"Be not afraid of going slowly, be
afraid of standing still."

- Old Chinese Proverb

University of Alberta

Strain Measurement of a Forming Process Using Digital Imaging

by

Hua Deng ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

in

Process Control

Department of Chemical and Materials Engineering

Edmonton, Alberta
Fall 2008

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-47202-6
Our file Notre référence
ISBN: 978-0-494-47202-6

献给我的父母:
邓日智，黄明华
感谢他们的爱;
感谢他们的支持和鼓励.

To my parents:
Deng Rizhi and Huang Minghua,
for their love, support and encouragement.

Abstract

The understanding of strain history during a metal forming operation is important as the continuous plastic deformation may alter yield stress of the finished product via work hardening. A non-invasive digital imaging procedure has been developed to quantify the local strain behavior during metal forming. From digital images, obtained during the forming process, discrete spatial data of the object profile was acquired. A B-spline approximation method with an optimal number of control points was then applied to obtain a continuous profile. The weights of the B-spline function were calculated using linear least square regression. The deformation strains were calculated from the curvature of the continuous profile, and a local averaging approach was used to smooth the fluctuations of the calculated curvature. The proposed procedure was applied to an electric resistance welded (ERW) pipe forming process, and validated using measured strain gauge data.

Acknowledgements

This thesis and the whole Master's experience could not have been possible without the help and support of many people. It is a pleasure to have the opportunity to express my gratitude for all of them.

I would like to extend my sincere gratitude to my supervisor, Dr. Amos Ben-Zvi, for his guidance, knowledge and support during my research and study. Working with him has been a great learning experience in my Master study.

I wish to express my thanks to Dr. J. Barry Wiskel and Dr. Hani Henein, for their innumerable suggestions on a variety of technical aspects of my work. My thanks also goes to Manuel D. Rieder and IPSCO Inc., for their valuable inputs and previous experimental work.

All my colleagues in the office 679B made it a convivial place to work. Hector, Venkat, Qi, and Kartik, many thanks for being your colleague. In particular, I would like to thank Hector for proofreading this thesis, and Venkat for helping me to start the work on B-spline.

My deepest gratitude goes to my family, past and present, for their unflagging love and wise education throughout my life. Thanks for providing the best possible environment for me to grow up and get educated. You have always been a source of strength and encouragement.

Contents

List of Figures

1
Introduction

In the pipe-rolling industry, accurate roll production is one of the major concerns for producers. Roll forming is a complex metal deformation process. The cyclic plastic deformation may alter yield stress of the finished product through work hardening or the Bauschinger effect (Wiskel *et al.*, 2004). However, measuring deformation in roll forming processes involves several challenges, including the large scale of the forming process, the in-plant environmental conditions, and the complexities of the arrangement and type of the forming equipments (Wiskel *et al.*, 2008).

Several techniques have been developed to measure or in addition to analyze the deformation imposed in forming systems. These techniques include: strain gauges (Rieder, 2003), grid measurements (Hsu, 2003), finite element method (FEM) (Yoshida *et al.*, 1995; Seki *et al.*, 1995), and digital imaging with the application of circle grid arrays. Both the strain gauges and grid measurements are invasive and relatively time consuming. Strain gauges provide only discrete forming information at the points where strain gauges are placed, while grid measurements are unavailable to provide the dynamic information. FEM simulation enable the designer to obtain very precise information about the deformation through the entire process without applying experiment. However, a full finite element computation may be computationally expensive and time-consuming.

Digital imaging is a suitable technique for analysis of roll forming processes. Although the strain measurement using digital imaging in combination with circle grid arrays is efficient and accurate (D.W.Manthey *et al.*, 1996), this method still has some disadvantages. First, marking and reading grids for roll forming are time consuming. Second, the grid may be damaged by the forming equipment. Given these challenges, the digital imaging measurement procedure presented in this work is conducted without the application of a grid.

1.1 Objectives

The aim of this work is to develop and implement an accurate and efficient digital imaging based numerical analysis method for the roll forming process. This method will be useful in the analysis of the geometric and mechanical properties of deformed materials. The specific objectives of this work are to:

1) Develop a non-invasive digital imaging procedure to quantify local strain behavior in roll forming;

2) Apply the proposed technique to an electric-resistance-welded (ERW) pipe forming process;

3) Select an effective and accurate numerical approximation method to describe the deformed-skelp geometry; and

4) Confirm the veracity of the proposed procedure by comparing the predicted strain values with the measured strain gauge data.

1.2 Thesis outline

In this thesis, a non-invasive digital imaging procedure has been developed to quantify the local strain behavior during a forming process. **Chapter 2** presents the background on the issues related to this work. These issues include: the importance and problem in roll forming study, the existing strain measurement techniques, the properties of the processed digital image data, and the available mathematical geometric modeling algorithms for the image data reconstruction. **Chapter 3** describes the proposed digital-imaging-based-strain-measurement procedure, uses an ERW pipe forming operation to illustrate this procedure, and validates the results with the strain gauge measurement data. **Chapter 4** investigates three different geometric modeling methods (one-dimensional, two-dimensional and trigonometric B-splines) in order to achieve an accurate description of the deformed-skelp geometry. **Chapter 5** summarizes the results of the present work, concludes the entire thesis, and recommends directions for future work. The MATLAB scripts of the B-spline approximation (1-D, 2-D, and trigonometric), the curvature calculation and smoothing, and the strain calculation used in this work are presented in **Appendix A**

This thesis has been presented in a paper-format according to the requirement of the Faculty of Graduate Studies and Research (FGSR), University of Alberta. In order to connect the materials in different chapters and at the same time ensuring completeness of individual chapters, there is some overlap between chapters.

References

D.W.Manthey, R.M.Pearce and D. Lee (1996). The need for surface strain measurement. *Metal Forming* **30**(5), 48–50.

Hsu, Q. C. (2003). Comparison of different analysis models to measure plastic strains on sheet metal forming parts by digital image processing. *International Journal of Machine Tools and Manufacture* **43**(5), 515–521.

Rieder, Manuel Dominique (2003). Mathematical modeling of the Bauschinger effect in ERW pipeforming. Master's thesis. University of Alberta.

Seki, Kazunori, Shinji Ida, Shin-ya Hayashi, Kenji Yamada, Shuichi Hamauzu and Matsuo Ataka (1995). Analysis of long steel product rolling by rigid-plastic finite element method. Technical Report 67. Nippon Steel Corporation.

Wiskel, J.B., H. Deng, M. Rieder, A. Ben-Zvi and H. Henein (2008). Digital imaging of an ERW pipe forming process. In: *Proceedings of IPC2008.* 7th International Pipeline Conference. ASME.

Wiskel, J.B., M. Rieder and H. Henein (2004). Kinematic behavior of micro alloyed steels under complex forming conditions. *Canadian Metallurgical Quarterly* **43**(1), 125–136.

Yoshida, Tohru, Matsuo Usuda and Koji Hashimoto (1995). 3-D FEM analysis of sheet metal deep drawability and stretchability. Technical Report 67. Nippon Steel Corporation.

2 Background

2.1 Roll forming study

Roll forming is a general term used to describe a large class of manufacturing processes by which a strip of sheet metal is deformed continuously into a desired shape (Walker and Pick, 1990). The flat strip of sheet metal, or skelp material, is uncoiled from a roll and passed through a series of forming rolls that bend and shape it into any desired shape. The forming process studied in the present work is the cage roll forming of thin walled pipe.

The geometry of the deformed sheet metal is a highly complex three dimensional shape. In a roll forming process, the complex skelp shape is difficult to control. Any error in the roll production will lead to scrapping rolls, re-machining rolls, and increasing set-up times (Halmos, 2006). These complexities also result in the unique strain histories during and between the forming steps. If the maximum strain occurring in the material exceeds a certain limit, various defects can be observed in the final section, such as edge waves and spring back (Halmos, 2006). If strain values are controlled within specified limits, section defects can be minimized. Therefore, it is important to develop an effective technique to measure or analyze these deformations.

2.2 Strain measurement and prediction techniques

Strain describes the amount of deformation in a body. When a body is deformed, points in that body are displaced. Strain must be defined in such a way that it excludes effects of rotation and translation (Hosford and Caddell, 2007). Strain evolution in a forming process may alter yield stress of the finished product, or may lead to failure (Wiskel *et al.*, 2004). The current strain measurement and prediction techniques for roll forming process include strain gauges (Rieder, 2003), grid marking (Vogel and Lee, 1990), finite element method (FEM) (Yoshida *et al.*, 1995; Seki *et al.*, 1995), and digital imaging with the application of circle grid arrays (Hsu, 2003).

2.2.1 Strain gauges

A strain gauge is a device used to measure the strain of an object. The gauge is attached to the object by a suitable adhesive. As the object is deformed the gauge is also deformed, causing the change of electrical resistance. Using strain gauges to measure the strain during roll forming can obtain dynamic strain measurements accurately (Wiskel *et al.*, 2008).

Since the location of the maximum strain is unknown prior to the measurement, it is difficult to use strain gauges to measure the maximum strain value or other critical values of interest (Das, 2003). Therefore, strain gauges can only determine local strains at specific points, not the deformation over the whole object. Surface preparation procedures are time-consuming and relatively difficult under plant conditions (Rieder, 2003). In addition, very small voltage differences can generate significant noise in the measured data (Rieder, 2003). Moreover, the material on which the strain attached has a maximum service temperature of approximately 200°C. These factors limit the application of strain gauges.

Rieder (2003) performed the strain gauge tests on CSA Grade 359 pipes with an outer diameter of 114.3 mm and a wall thickness of 4.0 mm. The measured data by Rieder (2003) was used in the present work to validate the digital imaging procedure.

Strain gauges were applied to the inside surface of the skelp after the preform rolls and were oriented primarily perpendicular to the rolling direction (transverse). A set of strain gauges applied to the skelp surface is shown in Figure 2.1. After the gauges were applied, the mill was operated as usual. The skelp was slowly moved through the process while data was collected.

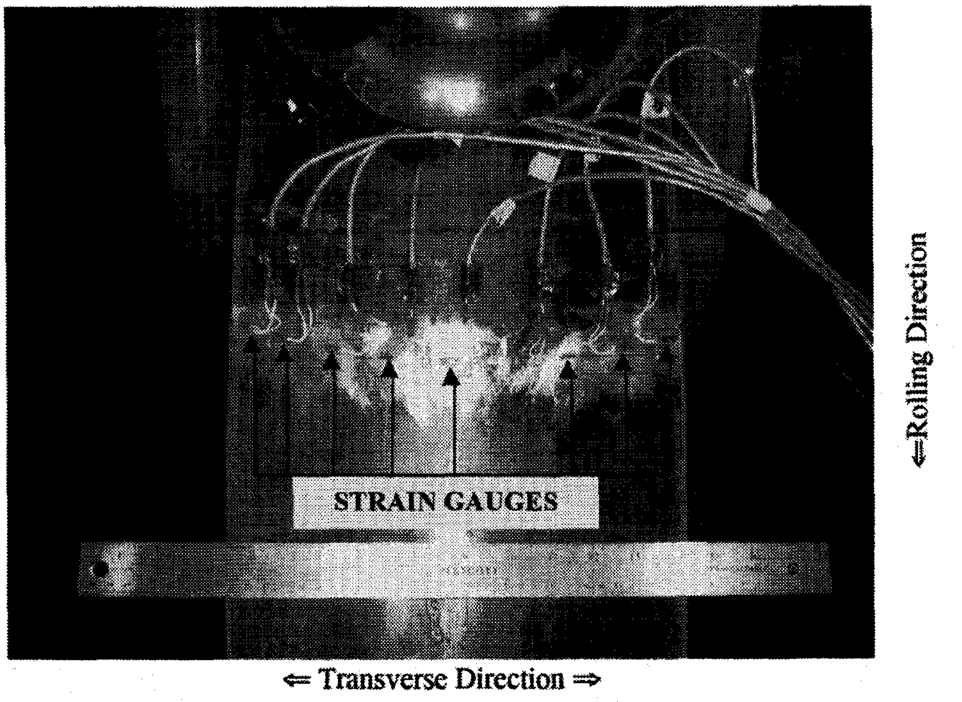


Figure 2.1: Strain gauges applied on the deformed skelp(Rieder, 2003)

The strain gauges were positioned for transverse measurements at 22.5°, 45°, 90°, 135°, 180°, 270°, 315°, and 352.5° on the completed cross section, shown in Figure 2.2. The strain gauge data obtained form this test is shown in **Appendix B**.

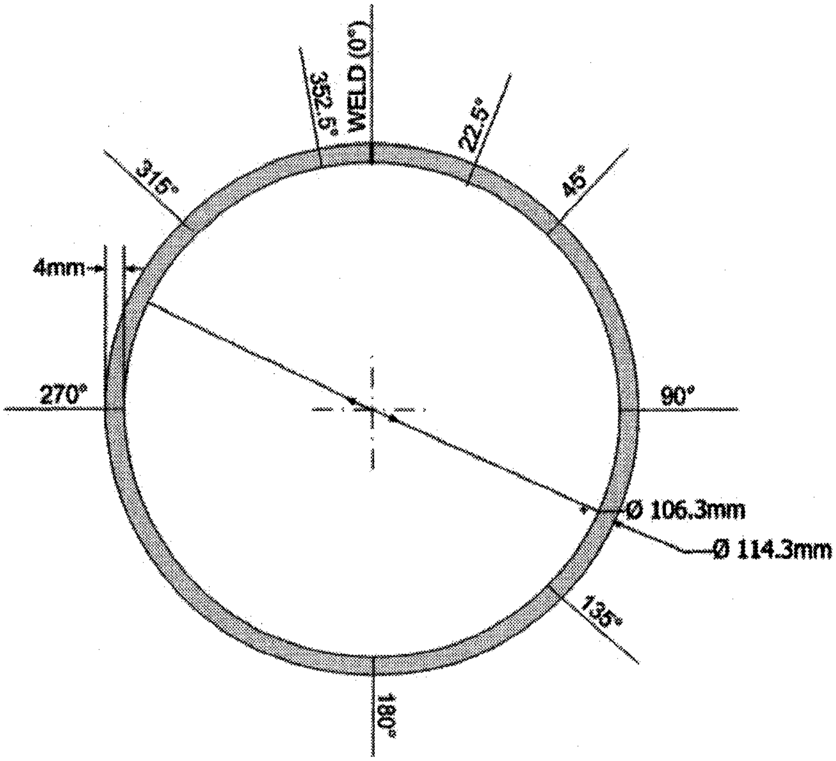


Figure 2.2: Strain gauge locations (Rieder, 2003)

2.2.2 Grid marking

Strain analysis by grid marking has been used effectively in metal forming (Vogel and Lee, 1990). By the grid marking method, the areas of high strain can be easily identified. The sheet is marked with the circle grids, before the forming process is carried out. After the sheet metal is deformed into the desired shape, the marked grids will deform into ellipses of different sizes (Figure 2.3). The strain distribution can be visualized and critical areas of strain can be found using the forming limit diagram (FLD) (Hsu, 2003).

Circle grid array (CGA) is relatively insensitive to measure low magnitude strain values in pipe forming (Wiskel *et al.*, 2008). Even under ideal conditions, the magnitude of strain errors using CGA measurement is around 0.7%. In pipe forming operations, the magnitude of strain values is usually less than 10%; therefore, the relative error is considerably high (approximately ±7%). In addition, only the total

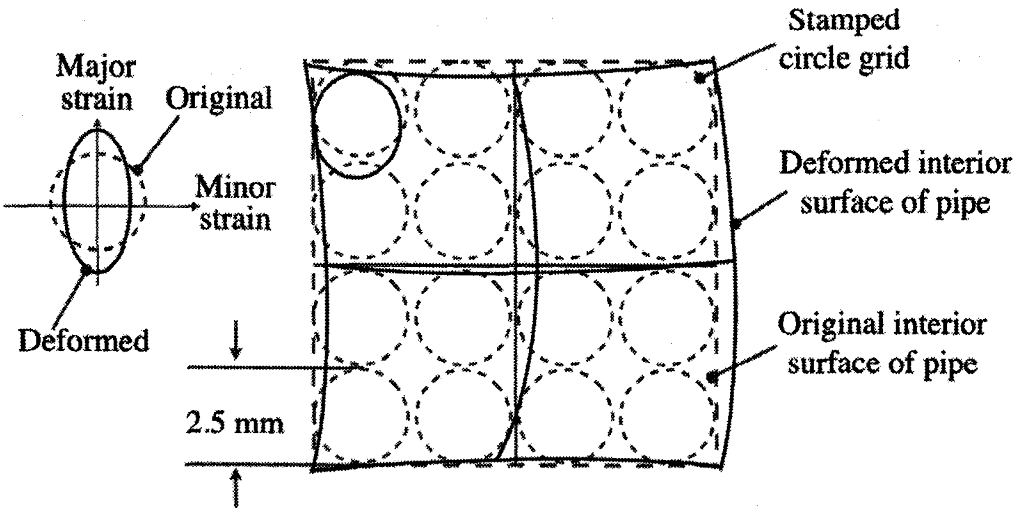

Major strain
Original
Minor strain
Deformed
2.5 mm
Stamped circle grid
Deformed interior surface of pipe
Original interior surface of pipe

Figure 2.3: Circle grid arrays, before and after deformation

strain at the end of the process can be obtained unless the grid is re-marked and examined at each stage in the process. Furthermore, this technique is difficult to apply under plant conditions and cannot provide dynamic strain information.

2.2.3 Finite element method

The finite element method (FEM) is a numerical technique for finding approximate solutions of field problems in solid and fluid mechanics, heat transfer and other areas. Most of these problems are described by partial differential equations (PDE) or integral equations over complex domains. The key steps of FEM are the division of a continuous and complex domain into a finite number of small and simple elements, and the application of simple algebraic equations on each element in order to approximate the equations to be studied.

Compared with the other direct measuring methods, finite element method (FEM) is a convenient technique to analyze strain distribution without performing costly and time-consuming plant trails or even small scale experimental tests (Alsamhan *et al.*, 2003; Rieder, 2003). Several finite element models (McClure and Li, 1995; Duggal *et al.*, 1996) and commercial FEM software packages (Senanayake *et al.*, 1994; McClure and Li, 1995) have been developed for strain analysis in metal forming. These FEM simulations enable the designer to get very precise information about stress and strain through the entire process.

A full finite element computation may be computationally expensive and time-consuming, as its iterative nature(Sedlmaier, 2005). Generally, the FEM simulation process needs to run on a separate computer, and it may take several days.

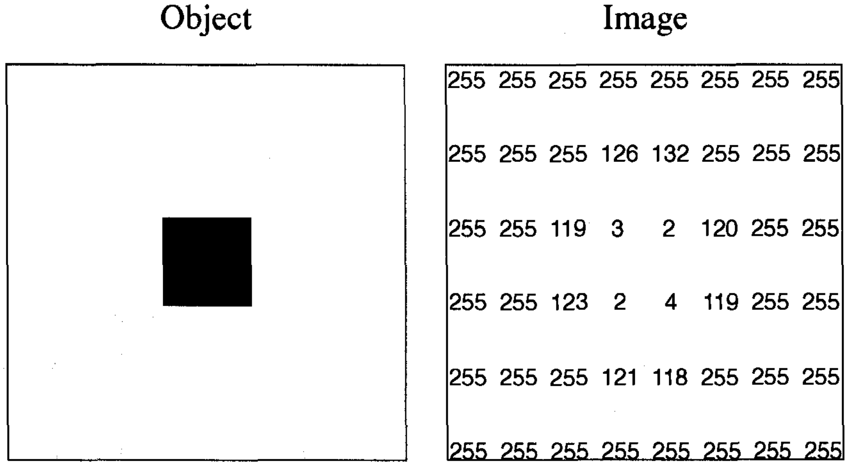
2.2.4 Digital imaging

Digital imaging is a technique that acquires a series of visual images from a physical object. Compared with traditional measurement devices, digital imaging techniques have several advantages. First, this technique is generally insensitive to environmental factors, for example high temperatures and electromagnetic fields. Second, the measuring process is non-invasive to the object being studied and is not limited to the position and type of forming equipments. Therefore, it can be applied either in close proximity or remotely to the object being considered. Third, the qualitative information included in digital images can be subsequently processed to provide the quantitative information as a continuous function, i.e. at every point on the studied object. These advantages of digital imaging make it a suitable technique for the strain measurement in the pipe roll forming.

2.3 Digital image

Computers have been used to process pictures for approximately fifty years (Green, 1988), and the technology of using digital imagery is continuing to develop rapidly. This technology has spread widely in diverse application fields for several reasons: first, a digital image has a wider dynamic range than the human eye or a photographic film; second, a single digital image can present a very large amount of information in a compact and easily interpretable form; third, a digital image can be processed and manipulated using methods that cannot be duplicated with non-digital technology (Green, 1988).

A digital image can be thought of as a matrix of numbers. A simple example of a digital image is shown in Figure 2.4 (Green, 1988). The scene is a black square on a white background. One possible digital representation of this scene is shown on the right of Figure 2.4. The digital representation of the scene is a sampled version of the continuous scene that is present in the object space (Green, 1988). Each point in the digital representation corresponds to an area in the object space, and a digital value is assigned at each point in the digital image that is related to the intensity of that area in the object space (Green, 1988).



255	255	255	255	255	255	255	255
255	255	255	126	132	255	255	255
255	255	119	3	2	120	255	255
255	255	123	2	4	119	255	255
255	255	255	121	118	255	255	255
255	255	255	255	255	255	255	255

Figure 2.4: Digital image example (Green, 1988)

The quality of the digital representation relates to two important parameters: the spatial sampling frequency and the intensity resolution.

The sampling frequency decides how many matrix elements will be used to represent the object. In Figure 2.4, the black square appears in the digital image as a 2-element by 2-element object (Green, 1988). The spatial resolution within a sampled digital image (finite) is less than the resolution of the actual continuous scene (infinite). As Green (1988) shows, the attempt to represent a continuous signal with a set of discrete points would cause the lack of sharp transition in the sampled digital image at the boundaries of the object. In Figure 2.4, the black-to-white transition at the edges of the black square is over a 3-pixel range, for example in line 3 from 255 to 199 to 3 (Green, 1988).

The digital intensity is represented by the value of individual component elements within the digital image which are referred to as picture elements (pixels). In Figure 2.4, digital intensity values can range from 0 to 255, with 0 representing black and 255 representing white (Green, 1988). The 4 pixels representing the uniform black square have digital intensity values of 3, 2, 2, and 4. Most systems that provide sampled digital representations of a scene introduce random noise into the sampled image (Green, 1988). If no noise was introduced during the sampling process, the digital intensity values of the 4 pixels would be the same (Green, 1988).

A higher sampling frequency and a higher intensity resolution would generate a higher quality image of the same scene, but a larger size of digital file and a more expensive equipment cost. The aim of the present work is neither to find the optimal sampling frequency nor the optimal intensity resolution, but rather to develop a procedure which is able to handle the constraints of the resolution and noise level. Digital images used in this work were acquired in JPEG format, at a resolution of 640×480 pixels with 24-bit color depth, using a standard digital camera (3Com HomeConnect PC Digital Camera, 3Com Corporation) (Rieder, 2003).

2.4 Geometric modeling

As described in Section 2.3, the information provided by the digital image is inherently discrete and noisy. Interpolation or other mathematical techniques need to be used to reconstruct a continuous and smooth geometry of the object from digital image data. The free-form curve or surface fitting are commonly used to generate an arbitrary curve or surface from measured data points. In the metal forming process, examples of free-form curves that have been used include cubic Bezier, B-spline and cubic B-spline.

Donovan (2002) proposed a numerical approximation method in the research of object recognition from a digital image. The proposed algorithm used cubic Bezier curves to represent identified the edges of the studied object in an image. The captured image was processed to find critical points on edges of the studied object. The least squares regression was used to represent these critical points as Bezier curves.

Cheng *et al.* (2002) developed a full-field speckle-pattern image-correlation method to directly determine the complete, two-dimensional deformation field. In the image correlation process, digital images were obtained using computer vision systems. A

B-spline function was used to represent the deformation field of the researched object through the entire image area.

