Thesis for the Degree of Licentiate of Engineering

Geometric Modelling of Form and Positional Errors of Mechanical Parts

Henrik Johansson

Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden
Göteborg, September 1999

Geometric Modelling of Form and Positional Errors of Mechanical Parts
HENRIK JOHANSSON

ISSN 0347-2809/NO 1999-36
Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31-772 1000

This work was supported by the Vehicular Research Programme
at the Swedish National Board for Industrial and Technical Development
(Programrådet för fordonsforskning, NUTEK) under grant 8531P-95-8870.

Matematiskt centrum
Göteborg 1999

Abstract

In manufacturing engineering, geometric quality assurance is carried out through dimensional inspection. A common way of inspecting mechanical parts is to take a set of discrete measurements on the surface of a manufactured part with a coordinate measurement machine (CMM). This report deals with geometrical issues related to dimensional inspection.

First, we describe how to recover the position of a mechanical part by analysing a set of discrete measurements. This method, known as the rigid best fit, can be used to remove a systematic error in the measurements that is due to incorrect positioning of a part.

Second, we study a special case of the rigid best fit problem, where the localisation algorithm is applied to the datum system of a part. A datum system defines a local coordinate system of a part. This report contributes to the stability analysis of datum systems in new ways of measuring and presenting both qualitative and quantitative information about the stability of a datum system. The stability information is suitable for visualisation in a computer-aided design (CAD) environment.

Third, we generalise the rigid best fit problem. The generalised model considers not only the information about part location but also the geometric form error inherent in a discrete set of CMM measurements of a mechanical part. The report is concerned with the special case when the part under consideration can be described by a curve. The restriction to curves is motivated by the possibility of applying the method to cross-sections of three-dimensional objects. The generalised localisation algorithm, the flexible best fit, creates a mathematical model of the form error that can be visualised in a CAD environment in parallel to the nominal CAD-model. The flexible best fit algorithm is useful in studying form errors of prototype parts. This report also demonstrates the performance of the flexible best fit in a case study with cross-sections taken from an automotive roof with CMM data from a corresponding prototype part.

The problems are approached from a practical, numerical viewpoint and algorithms and numerical examples are presented whenever possible.

Keywords: geometric modelling, datum system, least squares, quality assurance, rigid body motion, best fit, coordinate metrology, deformable models

AMS 1991 subject classification: 65D10 (65D17, 41A63)

Acknowledgements

First of all I would like to express my thanks to my advisors Associate Professor Bo I. Johansson at the Department of Mathematics at Chalmers University of Technology (CTH) and Daniel Johansson, formerly at Volvo Car Corporation (VCC) but currently at Volvo France. Daniel Johansson has paved the way for this project at VCC by his theoretical work and his enthusiasm on the benefits that can be achieved by geometric analysis in this field. I want to thank Bo I. Johansson for the many discussions we have had and for his encouragement. I am also indebted to him for his most valuable comments and suggestions to this report.

Furthermore, I am grateful to the Project Group that inspired me during our meetings. The group is chaired by Robert Häller (VCC) and consists of Andreas Westholm, Lars-Olof Johansson and Niklas Anderson at Volvo Information Technology, Professor Axel Ruhe and Associate Professor Bo I. Johansson at the Department of Mathematics (CTH) and Associate Professor Rikard Söderberg at the Department of Machine and Vehicle Design (CTH).

Thanks also to Patrik Högström, Roger Andersson and Stefan Demenfors at VCC and Johan Persson at Caran Systems AB for supplying me with data for the case study.

Finally, I would like to express my sincere gratitude to my fiancée Johanna for her continuous support and love throughout the duration of this project.

Contents

Chapter 1

Introduction

Geometric quality assurance is an essential part of the manufacturing process of mechanical parts. The geometric dimensions of the parts need to be checked for conformance to their specifications.

1.1 Historical Background

The geometric specifications consist of fourteen different types of tolerance specifications (ISO 1101, 1993). Until the breakthrough of the coordinate measuring machines (CMM), it was common to check conformance to the geometric specification using *hard gauges*. The appearance of CMMs enabled for inspection of general, sculptured surfaces by sampling discrete points on a part's surface. The new way of inspecting mechanical parts eliminates the time consuming and very expensive use and manufacture of high precision hard gauges, specific to any toleranced geometric feature of a part. Moreover, the CMM inspection methodology, also known as *soft gauging*, allows for automation of the inspection operations. In short, the appearance of CMMs meant a revolution in dimensional metrology.

Despite the potential offered by CMMs, there still seems to be some work to do in connecting the discrete coordinate measurements from the CMM with the tolerance specifications in determining the geometric quality of a mechanical part. Many of the problems that appeared in the early days of the CMMs have started to be overcome, whereas others are yet to be dealt with. The problem of relating the CMM data and the geometrical tolerances is due to that the tolerancing methodology was developed well before CMMs came into use. Hence, they implicitly assume that hard gauges are used for the inspection (Dowling et al., 1997).

1.2 Localisation by Fixture

Before starting to evaluate the geometric dimensions of a part, the part has to be localised; i.e. we have to find the position of the part. The part's position is

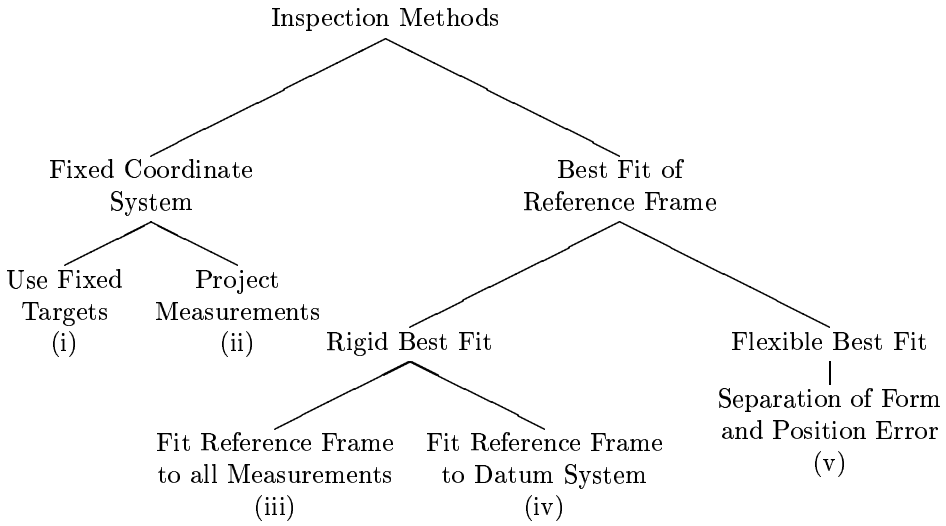

Figure 1.1: The tree illustrates different approaches to coordinate inspection.

defined by the location of the datum[1] system of the part. The datum system is defined by a set of *datum features*. A datum feature can, for instance, be a plain surface or a hole, see Figures 3.1–3.3. The datum features are important since they uniquely define a local coordinate system of the part. In the sequel, we also use the term *locating feature* to mean datum feature. A well-defined datum system can be mathematically decomposed into its *datum points* and corresponding *guiding directions*. Each and one of the six datum points is associated with a guiding direction. For example, if the datum feature is a plain surface, then the guiding direction of the datum feature is the principal normal of the surface. A hole, on the other hand, is mathematically described by a datum point defining the centre of the hole in addition to the two guiding directions that span the tangent plane of the hole. Since the hole restricts two degrees of freedom, it is said to have multiplicity two. There are also datum features with multiplicity three. The number of datum points for a well-defined datum system, counted with multiplicity, is six. The concept of a datum system is explained more carefully in Chapter 3.

In the trivial case the workpiece is put to rest on a fixture with known position that supports the part in its datum points. The inspection points are sampled with a CMM and the form error is evaluated by (i) taking the difference between the sampled measurements and their corresponding target points in the nominal model. Alternatively, if the target points are unknown we can (ii) project the sampled measurements in the nominal model, see Figure 1.1.

The drawback of this approach is that it requires a high precision fixture to be

[1] "**datum**. A point, line or surface to which dimensions are referred on engineering drawings and from which measurements are taken in machining or other engineering operations." from *Chambers Science and Technology Dictionary*, W & R Chambers Ltd., Edinburgh, 1988.

manufactured for each type of part subject to inspection. Thus, we have faced the same problem as manufacturing engineers had before the CMMs, i.e. the problem of having to waste time and money to prepare high precision tools just to be able to find out about the dimensions of the part. In this case, we have to prepare high accuracy fixtures, whereas before the CMMs the manufacturing engineers needed to prepare hard gauges.

1.3 Virtual Localisation

A great save in time and money is achieved by using the CMM measurements not only to evaluate the dimensions of the part but also to localise the part. By localisation we mean the process of locating the nominal object in the position where the (squared) sum of distances from the coordinate measurements to their target points in the nominal model take on its minimum. If the squared sum is minimised then we speak of a *least squares* minimisation. This process is also termed *rigid best fit* or simply *best fit*. The distance between the measurements and their target points are denoted *residuals* in the sequel.

The residuals of the best fit can be interpreted as the geometric deviation and compared with the geometric specification to check for conformance. This is method (iii) in the tree shown in Figure 1.1. The problem can be formulated as a least squares problem, which is subject to analysis in Chapter 2. One has to be careful when choosing inspection points since the solution of the least squares problem is sensitive to their distribution and, of course, the number of them.

An underlying assumption of method (iii) that we make in Chapter 2 is that the form errors are normally distributed and independent, a condition very rarely satisfied in reality where we usually have systematic form errors. For example, if a part deviates at a certain point the part is likely to deviate in a similar way in a close neighbourhood to the point. Put differently, there are spatial dependencies between measurements that are close in a geodetic sense. However, the assumption about normality and independence are likely to be correct for the part of the error that is due to measurement noise. On the other hand, the measurement noise is probably orders of magnitude smaller than the geometric deviations.

Method (iv) is a natural compromise of methods (ii) and (iii) where we inspect the part as usual, with the difference that the datum points are inspected as well. Secondly, we use the measurements of the datum points to find the rigid body transform that takes the workpiece from its position in the real world to the position of the nominal model in the CAD-environment. The rigid transform is again interpreted as the positional error. Thirdly, we evaluate the shape error by projecting the measurements in the surface of the nominal part as in method two. This method has the advantage of allowing for positional error without imposing any restrictions of the inspection points in terms of uniformness and so forth, since the inspection points are not involved in the least squares minimisation. Moreover, the six degrees of freedom of the rigid body transform is only determined by measurements of the datum points. In effect, this means that the system is not over-determined and hence no assumptions of independence and normality

has to be made since the residual of the non-linear least squares problem will be zero. On the other hand, the rigid body transform will be completely determined by the measurements of the datum points and hence errors purely related to the measurement process will be directly forwarded to the solution of the least squares problem. A slight enhancement of method (iv) to handle this problem would be to take several measurements of every datum point. Possible differences between measurements of the same point will then only reflect measurement noise, which can be assumed independent and normally distributed. Methods (iii) and (iv), also known as the rigid best fit, is the subject of Chapter 2 where we review a model for representing the position of a part and present algorithms with which it can be computed.

1.4 Form Error Modelling

The rigid best fit only allows for a rigid alignment of the nominal model to CMM data. To this end, we introduce another method in Chapter 4 that goes one step further and allows the model to be deformed. We call this method *flexible best fit* as opposed to the rigid best fit. Using this technique we can create a global model of the form error as well as the positional error, discard the measurements and then use this mathematical model of the geometric deviations to check for conformance to the geometric specifications. By applying conditions on the estimated form error, we can separate the form error from the positional error. This idea is represented by leaf (v) in the tree of Figure 1.1 and is developed in Chapter 4. However, due to the complexity of the method it is only applied to composite curves, which can be thought of as cross-sections of true three-dimensional objects.

The method applies to the early phases in the development of a new product. More specifically, it can be applied in the prototype phase in order to visualise the form error in an appealing way. The cause of the form error can then easily be pinpointed and correct adjustments made to e.g. a die used in a stamping operation to form a sheet metal part.

1.5 The Importance of Robust Datum Systems

The local coordinate system of a part as well as most of its geometric specifications is determined relative to the part's datum system. Moreover, during the inspection and many manufacturing operations involving the part, it is positioned by fixtures that support the object in its datum points. These facts reflect the importance of the datum system. Chapter 3 presents an application of the least squares rigid best fit method from Chapter 2 to datum system analysis.

The datum system is chosen at an early stage of the design process. A tool of the type presented in Chapter 3 is easy to use and provides the manufacturing engineer with graphical information about the quality of the datum system. Such a tool makes it possible to pinpoint deficiencies of a datum system early and can prevent problems to come up later in the manufacturing process that might arise if parts with faulty datum systems reach the production phase.

Chapter 2

Rigid Best Fit of Sculptured Curves and Surfaces

In this chapter, we try to solve the problem of finding the rigid body transformation of a discrete data set that best fits a curve or surface in some norm. Our aim is to find and remove the inherent rigid body transform of the data set and then interpret the residuals of the data as geometric defects. This problem arises naturally in quality assurance of mechanical parts in which case the discrete data set is produced by a coordinate measuring machine (CMM). The ideas presented are inspired by Johansson (1998) but here developed in some more detail.

2.1 Related Work

The excellent survey by Dowling et al. (1997) gives an account of current approaches to geometric feature inspection. They have reviewed work from several engineering fields and describe orthogonal least squares and part localisation as well as statistical analysis. A recent article by Rivest (1998) provides statistical analysis of an algorithm for localisation of prismatic parts. Through a linearisation of the used model, he derives statistical tests for geometric quality assurance based on the residual characteristics.

Apart from researchers in the statistical community, researchers from numerical analysis have paid attention to the localisation problem. One of the earlier publications worth mentioning is due to Hanson and Norris (1981), who use the singular value decomposition (SVD) to solve the localisation problem. They give an explicit, direct solution to the problem of matching two point clouds. The rigid transform is obtained through an application of the SVD. Zwick (1997) gives a short but complete overview of current research from numerical analysis.

As a representative from mechanical engineering we have chosen to put forward Yau (1997), who provides a best fit algorithm and a method for sensitivity analysis of the estimated parameters.

2.2 Formulating the Least Squares Problem

Consider a parameterised regular surface $\boldsymbol{c} : D \to \mathcal{M} \subset \mathbb{R}^3$, where D is a closed, bounded and connected subset of $\mathbb{R}^2$, in the simplest case $D = \{(u, v) \in \mathbb{R}^2 : 0 \leq u, v \leq 1\}$, the unit square. We require the surface to have continuously varying curvature, i.e. $\boldsymbol{c} \in C^2(D)$. We assume $\mathcal{M}$ to be our nominal CAD-model from which we manufacture a workpiece $\mathcal{P}$. The physical object $\mathcal{P}$ is allowed to differ from $\mathcal{M}$ to some extent. The allowed difference is defined by the geometrical tolerances with which the model $\mathcal{M}$ is associated. In order to estimate the geometric quality of $\mathcal{P}$ relative to $\mathcal{M}$ we take measurements $\boldsymbol{p}_j \in \mathbb{R}^3$, $j = 1, \ldots, n$ of $\mathcal{P}$ by, e.g. a coordinate measurement machine (CMM). The problem at hand is to compare the discrete measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ with the corresponding *target* points in the nominal model $\mathcal{M}$. The target points are not known in advance but are found through *least distance projection* of the coordinate measurements on the nominal model. By the least distance projection of a point $\boldsymbol{p} \in \mathbb{R}^3$ we mean the point in the object $\mathcal{M}$ that is closest to $\boldsymbol{p}$. We also sometimes speak of the *closest point* to $\boldsymbol{p}$. The problem of finding the least distance projection is denoted the *least distance problem* by Helfrich and Zwick (1996).

The measurements are contaminated by errors from three sources; (i) measurement noise $\boldsymbol{\varepsilon}$ from the CMM, (ii) shape errors $\boldsymbol{\gamma}$ from the physical object $\mathcal{P}$, and (iii) the localisation error $\mathcal{R}(\boldsymbol{x})$ that we want to estimate, i.e. the rigid body transform.

2.2.1 A Statistical Model

Let $\boldsymbol{u}_j = [u_j \; v_j]^T \in D$ denote the surface parameter of the closest point to $\boldsymbol{p}_j$ in $\mathcal{M}$ for all $j = 1, \ldots, n$. The measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ can be thought of as satisfying

$$\boldsymbol{p}_j = \mathcal{R}(\tilde{\boldsymbol{x}})\boldsymbol{c}(\boldsymbol{u}_j) + \boldsymbol{\varepsilon}_j + \boldsymbol{\gamma}_j, \quad j = 1, \ldots, n,$$

where $\boldsymbol{\varepsilon}_j \in \mathbb{R}^3$ represents the measurement error and $\boldsymbol{\gamma}_j \in \mathbb{R}^3$ the form error at the point of $\mathcal{P}$ corresponding to $\boldsymbol{c}(\boldsymbol{u}_j)$. Here, $\mathcal{R}(\tilde{\boldsymbol{x}})$ is a rigid body transform as defined in Appendix C. The rigid body transform has parameters $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{t}}^T \; \tilde{\boldsymbol{\omega}}^T]^T$ and act on a space point $\boldsymbol{p} \in \mathbb{R}^3$ like $\mathcal{R}(\tilde{\boldsymbol{x}})\boldsymbol{p} = \tilde{\boldsymbol{t}} + \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{p}$, where $\boldsymbol{R}(\tilde{\boldsymbol{\omega}})$ is a 3×3 rotation matrix parameterised by the rotation vector $\tilde{\boldsymbol{\omega}} \in \mathbb{R}^3$ and $\tilde{\boldsymbol{t}} \in \mathbb{R}^3$ is a translation vector. Furthermore, we define $\mathcal{R}(\tilde{\boldsymbol{x}}) = \mathcal{R}(\boldsymbol{x})^{-1}$, i.e. $\tilde{\boldsymbol{x}}$ are the rigid body transformation parameters that correspond to the inverse[1] of the rigid body transformation $\mathcal{R}(\boldsymbol{x})$, hence $\mathcal{R}(\tilde{\boldsymbol{x}})\boldsymbol{p} = \boldsymbol{R}(\boldsymbol{\omega})^{-1}(\boldsymbol{p} - \boldsymbol{t})$. Here, the entities without the tilde correspond to the inverse of same entities with a tilde and vice versa. See Appendix C for the details. We define the *transformed nominal model* by $\mathcal{M}_{\mathcal{R}}(\tilde{\boldsymbol{x}}) = \mathcal{R}(\tilde{\boldsymbol{x}})\mathcal{M}$.

In this setting, the independent variables $\boldsymbol{u}_j$ are unknown but can be recovered by least distance projection of the transformed points $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ on $\mathcal{M}$ or, equivalently, by least distance projection of the measurements $\boldsymbol{p}_j$ on $\mathcal{M}_{\mathcal{R}}(\tilde{\boldsymbol{x}})$. We discuss

[1] The reason for using the inverse is that we, in the formulation of the least squares problem that solves for $\boldsymbol{x}$, want to let the rigid body transform $\mathcal{R}(\boldsymbol{x})$ act on the measurements $\boldsymbol{p}_j$ instead of the model $\mathcal{M}$ in order to simplify the derivation of the derivatives we need for the numerical solution.

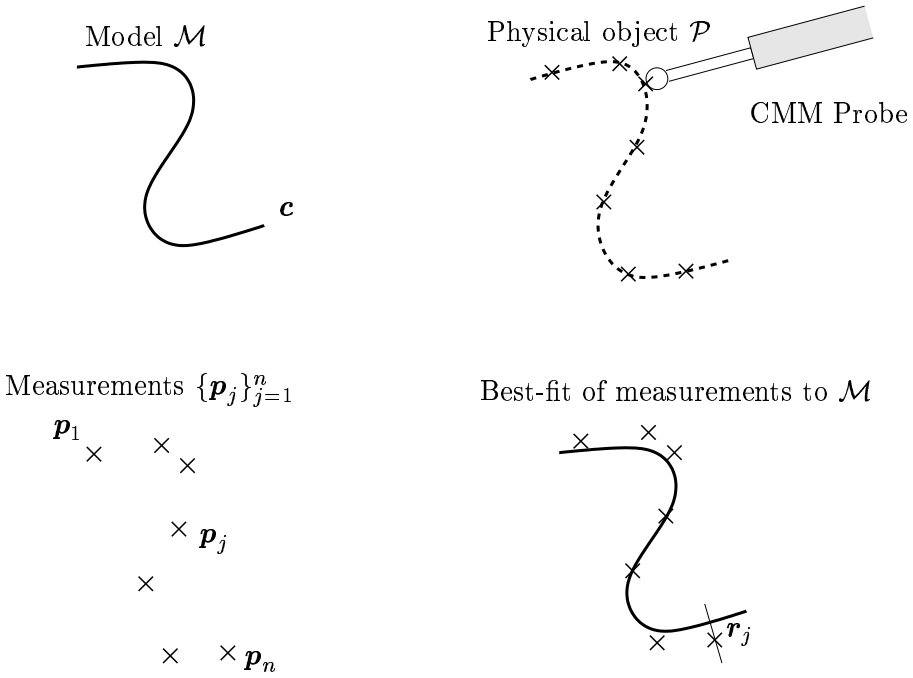

Figure 2.1: Conceptual example of a best fit of CMM measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ from a physical object $\mathcal{P}$ to its model $\mathcal{M}$.

issues related to projections in Section 2.3. In this chapter, we assume that the sum of the measurement errors $\boldsymbol{\varepsilon}_j$ and the form errors $\boldsymbol{\gamma}_j$ are independent and normally distributed with mean zero, i.e. $\boldsymbol{\varepsilon} + \boldsymbol{\gamma} \sim N_n(0, \boldsymbol{\Sigma})$, where $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_1^T \ \cdots \ \boldsymbol{\varepsilon}_n^T]^T$, $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^T \ \cdots \ \boldsymbol{\gamma}_n^T]^T$ and $\boldsymbol{\Sigma}$ is a block diagonal matrix with the 3×3 matrices $\mathrm{Cov}(\boldsymbol{\varepsilon}_j + \boldsymbol{\gamma}_j) = \sigma_j^2 \boldsymbol{I}$ along the diagonal. Here and in the sequel, all vectors are row vectors and are denoted by bold, lower-case letters. Similarly, matrices are denoted by bold, capital letters like, for example, the identity matrix $\boldsymbol{I}$ that we just introduced. Moreover, the transpose of a vector or a matrix is denoted by ^{T}, i.e. the transpose of a vector $\boldsymbol{y}$ is $\boldsymbol{y}^T$.

The assumption that the form errors $\boldsymbol{\gamma}_j$ are independent is, of course, not valid generally. On the contrary, the dependence of the form error of two discrete measurements is likely to be strong, in particular if they are close to each other. This is because the form error is in general a systematic error that is due to the connectedness of the part. Nevertheless, the least squares method we are about to formulate is often used without caring for this incorrect assumption. However, Chapter 4 introduces a geometric model where the independence assumption is dropped.

2.2.2 Stating the Problem

In order to formulate the least squares problem we denote the jth residual by $\boldsymbol{r}_j(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{c}(\boldsymbol{u}) - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ and its squared norm by $\varphi_j(\boldsymbol{x},\boldsymbol{u}) = w_j|\boldsymbol{r}_j(\boldsymbol{x},\boldsymbol{u})|^2/2$, where $w_j = \sigma_j^{-2}$. Then we can define the target function of the rigid best fit problem as $f(\boldsymbol{x},\bar{\boldsymbol{u}}) = \sum_{j=1}^n \varphi_j(\boldsymbol{x},\boldsymbol{u}_j)$, where $\bar{\boldsymbol{u}} = [\boldsymbol{u}_1 \ \cdots \ \boldsymbol{u}_n]^T$. The least squares problem then reads as follows.

Let $\{\boldsymbol{p}_j\}_{j=1}^n$ be a set of discrete coordinate measurements of the workpiece $\mathcal{P}$ manufactured from the nominal model $\mathcal{M}$. We are interested in finding the parameters $\boldsymbol{x}$ of the rigid body transform that best fits the measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ to the model $\mathcal{M}$. The rigid body transformation parameters are given by the solution to the optimisation problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^6} f\big(\boldsymbol{x},\bar{\boldsymbol{u}}(\boldsymbol{x})\big) \tag{2.1}$$

where

$$\boldsymbol{u}_j(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}\in D} \varphi_j(\boldsymbol{x},\boldsymbol{u}), \quad j = 1,\ldots,n. \tag{2.2}$$

The least distance problem (2.2) can be interpreted as requiring the measurement target point $\boldsymbol{c}(\boldsymbol{u}_j(\boldsymbol{x}))$ to be the point on the nominal model closest to the transformed measurement $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$. If the target points are fixed, i.e. if the points $\boldsymbol{c}(\boldsymbol{u}_j)$ that correspond to the transformed measurements $\boldsymbol{p}_j$ are known in advance for all $j = 1,\ldots,n$, then the problem (2.1) can be solved with the singular value decomposition (SVD) as suggested by Hanson and Norris (1981). This is because the rigid body transformation is nothing but a special type of affine transform, which can be recovered by the SVD. Anyway, in the current situation we cannot neglect the presence of the least distance problem (2.2), since the target points are assumed unknown. Because of the geometric interpretation of (2.1)–(2.2), problems of this type are often denoted *orthogonal least squares* in numerical analysis literature and *orthogonal distance regression* in the statistics community.

Ideally, we would like to use Newton's method for the solution of (2.1)–(2.2) since Newton's method is known to converge at a quadratic rate. However, Newton's method requires knowledge of the second derivatives. In the forthcoming section, we show that the second derivatives are available and thus Newton's method is the solution method of choice.

2.2.3 Alternative Formulations

Of course, (2.1) and (2.2) can be incorporated into one optimisation problem where minimisation is carried out for both $\boldsymbol{x}$ and $\bar{\boldsymbol{u}}$ simultaneously, but such a formulation gives rise to a larger optimisation problem. In particular, the Hessian of the problem will be of order $6+n$, giving rise to a large, sparse problem that grows linearly with the number of measurements n. Indeed, the problem will be very sparse since the last n equations corresponding to the reparameterisation are completely independent from each other and hence we gain in efficiency by solving these separately.

In Section 2.4.1 we incorporate the least distance problem (2.2) into the target function (2.1). We recover the optimal rigid body transform through a nested algorithm. Each evaluation of the target function requires the computation of the least distance projection of the transformed measurements on the nominal model $\mathcal{M}$.

2.3 The Least Distance Problem

This section describes how to solve the subproblem (2.2) for fixed $\boldsymbol{x}$ and j. In other words, we describe how the closest point $\boldsymbol{c}(\boldsymbol{u}_j^*)$ in the nominal model $\mathcal{M}$ can be found given a measurement $\boldsymbol{p}_j$ and the fixed rigid body transformation parameters $\boldsymbol{x}$. The point $\boldsymbol{c}(\boldsymbol{u}_j^*)$ on the nominal model $\mathcal{M}$ is denoted the *least distance projection* of $\mathcal{R}(\boldsymbol{x})$ on the nominal model $\mathcal{M}$. The least distance projection of a transformed point $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ is given by the optimisation problem

$$\min_{\boldsymbol{u}\in D} \varphi_j(\boldsymbol{x}, \boldsymbol{u}_j), \tag{2.3}$$

that gives the parameter $\boldsymbol{u}_j^*$ of the closest point $\boldsymbol{c}(\boldsymbol{u}_j^*)$ to $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$. We assume that $\boldsymbol{u}_j^*$ belongs to the interior of D for all $j = 1, \ldots, n$.

2.3.1 Local Existence on Surfaces

A necessary condition for $\boldsymbol{u}_j^*$ to be a local minimiser of (2.3) in the interior of D is that the first order condition $\varphi_{j,\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_j^*) = w_j \boldsymbol{r}_j^T \boldsymbol{r}_{j,\boldsymbol{u}} = 0$ is satisfied. Please observe that we denote the derivative by adding "$,\boldsymbol{u}$" to the subscript, i.e. $\partial \boldsymbol{r}_j / \partial \boldsymbol{u} = \boldsymbol{r}_{j,\boldsymbol{u}}$. See Appendix A for the definitions of the derivatives. Since $\boldsymbol{r}_{j,\boldsymbol{u}} = [\boldsymbol{r}_{j,u}\ \boldsymbol{r}_{j,v}]$ spans the tangent plane at $\boldsymbol{c}(\boldsymbol{u}_j^*)$ the first order condition means that $\boldsymbol{r}_j \perp \boldsymbol{r}_{j,u}$ and $\boldsymbol{r}_j \perp \boldsymbol{r}_{j,v}$. In other words, the residual is orthogonal to the tangent plane at $\boldsymbol{c}(\boldsymbol{u}_j^*)$ and hence is collinear with the principal normal at $\boldsymbol{c}(\boldsymbol{u}_j^*)$, i.e. $\boldsymbol{r}_j \parallel \boldsymbol{n}_j$, where $\boldsymbol{n}_j = \boldsymbol{c}_u \times \boldsymbol{c}_v / |\boldsymbol{c}_u \times \boldsymbol{c}_v|$. We have left out the parameter $\boldsymbol{u}_j^*$ for simplicity of notation.

A sufficient, second order condition for $\boldsymbol{u}_j^*$ to be a local minimiser of (2.3) in the interior of D is that $\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}, \boldsymbol{u}_j^*) = w_j(\boldsymbol{r}_{j,\boldsymbol{u}}^T \boldsymbol{r}_{j,\boldsymbol{u}} + \boldsymbol{r}_j^T \boldsymbol{r}_{j,\boldsymbol{uu}})$ is positive definite. Please note that $\boldsymbol{r}_{j,\boldsymbol{uu}}$ is a three-dimensional matrix and the last product has to be interpreted using tensor multiplication. Recalling that $\boldsymbol{r}_j(\boldsymbol{x}, \boldsymbol{u}_j^*)$ is collinear with $\boldsymbol{n}_j$ then the residual can be written as $\boldsymbol{r}_j(\boldsymbol{x}, \boldsymbol{u}_j^*) = \rho_j \boldsymbol{n}_j$ where $\rho_j = \boldsymbol{r}_j^T \boldsymbol{n}_j$. Using the new notation, the second order condition is expressed more clearly as

$$\begin{aligned}\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}, \boldsymbol{u}_j^*) &= w_j \left(\begin{bmatrix} |\boldsymbol{c}_u|^2 & \boldsymbol{c}_u^T \boldsymbol{c}_v \\ \boldsymbol{c}_v^T \boldsymbol{c}_u & |\boldsymbol{c}_v|^2 \end{bmatrix} + \rho_j \begin{bmatrix} \boldsymbol{n}_j^T \boldsymbol{c}_{uu} & \boldsymbol{n}_j^T \boldsymbol{c}_{uv} \\ \boldsymbol{n}_j^T \boldsymbol{c}_{vu} & \boldsymbol{n}_j^T \boldsymbol{c}_{vv} \end{bmatrix} \right) \\ &= w_j \left(\begin{bmatrix} E & F \\ F & G \end{bmatrix} + \rho_j \begin{bmatrix} e & f \\ f & g \end{bmatrix} \right) = w_j(\boldsymbol{N} + \rho_j \boldsymbol{Q}), \end{aligned} \tag{2.4}$$

where E, F and G are the coefficients of the first fundamental form and e, f and g are the coefficients of the second fundamental form of $\mathcal{M}$ at $\boldsymbol{c}(\boldsymbol{u}_j^*)$, all of them in local coordinates, see do Carmo (1976, p. 155).

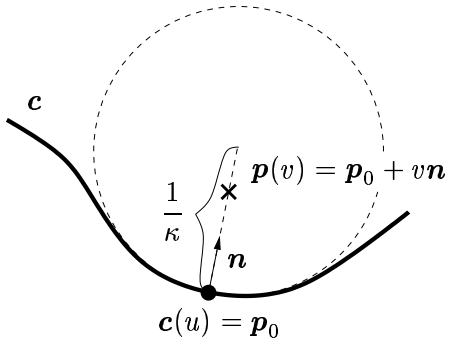

Figure 2.2: This illustrates the osculating circle at $\boldsymbol{c}(u) = \boldsymbol{p}(0)$ of the curve $\boldsymbol{c}$.

Imagine now that the least distance projection $\boldsymbol{c}(\boldsymbol{u}_j^*)$ with principal normal $\boldsymbol{n}_j$ of a point $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ is known. We are interested in the points for which the least distance projection of $\boldsymbol{c}(\boldsymbol{u}_j^*) + \rho_j \boldsymbol{n}_j$ exists locally, $\rho_j \in \mathbb{R}$.

The roots of $\det(\boldsymbol{N} + \rho_j \boldsymbol{Q}) = 0$ are $\rho_{j,i} = -1/\kappa_i$, $i = 1, 2$, where κ_1 and κ_2 are the *principal curvatures* of $\mathcal{M}$ at $\boldsymbol{c}(\boldsymbol{u}_j^*)$ (see Farin, 1997, p. 354). The roots $\rho_{j,1}$ and $\rho_{j,2}$ are known to be real and they divide the real line $\mathbb{R}$ into three intervals. If $\boldsymbol{c}(\boldsymbol{u}_j^*)$ is a *umbilical* point, i.e. if $\kappa_1 = \kappa_2$, then one interval is empty. It is obvious that $\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}, \boldsymbol{u}_j^*)$ is well-defined when $\rho_j = 0$ since $\boldsymbol{N} = \boldsymbol{r}_{\boldsymbol{u}}^T \boldsymbol{r}_{\boldsymbol{u}}$ is trivially positive definite. Thus, we conclude from continuity that $\boldsymbol{N} + \rho_j \boldsymbol{Q}$ is positive definite whenever ρ_j belongs to the interval containing the origin.

More precisely, assume without loss of generality that the roots are sorted so that $\rho_{j,1} < \rho_{j,2}$. This implies that $\kappa_1 < \kappa_2$. Form the intervals $R_0 = (-\infty, \rho_{j,1})$, $R_1 = (\rho_{j,1}, \rho_{j,2})$ and $R_2 = (\rho_{j,2}, \infty)$. Then, the least distance projection exists locally whenever ρ_j belongs to the interval containing the origin, i.e. whenever $\rho_j \in \mathcal{R}_i$ such that $0 \in \mathcal{R}_i$ for $i = 0, 1$ or 2.

2.3.2 Local Existence on Curves

Because of the great importance of least distance projections we also supply a more intuitive derivation of a local existence condition for the special case when $\mathcal{M}$ is a curve. In this case, the parameter $\boldsymbol{u}_j$ is a scalar and we write u_j instead.

A first order necessary condition for u_j^* to be a local minimiser of (2.3) in the interior of D is that $\varphi_{j,u}(\boldsymbol{x}, u_j^*) = 0$, i.e. that $\varphi_{j,u}(\boldsymbol{x}, u_j^*) = w_j \, \boldsymbol{r}_j^T \boldsymbol{r}_{j,u} = w_j \, \boldsymbol{r}_j^T \boldsymbol{c}_u = 0$. A geometric interpretation of this result is that the residual $\boldsymbol{r}_j$ is orthogonal to the curve tangent $\boldsymbol{c}_u(u_j^*)$ of the closest point $\boldsymbol{c}(u_j^*)$ to $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$. The second order sufficient condition for u_j^* to minimise (2.3) in the interior of D is that

$$\varphi_{j,uu}(\boldsymbol{x}, u_j^*) = w_j \big(|\boldsymbol{c}_u(u_j^*)|^2 + \boldsymbol{r}_j(\boldsymbol{x}, u_j^*)^T \boldsymbol{c}_{uu}(u_j^*) \big) > 0, \tag{2.5}$$

which is satisfied whenever $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ is within the centre of the osculating circle corresponding to $\boldsymbol{c}(u_j^*)$, the projection of $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$, see Figure 2.2.

In order to show this let $\boldsymbol{p}_0 = \boldsymbol{c}(u)$ for some fixed $u \in I$ and let $\boldsymbol{n}$ be the normal of $\boldsymbol{c}(u)$. The normal $\boldsymbol{n}$ can be written in terms of the derivatives of $\boldsymbol{c}$ as

$$\boldsymbol{n} = \frac{\boldsymbol{c}_{uu} - \frac{\boldsymbol{c}_u^T \boldsymbol{c}_{uu}}{|\boldsymbol{c}_u|^2}\boldsymbol{c}_u}{|\boldsymbol{c}_{uu} - \frac{\boldsymbol{c}_u^T \boldsymbol{c}_{uu}}{|\boldsymbol{c}_u|^2}\boldsymbol{c}_u|} = \frac{\boldsymbol{c}_{uu} - \frac{\boldsymbol{c}_u^T \boldsymbol{c}_{uu}}{|\boldsymbol{c}_u|^2}\boldsymbol{c}_u}{\sqrt{|\boldsymbol{c}_{uu}|^2 - \frac{(\boldsymbol{c}_u^T \boldsymbol{c}_{uu})^2}{|\boldsymbol{c}_u|^2}}}, \tag{2.6}$$

using basic differential geometry, see for instance Farin (1997). From the same reference we have that the curvature can be written as

$$\kappa^2 = \frac{|\boldsymbol{c}_u \times \boldsymbol{c}_{uu}|^2}{|\boldsymbol{c}_u|^6} = \frac{1}{|\boldsymbol{c}_u|^4}\big(|\boldsymbol{c}_{uu}|^2 - \frac{(\boldsymbol{c}_u^T \boldsymbol{c}_{uu})^2}{|\boldsymbol{c}_u|^2}\big). \tag{2.7}$$

By identifying the similarities between (2.6) and (2.7) we can simplify (2.6) to

$$\boldsymbol{n} = \frac{1}{\kappa|\boldsymbol{c}_u|^2}\big(\boldsymbol{c}_{uu} - \frac{\boldsymbol{c}_u^T \boldsymbol{c}_{uu}}{|\boldsymbol{c}_u|^2}\boldsymbol{c}_u\big). \tag{2.8}$$

Imagine now that we let a point $\boldsymbol{p}$ move along the normal $\boldsymbol{n}$ of $\boldsymbol{p}_0$ as $\boldsymbol{p}(v) = \boldsymbol{p}_0 + v\boldsymbol{n}$. We want to investigate for which values of v the projection $\boldsymbol{p}_0$ is well-defined, i.e. at what point $\boldsymbol{p}(v)$ that $\varphi_{uu}(0,u) = w(|\boldsymbol{c}_u|^2 + \boldsymbol{r}^T\boldsymbol{c}_{uu})$, where $\boldsymbol{r} = \boldsymbol{c}(u) - \boldsymbol{p}(v)$, ceases to be strictly positive. Inserting $\boldsymbol{r} = \boldsymbol{p}_0 - \boldsymbol{p}(v) = -v\boldsymbol{n}$ and (2.8) in (2.5) gives

$$\begin{aligned}\frac{1}{w}\varphi_{uu}(0,u) &= |\boldsymbol{c}_u|^2 - v\boldsymbol{c}_{uu}^T\boldsymbol{n} = |\boldsymbol{c}_u|^2 - \frac{v}{\kappa|\boldsymbol{c}_u|^2}\boldsymbol{c}_{uu}^T\big(\boldsymbol{c}_{uu} - \frac{\boldsymbol{c}_u^T\boldsymbol{c}_{uu}}{|\boldsymbol{c}_u|^2}\boldsymbol{c}_u\big) \\ &= |\boldsymbol{c}_u|^2 - \frac{v}{\kappa|\boldsymbol{c}_u|^2}\Big(|\boldsymbol{c}_{uu}|^2 - \frac{(\boldsymbol{c}_u^T\boldsymbol{c}_{uu})^2}{|\boldsymbol{c}_u|^2}\Big) = |\boldsymbol{c}_u|^2(1 - v\kappa),\end{aligned}$$

which is strictly positive for $v < \kappa^{-1}$, where κ^{-1} is the radius of curvature at $\boldsymbol{p}_0 = \boldsymbol{c}(u)$. In Figure 2.2, it is seen that the condition can be interpreted geometrically in terms of the osculating circle.

2.3.3 Global Conditions for Curves and Surfaces

The requirement that $\varphi_{\boldsymbol{uu}}$ is positive definite is a local condition for the projection to exist uniquely. For a global uniqueness condition for the projection we must consider the possibility that some other point on the model $\mathcal{M}$ apart from our original point $\boldsymbol{p}_0 = \boldsymbol{c}(u)$ can be the projection of the point $\boldsymbol{p}(v)$ even though the local existence conditions are satisfied. As always, global conditions are more difficult to deal with. However, intuitively the condition is clear. Figure 2.3 shows the regions for which a projection is well-defined for an example curve.

2.3.4 Computing the Least Distance Projection

Having argued for the existence and uniqueness of the least distance projection of a measurement on the model $\mathcal{M}$ it is time to illustrate how it can be computed.

For a set of measurement data $\{\boldsymbol{p}_j\}_{j=1}^n$ we want to find the corresponding projections in the model $\mathcal{M}$. The projection of a transformed measurement $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ is

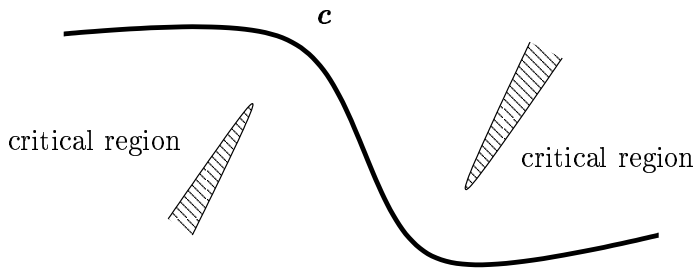

Figure 2.3: The figure illustrates the critical regions for a plane curve. Points in the critical regions do not have well-defined projections on the curve $\boldsymbol{c}$.

described by the surface/curve parameter $\boldsymbol{u}_j^*$. Formally, we can write this projection as $\boldsymbol{u}_j^* = \boldsymbol{c}^{-1} \circ \boldsymbol{P}_{\mathcal{M}}\ \mathcal{R}(\boldsymbol{x})\ \boldsymbol{p}_j$, where $\boldsymbol{P}_{\mathcal{M}}$ denotes the non-linear least distance projection taking space points to their closest point in the object $\mathcal{M}$. The parameter $\boldsymbol{u}_j^*$ is found by solving the non-linear least squares problem

$$\min_{\boldsymbol{u}\in D} \varphi_j(\boldsymbol{x}, \boldsymbol{u})$$

for fixed $\boldsymbol{x}$. It can be done iteratively by Newton's method. If $\boldsymbol{u}^{(0)}$ is an initial approximation to $\boldsymbol{u}_j^*$ then Newton's method generates a sequence

$$\boldsymbol{u}_j^{(l+1)} = \boldsymbol{u}_j^{(l)} - \varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}, \boldsymbol{u}_j^{(l)})^{-1} \varphi_{j,\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_j^{(l)}), \quad l = 0, 1, \ldots$$

that converges quadratically to the true solution $\boldsymbol{u}_j^*$. Algorithm 2.1 gives a systematic description of how the least distance projection can be computed.

Algorithm 2.1 Algorithm for the least distance projection of a point onto the nominal model $\mathcal{M}$. Formally we denote this non-linear projection procedure of a transformed space point $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ by $\boldsymbol{u}_j^* = \boldsymbol{c}^{-1} \circ \boldsymbol{P}_{\mathcal{M}}\, \mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$.

Input: Parameterised description $\boldsymbol{c}$ of nominal model $\mathcal{M}$, rigid body transformation parameter $\boldsymbol{x}$ and measurement to project $\boldsymbol{p}_j$.
Initialise $\boldsymbol{u}_j^{(0)}$ and relErrTol
relErr $\Leftarrow$ relErrTol + 1
$l \Leftarrow 0$
while not $0 <$ relErr $<$ relErrTol **do**
$\boldsymbol{u}_j^{(l+1)} \Leftarrow \boldsymbol{u}_j^{(l)} - \varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}, \boldsymbol{u}_j^{(l)})^{-1} \varphi_{j,\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_j^{(l)})$
relErr $\Leftarrow \left(\varphi_j(\boldsymbol{x}, \boldsymbol{u}_j^{(l)}) - \varphi_j(\boldsymbol{x}, \boldsymbol{u}_j^{(l+1)})\right) / \left(|\varphi_j(\boldsymbol{x}, \boldsymbol{u}_j^{(l+1)})| + \text{relErr}\right)$
$l \Leftarrow l + 1$
end while
$\boldsymbol{u}_j^* \Leftarrow \boldsymbol{u}_j^{(l)}$
Output: Parameter of least distance projection $\boldsymbol{u}_j^*$.

2.4 The Derivatives of the Least Squares Problem

In order to solve the least squares problem (2.1)–(2.2) it is convenient to have access to the derivatives of the problem. It turns out that the least distance problem (2.2) can be incorporated into the target function (2.1) and analytic expressions for both the first and second derivatives are available.

2.4.1 Including the Least Distance Problem

In this section, we use the implicit function theorem to incorporate the least distance problem (2.2) in the target function (2.1). For information about the implicit function theorem we refer to Rudin (1976, p. 223). A necessary condition for (2.2) to be minimised in the interior of D for fixed $\boldsymbol{x} = \boldsymbol{x}^*$ is that its derivative vanish at the solution $\boldsymbol{u}_j^*$ for all $j = 1, \ldots, n$. Hence, it must be that $\varphi_{j,\boldsymbol{u}}(\boldsymbol{x}^*, \boldsymbol{u}_j^*) = 0$ for all $j = 1, \ldots, n$. By the implicit function theorem there exists, if $\det(\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}^*, \boldsymbol{u}_j^*)) \neq 0$, a continuously differentiable function $\boldsymbol{u}_j = \boldsymbol{u}_j(\boldsymbol{x})$ such that $\varphi_{j,\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_j(\boldsymbol{x})) \equiv 0$ in a neighbourhood of $\boldsymbol{x}^*$. The condition $\det(\varphi_{j,\boldsymbol{uu}}) \neq 0$ is met if φ_j satisfies the second order sufficient condition for $\boldsymbol{u}_j^*$ to be a minimiser of (2.2), namely that (2.4) is positive definite.

Because of the above, the derivative of $\varphi_{j,\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_j(\boldsymbol{x}))$ with respect to $\boldsymbol{x}$ is also zero, i.e.

$$\frac{\partial \varphi_{j,\boldsymbol{u}}\big(\boldsymbol{x}, \boldsymbol{u}_j(\boldsymbol{x})\big)^T}{\partial \boldsymbol{x}} = \varphi_{j,\boldsymbol{xu}} + \varphi_{j,\boldsymbol{uu}} \boldsymbol{u}_{j,\boldsymbol{x}} = 0$$

by the chain rule. See Appendix A for the definitions of the derivative operators.

