   29  26.64   3.78;
2   28  26.19   3.95;
2   31  26.59   3.75;
3   30  25.71   3.76;
2   30  27.70   3.87;
3   31  26.17   3.71;
1   29  27.74   3.90;
];

X = X(:,1:6);
X(:,1) = X(:,1)/100;
X(:,2) = X(:,2)/100;
X(:,3) = X(:,3)/100;
X(:,4) = X(:,4)/1000;
X(:,5) = X(:,5)/100;
X(:,6) = X(:,6)/1000;

Y(:,1) = Y(:,1)/10;
Y(:,2) = Y(:,2)/100;
Y(:,3) = Y(:,3)/100;
Y(:,4) = Y(:,4)/10;
save datasim X Y

[rowx,colx] = size(X);
m = rowx/4;
m1 = m*3;
m2 = m;

for i = 0:m-1
```

```
    trX(i*3+1,:) = X(i*4+4,:);
    trY(i*3+1,:) = Y(i*4+4,:);
    trX(i*3+2,:) = X(i*4+3,:);
    trY(i*3+2,:) = Y(i*4+3,:);
    trX(i*3+3,:) = X(i*4+1,:);
    trY(i*3+3,:) = Y(i*4+1,:);
end;

for i = 1:m
    teX(i,:) = X(i*4-2,:);
    teY(i,:) = Y(i*4-2,:);
end;



%% Build the process model: net and inverse process model: netinv

[NS, INPUT] = size(trX);
[NS, OUT] = size(trY);
trX = trX';
trY = trY';
SPREAD = 8;
GOAL = 0.1;
net = network;
net = newrb(trX,trY,GOAL,SPREAD);
GOAL = 0.1;
netinv = newrb(trY,trX,GOAL,SPREAD);

[NS, INPUT] = size(teX);
[NS, OUT] = size(teY);
teX = teX';
teY = teY';

Yd = sim(net,teX);
error = Yd-teY;

mse(error)


%% Objective: Given a desired output, to find out the related inputs

eta = 0.00001;
ite = 100;
k = 0.01;
ymod_final = zeros(4,m);
xmod_final = zeros(6,m);

for i = 1:m
    ydes = teY(:,i);
    xmod = sim(netinv,ydes);
    ymod = sim(net,xmod);
    yerr = ydes - ymod;
```

```
    n = 0;
    while (sse(yerr) > eta)
        n = n+1;
        if (n > ite), break, end,
        ymodinv = ydes + k*yerr;
        if (ymodinv(1)>0.4)
            ymodinv(1) = 0.4;
        end
        if (ymodinv(1)<0)
            ymodinv(1) = 0;
        end
        if (ymodinv(2)>0.5)
            ymodinv(2) = 0.5;
        end
        if (ymodinv(2)<0.2)
            ymodinv(2) = 0.2;
        end
        if (ymodinv(3)>0.3)
            ymodinv(3) = 0.3;
        end
        if (ymodinv(3)<0.15)
            ymodinv(3) = 0.15;
        end
        if (ymodinv(4)>0.4)
            ymodinv(4) = 0.4;
        end
        if (ymodinv(4)<0.3)
            ymodinv(4) = 0.3;
        end
        xmod = sim(netinv, ymodinv);
        % to find the closest input in the database
        sum_error = 4;
        index = 1;
        for j = 1:46
            tx = trX(:,j);
            tmp = tx - xmod;
            if mse(tmp)<sum_error
                sum_error = mse(tmp);
                tmp_xmod = tx;
                index = j;
            end;
        end;
        ymod = sim(net,xmod);
        yact = trY(:,index);
        yerr = ymod - yact;
    end;
    ymod_final(:,i) = ymod;
    xmod_final(:,i) = xmod;
end;
%% bar chart for App I
x_error = zeros(6,m);
% <3%, <5%, <10%, >10%
x_errcnt = zeros(6,5);
x_error1 = teX - xmod_final;
for i = 1:6
```

```
    for j = 1:m
        x_error1(i,j) = abs(x_error1(i,j));
        x_error1(i,j) = x_error1(i,j)/teX(i,j);
        if x_error1(i,j) < 0.03
            x_errcnt(i,1) = x_errcnt(i,1) + 1;
        else
            if x_error1(i,j) < 0.05
                x_errcnt(i,2) = x_errcnt(i,2) + 1;
            else
                if x_error1(i,j) < 0.08
                    x_errcnt(i,3) = x_errcnt(i,3) + 1;
                else
                    if x_error1(i,j) < 0.10
                        x_errcnt(i,4) = x_errcnt(i,4) + 1;
                    else
                        x_errcnt(i,5) = x_errcnt(i,5) + 1;
                    end
                end
            end
        end
    end
end

xaxis = [0.02 0.05 0.08 0.10 0.20];
xaxis2 = [0 1 2 3 4];
yaxis = [0 5 10 15 20];
figure(1);
subplot(3,2,1);
bar(xaxis,x_errcnt(1,:));
title('Primary heat temperature');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,2);
bar(xaxis,x_errcnt(2,:));
title('Primary heat time');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,3);
bar(xaxis,x_errcnt(3,:));
title('Applied Pressure');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,4);
bar(xaxis,x_errcnt(4,:));
title('Balloon forming temperature');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,5);
bar(xaxis,x_errcnt(5,:));
title('Primary stretch distance');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,6);
bar(xaxis,x_errcnt(6,:));
title('Parison length');
```