Tong (1997) evaluated a whole-field, in-plane strain-mapping technique for monitoring plastic deformation patterns in sheet metals. Cubic B-splines and their first derivatives were used in computing the displacements and displacement gradients in this investigation. Displacement derivatives obtained directly by digital image correlation are highly oscillatory because of the high sensitivity to the displacement data noise. A method based on one-dimensional B-spline data smoothing routines was used to generalize the cross validation. The mean square prediction error criterion was used to automatically determine the optimum smoothness of the given data set.

2.4.1 Spline

Polynomial and spline functions are the two most common methods of representing arbitrary curves and surfaces in geometric modeling.

A polynomial function $f(x)$ is defined by

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0 \tag{2.1}$$

where n is a nonnegative integer and $a_0, a_1, a_2, \cdots, a_n$ are constant coefficients.

One of the attractive properties of polynomial functions is the smooth representation of a curve or surface. However, using just one polynomial segment to fit an arbitrary curve is often difficult. A high degree is required in order to satisfy a large number of constraints, or to accurately fit some complex shapes (Piegl and Tiller, 1997). For example, an $(n-1)^{th}$ degree is needed to pass a polynomial curve through n data points (Piegl and Tiller, 1997). However, high degree curves are inefficient to process and are numerically unstable (Piegl and Tiller, 1997). Moreover, there exist a number of curve types that cannot be represented precisely using polynomials, e.g. circles, ellipses, hyperbolas, etc. (Piegl and Tiller, 1997).

Spline is a special piecewise function defined by polynomials. A spline curve can be represented in the form (Piegl and Tiller, 1997):

$$C(u) = \sum_{i=0}^{n} f_i(u) P_i \tag{2.2}$$

where P_i are control points, and $\{f_i(u), 0 \leq i \leq m\}$ are piecewise polynomial basis functions with the desired degree and continuity (for a fixed knot sequence, $U = \{u_i\}$).

The continuity is determined by the basis functions. Therefore, the control points can be modified without altering the curve's continuity. Any of the standard polynomial forms can be used to represent $f_i(u)$ (Piegl and Tiller, 1997).

2.4.2 B-spline function

The B-spline function is one of the most efficient curve representation methods, and has been extensively used in computer-aided design and computer graphics. The B-spline functions possess very attractive properties such as convex hull property, transformation invariance, variation diminishing property and local support (Piegl and Tiller, 1997).

Given $n+1$ control points $P_0, P_1, \cdots, P_n$ and a knot vector $U = [u_0, u_1, \cdots, u_m]$, a B-spline curve of order k (i.e. degree $k-1$) defined by these control points and knot vector is:

$$C(t_j) = \sum_{i=0}^{n} N_{i,k}(t_j) P_i \tag{2.3}$$

where $N_{i,k}(t_j)$ is a B-spline basis function of order k, and $\{t_j\}$ is a set of parameter values.

Each parameter t_j is corresponded to a data point D_j. The u_i's are called knots, and the half open interval $[u_i, u_{i+1})$ is the i^{th} knot span. $U = [u_0, \cdots, u_m]$, where $m = n+k$, is a nondecreasing sequence of real numbers (i.e. $u_i \leq u_{i+1}$, $i = 0, \cdots, m-1$).

There are a number of ways to define the B-spline basis functions, such as De Boor (1978) recursion formula, divided difference of the truncated power function, continuity conditions, etc. (Ding and Davies, 1987). In this work, two recursion relations respectively defined by De Boor (1978) and Sanchez-Reyes (1992) were implemented.

The i^{th} B-spline basis function of order k defined by the De Boor (1978) recurrence formula is:

$$\begin{aligned} N_{i,1}(t_j) &= \begin{cases} 1 & \text{if } u_i \leq t_j \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,k}(t_j) &= \frac{t_j - u_i}{u_{i+k-1} - u_i} N_{i,k-1}(t_j) + \frac{u_{i+k} - t_j}{u_{i+k} - u_{i+1}} N_{i+1,k-1}(t_j) \end{aligned} \tag{2.4}$$

Computation of a set of basis functions requires specification of a knot vector U and the order k. Once the degree is fixed, the function $N_{i,k}(t_j)$ is only determined by the knot vector. Modifying the position of one or more knots will change the association between basis functions and knot spans, and hence change the shape of the curve. Knot vectors can be uniform, non-periodic, and nonuniform (Anand, 1993). In this work, only non-periodic knot vectors were considered.

A non-periodic knot vector has repeated knot values at the ends with multiplicity equal to the order of the B-spline function k and internal knots equally spaced (Anand, 1993). The non-periodic knot vector provides basis functions defined in the complete parameter range, and the curve always interpolates the first and last control points (Figure 2.6). A non-periodic knot vector can be calculated by the following equation:

$$\begin{aligned} u_i &= 0 && \text{for} \quad i < k \\ u_i &= i - k - 1 && \text{for} \quad k \leq i \leq n \\ u_i &= n - k + 2 && \text{for} \quad i > n \end{aligned} \tag{2.5}$$

Given a parameter value t_j, which is in a knot span $[u_i, u_{i+1})$, $N_{i,1}(t_j)$ is the only non-zero basis function of order k on the i^{th} knot span $[u_i, u_{i+1})$. For $k > 1$, $N_{i,k}(t_j)$ is a linear combination of two B-spline basis functions of order $(k-1)$ in the i^{th} and $(i+1)^{th}$ knot spans. Therefore, $N_{i,k}(t_j)$ can be generated using a truncated triangular table (Figure 2.5). All derivatives of $N_{i,k}(t_j)$ exist in the interior of a knot span (Piegl and Tiller, 1997).

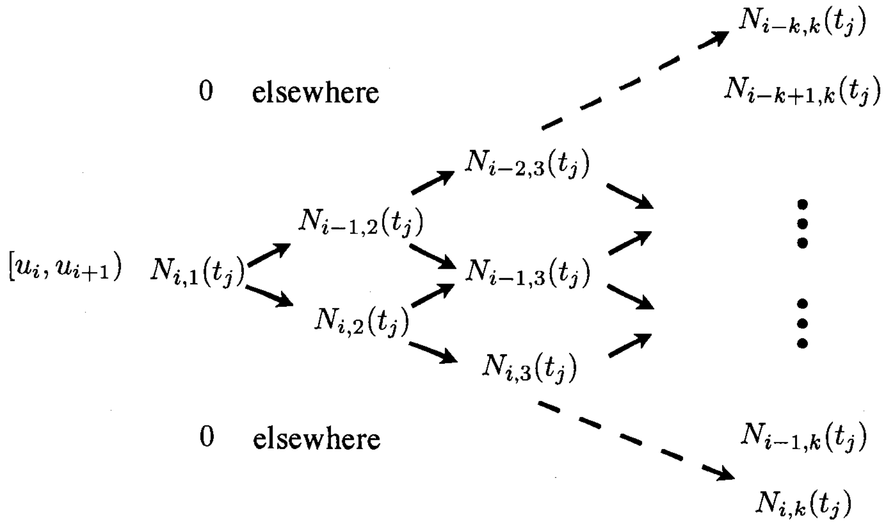


Figure 2.5: The truncated triangular table (Piegl and Tiller, 1997)

An example of a quadratic B-spline curve, interpolates a set of data points, is shown in Figure 2.6. The given data points are $D_0(0,1)$, $D_1(0.2,1.5)$, $D_2(0.5,1.8)$, $D_3(0.8,1.2)$, and $D_4(0.99,0.6)$, which correspond to a set of parameter values $t_0 = 0$, $t_1 = 0.2$, $t_2 = 0.5$, $t_3 = 0.8$, and $t_4 = 0.99$. This B-spline curve is defined by five control points $P_0, P_1, \cdots$, and P_4, and a non-periodic knot vector $[0, 0, 0, 1/3, 2/3, 1, 1, 1]$. A plot of the basis functions for this non-periodic quadratic B-spline over five control points is shown in Figure 2.7.

In summary, the shape of a B-spline curve can be changed by modifying one or more of these control parameters: the number and positions of control points, the number and positions of knots, and the order of the curve. The aim of the present work is modifying these control parameters correctly and efficiently in a B-spline curve fitting method.

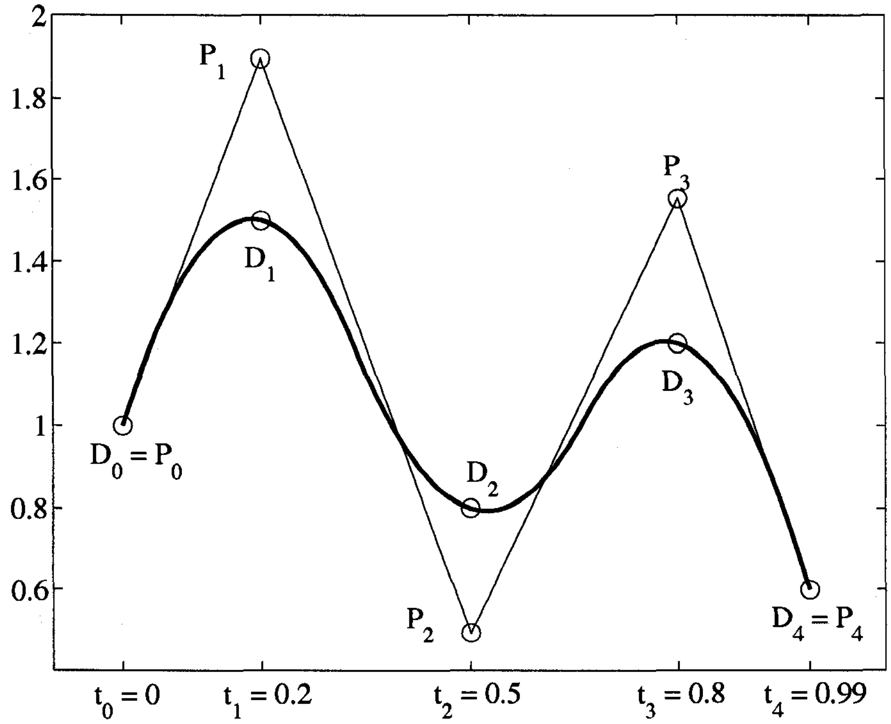


Figure 2.6: A quadratic B-spline curve

2.4.3 B-spline curve approximation

The mathematical approaches of curve fitting include interpolation and approximation. In interpolation, the curve passes through every data point precisely (Figure 2.8(a)). In approximation, the constructed curve passes near the original points smoothly (Figure 2.8(b)), rather than precisely satisfy the given data. The digital image data obtained from optical equipment are inherently noisy. In this case, using an interpolation approach introduces an unnecessary over-fitting. In the present work, an approximation approach was chosen because it can capture the general trend in the data and eliminate the undesired effects of noise.

In approximation, the fitting error need to be checked to assure it is within a tolerance (Piegl and Tiller, 1997). Usually, it is difficult to know in advance how many control points are required in order to obtain the desired accuracy. The iterative numerical optimization procedure is computationally expensive and time-consuming.

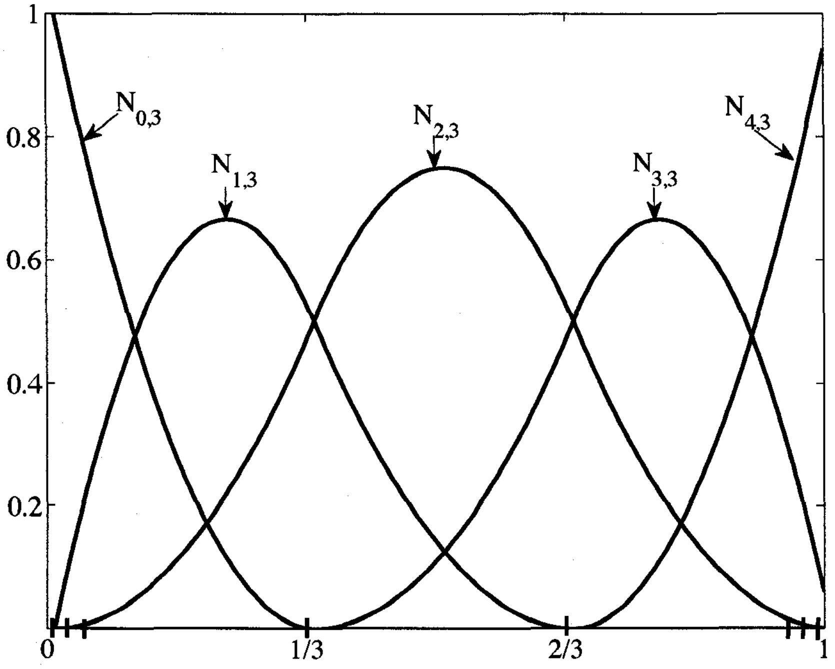


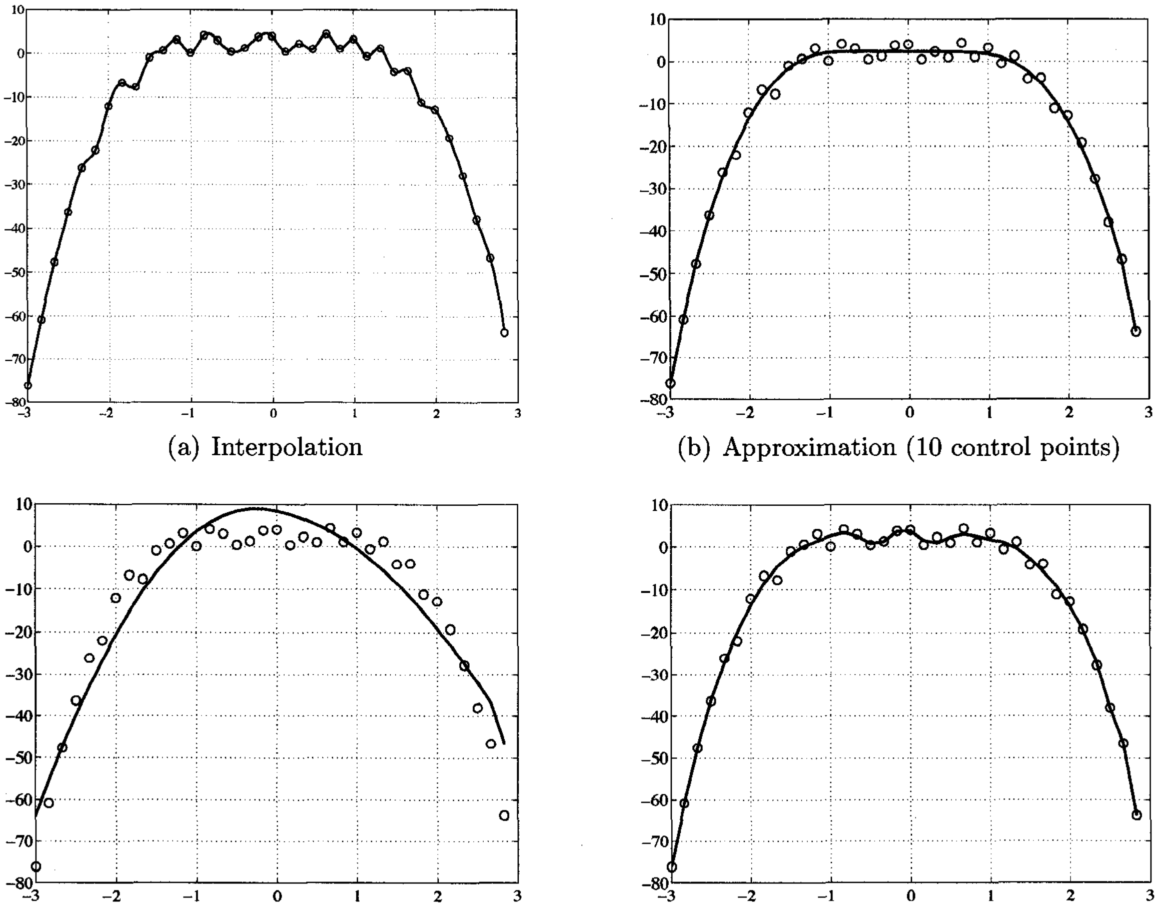
Figure 2.7: Plot of B-spline basis functions for a non-periodic quadratic B-spline over five control points

A better approach is to reformulate the problem in order to minimize the distance between the given data points and the curve. The key element of this method is to fit an approximation curve to the given data with a fixed number of control points. This is a linear optimization problem, as the control points are the only unknowns (degree and knots have been preselected).

The given data points are $D_0, D_1, \cdots, D_h$. Each data point D_j is corresponded to a parameter t_j. The corresponding point of t_j on the B-spline curve is $C(t_j)$. The fitting error is the distance between D_j and $C(t_j)$. The sum of all squared error distance is:

$$f(P_0, \cdots, P_n) = \sum_{i=0}^{h} |D_j - C(t_j)|^2 \tag{2.6}$$

The aim of the curve approximation is to find the control points that minimize the function $f(P_0, \cdots, P_n)$. This optimization problem can be solved using linear least square regression.



(a) Interpolation

(b) Approximation (10 control points)

(c) Approximation (3 control points)

(d) Approximation (18 control points)

Figure 2.8: B-spline interpolation and approximation

The accuracy of the B-spline approximation depends on the number of control points. In the approximation, the number of control points should be greater than the order of the spline (Piegl and Tiller, 1997). In general, the fit of the B-spline improves as the number of control points increase. However, as the number of control points approached the number of data points, undesirable shapes can occur (Piegl and Tiller, 1997). There exists a trade-off between approximation error and the number of control points. Figures 2.8(b) to 2.8(d) show examples of B-spline curve approximation with different number of control points. In Figure 2.8(b) a fit with ten control points gives an accurate approximation. Figure 2.8(c) shows a fit with three control points (inaccuracy), whereas Figure 2.8(d) depicts a fit with 18 control points (over-fitting).

References

Alsamhan, A., Hartely P and I. Pillinger (2003). The computer simulation of cold-roll-forming using FE methods and applied real time re-meshing techniques. *Journal of Materials Processing Technology* **142**(1), 102–111.

Anand, Vera B. (1993). *Computer Graphics and Geometric Modeling for Engineers.* John Wiley & Sons, Inc.

Cheng, Peng, Michael A. Sutton, Hubert W. Schreier and Stephen R. McNeill (2002). Full-field speckle pattern image correlation with B-spline deformation function. *Experimental Mechanics* **42**(3), 344–352.

Das, Sreekanta (2003). Fracture of Winkled Energy Pipelines. PhD thesis. University of Alberta. Edmonton, Alberta.

De Boor, C. (1978). *A Practical Guide to Splines.* Springer.

Ding, Qiulin and B. J. Davies (1987). *Surface engineering geometry for computer-aided design and manufacture.* Ellis Horwood Series in Mechanical Engineering. Ellis Horwood Limited.

Donovan, T. P. (2002). Object recognition in a digital image. In: *Visual Information Processing XI* (Zia ur Rahman, Robert A. Schowengerdt and Stephen E. Reichenbach, Eds.). Vol. 4736 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference.* pp. 1–8.

Duggal, N., M. A. Ahmetoglu, G. L. Kinzel and T. Altan (1996). Computer aided simulation of cold forming - a computer program for simple section profiles. *Journal of Materials Processing Technology* **59**(1-2), 41–48.

Green, William B. (1988). *Digital Image Processing.* 2nd ed.. Van Nostrand Reinhold.

Halmos, George T., Ed.) (2006). *Roll Forming Handbook.* Taylor & Francis Group, LLC.

Hosford, William F. and Robert M. Caddell (2007). *Metal Forming.* 3rd ed.. Cambridge University Press.

Hsu, Q. C. (2003). Comparison of different analysis models to measure plastic strains on sheet metal forming parts by digital image processing. *International Journal of Machine Tools and Manufacture* **43**(5), 515–521.

McClure, C. K. and H. Li (1995). Roll forming simulation using finite element analysis. *Manufacturing Review* **8**(2), 114–119.

Piegl, Les and Wayne Tiller (1997). *The NURBS Book.* 2nd ed.. Springer.

Rieder, Manuel Dominique (2003). Mathematical modeling of the Bauschinger effect in ERW pipeforming. Master's thesis. University of Alberta.

Sanchez-Reyes, J (1992). Single-valued Spline curves in polar coordinates. *Computer-aided Design* **24**(6), 307–315.

Sedlmaier, A. (2005). Predicting the properties of welded roll formed tubes for subsquent processes using the finite element method. In: *New Technologies for Tube & Pipe Production.* International Tube Association. pp. 27–41.

Seki, Kazunori, Shinji Ida, Shin-ya Hayashi, Kenji Yamada, Shuichi Hamauzu and Matsuo Ataka (1995). Analysis of long steel product rolling by rigid-plastic finite element method. Technical Report 67. Nippon Steel Corporation.

Senanayake, R. S., I. M. Cole and S. Thiruvarudchelvan (1994). The application of computational and experimental techniques to metal deformation in cold roll forming. *Journal of Materials Processing Technology* **45**(1-4), 155–160.

Tong, W. (1997). Detection of plastic deformation patterns in a binary aluminum alloy. *Experimental Mechanics* **37**(4), 452–459.

Vogel, J.H. and D. Lee (1990). An analytical method for deep drawing process design. *JOM* **42**(2), 8–13.

Walker, Terry R. and Roy J. Pick (1990). Approximation of the axial strains developed during the roll forming of ERW pipe. *Journal of Materials Processing Technology* **22**(1), 29–44.

Wiskel, J.B., H. Deng, M. Rieder, A. Ben-Zvi and H. Henein (2008). Digital imaging of an ERW pipe forming process. In: *Proceedings of IPC2008.* 7th International Pipeline Conference. ASME.

Wiskel, J.B., M. Rieder and H. Henein (2004). Kinematic behavior of micro alloyed steels under complex forming conditions. *Canadian Metallurgical Quarterly* **43**(1), 125–136.

Yoshida, Tohru, Matsuo Usuda and Koji Hashimoto (1995). 3-D FEM analysis of sheet metal deep drawability and stretchability. Technical Report 67. Nippon Steel Corporation.

3

Procedure for Strain Measurement by Digital Imaging [1]

3.1 Introduction

Digital imaging is a technique that acquires and subsequently processes a visual image of a physical object into a digital format. The digital image can be readily processed to provide quantitative information about the object being studied. Use of digital imaging ranges from large scale deformation analysis (Yoneyama *et al.*, 2007) to localized elastic stress behavior (Chu *et al.*, 1985). The advantages of digital imaging are that it is non-contacting, is technologically accessible (i.e. equipment necessary to obtain and manipulate digital images is readily available) and can be applied either in close proximity or remotely to the object being considered. These attributes of digital imaging make it a suitable technology for evaluation and analysis of material processing techniques such as pipe forming.

In the present work, a digital imaging procedure has been developed to calculate

[1] A version of this chapter has been accepted for publication by *Materials Science and Technology* on 8 February 2008

local strain in a forming process. An electric resistance welded (ERW) pipe forming operation was used to illustrate the proposed procedure. The procedure involves obtaining a series of digital images along the length of the pipe forming process. These images were processed to obtain discrete spatial data, i.e. x-y positioning, of the pipe shape as it is formed. A B-spline curve was fit to the discrete spatial data and used to compute the curvature of the profile. The curvature was then used to calculate the deformation strains around the circumference of the pipe. The deformation strains calculated from the digital imaging technique were compared to strain gauge measurements to validate the proposed technique.

3.2 Background

Digital imaging has been successfully used to obtain deformation information from large scale systems, i.e. deformation of a bridge (Yoneyama *et al.*, 2007), to micro scale behaviour such as elastic stress analysis (Chu *et al.*, 1985) or sheet metal forming (Lee and Hsu, 1994). For the large scale bridge analysis (Yoneyama *et al.*, 2007), displacement values were obtained by direct comparison of digital images of the bridge before and after loading. Conversely, in sheet metal forming (Lee and Hsu, 1994), measured strains are determined from imaging of a circle grid array applied to the sheet metal.

3.2.1 Deformation measurements using digital imaging

To quantify the effect of loading on bridge displacement Yoneyama *et al.* (2007), acquired digital images of a main bridge girder surface before and after loading. Using a digital image correlation technique, light intensity patterns of the bridge were obtained from both the unloaded bridge and loaded bridge. A subset from the undeformed image was chosen, and its location in the deformed image is then found. The displacement of this subset can be determined by the correlation calculation.