From this we see that $\boldsymbol{u}_{j,\boldsymbol{x}} = -\varphi_{j,\boldsymbol{uu}}^{-1} \varphi_{j,\boldsymbol{xu}}$. This is well-defined if our assumption that $\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}, \boldsymbol{u}_j)$ is invertible holds. The discussion in Section 2.3.1 describes conditions for $\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}^*, \boldsymbol{u}_j^*)$ to be positive definite which implies that it is invertible. By continuity it is invertible in a neighbourhood of thereof as well.

Having established the existence of $\boldsymbol{u}_j = \boldsymbol{u}_j(\boldsymbol{x})$ that solves (2.2) we can define $\bar{\boldsymbol{u}} = [\boldsymbol{u}_1 \ \cdots \ \boldsymbol{u}_n]^T = \bar{\boldsymbol{u}}(\boldsymbol{x})$ and restrict the target function f to $\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{x}, \bar{\boldsymbol{u}}(\boldsymbol{x}))$. In this way, we can incorporate the least distance problem (2.2) in the target function (2.1). Hence, the least squares problem (2.1)–(2.2) simplifies to

$$\min_{\boldsymbol{x} \in \mathbb{R}^6} \tilde{f}(\boldsymbol{x}). \tag{2.9}$$

Please note that the implicit function theorem only guarantees the existence of the parameter function $\bar{\boldsymbol{u}} = \bar{\boldsymbol{u}}(\boldsymbol{x})$. We do not have it in explicit form.

In order to find the solution $\boldsymbol{x}^*$ of (2.9) we can use Newton's method, which requires the first and second derivatives of the objective function $\tilde{f}$, i.e. the gradient $\boldsymbol{g}$ and the Hessian $\boldsymbol{H}$.

2.4.2 The Gradient

In this section, we calculate the analytic gradient $\boldsymbol{g}$ of $\tilde{f}$.

$$\boldsymbol{g}(\boldsymbol{x})^T = \frac{\partial \tilde{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial f\big(\boldsymbol{x}, \bar{\boldsymbol{u}}(\boldsymbol{x})\big)}{\partial \boldsymbol{x}} = f_{\boldsymbol{x}} + f_{\bar{\boldsymbol{u}}}\bar{\boldsymbol{u}}_{\boldsymbol{x}}$$
$$= \sum_{j=1}^{n} \varphi_{j,\boldsymbol{x}} + \varphi_{j,\boldsymbol{u}}\boldsymbol{u}_{j,\boldsymbol{x}} = \sum_{j=1}^{n} \varphi_{j,\boldsymbol{x}} - \varphi_{j,\boldsymbol{u}}\varphi_{j,\boldsymbol{uu}}^{-1}\varphi_{j,\boldsymbol{xu}}.$$

The transpose on the gradient is due to our definition of multivariate derivative (see Appendix A). The multivariate derivative of a scalar is defined to be a row vector whereas the gradient is supposed to be a column vector. The first order local condition for $\boldsymbol{u}_j(\boldsymbol{x})$ to solve (2.2) in the interior of D says that $\varphi_{j,\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}_j(\boldsymbol{x})) = 0$. Hence, the gradient reduces to

$$\boldsymbol{g}(\boldsymbol{x}) = \sum_{j=1}^{n} \varphi_{j,\boldsymbol{x}}\big(\boldsymbol{x}, \boldsymbol{u}_j(\boldsymbol{x})\big)^T = f_{\boldsymbol{x}}\big(\boldsymbol{x}, \bar{\boldsymbol{u}}(\boldsymbol{x})\big)^T.$$

So what is $f_{\boldsymbol{x}}$? Taking a closer look at the jth term $\varphi_{j,\boldsymbol{x}}$ of the sum $f_{\boldsymbol{x}}$ we omit the index j to clarify the notation, keeping in mind that we are working with the jth term of the sum that constitute the gradient. Thus, $\varphi_{\boldsymbol{x}} = w\,\boldsymbol{r}^T\boldsymbol{r}_{\boldsymbol{x}}$, where $\boldsymbol{r}_{\boldsymbol{x}} = -\frac{\partial}{\partial \boldsymbol{x}}(\mathcal{R}(\boldsymbol{x})\boldsymbol{p})$. From (C.2) in Appendix C we have that $\frac{\partial}{\partial \boldsymbol{x}}(\mathcal{R}(\boldsymbol{x})\boldsymbol{p}) = [\boldsymbol{I} - [\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}]]$ so the full gradient consequently sums up to

$$\boldsymbol{g}(\boldsymbol{x}) = \sum_{j=1}^{n} w_j\,\boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{r}_j, \tag{2.10}$$

where we have left out the parameters $(\boldsymbol{x}, \boldsymbol{u}_j(\boldsymbol{x}))$ to simplify the notation. Recall that $\boldsymbol{r}_{j,\boldsymbol{x}} = -[\boldsymbol{I} - [\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}_j]]$. Here we have introduced the *cross product matrix* of a vector $\boldsymbol{a} \in \mathbb{R}^3$ that is denoted by $[\boldsymbol{a}]$, see Appendix C for its definition.

2.4.3 The Hessian

Continuing with the Hessian $\boldsymbol{H}$ we have that

$$\boldsymbol{H}(\boldsymbol{x}) = \frac{\partial^2 \tilde{f}(\boldsymbol{x})}{\partial \boldsymbol{x}^2} = \frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial f_{\boldsymbol{x}}\big(\boldsymbol{x}, \bar{\boldsymbol{u}}(\boldsymbol{x})\big)^T}{\partial \boldsymbol{x}} = f_{\boldsymbol{xx}} + f_{\bar{\boldsymbol{u}}\boldsymbol{x}}\bar{\boldsymbol{u}}_{\boldsymbol{x}}$$
$$= \sum_{j=1}^{n} \varphi_{j,\boldsymbol{xx}} + \varphi_{j,\boldsymbol{ux}}\boldsymbol{u}_{j,\boldsymbol{x}} = \sum_{j=1}^{n} \varphi_{j,\boldsymbol{xx}} - \varphi_{j,\boldsymbol{ux}}\varphi_{j,\boldsymbol{uu}}^{-1}\varphi_{j,\boldsymbol{xu}}. \tag{2.11}$$

Ignoring the index j for a moment, we compute

$$\varphi_{\boldsymbol{xx}} = w(\boldsymbol{r}_{\boldsymbol{x}}^T\boldsymbol{r}_{\boldsymbol{x}} + \boldsymbol{r}^T\boldsymbol{r}_{\boldsymbol{xx}}), \tag{2.12}$$

where the last term is not well-defined since it is a product of a three-dimensional matrix and a vector. The dealing with three-dimensional matrices usually means

that we have to switch to tensor notation. In this case, however, the difficulty can be overcome by a little trick. First, we change variables from $\boldsymbol{x}$ to $\hat{\boldsymbol{x}}$ in the factor $\boldsymbol{r}_{\boldsymbol{xx}}$ that is affected by the derivative. Second, we move the derivative operator out. Finally, we apply a result from Appendix C. In detail,

$$\begin{aligned}\boldsymbol{r}^T\boldsymbol{r}_{\boldsymbol{xx}} = \frac{\partial^2}{\partial\hat{\boldsymbol{x}}^2}\big(\boldsymbol{r}(\boldsymbol{x},\boldsymbol{u})^T\boldsymbol{r}(\hat{\boldsymbol{x}},\boldsymbol{u})\big) &= \frac{\partial^2}{\partial\hat{\boldsymbol{x}}^2}\Big(\boldsymbol{r}^T\big(\boldsymbol{c}(\boldsymbol{u}) - \mathcal{R}(\hat{\boldsymbol{x}})\boldsymbol{p}\big)\Big) \\ &= -\frac{\partial^2}{\partial\hat{\boldsymbol{x}}^2}\big(\boldsymbol{r}^T\mathcal{R}(\hat{\boldsymbol{x}})\boldsymbol{p}\big) = -\begin{bmatrix}0 & 0\\ 0 & [\boldsymbol{r}][\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}] + [\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}][\boldsymbol{r}]\end{bmatrix},\end{aligned}$$

where we have used equation (C.3).

In order to calculate the second term of the Hessian $\varphi_{\boldsymbol{ux}}\varphi_{\boldsymbol{uu}}^{-1}\varphi_{\boldsymbol{xu}}$ we take a closer look at the mixed derivative $\varphi_{\boldsymbol{ux}}$.

$$\varphi_{\boldsymbol{ux}}\big(\boldsymbol{x},\boldsymbol{u}(\boldsymbol{x})\big) = \frac{\partial\varphi_{\boldsymbol{x}}^T}{\partial\boldsymbol{u}} = \frac{\partial(w\,\boldsymbol{r}_{\boldsymbol{x}}^T\boldsymbol{r})}{\partial\boldsymbol{u}} = w(\boldsymbol{r}_{\boldsymbol{ux}}^T\boldsymbol{r} + \boldsymbol{r}_{\boldsymbol{x}}^T\boldsymbol{r}_{\boldsymbol{u}}) = w\,\boldsymbol{r}_{\boldsymbol{x}}^T\boldsymbol{c}_{\boldsymbol{u}}, \tag{2.13}$$

where $\boldsymbol{r}_{\boldsymbol{xu}} = 0$, due to the absence of mixed terms in φ. This is where we benefit from letting the rigid body transformation $\mathcal{R}(\boldsymbol{x})$ act on the measurement $\boldsymbol{p}$ instead of the model $\boldsymbol{c}$ in the definition of the residual $\boldsymbol{r}$, see Section 2.2.2.

All the terms of the Hessian are now calculated, all there is to do is to put the results of (2.12) and (2.13) in (2.11) to get

$$\boldsymbol{H}(\boldsymbol{x}) = \sum_{j=1}^{n} w_j(\boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{r}_{j,\boldsymbol{x}} + \boldsymbol{r}_j^T\boldsymbol{r}_{j,\boldsymbol{xx}} - \boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{c}_{\boldsymbol{u}}\varphi_{j,\boldsymbol{uu}}^{-1}\boldsymbol{c}_{\boldsymbol{u}}^T\boldsymbol{r}_{j,\boldsymbol{x}}) \tag{2.14}$$

$$= \sum_{j=1}^{n} w_j\big(\boldsymbol{r}_{j,\boldsymbol{x}}^T(\boldsymbol{I} - \boldsymbol{c}_{\boldsymbol{u}}^T\varphi_{j,\boldsymbol{uu}}^{-1}\boldsymbol{c}_{\boldsymbol{u}})\boldsymbol{r}_{j,\boldsymbol{x}} + \boldsymbol{r}_j^T\boldsymbol{r}_{j,\boldsymbol{xx}}\big) = \sum_{j=1}^{n} w_j(\boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{T}_j\boldsymbol{r}_{j,\boldsymbol{x}} + \boldsymbol{r}_j^T\boldsymbol{r}_{j,\boldsymbol{xx}}),$$

where $\boldsymbol{T}_j(\boldsymbol{x}) = \boldsymbol{I} - \boldsymbol{c}_{\boldsymbol{u}}(\boldsymbol{u}_j(\boldsymbol{x}))^T\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x},\boldsymbol{u}_j(\boldsymbol{x}))^{-1}\boldsymbol{c}_{\boldsymbol{u}}(\boldsymbol{u}_j(\boldsymbol{x}))$. To summarise we have derived an explicit expression for the second order Taylor expansion of the objective function $\tilde{f}$ around $\boldsymbol{x}$. We have

$$\tilde{f}(\boldsymbol{x}+\Delta\boldsymbol{x}) \approx q^{(\boldsymbol{x})}(\Delta\boldsymbol{x}) = \tilde{f}(\boldsymbol{x}) + \Delta\boldsymbol{x}^T\boldsymbol{g}(\boldsymbol{x}) + \frac{1}{2}\Delta\boldsymbol{x}^T\boldsymbol{H}(\boldsymbol{x})\Delta\boldsymbol{x}, \tag{2.15}$$

with $\boldsymbol{g}$ as in (2.10) and $\boldsymbol{H}$ as in (2.14).

2.5 Sensitivity Analysis

An important question that often arises in optimisation is how much the optimum changes when the objective function changes. In particular we are interested in how much the optimal solution to the rigid best fit problem changes when the measurements are perturbed.

Assume we are given a data set $\{\boldsymbol{p}_j\}_{j=1}^n$ of a workpiece $\mathcal{P}$ manufactured from the nominal model $\mathcal{M}$. The target function of the rigid best fit problem is given in (2.9) and its optimal solution is denoted by $\boldsymbol{x}^*$. We are interested in how much

the solution changes when the measurements $\boldsymbol{p}_j$ are perturbed to $\tilde{\boldsymbol{p}}_j = \boldsymbol{p}_j + \varepsilon_j \boldsymbol{s}_j$, $|\boldsymbol{s}_j| = 1, j = 1, \ldots, n$, i.e. we want to know the solution to the following, perturbed problem.

Given a set of measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ a set of perturbation directions $\{\boldsymbol{s}_j\}_{j=1}^n$ and a vector of perturbation scalars $\boldsymbol{\varepsilon} = [\varepsilon_1 \ \cdots \ \varepsilon_n]^T$, find the rigid body transformation parameters $\boldsymbol{x}$ that minimises

$$\min_{\boldsymbol{x} \in \mathbb{R}^6} \tilde{f}(\boldsymbol{x}, \boldsymbol{\varepsilon}).$$

The target function $\tilde{f}$ is defined as in Section 2.2.2 with the difference that we have here added the perturbation parameter $\boldsymbol{\varepsilon}$. A first order local condition for $\boldsymbol{x}$ to be a minimiser of the above is that the gradient $\boldsymbol{g}$ vanishes at the solution, i.e. $\boldsymbol{g}(\boldsymbol{x}^*, \boldsymbol{\varepsilon}) = 0$. According to the implicit function theorem a continuously differentiable function $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\varepsilon})$ that depends on the perturbation parameter $\boldsymbol{\varepsilon}$ exists if $\det(\boldsymbol{g}_{\boldsymbol{x}}(\boldsymbol{x}^*, 0)) \neq 0$, where $\boldsymbol{x}^*$ is the solution to the unperturbed problem when $\boldsymbol{\varepsilon} = 0$. Please note that $\boldsymbol{g}_{\boldsymbol{x}}$ is the Hessian to the problem derived in Section 2.4.3 and denoted by $\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{\varepsilon})$ where we have added an argument for the perturbation $\boldsymbol{\varepsilon}$. The determinant of the Hessian is, of course, non-zero since we assume the solution $\boldsymbol{x}^*$ to the unperturbed problem to exist.

In a neighbourhood of $(\boldsymbol{x}^*, 0)$ for which $\boldsymbol{g}(\boldsymbol{x}^*, 0) = 0$ it must be that $\boldsymbol{g}(\boldsymbol{x}(\boldsymbol{\varepsilon}), \boldsymbol{\varepsilon}) = 0$ and hence

$$\frac{\partial}{\partial \boldsymbol{\varepsilon}} \big(\boldsymbol{g}(\boldsymbol{x}(\boldsymbol{\varepsilon}), \boldsymbol{\varepsilon})\big) = \boldsymbol{H}\boldsymbol{x}_{\boldsymbol{\varepsilon}} + \boldsymbol{g}_{\boldsymbol{\varepsilon}} = 0 \quad \Rightarrow \boldsymbol{x}_{\boldsymbol{\varepsilon}} = -\boldsymbol{H}^{-1}\boldsymbol{g}_{\boldsymbol{\varepsilon}}.$$

For the special case $\boldsymbol{\varepsilon} = 0$ we have $\boldsymbol{x}_{\boldsymbol{\varepsilon}}(0) = -\boldsymbol{H}(\boldsymbol{x}^*, 0)^{-1}\boldsymbol{g}_{\boldsymbol{\varepsilon}}(\boldsymbol{x}^*, 0)$. The change of the optimal solution due to the measurement perturbations is thus given to first order by $\boldsymbol{x}(\boldsymbol{\varepsilon}) - \boldsymbol{x}^* = \boldsymbol{x}_{\boldsymbol{\varepsilon}}(0)\,\boldsymbol{\varepsilon} + o(|\boldsymbol{\varepsilon}|\mathbb{1}) = -\boldsymbol{H}^{-1}\boldsymbol{g}_{\boldsymbol{\varepsilon}}\boldsymbol{\varepsilon}$.

In computing $\boldsymbol{g}_{\boldsymbol{\varepsilon}}$ we recall from (2.10) that $\boldsymbol{g}(\boldsymbol{x}) = \sum_{j=1}^n w_j \boldsymbol{r}_{j,\boldsymbol{x}}^T \boldsymbol{r}_j$ from which we see that

$$\boldsymbol{g}_{\boldsymbol{\varepsilon}} = \begin{bmatrix} \boldsymbol{r}_{1,\boldsymbol{x}}^T \boldsymbol{r}_{1,\varepsilon_1} + \boldsymbol{r}_{1,\varepsilon_1\boldsymbol{x}}^T \boldsymbol{r}_1 & \cdots & \boldsymbol{r}_{n,\boldsymbol{x}}^T \boldsymbol{r}_{n,\varepsilon_n} + \boldsymbol{r}_{n,\varepsilon_n\boldsymbol{x}}^T \boldsymbol{r}_n \end{bmatrix}$$

where the jth perturbed residual is defined as $\boldsymbol{r}_j(\boldsymbol{x}, \boldsymbol{u}_j, \varepsilon_j) = \boldsymbol{c}(\boldsymbol{u}_j) - \mathcal{R}(\boldsymbol{x})(\boldsymbol{p}_j + \varepsilon_j \boldsymbol{s}_j)$. The derivative of the unindexed residual with respect to the perturbation parameter ε is easily seen to be

$$\boldsymbol{r}_{\varepsilon} = \frac{\partial}{\partial \varepsilon}\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{u}, \varepsilon) = \frac{\partial}{\partial \varepsilon}\Big(\boldsymbol{c}(\boldsymbol{u}_j) - \big(\boldsymbol{t} + \boldsymbol{R}(\boldsymbol{\omega})(\boldsymbol{p} + \varepsilon \boldsymbol{s})\big)\Big) = -\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}.$$

It is also known that $\boldsymbol{r}_{\boldsymbol{x}} = -[\boldsymbol{I} \ -[\boldsymbol{R}(\boldsymbol{\omega})(\boldsymbol{p} + \varepsilon \boldsymbol{s})]]$ from which we see that

$$\boldsymbol{r}_{\boldsymbol{x}}^T \boldsymbol{r}_{\varepsilon} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s} \\ \big(\boldsymbol{R}(\boldsymbol{\omega})(\boldsymbol{p} + \varepsilon \boldsymbol{s})\big) \times \big(\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}\big) \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s} \\ \boldsymbol{R}(\boldsymbol{\omega})(\boldsymbol{p} \times \boldsymbol{s}) \end{bmatrix} + o\big((|\boldsymbol{\varepsilon}| + |\boldsymbol{x}|)\mathbb{1}\big). \tag{2.16}$$

We see also that $\boldsymbol{r}_{\varepsilon\boldsymbol{x}} = \begin{bmatrix} 0 & [\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}] \end{bmatrix}$ that gives

$$\boldsymbol{r}_{\varepsilon\boldsymbol{x}}^T \boldsymbol{r} = \begin{bmatrix} 0 \\ \boldsymbol{r} \times \big(\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}\big) \end{bmatrix}.$$

Together with (2.16) the derivative $\boldsymbol{g}_{\boldsymbol{\varepsilon}}$ is seen to be (to first order)

$$\boldsymbol{g}_{\boldsymbol{\varepsilon}} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}_1 & \cdots & \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}_n \\ \boldsymbol{R}(\boldsymbol{\omega})(\boldsymbol{p}_1 \times \boldsymbol{s}_1) + \boldsymbol{r}_1 \times (\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}_1) & \cdots & \boldsymbol{R}(\boldsymbol{\omega})(\boldsymbol{p}_n \times \boldsymbol{s}_n) + \boldsymbol{r}_n \times (\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{s}_n) \end{bmatrix} \boldsymbol{W}, \tag{2.17}$$

where $\boldsymbol{W}$ is a diagonal matrix with the weights along the diagonal, i.e. $\boldsymbol{W}$ is defined as $\boldsymbol{W} = \mathrm{diag}([w_1 \ \cdots \ w_n])$. Assume now that the perturbations $\boldsymbol{\varepsilon}$ are independent and normally distributed with mean zero, i.e. that $\boldsymbol{\varepsilon} \sim N_n(0, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$, where $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} = \boldsymbol{W}^{-1}$. Furthermore, assume that the measurements are perturbed in the normal direction of the measurement projection, i.e. let $\boldsymbol{s}_j = \boldsymbol{n}_j$. Finally we assume, for simplicity, that the solution to the unperturbed problem is zero, i.e. $\boldsymbol{x}^* = 0$. In this case, equation (2.17) becomes $\boldsymbol{g}_{\boldsymbol{\varepsilon}}(0, \boldsymbol{\varepsilon}) = \boldsymbol{A}\boldsymbol{W}$, where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{n}_1 & \cdots & \boldsymbol{n}_n \\ \boldsymbol{p}_1 \times \boldsymbol{n}_1 & \cdots & \boldsymbol{p}_n \times \boldsymbol{n}_n \end{bmatrix} \tag{2.18}$$

denotes the *sensitivity matrix*. Please note that $\boldsymbol{r}_j \times \boldsymbol{n}_j = 0$ since $\boldsymbol{r}_j$ is collinear with the normal direction.

Neglecting higher order terms we assume that the rigid body transformation parameters are approximately given by

$$\boldsymbol{x}(\boldsymbol{\varepsilon}) = -\boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{W}\boldsymbol{\varepsilon}. \tag{2.19}$$

Given that the perturbation parameters are distributed as assumed above, the mean is zero, i.e.

$$\boldsymbol{\mu}_{\boldsymbol{x}(\boldsymbol{\varepsilon})} = \mathbb{E}[\boldsymbol{x}(\boldsymbol{\varepsilon})] = \mathbb{E}[-\boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{W}\boldsymbol{\varepsilon}] = -\boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{W}\mathbb{E}[\boldsymbol{\varepsilon}] = 0.$$

More interestingly, the covariance can be computed explicitly by

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{x}(\boldsymbol{\varepsilon})} &= \mathbb{E}\big[(\boldsymbol{x}(\boldsymbol{\varepsilon}) - \boldsymbol{\mu}_{\boldsymbol{x}(\boldsymbol{\varepsilon})})(\boldsymbol{x}(\boldsymbol{\varepsilon}) - \boldsymbol{\mu}_{\boldsymbol{x}(\boldsymbol{\varepsilon})})^T\big] = \mathbb{E}\big[\boldsymbol{x}(\boldsymbol{\varepsilon})\boldsymbol{x}(\boldsymbol{\varepsilon})^T\big] \\ &= \mathbb{E}\big[\boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{W}\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\boldsymbol{W}\boldsymbol{A}^T\boldsymbol{H}^{-1}\big] = \boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{W}\,\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]\boldsymbol{W}\boldsymbol{A}^T\boldsymbol{H}^{-1} \\ &= \boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{W}\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}\boldsymbol{W}\boldsymbol{A}^T\boldsymbol{H}^{-1} = \boldsymbol{H}^{-1}\boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^T\boldsymbol{H}^{-1}. \end{aligned}$$

The variance of the estimated rigid body transformation parameters reveals information about the precision of the estimates. The sensitivity analysis is included because of its importance but is not used in the rest of this chapter.

2.6 Solving the Problem

The solution of the optimisation problem (2.9) can be found by starting with an initial iterate $\boldsymbol{x}^{(0)}$ and then solving a sequence of problems

$$\min_{\Delta\boldsymbol{x}} q^{(k)}(\Delta\boldsymbol{x}), \quad k = 0, 1, \ldots \tag{2.20}$$

where $q^{(k)} = q^{(\boldsymbol{x}^{(k)})}$ as defined in (2.15). The old solution $\boldsymbol{x}^{(k)}$ is updated with the step given by the solution $\Delta\boldsymbol{x}^{(k)}$ to (2.20) at the kth step according to $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}^{(k)}$. The sequence $\{\boldsymbol{x}^{(k)}\}_{k\geq 0}$ converges quadratically to the solution $\boldsymbol{x}^*$ as long as the Hessian $\boldsymbol{H}$ is positive definite. For more on numerical solution methods see e.g. Fletcher (1987).

2.6.1 Algorithm

Algorithm 2.2 describes in pseudo-code how an implementation of a solution method for the rigid best fit problem could be carried out.

Algorithm 2.2 The algorithm for finding the best rigid fit of a model to measurement data.

Input: Parameterised description $\boldsymbol{c}$ of nominal model $\mathcal{M}$, coordinate measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ and corresponding weights $\{w_j\}_{j=1}^n$.

Initialise relErrTol

relErr $\Leftarrow$ relErrTol $+ 1$

$\boldsymbol{x}^{(0)} \Leftarrow 0$

$k \Leftarrow 0$

while not $0 <$ relErr $<$ relErrTol **do**

$\boldsymbol{u}_j^{(k)} \Leftarrow \boldsymbol{c}^{-1} \circ \boldsymbol{P}_{\mathcal{M}}\, \mathcal{R}(\boldsymbol{x}^{(k)})\boldsymbol{p}_j, \quad j = 1, \ldots, n$ /* *see Algorithm 2.1* */

$\Delta\boldsymbol{x}^{(k)} \Leftarrow \arg\min_{\Delta\boldsymbol{x}} q^{(k)}(\Delta\boldsymbol{x})$ /* *Newton's method: Solve* $\boldsymbol{H}(\boldsymbol{x}^{(k)})\Delta\boldsymbol{x}^{(k)} = -\boldsymbol{g}(\boldsymbol{x}^{(k)})$ */

$\boldsymbol{x}^{(k+1)} \Leftarrow \boldsymbol{x}^{(k+1)} + \Delta\boldsymbol{x}^{(k)}$

relErr $\Leftarrow (\tilde{f}(\boldsymbol{x}^{(k)}) - \tilde{f}(\boldsymbol{x}^{(k+1)}))/\big(|\tilde{f}(\boldsymbol{x}^{(k+1)})| + \text{relErrTol}\big)$

$k \Leftarrow k + 1$

end while

$\boldsymbol{x}^* \Leftarrow \boldsymbol{x}^{(k)}$

Output: Rigid body transformation parameters $\boldsymbol{x}^*$.

The quadratic convergence of the algorithm is confirmed by numerical experiments using Newton's method. Please observe that the algorithm is nested with the least distance projection of the measurements as the inner loop. The least distance problem is described in Section 2.3 and a scheme for computing it is given in Algorithm 2.1.

2.7 Examples of Rigid Best Fit

In this section, we give some graphical examples of the use of the least squares rigid best fit algorithm.

2.7.1 Ideal Fitting

As nominal model we have chosen the surface shown in Figure 2.4. The surface is given a datum system and some inspection points. Recall that the concept of a datum system is discussed in Section 1.2 and in Section 3.1.

Figure 2.5 illustrates what happens if we simulate measurements from a dislocated copy of the surface and use Algorithm 2.2 to recover the rigid body transform hidden in the data. The leftmost picture shows the surface in nominal position. In order to distinguish the nominal position of the surface from the dislocated one we chosen to illustrate the surface in nominal position as transparent as opposed to the dislocated surface that is non-transparent.

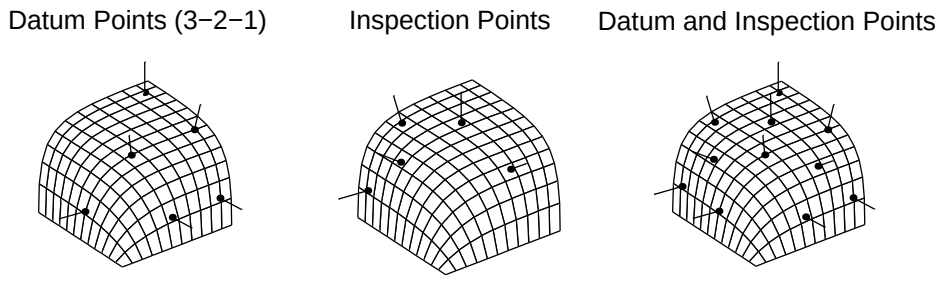

Figure 2.4: The model surface used to illustrate the rigid best fit. The leftmost picture shows the surface and the chosen datum system. In the middle the inspections points are visible. The union of the two point sets is shown to the right.

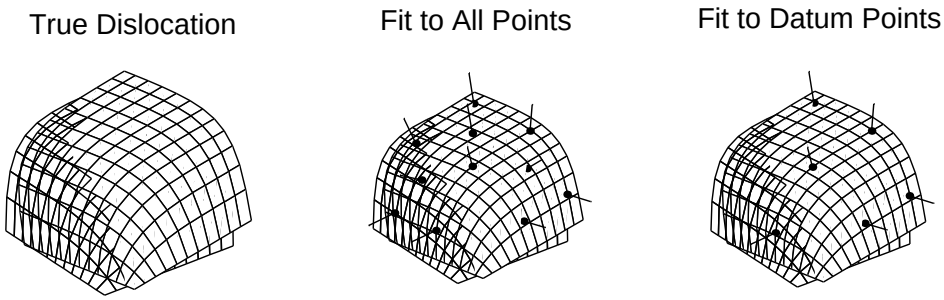

Figure 2.5: The leftmost figure shows the nominal surface as being transparent and the dislocated surface as being non-transparent. The middle picture shows the estimated dislocation from a rigid best fit to all measurements. The rightmost figure shows the rigid best fit to the datum points.

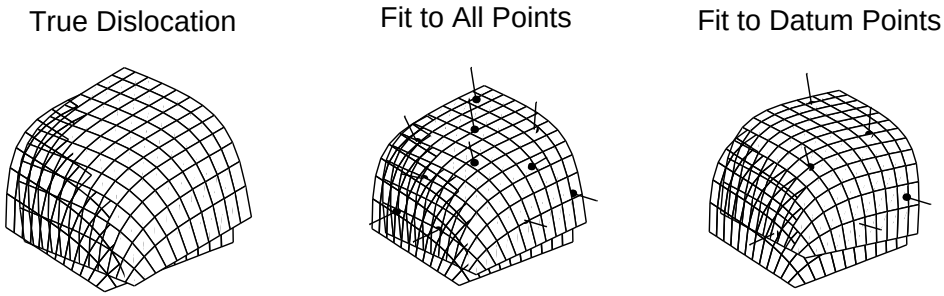

Figure 2.6: The same kind of figure as Figure 2.5 with the difference that we have added normally distributed noise to the simulated measurements.

The middle picture shows the result when we use both the "measurements" from the datum points as well as the inspection points to fit the surface. As seen, the estimated dislocation is close to the true one. This procedure, when all measurements are used for the estimation is discussed in Section 1.3. The method is also represented by leaf (iii) in Figure 1.1.

The rightmost picture of Figure 2.5 shows the result of a rigid best fit using the datum system only. This is maybe the most useful method in practice since the geometric specifications of a workpiece are usually defined relative to the datum why we want to know the deviation in the inspection points relative to the datum. There is a paragraph in Section 1.3 about this method as well which is represented in Figure 1.1 by leaf (iv).

As seen from Figure 2.5, the estimate when all points are used is the same as the estimate using only datum points. This is expected and due to the absence of noise that allows a perfect fit.

2.7.2 Fitting in Presence of Noise

In practice, workpieces are not ideal and measurements are polluted by noise. To this end we have made another example similar to the one presented in Section 2.7.1 but here we have also added normally distributed noise to the surface samples in addition to the rigid body movement. The measurements are perturbed in the normal direction of the nominal surface before the dislocation is applied to the sampled points.

The simulated dislocation is shown to the left in Figure 2.6 and the estimated dislocation using both the inspection points and the datum points in the middle. The result of a fit using only the datum points is seen to the right. In comparison to the true dislocation the dislocation estimated from the datum points is very different. This is due to the added noise that perturbs the estimate. When all points are used in the fit, the least squares method manages to find a proper estimate of the dislocation due to the many points used, which is why the estimated dislocation using all points that is shown in the middle of Figure 2.6 is very similar to the true one. A way of improving the fit when only datum points are used is to take repeated measurements of the datum points.

2.8 Minimisation in l^p by Weighted Least Squares

The described least squares problem that gives the l^2-solution to the localisation problem can be used to approximate the l^p-solution of the problem by solving a series of weighted least squares problems, recomputing the weights at each intermediate step.

Consider the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^6} \sum_{j=1}^{n} |\boldsymbol{r}_j|^p$$

whose solution we want to approximate by solving a series of the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^6} \sum_{j=1}^{n} w_j^{(k)} |\boldsymbol{r}_j|^2,$$

each time with different weights. The solution obtained for the kth weight set $\{w_j^{(k)}\}_{j=1}^n$ is denoted by $\boldsymbol{x}^{(k)}$. By choosing the sequence of weight sets in a clever way the solution sequence $\boldsymbol{x}^{(k)}$, $k = 0, 1, \ldots$ can be made to converge to the l^p solution, $1 < p < \infty$, see Björck (1996, p. 172). The two most interesting cases are $p = 1$ and $p = \infty$. The former because its solutions tend to be more robust against outliers and the latter one since it agrees well with the tolerancing standard. Neither of these can be solved exactly by the iteratively reweighted least squares method but we can achieve a good approximation.

2.8.1 The l^1 Case

In the l^1 case Gill et al. (1981) discuss a linear problem and they suggest the use of an initial weight set $w_j^{(0)} = n^{-1}$ for all $j = 1, \ldots, n$ and to update them successively by letting $w_j^{(k+1)} = |\boldsymbol{r}_j(\boldsymbol{x}^{(k)})|^{-1}$. The method is experienced to be unstable for the non-linear least squares problem considered in this chapter. However, its performance may be slightly improved if we update the weights according to $w_j^{(k+1)} = (|\boldsymbol{r}_j(\boldsymbol{x}^{(k)})| + 100\varepsilon)^{-1}$ where ε is the machine precision of the used computer. Figure 2.7 shows the rigid best fit of a curve to a set of simulated measurements indicated by $\times$ in the figure. The solution is computed using a weight set as above and should approximate the true l^1-solution.

2.8.2 The l^∞ Problem

The l^∞ problem is harder to solve than the l^1 problem. Gill et al. (1981) suggest it the to be solved using a weight sequence $w_j^{(k+1)} = S^{-1}|\boldsymbol{r}_j(\boldsymbol{x}^{(k)})|^2 w_j^{(k)}$, where $S = \sum_{j=1}^n |\boldsymbol{r}_j(\boldsymbol{x}^{(k)})|^2 w_j^{(k)}$. This weight sequence might work well for linear problems but certainly not for this non-linear problem where it turns out to be unstable. However, if the weight sequence is modified to $w_j^{(k+1)} = S^{-1}|\boldsymbol{r}_j(\boldsymbol{x}^{(k)})| w_j^{(k)}$ instead, with an obvious modification to S, the method is seen to approach the l^∞ solution, however slowly. An example of the l^∞ solution to a rigid curve fitting problem is shown in Figure 2.9. For comparison the l^2 solution to the same problem is shown in Figure 2.8.

2.9 Conclusion

We have reviewed a method for localisation of workpieces from a set of discrete coordinate measurements of a mechanical part. Furthermore, we have presented a fully practical algorithm based on the least squares approach that can be used to compute the best fit. Some examples using an implementation of the algorithm in MATLAB confirms the practicality of the algorithm. In addition, we have

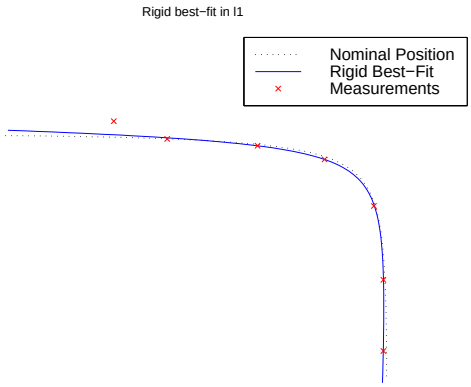

Figure 2.7: The l^1 solution of the rigid best fit problem.

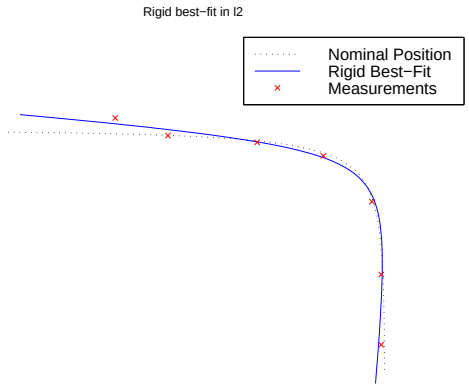

Figure 2.8: The l^2 solution of the rigid best fit problem.

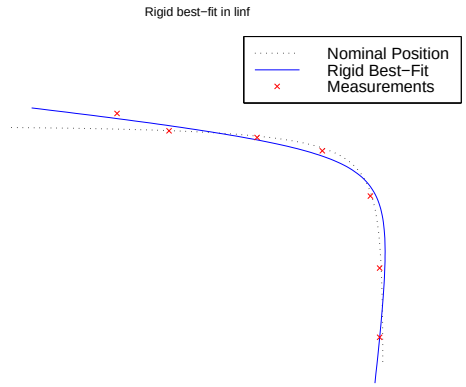

Figure 2.9: The l^∞ solution of the rigid best fit problem.

demonstrated how a sequence of iteratively reweighted least squares problems can be used to approximate the l^1 or l^∞ solution to the best fit problem.

Remember that we assumed the nominal model $\mathcal{M}$ to have continuously varying curvature. In fact, it is sufficient to require the surface of the part to have continuously varying curvature in a neighbourhood of the least distance projection of the measurements.

We remark, though, that the starting value of the rigid best fit algorithm must be close to the true solution for the suggested algorithm to be applicable. For an industrial strength implementation of the algorithm, methods for rough localisation should be added to make it more robust. This initial value can be found by, for instance, the SVD algorithm by Hanson and Norris (1981).

Chapter 3

Datum System Analysis

A method for analysing robustness of datum systems is presented. For a definition of datum systems see Sections 1.2 and 3.1. The method is based on a least squares problem that can be used to find a rigid body movement in a set of coordinate measurements of a mechanical part. The main tool of the derivation is a linearisation of the rotation and the geometry, where the former approximation effectively means that the parts are assumed to be prismatic and the latter that the rotational part of the rigid body transform is assumed to be small. The linear convergence of the approximation to the true solution is confirmed by a numerical experiment. Some examples are provided.

3.1 Introduction

A datum system is a set of points and associated guiding directions defined for each part in a mechanical assembly. The datum system determines the local coordinate system of a part and has six degrees of freedom. Recall that it takes six parameters to uniquely determine the position of a rigid object; three parameters for the translation vector and three more for the orientation. To determine all six degrees of freedom the datum system usually consists of six points. It can be a combination of different types of locating features such as *guide pins*, *holes* and *contact lugs*[1]. Figure 3.1 illustrates a contact lug and its locating feature. A contact lug locks one degree of freedom. A hole can also be used as a locating feature. If a hole is used as a locating feature the rigid body movement of the corresponding part is restricted in the plane that is tangent to the hole. Such a locating feature is modelled mathematically by a datum point defining the centre of the hole and two guiding directions that span the plane tangent to the hole. The datum point is counted with multiplicity two since the hole locks two degrees of freedom. Figure 3.2 illustrates another type of locating feature that also restricts two degrees of freedom of the part's possible rigid body movements. In principle,

[1] "**lug**. A cast or forged projection integral with, or attached to, an object, for supporting it, or attaching another part to it pivotally." from *Chambers Science and Technology Dictionary*, W & R Chambers Ltd., Edinburgh, 1988

it is equivalent to two contact lugs. The guide pin is yet another type of locating feature, see Figure 3.3. A guide pin restricts three degrees of freedom of the part i.e. it has multiplicity three.

The datum system locates each part throughout the assembly. It is also used to determine the reference coordinate system relative to which the geometric form deviations are evaluated. A geometrically ideal choice of datum system is to choose the points or datum features as far apart as possible to make them "cover" the part. Unfortunately, the ideal choice from a geometric viewpoint does not always coincide with other requirements imposed on the datum system.

Furthermore, most industrial standards, for example the STD 5026,2 (1996) of Volvo Car Corporation, specify that the guiding directions of the datum points shall coincide with the main coordinate axes x, y or z of the global coordinate system. This is a special case of the $3-2-1$ system that means that the datum points lie in three mutually orthogonal planes, where three points are located in the first plane, two in the second and one in the third. The planes are sometimes called primary, secondary and tertiary datum planes, respectively, see Figures 3.4 and 3.5. Another problem, which complicates the choice of datum system further, is the complexity of the geometry of the parts.

Taking all these aspects into consideration there are maybe a handful of plausible candidates left for datum systems of a part. Most often, the manufacturing engineer can intuitively determine which of these that form the most *robust* datum system. By a robust datum system we mean one that is insensitive to small perturbations of the datum points in the guiding directions, i.e. a small perturbation of one or more of the datum points in their guiding directions does not significantly alter the position of the part. In some cases, though, it is difficult even for an experienced manufacturing engineer to choose the most robust datum system. In such a case he needs a tool to support his choice.

In this chapter we present a method that determines the sensitivity of a datum system of a part. The results are both qualitative and quantitative in nature and can easily be visualised. The method is based on a linearisation of both the geometry of the part and the rigid body transform.

Carlson and Ahlmark (1997) have studied a similar problem but not in direct connection with the rigid best fit problem. They develop theory and methods for linear *root cause analysis* of fixture systems using measurement data from a manufacturing process. This ideas are developed further by Carlson (1999) who uses subspace techniques. Cai et al. (1997) present a method to optimise a fixture system based on a similar method to the one presented here.

3.2 Perturbation Analysis

Let $n = 6$ and let $\{\boldsymbol{p}_j\}_{n=1}^{6}$ be the datum points of the nominal model $\mathcal{M}$. Assume that we manufacture a part $\mathcal{P}$ that is *identical* to $\mathcal{M}$. During the subsequent assembly operations in the manufacturing process involving the object $\mathcal{P}$ it is located by a *fixture* that supports the workpiece $\mathcal{P}$ in its datum points. The workpiece is supported in the datum points during the inspection phase as well.

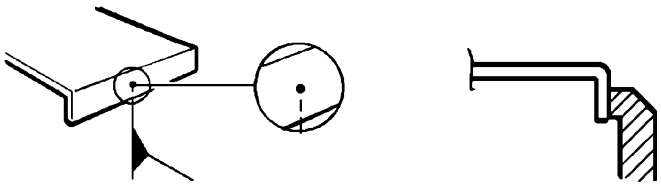

Figure 3.1: A contact lug and its locating feature. A contact lug guides the part in one direction.

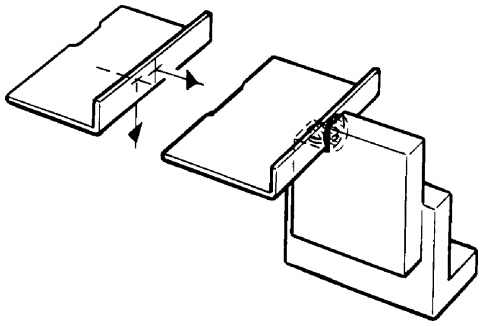

Figure 3.2: A locating feature for two guide functions.

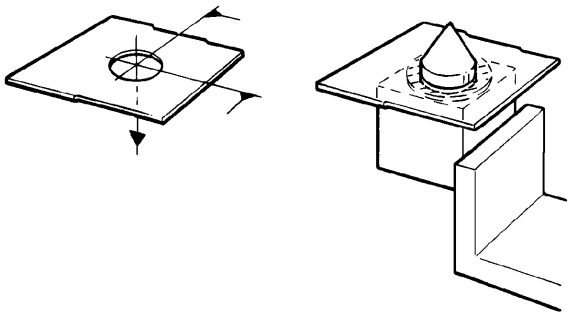

Figure 3.3: An example of a guide pin that guides a hole in three directions.

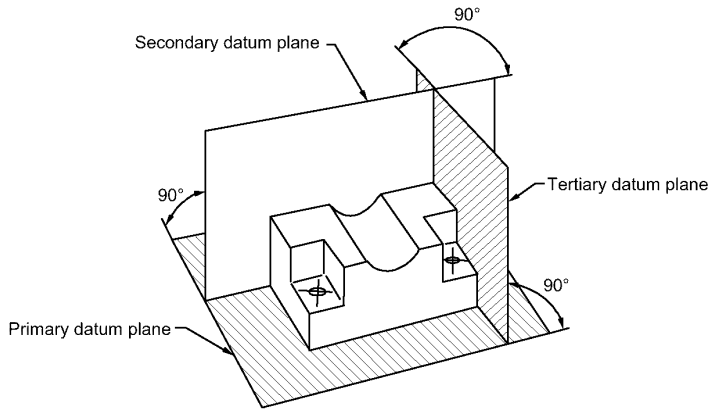

Figure 3.4: The figure illustrates three mutually orthogonal datum planes for a part. To be a 3–2–1 system, three distinct datum points shall be chosen from the primary datum plane, two from the secondary and three from the tertiary.

Since the fixtures are physical objects they are bound to suffer from geometric imperfections. The imperfect fixtures perturb the position of the part from nominal, see Figure 3.6. If the part is subject to inspection the perturbed position introduces a systematic error in the inspection measurements. The part may also be subject to a manufacturing operation in which case the workpiece dislocation may cause an unexpected manufacturing result.

In order to minimise the dislocation due to fixture imperfections the datum

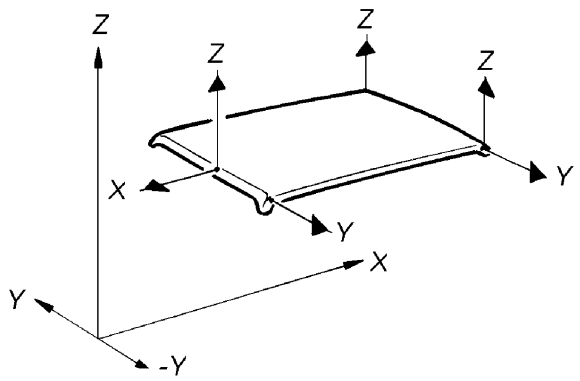

Figure 3.5: The datum system of an automotive roof. The black flags indicate the datum points and the associated letter x, y or z indicate the guiding direction. The part is seen to have three datum points that guide the workpiece in the z-direction, two in y and one in the x-direction.