```
xlabel('error');
ylabel('# of experiments');


error1_Y = ymod_final-teY;
error1_X = xmod_final-teX;
xerror1 = zeros(1,m);
for i=1:m
    xerror1(1,i) = mse(xmod_final(:,i));
end;
xerror1
mse(error1_X)

%% Control system - Approach II: direct control

xmod_final2 = zeros(6,m);
m = rowx/4;
for i=1:m
    ydes = teY(:,i);
    xmod2 = sim(netinv,ydes);
    xmod_final2(:,i) = xmod2;
end;
error2_X = xmod_final2-teX;
xerror2 = zeros(1,m);
for i=1:m
    xerror2(1,i) = mse(xmod_final2(:,i));
end;
xerror2
mse(error2_X)
%% Bar chart for the direct control
%% bar chart for App II
x_error = zeros(6,m);
% <3%, <5%, <10%, >10%
x_errcnt = zeros(6,5);
x_error = teX - xmod_final2;
for i = 1:6
    for j = 1:m
        x_error(i,j) = abs(x_error(i,j));
        x_error(i,j) = x_error(i,j)/teX(i,j);
        if x_error(i,j) < 0.03
            x_errcnt(i,1) = x_errcnt(i,1) + 1;
        else
            if x_error(i,j) < 0.05
                x_errcnt(i,2) = x_errcnt(i,2) + 1;
            else
                if x_error(i,j) < 0.08
                    x_errcnt(i,3) = x_errcnt(i,3) + 1;
                else
                    if x_error(i,j) < 0.10
                        x_errcnt(i,4) = x_errcnt(i,4) + 1;
                    else
                        x_errcnt(i,5) = x_errcnt(i,5) + 1;
                    end
                end
```

```
            end
        end
    end
end

xaxis = [0.02 0.05 0.08 0.10 0.20];
xaxis2 = [0 1 2 3 4];
figure(2);
subplot(3,2,1);
bar(xaxis,x_errcnt(1,:));
title('Primary heat temperature');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,2);
bar(xaxis,x_errcnt(2,:));
title('Primary heat time');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,3);
bar(xaxis,x_errcnt(3,:));
title('Applied Pressure');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,4);
bar(xaxis,x_errcnt(4,:));
title('Balloon forming temperature');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,5);
bar(xaxis,x_errcnt(5,:));
title('Primary stretch distance');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,6);
bar(xaxis,x_errcnt(6,:));
title('Parison length');
xlabel('error');
ylabel('# of experiments');

%% calculate how much the inverse model can recover the model
% input --> model --> output --> inv_model --> input

tempY = sim(net, teX);
tempX = sim(netinv, tempY);
Err = zeros(6,m);
ErrRate = zeros(6,5);
for i = 1:6
    for j = 1:m
        temp = abs(tempX(i,j)-teX(i,j));
        Err(i,j) = temp/teX(i,j);
        if Err(i,j) < 0.02
            ErrRate(i,1) = ErrRate(i,1)+1;
        else
            if Err(i,j) < 0.05
                ErrRate(i,2) = ErrRate(i,2)+1;
```

```
            else
                if Err(i,j) < 0.10
                    ErrRate(i,3) = ErrRate(i,3)+1;
                else
                    if Err(i,j) < 0.20
                        ErrRate(i,4) = ErrRate(i,4)+1;
                    else
                        ErrRate(i,5) = ErrRate(i,5)+1;
                    end
                end
            end
        end
    end
end

xaxis = [0.02 0.05 0.10 0.20 0.30];
figure(3);
subplot(3,2,1);
bar(xaxis,ErrRate(1,:));
title('Primary heat temperature');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,2);
bar(xaxis,ErrRate(2,:));
title('Primary heat time');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,3);
bar(xaxis,ErrRate(3,:));
title('Applied Pressure');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,4);
bar(xaxis,ErrRate(4,:));
title('Balloon forming temperature');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,5);
bar(xaxis,ErrRate(5,:));
title('Primary stretch distance');
xlabel('error');
ylabel('# of experiments');
subplot(3,2,6);
bar(xaxis,ErrRate(6,:));
title('Parison length');
xlabel('error');
ylabel('# of experiments');
%% process model validation
% input --> model --> output : calculate the error
tempY = sim(net, teX);
Err = zeros(4,m);
ErrRate = zeros(4,5);
for i = 1:4
    for j = 1:m
        if i == 1
```