Chu *et al.* (1985) developed a digital imaging technique for use in experimental stress analysis in which a digital camera was used to record the intensity patterns of deformed and undeformed objects. From each image, a set of discrete intensity values (grey-level) were obtained with intensity value corresponding to a geometric coordinate on the object surface. Bilinear and polynomial interpolation was used to fit surfaces to the discrete data. Finite-strain-tensor equations were applied to the continuous intensity pattern to obtain the stress profile.

Circle grid array (CGA) is a well established method of strain analysis in sheet-forming processes (Hsu, 2003). In this procedure a circle grid (of known dimensions) is physically etched on to the surface of the undeformed sheet metal. During the forming process, the grid will deform with the sheet metal. By comparing the deformed grid with the original grid, the major and minor forming strains can be obtained. To efficiently measure grid deformation, an automated digital-image-processing system has been developed by Lee and Hsu (1994). In this system, by applying pattern recognition algorithms, images of the deformed grids are digitized. The mathematical relationship between the deformed grid and the x-y coordinates of the sampling points is constructed by curve optimization (regression analysis). The localized major and minor strain can then calculated by using the length of major and minor axis of the deformed grid and the original circle radius.

Although the measurement of strain using digital imaging in conjunction with CGAs is efficient and accurate, the method has some disadvantages. Firstly, a grid must be applied to the sheet metal (non-contacting advantage of direct digital imaging is nullified). Secondly, to obtain dynamic forming information, the grids must be continually in the line of sight of the digital camera. For a sheet metal bulge test this is not difficult. However for other forming process, such as ERW pipe forming, the equipment may surround the pipe and limit the line-of-sight of the camera. Also, damage to the grid (by the forming equipment) can occur if the grid is applied to the outer surface of the pipe. If the grid is applied to the interior of the pipe (away from the damaging effects of the forming equipment) then it is difficult to keep the grid in the line-of-sight of the digital camera.

3.2.2 Electric resistance welded (ERW) pipe forming

Electric resistance welded (ERW) pipe forming is a technology used to produce steel pipe (< 600 mm) from skelp. Figure 3.1 is a schematic of the ERW forming process and shows the different forming stages in this process including: perform rolling, a series of adjustable cage forming rolls and finishing rolls. Each stage in the forming process combines together to convert the initial flat steel skelp into a near circular pipe in preparation for the actual ERW welding process. The complexity of forming process can result in different strain histories at different positions around the periphery of the pipe. These complex strain histories may affect the final mechanical properties of the pipe via the Bauschinger effect (Wiskel *et al.*, 2004). An effective and non-invasive strain measurement is required to understand these strain histories particularly as the forming setup is frequently changed to accommodate different pipe size.

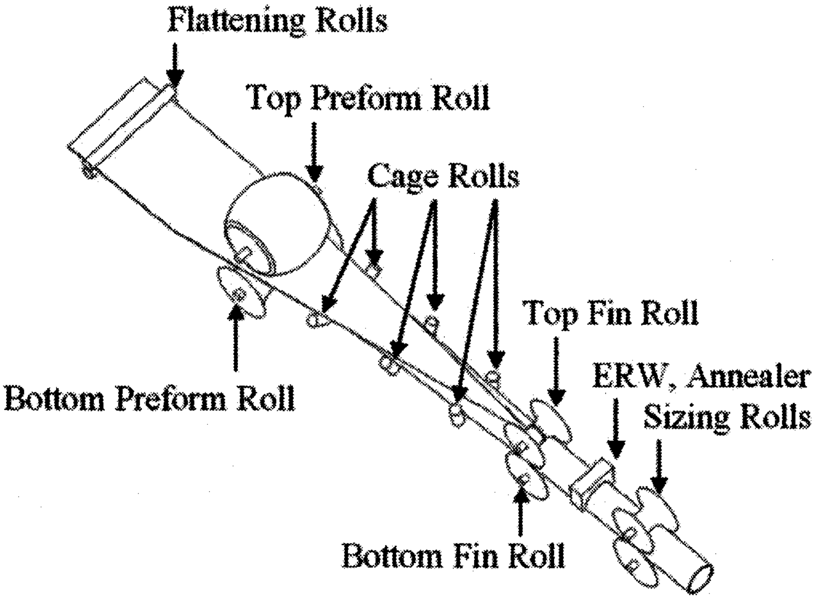


Figure 3.1: Schematic of ERW pipe forming process (Rieder, 2003)

Strain measurements during ERW pipe forming can be made using grid measurements (subject to the limitations discussed in the section on 'deformation measurements using digital imaging'), or strain gauges. Application of the latter is relatively time consuming, provides only discrete forming information (i.e. only where strain gauges are placed) and, as with a CGA, is invasive. A digital image based approach to strain measurement can overcome some of these issues.

In this work, a procedure to obtain local strain measurements in an ERW pipe forming process using digital imaging (without the application of grids) is proposed in which the forming strains are obtained directly from the localized curvature of the pipe profile. The procedure developed to achieve this requires firstly, acquisition of spatial data from digital images, secondly, the application of B-spline curve fitting to obtain a continuous pipe profile and finally, calculation of the local curvature (and hence strain) from the continuous B-spline pipe profile. These procedural steps are detailed in the following sections.

3.3 Acquiring spatial data by digital imaging

The acquisition of spatial data from a digital image was done in two steps. First, digital images of the object were captured by an image acquisition device. Second, image processing procedure was used to extract the spatial information of the object from the digital images.

3.3.1 Image acquisition

The most commonly used image acquisition device is Charge Coupled Device camera (CCD), which is a very mature technology and has high sensitivity at low illumination levels. CCD camera converts the captured visual image into a continuous electrical signal. These electrical signals are then converted into a digital image, i.e. a set of discrete picture elements or pixels. Each pixel contains a tonal value, e.g. grey scale value, which is stored in the form of binary code for future image processing. Figure 3.2 shows the image acquisition system inside the ERW forming process used to obtain the digital image at the cage 1 location for a 114.3 mm outside diameter (NPS 4) pipe (Rieder, 2003). The circular target at the centre of the image is a calibration target. The calibration target plane was orthogonal to the cameras line of sight. The magnetic tape was in the same plane as the calibration target (Rieder, 2003).

3.3.2 Image processing

Image processing involves extracting the pertinent features from the digital images, e.g. Figure 3.2. In the present work, image processing was carried out in two steps: image formation processing and feature extraction. Image formation processing includes geometric transformation, e.g. rotation or scaling, and colour corrections, e.g. brightness and contrast adjustments (Rieder, 2003). Feature extraction involves

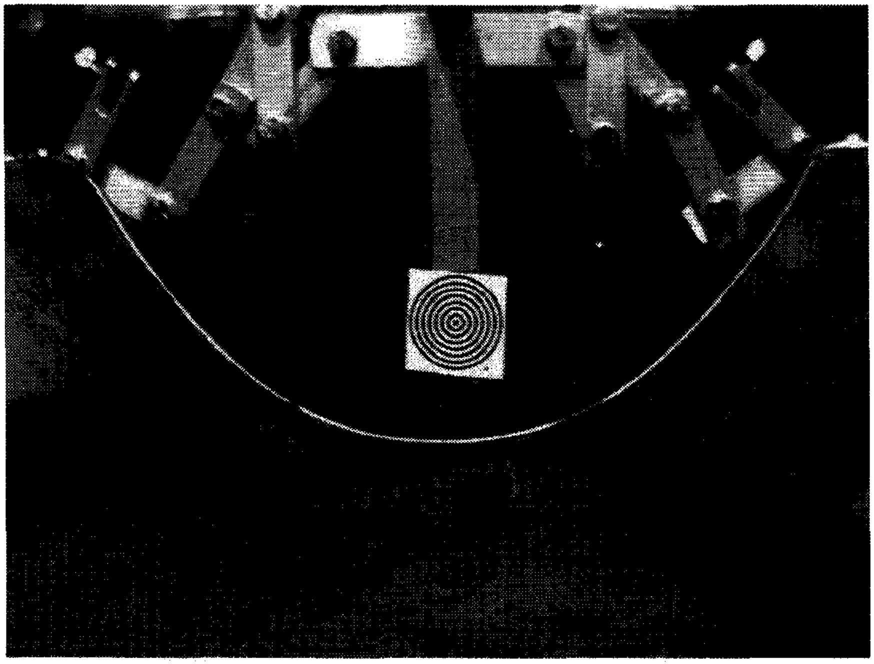


Figure 3.2: Digital imaging acquisition at cage 1

filtering out the less relevant information and preserving the important features of the image (Rieder, 2003). In the present work feature extraction amounts to obtaining x-y coordinates that correspond to the pipe profile in the raw image.

Feature extraction of the pipe profile was obtained in several steps. Firstly, image processing software was used to obtain the pipe image by evaluating the grey scale of each individual pixel of the image relative to a critical value. To enhance the ability to distinguish the pipe profile from extraneous information, a magnetic strip of low grey scale value was placed on the pipe prior to digital image acquisition. From a known calibration target, the local image coordinates (of each pixel exceeding the critical value) were converted to Cartesian coordinates (Rieder, 2003). Figure 3.3 shows the inside profile after processing, which is extracted from Figure 3.2.

3.4 Profile fitting calculation

The digital image processing discussed in Section 3.3 results in a spatial plot of the inside profile of the pipe, e.g. Figure 3.2. To obtain strain information it is necessary

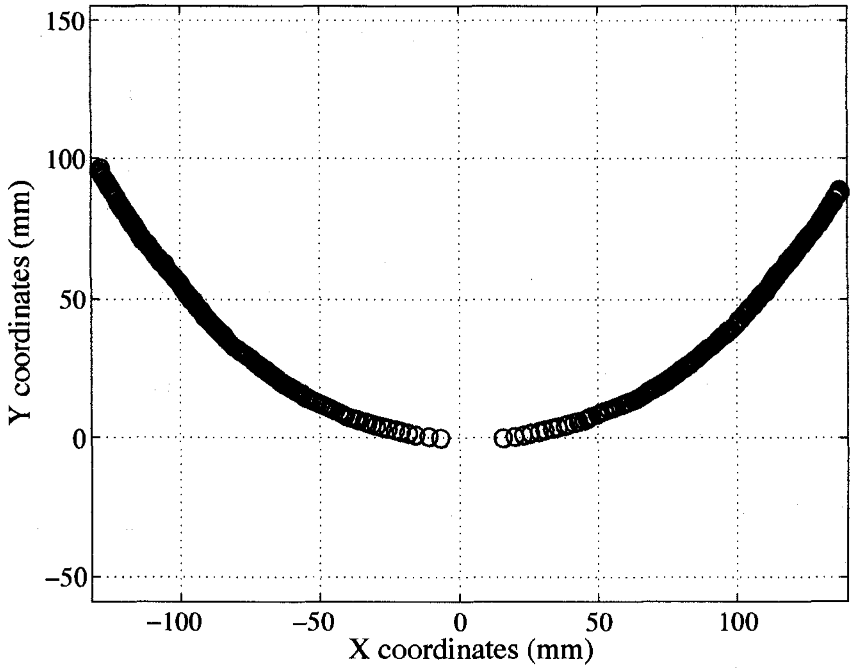


Figure 3.3: Cartesian plot of cage 1 (extrated digital image from Figure 3.2)

to fit a continuous curve to the discrete data. Mathematical approaches to curve fitting can be based on either interpolation, which constructs a curve, passes through every point, or approximation, which represents a curve that passes near the original points smoothly (Piegl and Tiller, 1997). In most digital imaging, data obtained from optical equipment will be inherently noisy. As a result an approximation approach is used because it can capture the general trend in the data.

Polynomial and spline functions are commonly used to approximate an arbitrary curve (Dewey, 1988; Anand, 1993). In this work, a B-spline curve approximation was chosen. The advantages of B-spline include the following. First, a very accurate approximation can be obtained using low-degree B-spline curves (Anand, 1993). Second, any degree of differentiability of the approximation can be guaranteed by choosing a sufficiently high-degree B-spline curve (Anand, 1993).

3.4.1 B-spline curve fitting

The discrete data points are (x_i, y_i), which can be represented by the following equation:

$$y_i = f(x_i) + \epsilon_i \qquad (i = 1, \cdots, m) \tag{3.1}$$

where $\{x_i\}$ is a nondecreasing sequence (i.e. $x_i \leq x_{i+1}$), $f(x_i)$ is an B-spline function fitting the curve, ϵ_i is the error of the i^{th} point.

The general form of B-spline function is given by (Anand, 1993):

$$f(x_i) = \sum_{j=0}^{n} p_j N_{j,k}(x_i) \tag{3.2}$$

where $\{p_j\}$ is the set of control points which can be considered as the coefficients of the B-spline function, and $N_{j,k}(x_i)$ is the j^{th} B-spline basis function of order k which can be calculated from the De Boor (1978) recursion relation:

$$\begin{aligned} N_{j,1}(x_i) &= \begin{cases} 1 & \text{if } t_j \leq t \leq t_{j+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{j,k}(x_i) &= \frac{x_i - t_j}{t_{j+k-1} - t_j} N_{j,k-1}(x_i) + \frac{t_{j+k} - x_i}{t_{j+k} - t_{j+1}} N_{j+1,k-1}(x_i) \end{aligned} \tag{3.3}$$

Let $\mathbf{T} = \{t_0, \cdots, t_j, \cdots, t_{n+k}\}$ be a nondecreasing sequence of real number, i.e. $t_j \leq t_{j+1}, (j = 0, \cdots, n+k)$. The $\{t_j\}$ are called knots, and $\mathbf{T}$ is the knot vector which is chosen with the following rule:

$$t_j = \begin{cases} a & \text{if } j \leq k \\ a + (j-k)\dfrac{b-a}{n-k+2} & \text{if } k < j \leq n+1 \\ b & \text{if } j > n+1 \end{cases} \tag{3.4}$$

where a is the minimum value of $\{x_i\}$; b is the maximum value of $\{x_i\}$.

The coefficients of the B-spline function $\{p_j\}$ can be estimated by minimize the objective function Equation (3.5):

$$\min_{p_j} \sum_{i=0}^{m} (y_i - f(x_i))^2 \tag{3.5}$$

Consider Equation (3.1) as a system of linear equations:

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} N_{0,k}(x_1) & N_{1,k}(x_1) & \dots & N_{n,k}(x_1) \\ N_{0,k}(x_2) & N_{1,k}(x_2) & \dots & N_{n,k}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,k}(x_m) & N_{1,k}(x_m) & \dots & N_{n,k}(x_m) \end{pmatrix}}_{\mathbf{N}} \underbrace{\begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{pmatrix}}_{\mathbf{p}} + \underbrace{\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{pmatrix}}_{\epsilon} \tag{3.6}$$

The linear least-squares estimator for $\mathbf{p}$ is:

$$\widehat{\mathbf{p}} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{y} \tag{3.7}$$

where $\widehat{\mathbf{p}} = \{\hat{p}_0, \cdots, \hat{p}_j, \cdots, \hat{p}_n\}$. This method is computationally easy to do for large data sets.

Thus the B-spline curve approximation values $f(x_i)$ are:

$$f(x_i) = \sum_{j=0}^{n} \hat{p}_j N_{j,k}(x_i) \tag{3.8}$$

A k^{th} order B-spline basis function generates a piecewise $(k-1)^{st}$ degree polynomial that is C^{k-2} continuous, which means the position and $[\ 1 \ \ \text{to} \ \ (k-2)\ ]$ derivatives are continuous. Considering the continuity of the curvature, i.e. the second derivatives are continuous, the third degree continuity of the curve must be satisfied, which means the order of the curve must be at least four. However, an order higher than fifth could produce undesirable shape perturbations (Piegl and Tiller, 1997). As a result, the ideal order for a B-spline for estimation of curvature is either four or five.

The accuracy of the B-spline fit also depends on the number of control points chosen. The number of control points must be greater than the order of the spline. In general the fit of the B-spline improves as the number of control points increases (Piegl and Tiller, 1997). However, as the number of control points approaches the number of data points, undesirable shapes can occur if the data exhibits noise or unwanted artifacts (Piegl and Tiller, 1997). The trade-off between approximation error and the number of control points can be calculate by the following equation.

$$\min_n S(n) = \frac{\text{SSE}(n)}{\text{SSE}(n-1)} + n \times \text{weight} \qquad (n > k) \tag{3.9}$$

where n is the number of control points, k is the order of B-spline curve, SSE(n) is the Sum of Squared Error:

$$\mathrm{SSE}(n) = \sum_{i=1}^{m}[y_i - \sum_{j=0}^{n}\widehat{p}_j N_{j,k}(x_i)]^2 \tag{3.10}$$

and w is a weighting factor chosen to ensure that the two terms on the right-hand side of Equation (3.9) will be of the same order of magnitude. In this work a weighting value of 0.05 was used. The value of n which can give the minimum value of $S(n)$ in Equation (3.9) is the optimal number of control points.

3.4.2 Application of B-spline curve fitting to cage 1

The proceeding mathematical concepts are applied to the Cartesian data for cage 1, Figure 3.2. The optimal number of control points for cage 1 was determined to be 11 and was obtained by minimizing the value of $S(n)$ in Equation (3.9), minimization is shown graphically in Figure 3.4.

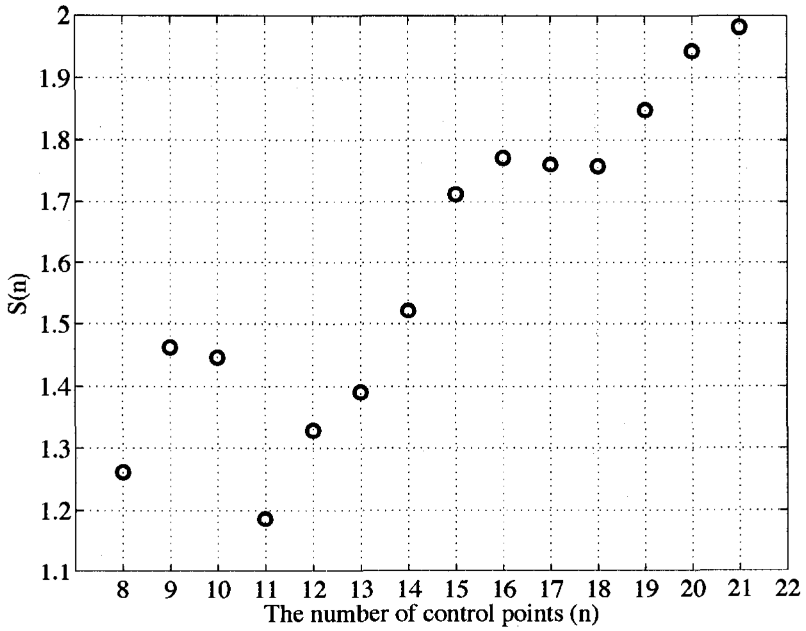


Figure 3.4: Optimal number of control points in cage 1 is 11

As detailed in Section 3.4.1, a fifth-order B-spline curve, 11 control points, was then used to approximate the image data for cage 1. A comparison between the fitted B-spline curve and the image data is shown in Figure 3.5.

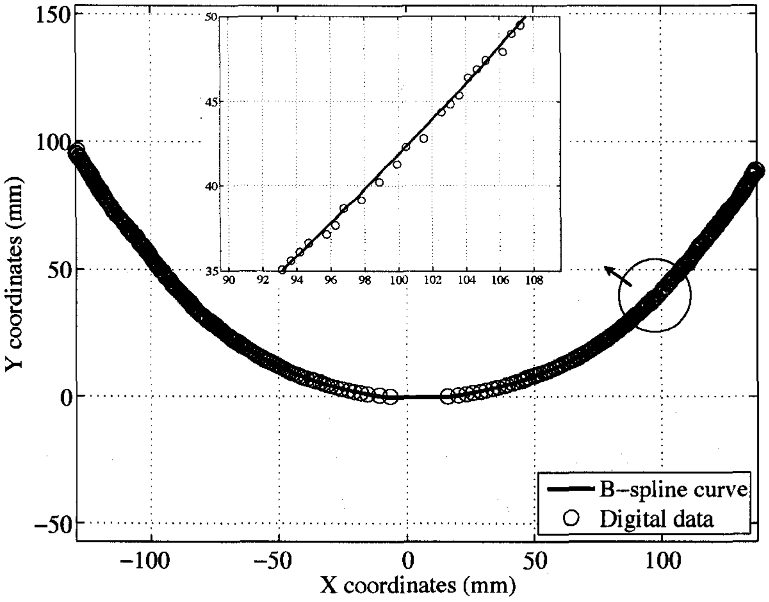


Figure 3.5: Measured curve for cage 1 v.s. calculated fifth order B-spline curve

3.4.3 Curvature of image

The bending strain along the pipe can be computed using the estimated pipe curvature. An equation for the calculation of the bending strains in the circumferential direction is presented by Noronha *et al.* (2005):

$$\varepsilon = \frac{t}{2}\left(\frac{1}{R_0} - \frac{1}{R_1}\right) \tag{3.11}$$

where R_0 is the radius of curvature of the undeformed pipe surface which is close to infinite, t is the wall thickness in the longitudinal direction, and R_1 is the external surface radii of curvature which is the inverse of curvature. Thus to obtain strain values from the digital image it is necessary to determine the local curvature of the pipe from the B-splines.

3.4.4 Local curvature calculations

The B-spline curve function $f(x_i)$, calculated in Equation (3.8), can be used to compute the curvature of the B-spline curve at any point, x_i, using the following equation (Stewart, 1994):

$$\kappa(x_i) = \frac{|f''(x_i)|}{\left[1 + (f'(x_i))^2\right]^{3/2}} \tag{3.12}$$

Since the basis function $N_{j,k}(x_i)$ is a function of x_i, and satisfies the second-degree differentiability, the first and second derivatives of the B-spline curve are (Barone, 2001):

$$\begin{cases} f'(x_i) = \sum\limits_{j=0}^{n} \hat{p}_j N'_{j,k}(x_i) \\ f''(x_i) = \sum\limits_{j=0}^{n} \hat{p}_j N''_{j,k}(x_i) \end{cases} \tag{3.13}$$

where $\{p_j\}$ is the set of control points, which are calculated by Equation (3.7), and $N'_{j,k}(x_i)$ and $N''_{j,k}(x_i)$ are the first and second derivatives of the B-spline basis function (Noronha *et al.*, 2005):

$$\begin{aligned} N'_{j,k}(x_i) &= \frac{N_{j,k-1}(x_i) + (x_i - t_j)N'_{j,k-1}(x_i)}{t_{j+k-1} - t_j} + \frac{(t_{j+k} - x_i)N'_{j+1,k-1}(x_i) - N_{j+1,k-1}(x_i)}{t_{j+k} - t_{j+1}} \\ N''_{j,k}(x_i) &= \frac{N_{j,k-1}(x_i) + (x_i - t_j)N''_{j,k-1}(x_i)}{t_{j+k-1} - t_j} + \frac{(t_{j+k} - x_i)N''_{j+1,k-1}(x_i) - N_{j+1,k-1}(x_i)}{t_{j+k} - t_{j+1}} \end{aligned} \tag{3.14}$$

Calculated predicted curvature values, using Equations (3.11–3.14), can exhibit fluctuations caused by numerical differentiation. In order to eliminate these processing artefacts, a local averaging approach was used. The averaged curvature value K is related to the raw curvature value κ by the following equation:

$$K(x) = \frac{1}{2S} \int_{x_0+S}^{x_0-S} \kappa(x)dx \tag{3.15}$$

where the span S can be used to specify the radius of smoothing, the larger the span the smoother the result. An example of the effect on local averaging on the calculated curvature is shown in Figure 3.6, for cage 1, which compares the calculated curvature and the smoothed curvature.