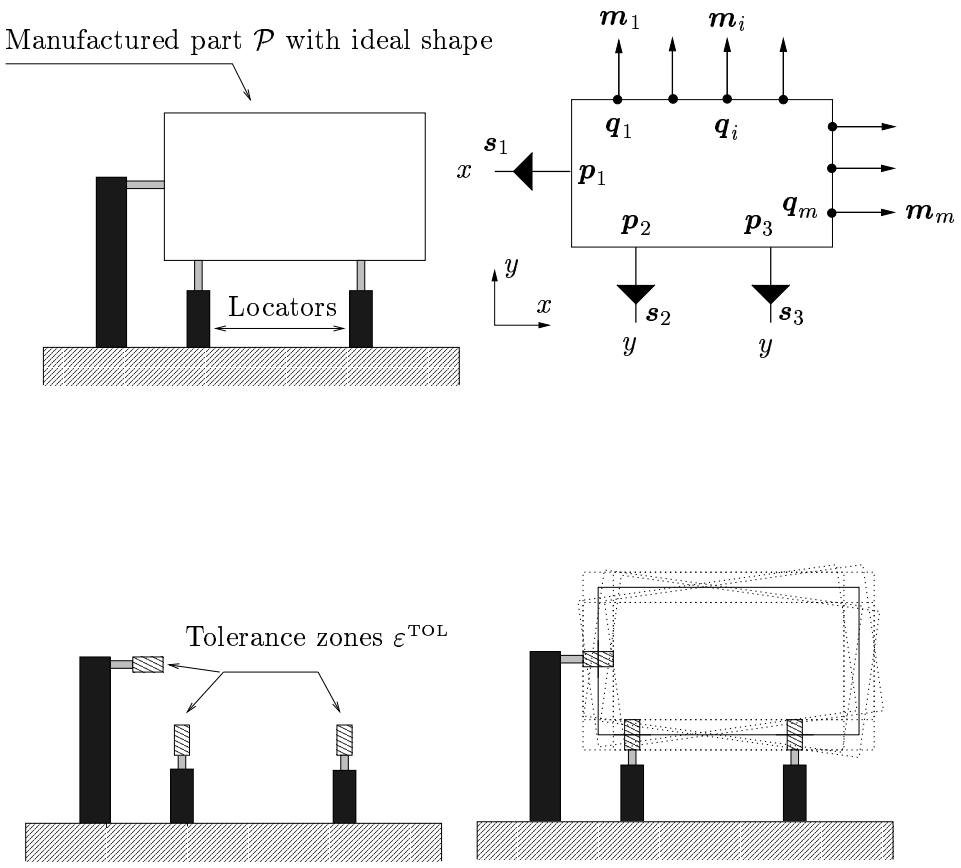

Figure 3.6: The upper left figure shows a box resting on the locators that constitute the fixture. The upper right figure illustrates the datum system of the box. The datum system consists of the datum points $\{\boldsymbol{p}_j\}_{j=1}^3$ and their guiding directions $\{\boldsymbol{s}_j\}_{j=1}^3$. Please observe that in the two dimensional case illustrated here it is sufficient to have three datum points since a rigid body motion in the plane only has three degrees of freedom. The upper right picture also illustrate the inspection points $\{\boldsymbol{q}_i\}_{i=1}^m$ and their evaluation directions $\{\boldsymbol{m}_i\}_{i=1}^m$. The lower left figure shows the tolerance zones of the locators. The lower right figure demonstrates the possible dislocations of the box due to locator perturbations within the tolerance zones.

system should be robust, i.e. insensitive to small perturbations of the locators of a fixture. This motivates the first order perturbation analysis of the rigid best fit problem for the special case when $n = 6$ that is presented in this chapter.

Let $\{\boldsymbol{p}_j\}_{j=1}^6$ be the coordinates of the locators that are in contact with the workpiece $\mathcal{P}$ in the points $\boldsymbol{c}(\boldsymbol{u}_j^0) = \boldsymbol{p}_j$, $j = 1, \ldots, 6$, where the superscript on $\boldsymbol{u}_j^0$ indicates that the parameterised model $\boldsymbol{c}$ equals $\boldsymbol{p}_j$. Assume that each of the six locators are perturbed by an amount ε_j in their guiding directions $\boldsymbol{n}_j$, $|\boldsymbol{n}_j| = 1$, for all $j = 1, \ldots, 6$. Let $\boldsymbol{x} = [\boldsymbol{t}^T \ \boldsymbol{\omega}^T]^T \in \mathbb{R}^6$ denote the parameters of the rigid body transform $\mathcal{R}(\boldsymbol{x})$. The rigid body transform is defined so as to take an arbitrary space point $\boldsymbol{p} \in \mathbb{R}^3$ to $\mathcal{R}(\boldsymbol{x})\boldsymbol{p} = \boldsymbol{t} + \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}$. The vector $\boldsymbol{t} \in \mathbb{R}^3$ is a translation vector and the vector $\boldsymbol{\omega} \in \mathbb{R}^3$ is the axis of rotation. More about rigid body transforms can be found in Appendix C.

We define the residual to be $\boldsymbol{r}_j(\boldsymbol{x}, \boldsymbol{u}_j, \varepsilon_j) = \boldsymbol{c}(\boldsymbol{u}_j) - \mathcal{R}(\boldsymbol{x})(\boldsymbol{p}_j + \varepsilon_j \boldsymbol{n}_j)$ and we also define $\varphi_j(\boldsymbol{x}, \boldsymbol{u}_j, \varepsilon_j) = |\boldsymbol{r}_j|^2/2$. Finally, we define the target function of the localisation problem $f(\boldsymbol{x}, \bar{\boldsymbol{u}}, \boldsymbol{\varepsilon}) = \sum_{j=1}^6 \varphi_j(\boldsymbol{x}, \boldsymbol{u}_j, \varepsilon_j)$, where $\boldsymbol{\varepsilon} = [\varepsilon_1 \ \cdots \ \varepsilon_6]^T$ is the vector of perturbation scalars and $\bar{\boldsymbol{u}} = [\boldsymbol{u}_1^T \cdots \boldsymbol{u}_6^T]^T$. The problem we want to solve is the following.

The datum system of the mechanical part $\mathcal{M}$ parameterised by $\boldsymbol{c}(\boldsymbol{u}), \boldsymbol{u} \in D$ is defined by the points $\{\boldsymbol{p}_j\}_{j=1}^6$ and guiding directions $\{\boldsymbol{n}_j\}_{j=1}^6$. For given perturbations $\boldsymbol{\varepsilon}$ we seek the rigid body transformation parameters $\boldsymbol{x} \in \mathbb{R}^6$ that solve

$$\min_{\boldsymbol{x} \in \mathbb{R}^6} f\big(\boldsymbol{x}, \bar{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{\varepsilon}), \boldsymbol{\varepsilon}\big) \tag{3.1}$$

where

$$\boldsymbol{u}_j(\boldsymbol{x}, \varepsilon_j) = \arg\min_{\boldsymbol{u} \in D} \varphi_j(\boldsymbol{x}, \boldsymbol{u}, \varepsilon_j), \quad j = 1, \ldots, 6. \tag{3.2}$$

As opposed to the general case (2.1)–(2.2) when $n \geq 6$ the target function will here attain zero since we only have six unknowns and six degrees of freedom in the rigid body transformation parameterised by $\boldsymbol{x} \in \mathbb{R}^6$. Since the target function f vanishes at its minimum each of the terms φ_j, $j = 1, \ldots, 6$ must vanish as well. The optimisation problem can be relaxed to only consider the residuals in the guiding directions of the datum points. To this end, define

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\varepsilon}) = \big[\boldsymbol{n}_1^T \boldsymbol{r}_1\big(\boldsymbol{x}, \boldsymbol{u}_1(\boldsymbol{x}, \varepsilon_1), \varepsilon_1\big) \quad \cdots \quad \boldsymbol{n}_6^T \boldsymbol{r}_6\big(\boldsymbol{x}, \boldsymbol{u}_6(\boldsymbol{x}, \varepsilon_6), \varepsilon_6\big)\big]^T,$$

where $\boldsymbol{u}_j(\boldsymbol{x}, \varepsilon_j) = \arg\min_{\boldsymbol{u} \in D} \varphi_j(\boldsymbol{x}, \boldsymbol{u}, \varepsilon_j)$ for all $j = 1, \ldots, 6$. The relaxed problem then reads as follows.

Let $\boldsymbol{\varepsilon} \in \mathbb{R}^6$ be a given perturbation vector that perturbs the datum system defined by the datum points $\{\boldsymbol{p}_j\}_{j=1}^6$ in its guiding directions $\{\boldsymbol{n}_j\}_{j=1}^6$. The problem is to find the rigid body transformation parameters $\boldsymbol{x} \in \mathbb{R}^6$ that solve

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\varepsilon}) = 0. \tag{3.3}$$

3.2.1 The Least Distance Problem

Before discussing ways of solving (3.3) we pay attention to the existence of the parameter function (3.2). The equation (3.2) establishes the connection between the surface parameters $\boldsymbol{u}_j$, the rigid body transformation $\boldsymbol{x}$ and the perturbation ε_j. The existence of such a connection $\boldsymbol{u}_j = \boldsymbol{u}_j(\boldsymbol{x}, \varepsilon_j)$ is guaranteed by the implicit function theorem, see Rudin (1976, p. 223). A first order necessary local condition for a solution $(\boldsymbol{x}^*, \boldsymbol{u}_j^*, \varepsilon_j^*)$ to minimise (3.2) in the interior of D is that $\varphi_{j,\boldsymbol{u}_j}(\boldsymbol{x}^*, \boldsymbol{u}_j^*, \varepsilon_j^*) = 0$. If also the second order local condition that $\varphi_{j,\boldsymbol{uu}}(\boldsymbol{x}^*, \boldsymbol{u}_j^*, \varepsilon_j^*)$ is positive definite holds then the local existence of $\boldsymbol{u}_j(\boldsymbol{x}, \varepsilon_j)$ is guaranteed in a neighbourhood of $(\boldsymbol{x}^*, \boldsymbol{u}_j^*, \varepsilon_j^*)$. As a consequence

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}}\big(\varphi_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, \varepsilon), \varepsilon)\big) &= \varphi_{\boldsymbol{xu}} + \varphi_{\boldsymbol{uu}}\boldsymbol{u}_{\boldsymbol{x}} = 0, \\ \Rightarrow \boldsymbol{u}_{\boldsymbol{x}}(0,0) &= -\varphi_{\boldsymbol{uu}}^{-1}\varphi_{\boldsymbol{xu}} = \big(\boldsymbol{c}_{\boldsymbol{u}}(\boldsymbol{u}^0)\boldsymbol{c}_{\boldsymbol{u}}(\boldsymbol{u}^0)^T\big)^{-1}\boldsymbol{c}_{\boldsymbol{u}}(\boldsymbol{u}^0)^T\left[\boldsymbol{I} \quad -[\boldsymbol{p}]\right]. \quad (3.4)\end{aligned}$$

Here we have neglected the subscript j for simplicity. In deriving equation (3.4) we used equation (2.4) which says that $\varphi_{\boldsymbol{uu}} = \boldsymbol{N} = \boldsymbol{c}_{\boldsymbol{u}}\boldsymbol{c}_{\boldsymbol{u}}^T$ when the corresponding measurement lies in the surface, i.e. when $\rho = 0$ which is the case here. We have also used the derivation of $\varphi_{\boldsymbol{xu}}$ in (2.13). Continuing to compute the dependence of $\boldsymbol{u}$ on ε we have

$$\begin{aligned}\frac{\partial}{\partial \varepsilon}\big(\varphi_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, \varepsilon), \varepsilon)\big) &= \varphi_{\boldsymbol{u}\varepsilon} + \varphi_{\boldsymbol{uu}}\boldsymbol{u}_{\varepsilon} = 0, \\ \Rightarrow \boldsymbol{u}_{\varepsilon}(0,0) &= -\varphi_{\boldsymbol{uu}}^{-1}\varphi_{\varepsilon\boldsymbol{u}} = \big(\boldsymbol{c}_u(\boldsymbol{u}^0)\boldsymbol{c}_u(\boldsymbol{u}^0)^T\big)^{-1}\boldsymbol{c}_u(\boldsymbol{u}^0)\boldsymbol{n},\end{aligned}$$

where $\varphi_{\varepsilon\boldsymbol{u}} = \boldsymbol{r}_{\boldsymbol{u}}^T\boldsymbol{r}_{\varepsilon} = -\boldsymbol{c}_{\boldsymbol{u}}(\boldsymbol{u}^0)^T\boldsymbol{n}$ when $(\boldsymbol{x}, \varepsilon) = (0,0)$ because the derivative of the residual with respect to ε is $\boldsymbol{r}_{\varepsilon} = -\partial \boldsymbol{r}/\partial\varepsilon = -\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{n}$.

3.2.2 The Sensitivity Matrix

A standard application of the implicit function theorem (Rudin, 1976, p. 223) on (3.3) guarantees the existence of a continuously differentiable function $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\varepsilon})$ that solves (3.3) whenever $\boldsymbol{h}_{\boldsymbol{x}}$ is invertible. More specifically it holds that $\boldsymbol{h}(\boldsymbol{x}(\boldsymbol{\varepsilon}), \boldsymbol{\varepsilon}) = 0$ in a neighbourhood of $(\boldsymbol{x}^*, \boldsymbol{\varepsilon}^*)$ that solves $\boldsymbol{h}(\boldsymbol{x}^*, \boldsymbol{\varepsilon}^*) = 0$ and from this we have that also the derivative is zero, i.e. $\frac{\partial}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{h}(\boldsymbol{x}(\boldsymbol{\varepsilon}), \boldsymbol{\varepsilon})) = \boldsymbol{h}_{\boldsymbol{x}}\boldsymbol{x}_{\boldsymbol{\varepsilon}} + \boldsymbol{h}_{\boldsymbol{\varepsilon}} = 0$ from which we have $\boldsymbol{x}_{\boldsymbol{\varepsilon}} = -\boldsymbol{h}_{\boldsymbol{x}}^{-1}\boldsymbol{h}_{\boldsymbol{\varepsilon}}$. Please note that (3.3) is trivially solved by $(\boldsymbol{x}^*, \boldsymbol{\varepsilon}^*) = (0,0)$.

In computing $\boldsymbol{h}_{\boldsymbol{x}}$ we take a closer look at its jth row $\boldsymbol{n}_j^T(\boldsymbol{r}_{j,\boldsymbol{x}} + \boldsymbol{r}_{j,\boldsymbol{u}}\boldsymbol{u}_{j,\boldsymbol{x}})$. To ease the burden of notation we drop the subscript j and write $\boldsymbol{r}_{\boldsymbol{x}}$ to mean $\boldsymbol{r}_{j,\boldsymbol{x}}$. From (C.2) we know that $\frac{\partial}{\partial \boldsymbol{x}}(\mathcal{R}(\boldsymbol{x})(\boldsymbol{p} + \varepsilon\boldsymbol{n})) = [\boldsymbol{I} \; -[\boldsymbol{R}(\boldsymbol{\omega})(\boldsymbol{p} + \varepsilon\boldsymbol{n})]]$ and hence that

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}}\Big(\boldsymbol{r}\big(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, \varepsilon), \varepsilon\big)\Big)\Big|_{(\boldsymbol{x},\varepsilon)=(0,0)} &= (\boldsymbol{r}_{\boldsymbol{x}} + \boldsymbol{r}_{\boldsymbol{u}}\boldsymbol{u}_{\boldsymbol{x}})\big|_{(\boldsymbol{x},\varepsilon)=(0,0)} \\ &= -\left[\boldsymbol{I} \quad -[\boldsymbol{p}]\right] + \boldsymbol{c}_{\boldsymbol{u}}(\boldsymbol{c}_{\boldsymbol{u}}\boldsymbol{c}_{\boldsymbol{u}}^T)^{-1}\boldsymbol{c}_{\boldsymbol{u}}^T\left[\boldsymbol{I} \quad -[\boldsymbol{p}]\right] = -\boldsymbol{T}\left[\boldsymbol{I} \quad -[\boldsymbol{p}]\right],\end{aligned}$$

where $\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{c_u}(\boldsymbol{c_u}\boldsymbol{c_u}^T)^{-1}\boldsymbol{c_u}^T$ is a projection matrix that takes vectors to the principal normal of $\boldsymbol{c}(\boldsymbol{u}^0)$. If we assume the guiding directions $\boldsymbol{n}$ to be collinear with the principal normal of $\boldsymbol{c}(\boldsymbol{u}^0)$ then $\boldsymbol{T}\boldsymbol{n} = \boldsymbol{n}$ and

$$\frac{\partial}{\partial \boldsymbol{x}}\Big(\boldsymbol{r}\big(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\varepsilon}), \boldsymbol{\varepsilon}\big)\Big)\Big|_{(\boldsymbol{x},\boldsymbol{\varepsilon})=(0,0)} = -\begin{bmatrix} \boldsymbol{n} \\ \boldsymbol{p} \times \boldsymbol{n} \end{bmatrix}^T.$$

Now we are in a position to define the *sensitivity matrix* $\boldsymbol{A} = -\boldsymbol{h_x}(0,0)^T$. More precisely

$$\boldsymbol{A} = -\boldsymbol{h_x}(0,0)^T = \begin{bmatrix} \boldsymbol{n}_1 & \cdots & \boldsymbol{n}_6 \\ \boldsymbol{p}_1 \times \boldsymbol{n}_1 & \cdots & \boldsymbol{p}_6 \times \boldsymbol{n}_6 \end{bmatrix} \tag{3.5}$$

We continue with the derivative of $\boldsymbol{h}$ with respect to $\boldsymbol{\varepsilon}$. Since $\boldsymbol{h}$ has components $\boldsymbol{n}_j^T\boldsymbol{r}_j$, $j = 1, \ldots, 6$ and since the jth component $\boldsymbol{n}_j^T\boldsymbol{r}_j$ only depend on ε_j and not on ε_k for $k \neq j$ then the matrix $\boldsymbol{h_\varepsilon}(\boldsymbol{x}, \boldsymbol{\varepsilon})$ is a diagonal matrix with

$$\boldsymbol{h_\varepsilon} = \operatorname{diag}\big(\begin{bmatrix} \boldsymbol{n}_1^T(\boldsymbol{r}_{1,\varepsilon_1} + \boldsymbol{r}_{1,\boldsymbol{u}}\boldsymbol{u}_{1,\varepsilon_1}) & \cdots & \boldsymbol{n}_6^T(\boldsymbol{r}_{6,\varepsilon_6} + \boldsymbol{r}_{6,\boldsymbol{u}}\boldsymbol{u}_{6,\varepsilon_6}) \end{bmatrix}\big).$$

Let us look at the jth term and drop the subscript temporarily. The derivative is

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\varepsilon}}\Big(\boldsymbol{r}\big(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\varepsilon}), \boldsymbol{\varepsilon}\big)\Big)\Big|_{(\boldsymbol{x},\boldsymbol{\varepsilon})=(0,0)} &= (\boldsymbol{r}_{\boldsymbol{u}}\boldsymbol{u}_{\boldsymbol{\varepsilon}} + \boldsymbol{r}_{\boldsymbol{\varepsilon}})\big|_{(\boldsymbol{x},\boldsymbol{\varepsilon})=(0,0)} \\ &= \boldsymbol{c_u}(\boldsymbol{c_u}\boldsymbol{c_u}^T)^{-1}\boldsymbol{c_u}^T\boldsymbol{n} - \boldsymbol{n} = -\boldsymbol{T}\boldsymbol{n} = -\boldsymbol{n}.\end{aligned}$$

Consequently, we have that $\boldsymbol{h_\varepsilon}(0,0) = -\operatorname{diag}([\boldsymbol{n}_1^T\boldsymbol{n}_1 \ \cdots \ \boldsymbol{n}_6^T\boldsymbol{n}_6]) = -\boldsymbol{I}$, i.e. the negative identity matrix.

Putting the pieces together we have that $\boldsymbol{x_\varepsilon} = -\boldsymbol{h_x}(0,0)^{-1}\boldsymbol{h_\varepsilon}(0,0) = -\boldsymbol{A}^{-T}$. We conclude that the sensitivity of the rigid body transformation parameters $\boldsymbol{x}$ as a function of locators perturbations $\boldsymbol{\varepsilon}$ is given to first order by

$$\boldsymbol{x}(\boldsymbol{\varepsilon}) = \boldsymbol{x}(0) + \boldsymbol{x_\varepsilon}(0)\boldsymbol{\varepsilon} + o(|\boldsymbol{\varepsilon}|\mathbb{1}) = -\boldsymbol{A}^{-T}\boldsymbol{\varepsilon} + o(|\boldsymbol{\varepsilon}|\mathbb{1}). \tag{3.6}$$

Please note that the sensitivity matrix $\boldsymbol{A}$ defined in (3.5) is a special case of the sensitivity matrix defined by (2.18) in Section 2.5. It is interesting, though, that the general sensitivity analysis of the rigid best fit problem that gives rise to expression (2.19) in Section 2.5 appears to be very different from the result (3.6). Further comparison and investigation of this difference is yet to be made.

3.3 The Effect on the Inspection Points

Consider a set of inspection points $\{\boldsymbol{q}_i\}_{i=1}^m$ of the part $\mathcal{P}$. We want to study the effect of locator perturbations $\boldsymbol{\varepsilon}$ in the inspection points, see the upper right illustration in Figure 3.6. We assume that the locators are perturbed in the guiding directions of the datum points. Furthermore, we assume that the locators support the surface as in Figure 3.1, i.e. that the guiding directions coincide with the principal normal of the surface in the datum points.

To each inspection point $\boldsymbol{q}_i$ we associate an *evaluation direction* $\boldsymbol{m}_i$. These are illustrated to the upper right in Figure 3.6. The geometric error is evaluated in the evaluation direction. Let us define the deviation in inspection point $\boldsymbol{q}_i$ as

$$\lambda_i(\boldsymbol{\varepsilon}) = \boldsymbol{m}_i^T \Big(\mathcal{R}\big(\boldsymbol{x}(\boldsymbol{\varepsilon})\big)^{-1} \boldsymbol{q}_i - \boldsymbol{q}_i \Big). \tag{3.7}$$

The deviation of the ith inspection point defined in (3.7) can be interpreted as the deviation in inspection point $\boldsymbol{q}_i$ due to locator perturbation $\boldsymbol{\varepsilon}$ projected in the evaluation direction $\boldsymbol{m}_i$. Please observe that the inverse transform is applied to the inspection point. Recall that the target function (3.1) is minimised with respect to the transform that best fits the measurement data to the model, not the opposite although they are equivalent. In a real situation, though, the workpiece moves as an effect of the perturbation of the locators. This movement is given by the inverse rigid body transform, hence the motivation for using the inverse transform in (3.7).

Claim C.6 asserts that the inverse of the rigid body transform applied to a point $\boldsymbol{p}$ is $\mathcal{R}(\boldsymbol{x})^{-1}\boldsymbol{q} = \boldsymbol{R}(-\boldsymbol{\omega})(\boldsymbol{q}-\boldsymbol{t})$. To first order it becomes $\mathcal{R}(\boldsymbol{x})^{-1}\boldsymbol{q} = \boldsymbol{q}-[\boldsymbol{I} \; -[\boldsymbol{q}]]\boldsymbol{x}+ o(|\boldsymbol{x}|\mathbb{1})$ that is inserted in (3.7) together with the first order approximation of the rigid body transform parameters from (3.6), namely $\boldsymbol{x}(\boldsymbol{\varepsilon}) = -\boldsymbol{A}^{-T}\boldsymbol{\varepsilon} + o(|\boldsymbol{\varepsilon}|\mathbb{1})$. Then we have

$$\lambda_i(\boldsymbol{\varepsilon}) = \boldsymbol{m}_i^T \begin{bmatrix} \boldsymbol{I} & -[\boldsymbol{q}_i] \end{bmatrix} \boldsymbol{A}^{-T}\boldsymbol{\varepsilon} + o(|\boldsymbol{\varepsilon}|\mathbb{1}) = \begin{bmatrix} \boldsymbol{m}_i \\ \boldsymbol{q}_i \times \boldsymbol{m}_i \end{bmatrix}^T \boldsymbol{A}^{-T}\boldsymbol{\varepsilon} + o(|\boldsymbol{\varepsilon}|\mathbb{1}).$$

If the *inspection matrix* is defined by

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \cdots & \boldsymbol{m}_m \\ \boldsymbol{q}_1 \times \boldsymbol{m}_1 & \boldsymbol{q}_2 \times \boldsymbol{m}_2 & \cdots & \boldsymbol{q}_m \times \boldsymbol{m}_m \end{bmatrix},$$

then the deviations can be then computed simultaneously for all inspection points by

$$\widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon}) = \boldsymbol{B}^T \boldsymbol{A}^{-T}\boldsymbol{\varepsilon}, \tag{3.8}$$

where $\widetilde{\boldsymbol{\Lambda}} = [\tilde{\lambda}_1 \; \cdots \; \tilde{\lambda}_m]^T$ and $\tilde{\lambda}_i$ is the linear part of λ_i where the remainder is dropped, i.e. $\boldsymbol{\Lambda}(\boldsymbol{\varepsilon}) = \widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon}) + o(|\boldsymbol{\varepsilon}|\mathbb{1})$.

3.3.1 Worst Case Analysis

Assuming that the perturbation of the locators $\boldsymbol{\varepsilon}$ is constrained to satisfy the bounds $|\varepsilon_j| \leq \varepsilon_j^{\text{TOL}}$ we are interested in finding the largest deviation $\tilde{\lambda}_i$ that a perturbation $\boldsymbol{\varepsilon}$ can give rise to for each inspection point $\boldsymbol{q}_i$, $i = 1, \ldots, m$. Thus we want to solve

$$\begin{aligned} &\max_{\boldsymbol{\varepsilon} \in E} |\tilde{\lambda}_i(\boldsymbol{\varepsilon})| \\ &E = \{\boldsymbol{\varepsilon} \in \mathbb{R}^6 : |\varepsilon_j| \leq \varepsilon_j^{\text{TOL}}, j = 1, \ldots, 6\}, \end{aligned}$$

for all $i = 1, \ldots, m$. The maximum is trivially given by the perturbation vector $\boldsymbol{\varepsilon}_i^* = \pm\text{diag}(\boldsymbol{\varepsilon}^{\text{TOL}})\text{sign}(\boldsymbol{A}^{-1}\boldsymbol{b}_i)$, where $\boldsymbol{\varepsilon}^{\text{TOL}} = [\varepsilon_1^{\text{TOL}}\ \varepsilon_2^{\text{TOL}}\ \cdots\ \varepsilon_6^{\text{TOL}}]^T$ and $\boldsymbol{b}_i$ is the ith column of inspection matrix $\boldsymbol{B}$, i.e. $\boldsymbol{b}_i = [\boldsymbol{m}_i^T\ (\boldsymbol{m}_i \times \boldsymbol{q}_i)^T]^T$. Thus the maximum deviation in modulus is simply

$$|\tilde{\lambda}_i(\boldsymbol{\varepsilon}_i^*)| = \boldsymbol{b}_i^T \boldsymbol{A}^{-T}\text{diag}(\boldsymbol{\varepsilon}^{\text{TOL}})\text{sign}(\boldsymbol{A}^{-1}\boldsymbol{b}_i).$$

If all locator tolerances are equal, i.e. if $\varepsilon_1^{\text{TOL}} = \varepsilon_2^{\text{TOL}} = \cdots = \varepsilon_6^{\text{TOL}} = \varepsilon^{\text{TOL}}$ then the maximum deviation at inspection point i is given by $|\lambda_i(\boldsymbol{\varepsilon}_i^*)| = \varepsilon^{\text{TOL}}\|\boldsymbol{A}^{-1}\boldsymbol{b}_i\|_1$, where $\|\cdot\|_1$ is the l^1 vector norm.

Assume that a profile tolerance relative to the datum system allows the deviation of the ith inspection point $\boldsymbol{q}_i$ in evaluation direction $\boldsymbol{m}_i$ to be λ_i^{TOL} at the most, i.e. we require $|\tilde{\lambda}_i(\boldsymbol{\varepsilon})| \leq \lambda_i^{\text{TOL}}$. It is then of interest to know how much of the tolerance that is occupied by variation due to fixture imperfections. To this end we introduce the *fixture influence coefficient* $\rho_i = |\tilde{\lambda}_i(\boldsymbol{\varepsilon})|/\lambda_i^{\text{TOL}}$. If all locator tolerances are equal then the fixture influence coefficient becomes

$$\rho_i = \frac{|\tilde{\lambda}_i(\boldsymbol{\varepsilon}_i^*)|}{\lambda_i^{\text{TOL}}} = \frac{\varepsilon^{\text{TOL}}}{\lambda_i^{\text{TOL}}}\|\boldsymbol{A}^{-1}\boldsymbol{b}_i\|_1, \quad i = 1, \ldots, m. \tag{3.9}$$

The quotient allows the impact of the locator perturbation on different areas of the model to be compared, where the areas do not necessarily need to have the same tolerance. The resulting fixture influence coefficients $\{\rho_i\}_{i=1}^m$ of the worst case analysis can be visualised in a computer-aided design (CAD) environment by colouring of the inspection part surface according to the magnitude of the coefficients.

3.3.2 Statistical Analysis

As the worst case analysis will give an overly pessimistic view of the situation it might be appropriate to consider a statistical method instead. We can, for example, assume that the perturbations ε_j are normally distributed and independent with zero mean and variance σ_j^2, i.e. $\boldsymbol{\varepsilon} \sim N_6(0, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$, where the covariance matrix is $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} = \text{diag}([\sigma_1^2\ \sigma_2^2\ \cdots\ \sigma_6^2])$. If the locator tolerances are given we can set the locator variances to be $\sigma_i = \varepsilon_i^{\text{TOL}}/3$, which is common in transitions between the deterministic and the stochastic worlds. The mean of the deviations $\widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon})$ due to stochastic locator perturbations $\boldsymbol{\varepsilon}$ is

$$\boldsymbol{\mu}_{\widetilde{\boldsymbol{\Lambda}}} = \mathbb{E}[\widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon})] = \mathbb{E}\left[\boldsymbol{B}^T\boldsymbol{A}^{-T}\boldsymbol{\varepsilon}\right] = \boldsymbol{B}^T\boldsymbol{A}^{-T}\mathbb{E}\left[\boldsymbol{\varepsilon}\right] = 0.$$

Moreover, the covariance is

$$\begin{aligned}\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{\Lambda}}} = \mathbb{E}[(\widetilde{\boldsymbol{\Lambda}} - \boldsymbol{\mu}_{\widetilde{\boldsymbol{\Lambda}}})(\widetilde{\boldsymbol{\Lambda}} - \boldsymbol{\mu}_{\widetilde{\boldsymbol{\Lambda}}})^T] &= \mathbb{E}[\widetilde{\boldsymbol{\Lambda}}\widetilde{\boldsymbol{\Lambda}}^T] = \mathbb{E}[\boldsymbol{B}^T\boldsymbol{A}^{-T}\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\boldsymbol{A}^{-1}\boldsymbol{B}] \\ &= \boldsymbol{B}^T\boldsymbol{A}^{-T}\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]\boldsymbol{A}^{-1}\boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{A}^{-T}\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}\boldsymbol{A}^{-1}\boldsymbol{B}\end{aligned}$$

and hence $\widetilde{\boldsymbol{\Lambda}} = [\tilde{\lambda}_1\ \tilde{\lambda}_2\ \cdots\ \tilde{\lambda}_m]^T \sim N_m(0, \boldsymbol{B}^T\boldsymbol{A}^{-T}\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}\boldsymbol{A}^{-1}\boldsymbol{B})$.

If the part tolerance allows the inspection point $\boldsymbol{q}_i$ to deviate at most λ_i^{TOL} in the evaluation direction $\boldsymbol{m}_i$ then we can adequately compare the effect of locator perturbation in point $\boldsymbol{q}_i$ to other points by the fixture influence coefficient

$$\rho_i = \frac{3\sigma_{ii}^{\widetilde{\boldsymbol{\Lambda}}}}{\lambda_i^{\text{TOL}}} = \frac{3\sqrt{\boldsymbol{b}_i^T \boldsymbol{A}^{-T} \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} \boldsymbol{A}^{-1} \boldsymbol{b}_i}}{\lambda_i^{\text{TOL}}},$$

where $\sigma_{ii}^{\widetilde{\boldsymbol{\Lambda}}} = (\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{\Lambda}}})_{ii}$ is the ith diagonal element of the covariance matrix $\boldsymbol{\Sigma}_{\widetilde{\boldsymbol{\Lambda}}}$. If the locator tolerances are all equal, i.e. $\varepsilon_1^{\text{TOL}} = \cdots = \varepsilon_6^{\text{TOL}} = \varepsilon^{\text{TOL}} = 3\sigma_i$, for all $i = 1, \ldots, m$, then $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} = (\varepsilon^{\text{TOL}}/3)^2 \boldsymbol{I}$ and consequently

$$\rho_i = \frac{\varepsilon^{\text{TOL}}}{\lambda_i^{\text{TOL}}} \|\boldsymbol{A}^{-1}\boldsymbol{b}_i\|_2, \quad i = 1, \ldots, m, \tag{3.10}$$

where $\|\cdot\|_2$ is the vector 2-norm. Observe the similarities between equations (3.9) and (3.10) that illustrate the close relationship between the worst case analysis and the statistical analysis.

3.3.3 Qualitative Analysis

In this section we will make use of the *singular value decomposition* (SVD) to pinpoint the weakness of a given datum system.

The Singular Value Decomposition

The value $\nu > 0$ is said to be a *singular value* of the $m \times n$ matrix $\boldsymbol{C}$ if there exist non-zero vectors $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^n$ such that

$$\boldsymbol{C}\boldsymbol{v} = \nu\boldsymbol{u} \quad \text{and} \quad \boldsymbol{C}^T\boldsymbol{u} = \nu\boldsymbol{v}.$$

The vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are *singular vectors* corresponding to ν. A matrix $\boldsymbol{C}$ can be completely decomposed into its singular values and singular vectors so that we can write $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{X}\boldsymbol{V}^T$, where $\boldsymbol{U} = [\boldsymbol{u}_1 \ \cdots \ \boldsymbol{u}_m]$ and $\boldsymbol{V} = [\boldsymbol{v}_1 \ \cdots \ \boldsymbol{v}_n]$. The matrix $\boldsymbol{X} = \text{diag}([\nu_1 \ \cdots \nu_p])$, $p = \min(m, n)$ contain the singular values in descending order. For more on the SVD see for example Golub and Van Loan (1996, p. 70).

SVD in Datum Analysis

From (3.6) we have $\boldsymbol{\varepsilon} = -\boldsymbol{A}^T\boldsymbol{x}$ when the locators are perturbed in the normal direction of the datum points. If the matrix $-\boldsymbol{A}^T$ is decomposed into its singular values then the singular value/vector triple $(\boldsymbol{x}_{\min}, \boldsymbol{\varepsilon}_{\min}, \nu_{\min})$, where $\nu_{\min}$ is the smallest singular value of $-\boldsymbol{A}^T$, correspond to the most critical rigid body motion $\boldsymbol{x}_{\min}$. Of course, the singular value $\nu_{\min}$ is required to be non-zero for the datum system to be acceptable, in which case we say that the datum system is *deterministic* (Cai et al., 1997). The singular value $\nu_{\min}$ gives qualitative information about the datum system described by the sensitivity matrix $\boldsymbol{A}$.

More specifically $\nu_{\min}\boldsymbol{\varepsilon}_{\min} = -\boldsymbol{A}^T\boldsymbol{x}_{\min}$, which means that $\boldsymbol{x}_{\min}$ is the parameter of the rigid body motion that causes the smallest effect on the locators, or

conversely, the perturbation $\boldsymbol{\varepsilon} = \nu_{\min}\boldsymbol{\varepsilon}_{\min}$ causes the rigid body movement of $\mathcal{P}$ parameterised by $\boldsymbol{x}_{\min}$. The singular value $\nu_{\min}$ is a measure of goodness of the datum system. For example, if the datum system is robust, then it is insensitive to small perturbations of the locators (by definition) and hence the smallest singular value will not be critically small. Conversely, if the datum system is of poor quality then even small perturbations $\boldsymbol{\varepsilon} = \nu_{\min}\boldsymbol{\varepsilon}_{\min}$ can cause large dislocations $\boldsymbol{x}_{\min}$. The rigid body motion parameters $\boldsymbol{x}_{\min}$ corresponds to the most critical dislocation.

The weakness of the datum system that is revealed by the singular vector $\boldsymbol{x}_{\min}$ can be visualised simply by applying the rigid body transform parameterised by $\boldsymbol{x}_{\min}$ to the part $\mathcal{M}$ under consideration. In this way the manufacturing engineer gets a clear picture of the weakness of the datum system and can make the appropriate changes to prevent the movement to occur by adjusting the datum points accordingly. For the modified datum system to be improved, the smallest singular value of the modified system $\nu_{\min}^{\text{new}}$ should be larger than the old singular value $\nu_{\min}$. See Section 3.6 for some examples of how the qualitative analysis can be used.

3.4 The Quality of the Approximation

In this section we compare the approximation (3.8) with the true solution to the original problem (3.7). As a prototype part we use the surface shown in Figure 3.7 with the datum system indicated by the arrows. We have tried to mimic the $3-2-1$ system by placing the reference points in nearly mutually orthogonal coordinate planes. The deviations are computed in the inspection points shown in Figure 3.8.

3.4.1 Checking the Convergence Rate

We perturb every point of the datum system equally much, solve the non-linear system of equations (3.3) by Newton's method and compare the deviations (3.8) in the chosen inspection points seen in Figure 3.8 with the approximation.

We denote the approximated deviation as defined in (3.8) by $\widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon}) = [\tilde{\lambda}_1 \ \cdots \ \tilde{\lambda}_m]^T$, and the true deviation is denoted by $\boldsymbol{\Lambda}(\boldsymbol{\varepsilon}) = [\lambda_1 \ \cdots \ \lambda_m]^T$ where $\lambda_i(\boldsymbol{\varepsilon})$ is as defined in (3.7). We consider perturbations $\boldsymbol{\varepsilon}$ of the form $\boldsymbol{\varepsilon} = \varepsilon \begin{bmatrix} \pm 1 & \pm 1 & \cdots & \pm 1 \end{bmatrix}^T$ so the magnitude of the perturbations at all datum points are the same. The relative error is measured by

$$E_{\max}(\varepsilon) = \frac{\|\boldsymbol{\Lambda}(\boldsymbol{\varepsilon}) - \widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon})\|_\infty}{\|\boldsymbol{\Lambda}(\boldsymbol{\varepsilon})\|_\infty},$$

where $\|\cdot\|_\infty$ denotes the maximum norm for vectors, see Golub and Van Loan (1996, p. 52). The relative error $E_{\max}(\varepsilon)$ is assumed to be of the form $E_{\max}(\varepsilon) = C\varepsilon^\alpha$, where $C > 0$ is a constant and α is the *rate of convergence* of the approximation $\widetilde{\boldsymbol{\Lambda}}$ to the true solution $\boldsymbol{\Lambda}$. For two subsequent values of ε_k, $k = 1, 2$ we can compute

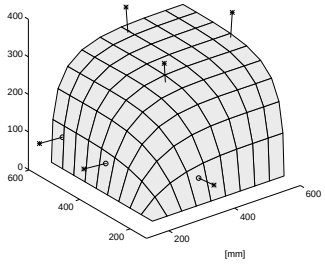

Figure 3.7: The model surface and the chosen datum system consisting of six datum points $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_6$ and their guiding directions $\boldsymbol{n}_1, \ldots, \boldsymbol{n}_6$. Observe that the datum system is of 3–2–1 type.

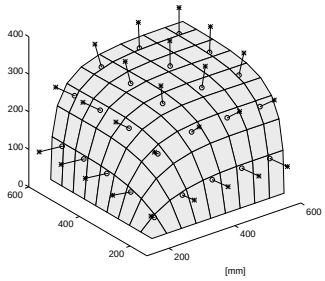

Figure 3.8: The chosen inspection points $\boldsymbol{q}_i$ and the corresponding inspection directions $\boldsymbol{m}_i$, $i = 1, \ldots, m$.

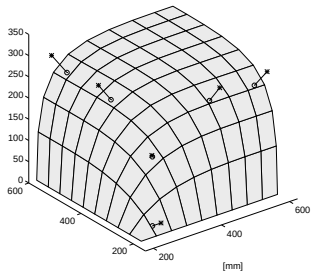

Figure 3.9: A datum system located at curved parts of the surface to illustrate the effect of curvature

ε $[mm]$	$E_{\max}(\varepsilon) \cdot 10^{-3}$	α
0.8	33.0	-
0.4	16.8	0.97
0.2	8.47	0.99
0.1	4.25	0.99
0.05	2.13	1.0
0.025	1.07	1.0
0.0125	0.534	1.0

Table 3.1: The table shows the convergence result for the locator perturbation $\boldsymbol{\varepsilon}^T = \varepsilon[-1, 1, -1, 1, -1, 1]$ for the datum system shown in Figure 3.7.

ε $[mm]$	$E_{\max}(\varepsilon) \cdot 10^{-3}$	α
0.8	19.4	-
0.4	9.52	1.0
0.2	4.72	1.0
0.1	2.35	1.0
0.05	1.17	1.0
0.025	0.585	1.0
0.0125	0.293	1.0

Table 3.2: The table shows the convergence result for the locator perturbation $\boldsymbol{\varepsilon}^T = \varepsilon[1, 1, 1, -1, -1, -1]$ for the datum system shown in Figure 3.7.

ε $[mm]$	$E_{\max}(\varepsilon) \cdot 10^{-3}$	α
0.8	396	-
0.4	241	0.71
0.2	139	0.79
0.1	76.9	0.86
0.05	40.8	0.91
0.025	21.1	0.95
0.0125	10.7	0.97

Table 3.3: The table shows the convergence result for the locator perturbation $\boldsymbol{\varepsilon}^T = \varepsilon[-1, 1, -1, 1, -1, 1]$ for the datum system shown in Figure 3.9.

ε $[mm]$	$E_{\max}(\boldsymbol{\varepsilon}) \cdot 10^{-3}$	α
0.8	123	-
0.4	47.8	1.4
0.2	22.0	1.1
0.1	10.6	1.1
0.05	5.22	1.0
0.025	2.59	1.0
0.0125	1.29	1.0

Table 3.4: The table shows the convergence result for the locator perturbation $\boldsymbol{\varepsilon}^T = \varepsilon[1, 1, 1, -1, -1, -1]$ for the datum system shown in Figure 3.9.

an estimate of α by

$$\alpha = \frac{\log\big(E_{\max}(\varepsilon_1)/E_{\max}(\varepsilon_2)\big)}{\log(\varepsilon_1/\varepsilon_2)}.$$

Tables 3.1 and 3.2 show the convergence results for the datum system shown in Figure 3.7 for two different perturbation patterns $\boldsymbol{\varepsilon}$. The results show that the approximation converges linearly to the true solution. The result is expected since the approximation is based on a linearisation of the rotation and the geometry.

To investigate the effect of curvature we carried out some investigations of a datum system where the datum points are located at curved parts of the surface as shown in Figure 3.9. The convergence result is shown in Tables 3.3 and 3.4

The rate of convergence is linear even here and the results show that the rate of convergence approaches unity for both of the perturbation patterns, however, the first converges to unity from below and the second from above. This unstable behaviour is probably due to the non-zero curvature at the datum points. The difference illustrates the non-linear effects that the curvature induces. These effects are not predicted by the linear approximation that neglects curvature.

3.4.2 A Statistical Test

According to the statistical analysis of the approximation in Section 3.3.2 the deviations $\widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon})$ are normally distributed with mean zero. In particular, we found that $\widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon}) \sim N_m(0, \boldsymbol{B}^T\boldsymbol{A}^{-T}\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}\boldsymbol{A}^{-1}\boldsymbol{B})$. In this section we perform a statistical test to verify if this is true for the true deviation $\boldsymbol{\Lambda}(\boldsymbol{\varepsilon})$ as well. In particular we will test the null hypothesis that the mean is zero for the true deviation $\tilde{\lambda}(\boldsymbol{\varepsilon})$ against the hypothesis that the mean is not zero. In order to do this we use Hotelling's T^2 test for normally distributed variables, see Johnson and Wichern (1992, p. 184).

Statistical Theory

Given an $m \times n$ matrix of variables $\boldsymbol{Y} = [\boldsymbol{y}_1\ \boldsymbol{y}_2 \cdots\ \boldsymbol{y}_m]^T$, where each row $\boldsymbol{y}_i^T$ is an observation and each column is a variable, we form the test statistics

$$T^2 = m\,\bar{\boldsymbol{y}}^T\boldsymbol{S}^{-1}\bar{\boldsymbol{y}},$$

where $\bar{\boldsymbol{y}} \in \mathbb{R}^n$ is the mean value of the m observations and $\boldsymbol{S}$ is the estimated covariance matrix defined as

$$\boldsymbol{S} = \frac{1}{m-1}\sum_{i=1}^{m}(\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T.$$

For a given significance level p the critical value T_0^2 is defined as

$$T_0^2 = \frac{(m-1)n}{m-n}F_p(n, m-n),$$

where the value of $F_p(n, m-n)$ can be looked up in a standard statistical table. We wish to test the hypothesis

$$H_0 : \boldsymbol{\mu} = 0 \tag{3.11}$$

versus

$$H_A : \boldsymbol{\mu} \neq 0 \tag{3.12}$$

at the level of significance $100p\%$. The null hypothesis that the mean is zero can be rejected if $T^2 \geq T_0^2$ at the level of significance $100p\%$, otherwise we cannot reject the null-hypothesis in favour of the alternative hypothesis H_A.