```
            temp = round(abs(tempY(i,j)*10-teY(i,j)*10));
            Err(i,j) = temp;
            if Err(i,j) == 0
                ErrRate(i,1) =  ErrRate(i,1) + 1;
            else
                if Err(i,j) == 1
                    ErrRate(i,2) =  ErrRate(i,2) + 1;
                else
                    if Err(i,j) == 2
                        ErrRate(i,3) =  ErrRate(i,3) + 1;
                    else
                        if Err(i,j) == 3
                            ErrRate(i,4) =  ErrRate(i,4) + 1;
                        else
                            if Err(i,j) == 4
                                ErrRate(i,5) =  ErrRate(i,5) + 1;
                            end
                        end
                    end
                end
            end
        else
            temp = abs(tempY(i,j)-teY(i,j));
            Err(i,j) = temp/teX(i,j);
            if Err(i,j) < 0.02
                ErrRate(i,1) = ErrRate(i,1)+1;
            else
                if Err(i,j) < 0.05
                    ErrRate(i,2) = ErrRate(i,2)+1;
                else
                    if Err(i,j) < 0.10
                        ErrRate(i,3) = ErrRate(i,3)+1;
                    else
                        if Err(i,j) < 0.20
                            ErrRate(i,4) = ErrRate(i,4)+1;
                        else
                            ErrRate(i,5) = ErrRate(i,5)+1;
                        end
                    end
                end
            end
        end
    end
end

xaxis = [0.02 0.05 0.10 0.20 0.30];
xaxis2 = [0 1 2 3 4];
figure(4);
subplot(2,2,1);
bar(xaxis2,ErrRate(1,:));
title('Quality');
xlabel('error');
ylabel('# of experiments');
subplot(2,2,2);
bar(xaxis,ErrRate(2,:));
```

```
title('Wall Thickness');
xlabel('error');
ylabel('# of experiments');
subplot(2,2,3);
bar(xaxis,ErrRate(3,:));
title('Burst Pressure');
xlabel('error');
ylabel('# of experiments');
subplot(2,2,4);
bar(xaxis,ErrRate(4,:));
title('Compliance');
xlabel('error');
ylabel('# of experiments');
```

Bibliography

1- Balloon Angioplasty and Stents, Texas Heart Institute, Website:

http://www.texasheartinstitute.org/HIC/Topics/Proced/angioplasty.cfm

2- Abdou Elhendy, and Robert I. Hamby, *Balloon Angioplasty*, website:

http://heart.health.ivillage.com/balloonangioplasty/balloonangioplasty.cfm

3- Dominic Lalli, *Cycle-to-Cycle Control of the Angioplasty Balloon Fabrication Process*, Master's Thesis, Department of Electrical and Engineering, McGill University

4- Coronary Artery Disease, website:

http://www.nhlbi.nih.gov/health/dci/Diseases/Cad/CAD_WhatIs.html

5- PTCA or Balloon Angioplasty, website:

http://www.heartsite.com/html/ptca.html

6- How is an Angioplasty Balloon Made , Website:

http://www.answers.com/topic/angioplasty-balloon

7- Heart Disease, website:

http://www.mayoclinic.com/health/coronary-artery-disease/DS00064/DSECTION=1

8- Angioplasty Balloon, website:

http://www.madehow.com/Volume-6/Angioplasty-Balloon.html

9- Mark A. Saab, *Applications of High-Pressure Balloons in the Medical Device Industry*, Advance Polymers, Inc.

10- *Using Experimental Design to Improve the Quality and Burst Pressure of Angioplasty balloons*, IMI

11- Stupecky, Josef, President, Interface Associates Incorporated, Website: www.interfaceusa.com

12- Zoe Sarrat-Cave, *BFM Process Schematic*, IMI, Private Communication.

13- Zoe Sarrat-Cave, *DES Schematic*, IMI, Private Communication.

14- Dominic Lalli, *Modeling*, IMI, Private Communication.

15- Knut Sauerteig, and Michael Giese, *The effect of Extrusion and Blow Molding Parameters on Angioplasty Balloon Production*, Journal of Applied Medical Polymers, 1999, Vol.3, No.1

16- K. Swingler, *Applying Neural Networks*, Harcourt Brace & Company, Academic PTR, 1996

17- G. W. Irwin, K. Warwick, and K. J. Hunt, *Neural Network Applications in Control*, The Institution of Electrical Engineers, 1995

18- S. Haykin, *Neural Networks – A Comprehensive Foundation* , 2rd Edition, Upper Saddle River, HJ: Prentice Hall PTR, 1999

19- M. Nørgaard, O. Ravn, N. K. Poulsen, and L.K. Hansen, *Neural Networks for Modeling and Control of Dynamic Systems*, Springer, 2000

20- Jose C. Principe, Neil R. Euliano, and W. Curt Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations*, John Wiley & Sons, INC, 2000

21- Neural Network ToolBox User's Guide, Version 3, 1998, The MathWorks Inc.

22- Z.Y. Song, *Application Research of Information Fusion Technology of Multi-sensor in Level Measurement*, Proc.3rd Internatioal Conference on Mechine Learning & Cybernetics, pp.3511-3514, 2004

23- W. Zhao, Y. Liu, and Y. He, *A New Method to Improve Pressure Sensor Precision Based on RBF Neural Network, Chinese Journal of Sensors and Actuators*, Vol.17, No.4, pp: 640-642, 2004

24- Y. Chen, B. Boulet, *Data Fusion based on RBF neural network for Error Compensation in Resistance Strain Gauge Force Transducers*, SNDP 2007