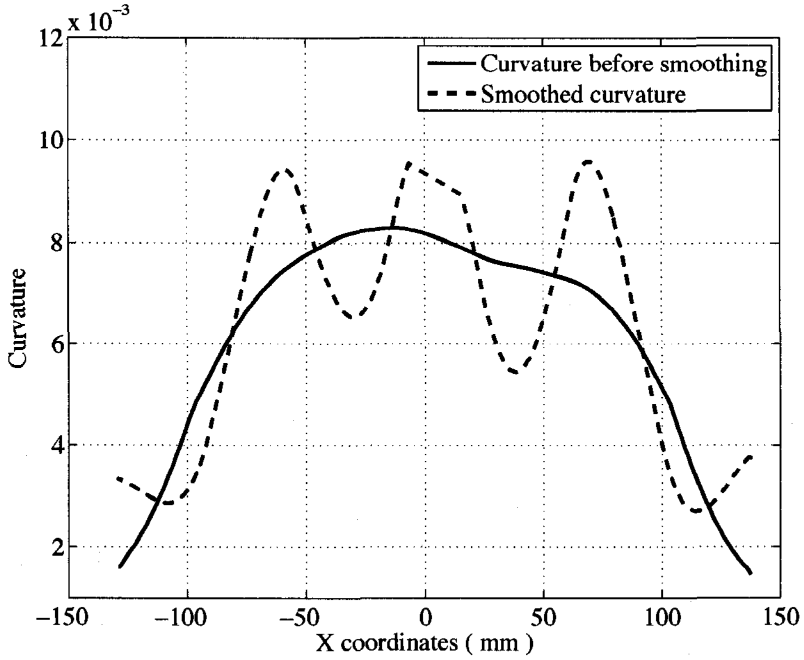


Figure 3.6: Curvature for cage 1 before and after smoothing

3.5 Summary of proposed procedure

Step 1: Obtain Cartesian coordinates of 2D pipe images at each forming stage using digital imaging processing described in Section 3.3

Step 2: Use fifth-order B-spline curve to fit the data points. The number of control points can be chosen to minimize the value of Equation (3.5). The weight $\{p_j\}$ of each spline curve in Equation (3.2) can be chosen by using linear least square parameter estimation as shown in Equation (3.7).

Step 3: Calculate the first and second derivative of B-spline curve function by using Equation (3.13).

Step 4: Calculate the local curvature of the B-spline curve by using Equation (3.12).

Step 5: Smooth the curvature of B-spline curve by using the Local Averaging approach using Equation (3.14).

Step 6: Calculate the bending strain in the circumferential direction by using Equation (3.11).

3.6 Comparison of digital strain measurements with strain gauges

The veracity of the proceeding procedure can be confirmed by comparing the calculated strain obtained via digital imaging with the measured strain obtained using strain gauges applied to the inside of 114.3 mm outside diameter (NPS 4) pipe.

As the strain measurement sensors are located along the pipe profile, the measured strain values are known as a function of arc length of the pipe profile. Therefore, the predicted strain values also need to be represented as a function of arc length. The first derivatives of the B-spline curve $f'(x)$ is continuous on $[c, d]$, then the length of the curve $y = f(x)$, $(c \leq x \leq d)$ is (Stewart, 1994):

$$L = \int_{c}^{d} \sqrt{1 + [f'(x)]^2} dx \tag{3.16}$$

Adapted Simpon's Rule (Cheney and Kincaid, 1999) was employed to calculate the integral in Equation (3.16).

The strains in the circumferential direction of the pipe were calculated by using Equation (3.11), where R_0 is infinite, and $1/R_1$ is the smoothed curvature value and t, the wall thickness in the longitudinal direction, is 4.0 mm. The predicted strains for cage 1 are shown in Figure 3.7. The procedure and comparisons were repeated for cage 2 and cage 3 (Figure 3.8 and Figure 3.9 respectively). The strains obtained from the strain gauges at the respective cages are included in Figures 3.7-3.9.

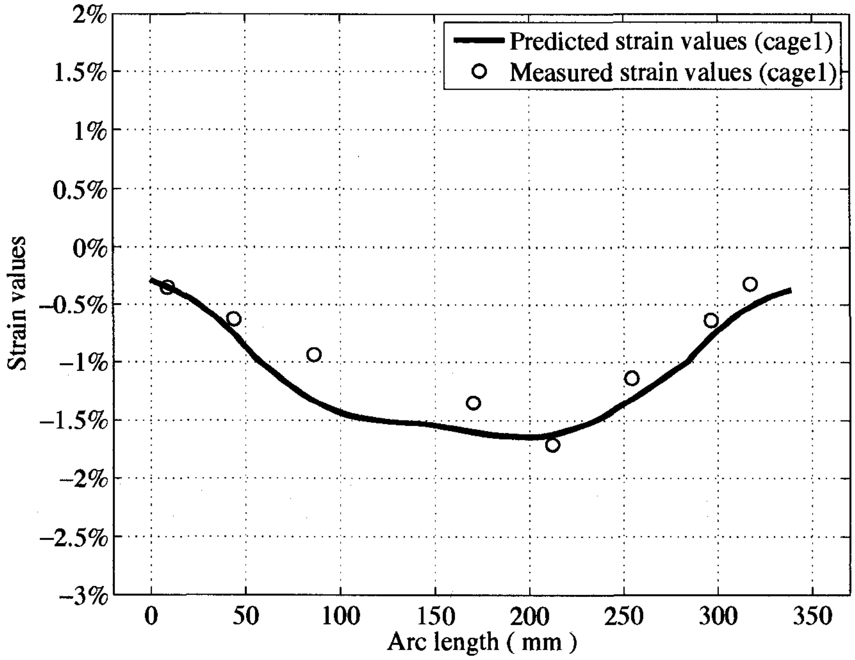


Figure 3.7: Comparison between predicted and measured strains (cage 1)

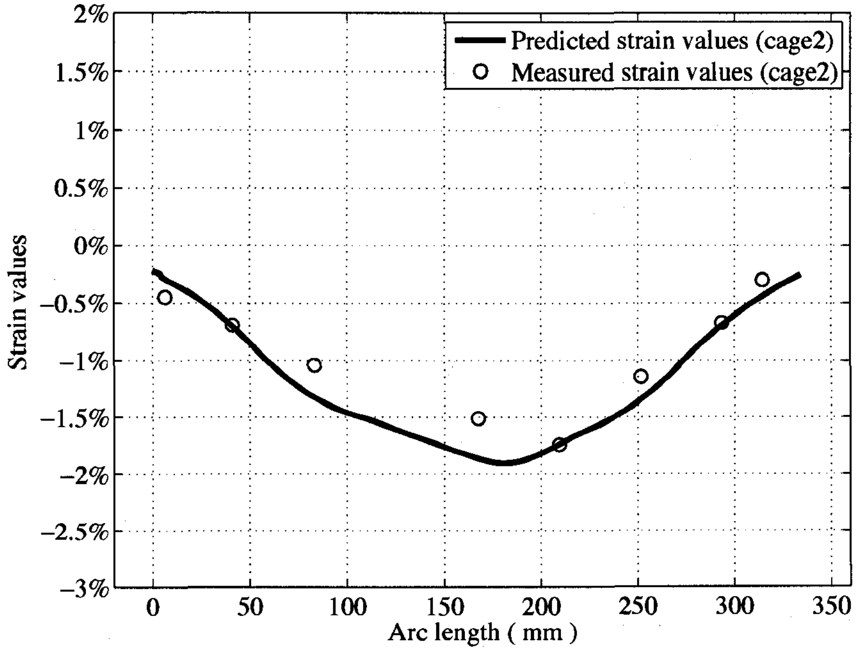


Figure 3.8: Comparison between predicted and measured strains (cage 2)

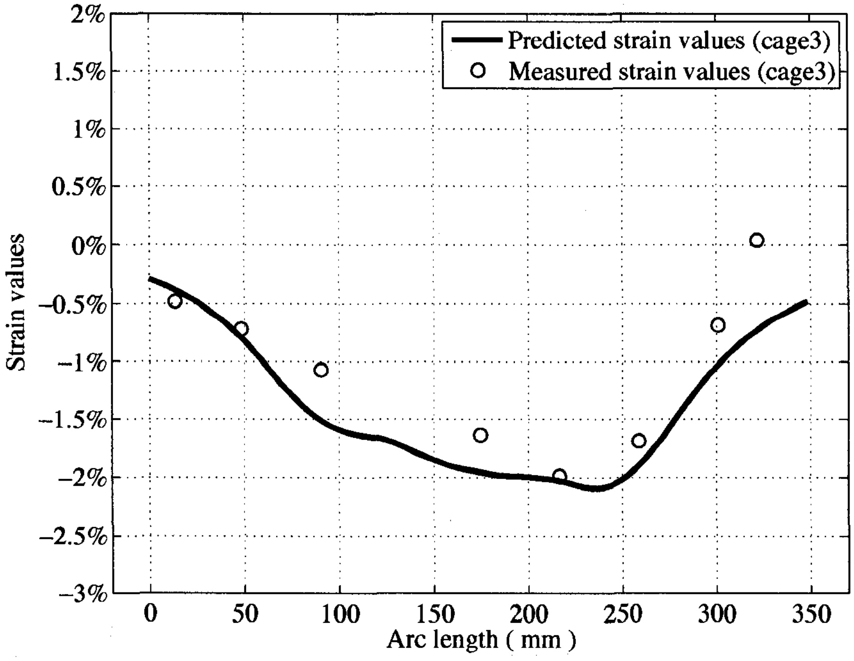


Figure 3.9: Comparison between predicted and measured strains (cage 3)

In all three figures, the trend in the predicted strain values obtained by the proposed procedure agrees reasonably well with the trend obtained (independently) by strain gauges. While specific strain gauge values may be slightly different than the values predicted by our approach, this difference maybe attributed to the physical dimensions of the strain gauges (10 mm long) and/or deviations in the exact forming setup between the digital image pipe analysis and the strain gauge measurements.

3.7 Conclusions

Digital imaging based techniques can be used to quickly and efficiently to analyze process information from a physical object. In this work, a method is proposed for measuring strain values in a pipe forming process, using a spatial pipe profile obtained from a digital image. Following spatial digitization of the image, a fifth order B-spline curve, with the optimal number of control points, was used to generate a continuous image profile. The weight of the B-spline function is calculated by using linear least square regression and a local averaging approach is applied to smooth the fluctuations of the calculated curvature of B-spline curve. The strain values are calculated by using the mathematical relation between the strain and the radius of curvature. The procedure was implemented for an electric resistance welded (ERW) pipe forming process and the technique validated against the strain gauge data.

References

Anand, Vera B. (1993). *Computer Graphics and Geometric Modeling for Engineers.* John Wiley & Sons, Inc.

Barone, S. (2001). Gear geometric design by B-spline curve fitting and sweep surface modelling. *Engineering with Computers* **17**(1), 66–74.

Cheney, W. and D. Kincaid (1999). *Numerical Mathematics and Computing.* 4th ed.. Brooks/Cole Publishing Company.

Chu, T. C., W. F. Ranson and M. A. Sutton (1985). Applications of digital image coorelation techniques to experimental mechanics. *Experimental Mechanics* **25**(23), 232–244.

De Boor, C. (1978). *A Practical Guide to Splines.* Springer.

Dewey, Bruce R. (1988). *Computer Graphics for Engineers.* Harper & Row.

Hsu, Q. C. (2003). Comparison of different analysis models to measure plastic strains on sheet metal forming parts by digital image processing. *International Journal of Machine Tools and Manufacture* **43**(5), 515–521.

Lee, R.S. and Q.C. Hsu (1994). Image-processing system for circular-grid analysis in sheet-metal forming. *Experimental Mechanics* pp. 108–115.

Noronha, Dauro B., Ricardo R. Martins, Breno P. Jacob and Eduardo Souza (2005). The use of B-spline in the assessment of strain levels associated with plain dents. Instituto Brasileiro de Petroleo e Gas - IBP. Rio de Janeiro, Brazier.

Piegl, Les and Wayne Tiller (1997). *The NURBS Book.* 2nd ed.. Springer.

Rieder, Manuel Dominique (2003). Mathematical modeling of the Bauschinger effect in ERW pipeforming. Master's thesis. University of Alberta.

Stewart, James (1994). *Calculus.* 3rd ed.. Brooks/Cole Publishing Company.

Wiskel, J.B., M. Rieder and H. Henein (2004). Kinematic behavior of micro alloyed steels under complex forming conditions. *Canadian Metallurgical Quarterly* **43**(1), 125–136.

Yoneyama, S., A. Kitagawa, S. Iwata, K. Tani, H. Kikuta, A. Kitagawa, K. Tani S. Iwata and H. Kikuta (2007). Bridge deflection measurement using digital image correlation. *Experimental Techniques* **31**(1), 34–40.

4

Numerical Approximation Methods for Geometric Modeling

4.1 Introduction

In the pipe-rolling industry, accurate roll production is one of the major concerns for producers. The complexity of the roll forming process can result in unique deformation histories within and between forming steps. An accurate understanding of these deformations is necessary to access their effects on mechanical properties of the material, and subsequently to improve the pipe quality. Several techniques have been developed to measure deformations in the roll forming process. These techniques include: strain gauges (Rieder, 2003), grid measurements (Hsu, 2003) and digital imaging (Deng *et al.*, 2008). Both strain gauges and grid measurements are invasive and relatively time consuming. Strain gauges provide only discrete forming information at the points where strain gauges are placed, while grid measurements are unavailable to provide the dynamic information.

Digital imaging is a technique that acquires a series of visual images from a physical object. Compared with the other two measurement devices, the digital imaging technique is generally insensitive to environmental factors, e.g. high temperature and electromagnetic fields. It is non-invasive to the object being studied, and is non-limited to the position and the type of forming equipment. Therefore, it can be applied either in close proximity or remote from the object being considered. The qualitative information included in digital images can be subsequently processed to provide the quantitative information as a continuous function, i.e. at every point on the studied object. These advantages of digital imaging make it a suitable technique for analyzing pipe forming processes.

Deng *et al.* (2008) developed a digital imaging based technique to quantify the local strain behavior in the metal forming process, using an electric resistance welded (ERW) pipe forming operation as a study case. The continuous curve function of the deformed skelp profile was generated from the discrete digital spatial data by one-dimensional B-spline approximation method. The local strains were calculated by evaluating the curvature of the skelp profile. The effectiveness and accuracy of this technique was confirmed by comparing the strains calculated from the digital imaging technique with strain gauge measurements.

The digital imaging based technique proposed by Deng *et al.* (2008) can be used to analyze the strain histories in a roll forming process quickly and efficiently. However, the one-dimensional approximation method implemented in this technique is only suitable for curves that project one-to-one onto the x-axis or y-axis. Because of this limitation, one-dimensional B-spline approximation method is difficult to be applied on the skelp profiles in the last few stages of the ERW pipe forming. Therefore, it is necessary to implement a suitable numerical approximation method for generating an accurate description of skelp profile and subsequently calculating the transverse plastic strains from the local curvature.

In the present work, three B-spline approximation methods (one-dimensional, two-dimensional and trigonometric B-spline) were investigated to reconstruct the continuous curve of the skelp profiles from the discrete digital data. These preceding mathematical concepts were applied to approximate the spatial data of cage 9 in an ERW forming operation, and subsequently to estimate the deformation strains using the procedure proposed by Deng *et al.* (2008). The strain values calculated respectively from the three B-spline approximation methods are compared with each other and validated against the strain gauge measurements.

4.2 Background

The digital imaging based technique has been successfully used to measure strain values in an electric resistance welded (ERW) pipe forming process (Deng *et al.*, 2008). The digital images of the pipe were captured at a series of locations along the length of the forming process. These images were processed to obtain the spatial data of pipe profiles, which was subsequently used to predict the strain values.

4.2.1 Electric resistance welded pipe forming

Electric resistance welded (ERW) pipe forming is a technique used to produce thin-walled small diameter pipe with high strength steels. A schematic of this process is shown in Figure 4.1. The forming process is typically comprised of a series of stages: flattening rolls, preform rolls, adjustable cage forming rolls and finishing rolls.

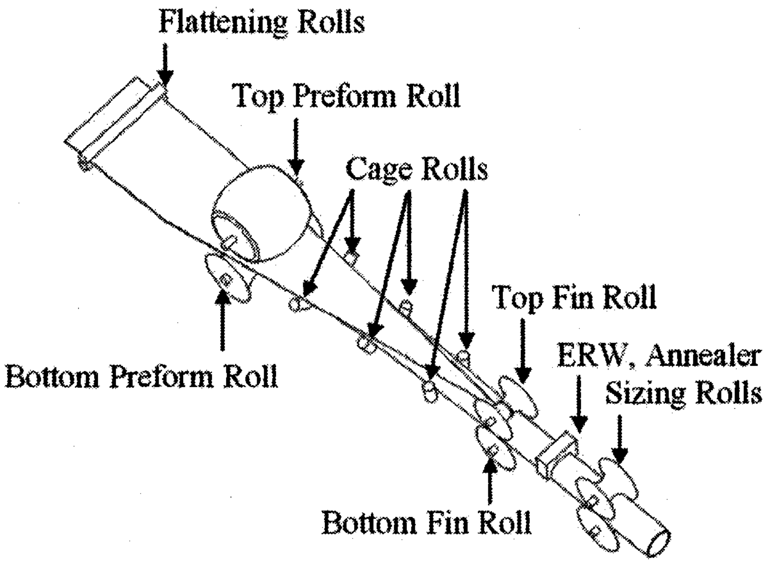


Figure 4.1: Schematic of ERW pipe forming process (Rieder, 2003)

The starting material (coiled steel skelp) is sent into the flattening rolls to create the flat steel skelp. Then the skelp is deformed into a curved cross-section by several preform rolls before entering the cage forming section. A series of adjustable cage rollers guide the edges of the skelp from the preform section to the fin section. At the end of the cage section, the cross-section shape of the skelp converts into a half ellipse. The opposing concave rollers in the fin section produce a progressively rounder shape. After the final fin roller, the two edges of the skelp are pressed together for the following electric resistance welding.

The complexity of this forming process, particularly as the forming setup is frequently changed to accommodate different pipe sizes, can result in different strain histories at different positions around the periphery of the pipe. To understand these deformations and their effect on subsequent geometry and mechanical properties of the pipe, it is necessary to develop an effective strain measurement for this process.

4.2.2 Acquiring spatial data by digital imaging

The acquisition of spatial data from a digital image was done in two steps: image acquisition and image processing. A series of digital images of the studied object were captured by an image acquisition device (CCD camera) at discrete locations (i.e. at each forming stage) along the length of the forming process. A bright identification strip was used to provide the inside profile of the pipe. The image processing software was used to enhance the contrast of the identification strip against the dark background. An imaging process software was used to isolate the identification strip from the dark background. Figure 4.2 shows the processed pipe profile image at cage 9.

Each digital image is composed of a series of pixels which contains a tonal value (e.g. gray scale value) and a position value (e.g. pixel coordinates). A MATLAB code was used to evaluate the gray scale of each individual pixel of the image relative to a critical value (the white strip provided the necessary contrast). The local image coordinates were converted to Cartesian coordinates by using the calibration target (Rieder, 2003). Figure 4.3 shows the Cartesian coordinates of the inside pipe profile obtained from the image shown in Figure 4.2.

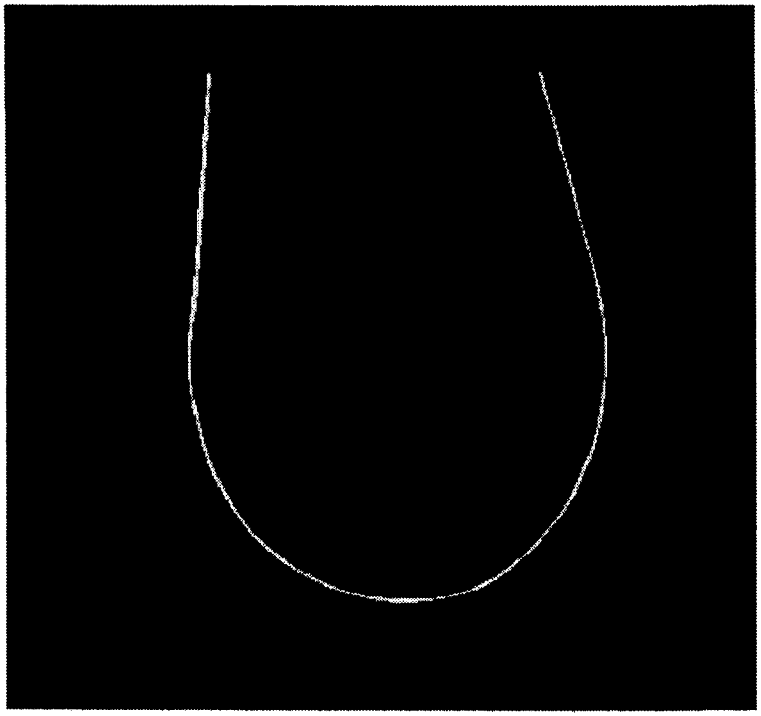


Figure 4.2: Digital imaging acquisition at cage 9

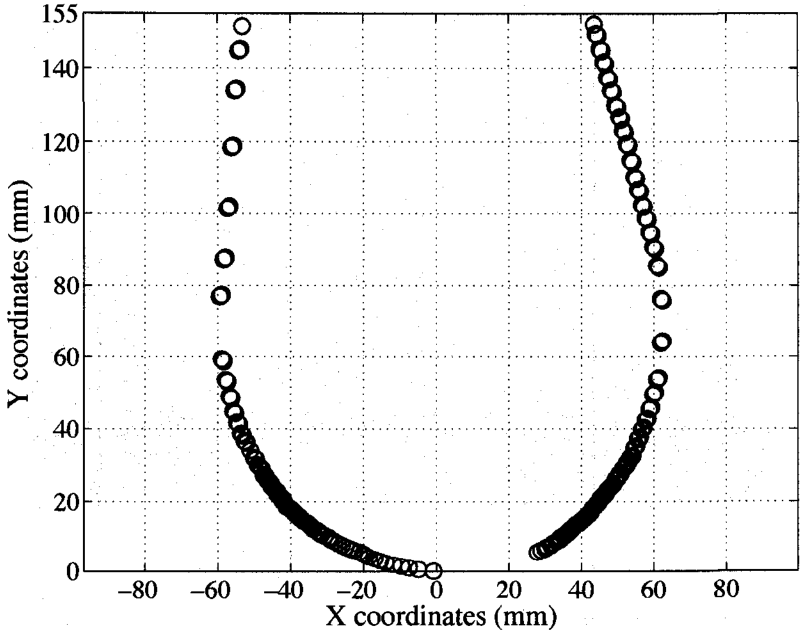


Figure 4.3: Cartesian plot of cage 9

4.3 Numerical approximation method

The digital image processing discussed in the Section 4.2.2 results in a spatial plot of the inside profile of the pipe (see Figure 4.3). To obtain strain information, it is necessary to generate a mathematical model that describes the skelp profile. A B-spline approximation method was chosen to generate the numerical model because it can capture the general trend in the data (Piegl and Tiller, 1997).

The accuracy of the B-spline fitting depends on the number of control points chosen (Piegl and Tiller, 1997). In general, the accuracy of the approximation improves as the number of control points increases. However, as the number of control points approaches the number of data points, over-fitting can occur if the data exhibits noise or unwanted artifacts. The trade-off between approximation error and the number of control points can be calculated by the Bayesian information criterion (BIC) (Sanchez *et al.*, 2007) (Equation 4.1):

$$\mathrm{BIC}(n) = n \ln(m) + m \ln\left(\frac{\mathrm{RSS(n)}}{m}\right) \tag{4.1}$$

where n is the number of the control points, m is the number of the data used for approximation, and RSS is the sum of squares error between the spatial data and the approximated curve with n control points.

The optimal number of control points can be calculated by minimizing the value of BIC from the same group of spatial data. For different spatial data sets, the optimal number of control points are different.

Before proceeding with an analysis of the strain information obtained by digital imaging, a comparison among the three B-spline approximation methods (1-D, 2-D and trigonometric) was given. These preceding mathematical concepts were applied to approximate the spatial data of cage 9, and to estimate the curvature of the approximated B-spline curve.

4.3.1 One-dimensional B-spline approximation

A one-dimensional B-spline approximation algorithm has been used to generate the geometry of the skelp in ERW pipe forming (Deng *et al.*, 2008). The constraint imposed on this algorithm is: the knot vector $T = \{t_0, \cdots, t_j, \cdots, t_{n+k}\}$ must be a non-decreasing order, i.e. $t_j \leq t_{j+1}$ (De Boor, 1978). This non-parametrical representation is the easiest one with which to work, but only curves that project one-to-one onto the x-axis can be described (Sanchez-Reyes, 1990).