Statistical Tests

We simulated random numbers that we used to perturb the datum systems of Figures 3.7 and 3.9 in the guiding directions for different values of the locator tolerance ε^{TOL}. We used the same tolerance for all locators. Then we solved the non-linear system of equations (3.1)–(3.2) to find the rigid body transform caused by the locator perturbations. Thereafter we computed the true deviations $\boldsymbol{\Lambda}(\boldsymbol{\varepsilon})$ in the inspection points shown in Figure 3.8. Assuming the deviations $\boldsymbol{\Lambda}(\boldsymbol{\varepsilon})$ come from a normal distribution $\boldsymbol{\Lambda}(\boldsymbol{\varepsilon}) \sim N_m(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{\boldsymbol{\Lambda}})$ we used Hotelling's T^2 test to see if we could reject the null hypothesis (3.11) that the mean is zero in favour of the alternative hypothesis (3.12) that the mean is non-zero. For both configurations of datum systems we did numerical simulations for a range of variances and studied the deviation in the $n = 25$ points for each of the $m = 1000$ observations. More specifically, we generated $m = 1000$ random vectors $\boldsymbol{\varepsilon}^{(k)}$ from the distribution $N_6(0, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$ where $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}} = \boldsymbol{I}\,(\varepsilon^{\text{TOL}}/3)^2$. For each $\boldsymbol{\varepsilon}^{(k)}$ we solved the system (3.1)–(3.2) in order to compute $\boldsymbol{x}^{(k)} = \boldsymbol{x}(\boldsymbol{\varepsilon}^{(k)})$. Finally, we computed the vector $\boldsymbol{\Lambda}^{(k)} = \boldsymbol{\Lambda}(\boldsymbol{\varepsilon}^{(k)})$, where the components $\lambda_i^{(k)}$ of $\boldsymbol{\Lambda}^{(k)}$ are computed from (3.7). In order to form the test statistics T^2 we computed the mean

$$\bar{\boldsymbol{\Lambda}} = \frac{1}{1000} \sum_{k=1}^{1000} \boldsymbol{\Lambda}^{(k)}$$

and the estimated covariance matrix

$$\boldsymbol{S}_{\boldsymbol{\Lambda}} = \frac{1}{1000-1} \sum_{k=1}^{1000} (\boldsymbol{\Lambda}^{(k)} - \bar{\boldsymbol{\Lambda}})(\boldsymbol{\Lambda}^{(k)} - \bar{\boldsymbol{\Lambda}})^T.$$

Then the test statistics is formed according to $T^2 = 1000\, \bar{\boldsymbol{\Lambda}}^T \boldsymbol{S}_{\boldsymbol{\Lambda}}^{-1} \bar{\boldsymbol{\Lambda}}$. The outcome of the statistical test can be studied in Tables 3.5 and 3.6. The tables display the value of the test statistics T^2 for different values of the locator variance. At the level of significance $p = 0.9$ the critical value T_0^2 is

$$T_0^2 = \frac{(1000-1)25}{1000-25} F_{0.9}(25, 1000-25) \approx 25.61 \cdot 1.38 \approx 35.4.$$

$\varepsilon^{\mathrm{TOL}}/3$	T^2	T_0^2	$T^2 \gtrless T_0^2$	reject/accept
0.8	1641	35.4	$>$	reject
0.4	2690	–"–	$>$	–"–
0.2	2873	–"–	$>$	–"–
0.1	3151	–"–	$>$	–"–
0.05	2985	–"–	$>$	–"–

Table 3.5: The table shows the outcome of the statistical experiment for the datum system in Figure 3.7.

$\varepsilon^{\mathrm{TOL}}/3$	T^2	T_0^2	$T^2 \gtrless T_0^2$	reject/accept
0.2	2444	35.4	$>$	reject
0.1	2607	–"–	$>$	–"–
0.05	892.4	–"–	$>$	–"–

Table 3.6: The table shows the outcome of the statistical experiment for the datum system in Figure 3.9.

As the critical value T_0^2 is consequently smaller than the observed test statistics T^2 for all values of the locator tolerance $\varepsilon^{\mathrm{TOL}}$ and for both datum systems the null hypothesis $\boldsymbol{\mu} = 0$ is reject in all cases. This illustrates the need for a better approximation of the solution to the perturbed datum system than the linear one we have derived in this report if we want to do statistical predictions. Our statistical analysis of the linear approximation in Section 3.3.2 predicts the linearised deviations $\widetilde{\boldsymbol{\Lambda}}(\boldsymbol{\varepsilon})$ in the inspection points to have zero mean when the perturbations $\boldsymbol{\varepsilon}$ is distributed as $\boldsymbol{\varepsilon} \sim N_6(0, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$, a result which is not correct according to our numerical investigations.

3.5 An Algorithm

In this section we state an algorithm for datum system analysis suitable for implementation in a CAD-environment, see Algorithm 3.1.

3.6 Examples

In this section we show examples of the worst case analysis, the statistical analysis and qualitative analysis as applied to a model surface. The results are computed by a MATLAB implementation of the algorithm described in Section 3.5.

The Figures 3.10–3.12 and 3.14–3.16 show the result of the datum system analysis. The surfaces are coloured according to the magnitude of the fixture influence coefficient ρ_i for the inspection points $\boldsymbol{q}_i$, $i = 1, \ldots, m$. The inspection points are not displayed separately but are chosen very densely along the surface. In the figures, both the inspection point tolerances λ_i^{TOL} and the locator tolerances

Algorithm 3.1 An outline of the algorithm for datum system analysis.

Give type and position of datum elements (e.g. , contact lug, hole, or guide pin), and hence define $\{\boldsymbol{p}_j\}_{j=1}^6$ and $\{\boldsymbol{n}_j\}_{j=1}^6$.
Give inspection points $\{\boldsymbol{q}_i\}_{i=1}^m$ and evaluation directions $\{\boldsymbol{m}_i\}_{i=1}^m$ /* *e.g. by discretising the surface.* */
Rescale and centre the points $\{\boldsymbol{p}_j\}_{j=1}^6$ and $\{\boldsymbol{q}_i\}_{i=1}^m$ /* *for numerical reasons* */
Initialise sensitivity matrix $\boldsymbol{A}$.
Initialise inspection matrix $\boldsymbol{B}$.
Perform qualitative analysis of $\boldsymbol{A}$, i.e. compute minimum smallest singular value $\nu_{\min}$ and the corresponding singular vector $\boldsymbol{x}_{\min}$ of $\boldsymbol{A}^T$.
Visualise the qualitative result by applying the rigid body transform parameterised by $\boldsymbol{x}_{\min}$.
Compute fixture influence coefficients by worst case analysis or statistical analysis.
Visualise the result by e.g. colouring of the part.

$\varepsilon_j^{\text{TOL}}$ are all equal and of unit size, i.e. $\lambda_i^{\text{TOL}} = \varepsilon_j^{\text{TOL}} = 1$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, 6$. Hence the fixture influence coefficients ρ_i can be seen as a magnification factor of the locator tolerances. The fixture influence coefficient is in this simplified setting $\rho_i = \|\boldsymbol{A}^{-1}\boldsymbol{b}_i\|_k$, where $k = 1$ for the worst case analysis and $k = 2$ for the statistical analysis.

The qualitative analysis is visualised by applying the most sensitive rigid body movement to the part subject to analysis. The rigid body movement is visualised together with the part in nominal position. The latter is represented by a transparent surface, whereas the dislocated part is illustrated as being non-transparent. Please observe that the amount that the part is moved is not important, but only the type of movement. The sensitivity of the displayed movement is given by the minimum singular value that corresponds to the visualised singular movement.

3.6.1 Comments on the 3–2–1 System

Figure 3.10 shows the datum system analysis of an initial configuration of a system consisting of contact lugs only. The next two Figures 3.11 and 3.12 show how the initial system is improved by moving one datum point at the time. Observe the reduction in maximum deviation for the worst case analysis from 8.94 for the first down to 5.66 for the last one. Also the maximum deviation for the statistical analysis decreases, here from 4.76 via 4.55 down to 3.23. The qualitative analysis is presented in the rightmost part of the figures and gives valuable information about the most sensitive rigid body motion. To be specific the qualitative analysis pictures show the rigid body movement that causes the smallest deviations in the locating points. The analysis result can guide the manufacturing engineer in placing datum points in order to prohibit this movement. It is important, though, to keep an eye on the minimum singular value. For the given example, the singular value increases from 0.211 up to 0.234, which indicates an improved system.

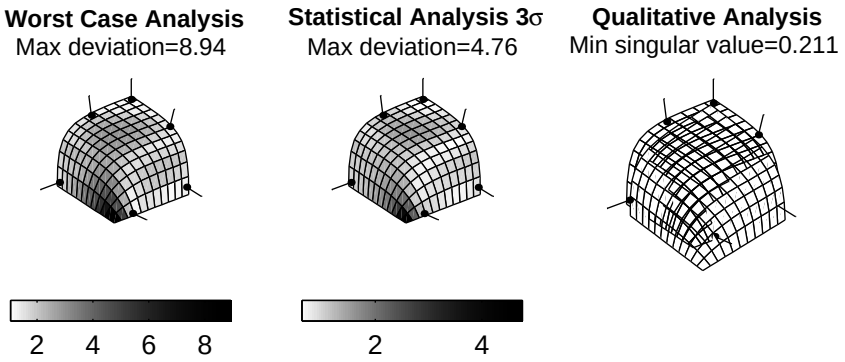

Figure 3.10: Analysis result of a datum system consisting purely of datum points with one guiding direction each. Commented in Section 3.6.1 on page 41.

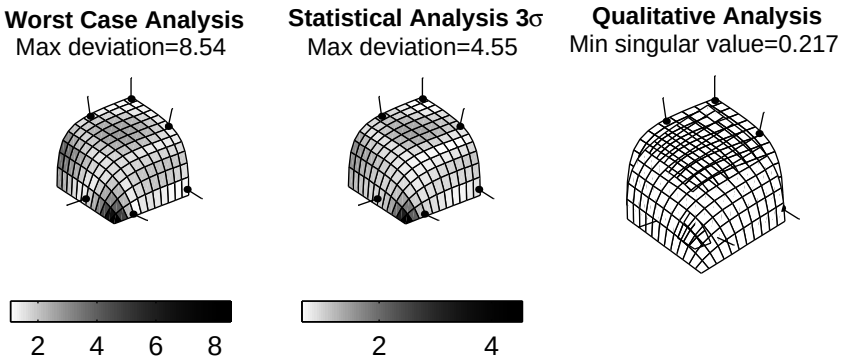

Figure 3.11: Almost the same datum system as in Figure 3.10 the only difference being the lower left datum point that is moved forwards to increase stability.

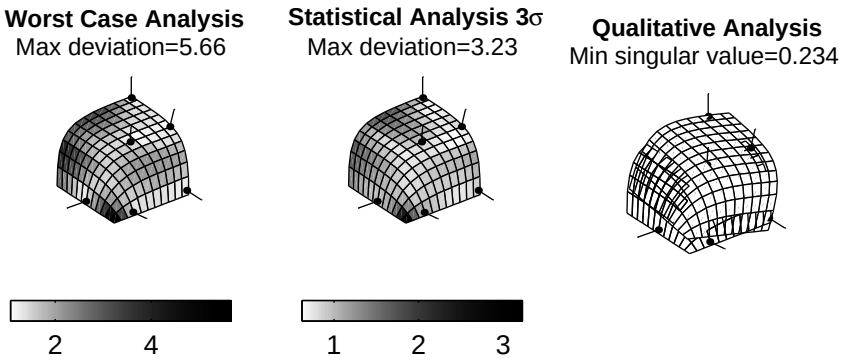

Figure 3.12: A significant improvement is made compared to Figure 3.11 by moving one of the upper datum points forward.

3.6.2 A Second Example

In order to represent different types of locating features such as holes and guide pins we have coinciding datum points with two and three guiding directions, respectively. A hole is represented by two guiding directions that spans the plane in which the hole lies and a guide pin is represented as a hole in combination with a contact lug, i.e. by three guiding directions, see Figure 3.13.

The second example consists of the Figures 3.14–3.16. Here we have chosen to apply the analysis to other types of datum features such as holes and guide pins apart from the usual contact lugs. However, the analysis results should be interpreted with caution as the approximation used for datum system analysis is derived for datum points of contact lug type that only guide the datum point in the normal direction. In this example, on the other hand, we have applied the method to cases when this assumption is not valid. It is likely, though, that the same approximation holds for this case as well, although there is no mathematical evidence to support it in this report.

The first figure in the series, i.e. Figure 3.14, shows the analysis result of the initial configuration of datum elements. The datum system is thought to be supported by a guide pin (see Figure 3.3), a locating feature for two guide functions (see Figure 3.2) and a contact lug (see Figure 3.1). The qualitative result to the right indicates that the component has a tendency to rise. This phenomenon is hopefully reduced by moving out the contact lug from the edge as in Figure 3.15. The analysis of the modified system gives a significant reduction in maximum deviation for both the worst case and the statistical analysis; they decrease from 8.06 and 3.69 to 6.82 and 2.98, respectively. However, the minimum singular value remains unchanged 0.3. The second attempt to improve the datum system fails as Figure 3.16 confirms. The contact lug is here moved once more, this time downward to try to prevent the movement detected by the current qualitative analysis. But this gives rise to an increase in maximum deviation (to 8 and 3.58) as well as a stability reduction.

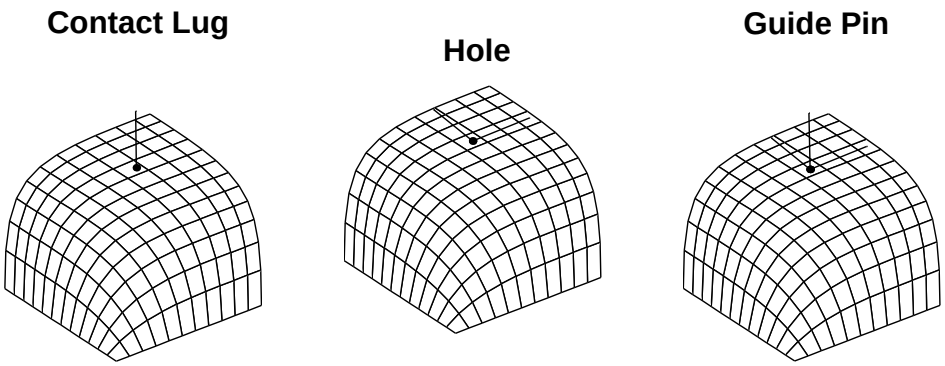

Figure 3.13: The figures illustrate how a contact lug, a hole and a guide pin are represented, respectively.

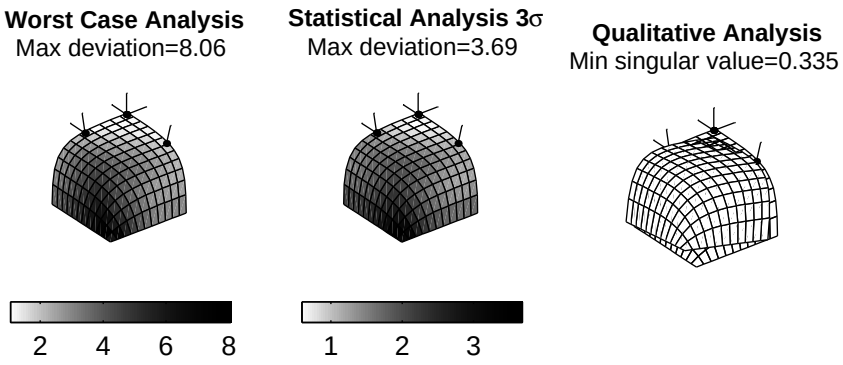

Figure 3.14: A datum system to be supported by a fixture consisting of a guide pin, a locating feature as in Figure 3.2 and a contact lug. See Section 3.6.2 on page 43 for additional comments.

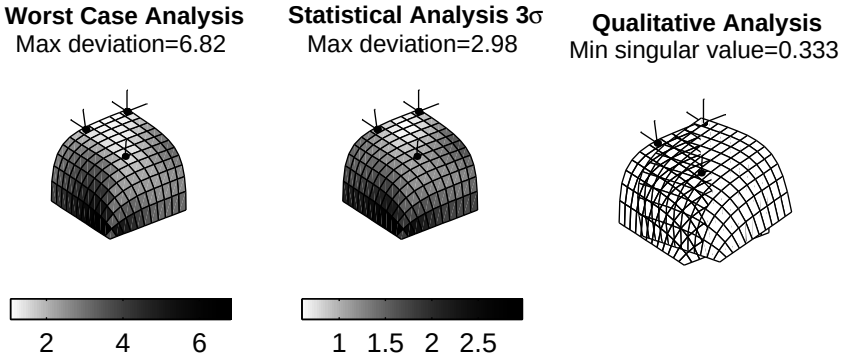

Figure 3.15: An improved variant the datum system in Figure 3.14.

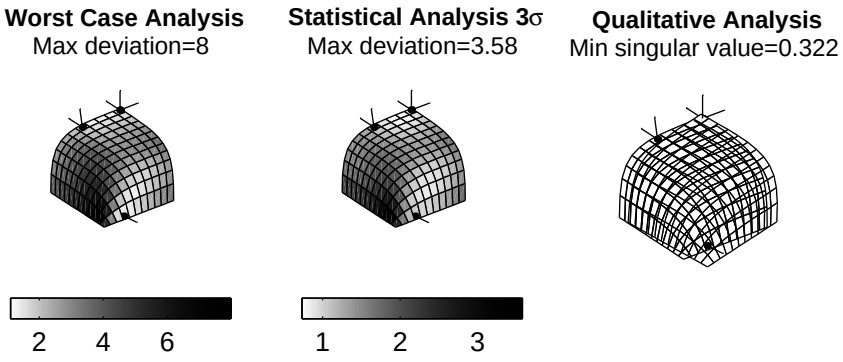

Figure 3.16: A failed attempt to further improve the datum system in Figure 3.15.

3.7 Conclusion

In this chapter we have applied the rigid best fit from Chapter 2 to the specific case where the "coordinate measurements" represent a datum system. In such a case the problem is not over-determined but simply a system of non-linear equations that can be solved by, for example, Newton's method. By linearising the rotation of the rigid body transform as well as the geometry of the part, we achieved a linear approximation of the non-linear system of equations. In applying the approximation to worst case, statistical and qualitative analysis of datum systems we achieved a fast and simple method of evaluating robustness of such systems.

The linear convergence of the approximation to the true solution of the non-linear system of equations was confirmed by numerical results. Moreover, we performed a statistical test to see whether the true solution has zero mean whenever the locator perturbations have. Unfortunately, the zero mean hypothesis was rejected which is an indication of the presence of a mean-shift not detected by the linear approximation.

We stated an algorithm to support understanding and implementation of the method. Lastly, we showed some examples of both the worst-case method and the statistical one when applied to a model surface.

In a choice between worst case analysis and statistical analysis we recommend the former since it is better in revealing poor datum systems whereas the statistical method tends to be averaging not giving as clear indications of how to improve a given datum system. The qualitative analysis could, and should, be used regardless of whether the worst case or statistical method is used.

Chapter 4

Flexible Best Fit of Curves

This chapter presents a way to model geometric form errors of mechanical parts. The aim is to create a mathematical model of the form error, a model that is based on the nominal form.

Consider a manufactured, physical part $\mathcal{P}$ with nominal form described by the ideal model $\mathcal{M}$. In order to determine the geometric quality of $\mathcal{P}$ relative to the nominal model $\mathcal{M}$ we inspect the part $\mathcal{P}$ by taking a set of discrete coordinate measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ of it. The measurements convey information about the appearance of $\mathcal{P}$, in particular they contain information about (i) the positional error and (ii) the form error. The positional error of the part is typically due to misalignment of the workpiece subject to inspection in the coordinate measuring machine (CMM). The form error has to satisfy the geometric specification of the part. The geometric specification consists of the part tolerances specified by the designer. In Chapter 2 we assumed the positional error of the part to appear as a systematic error in the measurements and derived an algorithm to filter out the rigid body transformation from the measurements. The method was based on the assumption that both the measurement error $\boldsymbol{\varepsilon}$ caused by the CMM and the geometric form error $\boldsymbol{\gamma}$ of the measured points are independent and normally distributed with zero mean. The residuals, i.e. the distance between the transformed measurements and their closest points in the nominal model are then interpreted as the form error and can be compared to the geometric tolerances to check the conformance of the part.

In cases when the form and measurement errors $\boldsymbol{\gamma}$ and $\boldsymbol{\varepsilon}$ do not satisfy the assumption about being independent and normally distributed, then we cannot justify the use of the least squares rigid best fit algorithm described in Chapter 2. This is typically the case when the form errors are systematic and is revealed by a strong dependence or correlation between neighbouring measurements even when the rigid body transform is filtered out. Then, we have to take one step further and try to model not only the localisation error $\mathcal{R}(\boldsymbol{x})$ but also the form error $\boldsymbol{\gamma}$ of the physical part $\mathcal{P}$ subject to inspection. In this case, the residuals only represent measurement error and the geometric model of the form error is compared to the geometric tolerances to see if the part is acceptable. Another

advantage of creating a mathematical description of the form error is that it can easily be visualised in the same CAD-system as the nominal part, either as a parallel, deformed object, or by colouring the nominal object according to the estimated deviations. The possibility of visualising the geometric deviations is important in aiding the manufacturing engineers to detect and pinpoint the main causes of the deviations. In this way they can make the correct decisions about the necessary corrections that need to be made to the manufacturing process in order to reduce the form errors.

This chapter presents a model that can incorporate both form and positional errors. Furthermore, we add conditions on the form error in order to separate it from the positional error. As a first step, the method is developed for curves. The motivation for using curves is that they can represent cross sections of three-dimensional objects. The described methods, however, should be possible to generalise also to work for surfaces.

The problem we want to solve can be stated as follows: Given measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ of a physical part $\mathcal{P}$ manufactured from the nominal, ideal model $\mathcal{M}$ with parametric description given by $\boldsymbol{c}(u)$, where $u \in I \subset \mathbb{R}$ (or $\mathbb{R}^2$ for surfaces), find an at least continuous vector field $\tilde{\boldsymbol{d}} : \mathcal{M} \to \mathbb{R}^3$ such that $\tilde{\boldsymbol{d}} \circ \boldsymbol{c}(u_j) = \boldsymbol{p}_j + \boldsymbol{\varepsilon}_j$, where $\boldsymbol{\varepsilon}_j$ is the measurement error. The parameter $u_j \in I$ is taken to be the parameter of the point of the deformed part given by $\tilde{\boldsymbol{d}} \circ \boldsymbol{c}$ that is closest to the measurement $\boldsymbol{p}_j$.

4.1 A Literature Review on Surface Deformation

The basic problem at hand is to approximate the measured form deviation by some function living on the part. This can be done in several ways and similar problems have been studied in the literature.

4.1.1 Free Form Deformation

A simple way to deform the nominal model is to use *free-form deformation*. Assume we are given a Bézier surface that we want to modify globally. The idea is to embed the surface in a Bézier volume. Placing the control points of the Bézier volume uniformly creates an identity map of a subset of $\mathbb{R}^3$ onto itself due to the linear precision property of the Bézier volume, see Farin (1997, p. 48). If the control points of the Bézier volume are moved the unit map is perturbed into a deformation of the initial volume. The idea is now to use the deformation map to alter the control points of the surface. This type of deformation has the disadvantage that it destroys continuity of adjacent patches since the continuity conditions of the surface are violated. Johansson (1995) has studied this problem further and transforms the geometry itself instead of just the control points of the surface. In this way, the continuity properties are preserved. A drawback of these methods are that they introduce a large number of degrees of freedom.

4.1.2 Creating a New Curve or Surface

Instead of modifying the old surface we can create a completely new curve or surface to the inspection data. After the new curve/surface is created we can compare it to the nominal and interpret the difference between them as the geometric defect of the inspected part.

Algorithms for creation of new curves and surfaces from measurement data is given by Bajaj et al. (1997). They develop a method that is able to generate both a geometric model of the measured shape and a scalar field over its surface. The algorithm requires a dense set of measurements so their method is suitable for scanned objects. By laser scanning it is possible to measure an abundance of points in short time. The method appears to be general and not to require any prior knowledge about the shape of the object to reconstruct.

Helfrich and Zwick (1996) present a trust region algorithm for optimising the shape of a parametric curve or surface by orthogonal distance regression. As already indicated, the structure of the curve or surface to model is required to be known and the paper deals with efficient algorithms for finding the optimal parameters of the model.

4.1.3 Scattered Data Interpolation

A survey of scattered data interpolation methods is given by Foley and Hagen (1994). Particularly interesting is the case where the independent data is three-dimensional and the dependent data is a scalar. For the application considered in this chapter, scattered data interpolation might be an interesting alternative since the independent, three-dimensional data could be taken to be the coordinates of the target points in the nominal model and the dependent scalar data could be the deviation of the nominal normal direction of the target points.

Scattered data interpolation methods can be divided into two groups; the global interpolants and the local interpolants, respectively. The global interpolants are typically *radial basis functions*. For the local interpolants it is common to use *triangulation methods* where the independent data generates a triangulation that interpolates the dependent data locally by e.g. barycentric combinations.

Scattered data interpolation methods have grown out of a demand from physicists to represent volumetric data. An important application arises in meteorology from the need to visualise weather data on the earth.

However, in presence of measurement error, which is the case in coordinate inspection, it is not of interest to interpolate the data. Unfortunately, there seems to be almost no literature on scattered data *approximation*.

4.1.4 Manufactured Part Models

Hulting (1995) presents a manufacturing process oriented approach to the CMM data analysis. Based on straight extrusions with square cross-section he develops a process control scheme to pinpoint and estimate form errors using *manufactured part models*. As he is targeted towards manufacturing processes he is interested in making the appropriate process correction based on the inspection data in order

to control the manufacturing process towards the nominal form. The ideas are interesting but the example he gives is far too simple to be applicable in the general case of sculptured surfaces. However, the work by Hulting is by far the most interesting we have found in this area, despite its limitations.

4.2 The Model

In Chapter 2 we modelled the positional error $\mathcal{R}(\tilde{\boldsymbol{x}})$ of the nominal part $\mathcal{M}$ relative to the true part $\mathcal{P}$. Thus, we allowed our model $\mathcal{M}$ to be transformed by the rigid body transform $\mathcal{R}(\tilde{\boldsymbol{x}})$ and tried to find the optimal transformation parameter $\tilde{\boldsymbol{x}}$. To this end we introduced the transformed model $\mathcal{M}_{\mathcal{R}}(\tilde{\boldsymbol{x}}) = \mathcal{R}(\tilde{\boldsymbol{x}})\mathcal{M}$. Recall, however, that we did not solve directly for $\tilde{\boldsymbol{x}}$ but instead for the parameters corresponding to the inverse of $\mathcal{R}(\tilde{\boldsymbol{x}})$, namely $\boldsymbol{x}$. In other words, we did not transform the model $\mathcal{M}$ but chose instead to transform the measurements $\boldsymbol{p}_j$, $j = 1, \ldots, n$. The derivation of the derivatives of the rigid best fit problem was significantly simplified by this choice.

Now, we want to extend the rigid model $\mathcal{M}_{\mathcal{R}}(\tilde{\boldsymbol{x}})$ to incorporate form deviations as well. Therefore, we introduce the *deformation parameter space* Ω and the *deformation space* $\mathcal{D}$. These definitions allow for the introduction of the deformation function $\tilde{\boldsymbol{d}} : \mathcal{M} \times \Omega \to \mathcal{D} \subset \mathbb{R}^3$ that gives the deformation vector corresponding to *deformation parameter* $\boldsymbol{\delta} \in \Omega$ for every point on $\mathcal{M}$. The deformation parameter space Ω can be thought of as $\mathbb{R}^{3n_{\boldsymbol{d}}}$ for the moment. In Section 4.11 the deformation parameter space Ω is restricted to a subset of $\mathbb{R}^{3n_{\boldsymbol{d}}}$.

The idea is that the function $\tilde{\boldsymbol{d}}$ is a vector field that gives the deformation for every point in $\mathcal{M}$. However, it is rather impractical to have a trivariate function that is only defined on a subset $\mathcal{M}$ of $\mathbb{R}^3$. It is more convenient to parameterise $\tilde{\boldsymbol{d}}$. To this end we define $\boldsymbol{d} = \tilde{\boldsymbol{d}} \circ \boldsymbol{c}$ that shares parameter space with $\boldsymbol{c}$. We have that $\boldsymbol{d} : I \times \Omega \to \mathcal{D}$, where $I = \boldsymbol{c}^{-1}(\mathcal{M})$. To be specific, we assume that the deformation $\boldsymbol{d}$ can be written as

$$\boldsymbol{d}(u;\boldsymbol{\delta}) = \sum_{i=0}^{n_{\boldsymbol{d}}-1} \bar{\boldsymbol{\delta}}_i B_i^{n_{\boldsymbol{d}}-1}(u), \quad \bar{\boldsymbol{\delta}}_i = \begin{bmatrix} \delta_{i_x} \\ \delta_{i_y} \\ \delta_{i_z} \end{bmatrix} \in \mathbb{R}^3,$$

where $B_i^{n_{\boldsymbol{d}}-1} \in C^2(I)$ for all $i = 0, \ldots, n_{\boldsymbol{d}}-1$ constitute a basis of the deformation. For convenience, we most usually write the deformation in matrix form as

$$\boldsymbol{d}(u;\boldsymbol{\delta}) = \begin{bmatrix} B_0^{n_{\boldsymbol{d}}-1}(u)\boldsymbol{I} & \cdots & B_{n_{\boldsymbol{d}}-1}^{n_{\boldsymbol{d}}-1}(u)\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\delta}}_0 \\ \vdots \\ \bar{\boldsymbol{\delta}}_{n_{\boldsymbol{d}}-1} \end{bmatrix} = \boldsymbol{B}(u)\boldsymbol{\delta}.$$

The description of the deformation as a function of the parameter space I simplifies our construction. In this way, the deformation can be described by, for example, a three-dimensional polynomial. It is seen from the above that the deformation $\boldsymbol{d}$ depends linearly on its deformation parameters $\boldsymbol{\delta} \in \Omega$. We can now introduce the *flexible model*

$$\mathcal{M}_{\boldsymbol{d}}(\boldsymbol{\delta}) = \{\boldsymbol{c}(u) + \boldsymbol{d}(u;\boldsymbol{\delta}) \in \mathbb{R}^3 : u \in I, \boldsymbol{\delta} \in \Omega\},$$

in terms of which we define the *transformed flexible model*

$$\mathcal{M}_{\mathcal{R},\boldsymbol{d}}(\tilde{\boldsymbol{x}},\boldsymbol{\delta}) = \mathcal{R}(\tilde{\boldsymbol{x}})\mathcal{M}_{\boldsymbol{d}}(\boldsymbol{\delta}).$$

As the goal is to determine the geometric quality of $\mathcal{P}$ relative to $\mathcal{M}$ we assume that a CMM has taken n discrete coordinate measurements $\{\boldsymbol{p}_j\}_{j=1}^n$ of $\mathcal{P}$. The measurements are the conveyors of the geometric information about $\mathcal{P}$. We assume that the measurements satisfy the relation

$$\boldsymbol{p}_j = \mathcal{R}(\tilde{\boldsymbol{x}})\big(\boldsymbol{c}(u_j) + \boldsymbol{d}(u_j;\boldsymbol{\delta})\big) + \boldsymbol{\varepsilon}_j, \quad j = 1,\ldots,n, \tag{4.1}$$

where $\boldsymbol{\varepsilon}_j$ denotes the measurement error of the jth measurement that we assume to be normally distributed with zero mean and variance $\mathrm{Cov}(\boldsymbol{\varepsilon}_j) = \sigma^2\boldsymbol{I}$. Please observe that we assume that the variance of the measurement noise is the same for all points.

The model of the form error $\boldsymbol{d}$ that we try to create can be viewed as a variant of the manufacturing part models developed by Hulting. The ideas behind the manufacturing part models are appealing but he provides no information on how to extend the manufacturing part models to general sculptured curves or surfaces. Recall that the example in Hulting (1995) is limited to a straight extrusion with square cross-section. In this chapter our aim is to extend the manufactured part models to more general objects. This chapter should be seen as a first step in that direction. Before we can develop methods for general sculptured surfaces, though, we have to try to see if we can develop a working method for curves. The curves can be seen as cross-sections of three-dimensional objects. Nevertheless, though it appears limited, the curve case is a very important indicator of the performance of these ideas. The surface case is bound to be more complicated, and things that work only poorly for curves cannot be expected to work at all for surfaces.

4.2.1 The Deformation Space

The deformation space $\mathcal{D}$ that describes the deformations we allow is crucial for the result. We want to allow deformations that are characteristic for $\mathcal{P}$. Selecting a basis of $\mathcal{D}$ is a very important task. If we know what types of deformation we want to detect it is relatively easy to limit the number of degrees of freedom (dof) of Ω. In order to know which deformations that are characteristic, we need a thorough knowledge of the typical physical part $\mathcal{P}$ or the limitations of the manufacturing process that prevents $\mathcal{P}$ from being identical to $\mathcal{M}$.

In this study we have chosen the Bézier polynomials as the basis of our deformation space. The reason is that they are easy to deal with, widely used and well understood in the computer-aided design (CAD) community. Moreover, the more general B-spline basis can be converted to Bézier form. However, the most general form of polynomials, the rational ones, have not been studied in this report.

Having chosen the deformation basis, there is still a large obstacle to overcome, that is the number of dof to use for the deformation. Naturally, we would like to have an abundance of measurements, a lot more than the dof of the deformation. Then, we can estimate the geometric deviations of the inspected part with narrow confidence intervals.

Alternatively, we let our deformation space have a large number of dof although the number of measurements will not determine all of them. Without restrictions on the deformation space this would give rise to an ill-posed problem. However, by imposing constraints on the deformation space the problem becomes well-posed with a unique solution. The disadvantage of this approach is that no statistical analysis can be carried out. Another disadvantage is the difficulty of selecting appropriate constraints.

The latter case allows for *a priori* information about the nominal form of the part to be taken into account. Remember that the deformation is expected to be small, and we want to use knowledge about the appearance of the nominal model $\mathcal{M}$ in the estimation procedure. Moreover, the constraints of the deformation space can be varied continuously while keeping its dof fixed. This allows for a continuous variation of the flexibility of $\mathcal{M}$. The third, and maybe most important advantage, at least from a practical point of view, is that the method with constrained deformation space means that the deformation can be achieved by varying the coefficients of the original representation of $\mathcal{M}$, i.e. without introducing a separate representation for the deformation.

4.3 The Least Squares Problem

Let $\boldsymbol{r}_j(\boldsymbol{x}, \boldsymbol{\delta}, u_j) = \boldsymbol{c}(u_j) + \boldsymbol{d}(u_j; \boldsymbol{\delta}) - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ denote the residual and let the squared residual be $\varphi_j(\boldsymbol{x}, \boldsymbol{\delta}, u_j) = |\boldsymbol{r}_j(\boldsymbol{x}, \boldsymbol{\delta}, u_j)|^2/2$. Then, we can define the target function of the optimisation problem to be $f(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}) = \sum_{j=1}^{n} \varphi_j(\boldsymbol{x}, \boldsymbol{\delta}, u_j)$. We have not introduced any weights in the target function. This is because the measurement error is assumed to have the same variance for all measurements.

Please note that we have defined the residual differently from what is suggested by the statistical model (4.1). The difference is that the rigid transform act on the measurements instead of the curve. In addition, the parameters $\tilde{\boldsymbol{x}}$ are exchanged for the parameters corresponding to the inverse of $\mathcal{R}(\tilde{\boldsymbol{x}})$, namely $\boldsymbol{x}$. This is motivated by the simpler derivation of derivatives of the target function in this setting compared to the original form. According to Claim C.7 these two ways of formulating the problem are equivalent.

We want to minimise f with respect to $\boldsymbol{x}$ and $\boldsymbol{\delta}$ using an optimal parameterisation $\boldsymbol{u} = [u_1 \ \cdots \ u_n]^T$. Hence, we want to solve the orthogonal distance regression problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^6, \boldsymbol{\delta} \in \Omega} f\big(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\delta})\big), \tag{4.2}$$

where

$$u_j(\boldsymbol{x}, \boldsymbol{\delta}) = \arg\min_{u \in I} \varphi_j(\boldsymbol{x}, \boldsymbol{\delta}, u), \quad j = 1, \ldots, n. \tag{4.3}$$

This problem is similar to the rigid best fit problem (2.1)–(2.2) but with the difference that we have added some flexibility to the underlying curve $\boldsymbol{c}$ in terms of the deformation $\boldsymbol{d}$. The least distance problem (4.3) gives the least distance

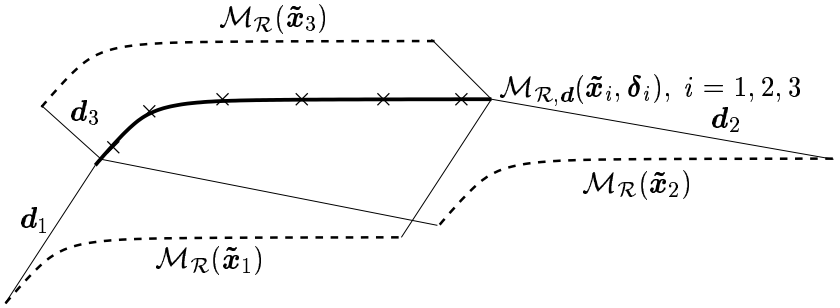

Figure 4.1: The figure illustrates three of the infinite number of solutions to the unconstrained problem (4.2)–(4.3) when the displacement and the deformation are unrelated. The thick solid line illustrates the deformed curve obtained from the flexible best fit and the dashed lines represent possible location of the rigid part of the flexible best fit.

projection of $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ on the deformed model $\mathcal{M}_{\boldsymbol{d}}$ and we refer to Section 2.3 for a discussion of existence and uniqueness of least distance projections.

For a unique solution $(\boldsymbol{x}^*, \boldsymbol{\delta}^*, \boldsymbol{u}^*)$ to exist, a necessary condition is that the number of inspection points n are equal to or greater than the number of dof of the deformation parameter space Ω. The rigid body transform $\mathcal{R}(\boldsymbol{x})$ has six degrees of freedom and the deformation $\boldsymbol{d}$ has dimension $3n_{\boldsymbol{d}}$ of Ω ($n_{\boldsymbol{d}}$ parameters for each coordinate). Hence, we require that $n \geq 6 + 3n_{\boldsymbol{d}}$. If we know that the curve is plane we can restrain the rigid body transformation and the deformation to lie in a plane in which case it is enough to require $n \geq 3 + 2n_{\boldsymbol{d}}$. Later we will discuss the under-determined case when $n < 6 + 3n_{\boldsymbol{d}}$. The problem is then ill-posed and we have to impose constraints to make the problem well-posed.

4.4 Separating Form and Positional Error

In the proposed setting, the minimisation problem (4.2)–(4.3) is not well-posed even when the condition about sufficiently many measurement points is satisfied. The observant reader might have noticed that nothing connects the rigid body transform $\mathcal{R}(\boldsymbol{x})$ to the deformation $\boldsymbol{d}$. In effect, there are infinitely many solutions to the problem. More specifically, for every position of the model $\mathcal{M}_{\mathcal{R}}(\tilde{\boldsymbol{x}})$ there is a deformation parameter $\boldsymbol{\delta}$ that solves the problem, see Figure 4.1 for an illustration.

The next few sections introduce and discuss two *separation functionals* that are essential to the methods developed here. They are essential because the separation functional makes it possible to separate the form and positional error.

4.4.1 Equilibrium Separation

From a physical viewpoint, the most natural way to define the position of the nominal model relative to the deformation is to interpret the deformation vector

field $\boldsymbol{d}$ as a force field acting on the nominal model $\mathcal{M}$. Then, we can require $\mathcal{M}$ to be in mechanical equilibrium. In order to achieve this, we introduce the force and moment functionals F_{force} and F_{moment}. Moreover, we require them to equal zero. To be specific, we require that

$$F_{\text{force}}(\boldsymbol{\delta};\boldsymbol{c}) = \int_I \boldsymbol{d}(u;\boldsymbol{\delta})|\boldsymbol{c}_u(u)|du = 0$$
$$F_{\text{moment}}(\boldsymbol{\delta};\boldsymbol{c}) = \int_I \boldsymbol{c}(u) \times \boldsymbol{d}(u;\boldsymbol{\delta})|\boldsymbol{c}_u(u)|du = 0.$$

These two equations determine three degrees of freedom each and thus we have determined the six degrees of freedom of the localisation. Recall that we assumed the deformation vector field $\boldsymbol{d}(u;\boldsymbol{\delta})$ to be linear in its deformation parameters and hence it can be written as $\boldsymbol{d}(u;\boldsymbol{\delta}) = \boldsymbol{B}(u)\boldsymbol{\delta}$, where $\boldsymbol{B}(u) \in \mathbb{R}^{3\times 3n_{\boldsymbol{d}}}$ are the deformation basis functions in matrix form. The conditions are linear in the deformation parameters and can be written in matrix form as

$$F_{\text{force}}(\boldsymbol{\delta}) = \int_I \boldsymbol{B}(u)\boldsymbol{\delta}\,|\boldsymbol{c}_u|du = \Big(\int_I \boldsymbol{B}(u)|\boldsymbol{c}_u|du\Big)\boldsymbol{\delta} = \boldsymbol{S}_{\text{force}}\boldsymbol{\delta} = 0$$
$$F_{\text{moment}}(\boldsymbol{\delta}) = \int_I \boldsymbol{c} \times (\boldsymbol{B}\boldsymbol{\delta})\,|\boldsymbol{c}_u|du = \int_I [\boldsymbol{c}]\boldsymbol{B}\boldsymbol{\delta}|\boldsymbol{c}_u|du$$
$$= \Big(\int_I [\boldsymbol{c}]\boldsymbol{B}|\boldsymbol{c}_u|du\Big)\boldsymbol{\delta} = \boldsymbol{S}_{\text{moment}}\boldsymbol{\delta} = 0.$$

These two conditions can be combined into the single condition

$$\begin{bmatrix}\boldsymbol{S}_{\text{force}}\\ \boldsymbol{S}_{\text{moment}}\end{bmatrix}\boldsymbol{\delta} = \boldsymbol{S}_{\text{eq}}\boldsymbol{\delta} = 0. \tag{4.4}$$

4.4.2 Datum Separation

When inspecting parts with a fixed coordinate system, the form error is evaluated relative to the *datum*, i.e. the reference frame. In this section, we introduce a way to evaluate the form error relative to the part's datum system[1]. We call this method *datum separation*.

In order to fix the reference frame in space we need six constraint equations. Recalling that a part is required to be equipped with a non-singular datum system consisting of six points $\{\boldsymbol{p}_k\}_{k=1}^6 \subset \mathcal{M}$, each one paired with its guiding direction $\{\boldsymbol{s}_k\}_{k=1}^6$, we define our virtual reference frame. Assume that $\{u_k\}_{k=1}^6$ is a set of parameters such that $\boldsymbol{p}_k = \boldsymbol{c}(u_k)$, $k = 1,\ldots,6$. Then our virtual reference frame is defined by the constraints

$$\boldsymbol{s}_k^{\mathrm{T}}\boldsymbol{d}(u_k;\boldsymbol{\delta}) = 0, \quad k = 1,\ldots,6.$$

[1] A datum system is a set of six points and their controlling directions on a workpiece. Adequately located, the reference system uniquely defines a local coordinate system, c.f. 3–2–1 system. For details, see Sections 1.2 and 3.1.

Since the deformation is linear in its parameters we can conveniently combine the constraints in matrix form as

$$\begin{bmatrix} \boldsymbol{s}_1^T \boldsymbol{B}(u_1) \\ \cdots \\ \boldsymbol{s}_6^T \boldsymbol{B}(u_6) \end{bmatrix} \boldsymbol{\delta} = \boldsymbol{S}_{\mathrm{ds}} \boldsymbol{\delta} = 0. \tag{4.5}$$

4.4.3 Other Options

In principle, any restraint on $\boldsymbol{d}$ giving rise to a rank six matrix is an adequate separation condition. Other options that could be considered are for example to locate the nominal model where the bending energy of the deformation is smallest or where the L^2 norm of the deformation has its minimum. These constraints, however, give rise to non-linear separation conditions.

4.5 Incorporating the Least Distance Problem

It is possible to incorporate the least distance problem (4.3) into the target function (4.2). This section describes how this is done and the argumentation follows the same pattern as in Section 2.4.1. Thus, we argue that for $(\boldsymbol{x}, \boldsymbol{\delta})$ in a neighbourhood of the triple $(\boldsymbol{x}^*, \boldsymbol{\delta}^*, u_j^*)$ that solves (4.3) for some j there is a continuously differentiable function $u_j = u_j(\boldsymbol{x}, \boldsymbol{\delta})$ in the interior of D that satisfies $\varphi_{j,u}(\boldsymbol{x}, \boldsymbol{\delta}, u_j(\boldsymbol{x}, \boldsymbol{\delta})) = 0$ if $\varphi_{j,uu}(\boldsymbol{x}^*, \boldsymbol{\delta}^*, u_j^*) \neq 0$. That the latter condition is guaranteed to hold follows from our assumption about the existence of a triple $(\boldsymbol{x}^*, \boldsymbol{\delta}^*, u_j^*)$ in which case condition (2.5) holds. Please note that the discussion about least distance projections in Section 2.3 goes through for this case too with $\boldsymbol{c}$ replaced by $\boldsymbol{c} + \boldsymbol{d}$. It follows by the chain rule that

$$\frac{\partial}{\partial \boldsymbol{x}} \Big(\varphi_{j,u} \big(\boldsymbol{x}, \boldsymbol{\delta}, u_j(\boldsymbol{x}, \boldsymbol{\delta}) \big) \Big) = \varphi_{j,\boldsymbol{x}u} + \varphi_{j,uu} u_{j,\boldsymbol{x}} = 0 \Rightarrow u_{j,\boldsymbol{x}} = -\varphi_{j,uu}^{-1} \varphi_{j,\boldsymbol{x}u} \tag{4.6}$$

$$\frac{\partial}{\partial \boldsymbol{\delta}} \Big(\varphi_{j,u} \big(\boldsymbol{x}, \boldsymbol{\delta}, u_j(\boldsymbol{x}, \boldsymbol{\delta}) \big) \Big) = \varphi_{j,\boldsymbol{\delta}u} + \varphi_{j,uu} u_{j,\boldsymbol{\delta}} = 0 \Rightarrow u_{j,\boldsymbol{\delta}} = -\varphi_{j,uu}^{-1} \varphi_{j,\boldsymbol{\delta}u}. \tag{4.7}$$

Redefining the target function as $\tilde{f}(\boldsymbol{x}, \boldsymbol{\delta}) = f(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\delta}))$ we can state the modified minimisation problem as

$$\min_{\boldsymbol{x} \in \mathbb{R}^6, \boldsymbol{\delta} \in \Omega} \tilde{f}(\boldsymbol{x}, \boldsymbol{\delta}) \tag{4.8}$$

$$\text{subject to } \boldsymbol{S}\boldsymbol{\delta} = 0, \tag{4.9}$$

where $\boldsymbol{S}$ is the separation constraint and could be either of (4.4) or (4.5) or some other user-defined separation constraint. Equations (4.6) and (4.7) are useful later when the derivatives of $\tilde{f}$ are calculated.