For a curve which is not a single-valued function of x coordinates, the approximation needs to be applied to three or more separate segments of the curve. The profiles of cage 9, shown in Figure 4.4, is a curve sweeping more than 180°. For this curve, a given x-coordinate value x_1 has two corresponding y-coordinate values y_1 and y_2. Before proceeding with the one-dimensional B-spline approximation, the profile curve was separated to left segment, middle segment and right segment (Figure 4.4).

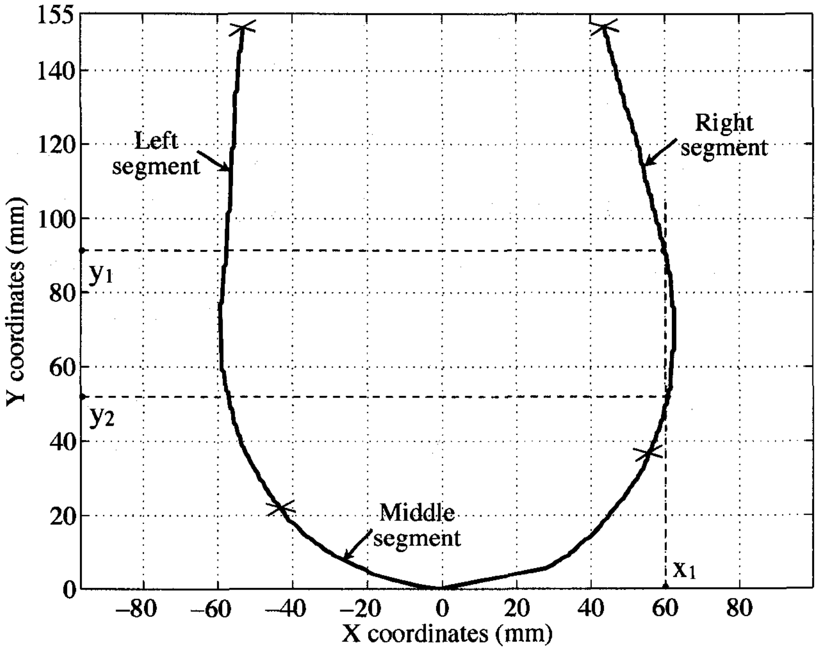


Figure 4.4: A curve sweeping more than 180°

The discrete data points at left, right and middle segments are (x_{l_i}, y_{l_i}), (x_{r_i}, y_{r_i}), and (x_{m_i}, y_{m_i}), which can be represented by the following equation:

$$\begin{cases} x_{l_i} = f_l(y_{l_i}) + \epsilon_{l_i} \\ x_{r_i} = f_r(y_{r_i}) + \epsilon_{r_i} \\ y_{m_i} = f_m(x_{m_i}) + \epsilon_{m_i} \end{cases} \tag{4.2}$$

where $\{y_{l_i}\}$, $\{y_{r_i}\}$ and $\{x_{m_i}\}$ are nondecreasing sequences, $f_l(y_{l_i})$, $f_r(y_{r_i})$ and $f_m(x_{m_i})$ are B-spline functions fitting the segments, and ϵ_{l_i}, ϵ_{r_i} and ϵ_{m_i} are the errors of the i^{th} point on each segments.

The B-spline function of each segment is given by (Anand, 1993):

$$\begin{cases} f_l(y_{l_i}) = \sum_{j=0}^{n} p_{l_j} N_{l_{j,k}}(y_{l_i}) \\ f_r(y_{r_i}) = \sum_{j=0}^{n} p_{r_j} N_{r_{j,k}}(y_{r_i}) \\ f_m(x_{m_i}) = \sum_{j=0}^{n} p_{m_j} N_{m_{j,k}}(x_{m_i}) \end{cases} \tag{4.3}$$

where $\{p_{l_j}\}$, $\{p_{r_j}\}$ and $\{p_{m_j}\}$ are the sets of control points which can be considered as the coefficients of the B-spline functions of each segment, and $N_{l_{j,k}}(y_{l_i})$, $N_{r_{j,k}}(y_{r_i})$ and $N_{m_{j,k}}(x_{m_i})$ are the j^{th} B-spline basis functions of each segments with order k, which can be calculated by Equation (4.4).

The general B-spline basis function $N_{j,k}(u_i)$ is defined by the De Boor (1978) recursion relation:

$$\begin{gathered} N_{j,1}(u_i) = \begin{cases} 1 & \text{if } t_j \leq t \leq t_{j+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{j,k}(u_i) = \frac{u_i - t_j}{t_{j+k-1} - t_j} N_{j,k-1}(u_i) + \frac{t_{j+k} - u_i}{t_{j+k} - t_{j+1}} N_{j+1,k-1}(u_i) \end{gathered} \tag{4.4}$$

where k is the order of the B-spline, $\mathbf{T} = \{\underbrace{a, \cdots, a,}_{k} t_k, \cdots, t_n, \underbrace{b, \cdots, b}_{k}\}$ is a nonperiodic knot vector (Anand, 1993). The first and last knots have multiplicity k. Here a is the minimum value of $\{u_i\}$, b is the maximum value of $\{u_i\}$, and $\{t_k \leq \cdots \leq t_n\}$.

The coefficients of the B-spline function $\{p_j\}$ can be estimated by minimizing the objective function Equation 4.5:

$$\begin{cases} \sum_{i=0}^{m}\left(x_{l_i} - \sum_{j=0}^{n} p_{l_j} N_{l_{j,k}}(y_{l_i})\right)^2 \\ \sum_{i=0}^{m}\left(x_{r_i} - \sum_{j=0}^{n} p_{r_j} N_{r_{j,k}}(y_{r_i})\right)^2 \\ \sum_{i=0}^{m}\left(y_{m_i} - \sum_{j=0}^{n} p_{m_j} N_{m_{j,k}}(x_{m_i})\right)^2 \end{cases} \tag{4.5}$$

The linear least-squares estimators for $\mathbf{p}_l$, $\mathbf{p}_r$, and $\mathbf{p}_m$ are calculated by:

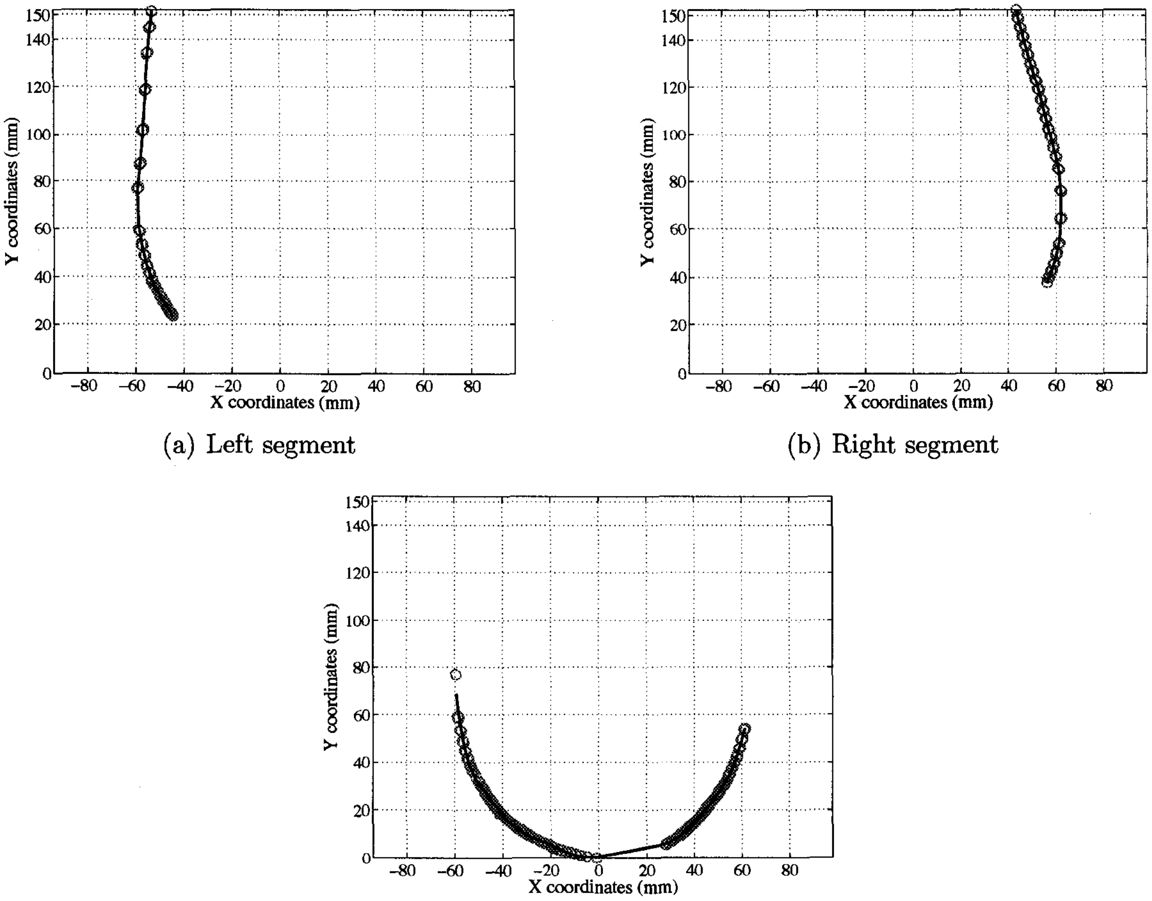
$$\begin{cases} \widehat{\mathbf{p}}_l = (\mathbf{N}_l{}^T\mathbf{N}_l)^{-1}\mathbf{N}_l{}^T\mathbf{x}_l \\ \widehat{\mathbf{p}}_r = (\mathbf{N}_r{}^T\mathbf{N}_r)^{-1}\mathbf{N}_r{}^T\mathbf{x}_r \\ \widehat{\mathbf{p}}_m = (\mathbf{N}_m{}^T\mathbf{N}_m)^{-1}\mathbf{N}_m{}^T\mathbf{y}_m \end{cases} \tag{4.6}$$

where $\widehat{\mathbf{p}}_l = \{\hat{p}_{l_0}, \cdots, \hat{p}_{l_j}, \cdots, \hat{p}_{l_n}\}$, $\widehat{\mathbf{p}}_r = \{\hat{p}_{r_0}, \cdots, \hat{p}_{r_j}, \cdots, \hat{p}_{r_n}\}$, and $\widehat{\mathbf{p}}_m = \{\hat{p}_{m_0}, \cdots, \hat{p}_{m_j}, \cdots, \hat{p}_{m_n}\}$.

Thus the B-spline curve approximation values for each segment are given by Equation (4.7).

$$\begin{cases} f_l(y_{l_i}) = \sum_{j=0}^{n} \hat{p}_{l_j} N_{l_{j,k}}(y_{l_i}) \\ f_r(y_{r_i}) = \sum_{j=0}^{n} \hat{p}_{r_j} N_{r_{j,k}}(y_{r_i}) \\ f_m(x_{m_i}) = \sum_{j=0}^{n} \hat{p}_{m_j} N_{m_{j,k}}(x_{m_i}) \end{cases} \tag{4.7}$$

A comparison between the three approximation B-spline curves and the image data is shown in Figure 4.5. For each segment, the approximation is accurate. However, the conjunctions between each segment are discontinuous (Figure 4.6).



(a) Left segment

(b) Right segment

(c) Middle segment

Figure 4.5: Approximated 1-D B-spline curve v.s. the spatial data

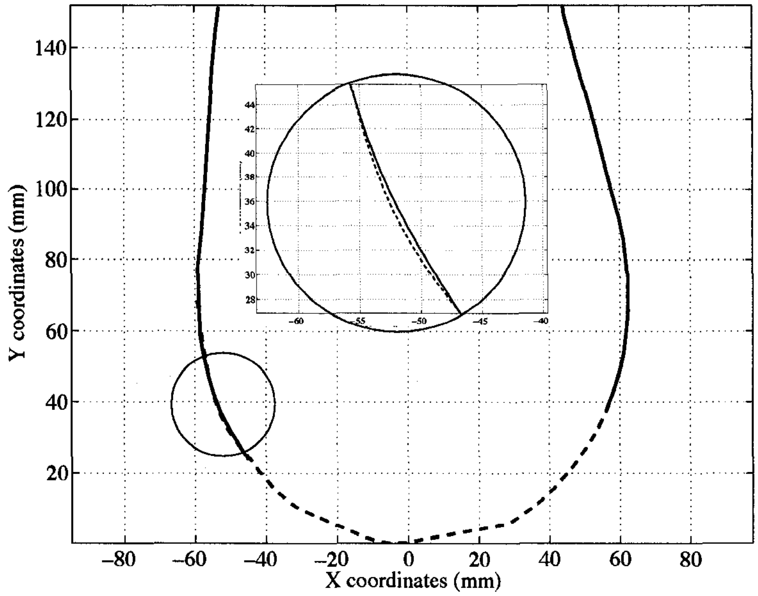


Figure 4.6: The combination of the three separated segments

After the approximation of each segment, the curvature of each segment was obtained by using the local curvature calculation described by Deng *et al.* (2008). The smoothed local curvatures of the three approximation segments are shown in Figure 4.7. These curvatures are not continuous between each segment. The discontinuous parts can be connected smoothly by using local interpolation with weight functions (Pobegailo, 1991):

$$Q(x_i) = \big(1 - w(x_i)\big)\, D_1(x_i) + w(x_i)\, D_2(x_i) \tag{4.8}$$

where $D_1(x_i)$ and $D_2(x_i)$ are the points on the two segments need to be connected, $w(x_i)$ is a specified weight function which satisfy $0 \leq w(x_i) \leq 1$. The smoothness can be achieved by changing the weights. The connected curvatures are show in Figure 4.7. The conjunction of these curves leaves sharp edges in the final strain estimation (Figure 4.12).

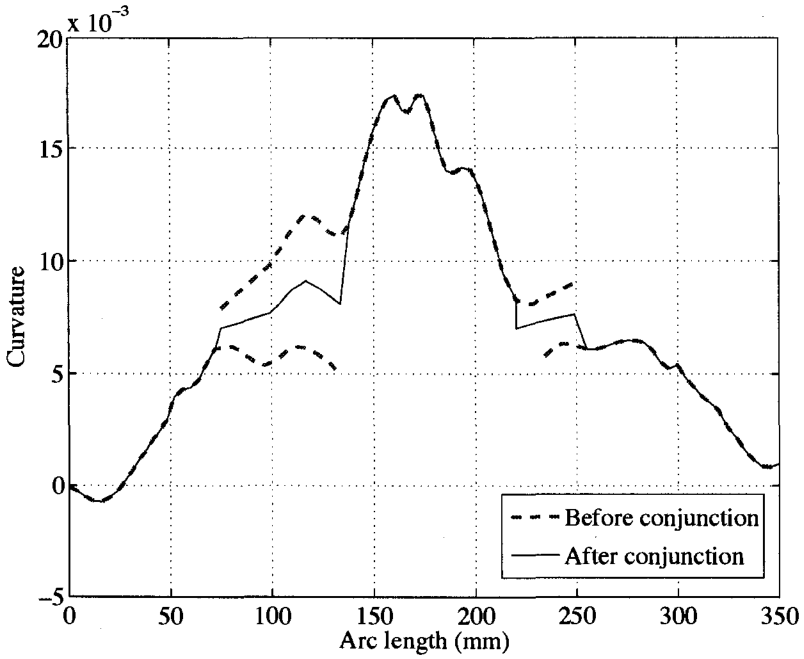


Figure 4.7: Calculated curvature of cage 9, 1-D B-spline

4.3.2 Two-dimensional B-Spline approximation

In the one-dimensional B-spline approximation method, a linear optimization problem was set up with the control points as the only unknowns. The least square regression was applied to minimize the sum of the error between the fitting points and the approximated curve. The control points lie over abscissas regularly spaced along the x-axis.

In two-dimensional B-spline approximation, the least square regression is employed to solve two linear optimization problems, one with $y = f_x(u_i)$ that minimizes the sum of the vertical distances and the other with $x = f_y(u_i)$ that minimizes the sum of the horizontal distances from the known points to the approximated curve (Tsai and Chen, 1994). The discrete data points are (x_i, y_i), $(i = 1, \cdots, m)$, which can be represented by the following equation:

$$\begin{cases} x_i = f_x(u_i) + \epsilon_{x_i} \\ y_i = f_y(u_i) + \epsilon_{y_i} \end{cases} \tag{4.9}$$

where $\{u_i\}$ is a nondecreasing sequence (i.e. $u_i \leq u_{i+1}$), $f_x(u_i)$ and $f_y(u_i)$ are the x

and y components of B-spline function fitting the curve, ϵ_{x_i} and ϵ_{y_i} are the x and y components of error at the i^{th} point.

The form of two-dimensional B-spline function is given by:

$$\begin{cases} f_x(u_i) = \sum_{j=0}^{n} p_{x_j} N_{j,k}(u_i) \\ \\ f_y(u_i) = \sum_{j=0}^{n} p_{y_j} N_{j,k}(u_i) \end{cases} \tag{4.10}$$

where $\{p_{x_j}\}$ and $\{p_{y_j}\}$ are sets of control points which can be considered to be the coefficients of the B-spline function, and $N_{j,k}(u_i)$ is the j^{th} B-spline basis function of order k which can be calculated from the De Boor (1978) recursion relation (Equation 4.4).

The coefficients or control points of the B-spline function $\{p_{x_j}\}$ and $\{p_{y_j}\}$ can be estimated by minimizing the value of Equation (4.11):

$$\begin{cases} \min\limits_{p_{x_j}} \sum_{i=0}^{m} (x_i - f_x(u_i))^2 \\ \\ \min\limits_{p_{y_j}} \sum_{i=0}^{m} (y_i - f_y(u_i))^2 \end{cases} \tag{4.11}$$

The linear least squares estimators for $\mathbf{p}_x$ and $\mathbf{p}_y$ are:

$$\begin{cases} \widehat{\mathbf{p}}_x = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{x} \\ \\ \widehat{\mathbf{p}}_y = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{y} \end{cases} \tag{4.12}$$

where $\widehat{\mathbf{p}}_x = \{\hat{p}_{x,0}, \cdots, \hat{p}_{x,j}, \cdots, \hat{p}_{x,n}\}$, and $\widehat{\mathbf{p}}_y = \{\hat{p}_{y,0}, \cdots, \hat{p}_{y,j}, \cdots, \hat{p}_{y,n}\}$.

Thus the two-dimensional B-spline curve approximation values $\big(f_x(u_i), f_y(u_i)\big)$ can be calculated by Equation (4.13).

$$\begin{cases} f_x(u_i) = \sum_{j=0}^{n} \hat{p}_{x,j} N_{j,k}(u_i) \\ \\ f_y(u_i) = \sum_{j=0}^{n} \hat{p}_{y,j} N_{j,k}(u_i) \end{cases} \tag{4.13}$$

The curvature of the two-dimensional B-spline curve can be calculated by the following equation:

$$\kappa(u_i) = \frac{|f_x' f_y'' - f_y' f_x''|}{(f_x'^2 + f_y'^2)^{3/2}} \tag{4.14}$$

Figure 4.8 represents the approximation curve with 30 control points (the optimal number, calculated using BIC). This fitting curve contains undesired fluctuation shape at the middle part of the profile. The calculated curvature (before and after smoothing) of cage 9 using 2-D B-spline approximation is shown in Figure 4.9. It was found that the 2-D B-spline approximation is not accurate and is not suitable for the subsequent curvature calculation.

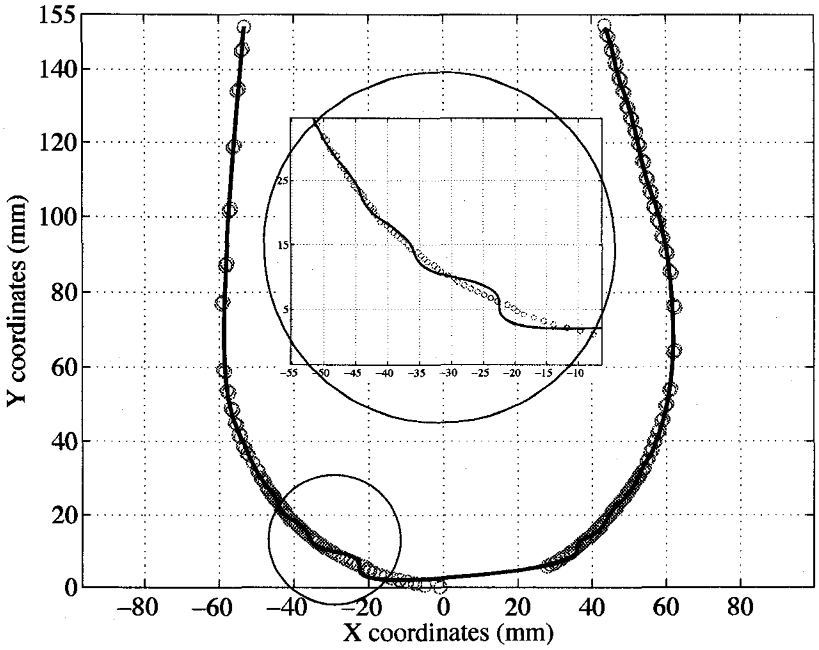


Figure 4.8: Approximated 2-D B-spline curve v.s. the spatial data

4.3.3 Trigonometric B-spline approximation

In the B-spline approximation methods described in Section 4.3.1 and 4.3.2, planar curves were described as the graph of a single-valued function $y = f(x)$ in Cartesian

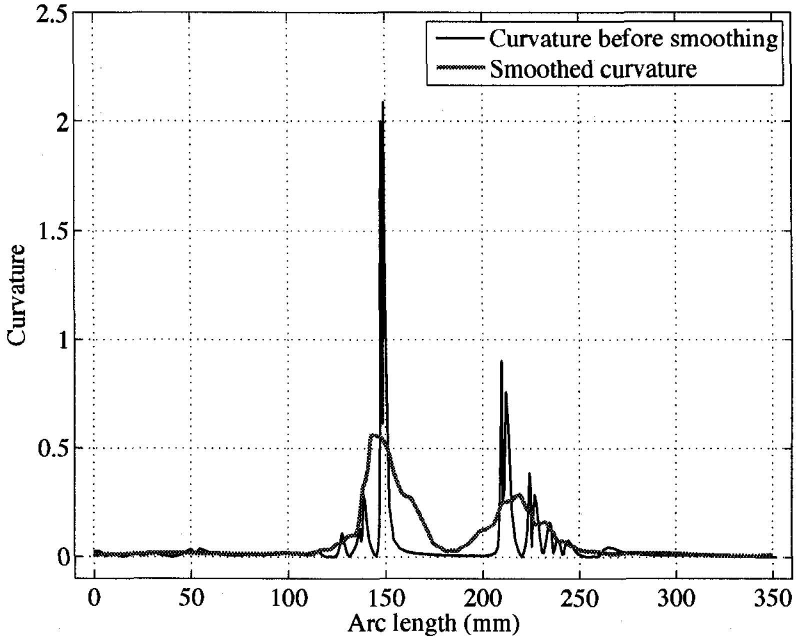


Figure 4.9: Calculated curvature of cage 9, 2-D B-spline

coordinates (x, y). Similarly, single-valued curves $r = r(\theta)$ can be defined in a system of polar coordinates (r, θ). In some applications, like the the near circular pipe profile in cage 9, it is very convenient to define the profile in terms of $r = r(\theta)$, to analyze the geometry of the skelp of the pipe.

The data points in Cartesian coordinates (x_i, y_i) can be converted to polar coordinates (r_i, θ_i) by the following equations:

$$\begin{cases} r_i = \sqrt{{x_i}^2 + {y_i}^2} \\ \theta_i = \tan^{-1}\left(\dfrac{y_i}{x_i}\right) \end{cases} \tag{4.15}$$

Then these discrete data can be represented as:

$$r_i = \delta(\theta_i) + \epsilon_i \qquad (i = 1, \cdots, m) \tag{4.16}$$

where $\{\theta_i\}$ is a nondecreasing sequence (i.e. $\theta_i \leq \theta_{i+1}$), $\delta(\theta_i)$ is a trigonometric B-spline function fitting the curve, and ϵ_i is the error of the i^{th} point.