4.6 A Separable Problem

The minimisation problem (4.8)–(4.9) is naturally separated in the rigid body transformation variables $\boldsymbol{x}$ and the deformation variables $\boldsymbol{\delta}$. Moreover, for fixed $\boldsymbol{\delta}$

the subproblem of finding the optimal $\boldsymbol{x}$ is an unconstrained least squares problem of order six. Methods for finding the optimal $\boldsymbol{x}$ given $\boldsymbol{\delta}$ are equivalent to the problem covered in Chapter 2. We can take advantage of the separated structure of the problem in order to compute a solution. More specifically, the subproblem

$$\min_{\boldsymbol{x}\in\mathbb{R}^6} \tilde{f}(\boldsymbol{x},\boldsymbol{\delta}), \tag{4.10}$$

is easy to solve numerically for every fixed $\boldsymbol{\delta} \in \Omega$ satisfying the separation constraint (4.9). Let $\boldsymbol{x}(\boldsymbol{\delta})$ denote the solution of (4.10) and define $\psi(\boldsymbol{\delta}) = \tilde{f}(\tilde{\boldsymbol{x}}(\boldsymbol{\delta}),\boldsymbol{\delta})$. The existence of a continuously differentiable function $\boldsymbol{x}(\boldsymbol{\delta})$ is guaranteed by the implicit function theorem in a neighbourhood of a pair $(\boldsymbol{x}^*,\boldsymbol{\delta}^*)$ that solves (4.10), see Rudin (1976, p. 223). If the pair $(\boldsymbol{x}^*,\boldsymbol{\delta}^*)$ is a minimiser of (4.10) then the first order condition necessary $\tilde{f}_{\boldsymbol{x}}(\boldsymbol{x}^*,\boldsymbol{\delta}^*) = 0$ is satisfied since the problem is unconstrained in $\boldsymbol{x}$. The existence of $\boldsymbol{x}(\boldsymbol{\delta})$ is then immediately justified since the Hessian of $\tilde{f}$ with respect to $\boldsymbol{x}$ is necessarily positive definite in $(\boldsymbol{x}^*,\boldsymbol{\delta}^*)$ if the pair is a minimiser. Recall that the positive definiteness of the Hessian of $\tilde{f}$ is a sufficient requirement for the implicit function theorem to hold.

The problem is then to solve

$$\begin{aligned} &\min_{\boldsymbol{\delta}\in\Omega} \psi(\boldsymbol{\delta}) \\ &\text{subject to } \boldsymbol{S}\boldsymbol{\delta} = 0. \end{aligned} \tag{4.11}$$

Please observe that each evaluation of the function $\psi(\boldsymbol{\delta})$ involves the solution of the subproblem (4.10) to find $\boldsymbol{x}(\boldsymbol{\delta})$. The next section is devoted to calculating the derivatives of ψ.

4.7 Derivative Information

4.7.1 The Gradient

The gradient of ψ is

$$\psi_{\boldsymbol{\delta}} = \frac{\partial \tilde{f}(\boldsymbol{x}(\boldsymbol{\delta}),\boldsymbol{\delta})}{\partial \boldsymbol{\delta}} = \tilde{f}_{\boldsymbol{x}}\boldsymbol{x}_{\boldsymbol{\delta}} + \tilde{f}_{\boldsymbol{\delta}}.$$

The vector $\tilde{f}_{\boldsymbol{x}}$ occurring in the first term is almost identical to (2.10) except for an obvious modification due to the deformation function $\boldsymbol{d}$. Moreover, from our discussion in Section 4.6 it is evident that the first order condition $\tilde{f}_{\boldsymbol{x}}(\boldsymbol{x}(\boldsymbol{\delta}),\boldsymbol{\delta}) = 0$ holds and hence we only need to consider the second term $\tilde{f}_{\boldsymbol{\delta}}$. Thus

$$\tilde{f}_{\boldsymbol{\delta}} = \frac{\partial \tilde{f}(\boldsymbol{x},\boldsymbol{\delta})}{\partial \boldsymbol{\delta}} = \frac{\partial f(\boldsymbol{x},\boldsymbol{\delta},\boldsymbol{u}(\boldsymbol{x},\boldsymbol{\delta}))}{\partial \boldsymbol{\delta}} = f_{\boldsymbol{\delta}} + f_{\boldsymbol{u}}\boldsymbol{u}_{\boldsymbol{\delta}} = \sum_{j=1}^{n} \varphi_{j,\boldsymbol{\delta}} + \varphi_{j,u}u_{j,\boldsymbol{\delta}},$$

where the last terms containing $\varphi_{j,u}(\boldsymbol{x},\boldsymbol{\delta},u_j(\boldsymbol{x},\boldsymbol{\delta}))$ vanish due to the first order condition on $u_j(\boldsymbol{x},\boldsymbol{\delta})$ to solve (4.3). Hence, the derivative $\psi_{\boldsymbol{\delta}}$ takes the form

$$\psi_{\boldsymbol{\delta}} = \tilde{f}_{\boldsymbol{\delta}} = f_{\boldsymbol{\delta}} = \sum_{j=1}^{n} \varphi_{j,\boldsymbol{\delta}} = \sum_{j=1}^{n} \boldsymbol{r}_j^T \boldsymbol{d}_{\boldsymbol{\delta}} = \sum_{j=1}^{n} \boldsymbol{r}_j^T \boldsymbol{B}_j, \tag{4.12}$$

where we have left out the arguments of $\boldsymbol{r}_j$ and $\boldsymbol{B}_j$ to simplify the notation. Recall, though, that $\boldsymbol{r}_j = \boldsymbol{r}_j(\boldsymbol{x}(\boldsymbol{\delta}), \boldsymbol{\delta}, u_j(\boldsymbol{x}(\boldsymbol{\delta}), \boldsymbol{\delta}))$ and $\boldsymbol{B}_j = \boldsymbol{B}(u_j(\boldsymbol{x}(\boldsymbol{\delta}), \boldsymbol{\delta}))$.

4.7.2 The Hessian

The Hessian of $\psi(\boldsymbol{\delta})$ is

$$\psi_{\boldsymbol{\delta}\boldsymbol{\delta}}(\boldsymbol{\delta}) = \frac{\partial \psi_{\boldsymbol{\delta}}^T}{\partial \boldsymbol{\delta}} = \frac{\partial \tilde{f}_{\boldsymbol{\delta}}\big(\boldsymbol{x}(\boldsymbol{\delta}), \boldsymbol{\delta}\big)^T}{\partial \boldsymbol{\delta}} = \tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}}\boldsymbol{x}_{\boldsymbol{\delta}} + \tilde{f}_{\boldsymbol{\delta}\boldsymbol{\delta}}, \tag{4.13}$$

where $\boldsymbol{x}_{\boldsymbol{\delta}}$ is determined from the necessary first order condition for $\boldsymbol{x}(\boldsymbol{\delta})$ to be a solution to (4.10). The condition says that $\tilde{f}_{\boldsymbol{x}}(\boldsymbol{x}(\boldsymbol{\delta}), \boldsymbol{\delta}) = 0$ whose derivative with respect to $\boldsymbol{\delta}$ is

$$\frac{\partial \tilde{f}_{\boldsymbol{x}}\big(\boldsymbol{x}(\boldsymbol{\delta}), \boldsymbol{\delta}\big)}{\partial \boldsymbol{\delta}} = \tilde{f}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{x}_{\boldsymbol{\delta}} + \tilde{f}_{\boldsymbol{\delta}\boldsymbol{x}} = 0$$

from which we get $\boldsymbol{x}_{\boldsymbol{\delta}} = -\tilde{f}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\tilde{f}_{\boldsymbol{\delta}\boldsymbol{x}}$. Substituting back into (4.13) we get

$$\psi_{\boldsymbol{\delta}\boldsymbol{\delta}} = \tilde{f}_{\boldsymbol{\delta}\boldsymbol{\delta}} - \tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}}\tilde{f}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\tilde{f}_{\boldsymbol{\delta}\boldsymbol{x}}. \tag{4.14}$$

The Mixed Derivative

The mixed derivative of $\tilde{f}$ is

$$\begin{aligned}
\tilde{f}_{\boldsymbol{\delta}\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\delta}) &= \frac{\partial \tilde{f}_{\boldsymbol{x}}^T}{\partial \boldsymbol{\delta}} = \frac{\partial f_{\boldsymbol{x}}\big(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\delta})\big)^T}{\partial \boldsymbol{\delta}} = f_{\boldsymbol{\delta}\boldsymbol{x}} + f_{\boldsymbol{u}\boldsymbol{x}}\boldsymbol{u}_{\boldsymbol{\delta}} \\
&= \sum_{j=1}^{n} \varphi_{j,\boldsymbol{\delta}\boldsymbol{x}} - \varphi_{j,u\boldsymbol{x}}\varphi_{j,uu}^{-1}\varphi_{j,\boldsymbol{\delta}u} = \sum_{j=1}^{n} \boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{r}_{j,\boldsymbol{\delta}} - \boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{r}_{j,u}\varphi_{j,uu}^{-1}\boldsymbol{r}_{j,u}^T\boldsymbol{r}_{j,\boldsymbol{\delta}} \\
&= \sum_{j=1}^{n} \boldsymbol{r}_{j,\boldsymbol{x}}^T\big(\boldsymbol{I} - \boldsymbol{r}_{j,u}\varphi_{j,uu}^{-1}\boldsymbol{r}_{j,u}^T\big)\boldsymbol{r}_{j,\boldsymbol{\delta}} \qquad (4.15) \\
&= \sum_{j=1}^{n} \boldsymbol{r}_{j,\boldsymbol{x}}^T\big(\boldsymbol{I} - (\boldsymbol{c}_u + \boldsymbol{d}_u)\varphi_{j,uu}^{-1}(\boldsymbol{c}_u + \boldsymbol{d}_u)^T\big)\boldsymbol{d}_{j,\boldsymbol{\delta}} = \sum_{j=1}^{n} \boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{T}_j\boldsymbol{B}_j,
\end{aligned}$$

where $\boldsymbol{T}_j = \boldsymbol{I} - (\boldsymbol{c}_u + \boldsymbol{d}_u)\varphi_{j,uu}^{-1}(\boldsymbol{c}_u + \boldsymbol{d}_u)^T$ and we know from (C.2) in Appendix C that $\boldsymbol{r}_{j,\boldsymbol{x}} = -[\,\boldsymbol{I} \;\; -[\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}_j]]$, see also Section 2.4.2. If $\boldsymbol{r}_j = 0$ then $\varphi_{j,uu} = |\boldsymbol{r}_{j,u}|^2$ and consequently $\boldsymbol{T}_j$ is a projection matrix that spans the normal plane of $\boldsymbol{c}(u_j)$.

Second Deformation Derivative

The second derivative of $\tilde{f}$ with respect to the deformation parameters $\boldsymbol{\delta}$ is

$$\begin{aligned}
\tilde{f}_{\boldsymbol{\delta}\boldsymbol{\delta}} &= \frac{\partial \tilde{f}_{\boldsymbol{\delta}}^T}{\partial \boldsymbol{\delta}} = \frac{\partial f_{\boldsymbol{\delta}}(\boldsymbol{x},\boldsymbol{\delta},\boldsymbol{u}(\boldsymbol{x},\boldsymbol{\delta}))^T}{\partial \boldsymbol{\delta}} = f_{\boldsymbol{\delta}\boldsymbol{\delta}} + f_{\boldsymbol{u}\boldsymbol{\delta}}\boldsymbol{u}_{\boldsymbol{\delta}} = \sum_{j=1}^{n} \varphi_{j,\boldsymbol{\delta}\boldsymbol{\delta}} - \varphi_{j,u\boldsymbol{\delta}}\varphi_{j,uu}^{-1}\varphi_{j,\boldsymbol{\delta}u} \\
&= \sum_{j=1}^{n} \boldsymbol{r}_{j,\boldsymbol{\delta}}^T\boldsymbol{r}_{j,\boldsymbol{\delta}} - (\boldsymbol{r}_{j,\boldsymbol{\delta}}^T\boldsymbol{r}_{j,u} + \boldsymbol{r}_{j,u\boldsymbol{\delta}}^T\boldsymbol{r})\varphi_{j,uu}^{-1}(\boldsymbol{r}_{j,u}^T\boldsymbol{r}_{j,\boldsymbol{\delta}} + \boldsymbol{r}_j^T\boldsymbol{r}_{j,u\boldsymbol{\delta}}) \\
&= \sum_{j=1}^{n} \begin{bmatrix} \boldsymbol{r}_{j,\boldsymbol{\delta}} \\ \boldsymbol{r}_{j,u\boldsymbol{\delta}} \end{bmatrix}^T \underbrace{\left(\begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{r}_{j,u} \\ \boldsymbol{r}_j \end{bmatrix} \varphi_{j,uu}^{-1} \begin{bmatrix} \boldsymbol{r}_{j,u} \\ \boldsymbol{r}_j \end{bmatrix}^T \right)}_{=\boldsymbol{U}_j} \begin{bmatrix} \boldsymbol{r}_{j,\boldsymbol{\delta}} \\ \boldsymbol{r}_{j,u\boldsymbol{\delta}} \end{bmatrix} \\
&= \sum_{j=1}^{n} \begin{bmatrix} \boldsymbol{d}_{\boldsymbol{\delta}} \\ \boldsymbol{d}_{u\boldsymbol{\delta}} \end{bmatrix}^T \boldsymbol{U}_j \begin{bmatrix} \boldsymbol{d}_{\boldsymbol{\delta}} \\ \boldsymbol{d}_{u\boldsymbol{\delta}} \end{bmatrix} = \sum_{j=1}^{n} \begin{bmatrix} \boldsymbol{B}_j \\ \dot{\boldsymbol{B}}_j \end{bmatrix}^T \boldsymbol{U}_j \begin{bmatrix} \boldsymbol{B}_j \\ \dot{\boldsymbol{B}}_j \end{bmatrix}, \qquad (4.16)
\end{aligned}$$

where $\boldsymbol{B}_j = \boldsymbol{B}(u_j(\boldsymbol{x}(\boldsymbol{\delta}),\boldsymbol{\delta}))$ and $\dot{\boldsymbol{B}}_j = \frac{\partial}{\partial u}\boldsymbol{B}_j$. By substitution of (4.15) and (4.16) in (4.14) the second derivative of ψ is seen to be

$$\psi_{\boldsymbol{\delta}\boldsymbol{\delta}} = \sum_{j=1}^{n} \begin{bmatrix} \boldsymbol{B}_j \\ \dot{\boldsymbol{B}}_j \end{bmatrix}^T \boldsymbol{U}_j \begin{bmatrix} \boldsymbol{B}_j \\ \dot{\boldsymbol{B}}_j \end{bmatrix} - \Big(\sum_{j=1}^{n} \boldsymbol{B}_j^T\boldsymbol{T}_j\boldsymbol{r}_{j,\boldsymbol{x}}\Big)\tilde{f}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\Big(\sum_{j=1}^{n} \boldsymbol{r}_{j,\boldsymbol{x}}^T\boldsymbol{T}_j\boldsymbol{B}_j\Big) \qquad (4.17)$$

To sum up, we have calculated the derivatives of the objective function $\psi(\boldsymbol{\delta})$ to second order, i.e.

$$\psi(\boldsymbol{\delta} + \Delta\boldsymbol{\delta}) = \psi(\boldsymbol{\delta}) + \psi_{\boldsymbol{\delta}}(\boldsymbol{\delta})\Delta\boldsymbol{\delta} + \frac{1}{2}\Delta\boldsymbol{\delta}^T\psi_{\boldsymbol{\delta}\boldsymbol{\delta}}(\boldsymbol{\delta})\Delta\boldsymbol{\delta} + o(|\Delta\boldsymbol{\delta}|^2),$$

where $\psi_{\boldsymbol{\delta}}$ is taken from (4.12) and $\psi_{\boldsymbol{\delta}\boldsymbol{\delta}}$ is seen in (4.17). The expansion is useful in solving the optimisation problem (4.11) numerically.

4.8 Composite Curves

Until now we have assumed the nominal model $\mathcal{M}$ to be described by a single curve $\boldsymbol{c} \in C^2(I)$. In this section we will generalise the model and allow $\mathcal{M}$ to be described by a composite curve.

Assume that the underlying curve $\boldsymbol{c}$ is a composite curve consisting of piecewise curves, here denoted by *arcs*. Thus, we assume that our model $\mathcal{M}$ is described by the composite curve $\boldsymbol{c} : I \to \mathbb{R}^3$, where the parameter interval $I = [a, b]$ can be described by the union of its components $I = \cup_{l=1}^{L} I^{(l)}$. Each parameter interval is $I^{(l)} = [s^{(l-1)}, s^{(l)}]$, where $a = s^{(0)} < s^{(1)} < \cdots < s^{(L)} = b$. Please note that this definition of the parameter interval is not entirely satisfactory since the curve is two-valued in the joints. More specifically, at the lth joint the curve $\boldsymbol{c}(s^{(l)})$ takes on the both the value $\boldsymbol{c}^{(l)}(s^{(l)})$ and $\boldsymbol{c}^{(l+1)}(s^{(l)})$. However, this is not an issue when we only consider the trace of the curve since it is assumed to be continuous, i.e. we have $\boldsymbol{c}^{(l)}(s^{(l)}) = \boldsymbol{c}^{(l+1)}(s^{(l)})$. On the other hand, when considering the derivatives

of the curve, this definition causes a problem since the curve is not continuously differentiable, i.e. does not belong to $C^1(I)$. Nevertheless, we stick to this definition and try to stay out of trouble by dealing with the joints separately.

The breakpoints $s^{(l)}$, $l = 0, \ldots, L$ are called *knots* and the corresponding points on the curve $\boldsymbol{c}(s^{(l)})$ are denoted *joints*. The curve $\boldsymbol{c}$ can be described as

$$\boldsymbol{c}(u) = \boldsymbol{c}^{(l)}(u), \quad u \in I^{(l)}.$$

As already mentioned, we assume that the nominal curve $\boldsymbol{c}$ is continuous on the whole of I and we assume it second order continuous on each parameter interval. Put mathematically, we assume that $\boldsymbol{c} \in C(I) \cap (\cup_{l=1}^{L} C^2(I^{(l)}))$ for each coordinate. Some of the joints $\boldsymbol{c}(s^{(l)})$ might be G^1 or G^2 continuous (see e.g. Farin (1997) for a definition of G^n continuity).

As for the deformation, we want it to have the same properties as the nominal curve. In particular, we want it to share parameter space with $\boldsymbol{c}$ and we want it to consist of piecewise arcs with the same knot sequence as $\boldsymbol{c}$. Moreover, we want the deformation to inherit the continuity properties of $\boldsymbol{c}$, i.e. if $\boldsymbol{c}(s^{(l)})$ is a G^1 joint for some l we want $(\boldsymbol{c} + \boldsymbol{d})(s^{(l)})$ to be a G^1 joint as well.

How does the extension to composite curves affect the structure of the derivatives? Attempting to answer this question we recall the definition of the target function

$$f(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}) = \sum_{j=1}^{n} \varphi_j(\boldsymbol{x}, \boldsymbol{\delta}, u_j),$$

where u_j is the parameter of the least distance projection of $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ on $\mathcal{M}_{\boldsymbol{d}}$. Recall also that the jth term of the sum φ_j is the squared distance between the transformed measurement $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$ and its least distance projection on $\mathcal{M}_{\boldsymbol{d}}$ given by parameter u_j. When $\boldsymbol{c}+\boldsymbol{d}$ is a composite curve we can group the measurements according to which arc their least distance projection hits (assuming it exists). If we also create an internal ordering within these groups we can define an index function $j = j(l,k)$ for all $l = 1, \ldots, L$, $k = 1, \ldots, n_l$ that gives the original measurement index j as a function of the arc l that contains its least distance projection and its internal number k within group l. This construction allows us to write the target function as

$$f(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}) = \sum_{l=1}^{L} \sum_{k=1}^{n_l} \varphi_{j(l,k)}\big(\boldsymbol{x}, \boldsymbol{\delta}, u_{j(l,k)}\big).$$

Let us assume that $\boldsymbol{u}$ are the parameters of the deformed object that correspond to the least distance projection of $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}_j$. A standard application of the implicit function theorem asserts the existence of a continuously differentiable function $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\delta})$ and we can define $\tilde{f}(\boldsymbol{x}, \boldsymbol{\delta}) = f(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\delta}))$. The problem can be separated as in Section 4.6 and another standard application of the implicit function theorem guarantees that there exists a continuously differentiable function $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\delta})$ and we can therefore define $\psi(\boldsymbol{\delta}) = \tilde{f}(\boldsymbol{x}(\boldsymbol{\delta}), \boldsymbol{\delta})$. Working through the derivations of the derivatives in Section 4.7 with $\boldsymbol{\delta} = [\boldsymbol{\delta}^{(1)^T} \cdots \boldsymbol{\delta}^{(L)^T}]^T$ we get the

result that the first derivative is $\psi_{\boldsymbol{\delta}^{(l)}} = \tilde{f}_{\boldsymbol{\delta}^{(l)}}$ as in (4.12) and the second derivative is

$$\psi_{\boldsymbol{\delta}^{(k)}\boldsymbol{\delta}^{(l)}} = \begin{cases} \tilde{f}_{\boldsymbol{\delta}^{(k)}\boldsymbol{\delta}^{(l)}} - \tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}^{(k)}}\tilde{f}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\tilde{f}_{\boldsymbol{\delta}^{(l)}\boldsymbol{x}}, & l = k \\ -\tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}^{(k)}}\tilde{f}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\tilde{f}_{\boldsymbol{\delta}^{(l)}\boldsymbol{x}}, & l \neq k. \end{cases}$$

From this the composite gradient is seen to be

$$\psi_{\boldsymbol{\delta}} = \begin{bmatrix} \tilde{f}_{\boldsymbol{\delta}^{(1)}} \\ \vdots \\ \tilde{f}_{\boldsymbol{\delta}^{(L)}} \end{bmatrix}$$

and the composite Hessian

$$\psi_{\boldsymbol{\delta}\boldsymbol{\delta}} = \begin{bmatrix} \tilde{f}_{\boldsymbol{\delta}^{(1)}\boldsymbol{\delta}^{(1)}} & & 0 \\ & \ddots & \\ 0 & & \tilde{f}_{\boldsymbol{\delta}^{(L)}\boldsymbol{\delta}^{(L)}} \end{bmatrix} - \begin{bmatrix} \tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}^{(1)}} \\ \vdots \\ \tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}^{(L)}} \end{bmatrix} \tilde{f}_{\boldsymbol{x}\boldsymbol{x}}^{-1} \begin{bmatrix} \tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}^{(1)}} \\ \vdots \\ \tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}^{(L)}} \end{bmatrix}^T,$$

where $\tilde{f}_{\boldsymbol{\delta}^{(l)}}$, $\tilde{f}_{\boldsymbol{x}\boldsymbol{\delta}^{(l)}}$ and $\tilde{f}_{\boldsymbol{\delta}^{(l)}\boldsymbol{\delta}^{(l)}}$ are computed from (4.12), (4.15) and (4.16), respectively, with $\boldsymbol{\delta}^{(l)}$ here corresponding to $\boldsymbol{\delta}$ there.

4.9 Geometric Continuity

This section is devoted to deriving conditions for continuity and first order geometric continuity of the deformation $\boldsymbol{d}$.

4.9.1 G^0 Continuity

In order to derive conditions for continuity of the deformation we consider the lth joint of the curve and simply require the deformed curve to be continuous in this point. This means that the pair of arcs that the constitute the curve around this point should have one point in common. Put mathematically, continuity of $(\boldsymbol{c}+\boldsymbol{d})$ at the point $(\boldsymbol{c}+\boldsymbol{d})(s)$ implies that

$$\lim_{t\to u^-}(\boldsymbol{c}+\boldsymbol{d})(t) = \lim_{t\to u^+}(\boldsymbol{c}+\boldsymbol{d})(t) = (\boldsymbol{c}+\boldsymbol{d})(u),$$

for all $u \in I$. In particular, the condition holds for the curve joints, i.e. when $t = \boldsymbol{s}^{(l)}$, $l = 1, \ldots, L-1$. Stating the condition in terms of the arcs themselves we have

$$\lim_{u\to s^{(l)}-}(\boldsymbol{c}^{(l)}+\boldsymbol{d}^{(l)})(u) = \lim_{u\to s^{(l)}+}(\boldsymbol{c}^{(l+1)}+\boldsymbol{d}^{(l+1)})(u).$$

Because of continuity of the curve $\boldsymbol{c}$, the condition above implies that the deformation itself is continuous, i.e. that $\boldsymbol{d}^{(l)}(s^{(l)};\boldsymbol{\delta}^{(l)}) = \boldsymbol{d}^{(l+1)}(s^{(l)};\boldsymbol{\delta}^{(l+1)})$. Recalling that we assume the deformation to be linear in its deformation parameters $\boldsymbol{\delta}$ we

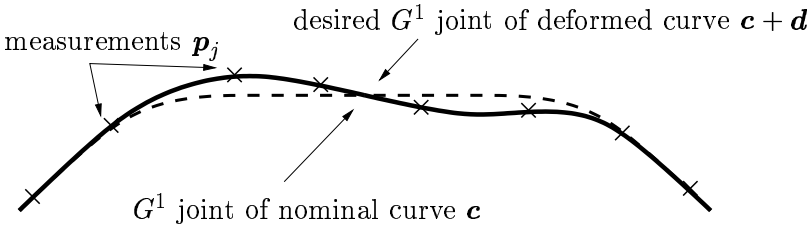

Figure 4.2: The example illustrates a case when we deformed curve is desired to inherit the G^1 condition of the nominal curve. The thick solid line illustrate the deformed curve of the flexible best fit to the measurements and the dashed line represent the nominal model.

can write $\boldsymbol{d}^{(l)}(s^{(l)};\boldsymbol{\delta}^{(l)}) = \boldsymbol{B}^{(l)}(s^{(l)})\boldsymbol{\delta}^{(l)}$. Because of the linearity we can write the continuity condition of the lth joint as

$$\begin{bmatrix}\boldsymbol{B}^{(l)}(s^{(l)}) & -\boldsymbol{B}^{(l+1)}(s^{(l)})\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}^{(l)}\\ \boldsymbol{\delta}^{(l+1)}\end{bmatrix} = 0.$$

By combining the continuity conditions for all of the $L-1$ joints the final G^0 condition can be written as

$$\begin{bmatrix}\boldsymbol{B}^{(1)}(s^{(1)}) & -\boldsymbol{B}^{(2)}(s^{(1)}) & & 0\\ & \ddots & \ddots & \\ 0 & & \boldsymbol{B}^{(L-1)}(s^{(L-1)}) & -\boldsymbol{B}^{(L)}(s^{(L-1)})\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}^{(1)}\\ \boldsymbol{\delta}^{(2)}\\ \vdots\\ \boldsymbol{\delta}^{(L-1)}\\ \boldsymbol{\delta}^{(L)}\end{bmatrix} = \boldsymbol{G}_0\boldsymbol{\delta} = 0, \tag{4.18}$$

where $\boldsymbol{G}_0$ is the G^0 continuity constraint matrix, independent of the deformation parameters $\boldsymbol{\delta}$.

4.9.2 G^1 Continuity

Consider the lth joint of the nominal composite curve $\boldsymbol{c}$. Set $\boldsymbol{t}_{\mathrm{L}} = \boldsymbol{c}_u^{(l)}(s^{(l)})$ and $\boldsymbol{t}_{\mathrm{R}} = \boldsymbol{c}_u^{(l+1)}(s^{(l)})$. If $\boldsymbol{c}(s^{(l)})$ is a G^1 joint, then the angle between the vectors $\boldsymbol{t}_{\mathrm{L}}$ and $\boldsymbol{t}_{\mathrm{R}}$ is zero, i.e. we conclude from the well-known identity $\boldsymbol{t}_{\mathrm{L}}^T\boldsymbol{t}_{\mathrm{R}} = |\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|\cos\alpha$, where α is the angle between $\boldsymbol{t}_{\mathrm{L}}$ and $\boldsymbol{t}_{\mathrm{R}}$, that

$$\boldsymbol{t}_{\mathrm{L}}^T\boldsymbol{t}_{\mathrm{R}} = |\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|. \tag{4.19}$$

Let us now introduce a shorthand notation for the right and left limits of the derivatives of the deformation $\boldsymbol{d}$ in $s^{(l)}$ Let $\boldsymbol{u}_{\mathrm{L}} = \boldsymbol{d}_u^{(l)}(s^{(l)};\boldsymbol{\delta}^{(l)})$ and $\boldsymbol{u}_{\mathrm{R}} = \boldsymbol{d}_u^{(l+1)}(s^{(l)};\boldsymbol{\delta}^{(l+1)})$. Recalling that $\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}}$ and $\boldsymbol{t}_{\mathrm{R}}+\boldsymbol{u}_{\mathrm{R}}$ are the left and right limits of the tangent of the deformed curve at joint l we want the identity

$$(\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}})^T(\boldsymbol{t}_{\mathrm{R}}+\boldsymbol{u}_{\mathrm{R}}) = |\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}+\boldsymbol{u}_{\mathrm{R}}| \tag{4.20}$$

to hold in order for the tangent continuity to be preserved by the deformed curve. Note that equation (4.20) is the G^1 condition for the deformed curve. Since the vectors $\boldsymbol{u}_{\mathrm{L}}$ and $\boldsymbol{u}_{\mathrm{R}}$ are assumed small (due to the deformation $\boldsymbol{d}$) we can approximate the right hand side of (4.20) by a second order Taylor series (see equation (B.1))

$$|\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}+\boldsymbol{u}_{\mathrm{R}}| = |\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}| + \frac{|\boldsymbol{t}_{\mathrm{L}}|}{|\boldsymbol{t}_{\mathrm{R}}|}\boldsymbol{t}_{\mathrm{R}}^T\boldsymbol{u}_{\mathrm{R}} + \frac{|\boldsymbol{t}_{\mathrm{R}}|}{|\boldsymbol{t}_{\mathrm{L}}|}\boldsymbol{t}_{\mathrm{L}}^T\boldsymbol{u}_{\mathrm{L}} + \frac{|\boldsymbol{t}_{\mathrm{L}}|}{2|\boldsymbol{t}_{\mathrm{R}}|}\boldsymbol{u}_{\mathrm{R}}^T\Big(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{R}}\boldsymbol{t}_{\mathrm{R}}^T}{|\boldsymbol{t}_{\mathrm{R}}|^2}\Big)\boldsymbol{u}_{\mathrm{R}}$$
$$+\frac{|\boldsymbol{t}_{\mathrm{R}}|}{2|\boldsymbol{t}_{\mathrm{L}}|}\boldsymbol{u}_{\mathrm{L}}^T\Big(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{L}}^T}{|\boldsymbol{t}_{\mathrm{L}}|^2}\Big)\boldsymbol{u}_{\mathrm{L}} + \frac{\boldsymbol{u}_{\mathrm{L}}^T\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{R}}^T\boldsymbol{u}_{\mathrm{R}}}{|\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|} + o\big(|\boldsymbol{u}_{\mathrm{L}}|^2+|\boldsymbol{u}_{\mathrm{R}}|^2\big). \quad (4.21)$$

Using that $\boldsymbol{t}_{\mathrm{L}}^T\boldsymbol{t}_{\mathrm{R}} = |\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|$ from equation (4.19) and collecting the first order terms on the left hand side and the second order terms on the right hand side in equation (4.21) we have

$$(\boldsymbol{t}_{\mathrm{R}}-\frac{|\boldsymbol{t}_{\mathrm{R}}|}{|\boldsymbol{t}_{\mathrm{L}}|}\boldsymbol{t}_{\mathrm{L}})^T\boldsymbol{u}_{\mathrm{L}} + (\boldsymbol{t}_{\mathrm{L}}-\frac{|\boldsymbol{t}_{\mathrm{L}}|}{|\boldsymbol{t}_{\mathrm{R}}|}\boldsymbol{t}_{\mathrm{R}})^T\boldsymbol{u}_{\mathrm{R}} = \frac{|\boldsymbol{t}_{\mathrm{L}}|}{2|\boldsymbol{t}_{\mathrm{R}}|}\boldsymbol{u}_{\mathrm{R}}^T\Big(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{R}}\boldsymbol{t}_{\mathrm{R}}^T}{|\boldsymbol{t}_{\mathrm{R}}|^2}\Big)\boldsymbol{u}_{\mathrm{R}}$$
$$+\frac{|\boldsymbol{t}_{\mathrm{R}}|}{2|\boldsymbol{t}_{\mathrm{L}}|}\boldsymbol{u}_{\mathrm{L}}^T\Big(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{L}}^T}{|\boldsymbol{t}_{\mathrm{L}}|^2}\Big)\boldsymbol{u}_{\mathrm{L}} - \boldsymbol{u}_{\mathrm{L}}\Big(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{R}}^T}{|\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|}\Big)\boldsymbol{u}_{\mathrm{R}} + o\big(|\boldsymbol{u}_{\mathrm{L}}|^2+|\boldsymbol{u}_{\mathrm{R}}|^2\big). \quad (4.22)$$

The vectors $\boldsymbol{t}_{\mathrm{L}}$ and $\boldsymbol{t}_{\mathrm{R}}$ are collinear due to the tangent continuity of $\boldsymbol{c}$, hence by letting $\boldsymbol{t}^{(l)} = \boldsymbol{t}_{\mathrm{L}}/|\boldsymbol{t}_{\mathrm{L}}|$ we also have $\boldsymbol{t}^{(l)} = \boldsymbol{t}_{\mathrm{R}}/|\boldsymbol{t}_{\mathrm{R}}|$. The notation will be simplified by defining $\boldsymbol{U}^{(l)} = \boldsymbol{I} - \boldsymbol{t}^{(l)}\boldsymbol{t}^{(l)T}$. By substituting these new definitions into (4.22) the left hand side vanishes and thus

$$0 = \frac{1}{2|\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|}(|\boldsymbol{t}_{\mathrm{R}}|^2\boldsymbol{u}_{\mathrm{L}}^T\boldsymbol{U}^{(l)}\boldsymbol{u}_{\mathrm{L}} + |\boldsymbol{t}_{\mathrm{L}}|^2\boldsymbol{u}_{\mathrm{R}}^T\boldsymbol{U}^{(l)}\boldsymbol{u}_{\mathrm{R}} - 2|\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|\boldsymbol{u}_{\mathrm{L}}^T\boldsymbol{U}^{(l)}\boldsymbol{u}_{\mathrm{R}})$$
$$= \frac{1}{2|\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{R}}|}(|\boldsymbol{t}_{\mathrm{R}}|\boldsymbol{u}_{\mathrm{L}} - |\boldsymbol{t}_{\mathrm{L}}|\boldsymbol{u}_{\mathrm{R}})^T\boldsymbol{U}^{(l)}(|\boldsymbol{t}_{\mathrm{R}}|\boldsymbol{u}_{\mathrm{L}} - |\boldsymbol{t}_{\mathrm{L}}|\boldsymbol{u}_{\mathrm{R}}), \quad (4.23)$$

where we have dropped the remainder, and thus the equality holds to second order. Recall that $\boldsymbol{u}_{\mathrm{L}} = \boldsymbol{d}_u^{(l)}(s^{(l)};\boldsymbol{\delta}^{(l)}) = \dot{\boldsymbol{B}}^{(l)}(s^{(l)})\boldsymbol{\delta}^{(l)}$ and $\boldsymbol{u}_{\mathrm{R}} = \boldsymbol{d}_u^{(l+1)}(s^{(l)};\boldsymbol{\delta}^{(l+1)}) = \dot{\boldsymbol{B}}^{(l+1)}(s^{(l)})\boldsymbol{\delta}^{(l+1)}$. We can then write

$$|\boldsymbol{t}_{\mathrm{R}}|\boldsymbol{u}_{\mathrm{L}} - |\boldsymbol{t}_{\mathrm{L}}|\boldsymbol{u}_{\mathrm{R}} = \begin{bmatrix}|\boldsymbol{t}_{\mathrm{R}}|\dot{\boldsymbol{B}}^{(l)}(s^{(l)}) & -|\boldsymbol{t}_{\mathrm{L}}|\dot{\boldsymbol{B}}^{(l+1)}(s^{(l)})\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}^{(l)}\\ \boldsymbol{\delta}^{(l+1)}\end{bmatrix} \quad (4.24)$$

$$= \boldsymbol{V}^{(l)}\begin{bmatrix}\boldsymbol{\delta}^{(l)}\\ \boldsymbol{\delta}^{(l+1)}\end{bmatrix} = \begin{bmatrix}0 & \cdots & 0 & \boldsymbol{V}^{(l)} & 0 & \cdots & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}^{(1)}\\ \vdots\\ \boldsymbol{\delta}^{(l)}\\ \boldsymbol{\delta}^{(l+1)}\\ \vdots\\ \boldsymbol{\delta}^{(L)}\end{bmatrix} = \tilde{\boldsymbol{V}}^{(l)}\boldsymbol{\delta}.$$

Let us now rewrite equation (4.23) in terms of (4.24) to get

$$\boldsymbol{\delta}^T\tilde{\boldsymbol{V}}^{(l)^T}\boldsymbol{U}^{(l)}\tilde{\boldsymbol{V}}^{(l)}\boldsymbol{\delta} = \boldsymbol{\delta}^T\tilde{\boldsymbol{U}}^{(l)}\boldsymbol{\delta} = 0,$$

which is the approximate G^1 constraint for the lth joint of the deformed curve. If the set of G^1 joints is denoted by

$$\mathcal{I}_1 = \{l \in \{1, \ldots, L\} : \boldsymbol{c}(s^{(l)}) \text{ is a } G^1 \text{ joint.}\},$$

then the G^1 constraints

$$\boldsymbol{\delta}^T \tilde{\boldsymbol{U}}^{(l)} \boldsymbol{\delta} = 0, \quad l \in \mathcal{I}_1 \tag{4.25}$$

are equivalent to the constraint

$$\boldsymbol{\delta}^T \boldsymbol{G}_1 \boldsymbol{\delta} = 0, \quad \boldsymbol{G}_1 = \sum_{l \in \mathcal{I}_1} \tilde{\boldsymbol{U}}^{(l)}, \tag{4.26}$$

where $\boldsymbol{G}_1$ is independent of the deformation parameters $\boldsymbol{\delta}$. That constraint (4.25) implies (4.26) is trivial. The reverse implication follows by the semi-positive definiteness of the matrices $\tilde{\boldsymbol{U}}^{(l)}$, $l \in \mathcal{I}_1$.

4.10 Regularisation Using A Priori Information

Until now we have only considered the energy of the deformation in terms of the distance from the measurements to their projections in the deformed object $\mathcal{M}_{\boldsymbol{d}}$. We have not taken the initial form of the ideal model $\mathcal{M}$ into consideration but have only focused on getting the best match possible given the degrees freedom of the deformation.

However, there is a need for a mechanism that tries to give weight to the nominal form $\mathcal{M}$. This is particularly true when the number of measurements of the physical object $\mathcal{P}$ is not sufficient to make the minimisation problem well-posed. This case can occur if the underlying, nominal model is complex and consists of many arcs. Since we have assumed that the deformation consists of the same number of arcs as the underlying curve it might be that the total number of degrees of freedom (dof) of the deformation exceeds the number of available measurements although the number of dof of each arc might be low.

Another case is when the number of measurements equal the number of dof of the deformation. In such a case, measurement data will be interpolated. Interpolation is something that we definitely want to avoid since the measurements are polluted by measurement noise that we want to filter out. In this case we can hold back the match of the deformation to data by putting more emphasis on the nominal form.

The idea is not only to consider the measurements but also to use *a priori* information about the shape of the object given by the nominal model. Thus, we restrict the deformation space $\mathcal{D}$ to a neighbourhood of the nominal model. In practice, we know that the deformations we are looking for are small. The limitation on the deformation space makes the optimisation problem easier to solve since the set of plausible solutions is smaller. Nevertheless, immediately another question arises; what kind restriction should be put on the deformation space and to what extent? This section tries to answer the first of these two questions by

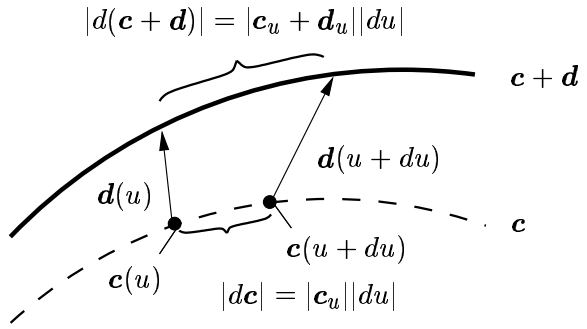

Figure 4.3: The figure shows an infinitesimal curve element $|d\boldsymbol{c}|$ and its deformed counterpart $|d(\boldsymbol{c}+\boldsymbol{d})|$.

deriving inequality constraints for the length change, shape change and joint angle change.

4.10.1 Length Preservation

In order to constrain the length of the deformed curve we introduce a functional that measures the change of length of the deformed curve $\mathcal{M}_{\boldsymbol{d}}$ relative to $\mathcal{M}$. Consider an infinitesimal distance $|d\boldsymbol{c}^{(l)}| = |\boldsymbol{c}_u^{(l)}||du|$ along the lth arc $\boldsymbol{c}^{(l)}$ and the corresponding distance $|d(\boldsymbol{c}^{(l)} + \boldsymbol{d}^{(l)})| = |\boldsymbol{c}_u^{(l)} + \boldsymbol{d}_u^{(l)}||du|$ along the deformed arc $\boldsymbol{c}^{(l)} + \boldsymbol{d}^{(l)}$, see Figure 4.3. The change in length of the deformation relative to the original curve can be expressed as

$$\frac{\big||d(\boldsymbol{c}^{(l)} + \boldsymbol{d}^{(l)})| - |d\boldsymbol{c}^{(l)}|\big|}{|d\boldsymbol{c}^{(l)}|} = \frac{\big||\boldsymbol{c}_u^{(l)} + \boldsymbol{d}_u^{(l)}| - |\boldsymbol{c}_u^{(l)}|\big|}{|\boldsymbol{c}_u^{(l)}|} \leq \frac{|\boldsymbol{c}_u^{(l)} + \boldsymbol{d}_u^{(l)} - \boldsymbol{c}_u^{(l)}|}{|\boldsymbol{c}_u^{(l)}|} = \frac{|\boldsymbol{d}_u^{(l)}|}{|\boldsymbol{c}_u^{(l)}|} \tag{4.27}$$

The non-smoothness of the quotient (4.27) can be handled by considering the squared relative change of length instead. Integration of the square of (4.27) over the parameter interval $I^{(l)}$ gives a measure of the change of length of the deformed object $\mathcal{M}_{\boldsymbol{d}}$ relative to its nominal form $\mathcal{M}$. A functional measuring the length discrepancy of the deformed curve along the lth arc is

$$F_{\text{length}}(\boldsymbol{d}^{(l)}; \boldsymbol{c}^{(l)}) = \int_{I^{(l)}} \frac{|\boldsymbol{d}_u^{(l)}|^2}{|\boldsymbol{c}_u^{(l)}|^2} |\boldsymbol{c}_u^{(l)}| du = \int_{I^{(l)}} \frac{|\boldsymbol{d}_u^{(l)}|^2}{|\boldsymbol{c}_u^{(l)}|} du. \tag{4.28}$$

Assuming $\boldsymbol{d}^{(l)}$ to be of the form $\boldsymbol{d}^{(l)}(u) = \boldsymbol{B}^{(l)}(u)\boldsymbol{\delta}^{(l)}$, then (4.28) is seen to be the quadratic form

$$\begin{aligned} F_{\text{length}}(\boldsymbol{d}^{(l)}; \boldsymbol{c}^{(l)}) &= \int_{I^{(l)}} {\boldsymbol{\delta}^{(l)}}^T \boldsymbol{B}^{(l)}(u)^T \boldsymbol{B}^{(l)}(u) \boldsymbol{\delta}^{(l)} |\boldsymbol{c}_u^{(l)}| du \\ &= {\boldsymbol{\delta}^{(l)}}^T \Big(\int_{I^{(l)}} \boldsymbol{B}^{(l)}(u)^T \boldsymbol{B}^{(l)}(u) |\boldsymbol{c}_u^{(l)}| du \Big) \boldsymbol{\delta}^{(l)} = {\boldsymbol{\delta}^{(l)}}^T \boldsymbol{V}^{(l)} \boldsymbol{\delta}^{(l)}. \end{aligned}$$

Let $\gamma^{(l)}$ denote the nominal length of arc l. It can be computed by $\gamma^{(l)} = F_{\text{length}}(\boldsymbol{c}^{(l)}; \boldsymbol{c}^{(l)})$. Suppose now we want to constrain the length change of the deformed curve to be less than one per cent, say. Then, this can approximately be accomplished by letting $v = 0.01$ and requiring that

$$\boldsymbol{\delta}^{(l)^T} \boldsymbol{V}^{(l)} \boldsymbol{\delta}^{(l)} \leq v\gamma^{(l)}, \quad \text{for all } l = 1, \ldots, L.$$

If a global relative length constraint is sufficient we can relax the constraints by adding them together and simply require that

$$\gamma^{-1} \sum_{l=1}^{L} \boldsymbol{\delta}^{(l)^T} \boldsymbol{V}^{(l)} \boldsymbol{\delta}^{(l)} = \gamma^{-1} \begin{bmatrix} \boldsymbol{\delta}^{(1)} \\ \vdots \\ \boldsymbol{\delta}^{(L)} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{V}^{(1)} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{V}^{(L)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}^{(1)} \\ \vdots \\ \boldsymbol{\delta}^{(L)} \end{bmatrix} = \boldsymbol{\delta}^T \boldsymbol{V} \boldsymbol{\delta} \leq v, \tag{4.29}$$

where γ is the length of the whole of $\mathcal{M}$, i.e. $\gamma = \sum_{l=1}^{L} \gamma^{(l)}$.

4.10.2 Shape Preservation

The difference in shape between the nominal arc $\boldsymbol{c}^{(l)}$ and the deformed arc $\boldsymbol{c}^{(l)} + \boldsymbol{d}^{(l)}$ can be measured by comparing the difference in curvature between the two. Following Wesselink (1996) we introduce the *smoothness functional*

$$F_{\text{shape}}(\boldsymbol{e}) = \int_{I^{(l)}} \kappa_{\boldsymbol{e}}^2 |\boldsymbol{e}_u| du, \tag{4.30}$$

where $\kappa_{\boldsymbol{e}}$ is the curvature of the curve $\boldsymbol{e}$ defined on $I^{(l)}$. Suppose we want to measure the shape difference of the curve $\boldsymbol{e}$ relative to $\boldsymbol{c}^{(l)}$. We assume $\boldsymbol{e}$ to be close to $\boldsymbol{c}^{(l)}$ in shape. In order to do this we introduce the Laplace-Beltrami operator $\Delta_{\boldsymbol{c}^{(l)}}(\boldsymbol{e})$ defined on the arc $\boldsymbol{c}^{(l)}$.

$$\Delta_{\boldsymbol{c}^{(l)}}(\boldsymbol{e}) = \frac{\boldsymbol{e}_{uu}}{|\boldsymbol{c}_u^{(l)}|^2} - \frac{(\boldsymbol{c}_u^{(l)^T} \boldsymbol{c}_{uu}^{(l)}) \boldsymbol{e}_u}{|\boldsymbol{c}_u^{(l)}|^4}.$$

In particular $\kappa_{\boldsymbol{c}^{(l)}} = |\Delta_{\boldsymbol{c}^{(l)}}(\boldsymbol{c}^{(l)})|$. According to Wesselink (1996) we can approximate (4.30) by

$$\hat{F}_{\text{shape}}(\boldsymbol{e}; \boldsymbol{c}^{(l)}) = \int_{I^{(l)}} |\Delta_{\boldsymbol{c}^{(l)}}(\boldsymbol{e})|^2 |\boldsymbol{c}_u^{(l)}| du, \tag{4.31}$$

as long as $\boldsymbol{e}$ is close to $\boldsymbol{c}^{(l)}$. The approximation is trivially exact for $\boldsymbol{e} = \boldsymbol{c}^{(l)}$, i.e. $F_{\text{shape}}(\boldsymbol{c}^{(l)}) = \hat{F}_{\text{shape}}(\boldsymbol{c}^{(l)}; \boldsymbol{c}^{(l)})$.

We can use this result to measure the smoothness of the deformed curve by letting $\boldsymbol{e} = \boldsymbol{c}^{(l)} + \boldsymbol{d}^{(l)}$ in (4.31). However, this is not exactly what we want. We only want to measure the difference in shape of $\boldsymbol{c}^{(l)} + \boldsymbol{d}^{(l)}$ relative to $\boldsymbol{c}^{(l)}$. In order to do this we measure the relative curvature of $\boldsymbol{d}^{(l)}$ with respect to $\boldsymbol{c}^{(l)}$ using the

Laplace-Beltrami operator $\Delta_{\boldsymbol{c}^{(l)}}(\boldsymbol{d}^{(l)})$. The deformation measure is then taken to be the integral

$$\hat{F}_{\text{shape}}(\boldsymbol{d}^{(l)};\boldsymbol{c}^{(l)}) = \int_{I^{(l)}} |\Delta_{\boldsymbol{c}^{(l)}}(\boldsymbol{d}^{(l)})|^2 |\boldsymbol{c}_u^{(l)}| du \tag{4.32}$$

The functional (4.32) turns out to be a quadratic form in the deformation parameters $\boldsymbol{\delta}^{(l)}$ if $\boldsymbol{d}^{(l)}$ is assumed linear in $\boldsymbol{\delta}^{(l)}$. Thus

$$\begin{aligned}\hat{F}_{\text{shape}}(\boldsymbol{d}^{(l)};\boldsymbol{c}^{(l)}) &= \int_{I^{(l)}} \left| \frac{\ddot{\boldsymbol{B}}^{(l)}\boldsymbol{\delta}^{(l)}}{|\boldsymbol{c}_u^{(l)}|^2} - \frac{(\boldsymbol{c}_u^{(l)^T}\boldsymbol{c}_{uu}^{(l)})\dot{\boldsymbol{B}}^{(l)}\boldsymbol{\delta}^{(l)}}{|\boldsymbol{c}_u^{(l)}|^4} \right|^2 |\boldsymbol{c}_u^{(l)}| du \\ &= \boldsymbol{\delta}^{(l)^T} \left(\int_I \Big(\frac{\ddot{\boldsymbol{B}}^{(l)}}{|\boldsymbol{c}_u^{(l)}|^2} - \frac{(\boldsymbol{c}_u^{(l)^T}\boldsymbol{c}_{uu}^{(l)})\dot{\boldsymbol{B}}^{(l)}}{|\boldsymbol{c}_u^{(l)}|^4} \Big)^T \Big(\frac{\ddot{\boldsymbol{B}}^{(l)}}{|\boldsymbol{c}_u^{(l)}|^2} - \frac{(\boldsymbol{c}_u^{(l)^T}\boldsymbol{c}_{uu}^{(l)})\dot{\boldsymbol{B}}^{(l)}}{|\boldsymbol{c}_u^{(l)}|^4} \Big) |\boldsymbol{c}_u^{(l)}| du \right) \boldsymbol{\delta}^{(l)} \\ &= \boldsymbol{\delta}^{(l)^T} \boldsymbol{W}^{(l)} \boldsymbol{\delta}. \end{aligned} \tag{4.33}$$

The total shape change of the deformed arc $\boldsymbol{c}^{(l)} + \boldsymbol{d}^{(l)}$ can thus be restricted by constraining (4.33) divided by the length of the lth arc $\gamma^{(l)}$ in order to constrain the shape change per unit length. More precisely, require that

$$\boldsymbol{\delta}^{(l)^T} \boldsymbol{W}^{(l)} \boldsymbol{\delta} \leq w^{(l)} \gamma^{(l)}, \quad \text{for all } l = 1, \ldots, L. \tag{4.34}$$

It is sufficient to constrain the quadratic form from above since the matrix $\boldsymbol{W}^{(l)}$ is semi-positive definite. Instead of having L constraints of the form (4.34) we can relax them by incorporating them into one single constraint whose sum is restricted from above. To put it mathematically we have that

$$\gamma^{-1} \sum_{l=1}^{L} \boldsymbol{\delta}^{(l)^T} \boldsymbol{W}^{(l)} \boldsymbol{\delta}^{(l)} = \gamma^{-1} \begin{bmatrix} \boldsymbol{\delta}^{(1)} \\ \vdots \\ \boldsymbol{\delta}^{(L)} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{W}^{(1)} & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{W}^{(L)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}^{(1)} \\ \vdots \\ \boldsymbol{\delta}^{(L)} \end{bmatrix} = \boldsymbol{\delta}^T \boldsymbol{W} \boldsymbol{\delta} \leq w, \tag{4.35}$$

where $\boldsymbol{W}$ is independent of the deformation parameters $\boldsymbol{\delta}$. It is difficult to give a quantitative interpretation of the constraint parameter w in (4.35) and $w^{(l)}$ in (4.34). Nevertheless, they are very important in controlling the bending energy of the deformation.

4.10.3 Joint Angle Preservation

The third and last property of the deformed curve that we want to be able to constrain is the difference in joint angle of the deformed curve relative to the nominal. In some sense this corresponds to restricting the bending energy of the joint. Of course, this constraint does not apply to the G^1 joints whose angle are not allowed to deviate at all, see Section 4.9.2.

In practice, the true joint angle difference of a joint relative to the nominal model can be assumed small, typically within five degrees, say. Of course, the

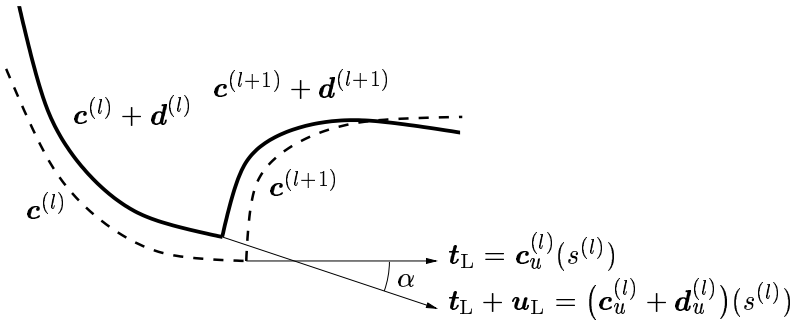

Figure 4.4: An illustration of the joint angle difference between the nominal and the deformed curve.

maximum expected joint angle difference will depend on the manufacturing process through which the workpiece under consideration is produced. The aim of this section is to derive constraints so that this physical limitation of the joint angle difference is reflected in our geometric model of the deformation. A side-effect of the constraint is of mathematical importance in that the set of possible solutions to the minimisation problem is restricted.

Mathematically, the angle difference of a joint can be constrained by considering the left and right limits of the tangents of the nominal and deformed curve separately, see Figure 4.4 for an illustration.

Joint Angle Mathematics.

The angle α in Figure 4.4 can be computed through the relation

$$\boldsymbol{t}_{\mathrm{L}}^T(\boldsymbol{t}_{\mathrm{L}} + \boldsymbol{u}_{\mathrm{L}}) = |\boldsymbol{t}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{L}} + \boldsymbol{u}_{\mathrm{L}}| \cos\alpha, \tag{4.36}$$

where $\boldsymbol{t}_{\mathrm{L}}$ and $\boldsymbol{u}_{\mathrm{L}}$ are defined in Section 4.9.2. Our aim is to simplify this non-linear condition and derive an explicit inequality constraint for the joint angle difference.

A second order expansion of $\cos\alpha$ around the origin is $\cos\alpha = 1 - \alpha^2/2 + o(\alpha^2)$ and substitution in (4.36) gives after rearrangement

$$\frac{|\boldsymbol{t}_{\mathrm{L}} + \boldsymbol{u}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{L}}| - \boldsymbol{t}_{\mathrm{L}}^T(\boldsymbol{t}_{\mathrm{L}} + \boldsymbol{u}_{\mathrm{L}})}{|\boldsymbol{t}_{\mathrm{L}} + \boldsymbol{u}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{L}}|} = \frac{\alpha^2}{2} + o(\alpha^2). \tag{4.37}$$

Expanding $|\boldsymbol{t}_{\mathrm{L}} + \boldsymbol{u}_{\mathrm{L}}|$ with respect to $\boldsymbol{u}_{\mathrm{L}}$ around the origin (see equation (B.1)) we see that the numerator of (4.37) can be approximated to second order by

$$\begin{aligned} |\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{L}}|-\boldsymbol{t}_{\mathrm{L}}^T(\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}}) &= \left(|\boldsymbol{t}_{\mathrm{L}}|+\frac{\boldsymbol{t}_{\mathrm{L}}^T\boldsymbol{u}_{\mathrm{L}}}{|\boldsymbol{t}_{\mathrm{L}}|}+\frac{1}{2|\boldsymbol{t}_{\mathrm{L}}|}\boldsymbol{u}_{\mathrm{L}}^T\Big(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{L}}^T}{|\boldsymbol{t}_{\mathrm{L}}|^2}\Big)\boldsymbol{u}_{\mathrm{L}}+o\big(|\boldsymbol{u}_{\mathrm{L}}|^2\big)\right)|\boldsymbol{t}_{\mathrm{L}}| \\ -\boldsymbol{t}_{\mathrm{L}}^T(\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}}) &= \frac{1}{2}\boldsymbol{u}_{\mathrm{L}}^T\Big(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{L}}^T}{|\boldsymbol{t}_{\mathrm{L}}|^2}\Big)\boldsymbol{u}_{\mathrm{L}}+o\big(|\boldsymbol{u}_{\mathrm{L}}|^2\big). \end{aligned} \tag{4.38}$$

To get a second order approximation of the left-hand side of (4.37) is suffices to expand its denominator to zeroth order since the numerator appears to be of second order judging from (4.38). Hence the denominator is expanded to

$$\frac{1}{|\boldsymbol{t}_{\mathrm{L}}+\boldsymbol{u}_{\mathrm{L}}||\boldsymbol{t}_{\mathrm{L}}|}=\Big(\frac{1}{|\boldsymbol{t}_{\mathrm{L}}|}+o(1)\Big)\frac{1}{|\boldsymbol{t}_{\mathrm{L}}|}=\frac{1}{|\boldsymbol{t}_{\mathrm{L}}|^2}+o(1), \tag{4.39}$$

according to equation (B.2). Using (4.38) and (4.39) a second order approximation to (4.37) is

$$\frac{1}{|\boldsymbol{t}_{\mathrm{L}}|^2}\boldsymbol{u}_{\mathrm{L}}^T\left(\boldsymbol{I}-\frac{\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{L}}^T}{|\boldsymbol{t}_{\mathrm{L}}|^2}\right)\boldsymbol{u}_{\mathrm{L}}=\alpha^2+o\big(\alpha^2+|\boldsymbol{u}_{\mathrm{L}}|^2\big).$$

Recall that $\boldsymbol{u}_{\mathrm{L}}=\dot{\boldsymbol{B}}^{(l)}(s^{(l)})\boldsymbol{\delta}^{(l)}$ and define the matrix $\tilde{\boldsymbol{X}}^{(l)}_{\text{left}}=|\boldsymbol{t}_{\mathrm{L}}|^{-2}(\boldsymbol{I}-|\boldsymbol{t}_{\mathrm{L}}|^{-2}\boldsymbol{t}_{\mathrm{L}}\boldsymbol{t}_{\mathrm{L}}^T)$. Then the angle α is approximately given by

$$\boldsymbol{\delta}^{(l)^T}\dot{\boldsymbol{B}}^{(l)^T}(s^{(l)})\tilde{\boldsymbol{X}}_{\text{left}}\dot{\boldsymbol{B}}^{(l)}(s^{(l)})\boldsymbol{\delta}=\boldsymbol{\delta}^{(l)^T}\boldsymbol{X}^{(l)}_{\text{left}}\boldsymbol{\delta}^{(l)}\approx\alpha^2. \tag{4.40}$$

The angle corresponding to the right limit of the deformed curve tangent is denoted by β and satisfies the similar relation

$$\boldsymbol{\delta}^{(l+1)^T}\boldsymbol{X}^{(l)}_{\text{right}}\boldsymbol{\delta}^{(l+1)}\approx\beta^2. \tag{4.41}$$

Suppose now that we want the total joint angle difference of joint l to be limited by some angle $\Delta\theta^{(l)}$, i.e. that

$$|\alpha+\beta|\leq\Delta\theta^{(l)}.$$

Then we have a problem in that we do not know the signs of α or β. Moreover, the derived expressions for them determine the square of the deformed angle. However, observing that[2] $|\alpha+\beta|^2\leq 2(\alpha^2+\beta^2)$, it is clear that a more restrictive condition on the deformed joint angle is

$$2(\alpha^2+\beta^2)\leq\big(\Delta\theta^{(l)}\big)^2. \tag{4.42}$$

On the other hand, this condition is easier to compute. A matrix version of the constraint is achieved by formulating it in terms of (4.40) and (4.41), i.e.

$$\begin{bmatrix}\boldsymbol{\delta}^{(l)}\\ \boldsymbol{\delta}^{(l+1)}\end{bmatrix}^T\begin{bmatrix}2\boldsymbol{X}^{(l)}_{\text{left}} & 0\\ 0 & 2\boldsymbol{X}^{(l)}_{\text{right}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta}^{(l)}\\ \boldsymbol{\delta}^{(l+1)}\end{bmatrix}\leq\big(\Delta\theta^{(l)}\big)^2.$$

[2]Follows from $0\leq|\alpha+\beta|^2$ by which we see that $-2\alpha\beta\leq 2|\alpha\beta|\leq\alpha^2+\beta^2$. Now, expanding the square once again we have $|\alpha+\beta|^2\leq\alpha^2+\beta^2+2\alpha\beta\leq\alpha^2+\beta^2+2|\alpha\beta|\leq 2(\alpha^2+\beta^2)$ which concludes the proof. □

Alternatively, the condition can be formulated in terms of all deformation parameters, i.e.

$$\begin{bmatrix} \vdots \\ \boldsymbol{\delta}^{(l)} \\ \boldsymbol{\delta}^{(l+1)} \\ \vdots \end{bmatrix}^T \begin{bmatrix} \ddots & & & & & & 0 \\ & 0 & & & & & \\ & & 2\boldsymbol{X}^{(l)}_{\text{left}} & & & & \\ & & & & 2\boldsymbol{X}^{(l)}_{\text{right}} & & \\ & & & & & 0 & \\ 0 & & & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{\delta}^{(l)} \\ \boldsymbol{\delta}^{(l+1)} \\ \vdots \end{bmatrix} = \boldsymbol{\delta}^T \boldsymbol{X}^{(l)} \boldsymbol{\delta} \leq (\Delta\theta^{(l)})^2 \tag{4.43}$$

Why Two Conditions for G^1 Continuity?

Imagine that we use equation (4.43) to impose G^1 continuity on the deformation. How is this different from the other G^1 condition (4.26)?

Consider the lth joint of the curve as depicted in Figure 4.4 and suppose that we try to make the deformation $\boldsymbol{d}$ tangent continuous by letting $\Delta\theta^{(l)} = 0$ in (4.43). In the present situation the joint angle constraint is modelled by constraining the squared, tangential angle differences α^2 and β^2. Hence letting $\Delta\theta^{(l)}$ to zero will force the sub-angles α and β to zero as well. Apart from achieving G^1 continuity of $\boldsymbol{d}$, the constraint will disallow rotation of the joint, an undesirable property. See Figure 4.2 for an example of a situation that we want to allow, that would not be allowed using (4.43) with $\Delta\theta^{(l)} = 0$.

4.10.4 Summary of Regularisation

We have derived constraints that can be used to include information about length, shape and joint angles of the nominal model. They take the form

$$\boldsymbol{\delta}^T \boldsymbol{V} \boldsymbol{\delta} \leq v, \qquad \text{(length)} \tag{4.44}$$

$$\boldsymbol{\delta}^T \boldsymbol{W} \boldsymbol{\delta} \leq w, \qquad \text{(shape)} \tag{4.45}$$

$$\boldsymbol{\delta}^T \boldsymbol{X}^{(l)} \boldsymbol{\delta} \leq (\Delta\theta^{(l)})^2, \quad l \in \mathcal{I}_1^C, \qquad \text{(joint angle).} \tag{4.46}$$

The constraints are in block matrix form. It is likely that the problem can be solved efficiently by exploiting the block form in a solution algorithm. However, methods of optimisation are not within the scope of this study where we are mostly concerned with the geometric model and its implications.

Relating the Constraints to Each Other

Is there a close relationship between the constraints (4.44)–(4.46)? These constraints add parameters to the optimisation problem, parameters that has to be set somehow. It would be highly advantageous if we can combine the three types of constraints to reduce the number of variables. In particular, the second and third constraint type are special in that both of them measure bending energy. The

second constraint (4.45) measures the bending energy in terms of the curvature of the deformation on each arc, and (4.46) measures the joint angle difference. The latter can be seen as discrete curvature. However, as of today, this issue is yet to be dealt with.

4.11 Restricting the Deformation Space

In Sections 4.4, 4.9 and 4.10 we derived constraints on the deformation parameters. With the help of these constraints we can restrict the deformation parameter space. In particular, let $\Omega = \Omega(v, w, \Delta\theta)$ be defined as

$$\Omega(v, w, \Delta\theta) = \big\{\boldsymbol{\delta} \in \mathbb{R}^{3(n_{\boldsymbol{d}}^{(1)}+\cdots+n_{\boldsymbol{d}}^{(L)})} : \boldsymbol{S}\boldsymbol{\delta} = 0,\ \boldsymbol{G}_0\boldsymbol{\delta} = 0,\ \boldsymbol{\delta}^T\boldsymbol{G}_1\boldsymbol{\delta} = 0, \dots \\ \dots\ \boldsymbol{\delta}^T\boldsymbol{V}\boldsymbol{\delta} \le v,\ \boldsymbol{\delta}^T\boldsymbol{W}\boldsymbol{\delta} \le w,\ \boldsymbol{\delta}^T\boldsymbol{X}^{(l)}\boldsymbol{\delta} \le (\Delta\theta)^2 \text{ for all } l \in \mathcal{I}_1^C\big\}.$$

The type of separation matrix is here left unspecified. It can be either $\boldsymbol{S}_{\text{eq}}$ or $\boldsymbol{S}_{\text{ds}}$. For simplicity we have chosen to use the global versions of the length and shape constraints, i.e. equations (4.29) and (4.35), respectively. Moreover, the joint angles are constrained by the same angle. The deformation is then required to be in the set

$$\mathcal{D} = \big\{\boldsymbol{d} \in C(I) \cap \big(\cup_{l=1}^{L} C^2(I^{(l)})\big) :\ \boldsymbol{d} : I \times \Omega(v, w, \Delta\theta) \to \mathbb{R}^3, \dots \\ \dots\ \boldsymbol{d}(u; \boldsymbol{\delta}) = \boldsymbol{d}^{(l)}(u; \boldsymbol{\delta}^{(l)}) = \boldsymbol{B}^{(l)}(u)\boldsymbol{\delta}^{(l)},\ u \in I^{(l)}\big\},$$

where $\boldsymbol{B}^{(l)}$ are matrices consisting of the deformation bases for each arc. The notation presented is somewhat misleading since the parameter space Ω does not appear to depend on the deformation space $\mathcal{D}$. However, the constraints defining Ω are defined in terms of the bases $\boldsymbol{B}^{(l)}$, $l = 1, \dots, L$ and are thus dependent on the representation of $\boldsymbol{d}$.

To conclude, given a model $\mathcal{M}$ and a set of measurements $\boldsymbol{p}_j$ $j = 1, \dots, n$ of a real workpiece $\mathcal{P}$, find the deformation $\boldsymbol{d} \in \mathcal{D}$ of $\mathcal{P}$ by solving

$$\min_{\boldsymbol{\delta} \in \Omega(v,w,\Delta\theta)} \psi(\boldsymbol{\delta}).$$

Recall that each evaluation of ψ requires the solution of a rigid best fit problem to find $\boldsymbol{x}(\boldsymbol{\delta})$. Moreover, each evaluation of the rigid best fit problem requires the solution of n least distance problems to find $u_j(\boldsymbol{x}, \boldsymbol{\delta})$.

4.11.1 A Discussion on Parameter Selection

How can we assign adequate values to the parameters? Except for the dof of the deformation $3(n_{\boldsymbol{d}}^{(1)} + \cdots + n_{\boldsymbol{d}}^{(L)})$ we have introduced constraints with parameters $(v, w, \Delta\theta)$ on the parameter space Ω that have to be set. The number of dof of the deformation together with the newly introduced regularisation parameters control the flexibility of the deformation. In general, the deformation should be flexible enough to incorporate all of the form error, but should at the same time be sufficiently stiff to filter out the measurement noise. This section discusses the regularisation parameters, one by one.

Length Constraint

The length restriction is the easiest parameter to deal with. The length change has to be restricted due to the lack of control in the tangential direction of the curve. By the definition of the residuals, the measurements only control the deformed curve in its normal direction. Recall that the target point of a measurement is always taken to be the least distance projection of the transformed point on the deformed curve. The length constraint has a very clear quantitative interpretation; the length of the deformed curve is only allowed to differ from its nominal length by $100v\%$. A typical value of the length constraint is $v = 0.01$.

Joint Angle Constraint

Also the joint angle constraint is easily understood. It is easy to imagine what it means for a workpiece when its joint angles differ from nominal. Physically, a joint angle difference is due to incorrect bending of a sheet metal part or to inaccurate machining of a solid part during the manufacturing process. Therefore, there will be some physical limitations of the plausible size of the joint angle difference can take. In general, the joint angle difference can be expected to lie within five degrees, say.

Shape Constraint vs. Deformation Dof

The most difficult parameters to select are the shape constraint and the deformation dof. The dof of the deformation should, of course, be significantly lower than the number of measurements for the optimisation problem to be well-posed. Nevertheless, using the regularisation constraints we can solve under-constrained problems as well.

Consider for a moment the lth arc of the curve $\boldsymbol{c}^{(l)}$ and its deformation $\boldsymbol{d}^{(l)}$. There seems to be a close relationship between the deformation degree of the arc $n_{\boldsymbol{d}}^{(l)}$ and the shape constraint parameter $w^{(l)}$. For a fixed value of the degree of freedom of the arc $n_{\boldsymbol{d}}^{(l)}$ the deformation is becoming increasingly stiff as the shape deformation parameter $w^{(l)}$ decreases to zero. Conversely, consider a fixed value of $w^{(l)}$. Then the flexibility of the curve decreases as $n_{\boldsymbol{d}}^{(l)}$ decreases discretely to zero. We have not derived the explicit relationship between these two parameters, though.

Taking a step backwards to consider the whole curve instead the same relationship holds globally. The trouble in the global case is that the composite deformation is connected through joints, not all of them G^1. The joint angles contribute to the same type of flexibility as the arcs, but on a discrete level. An interesting issue would be to try to connect the joint angle constraints to the shape constraint.

4.11.2 The Typical Case

In a typical application, the nominal curve $\boldsymbol{c}$ is a cross-section of a CAD-model. Usually a CAD-model is very complex and consists of hundreds of surface patches

which means that a cross-section is described by dozens of arcs. Considering that each arc is a three-dimensional polynomial of degree four, say, this gives a hint of the number of measurements required to over-determine the deformation. This experiment of thought motivates the introduction of constraints on the parameter space, i.e. the regularisation.

The measurements are usually collected with a CMM or laser scanner. A CMM is a measurement robot that registers discrete positions of a workpiece through a touch probe. Inspection using a CMM is a time-consuming process that only allows a limited number of points to be inspected in practice. The maximum rate is about 50–60 points per minute (Ramsdale, 1999). Laser scanning is another technique gaining in popularity. Then, the measuring is carried out by one or more laser cameras that measure a huge amount of points by a non-contact technique. Dimensional inspection with laser scanners puts higher demands on data analysis as it becomes necessary to process the data in order to reduce the gigantic data set and extract the desired dimensional information.

In the first case, that we denote by *sparse* measuring, the number of measurements does in general not exceed the degree of freedom of the deformation $3(n_{\boldsymbol{d}}^{(1)} + \cdots + n_{\boldsymbol{d}}^{(L)})$ and the deformation parameter space $\Omega(v, w, \Delta\theta)$ has to be constrained so that we can find a unique best deformation. The constraints are defined to represent the nominal geometry, i.e. lowering the bounds on the constraints means that more care is taken to the nominal geometry in the search of the most plausible deformation.

The second case, when we have a *dense* set of measurements, can be treated without much care taken to the nominal geometry since the number of measurements is likely to be well above the deformation dof.

4.12 Example: Bézier Curves

In this section we are more specific about the representation of the deformation. So far, we have not put any assumptions on the representation of the curve $\boldsymbol{c}$ other than that it is required to be regular, continuous and piecewise curvature continuous, i.e. $|\boldsymbol{c}_u(u)| \neq 0, u \in I$ and $\boldsymbol{c} \in C(I) \cap (\cup_{l=1}^{L} C^2(I^{(l)}))$. Almost the same the same restrictions have been put on the deformation $\boldsymbol{d}$, except for the regularity requirement, since it is sufficient that $\boldsymbol{c} + \boldsymbol{d}$ is regular. The deformation itself does not have to be regular.

Nevertheless, in this section we show how the continuity constraints are simplified when the deformation is a Bézier polynomial. We will not be more specific about the nominal curve $\boldsymbol{c}$, though, but instead leave it in its general form.

Let us assume that each arc of the deformation $\boldsymbol{d}^{(l)}$ is described by a polynomial

$$\boldsymbol{d}^{(l)}\big(u; \boldsymbol{\delta}^{(l)}\big) = \sum_{i=0}^{m_l} \bar{\boldsymbol{\delta}}_i^{(l)} B_i^{m_l}\Big(\frac{u - s^{(l-1)}}{s^{(l)} - s^{(l-1)}}\Big), \quad u \in I^{(l)}, \quad \bar{\boldsymbol{\delta}}_i^{(l)} = \begin{bmatrix} \delta_{i_x}^{(l)} \\ \delta_{i_y}^{(l)} \\ \delta_{i_z}^{(l)} \end{bmatrix} \in \mathbb{R}^3, \tag{4.47}$$

where $m_l = n_{\boldsymbol{d}}^{(l)} - 1$, $\boldsymbol{\delta}^{(l)} = [\bar{\boldsymbol{\delta}}_0^{(l)T} \ \bar{\boldsymbol{\delta}}_1^{(l)T} \ \cdots \ \bar{\boldsymbol{\delta}}_{m_l}^{(l)T}]^T \in \mathbb{R}^{3n_{\boldsymbol{d}}^{(l)}}$ and $\{B_i^{m_l}(t)\}_{i=0}^{m_l}$ is a

polynomial basis of degree m_l. If they are the Bernstein polynomials then $\boldsymbol{d}^{(l)}$ is of Bézier type. The Bernstein polynomials are defined by

$$B_i^m(t) = \binom{m}{i} t^i (1-t)^{m-i}.$$

Since a Bézier curve is invariant under affine parameter transformations (Farin, 1997) we can write (4.47) as

$$\boldsymbol{d}^{(l)}\left(u; \boldsymbol{\delta}^{(l)}\right) = \sum_{i=0}^{m_l} \bar{\boldsymbol{\delta}}_i^{(l)} B_i^{m^{(l)}}(u), \quad u \in I^{(l)}.$$

Please note that $\boldsymbol{d}^{(l)}$ is linear in its coefficients $\boldsymbol{\delta}^{(l)}$. We can express (4.47) in matrix form as

$$\boldsymbol{d}^{(l)}(u; \boldsymbol{\delta}^{(l)}) = \underbrace{\begin{bmatrix} B_0^{m_l}(t^{(l)}(u))\boldsymbol{I} & B_1^{m_l}(t^{(l)}(u))\boldsymbol{I} & \cdots & B_{m^{(l)}}^{m_l}(t^{(l)}(u))\boldsymbol{I} \end{bmatrix}}_{=\boldsymbol{B}^{(l)}(u)} \begin{bmatrix} \bar{\boldsymbol{\delta}}_0^{(l)} \\ \bar{\boldsymbol{\delta}}_1^{(l)} \\ \vdots \\ \bar{\boldsymbol{\delta}}_{m_l}^{(l)} \end{bmatrix}$$
$$= \boldsymbol{B}^{(l)}(u)\boldsymbol{\delta}^{(l)},$$

where $t^{(l)} = t^{(l)}(u) = (u - s^{(l-1)})/(s^{(l)} - s^{(l-1)}) \in [0,1]$.

4.12.1 G^0 Continuity in the Bézier Representation

From Section 4.9.1 it is clear that continuity of the lth joint of the deformed curve implies that

$$\begin{bmatrix} \boldsymbol{B}^{(l)}(s^{(l)}) & -\boldsymbol{B}^{(l+1)}(s^{(l)}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}^{(l)} \\ \boldsymbol{\delta}^{(l+1)} \end{bmatrix} = 0.$$

Since $t^{(l)}(s^{(l)}) = 1$ and $t^{(l+1)}(s^{(l)}) = 0$ then $\boldsymbol{B}^{(l)}(s^{(l)}) = \bar{\boldsymbol{\delta}}_{m_l}^{(l)}$ and $\boldsymbol{B}^{(l+1)}(s^{(l)}) = \bar{\boldsymbol{\delta}}_0^{(l+1)}$. Hence the above condition reduces to

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\delta}}_{m_l}^{(l)} \\ \bar{\boldsymbol{\delta}}_0^{(l+1)} \end{bmatrix} = \boldsymbol{I_2} \begin{bmatrix} \bar{\boldsymbol{\delta}}_{m_l}^{(l)} \\ \bar{\boldsymbol{\delta}}_0^{(l+1)} \end{bmatrix} = 0,$$

which we write in matrix form as

$$\begin{bmatrix} \cdots & \boldsymbol{I_2} & \cdots & & & \\ & & \ddots & & & \\ & & & \cdots & \boldsymbol{I_2} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \bar{\boldsymbol{\delta}}_{m_1}^{(1)} \\ \bar{\boldsymbol{\delta}}_0^{(2)} \\ \vdots \\ \bar{\boldsymbol{\delta}}_{m_{L-1}}^{(L-1)} \\ \bar{\boldsymbol{\delta}}_0^{(L)} \\ \vdots \end{bmatrix} = \boldsymbol{G}_0\boldsymbol{\delta} = 0$$

4.12.2 G^1 Continuity in the Bézier Representation

Recall that the left tangent of the deformation at the lth joint is $\boldsymbol{u}_{\mathrm{L}} = \dot{\boldsymbol{B}}^{(l)}(s^{(l)})\boldsymbol{\delta}^{(l)}$ and that the corresponding right tangent is $\boldsymbol{u}_{\mathrm{R}} = \dot{\boldsymbol{B}}^{(l+1)}(s^{(l)})\boldsymbol{\delta}^{(l+1)}$ as defined in Section 4.9.2. When the deformation is represented by a Bézier polynomial then

$$\boldsymbol{u}_{\mathrm{L}} = \dot{\boldsymbol{B}}^{(l)}(s^{(l)})\boldsymbol{\delta}^{(l)} = m_l(\bar{\boldsymbol{\delta}}^{(l)}_{m_l} - \bar{\boldsymbol{\delta}}^{(l)}_{m_l-1}) = -m_l\boldsymbol{I_2}\begin{bmatrix}\bar{\boldsymbol{\delta}}^{(l)}_{m_l-1}\\ \bar{\boldsymbol{\delta}}^{(l)}_{m_l}\end{bmatrix}$$

and similarly

$$\boldsymbol{u}_{\mathrm{R}} = -m_{l+1}\boldsymbol{I_2}\begin{bmatrix}\bar{\boldsymbol{\delta}}^{(l+1)}_{0}\\ \bar{\boldsymbol{\delta}}^{(l+1)}_{1}\end{bmatrix}.$$

In terms of the deformation parameters $\boldsymbol{\delta}$ we can write

$$\begin{aligned}|\boldsymbol{t}_{\mathrm{L}}|\boldsymbol{u}_{\mathrm{R}} - |\boldsymbol{t}_{\mathrm{R}}|\boldsymbol{u}_{\mathrm{L}} &= \underbrace{\boldsymbol{I_2}\begin{bmatrix}m_l|\boldsymbol{t}_{\mathrm{R}}| & 0\\ 0 & m_{l+1}|\boldsymbol{t}_{\mathrm{L}}|\end{bmatrix}\begin{bmatrix}\boldsymbol{I_2} & 0\\ 0 & \boldsymbol{I_2}\end{bmatrix}}_{=\hat{\boldsymbol{V}}^{(l)}}\begin{bmatrix}\bar{\boldsymbol{\delta}}^{(l)}_{m_l-1}\\ \bar{\boldsymbol{\delta}}^{(l)}_{m_l}\\ \bar{\boldsymbol{\delta}}^{(l+1)}_{0}\\ \bar{\boldsymbol{\delta}}^{(l+1)}_{1}\end{bmatrix}\\ &= \underbrace{\begin{bmatrix}0 & \cdots & 0 & \hat{\boldsymbol{V}}^{(l)} & 0 & \cdots & 0\end{bmatrix}}_{=\boldsymbol{V}^{(l)}}\begin{bmatrix}\boldsymbol{\delta}^{(l)}\\ \boldsymbol{\delta}^{(l+1)}\end{bmatrix}\\ &= \begin{bmatrix}0 & \cdots & 0 & \boldsymbol{V}^{(l)} & 0 & \cdots & 0\end{bmatrix}\boldsymbol{\delta} = \tilde{\boldsymbol{V}}^{(l)}\boldsymbol{\delta}.\end{aligned}$$

Recalling that $\boldsymbol{U}^{(l)} = \boldsymbol{I} - \boldsymbol{t}^{(l)T}\boldsymbol{t}^{(l)}$, where $\boldsymbol{t}^{(l)}$ is tangent to the nominal curve at $\boldsymbol{c}(s^{(l)})$ we write

$$\boldsymbol{\delta}^T\tilde{\boldsymbol{V}}^{(l)^T}\boldsymbol{U}^{(l)}\tilde{\boldsymbol{V}}^{(l)}\boldsymbol{\delta} = 0, \quad \text{for all } l \in \mathcal{I}_1.$$

that sums to the final G^1 condition

$$\boldsymbol{\delta}^T\boldsymbol{G}_1\boldsymbol{\delta} = 0.$$

See Section (4.9.2) for the details.

4.12.3 Joint Angle Constraints for Bézier Curves

When $\boldsymbol{d}$ is a piecewise Bézier polynomial then the joint angle constraint can be simplified somewhat. This section presents a such a simplification of the joint angle constraint defined in Section 4.10.3. Just as in the previous section we define $\boldsymbol{u}_{\mathrm{L}} = \dot{\boldsymbol{B}}^{(l)}(s^{(l)}) = -m_l\boldsymbol{I_2}[\bar{\boldsymbol{\delta}}^{(l)T}_{m_l-1}\ \bar{\boldsymbol{\delta}}^{(l)T}_{m_l}]^T$ and we can simplify condition (4.40)

to

$$\begin{bmatrix} \bar{\boldsymbol{\delta}}^{(l)}_{m_l-1} \\ \bar{\boldsymbol{\delta}}^{(l)}_{m_l} \end{bmatrix}^T m_l \boldsymbol{I_2}^T \tilde{\boldsymbol{X}}^{(l)}_{\text{left}} m_l \boldsymbol{I_2} \begin{bmatrix} \bar{\boldsymbol{\delta}}^{(l)}_{m_l-1} \\ \bar{\boldsymbol{\delta}}^{(l)}_{m_l} \end{bmatrix}$$
$$= \boldsymbol{\delta}^{(l)^T} \begin{bmatrix} 0 & \cdots & 0 & m_l \boldsymbol{I_2} & 0 & \cdots & 0 \end{bmatrix}^T \tilde{\boldsymbol{X}}^{(l)}_{\text{left}} \begin{bmatrix} 0 & \cdots & 0 & m_l \boldsymbol{I_2} & 0 & \cdots & 0 \end{bmatrix} \boldsymbol{\delta}^{(l)}$$
$$= \boldsymbol{\delta}^{(l)^T} \boldsymbol{X}^{(l)}_{\text{left}} \boldsymbol{\delta}^{(l)} \approx \alpha^2.$$

The matrix $\boldsymbol{X}^{(l)}_{\text{right}}$ is defined analogously. Substitution in (4.43) gives the desired constraint.

4.12.4 Plane Bézier Curves

When the nominal curve is a cross-section then the curve is plane. Assume that $\boldsymbol{c}$ is a plane curve $\boldsymbol{c}$ with tangent $\boldsymbol{t} = \boldsymbol{c}_u(u_0)$ and well-defined normal $\boldsymbol{n}$ at a point $\boldsymbol{c}(u_0)$ of the curve. The curve then lies in the $\boldsymbol{tn}$-plane and its deformed counterpart is then expected to do the same. Let the vector orthogonal to both $\boldsymbol{t}$ and $\boldsymbol{n}$ be $\boldsymbol{b}$. It can be defined by $\boldsymbol{b} = \boldsymbol{t} \times \boldsymbol{n}$. Since the deformation $\boldsymbol{d}$ is a vector field that is added to $\boldsymbol{c}$ the deformed curve is guaranteed to lie in the same plane as $\boldsymbol{c}$ if $\boldsymbol{d}(u;\boldsymbol{\delta})^T\boldsymbol{b} = 0$ for all $u \in I$.

When each arc is represented by a Bézier polynomial then we have that

$$\boldsymbol{d}^{(l)}(u;\boldsymbol{\delta}^{(l)})^T\boldsymbol{b} = \Big(\sum_{i=1}^{m_l} \bar{\boldsymbol{\delta}}^{(l)}_i B^{m_l}_i\big(t^{(l)}(u)\big)\Big)^T \boldsymbol{b} = \sum_{i=1}^{m_l} \bar{\boldsymbol{\delta}}^{(l)^T}_i \boldsymbol{b} B^{m_l}_i\big(t^{(l)}(u)\big) = 0,$$

when $u \in [s^{(l-1)}, s^{(l)}]$. Since the Bernstein polynomials form a basis, this implies that $\bar{\boldsymbol{\delta}}^{(l)^T}_i \boldsymbol{b} = 0$ for all $i = 0, \ldots, m_l$. In matrix form we can write

$$\boldsymbol{b}^T \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \end{bmatrix} \boldsymbol{\delta} = \boldsymbol{P_b}\boldsymbol{\delta} = 0.$$

4.13 Algorithm

This section outlines an algorithm that solves the flexible best fit problem. As we are not concerned with methods of optimisation in this report we assume the existence of a numerical solver that efficiently solve the problem

$$\begin{aligned} \min_{\Delta \boldsymbol{y}} \frac{1}{2} \Delta\boldsymbol{y}^T \boldsymbol{H} \Delta\boldsymbol{y} + \boldsymbol{g}^T \Delta\boldsymbol{y}, \quad & \text{where } \boldsymbol{y} = \boldsymbol{y}_0 + \Delta\boldsymbol{y} \\ \text{subject to } \boldsymbol{A}\boldsymbol{y} &\le \boldsymbol{a} \\ \boldsymbol{B}\boldsymbol{y} &= \boldsymbol{b} \\ \boldsymbol{y}^T \boldsymbol{C}_i \boldsymbol{y} &\le \boldsymbol{c}_i, \quad i = 1, \ldots, n_{\text{I}} \\ \boldsymbol{y}^T \boldsymbol{D}_j \boldsymbol{y} &= \boldsymbol{d}_j, \quad j = 1, \ldots, n_{\text{E}}, \end{aligned}$$

where $\boldsymbol{y}_0$ is the starting value for the solver. The syntax of the solver is assumed to be

$$\Delta \boldsymbol{y}^* = \mathsf{Solve}(\boldsymbol{H}, \boldsymbol{g}, \boldsymbol{A}, \boldsymbol{a}, \boldsymbol{B}, \boldsymbol{b}, [\boldsymbol{C}_1\ \boldsymbol{C}_2\ \cdots], [\boldsymbol{c}_1\ \boldsymbol{c}_2\ \cdots], [\boldsymbol{D}_1\ \boldsymbol{D}_2\ \cdots], [\boldsymbol{d}_1\ \boldsymbol{d}_2\ \cdots], \boldsymbol{y}_0),$$

where $\Delta \boldsymbol{y}^*$ denotes the solution. The algorithm that uses the Solve() function to solve the flexible best fit problem is outlined in Algorithm 4.1.

Algorithm 4.1 The algorithm computes the flexible best fit of a curve to measurements.

Input: The nominal curve $\boldsymbol{c}$, measurements $\{\boldsymbol{p}_j\}_{j=1}^n$, deformation dof $\{n_{\boldsymbol{d}}^{(l)}\}_{l=1}^L$ and regularisation parameters $(v, w, \Delta\theta)$.

$\mathcal{I}_1 \Leftarrow \{l \in \{1, \ldots, L\} : \boldsymbol{c}(s^{(l)})$ is a G^1 joint.$\}$

$\boldsymbol{S} \Leftarrow$ Separation matrix

$\boldsymbol{G}_0$ and $\boldsymbol{G}_1 \Leftarrow$ Continuity matrices.

$\boldsymbol{V}$, $\boldsymbol{W}$ and $\boldsymbol{X}^{(l)}$, $l \in \mathcal{I}_1^C \Leftarrow$ Regularisation matrices.

if $\boldsymbol{c}$ is a plane curve **then**

  $\boldsymbol{b} \Leftarrow$ The binormal direction.

  $\boldsymbol{P}_{\boldsymbol{b}} \Leftarrow$ Plane curve constraint matrix.

else

  $\boldsymbol{P}_{\boldsymbol{b}} \Leftarrow \emptyset$

end if

Initialise relErrTol

relErr $\Leftarrow$ relErrTol $+ 1$

$\boldsymbol{\delta}^{(0)} \Leftarrow 0$

$k \Leftarrow 0$

while not $0 <$ relErr $<$ relErrTol **do**

  $\boldsymbol{g}^{(k)} \Leftarrow \psi_{\boldsymbol{\delta}}(\boldsymbol{\delta}^{(k)})^T$

  $\boldsymbol{H}^{(k)} \Leftarrow \psi_{\boldsymbol{\delta}\boldsymbol{\delta}}(\boldsymbol{\delta}^{(k)})$

  $\Delta\boldsymbol{\delta}^{(k)} \Leftarrow \mathsf{Solve}(\boldsymbol{H}^{(k)}, \boldsymbol{g}^{(k)}, \emptyset, \emptyset, \begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{G}_0 \\ \boldsymbol{P}_{\boldsymbol{b}} \end{bmatrix}, 0, [\boldsymbol{V}\ \boldsymbol{W}\ \boldsymbol{X}^{(l_1)}\ \cdots], [v\ w\ \Delta\theta\ \cdots], \boldsymbol{G}_1, 0, \boldsymbol{\delta}^{(k)})$

  $\boldsymbol{\delta}^{(k+1)} \Leftarrow \boldsymbol{\delta}^{(k)} + \Delta\boldsymbol{\delta}^{(k)}$

  relErr $\Leftarrow \big(\psi(\boldsymbol{\delta}^{(k)}) - \psi(\boldsymbol{\delta}^{(k+1)})\big) / \big(|\psi(\boldsymbol{\delta}^{(k+1)})| + \mathsf{relErrTol}\big)$

  $k \Leftarrow k + 1$

end while

$\boldsymbol{\delta}^* \Leftarrow \boldsymbol{\delta}^{(k)}$

$\boldsymbol{x}^* \Leftarrow \boldsymbol{x}(\boldsymbol{\delta}^*)$ /* *by Algorithm 2.2 with* $\boldsymbol{c} \Leftarrow \boldsymbol{c} + \boldsymbol{d}(\,\cdot\,; \boldsymbol{\delta}^{(k)})$ */

Output: The optimal deformation parameters $\boldsymbol{\delta}^*$ and the corresponding rigid body transformation parameter $\boldsymbol{x}^*$.