A trigonometric B-spline function is defined as (Sanchez-Reyes, 1992):

$$\delta(\theta_i) = \sum_{j=0}^{n} \delta_j M_{j,k}(\theta_i) \tag{4.17}$$

where δ_j is the set of control points which can be considered to be the coefficients of the trigonometric B-spline function, and $M_{j,k}(\theta_i)$ is the normalized trigonometric B-spline satisfying the following recursion relation (Sanchez-Reyes, 1992):

$$\begin{gathered} M_{j,1}(\theta_i) = \begin{cases} 1 & \text{if} \quad t_j \leq \theta_i \leq t_{j+1} \\ 0 & \text{otherwise} \end{cases} \\ M_{j,k}(\theta_i) = \frac{\sin(\theta_i - t_j)}{\sin(t_{j+k-1} - t_j)} M_{j,k-1}(\theta_i) + \frac{\sin(t_{j+k} - \theta_i)}{\sin(t_{j+k} - t_{j+1})} M_{j+1,k-1}(\theta_i) \end{gathered} \tag{4.18}$$

where $\{t_j\}$ are called knots. Let $\mathbf{T}$ be the knot sequence (Koch *et al.*, 1995):

$$\mathbf{T} = \{t_0 \leq t_1 \leq \cdots \leq t_{n+k}\}$$

where

$$\theta_a = t_0 = \cdots = t_{k-1}, \qquad t_{n+1} = \cdots = t_{n+k} = \theta_b,$$

and where

$$\{t_k \leq \cdots \leq t_n\}$$

where θ_a and θ_b are the minimum and maximum value of $\{\theta_i\}$, and $(\theta_b - \theta_a) \leq 2\pi$ (Casciola and Morigi, 1996).

The coefficients of the B-spline function $\{\delta_j\}$ can be estimated by minimizing the value of Equation (4.19):

$$\sum_{i=0}^{n} (r_i - \delta(\theta_i))^2 = \sum_{i=0}^{n} \left(r_i - \sum_{j=0}^{n} \delta_j M_{j,k}(\theta_i) \right)^2 \tag{4.19}$$

The linear least-squares estimator for δ is:

$$\hat{\delta} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{r} \tag{4.20}$$

where $\hat{\delta} = \{\hat{\delta}_0, \cdots, \hat{\delta}_j, \cdots, \hat{\delta}_n\}$

Thus the approximated trigonometric B-spline curve $\delta(\theta_i)$ is:

$$\delta(\theta_i) = \sum_{j=0}^{n} \hat{\delta}_j M_{j,k}(\theta_i) \tag{4.21}$$

A 5^{th} order trigonometric B-spline curve with 12 control points (the optimal number, calculated using BIC) was used to approximate the image data for cage 9. A comparison between the fitted B-spline curve and the image data is shown in Figure (4.10).

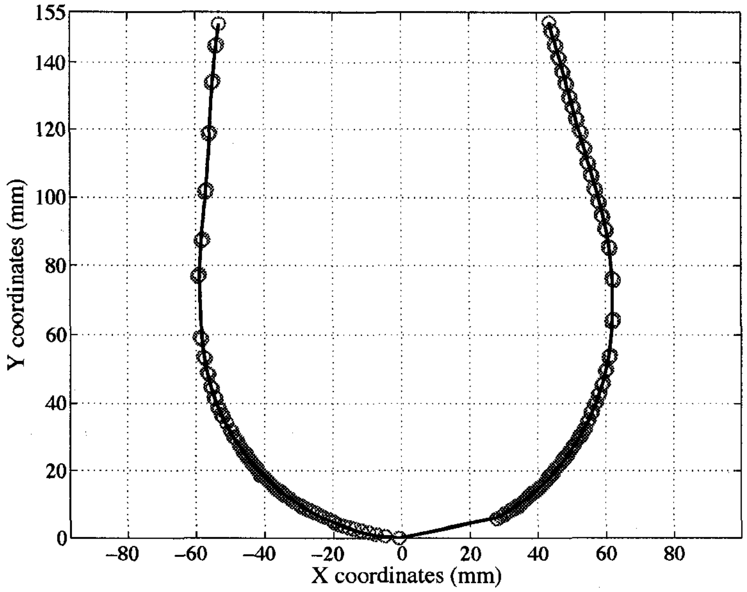


Figure 4.10: Approximated trigonometric B-spline curve v.s. the spatial data

If a curve is defined in polar coordinates as $\delta(\theta)$, then its curvature is:

$$\kappa(\theta) = \frac{\delta^2 + 2\delta'^2 - \delta\delta''}{\left(\delta^2 + \delta'^2\right)^{3/2}} \tag{4.22}$$

where $\kappa(\theta)$ is the curvature, δ' and δ'' are the first and second derivatives of $\delta(\theta)$ which are given by Equation 4.21:

$$\begin{aligned} \delta'(\theta_i) &= \sum_{j=0}^{n} \hat{\delta}_j M'_{j,k}(\theta_i) \\ \delta''(\theta_i) &= \sum_{j=0}^{n} \hat{\delta}_j M''_{j,k}(\theta_i) \end{aligned} \tag{4.23}$$

where $\{\hat{\delta}_j\}$ is a set of control points which are calculated by the Equation 4.20, and $M'_{j,k}(\theta_i)$ and $M''_{j,k}(\theta_i)$ are the first and second derivative of trigonometric B-spline basis function:

$$\begin{aligned}
M'_{j,k}(\theta_i) &= \frac{\cos(\theta_i - t_j)}{\sin(t_{j+k-1} - t_j)} M_{j,k-1}(\theta_i) + \frac{\sin(\theta_i - t_j)}{\sin(t_{j+k-1} - t_j)} M'_{j,k-1}(\theta_i) \\
&\quad - \frac{\cos(t_{j+k} - \theta_i)}{\sin(t_{j+k} - t_{j+1})} M_{j+1,k-1}(\theta_i) + \frac{\sin(t_{j+k} - \theta_i)}{\sin(t_{j+k} - t_{j+1})} M'_{j+1,k-1}(\theta_i) \\
M''_{j,k}(\theta_i) &= -\frac{\sin(\theta_i - t_j)}{\sin(t_{j+k-1} - t_j}) M_{j,k-1}(\theta_i) + \frac{2\cos(\theta_i - t_j)}{\sin(t_{j+k-1} - t_j)} M'_{j,k-1}(\theta_i) \qquad (4.24) \\
&\quad + \frac{\sin(\theta_i - t_j)}{\sin(t_{j+k-1} - t_j}) M''_{j,k-1}(\theta_i) - \frac{\sin(t_{j+k} - \theta_i)}{\sin(t_{j+k} - t_{j+1})} M_{j+1,k-1}(\theta_i) \\
&\quad - \frac{2\cos(t_{j+k} - \theta_i)}{\sin(t_{j+k} - t_{j+1})} M'_{j+1,k-1}(\theta_i) + \frac{\sin(t_{j+k} - \theta_i)}{\sin(t_{j+k} - t_{j+1})} M''_{j+1,k-1}(\theta_i)
\end{aligned}$$

Calculated predicted curvature values (using Equation 4.22-4.24) can exhibit fluctuations caused by numerical differentiation. In order to eliminate these processing artifacts, a local averaging approach proposed by Deng *et al.* (2008) was used. As the size of the strain gauge is about 10mm, the chosen span value used to specify the radius of smoothing was $\pi/20$. Figure 4.11 shows the estimated curvature value for cage 9 before and after smoothing.

4.4 Strain calculation

The surface transverse strain along the pipe was computed using the estimated pipe curvature. An equation for the calculation of the bending strains in the circumferential direction is presented by Noronha *et al.* (2005):

$$\varepsilon = \frac{t}{2}\left(\frac{1}{R_0} - \frac{1}{R_1}\right) \tag{4.25}$$

where R_0 is the radius of curvature of the undeformed pipe surface which is close to infinite, t is the wall thickness in the longitudinal direction, and R_1 is the external surface radii of curvature which is the inverse of curvature.

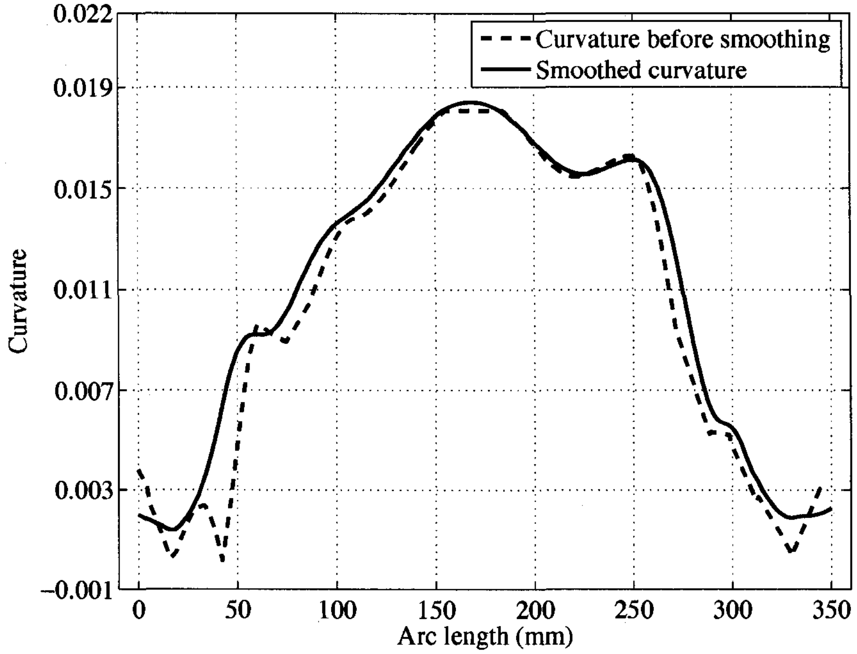


Figure 4.11: Calculated curvature of cage 9, trigonometric B-spline

The predicted strains for cage 9 (using 1-D, 2-D and trigonometric B-spline approximation methods respectively) are shown in Figure 4.12-4.14. The strains obtained from the strain gauges at cage 9 are included in Figures 4.12-4.14. It was found that the trigonometric B-spline provided the most accurate strain prediction among these three methods.

The procedure of strain prediction using trigonometric B-spline approximation method were repeated for cage 8 and cage 2. The comparisons between the predicted strain values and measured strain gauge data are shown in Figure 4.15 and Figure 4.16 respectively.

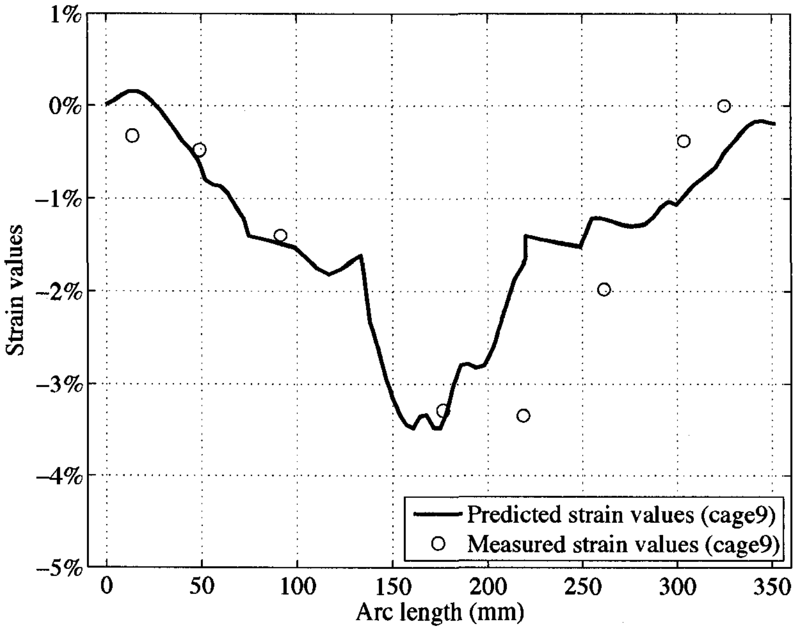


Figure 4.12: Comparison between predicted and measured strain (1-D)

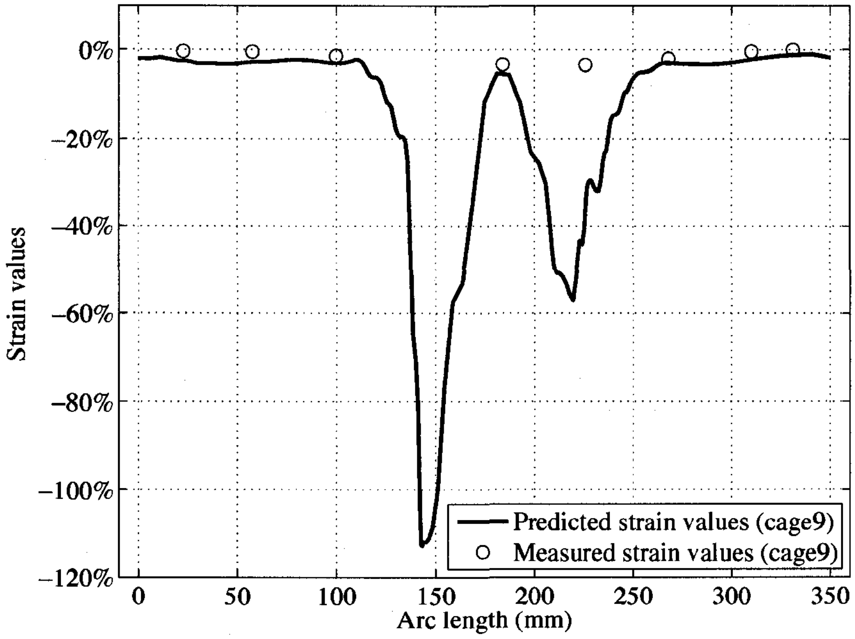


Figure 4.13: Comparison between predicted and measured strain (2-D)

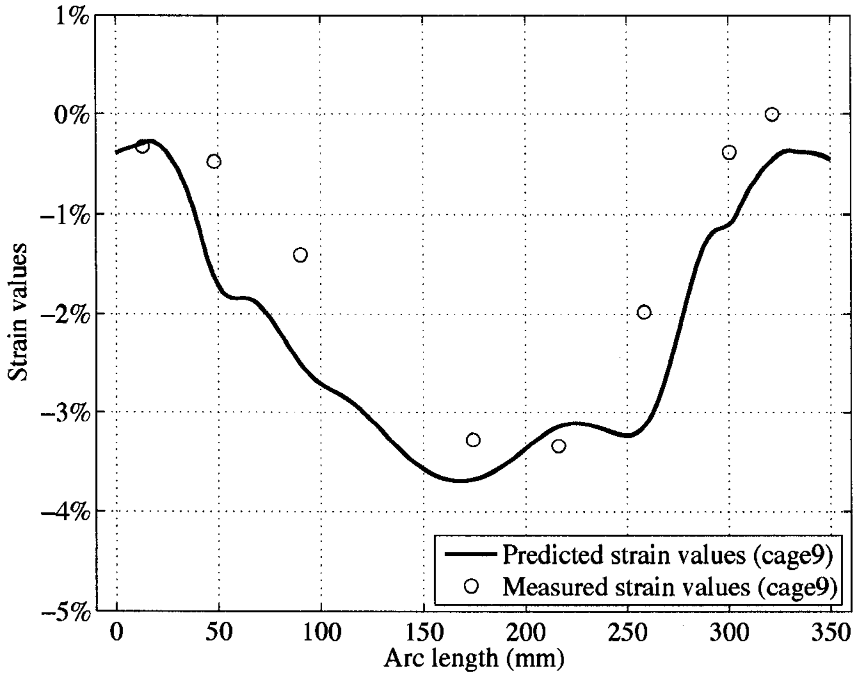


Figure 4.14: Comparison between predicted and measured strain (trigonometric)

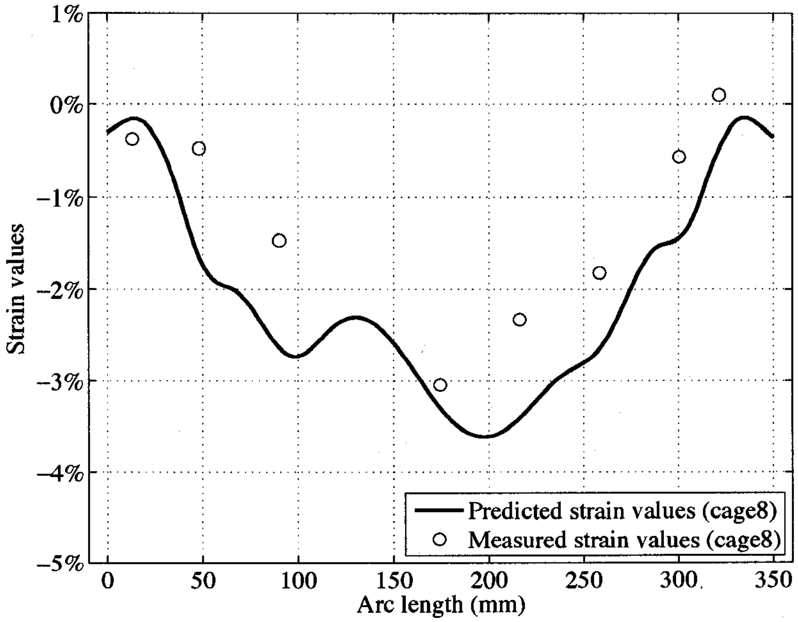


Figure 4.15: Comparison between predicted and measured strain (trigonometric)

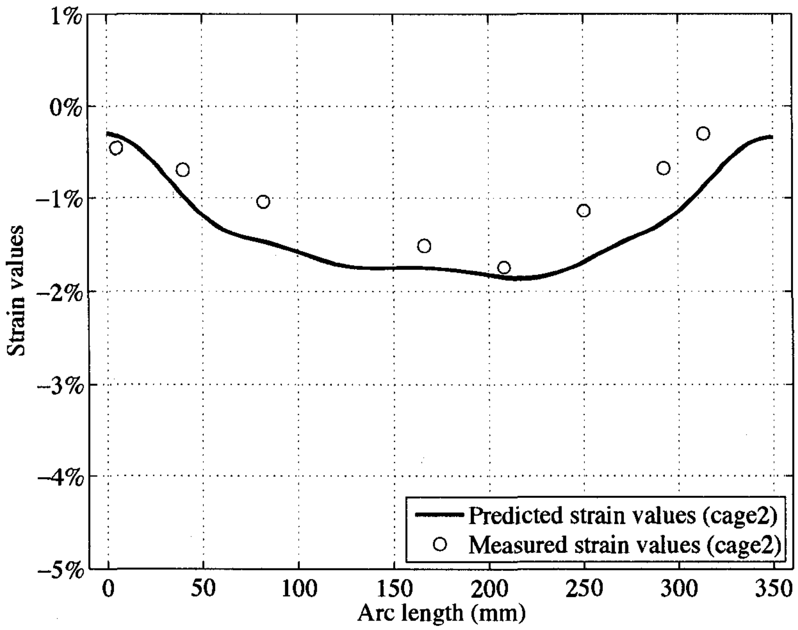


Figure 4.16: Comparison between predicted and measured strain (trigonometric)

4.5 Conclusions

Digital imaging based techniques have been developed to analyze process information in roll forming quickly and efficiently. In the present work, an investigation among three B-spline curve approximation methods (1-D, 2-D and trigonometric) was conducted to assess their suitability for generating an accurate description of skelp profiles from digital spatial data. These methods were applied to an electric resistance welded (ERW) pipe forming process to obtain the mathematical model of the profile spatial data. Bayesian information criterion (BIC) was used to optimize the value of the number of control points in each model.

One-dimensional B-spline is the most commonly used approximation method among those evaluated, but only curves that project one-to-one onto the x-axis can be described. For curves that do not project one-to-one onto the x-axis, the one-dimensional B-spline needs to be applied to three or more separate segments of the curve, each one less than 180°. Two-dimensional and trigonometric B-splines can overcome this limitation. It was found that the two-dimensional B-spline cannot provide an accurate approximation, even with a large number of control points (Figure 4.8). Trigonometric B-splines provide a more accurate curve fitting and strain prediction.

The local curvature of pipe profile was calculated from the continuous trigonometric B-spline curve. A local averaging approach with the smoothing span equal to the strain gauge size was applied to smooth the fluctuations of the calculated curvature of B-spline curve. The transverse surface strain along the pipe was calculated by using the mathematical relation between the strain and radius of curvature. The predicted strain values were validated against the strain gauge data.

References

Anand, Vera B. (1993). *Computer Graphics and Geometric Modeling for Engineers.* John Wiley & Sons, Inc.

Casciola, Giulio and Serena Morigi (1996). Spline curves in polar and Cartesian coordinates. In: *Curves and Surfaces with Application in CAGD* (A. Le Mehaute, C. Rabut and L. L. Schumaker, Eds.). Vanderbilt University Press. pp. 1–8.

De Boor, C. (1978). *A Practical Guide to Splines.* Springer.

Deng, H., J. B. Wiskel, A. Ben-Zvi, M. D. Rieder and H. Henein (2008). Strain measurement of a forming process using digital imaging. *Materials Science and Technology.*

Hsu, Q. C. (2003). Comparison of different analysis models to measure plastic strains on sheet metal forming parts by digital image processing. *International Journal of Machine Tools and Manufacture* **43**(5), 515–521.

Koch, Per Erik, Tom Lyche, Marian Neamtu and Larry L. Schumaker (1995). Control curves and knot insertion for trigonometric Splines. *Advances in Computational Mathematics* **3**(1), 405–424.

Noronha, Dauro B., Ricardo R. Martins, Breno P. Jacob and Eduardo Souza (2005). The use of B-spline in the assessment of strain levels associated with plain dents. Instituto Brasileiro de Petroleo e Gas - IBP. Rio de Janeiro, Brazier.

Piegl, Les and Wayne Tiller (1997). *The NURBS Book.* 2nd ed.. Springer.

Pobegailo, A P (1991). Local interpolation with weight functions for variable-smoothness curve design. *Computer-aided design* **23**(8), 579–582.

Rieder, Manuel Dominique (2003). Mathematical modeling of the Bauschinger effect in ERW pipeforming. Master's thesis. University of Alberta.

Sanchez-Reyes, J (1990). Single-valued curves in polar coordinates. *Computer-aided Design* **22**(1), 19–26.

Sanchez-Reyes, J (1992). Single-valued Spline curves in polar coordinates. *Computer-aided Design* **24**(6), 307–315.

Sanchez, Sergio, Jorge Ancheyta and William C. McCaffrey (2007). Comparison of distribution functions for fitting distillation curves of petroleum. *Energy & Fuels* **21**(5), 2955–2963.

Tsai, Du-Ming and Ming-Fong Chen (1994). Curve fitting approach for tangent angle and curvature measurements. *Pattern Recognition* **27**(5), 699–711.

5
Summary and Conclusions

5.1 Summary

The general procedure of the digital imaging based strain measurement technique developed in the present work is summarized in Figure 5.1. The analytical information generated from the technique developed in the present work can be used for the optimum design of roll profiles, pass schedule and forming mills.

Profiles of deformed-skelp obtained at various longitudinal positions along the forming process can be arranged in order to represent the geometric information of the deformed-skelp, as shown in Figure 5.2. This geometric information can be used to adjust the mill setup to optimize the roll profile for uniform deformation.

An alternate view of Figure 5.2 along the forming axis is shown below in Figure 5.3. The roll profiles with a descending pass line provide a flower diagram of the forming process to assess qualitatively the forming operation (non-symmetric behavior, abrupt changes in profile shape, etc.). This information has the potential to be used for the optimal design of roll profiles.