4.14 Examples

In this section we give examples of how the flexible best fit works in practice. The examples are computed using an implementation of Algorithm 4.1 in MATLAB

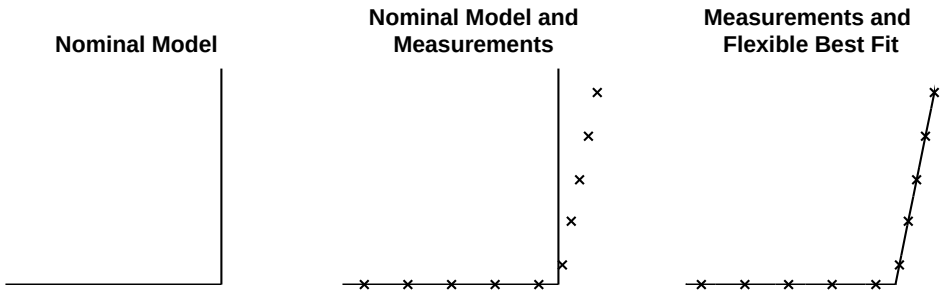

Figure 4.5: A typical example of a flexible best fit. The nominal model is seen to the left and the simulated measurements in the middle. The flexible best fit is shown to the right.

equipped with Version 2 of the Optimization Toolbox. First, we give a series of examples that demonstrate how the different parameters and options affect the result. Second, we show how the flexible best fit performs on a curve representing a cross section of an automotive roof.

4.14.1 The Principle

To illustrate the basic idea of the flexible best fit we have chosen a nominal geometry $\mathcal{M}$ consisting of two straight lines meeting each other in an orthogonal joint. The model, shaped like a mirrored "L", is shown to the left in Figure 4.5. We simulate a set of noise-free measurements to represent a deformed object, whose form we want to recover through the flexible best fit. The set of measurements are shown together with the nominal model in the middle of Figure 4.5.

The nominal model $\mathcal{M}$ and the measurements are input to the flexible best fit algorithm that returns the flexible best fit depicted to the right in Figure 4.5 together with the measurement data. The flexible best fit is obtained through a piecewise linear deformation represented as Bézier polynomials. Since we want the deformed curve to stay continuous we have used the continuity condition (4.18) in the fitting process.

4.14.2 Separation of Form and Positional Error

As discussed in Section 4.4 the flexible best fit allows the form and positional errors to be separated. We introduced two possibilities for separation, namely *equilibrium separation* and *datum separation*.

In order to demonstrate the two possibilities we equipped the nominal geometry of Figure 4.5 with a *datum system* (see Sections 1.2 and 3.1 where this concept is defined). The nominal model $\mathcal{M}$ and its datum system are illustrated to the left in Figure 4.6. Since the curve is restricted to the plane it is sufficient with three datum points and three corresponding guiding directions. Recall that a rigid

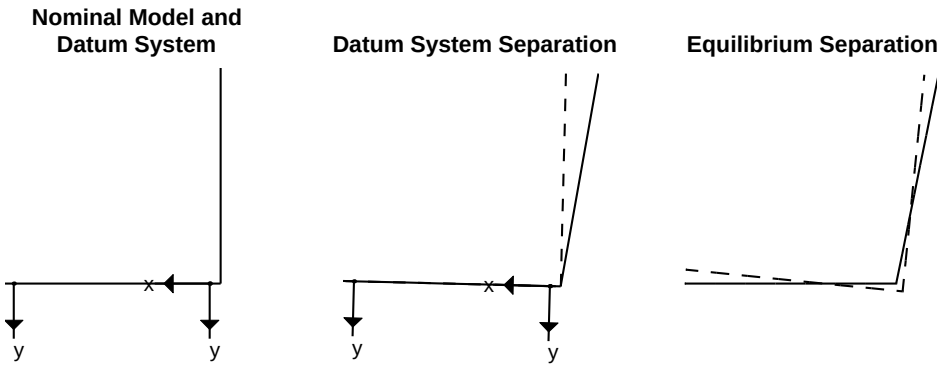

Figure 4.6: Separation methods. To the left is the nominal model and its datum system used to define the datum separation in the middle. The equilibrium separation is seen to the right.

body movement in the plane has three degrees of freedom, two for the translation and one for the orientation. For the part under consideration we have chosen one datum point to guide the part in the y-direction (the leftmost datum point in the current picture) and one datum point to guide the part in both the x and y-directions (the rightmost datum point in the current figure). Recall that the latter datum point is said to have multiplicity two, see Section 1.2.

The picture in the middle of Figure 4.6 shows the flexible best fit to the same set of measurements that we used in Figure 4.5. To make the picture clearer we have chosen not to display the measurements. The flexible best fit is represented by a solid line and the rigid part of the flexible best fit is represented by a dashed line. The deformation is defined relative to the rigid part of the flexible best fit. The position of the rigid part of the flexible best fit is defined by the position of the datum system. Remember that the deformation is required to be zero in the datum points along the guiding directions. The flexible best fit can be thought of as consisting of two steps. In the first step, the flexible deformation is fitted to the measurements. Second, the rigid part of the deformation is defined by the separation condition. When the datum separation is used, as is the case in this example, then the rigid part is located so that the form error is zero in the datum points. However, in reality these two steps are carried out simultaneously since the deformable part of the flexible best fit is defined as a function living on the rigid part.

The equilibrium separation is an alternative to the datum separation. The rightmost picture in Figure 4.6 shows where the nominal geometry is positioned relative to the flexible best fit when the equilibrium separation is applied. Recall that the deformation vector field acting on the nominal geometry satisfies *static equilibrium conditions* when interpreted as a force field, hence the name.

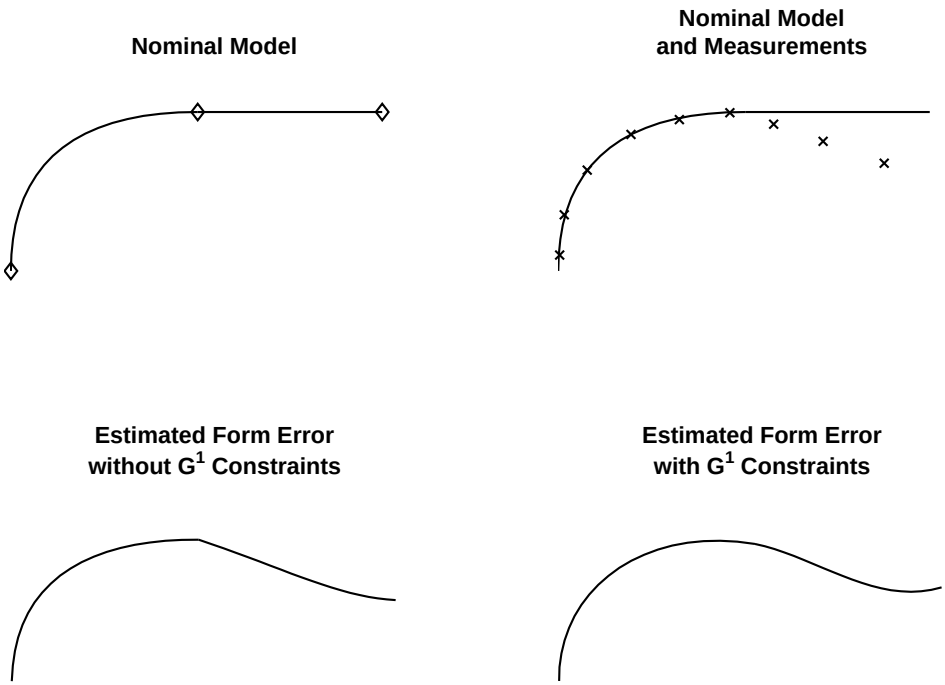

Figure 4.7: G^1 constraints for a nominal model consisting of two arcs (upper left) using simulated data (upper right). Form estimates with (lower right) and without (lower left) G^1 conditions.

4.14.3 First Order Continuity

Since a single arc is not sufficient to describe general geometries it is common to connect several of them to form so-called composite curves. In the sequel we will simply call them curves. Each arc is taken to be a polynomial in the Bézier representation. In order to look visually pleasing the arcs are often connected through tangent continuous joints. Performing a flexible best fit by applying a deformation vector field to such a curve we want the deformed curve to inherit the tangent continuity from the nominal model. This is illustrated in Figure 4.7. The upper left picture of Figure 4.7 depicts the nominal model we have chosen for this example. The curve consists of two arcs connected through a tangent continuous joint. A tangent continuous joint is also denoted a G^1 joint, the G meaning *geometric* as in geometric continuity and the one meaning first order, see Farin (1997, p. 181). The joints of the curve are marked by small diamonds.

The upper right figure of Figure 4.7 shows the nominal geometry again but here accompanied by a set of simulated measurements. The measurements are chosen to provoke the flexible best fit not to be G^1. Please note that the simulated measurements do not intend to illustrate a physical phenomena observed in

measurement data from a real situation but only serves to illustrate the properties of the flexible best fit.

The lower left illustration of Figure 4.7 demonstrates the effect of a flexible best fit to the given data set without imposing any G^1 constraints. Of course, the flexible best fit is subject to a continuity condition. A continuity condition is sometimes denoted a G^0 condition. Please observe the cusp that appears at the joint that joins the two arcs. Finally, the lower right subfigure of Figure 4.7 shows the effect of imposing a G^1 condition on the flexible best fit. The result is a smooth curve that does not reveal its joint by a cusp.

4.14.4 Regularisation Constraints

The flexible best fit generally introduces a large number of degrees of freedom. These degrees of freedom has to be determined by the information in the measurement data. However, when measuring is carried out using a touch-trigger probe, then measurements can only be sampled at a rate of 50-60 points a minute (Ramsdale, 1999). In such a case it is unreasonable to sample enough points to make the optimisation problem of the flexible best fit well-posed. Nevertheless, it might still be interesting to make a flexible best fit. The flexible best fit problem is then ill-posed and need to be regularised. Regularisation is described in Section 4.10 and is accomplished by adding information about the nominal form of the workpiece in the optimisation process. More specifically, we described how to regularise the problem by including information about the length of the nominal curve, its shape and the magnitude of the joint angles whenever they are not G^1. Here, we give examples of the effect of each of these three parameters.

Regularising the Length

The optimisation problem (4.2)–(4.3) of the flexible best fit stated in Section 4.3 is formulated to minimise the orthogonal distance between the measurements and the nominal model. Hence, the measurements do not control the deformed curve in any other directions than those that lie in the normal plane of least distance projection of the measurements. Consequently, the curve can change its length without affecting the value of the target function of the flexible best fit problem at all, hence it is singular. This motivates the introduction of a constraint that restricts the length of the deformed curve to deviate excessively from the length of the nominal curve. The length constraint is discussed and derived in Section 4.10.1. We try to illustrate the length constraint in Figure 4.8. The curve to the left in Figure 4.8 is chosen as the nominal model. The crosses in the same figure represent measurement data. In the middle of Figure 4.8 we show the result of a flexible best fit to the simulated measurements. The length of the deformed curve has decreased by 5.7% in comparison to the length of the nominal curve. A length change this large is unacceptable in practice. The length change of the flexible best fit shown to the right in Figure 4.8 is reduced to 2.6% as an effect of applying the length constraint with parameter $v = 0.01$.

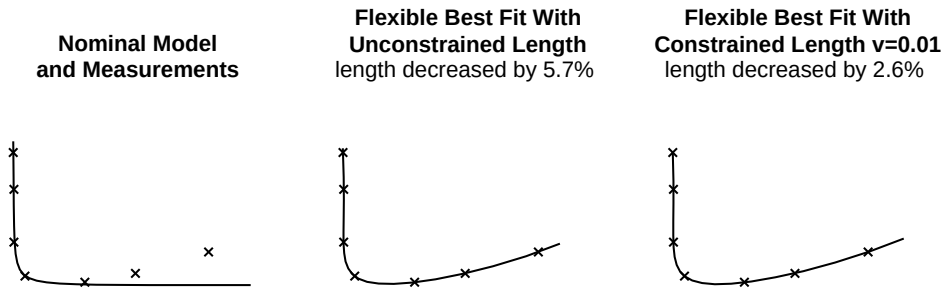

Figure 4.8: Length preservation demonstrated with a flexible best fit to the measurement data shown to the left. The left picture also depicts the nominal model. Flexible fits with and without length constraint are shown in to the right and in the middle, respectively. The deformation is a single cubic polynomial.

Constraining the Shape Change

Perhaps the most important way to regularise the flexible best fit is to constrain the shape change of the deformed curve. The shape difference between the deformed and the nominal curve is measured by the curvature of the deformed curve relative to the nominal. The details are described in Section 4.10.2.

The nominal model we have chosen to demonstrate the shape constraint is shown to the left in Figure 4.9 and consists of three arcs joined in two G^1 joints. We simulated measurements by picking ten points distributed uniformly along the curve. The points are perturbed by adding sinusoidal perturbations in the normal directions of the points. The added perturbations are meant to represent a plausible form error of a corresponding physical workpiece subject to inspection. The simulated measurements are shown together with the nominal model in the middle of Figure 4.9. Our intention is to recover the simulated form error through the flexible best fit. We have chosen the deformation to consist of piecewise polynomials of degree four. Such a polynomial is chosen to represent the form error on each of the three arcs constituting the nominal model. Since the joints of the nominal model are G^1, we apply such a condition on the deformation in addition to the

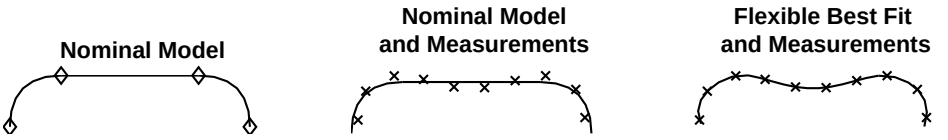

Figure 4.9: Nominal model to the left, consisting of three arcs connected through G^1 joints. Measurements with a sinusoidal perturbation in the middle and the flexible best fit to them shown to the right.

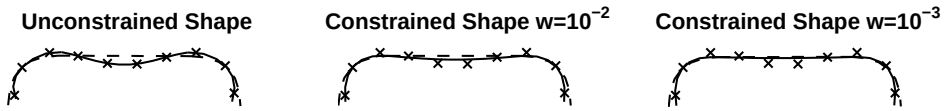

Figure 4.10: The flexible best fit (solid line) with increasingly constrained shape to the right. Please note how the flexible best fit approaches the rigid part of the deformation (dashed line) as the shape constraint parameter is decreased.

length constraint $v = 0.01$. The deformation has 24 degrees of freedom. This is because a fourth degree polynomial in the plane has $5 \times 2 = 10$ degrees of freedom that adds up to 30 for the three arcs. The continuity condition reduces the degrees of freedom by two for each joint and the first order geometric continuity condition reduces the degrees of freedom by one for each joint. In conclusion, the degrees of freedom for the deformation is $30 - 2 \times 2 - 2 \times 1 = 24$. Recall that we have simulated ten measurements that determine one degree of freedom each. Hence, there are $24 - 10 = 14$ degrees of freedom left undetermined. The result of the flexible best fit is shown to the right in Figure 4.9.

If the problem is under-determined because of too few measurements as in this case we can use the shape constraint to regularise the problem by including shape information about the nominal form. In the under-determined case we face the risk of including the noise of the measurements in the deformation function of the flexible best fit. Then we can use the shape constraint to hold back the fit of the deformation to the measurements. Figure 4.10 shows an example of this. The picture to the left in Figure 4.10 shows the same solution as is depicted to the right in Figure 4.9. The difference between the two is that we have included the rigid part of the deformation in the latter figure. The rigid part of the deformation is shown by the dashed line. The rigid part of the flexible best fit is computed using the equilibrium separation condition. The pictures in the middle and to the right in Figure 4.10 show the result of the flexible best fit using the shape constraint with parameters $w = 10^{-2}$ and $w = 10^{-3}$, respectively. As seen in the figures, the deformation is increasingly restricted as the shape constraint parameter decreases. In presence of noise, the shape constraint can be useful in holding back the fit so as not to interpolate the noise in the estimate of the form error. The size and distribution of the residuals can be used as a guideline for the selection of the shape constraint parameter. Ideally, we want the systematic form errors to be modelled by the deformation function $\boldsymbol{d}$. A method for selecting the shape constraint parameter could be to start with a small value of w and increase it until there is no systematic error left in the residuals. The absence of systematic errors can be detected by the residuals being independent and normally distributed, assuming the noise has these properties.

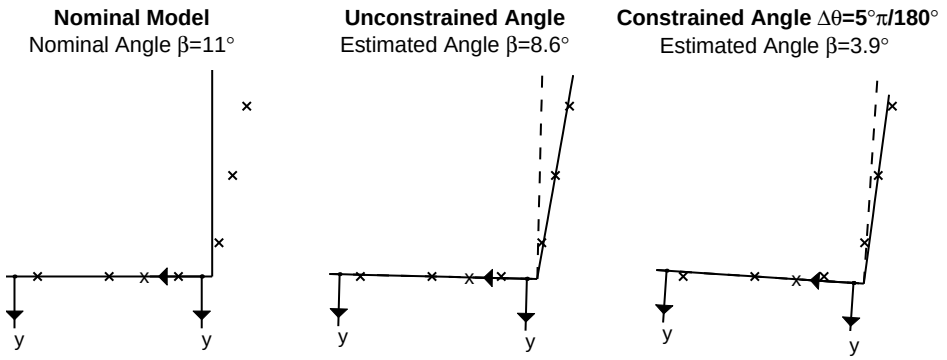

Figure 4.11: The joint angle constraint is illustrated using a nominal model equipped with a datum system. The flexible best fit with and without joint angle constraint to the right and in the middle, respectively.

The Joint Angle Constraint

In cases where the joints connecting the arcs of a curve are not G^1, then there is nothing that limits the corresponding joints of the deformed curve in the flexible best fit. Our aim to restrict the joints of the deformed curve has motivated the development of the joint angle constraint. This constraint, which is derived in Section 4.10.3, can be thought of as a discrete analogue of the shape constraint exemplified in the previous section.

In Figure 4.11 we use the same nominal model as in Figure 4.5. The measurements, visible along with the nominal model in the left picture of Figure 4.11, represent an object that has a joint angle that is larger than that of the nominal model. The word angle refers to the angle between the two linear arcs that constitute the nominal model. Please note that the angle of the nominal model is 90°, i.e. the two arcs of the nominal model are orthogonal. A flexible best fit of the nominal model to the simulated measurement data is shown in the middle of Figure 4.11. The estimated joint angle difference β between the joint of the deformed curve and the joint of the nominal curve is 8.6°. If we want to limit the joint angle difference we can use the joint angle constraint. The picture to the right in Figure 4.11 shows the result of letting the joint angle constraint parameter be $\Delta\theta = 5° \frac{\pi}{180°}$. Recall from equation (4.42) that the joint angle constraint satisfies the relation $2(\alpha^2 + \beta^2) = \Delta\theta$ when the constraint is active, which is certainly the case here. The angles α and β are the left and right limit angles of the current joint between the rigid and the flexible part of the deformation, respectively. For the model under consideration in this example, the left limit angle α is zero due to our choice of datum system. Hence, the maximal theoretical value of the deformation is $\beta = \Delta\theta/\sqrt{2} = 5°/\sqrt{2} \approx 3.5°$, which is close to the estimated angle $\beta = 3.9°$. Recall, though, that the derived constraint is based on an approximation of the true angle, hence the discrepancy between the estimate and the theoretical value.

4.15 Case Study: An Automotive Roof

In order to verify how the flexible best fit performs in practice we have done a case study. We have chosen to investigate a prototype of an automotive roof. The nominal model of the roof is depicted in Figure 4.12. The prototype roof is measured along five cross-sections denoted by X1 to X5, see Figure 4.13. The nominal model of cross-section X1 is shown in Figure 4.14. The left picture of Figure 4.14 shows the nominal model and the datum system that we have chosen for the cross-section. Please note that we have defined separate datum systems for each of the five cross-sections. Since the cross-sections are plane curves, the datum system only consists of three datum points and three guiding directions. We remark that the datum systems of the curves should only be considered separately, i.e. the union of them cannot be interpreted as a datum system for the roof itself. The reason for choosing a datum system for each cross-section is to allow demonstration of the datum separation condition.

The picture in the middle of Figure 4.14 shows the arcs that constitute the nominal model of cross-section X1. The joints of the curve that connect the arcs are denoted by diamonds. We apologise for the difficulty of seeing the composition of the curve close to the ends. However, the construction of the curve close to the ends is better seen in Figure 4.15. The nominal model of cross-section X1 is made up from 20 arcs of degree five, i.e. quintics. Cross-sections X2 and X3, which are not displayed separately due to their similarity with cross-section X1, are made up from 18 and 20 quintic arcs, respectively. The cross-sections X1, X2 and X3 are measured in 37 points indicated by the markers in the right picture of Figure 4.14. In this section we focus on the results from cross-section X3 why we leave the figure that illustrates cross-sections X4 and X5 to Appendix D, see Figure D.7. The figure displays cross-sections X4 and X5 in the same way as Figure 4.14 displays cross-sections X1, X2 and X3. We continue this section by discussing the results from cross-section X1 in some detail and leave the results from the other cross-sections to Appendix D.

4.15.1 The Measuring

Measurement data from a prototype part $\mathcal{P}$ corresponding to the nominal model of the roof is used as input in the flexible best fit. This section briefly describes how the measuring was carried out. The prototype was put to rest on a fixture whose locators supported the roof in its datum points. Please note that we are only interested in analysing cross-sections why we have not displayed the true datum system of the roof in Figure 4.12. The interested reader can think of the datum system of the automotive roof as the one illustrated in Figure 3.5. The position of the fixture locators were required to satisfy a certain tolerance bound $\pm\varepsilon^{\text{TOL}}$ in order for the part to be correctly located in the CMM coordinate system. The part was then measured in the fixed coordinate system as described in Section 1.2. The measurements were taken along the cross-sections seen in Figure 4.13. The cross-sections were sampled in the points shown in Figures 4.14 and D.7.

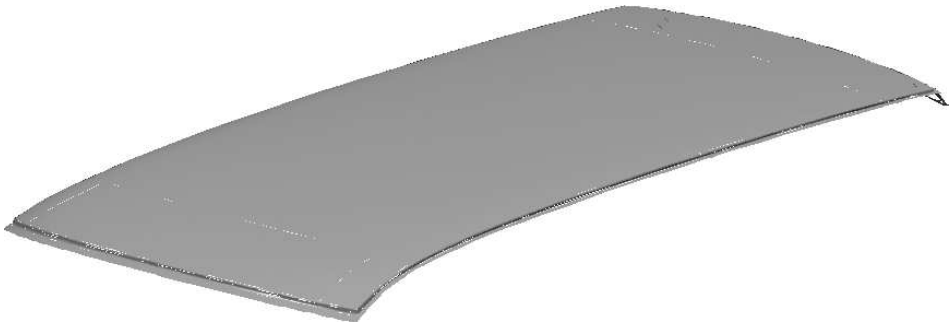

Figure 4.12: Snapshot of a CAD-model of an automotive roof.

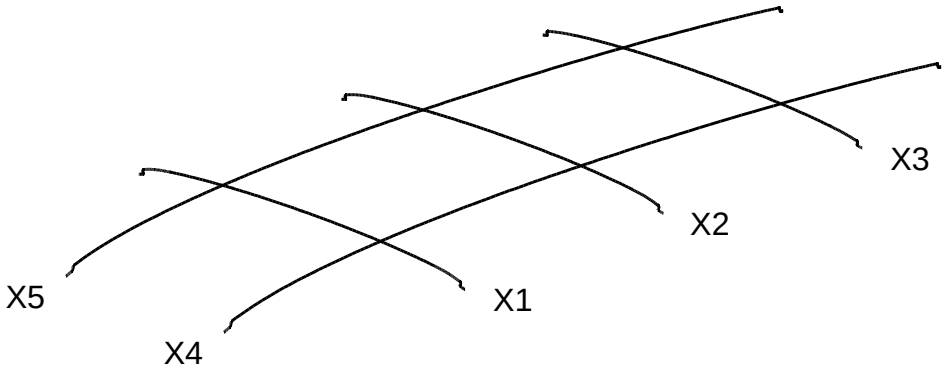

Figure 4.13: The figure shows five cross-sections taken from the automotive roof illustrated in Figure 4.12. The cross-sections are numbered X1 to X5.

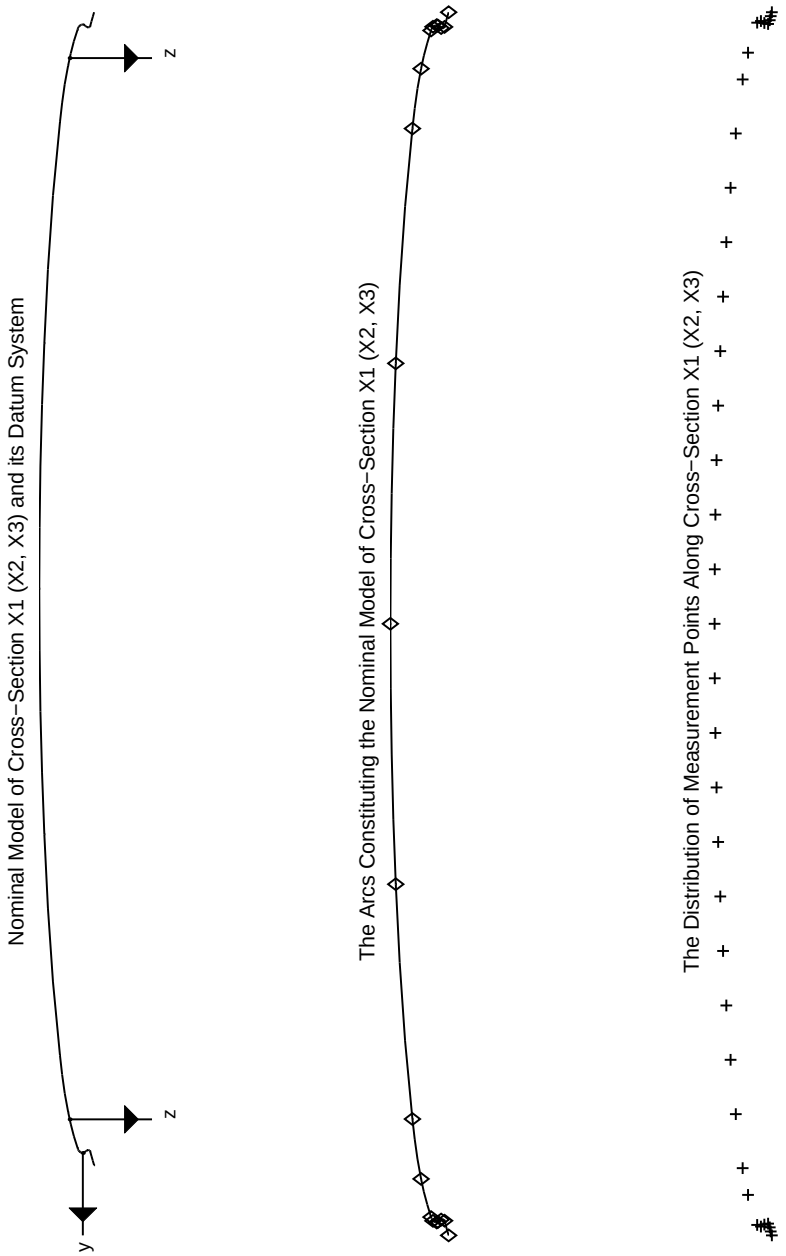

Figure 4.14: A closer view of the cross-sections from Figure 4.13. The cross-section X1 is taken to represent X2 and X3 as well since they are very similar. The pictures illustrate the appearance of the model as well as its composition of arcs. The points subject to inspection are also indicated.

4.15.2 Data Analysis

This section concentrates on the result of cross-section X1. The results from the other cross-sections are put in Appendix D. Since it is difficult to view the result for the whole cross-section we have chosen to concentrate on the left end of cross-section X1, see top of Figure 4.15. The figure also illustrates the arcs that constitute the curve. The joints of the arcs are marked by diamonds. The second picture of Figure 4.15 shows the CMM measurements together with the nominal model of cross-section X1 in the fixed coordinate system. The deviation of the measurements from the nominal model of the cross-section, in the sequel denoted by the *residuals*, are displayed in the top diagram of Figure 4.16. The residuals are computed as the signed distance from the measurements to their closest points in the nominal model. This corresponds to leaf (ii) in the tree depicted in Figure 1.1. The residuals are sorted from left to right so that the first eight bars in the diagram correspond to the measurements in the figure. The second diagram of Figure 4.16 shows the residuals after performing a rigid best fit of the measurements to the nominal model as described in Chapter 2. This idea is represented by leaf (iii) in the tree of Figure 1.1. Comparing the initial residuals with the residuals after the rigid best fit we see that there is only a small change.

The third picture of Figure 4.15 shows a flexible best fit of the nominal model to the measurements and the bottom diagram of Figure 4.16 shows the residuals after the flexible best fit. The method is represented in Figure 1.1 by leaf (v). The flexible best fit is obtained by adding a linear polynomial to each arc of the cross-section, i.e. $n_{\boldsymbol{d}}^{(l)} = 2$ for all $l = 1, \ldots, L = 20$. The continuity of the deformation is guaranteed by the G^0 condition (4.18) of Section 4.9.1. The cross-section is made up from 20 plane arcs that are linear which implies that the deformation has 42 degrees of freedom. Since we only have 37 measurements that contribute to one degree of freedom each the problem is ill-posed. Recall that the measurements only add information in the normal direction of the least distance projection of the curve. In order to achieve an estimate of the form error through the flexible best fit we regularised the problem by constraining the length and joint angles of the deformed curve. The constraints are described in Section 4.10. We chose the joint angle constraint parameter to be $v = 0.01$ and the joint angle constraint parameter to be $\Delta\theta = 5^\circ \frac{\pi}{180^\circ}$. We computed the flexible best fit using both the equilibrium separation and datum separation. Recall that the separation condition is necessary in order to separate the form error from the positional error. The result of the equilibrium separation is seen in the fourth picture of Figure 4.15. The solid line is the flexible best fit, the dashed line is the rigid part of the flexible best fit as defined by the equilibrium separation condition and the dotted line is the nominal position of the nominal cross-section. The bottom picture of Figure 4.15 shows the result of the flexible best fit using the datum separation condition.

The upper two diagrams of Figure 4.17 show the form and positional error as estimated by the flexible best fit using equilibrium separation conditions. The form and positional errors are computed in the points where the cross-section was measured. In this way the estimated errors can be compared with the initial residuals and the residuals after a rigid best fit. Please remember, though, that the

flexible best fit gives an estimate of the form error anywhere on the cross-section. The lower two diagrams of Figure 4.17 show the estimated form and positional errors of cross-section X1 using the datum separation.

4.15.3 Conclusion

The residuals after the flexible best fit seen in the lower two diagrams of Figure 4.16 are small compared to the initial residuals shown in the top diagram of the same figure. Thus we conclude that our geometric model manages to incorporate a substantial part of the geometric deviation indicated by the measurements. However, there is still some systematic errors left in the residuals that our geometric model of the deformation does not manage to incorporate. Moreover, the residuals in points 4–6 and 32–34 are too large for the model to be completely satisfactory.

Nevertheless, by comparing the estimated form error in Figure 4.17 with the initial residuals in Figure 4.16 we see that the flexible best fit manages to capture the main characteristics of the geometric deviations seen in the initial residuals. If a better fit is desired then the flexibility of the deformation has to be increased either by increasing the degree of the deformation or by decreasing the joint angle constraint.

We can observe in the data that the form error estimated through a flexible best fit using the equilibrium condition appears to be similar to the residuals from the rigid best fit. This is not clearly seen in the results of cross-section X1 where there seems to be essentially no rigid body movement inherent in the initial residuals. However, this observation can be made from the results of cross-section X4. The similarities in the upper diagram of Figure D.10 that shows the estimated form error using equilibrium separation and the second diagram of Figure D.9 that display the residuals after a rigid best fit are evident. However, we recommend the datum separation condition to be used in the flexible best fit since the form error is related to the datum system in that case. The equilibrium separation relates the form error to its average which makes it difficult to pinpoint its cause.

In conclusion, the flexible best fit succeeds in capturing the essence of the form error. The resulting mathematical model that the flexible best fit gives rise to can be used to visualise the form error in the same CAD-environment as the nominal model. Moreover, by the flexible best fit we can separate the form error from the positional error. Bear in mind, though, that the form error is defined relative to the positional error in terms of the separation condition.

4.16 Cross-Section X1

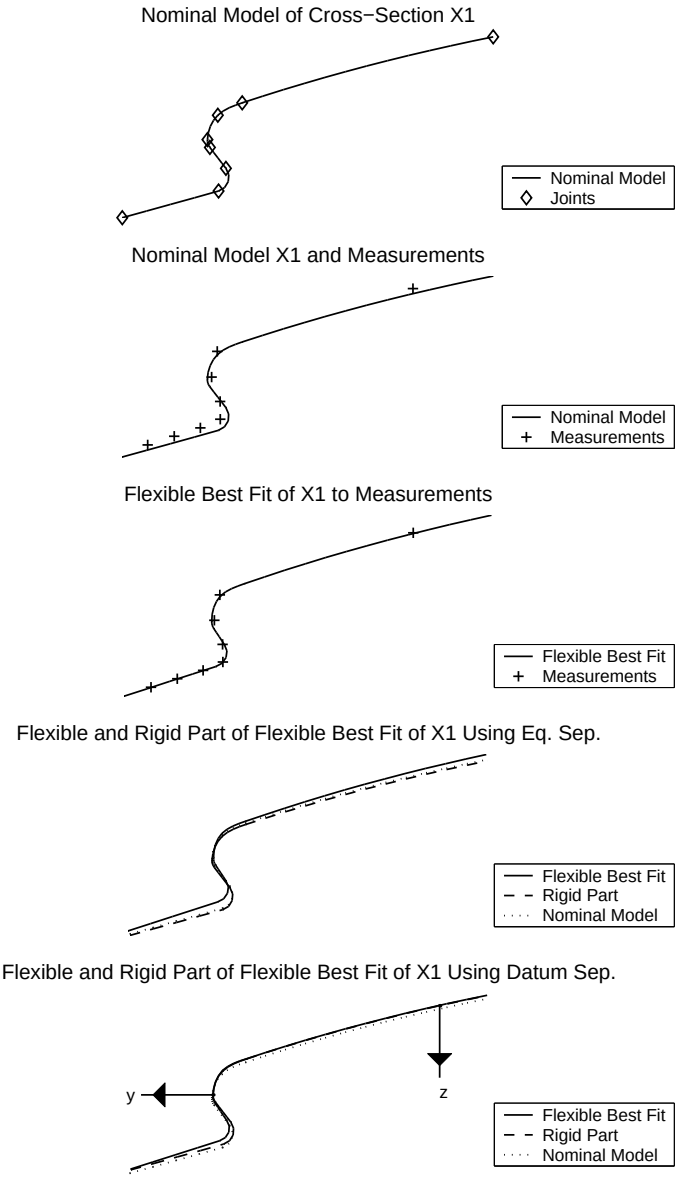

Figure 4.15: The five pictures show the result of a flexible best fit of cross-section X1 to measurement data. From top to bottom they show the joints of the nominal model, the measurements, the flexible best fit, the rigid and flexible parts of the flexible best fit using equilibrium separation and datum separation, respectively.

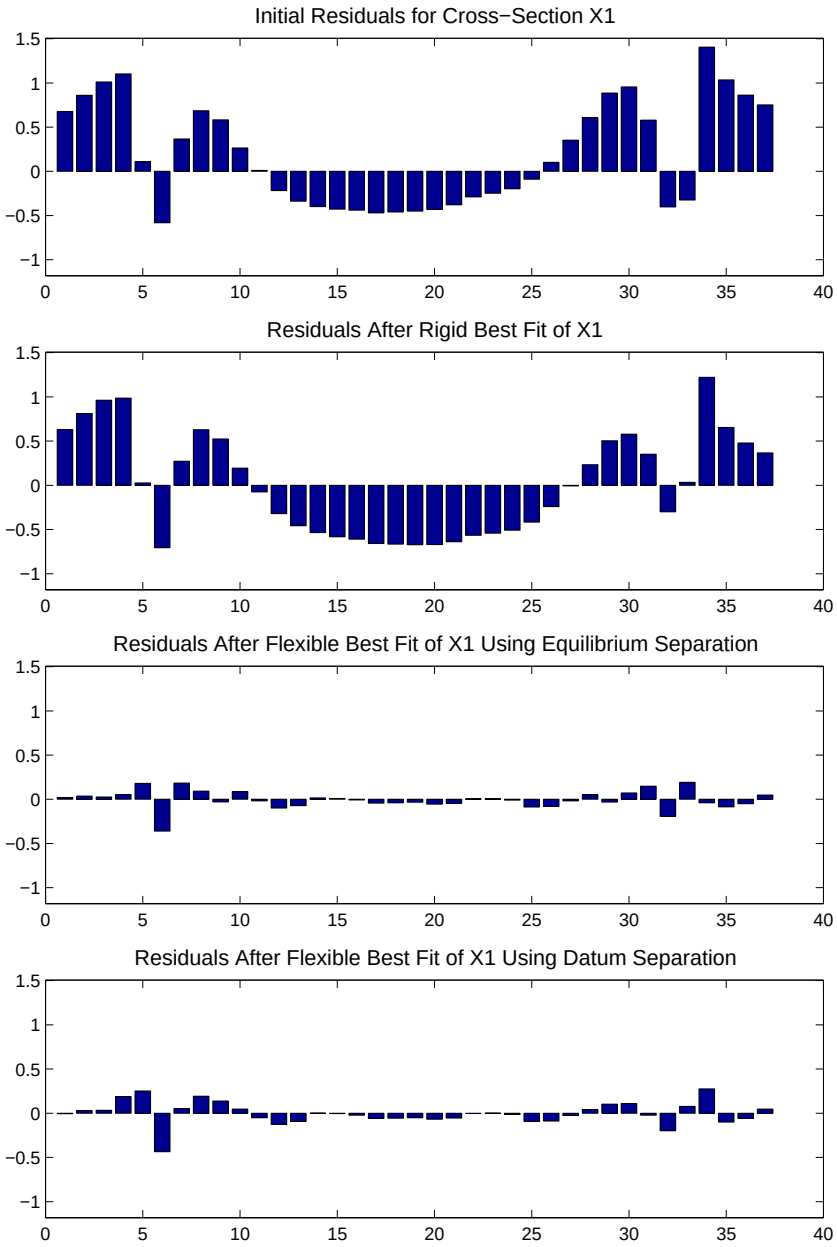

Figure 4.16: The top diagram shows the initial residuals of cross-section X1. The second diagram shows the residuals after a rigid best fit to all measurements. The two diagrams at the bottom show the residuals after a flexible best fit with equilibrium separation and datum separation, respectively.

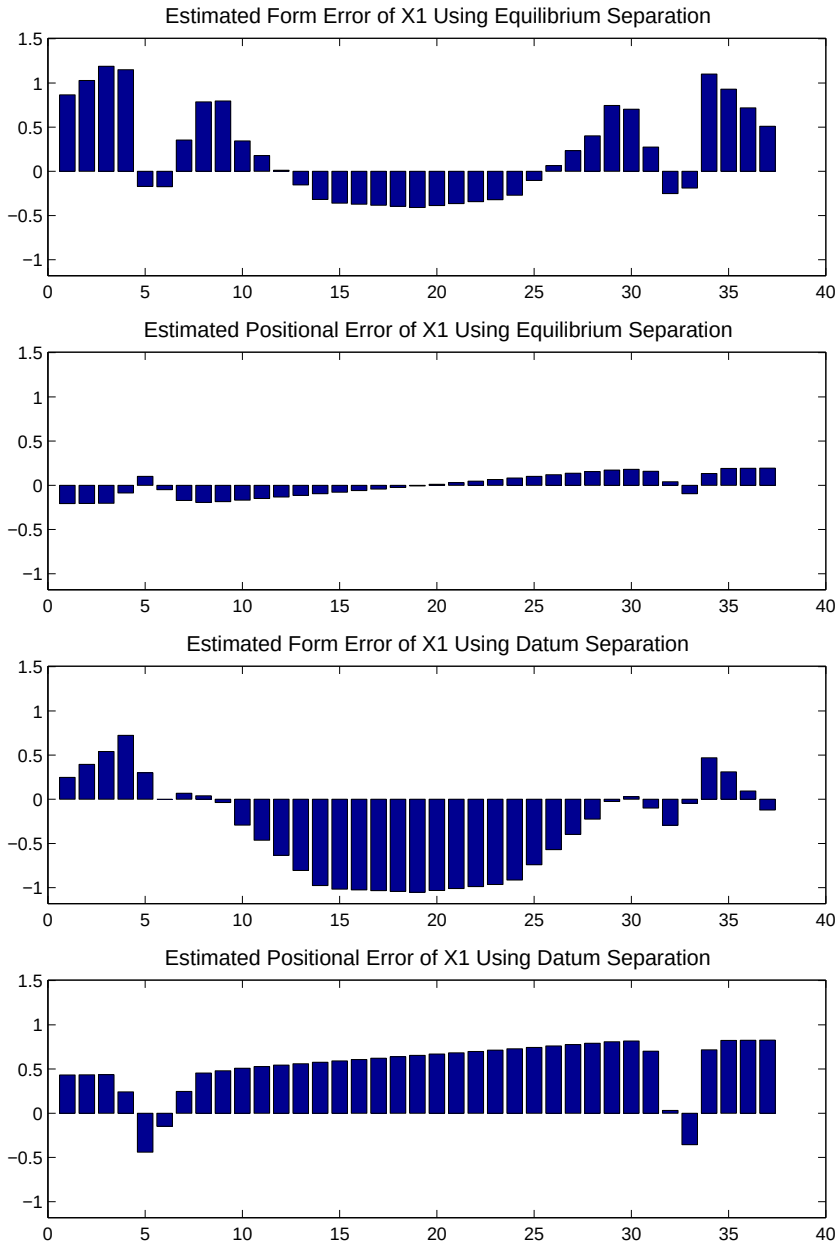

Figure 4.17: The two upper diagrams illustrate the form and positional error estimated by the flexible best fit of X1 using equilibrium separation and the two lower diagrams show same result using datum separation instead. The residuals are computed in the same points that were measured to allow for comparison.

4.17 Summary and Conclusion

We have stated a mathematical model that is able to model the form and positional errors from coordinate measurements of mechanical workpieces. The model deals with curves, which is useful in analysing geometric deviations of cross-sections.

We formulated an orthogonal least squares problem with the sum of the squared residuals as the target function, the residuals being defined as the distance from the measurements to their closest points in the nominal model. Identifying the problem as being separable we derived the derivatives of the separated problem. Recall that knowledge of the derivatives is essential for numerical solution.

Furthermore, we derived explicit continuity and first order continuity constraints. In order to deal with situations in which the number of measurements is not sufficient to determine the dof of the deformation and rigid body transformation, we discussed how to include a priori information about the nominal shape in the optimisation.

Finally, we specialised to the case when the deformation function is a composite Bézier curve and applied it to a number of examples illustrating different features of geometric model such as the tangent continuity and the regularisation conditions. The flexible best fit was then applied to cross-sections of an automotive roof. Measurements from a real prototype of the automotive roof was used as input to the flexible best fit that succeeded in creating a visually pleasing, mathematical model of the form error.

4.17.1 Open Issues

This study of the flexible best fit should be seen as a first step in trying to create a mathematical model of form errors of mechanical parts. Of course, there is a need for further investigation of several problems. For example, there is a need for a better optimiser, as MATLAB's fmincon in version 2 of the Optimisation Toolbox cannot handle second order information about the target function and the nonlinear constraints. As we have seen in the report, the second order information is available, and it would be advantageous to have an optimiser that can take care of it.

A very important issue left open is how to select adequate parameters for the flexible best fit. In addition to the degrees of freedom of the deformation, the regularisation parameters has to be set as well. For an industrial strength version of the flexible best fit, it is necessary to set these parameters automatically.

Finally, it would be interesting to generalise the flexible best fit to surfaces. Form and positional errors are three-dimensional phenomena and should be studied in a three-dimensional environment.

Appendix A

Derivative Definitions

Throughout the report we use the following definitions:

Definition A.1. *Let a function $f = f(\boldsymbol{x})$ be such that $f : \mathbb{R}^m \to \mathbb{R}$, where $\boldsymbol{x} \in \mathbb{R}^m$. Then we define the* derivative *of f with respect to $\boldsymbol{x}$ as*

$$f_{\boldsymbol{x}} = \frac{\partial f}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_m} \end{bmatrix}.$$

Definition A.2. *We define the derivative of a vector valued function $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^n$ with respect to $\boldsymbol{x} \in \mathbb{R}^n$ to be*

$$\boldsymbol{f}_{\boldsymbol{x}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & & \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix}.$$

Consequence A.3. *The second mixed derivative of a scalar valued function $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ with respect to $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$ is*

$$f_{\boldsymbol{xy}} = \frac{\partial^2 f}{\partial \boldsymbol{x} \partial \boldsymbol{y}} = \frac{\partial}{\partial \boldsymbol{x}} \Big(\frac{\partial f}{\partial \boldsymbol{y}} \Big)^T = \frac{\partial}{\partial \boldsymbol{x}} \begin{bmatrix} \frac{\partial f}{\partial y_1} \\ \frac{\partial f}{\partial y_2} \\ \vdots \\ \frac{\partial f}{\partial y_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial y_1} & \frac{\partial^2 f}{\partial x_2 \partial y_1} & \cdots & \frac{\partial^2 f}{\partial x_m \partial y_1} \\ \frac{\partial^2 f}{\partial x_1 \partial y_2} & \frac{\partial^2 f}{\partial x_2 \partial y_2} & & \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial y_n} & & \cdots & \frac{\partial^2 f}{\partial x_m \partial y_n} \end{bmatrix}.$$

Claim A.4. *If $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A}$ is an $m \times n$ matrix, then $\boldsymbol{f}_{\boldsymbol{x}} = \boldsymbol{A}$.*

Proof. The ith component of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ is $(\boldsymbol{f}(\boldsymbol{x}))_i = (\boldsymbol{A}\boldsymbol{x})_i = (\sum_{j=1}^n a_{ij} x_j)_i$ and since the (k,l)th component of $\boldsymbol{f}_{\boldsymbol{x}}$ is $\frac{\partial f_k}{\partial x_l}$ we have that

$$(\boldsymbol{f}_{\boldsymbol{x}})_{kl} = \frac{\partial f_k}{\partial x_l} = \frac{\partial}{\partial x_l} (\sum_{j=1}^n a_{kj} x_j) = a_{kl}.$$

□

Appendix B

Taylor Expansions

Here we derive Taylor expansions that are useful in Chapter 4.