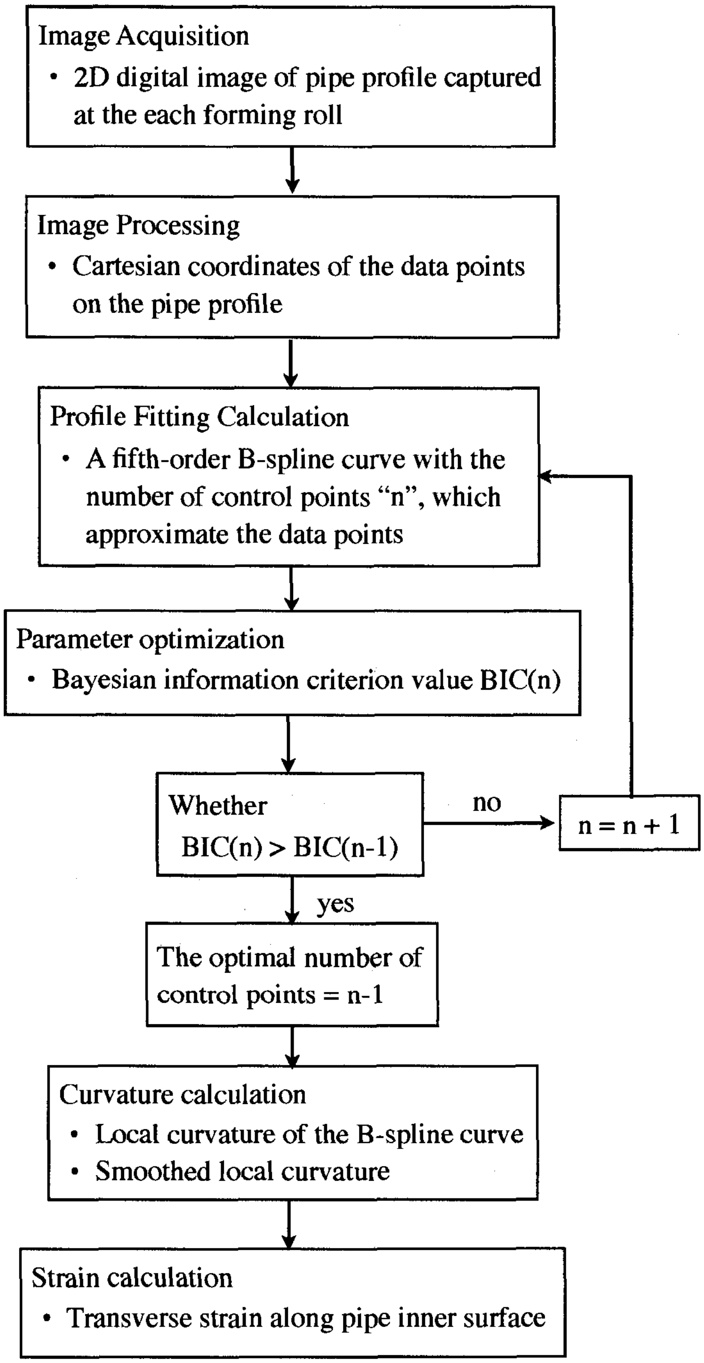


Image Acquisition
• 2D digital image of pipe profile captured at the each forming roll
Image Processing
• Cartesian coordinates of the data points on the pipe profile
Profile Fitting Calculation
• A fifth-order B-spline curve with the number of control points "n", which approximate the data points
Parameter optimization
• Bayesian information criterion value BIC(n)
Whether BIC(n) > BIC(n-1)
no
n = n + 1
yes
The optimal number of control points = n-1
Curvature calculation
• Local curvature of the B-spline curve
• Smoothed local curvature
Strain calculation
• Transverse strain along pipe inner surface

Figure 5.1: Digital imaging strain measurement procedure

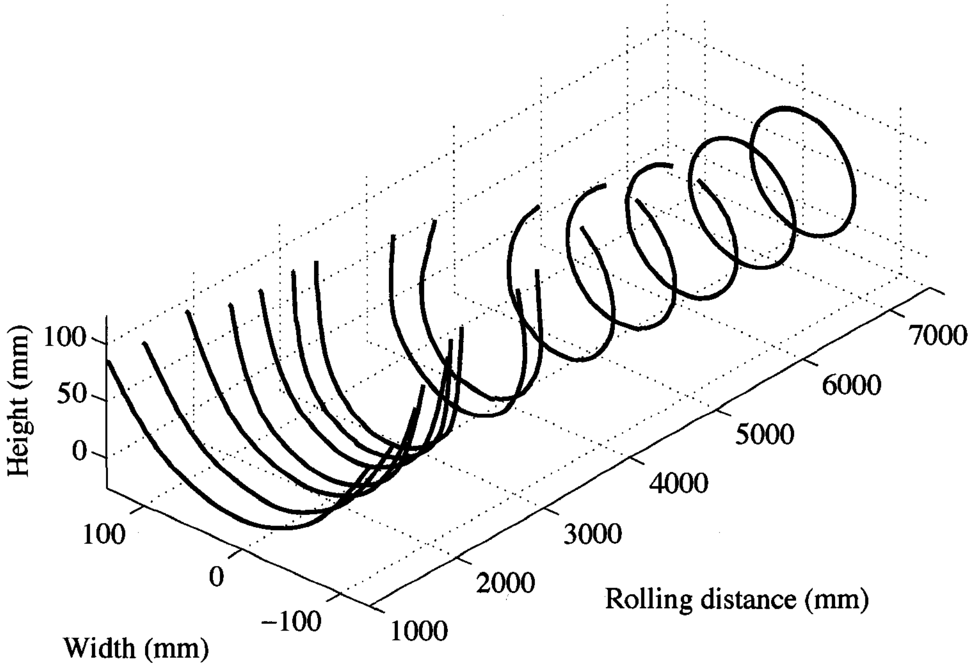


Figure 5.2: Pipe profiles along length of process

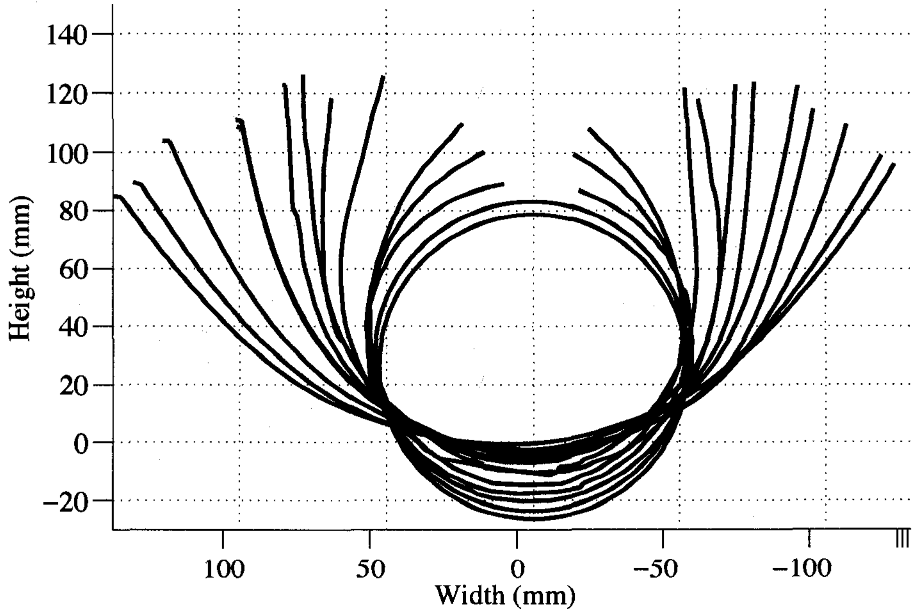


Figure 5.3: Pipe profiles viewed along the forming axis

The strains predicted for the forming process at each stage can be combined in a 3D strain contour plot as shown in Figure 5.4. This type of contour plot can be used to quantitatively assess the overall pipe forming operation or the presence of strain reversals during pipe forming (Wiskel *et al.*, 2008). In addition, the strain distributions throughout the product can be used to analyze the thermo-mechanical mechanisms that lead to defects like cracking, flake etc.

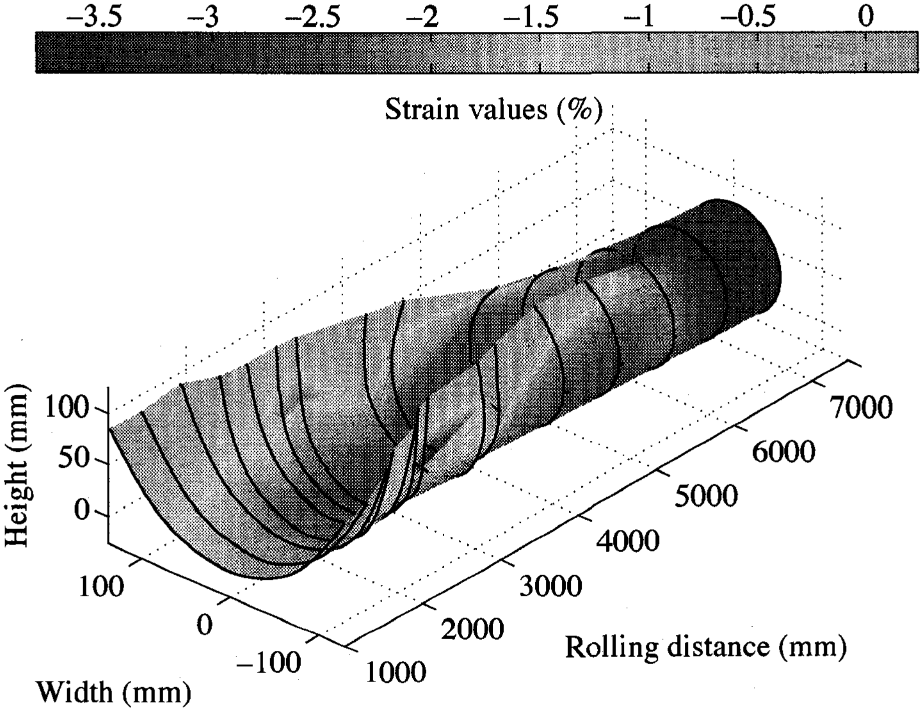


Figure 5.4: 3-D strain contour plot

5.2 Conclusions

Digital imaging techniques can be used to analyze process information from physical objects. The present work has established an effective and accurate digital imaging based technique for the non-invasive evaluation of strain values in forming processes. This technique was successfully applied to an ERW pipe forming operation.

Digital image data obtained from optical equipment are inherently discrete and noisy. In order to obtain an accurate mathematical model of the deformed-skelp geometry from the digital image, three B-spline approximation methods were investigated: one-dimensional, two-dimensional, and trigonometric.

One-dimensional B-spline is the most commonly used approximation method among those evaluated, and it was successfully used to generate the continuous image profile in the first three cage sections of an ERW pipe forming process. However, only curves that project one-to-one onto the x-axis can be described using one-dimensional B-splines.

For curves that do not project one-to-one onto the x-axis, the one-dimensional B-spline needs to be applied to three or more separate segments of the curve, each one less than 180°. Two-dimensional and trigonometric B-splines can overcome this limitation. It was found that the two-dimensional B-spline cannot provide an accurate approximation even with a large number of control points (Figure 4.8). Trigonometric B-splines provide a more accurate curve fitting and curvature prediction.

A fifth degree B-spline was selected in order to guarantee the third degree continuity of the curvature, and to prevent shape perturbations. A nonperiodic knot vector was chosen, as it provides basis functions defined in the complete parameter range. The accuracy of the B-spline approximation depends on the number and location of the control points. The location of the control points was successfully calculated using linear least square regression, avoiding the use of time-consuming iterative optimization methods. The Bayesian information criterion (BIC) was used to optimize the number of control points for the B-spline curve, as BIC provides a trade-off between under and over-fitting. The optimal number of control points for different data set was calculated respectively.

The local curvature of the pipe profile was calculated from the approximated B-spline curve. Due to the numerical errors in the calculation procedure, the calculated curvature has undesirable fluctuations. It was found that a local averaging approach is able to eliminate these fluctuations. The smoothing span was chosen equal to the strain gauge size (10 mm).

The transverse plastic strains along the deformed-skelp were calculated, using the mathematical relation between the strain and the radius of the curvature. The veracity of the proposed technique was confirmed by comparing the calculated deformation strains at each cage section with the measured strain gauge data.

5.3 Recommendations for future work

The digital imaging based technique has been successfully developed and implemented in the present work. In order to achieve more confidence in the understanding of the strain behaviors and their effects on the forming process, more research works are recommended.

1: In the image processing procedure described by Rieder (2003), a line scan was performed moving stepwise by pixel (row, column) in horizontal direction. In a given line, the first instance of a pixel with a gray scale value greater than the critical value would be stored and the scan would restart on the next line. This caused the missing points of the spatial data at the bottom of the curve (Figure 3.3 and 4.3). In the future, this procedure should be done by scanning in both the horizontal and vertical directions.

2: The definition of the transverse strain presented by Noronha *et al.* (2005), was developed with the assumption that the length of the center line of the metal strip will not change during the forming process. However, in the real forming process, the length of the center line will change, especially in the fin regions. This limited the implementation of the proposed method in the fin regions. To overcome this limitation, it is necessary to choose a more accurate strain definition for the roll forming process.

3: For a long time, the design and operation of roll forming processes have been based only on past experience obtained by time-consuming and expensive trial-and-error approaches (Halmos, 2006). Modern trends in the rolling industry have been towards using computer software to simulate and conduct the forming process. It is essential to extend the present work to develop a computer-aided quality-control system. Figure 5.5 shows the conceptual illustration of a computer-aided engineering (CAE) and computer-aided design (CAD) system for roll forming by using the digital imaging method proposed in the present work.

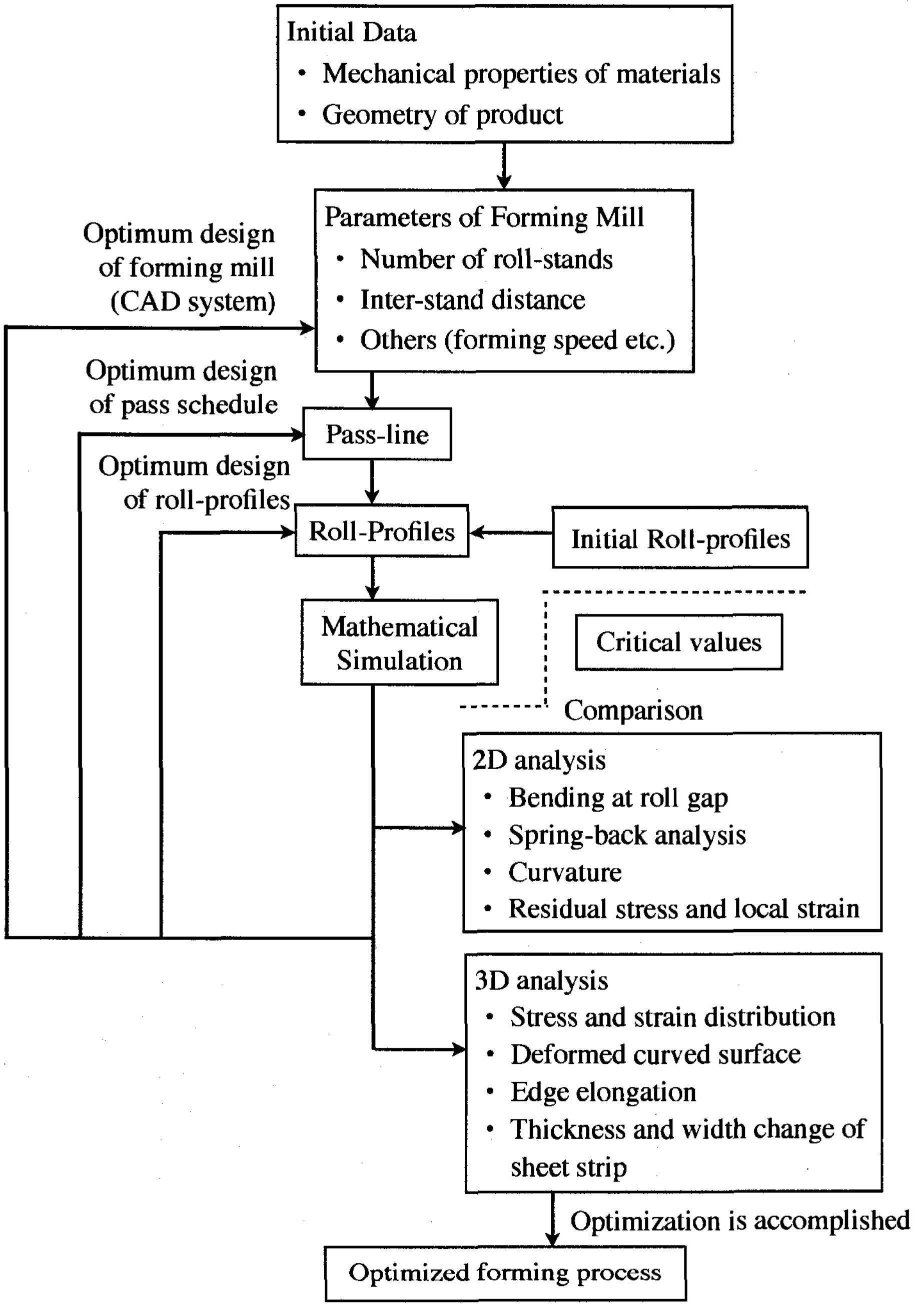

Initial Data
• Mechanical properties of materials
• Geometry of product
Parameters of Forming Mill
• Number of roll-stands
• Inter-stand distance
• Others (forming speed etc.)
Optimum design of forming mill (CAD system)
Optimum design of pass schedule
Pass-line
Optimum design of roll-profiles
Roll-Profiles
Initial Roll-profiles
Mathematical Simulation
Critical values
Comparison
2D analysis
• Bending at roll gap
• Spring-back analysis
• Curvature
• Residual stress and local strain
3D analysis
• Stress and strain distribution
• Deformed curved surface
• Edge elongation
• Thickness and width change of sheet strip
Optimization is accomplished
Optimized forming process

Figure 5.5: Framework of CAE system for roll forming (Halmos, 2006)

References

Halmos, George T., Ed.) (2006). *Roll Forming Handbook.* Taylor & Francis Group, LLC.

Noronha, Dauro B., Ricardo R. Martins, Breno P. Jacob and Eduardo Souza (2005). The use of B-spline in the assessment of strain levels associated with plain dents. Instituto Brasileiro de Petroleo e Gas - IBP. Rio de Janeiro, Brazier.

Rieder, Manuel Dominique (2003). Mathematical modeling of the Bauschinger effect in ERW pipeforming. Master's thesis. University of Alberta.

Wiskel, J.B., H. Deng, M. Rieder, A. Ben-Zvi and H. Henein (2008). Digital imaging of an ERW pipe forming process. In: *Proceedings of IPC2008.* 7th International Pipeline Conference. ASME.

A

MATLAB® scripts for the strain estimation procedure

This appendix presents the MATLAB® implementations of the numerical methods for the digital imaging based strain measurement technique proposed in **Chapter 3** and **Chapter 4**.

A.1 One-dimensional B-spline approximation

```
function [ X, Y_app, K, K_smooth, Arclength, strain ] =
                OneD_approximation( data, n, k, Span, dL )
%===========================================================================
% File: OneD_approximation.m
% Description: One-dimensional B-spline curve approximation.
% Inputs:    data = the digital spatial data (x and y coordinates)
%            n+1  = the number of control points
%            k    = the order of the B-spline curve
%            Span = the smoothing span for curvature smoothing
%            dL   = the distance between the smoothing points
% Outputs:  X         =  the x coordinates of the approximation curve
%           Y_app     =  the y coordinates of the approximation curve
%           K         =  the calculated curvature
%           K_smooth  =  the smoothed curvature
%           Arclength =  the arc length of each point on the curve
%           strain    =  the transverse surface strain values
%=====================================================================
m = n + k ;
a = length(data) ;

for i = 1 : a
    X( i, 1 ) = data( i, 1 ) ;
    Y( i, 1 ) = data( i, 2 ) ;
end
%---------------------------------------------------------------------
% Get the basis functions and the control points
%---------------------------------------------------------------------
p = max( X ) ;
q = min( X ) ;

for i = 1 : a
    [ Blend( i, : ), Blend1( i, : ), Blend2( i, : ) ] =
                      OneD_BSpline( X( i, 1 ), p, q, n, k ) ;
end

Delta =( ( Blend' * Blend ) \ Blend' ) * Y ;

save('data_1D', 'n', 'k', 'p', 'q', 'Delta' )
%---------------------------------------------------------------------
% Curvature calculation and smoothing
%---------------------------------------------------------------------
Y_app = Blend * Delta ;
Y1_app = Blend1 * Delta ;     % first derivative
Y2_app = Blend2 * Delta ;     % second derivative

for i = 1 : a
    k(i,1) = Y2_app(i,1)/(1+Y1_app(i,1)^2)^(3/2) ;
end

for i = 1 : a
    K_smooth(i,1) = OneD_integral(X(i,1)-Span,X(i, 1)+Span)/(2*Span);
    Arclength(i,1) = OneD_Arc_Length(X(a,1),X(i,1)) ;
end
%---------------------------------------------------------------------
% Strain calculation
%---------------------------------------------------------------------
strain = -200 * K_smooth ;
```

```
function [ Blend, Blend1, Blend2 ] =  OneD_BSpline( u, p, q, n, k )
%===========================================================================
% File: OneD_BSpline.m
% Description: the B-spline basis function
% Input:  u = the value of a given point
%         p = the maximum value of control points
%         q = the minimum value of control points
%         n+1 = the number of control points
%         k = order of the B-Spline curve
% Output: Blend = the general B-spline basis function
%         Blend1 = the first derivative of the basis function
%         Blend2 = the second derivative of the basis function
%===========================================================================

%---------------------------------------------------------------------------
% The nonperiodic knot vector
%---------------------------------------------------------------------------
for i = 1 : n+k+1
    if ( i ≤ k )
        t(i) = p ;
    elseif ( k+1 ≤ i && i ≤ n+1 )
        t(i) = q + (i-k) * (p-q) / (n-k+2) ;
    elseif ( i > n+1 )
        t(i) = p ;
    end
end
%---------------------------------------------------------------------------
% The blending function N_i,k(u)
%---------------------------------------------------------------------------
% N_i,1
for i = 1 : k-1
    if ( u == t(i) )
        N( i, 1 ) = 1 ;
    else
        N( i, 1 ) = 0 ;
    end
end

for i = k : n+1
    if ( t(i) ≤ u && u < t(i+1) )
        N( i, 1 ) = 1 ;
    else
        N( i, 1 ) = 0 ;
    end
end

for i = n+2 : n+k
    if ( u == t(i) )
```

```
        N( i, 1 ) = 1 ;
    else
        N( i, 1 ) = 0 ;
    end
end

% N1_i,1, N2_i,1
for i = 1 : n+k
    N1( i, 1 ) = 0 ;
    N2( i, 1 ) = 0 ;
end

% N_i,j
for j = 1 : k-1
    for i = 1 : n+k-j
        if( ( t(i+j)-t(i) ) == 0 && ( t(i+j+1)-t(i+1) ) == 0 )
            N(i,j+1)  = 0 ;
            N1(i,j+1)  = 0 ;
            N2(i,j+1)  = 0 ;
        elseif( ( t(i+j)-t(i) ) == 0 )
            N(i,j+1)  = ((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N(i+1,j) ;
            N1(i,j+1)  = -(1/(t(i+j+1)-t(i+1)))*N(i+1,j)
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N1(i+1,j) ;
            N2(i,j+1)  = -(2/(t(i+j+1)-t(i+1)))*N1(i+1,j)
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N2(i+1,j) ;
        elseif( ( t(i+j+1)-t(i+1) ) == 0 )
            N(i,j+1)  = ((u-t(i))/(t(i+j)-t(i)))*N(i,j) ;
            N1(i,j+1)  = (1/(t(i+j)-t(i)))*N(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N1(i,j) ;
            N2(i,j+1)  = (2/(t(i+j)-t(i)))*N1(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N2(i,j) ;
        else
            N(i,j+1)  = ((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N(i+1,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N(i,j) ;
            N1(i,j+1)  = (1/(t(i+j)-t(i)))*N(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N1(i,j)
                        -(1/(t(i+j+1)-t(i+1)))*N(i+1,j)
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N1(i+1,j) ;
            N2(i,j+1)  = (2/(t(i+j)-t(i)))*N1(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N2(i,j)
                        -(2/(t(i+j+1)-t(i+1)))*N1(i+1,j)
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N2(i+1,j);
        end
    end
end

for i = 1 : n+1
    Blend( i ) = N( i, k ) ;
    Blend1( i ) = N1( i, k ) ;
    Blend2( i ) = N2( i, k ) ;
end

Blend = Blend' ;
Blend1 = Blend1' ;
Blend2 = Blend2' ;
```

```
function K = OneD_integral( lower, upper )
%==========================================================================
% File: OneD_integral.m
% Description: Calculate the integration of the local curvature
%              over a interval [lower, upper]
% Input: lower = lower limit of integration
%        upper = upper limit of integration
% Output: K = the result of the integration
%==========================================================================
K = 0 ;
h = (upper-lower)/50 ;     % the distance of interval

for i = 1 : n
    t(i) = lower + h * i ;
    t(i+1) = lower + h * ( i+ 1 ) ;
    K = K+(h/6)*(OneD_Fun(t(i))+4*OneD_Fun((t(i)+t(i+1))/2)
                                            +OneD_Fun((t(i+1)));
end
```

```
function Kappa = OneD_Fun( u )
%========================================================================
% File: OneD_Fun.m
% Description: the function of local curvature calculation for
%              one-dimensional B-spline curve
% Inputs:  u      = the x coordinates value of a given point
% data_1D.mat includes:
%          p      = the maximum value of the x coordinates
%          q      = the minmum value of the x coordinates
%          n+1    = the number of control points
%          k      = the order of the B-spline curve
%          Delta  = the control points
% Outputs: Kappa  = the local curvature at a given point
%========================================================================
load data_1D.mat

[ N, N1, N2 ] = OneD_BSpline( u, p, q, n, k ) ;

Y = N' * Delta ;
```

```
Y1 = N1' * Delta ;     % first derivative
Y2 = N2' * Delta ;     % second derivative

Kappa = Y2/(1+Y1^2)^(3/2) ;
```

```
function L = OneD_Arc_Length( lower, upper )
%===================================================================
% File: OneD_Arc_Length.m
% Description: Calculate the arc length between the two points on
%              the curve over an interval [lower, upper]
% Input: lower = lower limit of integration
%        upper = upper limit of integration
% Output: L = the arc length
%===================================================================
L = 0 ;
h = (upper-lower)/50 ;     % the distance of interval

for i = 1 : n
    t(i) = lower + h * i ;
    t(i+1) = lower + h * ( i+ 1 ) ;
    L = L+(h/6)*(OneD_Arc_Fun(t(i))+4*OneD_Arc_Fun((t(i)+t(i+1))/2)
                                           +OneD_Arc_Fun((t(i+1)));
end
```

```
function f = OneD_Arc_Fun( u )
%=================================================================
% File: OneD_Arc_Fun.m
% Description: the function of arc length calculation for
%              One-dimensional B-spline curve
% Inputs: u      = the parameter value of a given point
% data_1D.mat includes:
%          n+1   = the number of control points
%          k     = the order of the B-spline curve
%          p     = the maximum value of control points
%          q     = the minimum value of control points
%          Delta = the control points along the x-axis
% Outputs: f = the value of the function at a given point
%=================================================================
load data_1D.mat

[ N, N1, N2 ] = OneD_BSpline( u, p, q, n, k ) ;

f = ( 1 + ( ( N1' * Delta ) ^ 2 ) ) ^ (1/2) ;
```

A.2 Two-dimensional B-spline approximation

```
function [ X, Y, X_app, Y_app, K, K_smooth, Arclength, strain ] =
                  TwoD_approximation( data, n, k, Span, dL )
%==========================================================================
% File: TwoD_approximation.m
% Description: Two-dimensional B-spline curve approximation.
% Inputs:  data = the digital spatial data (x and y coordinates)
%          n+1  = the number of control points
%          k    = the order of the B-spline curve
%          Span = the smoothing span for curvature smoothing
%          dL   = the distance between the smoothing points
% Outputs: X         =  the x coordinates of the digital data
%          Y         =  the y coordinates of the digital data
%          X_app     =  the x coordinates of the approximation curve
%          Y_app     =  the y coordinates of the approximation curve
%          K         =  the calculated curvature
%          Arclength =  the arc length of each point on the curve
%          K_smooth =  the smoothed curvature
%          strain   =  the transverse surface strain values
%==========================================================================
m = n + k ;
a = length( data ) ;

for i = 1 : a
    X( i, 1 ) = data( i, 1 ) ;
    Y( i, 1 ) = data( i, 2 ) ;
end
%--------------------------------------------------------------------------
% Get the basis functions and the control points
%--------------------------------------------------------------------------
for i = 1 : a
     u( i, 1 ) = ( i - 1 ) / ( a - 1 );
end

for i = 1 : a
    [ B( i, : ), B1( i, : ), B2( i, : ) ]
                    = TwoD_BSpline( u( i, 1 ), n, k ) ;
end

Delta_X = ( ( B' * B ) \ B' ) * X ;
Delta_Y = ( ( B' * B ) \ B' ) * Y ;

save('data_2D', 'n', 'k', 'Delta_X', 'Delta_Y' )
%--------------------------------------------------------------------------
% Curvature calculation
%--------------------------------------------------------------------------
```

```
X_app = B * Delta_X ;
Y_app = B * Delta_Y ;

X1_app = B1 * Delta_X ;
Y1_app = B1 * Delta_Y ;

X2_app = B2 * Delta_X ;
Y2_app = B2 * Delta_Y ;

for i = 1 : a
    K(i,1) = abs(x1_app(i,1)*Y2_app(i,1)-Y1_app(i,1)*X2(i,1)) /
                         ((X1_app(i,1)^2+Y1_app(i,1)^2)^(3/2)) ;
end

for i = 1 : a
    K_smooth(i,1) = TwoD_integral(u(i,1)-Span,u(i, 1)+Span)/(2*Span);
    Arclength(i,1) = TwoD_Arc_Length(u(1,1),u(i,1)) ;
end
%----------------------------------------------------------------------
% Strain calculation
%----------------------------------------------------------------------
strain = -200 * K_smooth ;
```

```
function [ Blend, Blend1, Blend2 ] =  TwoD_BSpline( u, n, k )
%======================================================================
% File: TwoD_BSpline.m
% Description: the B-spline basis function
% Input:  u = the value of a given point
%         n+1 = the number of control points
%         k = order of the B-Spline curve
% Output: Blend = the general B-spline basis function
%         Blend1 = the first derivative of the basis function
%         Blend2 = the second derivative of the basis function
%======================================================================

%----------------------------------------------------------------------
% The nonperiodic knot vector
%----------------------------------------------------------------------
for i = 1 : n+k+1
    if ( i ≤ k )
        T(i) = 0 ;
    elseif ( k+1 ≤ i && i ≤ n+1 )
        T(i) = i-k ;
    elseif ( i > n+1 )
        T(i) = n-k+2 ;
    end
end
```

```

t = T / ( max(T) - min(T) ) ;

%------------------------------------------------------------------------------------------
% The blending function N_i,k(u)
%------------------------------------------------------------------------------------------
% N_i,1
for i = 1 : k-1
    if ( u == t(i) )
        N( i, 1 ) = 1 ;
    else
        N( i, 1 ) = 0 ;
    end
end

for i = k : n+1
    if ( t(i) ≤ u && u < t(i+1) )
        N( i, 1 ) = 1 ;
    else
        N( i, 1 ) = 0 ;
    end
end

for i = n+2 : n+k
    if ( u == t(i) )
        N( i, 1 ) = 1 ;
    else
        N( i, 1 ) = 0 ;
    end
end

% N1_i,1, N2_i,1
for i = 1 : n+k
    N1( i, 1 ) = 0 ;
    N2( i, 1 ) = 0 ;
end

% N_i,j
for j = 1 : k-1
    for i = 1 : n+k-j
        if( ( t(i+j)-t(i) ) == 0 && ( t(i+j+1)-t(i+1) ) == 0 )
            N(i,j+1) = 0 ;
            N1(i,j+1) = 0 ;
            N2(i,j+1) = 0 ;
        elseif( ( t(i+j)-t(i) ) == 0 )
            N(i,j+1) = ((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N(i+1,j) ;
            N1(i,j+1) = -(1/(t(i+j+1)-t(i+1)))*N(i+1,j)
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N1(i+1,j) ;
            N2(i,j+1) = -(2/(t(i+j+1)-t(i+1)))*N1(i+1,j)
```

```
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N2(i+1,j) ;
        elseif( ( t(i+j+1)-t(i+1) ) == 0 )
            N(i,j+1) = ((u-t(i))/(t(i+j)-t(i)))*N(i,j) ;
            N1(i,j+1) = (1/(t(i+j)-t(i)))*N(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N1(i,j) ;
            N2(i,j+1) = (2/(t(i+j)-t(i)))*N1(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N2(i,j) ;
        else
            N(i,j+1) = ((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N(i+1,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N(i,j) ;
            N1(i,j+1) = (1/(t(i+j)-t(i)))*N(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N1(i,j)
                        -(1/(t(i+j+1)-t(i+1)))*N(i+1,j)
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N1(i+1,j) ;
            N2(i,j+1) = (2/(t(i+j)-t(i)))*N1(i,j)
                        +((u-t(i))/(t(i+j)-t(i)))*N2(i,j)
                        -(2/(t(i+j+1)-t(i+1)))*N1(i+1,j)
                        +((t(i+j+1)-u)/(t(i+j+1)-t(i+1)))*N2(i+1,j);
        end
    end
end

for i = 1 : n+1
    Blend( i ) = N( i, k ) ;
    Blend1( i ) = N1( i, k ) ;
    Blend2( i ) = N2( i, k ) ;
end

Blend = Blend' ;
Blend1 = Blend1' ;
Blend2 = Blend2' ;
```

```
function K = TwoD_integral( lower, upper )
%===========================================================================
% File: TwoD_integral.m
% Description: Calculate the integration of the local curvature
%               over an interval [lower, upper]
% Input: lower = lower limit of integration
%        upper = upper limit of integration
% Output: K = the result of the integration
%===========================================================================
K = 0 ;
h = (upper-lower)/50 ;     % the distance of interval

for i = 1 : n
    t(i) = lower + h * i ;
    t(i+1) = lower + h * ( i+ 1 ) ;
    K = K+(h/6)*(TwoD_Fun(t(i))+4*TwoD_Fun((t(i)+t(i+1))/2)
                                        +TwoD_Fun((t(i+1)));
end
```

```
function Kappa = TwoD_Fun( u )
%==================================================================
% File: TwoD_Fun.m
% Description: the function of local curvature calculation for
%              two dimensional B-spline curve
% Inputs:  u      = the parameter value of a given point
% data_2D.mat includes:
%          n+1    = the number of control points
%          k      = the order of the B-spline curve
%          Delta_X = the control points along the x-axis
%          Delta_Y = the control points along the y-axis
% Outputs: Kappa = the local curvature at a given point
%==================================================================
load data_2D.mat

[ N, N1, N2 ] = TwoD_BSpline( u, n, k ) ;

X_app = N' * Delta_X ;
Y_app = N' * Delta_Y ;

X1_app = N1' * Delta_X ;
Y1_app = N1' * Delta_Y ;

X2_app = N2' * Delta_X ;
Y2_app = N2' * Delta_Y ;

Kappa = abs(X1_app*Y2_app-Y1_app*X2_app) /
                         ((X1_app^2+Y1_app^2)^(3/2)) ;
```

```
function L = TwoD_Arc_Length( lower, upper )
%====================================================================
% File: TwoD_Arc_Length.m
% Description: Calculate the arc length between the two points on
%              the curve over an interval [lower, upper]
% Input: lower = lower limit of integration
%        upper = upper limit of integration
% Output: L = the arc length
%====================================================================
L = 0 ;
h = (upper-lower)/50 ;     % the distance of interval

```

```
for i = 1 : n
    t(i) = lower + h * i ;
    t(i+1) = lower + h * ( i+ 1 ) ;
    L = L+(h/6)*(TwoD_Arc_Fun(t(i))+4*TwoD_Arc_Fun((t(i)+t(i+1))/2)
                                          +TwoD_Arc_Fun((t(i+1)));
end
```

```
function f = TwoD_Arc_Fun( u )
%==========================================================================
% File: TwoD_Arc_Fun.m
% Description: the function of arc length calculation for
%              two-dimensional B-spline curve
% Inputs:  u     = the parameter value of a given point
% data_2D.mat includes:
%          n+1   = the number of control points
%          k     = the order of the B-spline curve
%          Delta_X = the control points along the x-axis
%          Delta_Y = the control points along the y-axis
% Outputs: f = the value of the function at a given point
%==========================================================================
load data_2D.mat

[ N, N1, N2 ] = TwoD_BSpline( u, n, k ) ;

f = (((N1'*Delta_X)^2)+((N1'*Delta_Y)^2))^(1/2) ;
```

A.3 Trigonometric B-spline approximation

```
function [X_app, Y_app, K, K_smooth, Arclength, strain] =
            Tri_approximation( data, n, k, Span, dL )
%==========================================================================
% File: Tri_approximation.m
% Description: Trigonometric B-spline curve approximation.
% Inputs:     data = the digital spatial data (x and y coordinates)
%             n+1  = the number of control points
%             k    = the order of the B-spline curve
%             Span = the smoothing span for curvature smoothing
%             dL   = the distance between the smoothing points
% Outputs:    X_app      =  the x coordinates of the approximation curve
%             Y_app      =  the y coordinates of the approximation curve
%             K          =  the calculated curvature
%             K_smooth   =  the smoothed curvature
%           Arclength =  the arc length of each point on the curve
%           strain    =  the transverse surface strain values
%=========================================================================
m = n + k ;
a = length(data) ;

for i=1 : a
    X(i,1)=data(i,1) ;
    Y(i,1)=data(i,2) - max(data(:,2)) / 2 ;
end
%-------------------------------------------------------------------------
% Get the radius and angular for each point
%-------------------------------------------------------------------------
for i=1 : a
    R(i,1)=sqrt(X(i,1)^2 + Y(i,1)^2) ;
    if(X(i,1)>0 && Y(i,1) ≥ 0)
        Theta(i,1)=atan(Y(i,1)/X(i,1)) ;
     elseif(X(i,1)>0 && Y(i,1)<0)
        Theta(i,1)=atan(Y(i,1)/ X(i,1))  ;
    elseif(X(i,1)<0)
        Theta(i,1)=atan(Y(i,1)/ X(i,1)) - pi ;
    elseif(X(i,1) == 0 && Y(i,1)>0)
        Theta(i,1)=-3*pi/2 ;
    elseif(X(i,1) == 0 && Y(i,1)<0)
        Theta(i,1)=-pi/2 ;
    end
end
%-------------------------------------------------------------------------
% Get the basis functions and the control points
%-------------------------------------------------------------------------
p = min(Theta) ;
q = max(Theta) ;

for i = 1 : a
    [ B(i,:), B1(i,:), B2(i,:) ] =
                 Tri_BSpline( Theta(i,1), p, q, n, k ) ;
end

Delta =((B' * B) \ B') * R ;
R_app = B * Delta ;

for i=1 : a
    X_app(i,1)=R_app(i,1)*cos(Theta(i,1)) ;
    Y_app(i,1)=R_app(i,1)*sin(Theta(i,1)) + max(data(:,2)) / 2 ;
end

save( 'data_tri', 'p', 'q', 'n', 'k', 'Delta')
%-------------------------------------------------------------------------
% Curvature calculation and smoothing
```

```
%---------------------------------------------------------------------------
for i=1 : a
    K(i,1)=abs((R_app(i,1)^2+2*(R1_app(i,1)^2)
                             -R_app(i,1)*R2_app(i,1))) /
                          ((R_app(i,1)^2+R1_app(i,1)^2)^(3/2)) ;
end

for i = 1 : a
    K_smooth(i,1) = Tri_integral(Theta(i,1)-Span,Theta(i, 1)+Span)
                                                            /(2*Span);
    Arclength(i,1) = Tri_Arc_Length(Theta(a,1),Theta(i,1)) ;
end
%---------------------------------------------------------------------------
%  Strain calculation
%---------------------------------------------------------------------------
strain = -200 * K_smooth ;
```

```
function  [ B, B1, B2 ] = Tri_BSpline( u, p, q, n, k )
%=========================================================================
% File: Tri_BSpline.m
% Description: the B-spline basis function
% Input:  u = the angular coordinates of a given point
%         p = the minimum value of the anglar coordinates
%         q = the maximum value of the angular coordinates
%         n+1 = the number of control points
%         k = the order of the B-spline
% Output: B = trigonometric B-spline basis function
%         B1 = the first derivative of the basis function
%         B2 = the Second derivative of the basis function
%=========================================================================

%-------------------------------------------------------------------------
% Knot vectors
%-------------------------------------------------------------------------
for i = 1 : n+k+1
    if (i ≤ k)
        t(i) = p;
    elseif (k+1 ≤ i && i ≤ n+1)
        t(i) = p+(i-k)*(q-p)/(n-k+2);
    elseif (i > n+1)
        t(i) = q;
    end
end
%-------------------------------------------------------------------------
% The basis function
%-------------------------------------------------------------------------
for i = 1 : k-1
```

```
    if (u == t(i))
        N(i,1) = 1;
    else
        N(i,1) = 0;
    end
end

for i = k : n+1
    if (t(i) ≤ u && u < t(i+1))
        N(i,1) = 1;
    else
        N(i,1) = 0;
    end
end

for i = n+2 : n+k
    if (u == t(i))
        N(i,1) = 1;
    else
        N(i,1) = 0;
    end
end

% N1_i,1,N2_i,1
for i = 1 : n+k
    N1(i,1) = 0;
    N2(i,1) = 0;
end

% N_i,2


% N_i,j
for j = 1 : k-1
    for i = 1 : n+k-j
        if((t(i+j)-t(i)) == 0 && (t(i+j+1)-t(i+1)) == 0)
            N(i,j+1) = 0;
            N1(i,j+1) = 0;
            N2(i,j+1) = 0;
        elseif((t(i+j)-t(i)) == 0)
            N(i,j+1)=(sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N(i+1,j);
            N1(i,j+1)=(-cos(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N(i+1,j)
                +(sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N1(i+1,j);
            N2(i,j+1)=(-sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N(i+1,j)
              +(-2*cos(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N1(i+1,j)
              +(sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N2(i+1,j);
        elseif((t(i+j+1)-t(i+1)) == 0)
            N(i,j+1)=(sin(u-t(i))/sin(t(i+j)-t(i)))*N(i,j);
            N1(i,j+1)=(cos(u-t(i))/sin(t(i+j)-t(i)))*N(i,j)
```

```
              +(sin(u-t(i))/sin(t(i+j)-t(i)))*N1(i,j);
            N2(i,j+1)=(-sin(u-t(i))/sin(t(i+j)-t(i)))*N(i,j)
               +(2*cos(u-t(i))/sin(t(i+j) -t(i)))*N1(i,j)
               +(sin(u-t(i))/sin(t(i+j)-t(i)))*N2(i,j);
        else
            N(i,j+1)=(sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N(i+1,j)
                    +(sin(u-t(i))/sin(t(i+j)-t(i)))*N(i,j);
            N1(i,j+1)=(-cos(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N(i+1,j)
               +(sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N1(i+1,j)
               +(cos(u-t(i))/sin(t(i+j)-t(i)))*N(i,j)
               +(sin(u-t(i))/sin(t(i+j)-t(i)))*N1(i,j);
            N2(i,j+1)=(-sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N(i+1,j)
               +(-2*cos(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N1(i+1,j)
               +(sin(t(i+j+1)-u)/sin(t(i+j+1)-t(i+1)))*N2(i+1,j)
               +(-sin(u-t(i))/sin(t(i+j)-t(i)))*N(i,j)
               +(2*cos(u-t(i))/sin(t(i+j) -t(i)))*N1(i,j)
               +(sin(u-t(i))/sin(t(i+j)-t(i)))*N2(i,j);
        end
    end
end

for i = 1 : n+1
    Blend(i) = N(i,k);
    Blend1(i) = N1(i,k);
    Blend2(i) = N2(i,k);
end

B = Blend';
B1 = Blend1';
B2 = Blend2';
```

```
function K = Tri_integral( lower, upper )
%===================================================================
% File: Tri_integral.m
% Description: Calculate the integration of the local curvature
%              over a interval [lower, upper]
% Input: lower = lower limit of integration
%        upper = upper limit of integration
% Output: K = the result of the integration
%===================================================================
K = 0 ;
h = (upper-lower)/50 ;     % the distance of interval

for i = 1 : n
    t(i) = lower + h * i ;
    t(i+1) = lower + h * ( i+ 1 ) ;
    K = K+(h/6)*(Tri_Fun(t(i))+4*Tri_Fun((t(i)+t(i+1))/2)
                                                 +Tri_Fun((t(i+1)));
end
```

```
function Kappa = Tri_Fun( u )
%=====================================================================
% File: Tri_Fun.m
% Description: the function of local curvature calculation for
%              trigonometric B-spline curve
% Inputs:  u     = the angular coordinate value of a given point
% data_tri.mat includes:
%          p     = the minimum value of the angular coordinates
%          q     = the maximum value of the angular coordinates
%          n+1   = the number of control points
%          k     = the order of the B-spline curve
%          Delta = the control points
% Outputs: Kappa = the local curvature at a given point
%=====================================================================
load data_tri.mat

[ N, N1, N2 ] = Tri_BSpline( u, p, q, n, k ) ;

r = N' * Delta ;
r1 = N1' * Delta ;
r2 = N2' * Delta ;

Kappa = abs((r^2+2*(r1^2)-r*r2))/((r^2+r1^2)^(3/2)) ;
```

```
function L = Tri_Arc_Length( lower, upper )
%=======================================================================
% File: Tri_Arc_Length.m
% Description: Calculate the arc length between the two points on
%              the curve over an interval [lower, upper]
% Input: lower = lower limit of integration
%        upper = upper limit of integration
% Output: L = the arc length
%=======================================================================
L = 0 ;
h = (upper-lower)/50 ;     % the distance of interval

for i = 1 : n
    t(i) = lower + h * i ;
    t(i+1) = lower + h * ( i+ 1 ) ;
    L = L+(h/6)*(Tri_Arc_Fun(t(i))+4*Tri_Arc_Fun((t(i)+t(i+1))/2)
                                              +Tri_Arc_Fun((t(i+1)));
end
```

```
function f = Tri_Arc_Fun( u )
%==========================================================================
% File: Tri_Arc_Fun.m
% Description: the function of arc length calculation for
%              trigonomentric B-spline curve
% Inputs:  u     = the parameter value of a given point
% data_tri.mat includes:
%          n+1   = the number of control points
%          k     = the order of the B-spline curve
%          p     = the minimum value of the angular coordinates
%          q     = the maximum value of the angular coordinates
%          Delta = the control points
% Outputs: f = the value of the function at a given point
%==========================================================================
load data_tri.mat

[ N, N1, N2 ] = Tri_BSpline( u, p, q, n, k ) ;

f = ( ( ( N' * Delta ) ^ 2 + ( ( N1' * Delta ) ^ 2 ) ) ^ (1/2) ;
```

B

Strain gauge test data

Table B.1: Strain gauge data for CSA Grade 359, O.D.=114.3 mm, t=4.0 mm

location	cage1	cage2	cage3	cage4	cage5	cage6	cage7	cage8	cage9
12:45	-0.32	-0.30	0.04	0.08	0.13	0.06	0.09	0.10	0.00
1:30	-0.64	-0.67	-0.68	-0.63	-0.52	-0.63	-0.53	-0.56	-0.38
3:00	-1.14	-1.14	-1.69	-1.74	-1.54	-1.93	-1.88	-1.82	-1.98
4:30	-1.71	-1.74	-1.99	-2.07	-2.00	-2.31	-2.36	-2.33	-3.34
6:00	-1.35	-1.51	-1.64	-1.92	-2.09	-2.28	-2.77	-3.04	-3.28
9:00	-0.94	-1.04	-1.07	-1.29	-1.22	-1.51	-1.51	-1.47	-1.40
10:30	-0.63	-0.69	-0.72	-0.77	-0.68	-0.58	-0.63	-0.47	-0.48
11:45	-0.35	-0.45	-0.48	-0.57	-0.42	-0.42	-0.33	-0.37	-0.33