Claim B.1. *If $\boldsymbol{a} \in \mathbb{R}^m$, $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{x} \in D \subset \mathbb{R}^n$ are such that $|\boldsymbol{Bx}| < |\boldsymbol{a}|$ whenever $\boldsymbol{x} \in D$, then the second order Taylor expansion of $|\boldsymbol{a}+\boldsymbol{Bx}|$ around the origin is*

$$|\boldsymbol{a}+\boldsymbol{Bx}| = |\boldsymbol{a}| + \frac{\boldsymbol{a}^T\boldsymbol{Bx}}{|\boldsymbol{a}|} + \frac{1}{2\,|\boldsymbol{a}|}\boldsymbol{x}^T\boldsymbol{B}^T(\boldsymbol{I} - \frac{\boldsymbol{a}\boldsymbol{a}^T}{|\boldsymbol{a}|^2})\boldsymbol{Bx} + o(|\boldsymbol{x}|^2). \tag{B.1}$$

Proof. Let $f(\boldsymbol{x}) = |\boldsymbol{a}+\boldsymbol{Bx}|$, then the zeroth order term of the expansion is $f(0) = |\boldsymbol{a}|$. For the first order term we need to differentiate f,

$$\begin{aligned} f_{\boldsymbol{x}}(\boldsymbol{x}) &= \frac{\partial}{\partial \boldsymbol{x}}\big(|\boldsymbol{a}+\boldsymbol{Bx}|\big) = \frac{\partial}{\partial \boldsymbol{x}}\Big(\big(|\boldsymbol{a}+\boldsymbol{Bx}|^2\big)^{\frac{1}{2}}\Big) \\ &= \frac{1}{2}\big(|\boldsymbol{a}+\boldsymbol{Bx}|^2\big)^{-\frac{1}{2}}\,2(\boldsymbol{a}+\boldsymbol{Bx})^T\boldsymbol{B} = \frac{(\boldsymbol{a}+\boldsymbol{Bx})^T}{|\boldsymbol{a}+\boldsymbol{Bx}|}\boldsymbol{B}. \end{aligned}$$

From the above it is obvious that $f_{\boldsymbol{x}}(0) = \boldsymbol{a}^T\boldsymbol{B}/|\boldsymbol{a}|$. Continuing with the second derivative we observe that $f_{\boldsymbol{x}}(\boldsymbol{x}) = f(\boldsymbol{x})^{-1}(\boldsymbol{a}+\boldsymbol{Bx})^T\boldsymbol{B}$ and from this we have

$$\begin{aligned} f_{\boldsymbol{xx}} &= \frac{\partial f_{\boldsymbol{x}}^T}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}}\Big(f(\boldsymbol{x})^{-1}\big(\boldsymbol{B}^T(\boldsymbol{a}+\boldsymbol{Bx})\big)\Big) \\ &= f(\boldsymbol{x})^{-1}\boldsymbol{B}^T\boldsymbol{B} - f(\boldsymbol{x})^{-2}\boldsymbol{B}^T(\boldsymbol{a}+\boldsymbol{Bx})f_{\boldsymbol{x}}(\boldsymbol{x}) \\ &= \frac{1}{f(\boldsymbol{x})}\boldsymbol{B}^T\Big(\boldsymbol{I} - \frac{(\boldsymbol{a}+\boldsymbol{Bx})(\boldsymbol{a}+\boldsymbol{Bx})^T}{f(\boldsymbol{x})^2}\Big)\boldsymbol{B} \\ &= \frac{1}{|\boldsymbol{a}+\boldsymbol{Bx}|}\boldsymbol{B}^T\Big(\boldsymbol{I} - \frac{(\boldsymbol{a}+\boldsymbol{Bx})(\boldsymbol{a}+\boldsymbol{Bx})^T}{|\boldsymbol{a}+\boldsymbol{Bx}|^2}\Big)\boldsymbol{B}. \end{aligned}$$

Evaluated at the origin

$$f_{\boldsymbol{xx}}(0) = \frac{1}{|\boldsymbol{a}|}\boldsymbol{B}^T(\boldsymbol{I} - \frac{\boldsymbol{a}\boldsymbol{a}^T}{|\boldsymbol{a}|^2})\boldsymbol{B}.$$

In conclusion, our derivations of $f(0)$, $f_{\boldsymbol{x}}(0)$ and $f_{\boldsymbol{xx}}(0)$ justify the stated Taylor expansion. □

Claim B.2. *Under the same circumstances as in Claim B.1 the second order Taylor expansion of* $|\boldsymbol{a}+\boldsymbol{B}\boldsymbol{x}|^{-1}$ *is*

$$\frac{1}{|\boldsymbol{a}+\boldsymbol{B}\boldsymbol{x}|}=\frac{1}{|\boldsymbol{a}|}-\frac{\boldsymbol{a}^T\boldsymbol{B}\boldsymbol{x}}{|\boldsymbol{a}|^3}+\frac{1}{2|\boldsymbol{a}|^3}\boldsymbol{x}^T\boldsymbol{B}^T\big(3\frac{\boldsymbol{a}\boldsymbol{a}^T}{|\boldsymbol{a}|^2}-\boldsymbol{I}\big)\boldsymbol{B}\boldsymbol{x}+o(|\boldsymbol{x}|^2). \tag{B.2}$$

Proof. Let $g(\boldsymbol{x})=1/f(\boldsymbol{x})$, where $f(\boldsymbol{x})=|\boldsymbol{a}+\boldsymbol{B}\boldsymbol{x}|$ as in the proof of Claim B.1. By observing that $g_{\boldsymbol{x}}=-f_{\boldsymbol{x}}/f^2$ and that

$$g_{\boldsymbol{x}\boldsymbol{x}}=2\frac{f_{\boldsymbol{x}}^T f_{\boldsymbol{x}}}{f^3}-\frac{f_{\boldsymbol{x}\boldsymbol{x}}}{f^2}=\frac{1}{f^2}\Big(2\frac{f_{\boldsymbol{x}}^T f_{\boldsymbol{x}}}{f}-f_{\boldsymbol{x}\boldsymbol{x}}\Big),$$

then a second order expansion of g around the origin takes the form

$$g(\boldsymbol{x})=\frac{1}{f(0)}-\frac{f_{\boldsymbol{x}}(0)\boldsymbol{x}}{f^2(0)}+\frac{1}{2f^2(0)}\boldsymbol{x}^T\Big(2\frac{f_{\boldsymbol{x}}(0)^T f_{\boldsymbol{x}}(0)}{f(0)}-f_{\boldsymbol{x}\boldsymbol{x}}(0)\Big)\boldsymbol{x}+o(|\boldsymbol{x}|^2).$$

The proof is completed by insertion of the derived expressions for $f(0)$, $f_{\boldsymbol{x}}(0)$ and $f_{\boldsymbol{x}\boldsymbol{x}}(0)$ from the proof of Claim B.1. □

Appendix C

Rigid Body Calculus

The technique of parametrising a rotation matrix in terms of its *axis of rotation* is described in detail by Angeles and Kecskeméthy (1995). The key observation they describe, that was known already to Euler, is that the invariants of a rigid body rotation are the axis and the angle of rotation. Furthermore, they show that if $\boldsymbol{\omega}$ is the axis of rotation for a *rotation matrix* $\boldsymbol{R}$ with rotation angle $\theta = |\boldsymbol{\omega}|$, then $\boldsymbol{R}(\boldsymbol{\omega}) = e^{[\boldsymbol{\omega}]}$, where

$$[\boldsymbol{\omega}] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

defines the *cross product matrix* of the vector $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$. The name is due to the fact that we can rewrite the cross product of two vectors by a matrix multiplication, i.e. $\boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]\boldsymbol{b}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors in $\mathbb{R}^3$. Obviously, we also have that $(\boldsymbol{a} \times \boldsymbol{b})^T = \boldsymbol{a}^T[\boldsymbol{b}]$. Furthermore, the cross product matrix is skew-symmetric, i.e. $[\boldsymbol{a}]^T = -[\boldsymbol{a}]$

Having identified the rotation matrix as the matrix exponential of the cross product matrix of its rotation vector we can easily expand the rotation matrix in a Taylor series, hence

$$\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p} = \boldsymbol{p} + \boldsymbol{\omega} \times \boldsymbol{p} + \frac{1}{2!}\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{p}) + \cdots + \frac{1}{k!}\boldsymbol{\omega} \times \big(\boldsymbol{\omega} \times \cdots \times (\boldsymbol{\omega} \times \boldsymbol{p})\big) + \cdots . \tag{C.1}$$

C.1 Derivatives of the Rotation Matrix

The first and second derivatives of the rotation matrix are derived in this section.

Claim C.1. *The first derivative of* $\boldsymbol{g}(\boldsymbol{\omega}) = \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}$*, where* $\boldsymbol{R}(\boldsymbol{\omega})$ *is a rigid body rotation matrix parametrised by its axis of rotation and* $\boldsymbol{p}$ *is a point in* $\mathbb{R}^3$*, is*

$$\boldsymbol{g}_{\boldsymbol{\omega}}(\boldsymbol{\omega}) = \frac{\partial}{\partial \boldsymbol{\omega}}\big(\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}\big) = -[\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}].$$

Proof. It is immediately seen from the expansion (C.1) that the derivative of $\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}$ evaluated at the origin is $-[\boldsymbol{p}]$. Suppose now we are interested in the derivative at $\boldsymbol{\omega}_0$, say. Then

$$\begin{aligned}\boldsymbol{g}_{\boldsymbol{\omega}}(\boldsymbol{\omega}_0) &= \frac{\partial}{\partial \boldsymbol{\omega}}\big(\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}\big)\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0} = \frac{\partial}{\partial \boldsymbol{\omega}}\big(\boldsymbol{R}(\boldsymbol{\omega}-\boldsymbol{\omega}_0)\boldsymbol{R}(\boldsymbol{\omega}_0)\boldsymbol{p}\big)\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0} \\ &= \{\tilde{\boldsymbol{p}} = \boldsymbol{R}(\boldsymbol{\omega}_0)\boldsymbol{p}\} = \frac{\partial}{\partial \boldsymbol{\omega}}\big(\boldsymbol{R}(\boldsymbol{\omega})\tilde{\boldsymbol{p}}\big)\Big|_{\boldsymbol{\omega}=0} = -[\tilde{\boldsymbol{p}}] = -[\boldsymbol{R}(\boldsymbol{\omega}_0)\boldsymbol{p}],\end{aligned}$$

or more generally $\boldsymbol{g}_{\boldsymbol{\omega}}(\boldsymbol{\omega}) = \frac{\partial}{\partial \boldsymbol{\omega}}\big(\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}\big) = -[\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}]$. □

In order to compute the second derivative without having to resort to tensor notation due to that we get three-dimensional matrices we compute the second derivatives for the scalar function $h(\boldsymbol{\omega}) = \boldsymbol{q}^T\boldsymbol{g}(\boldsymbol{\omega})$. For our purposes it suffices to have the second derivative in this form.

Claim C.2. *The second derivative of $h(\boldsymbol{\omega}) = \boldsymbol{q}^T\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}$, where $\boldsymbol{R}(\boldsymbol{\omega})$ is a rotation matrix parametrised by its axis of rotation and $\boldsymbol{p}$ and $\boldsymbol{q}$ are points in $\mathbb{R}^3$, is*

$$h_{\boldsymbol{\omega}\boldsymbol{\omega}}(\boldsymbol{\omega}) = \frac{\partial^2}{\partial \boldsymbol{\omega}^2}\big(\boldsymbol{q}^T\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}\big) = [\boldsymbol{q}][\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}] + [\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}][\boldsymbol{q}].$$

Proof. Using the expansion (C.1) we can write the function h as

$$h(\boldsymbol{\omega}) = \boldsymbol{q}^T\boldsymbol{p} - \boldsymbol{q}^T[\boldsymbol{p}]\boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\omega}^T\big([\boldsymbol{p}][\boldsymbol{q}] + [\boldsymbol{q}][\boldsymbol{p}]\big)\boldsymbol{\omega} + o\big(|\boldsymbol{\omega}|^2\mathbb{1}\big).$$

From the above it is evident that $h_{\boldsymbol{\omega}\boldsymbol{\omega}}(0) = [\boldsymbol{p}][\boldsymbol{q}] + [\boldsymbol{q}][\boldsymbol{p}]$. We can evaluate the derivative at an arbitrary parameter $\boldsymbol{\omega}_0$ using the same technique as in the proof of Claim C.1, i.e.

$$\begin{aligned}h_{\boldsymbol{\omega}\boldsymbol{\omega}}(\boldsymbol{\omega}_0) &= \frac{\partial^2}{\partial \boldsymbol{\omega}^2}(\boldsymbol{q}^T\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p})\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0} = \frac{\partial^2}{\partial \boldsymbol{\omega}^2}(\boldsymbol{q}^T\boldsymbol{R}(\boldsymbol{\omega}-\boldsymbol{\omega}_0)\boldsymbol{R}(\boldsymbol{\omega}_0)\boldsymbol{p})\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_0} \\ &= \{\tilde{\boldsymbol{p}} = \boldsymbol{R}(\boldsymbol{\omega}_0)\boldsymbol{p}\} = \frac{\partial^2}{\partial \boldsymbol{\omega}^2}(\boldsymbol{q}^T\boldsymbol{R}(\boldsymbol{\omega})\tilde{\boldsymbol{p}})\Big|_{\boldsymbol{\omega}=0} \\ &= [\boldsymbol{q}][\tilde{\boldsymbol{p}}] + [\tilde{\boldsymbol{p}}][\boldsymbol{q}] = [\boldsymbol{q}][\boldsymbol{R}(\boldsymbol{\omega}_0)\boldsymbol{p}] + [\boldsymbol{R}(\boldsymbol{\omega}_0)\boldsymbol{p}][\boldsymbol{q}].\end{aligned}$$

By dropping the subscript on $\boldsymbol{\omega}_0$ we can write $h_{\boldsymbol{\omega}\boldsymbol{\omega}}(\boldsymbol{\omega}) = [\boldsymbol{q}][\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}] + [\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}][\boldsymbol{q}]$. □

C.2 The Rigid Body Transform

Definition C.3. *The map $\mathcal{R} : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^3$ with parameters $\boldsymbol{x} = [\boldsymbol{t}^T\ \boldsymbol{\omega}^T]^T \in \mathbb{R}^6$, where $\boldsymbol{t}, \boldsymbol{\omega} \in \mathbb{R}^3$, is defined by $\mathcal{R}(\boldsymbol{x})\boldsymbol{p} = \boldsymbol{t} + \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}$, where $\boldsymbol{R}(\boldsymbol{\omega})$ is a rotation matrix and $\boldsymbol{p} \in \mathbb{R}^3$ is a point. The map $\mathcal{R}$ is called a* rigid body transform.

Please remark that the rigid body transform is a special case of an affine map. The parameter $\boldsymbol{t}$ is called the *translation vector* and $\boldsymbol{\omega}$ is called the *rotation vector.*

C.3 Derivatives of the Rigid Body Transform

Since the rigid body transformation $\mathcal{R}(\boldsymbol{x})$ with transformation parameters $\boldsymbol{x} = [\boldsymbol{t}^T \ \boldsymbol{\omega}^T]^T$ is linear in $\boldsymbol{t}$ with unit coefficient its first derivative with respect to $\boldsymbol{t}$ is the identity and the second derivative vanishes. Thus we can sum up by stating its derivatives with respect to $\boldsymbol{x}$ as direct consequences of the above and the Claims C.1 and C.2.

Consequence C.4. *If $\mathcal{R}(\boldsymbol{x})$ is a rigid body transform and $\boldsymbol{p}$ a point in $\mathbb{R}^3$ then the derivative of $\mathcal{R}(\boldsymbol{x})\boldsymbol{p}$ with respect to $\boldsymbol{x}$ is*

$$\frac{\partial}{\partial \boldsymbol{x}}\big(\mathcal{R}(\boldsymbol{x})\boldsymbol{p}\big) = \begin{bmatrix} \boldsymbol{I} & -[\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}] \end{bmatrix}. \tag{C.2}$$

Consequence C.5. *If $\mathcal{R}(\boldsymbol{x})$ is a rigid body transform and $h(\boldsymbol{x}) = \boldsymbol{q}^T\mathcal{R}(\boldsymbol{x})\boldsymbol{p}$, where $\boldsymbol{p}$ and $\boldsymbol{q}$ are points in $\mathbb{R}^3$ then the second derivative of h with respect to $\boldsymbol{x}$ is*

$$h_{\boldsymbol{xx}}(\boldsymbol{x}) = \frac{\partial^2}{\partial \boldsymbol{x}^2}\big(\boldsymbol{q}^T\mathcal{R}(\boldsymbol{x})\boldsymbol{p}\big) = \begin{bmatrix} 0 & 0 \\ 0 & [\boldsymbol{q}][\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}] + [\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}][\boldsymbol{q}] \end{bmatrix}. \tag{C.3}$$

C.4 The Inverse of the Rigid Body Transform

Claim C.6. *The inverse of the rigid body transform $\mathcal{R}(\boldsymbol{x})$ acting on a point $\boldsymbol{p}$ in $\mathbb{R}^3$ is*

$$\mathcal{R}(\boldsymbol{x})^{-1}\boldsymbol{p} = \boldsymbol{R}(\boldsymbol{\omega})^{-1}(\boldsymbol{p} - \boldsymbol{t})$$

and has parameters

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} -\boldsymbol{R}(-\boldsymbol{\omega})\boldsymbol{t} \\ -\boldsymbol{\omega} \end{bmatrix}.$$

Proof. The inverse of $\mathcal{R}$ is defined by $\mathcal{R}(\boldsymbol{x})^{-1}\mathcal{R}(\boldsymbol{x})\boldsymbol{p} = \boldsymbol{p}$. Let the inverse have parameters $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{t}}^T \ \tilde{\boldsymbol{\omega}}^T]^T$, i.e. $\mathcal{R}(\tilde{\boldsymbol{x}}) = \mathcal{R}(\boldsymbol{x})^{-1}$. From direct calculation

$$\begin{aligned} \boldsymbol{p} &= \mathcal{R}(\boldsymbol{x})^{-1}\mathcal{R}(\boldsymbol{x})\boldsymbol{p} = \mathcal{R}(\tilde{\boldsymbol{x}})\mathcal{R}(\boldsymbol{x})\boldsymbol{p} = \mathcal{R}(\tilde{\boldsymbol{x}})\big(\boldsymbol{t} + \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}\big) \\ &= \tilde{\boldsymbol{t}} + \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\big(\boldsymbol{t} + \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}\big) = \tilde{\boldsymbol{t}} + \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{t} + \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p}. \end{aligned}$$

Equating the left and right and side it must be that

$$\tilde{\boldsymbol{t}} + \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{t} = 0, \quad \text{and} \quad \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{R}(\boldsymbol{\omega}) = \boldsymbol{I}.$$

The latter gives

$$\boldsymbol{R}(\tilde{\boldsymbol{\omega}}) = \boldsymbol{R}(\boldsymbol{\omega})^{-1} = e^{-[\boldsymbol{\omega}]} = e^{[-\boldsymbol{\omega}]} = \boldsymbol{R}(-\boldsymbol{\omega})$$

from which we conclude that $\tilde{\boldsymbol{\omega}} = -\boldsymbol{\omega}$ and from the former we see that

$$\tilde{\boldsymbol{t}} = -\boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{t} = -\boldsymbol{R}(-\boldsymbol{\omega})\boldsymbol{t},$$

hence $\tilde{\boldsymbol{x}} = [-\big(\boldsymbol{R}(-\boldsymbol{\omega})\boldsymbol{t}\big)^T \ -\boldsymbol{\omega}^T]^T$. As a consequence

$$\mathcal{R}(\tilde{\boldsymbol{x}})\boldsymbol{p} = \tilde{\boldsymbol{t}} + \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{p} = -\boldsymbol{R}(-\boldsymbol{\omega})\boldsymbol{t} + \boldsymbol{R}(-\boldsymbol{\omega})\boldsymbol{p} = \boldsymbol{R}(-\boldsymbol{\omega})\big(\boldsymbol{p}-\boldsymbol{t}\big) = \boldsymbol{R}(\boldsymbol{\omega})^{-1}\big(\boldsymbol{p}-\boldsymbol{t}\big).$$

□

Claim C.7. *Let $\mathcal{R}$ be a rigid body transform and assume $\boldsymbol{c}$ and $\boldsymbol{p}$ belong to $\mathbb{R}^3$. If $\boldsymbol{r} = \mathcal{R}(\tilde{\boldsymbol{x}})\boldsymbol{c} - \boldsymbol{p}$ then $|\boldsymbol{r}|^2 = |\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}|^2$, where $\tilde{\boldsymbol{x}}$ are the inverse rigid body transformation parameters to $\boldsymbol{x}$.*

Proof. Direct calculation gives

$$\begin{aligned}
\boldsymbol{r} &= \mathcal{R}(\tilde{\boldsymbol{x}})\boldsymbol{c} - \boldsymbol{p} = \tilde{\boldsymbol{t}} + \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\boldsymbol{c} - \boldsymbol{p} \\
&= \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\big(\boldsymbol{R}(\tilde{\boldsymbol{\omega}})^{-1}\tilde{\boldsymbol{t}} + \boldsymbol{c} - \boldsymbol{R}(\tilde{\boldsymbol{\omega}})^{-1}\boldsymbol{p}\big) = \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\big(\boldsymbol{R}(-\tilde{\boldsymbol{\omega}})\tilde{\boldsymbol{t}} + \boldsymbol{c} - \boldsymbol{R}(-\tilde{\boldsymbol{\omega}})\boldsymbol{p}\big) \\
&= \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\Big(\boldsymbol{c} - \big(-\boldsymbol{R}(-\tilde{\boldsymbol{\omega}})\tilde{\boldsymbol{t}} + \boldsymbol{R}(-\tilde{\boldsymbol{\omega}})\boldsymbol{p}\big)\Big) \\
&= \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\big(\boldsymbol{c} - (\boldsymbol{t} + \boldsymbol{R}(\boldsymbol{\omega})\boldsymbol{p})\big) = \boldsymbol{R}(\tilde{\boldsymbol{\omega}})\big(\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}\big)
\end{aligned}$$

and since the rotation matrix $\boldsymbol{R}(\tilde{\boldsymbol{\omega}})$ is orthogonal we know that $\boldsymbol{R}(\tilde{\boldsymbol{\omega}})^T\boldsymbol{R}(\tilde{\boldsymbol{\omega}}) = \boldsymbol{I}$ which is useful in computing

$$\begin{aligned}
|\boldsymbol{r}|^2 = \boldsymbol{r}^T\boldsymbol{r} &= \big(\boldsymbol{R}(\tilde{\boldsymbol{\omega}})(\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p})\big)^T\boldsymbol{R}(\tilde{\boldsymbol{\omega}})(\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}) \\
&= (\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p})^T\boldsymbol{R}(\tilde{\boldsymbol{\omega}})^T\boldsymbol{R}(\tilde{\boldsymbol{\omega}})(\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}\big) \\
&= (\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p})^T(\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}) = |\boldsymbol{c} - \mathcal{R}(\boldsymbol{x})\boldsymbol{p}|^2.
\end{aligned}$$

□

Appendix D

Results From Case Study

This appendix contains some of the results from the case study described in Section 4.15. The section also contain the results from one out of the five measured cross-sections that constitute that case study. The results from the other four cross-sections X2 to X5 are displayed here. The results are left uncommented since the comments made about the results of cross-section X1 in Section 4.15 applies to the cross-sections displayed here as well.

D.1 Cross-Section X2

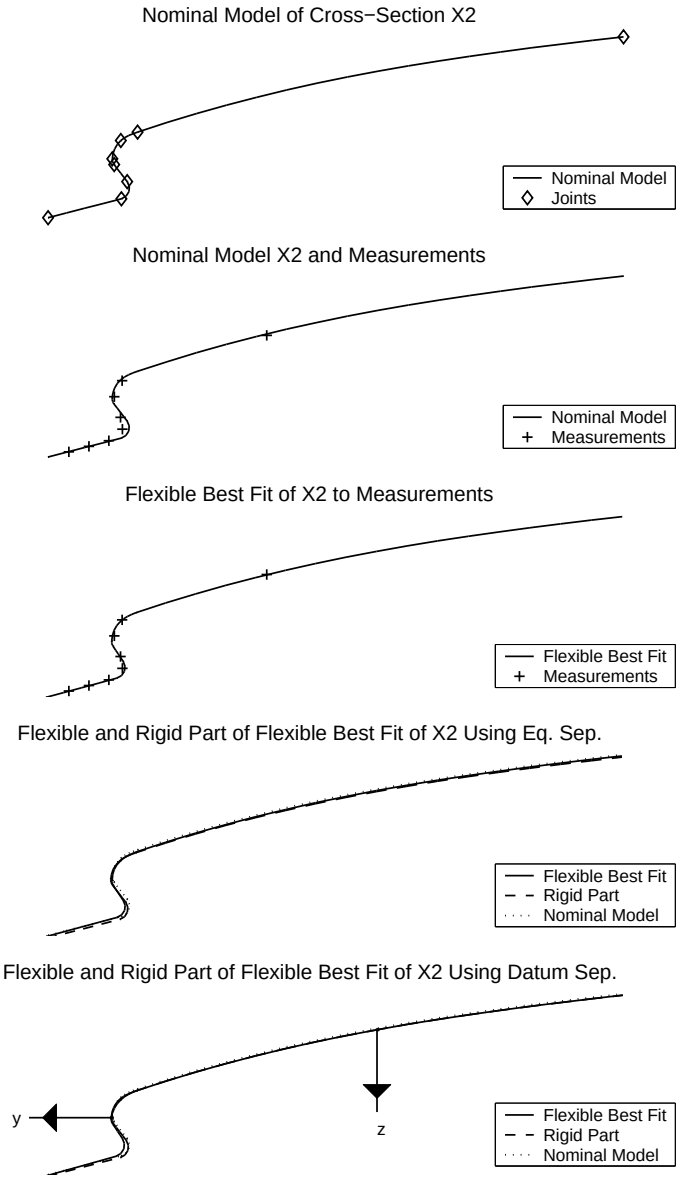

Figure D.1: The five pictures show the result of a flexible best fit of cross-section X2 to measurement data. From top to bottom they show the joints of the nominal model, the measurements, the flexible best fit, the rigid and flexible parts of the flexible best fit using equilibrium separation and datum separation, respectively.

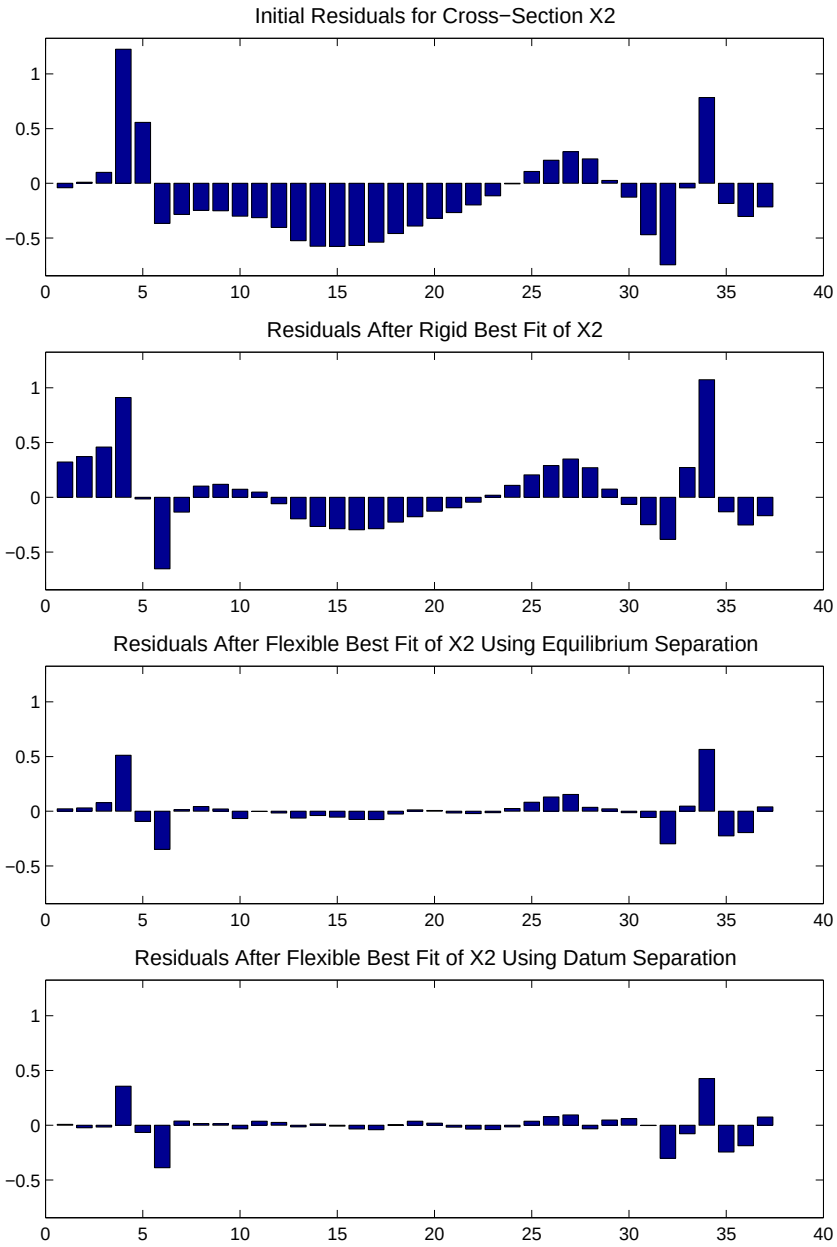

Figure D.2: The top diagram shows the initial residuals of cross-section X2. The second diagram shows the residuals after a rigid best fit to all measurements. The two diagrams at the bottom show the residuals after a flexible best fit with equilibrium separation and datum separation, respectively.

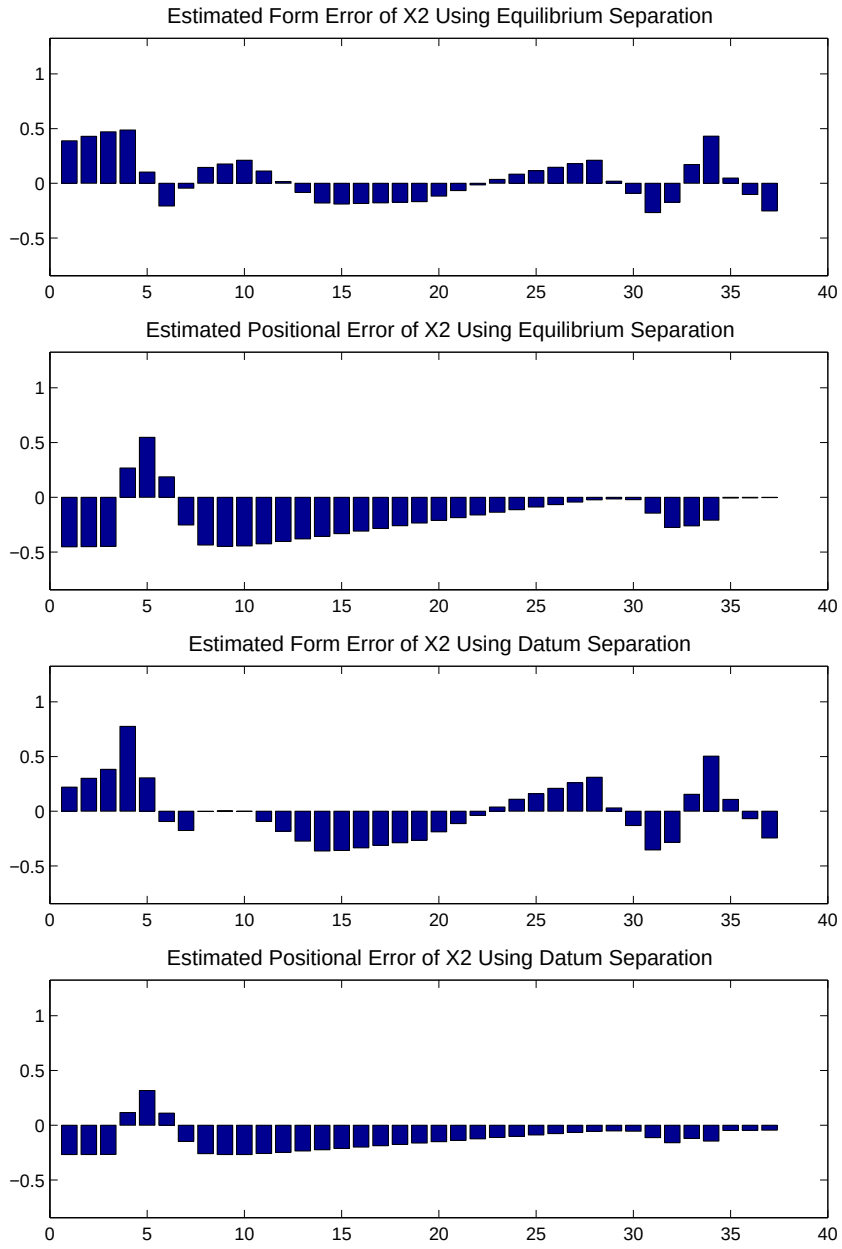

Figure D.3: The two upper diagrams illustrate the form and positional error estimated by the flexible best fit of X2 using equilibrium separation and the two lower diagrams show same result using datum separation instead. The residuals are computed in the same points that were measured to allow for comparison.

D.2 Cross-Section X3

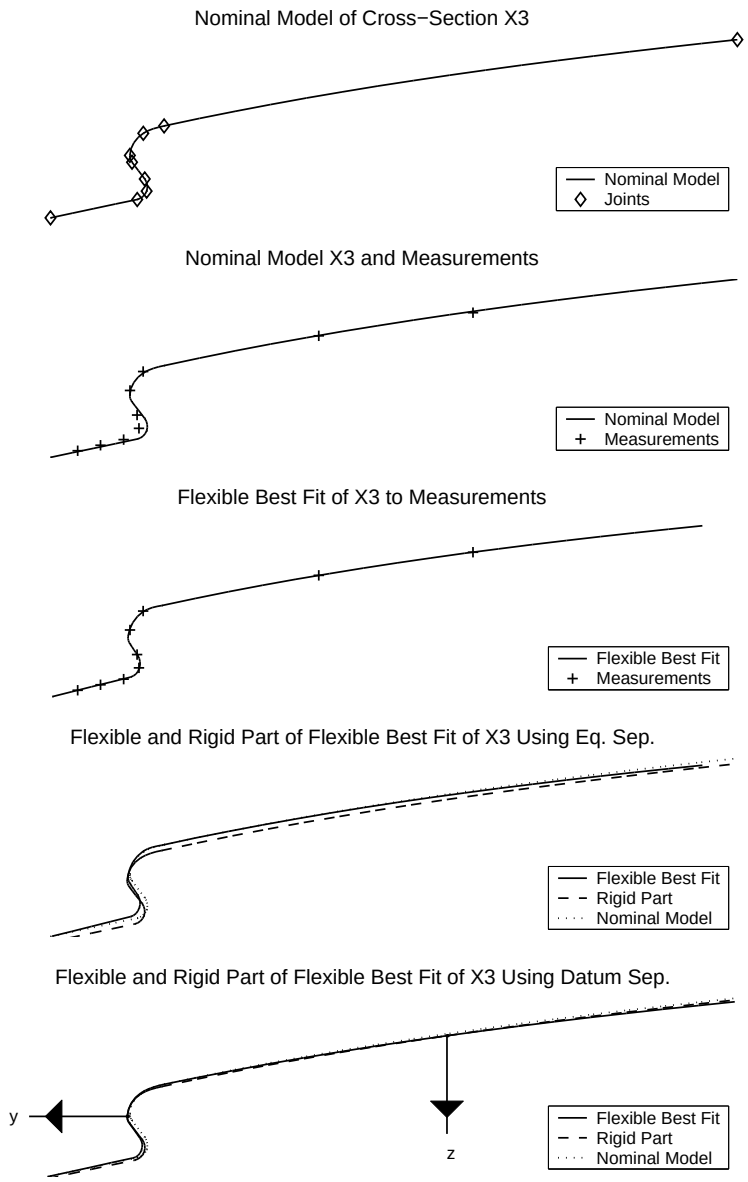

Figure D.4: The five pictures show the result of a flexible best fit of cross-section X3 to measurement data. From top to bottom they show the joints of the nominal model, the measurements, the flexible best fit, the rigid and flexible parts of the flexible best fit using equilibrium separation and datum separation, respectively.

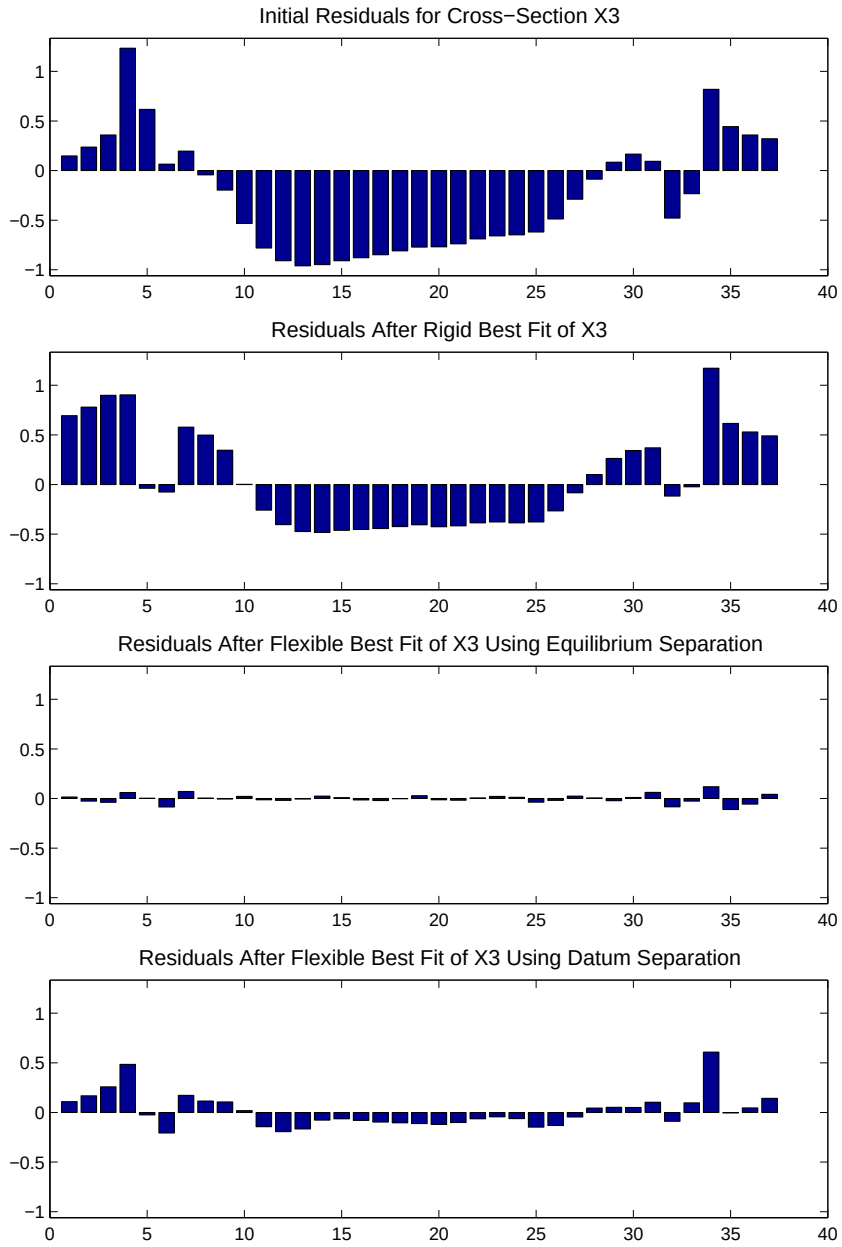

Figure D.5: The top diagram shows the initial residuals of cross-section X3. The second diagram shows the residuals after a rigid best fit to all measurements. The two diagrams at the bottom show the residuals after a flexible best fit with equilibrium separation and datum separation, respectively.

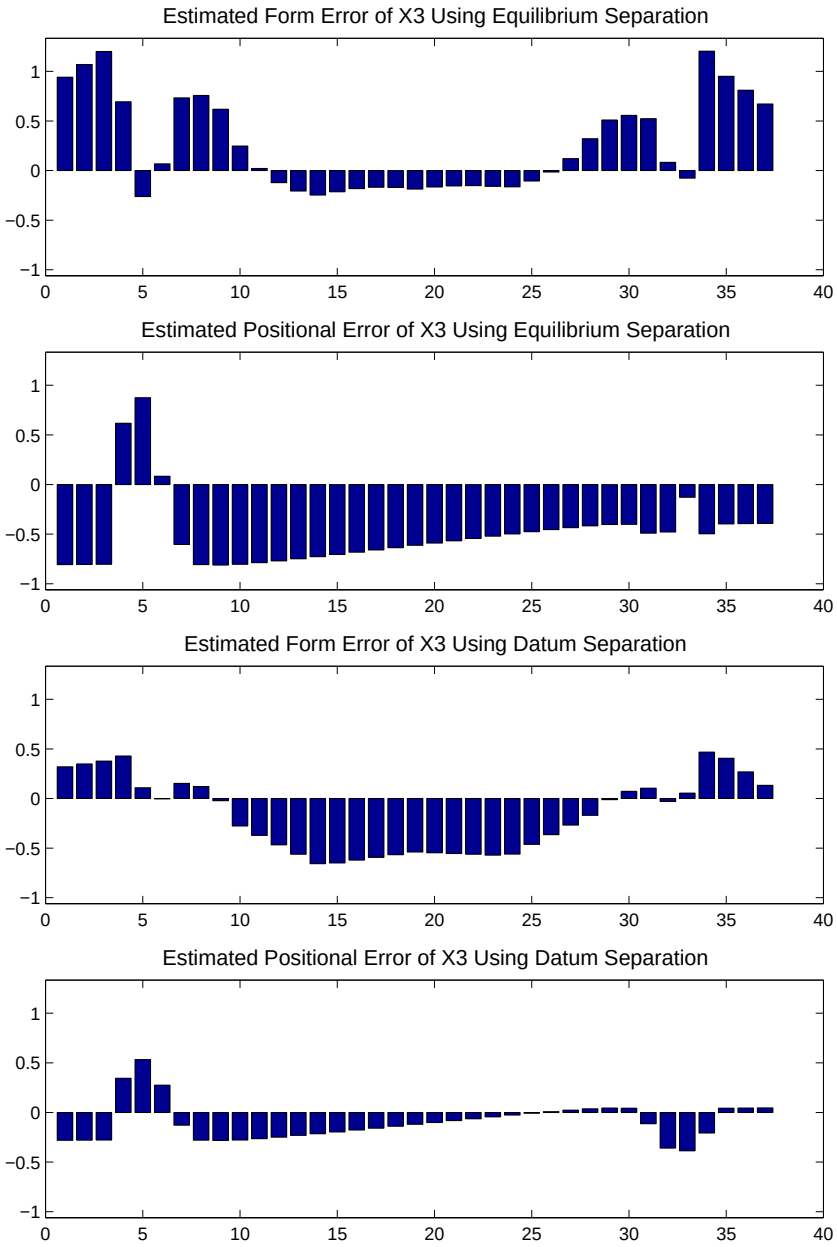

Figure D.6: The two upper diagrams illustrate the form and positional error estimated by the flexible best fit of X3 using equilibrium separation and the two lower diagrams show same result using datum separation instead. The residuals are computed in the same points that were measured to allow for comparison.

D.3 Cross-Section X4

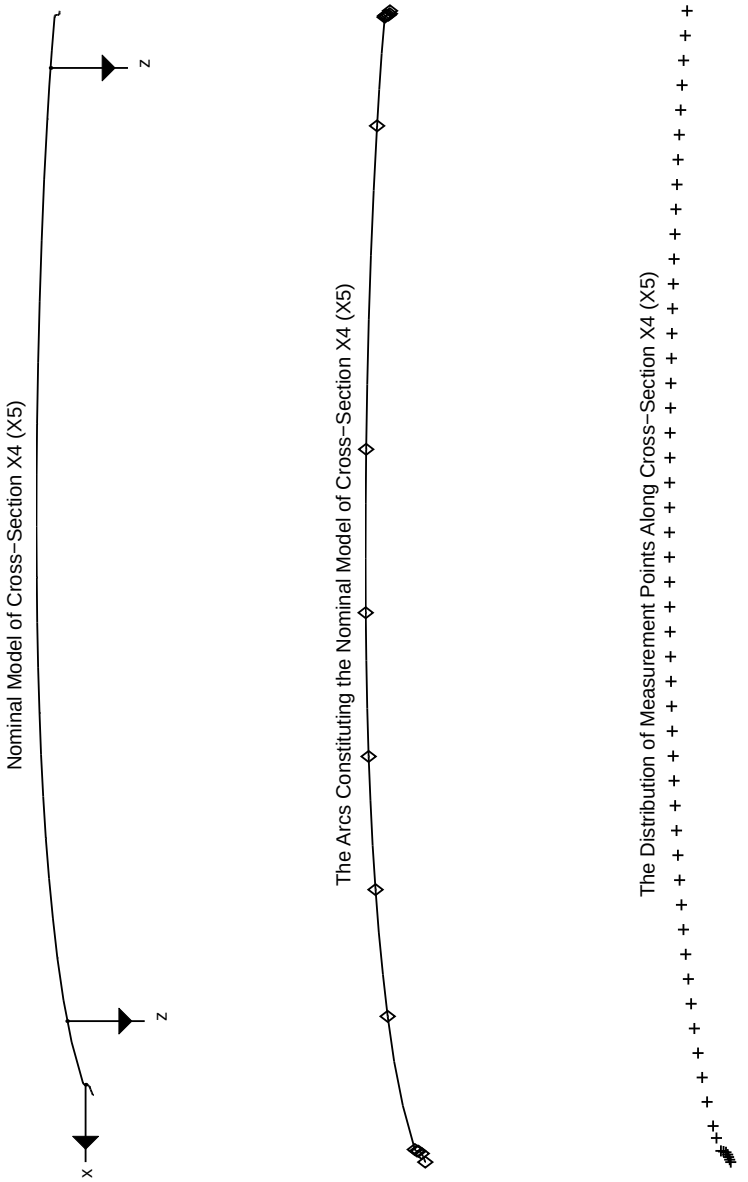

Figure D.7: A closer view of the cross-sections from Figure 4.13. Cross-section X4 is taken to represent X5 as well since are indentical. The pictures illustrate the appearance of the models as well as its composition of arcs. The points subject to inspection are also indicated.

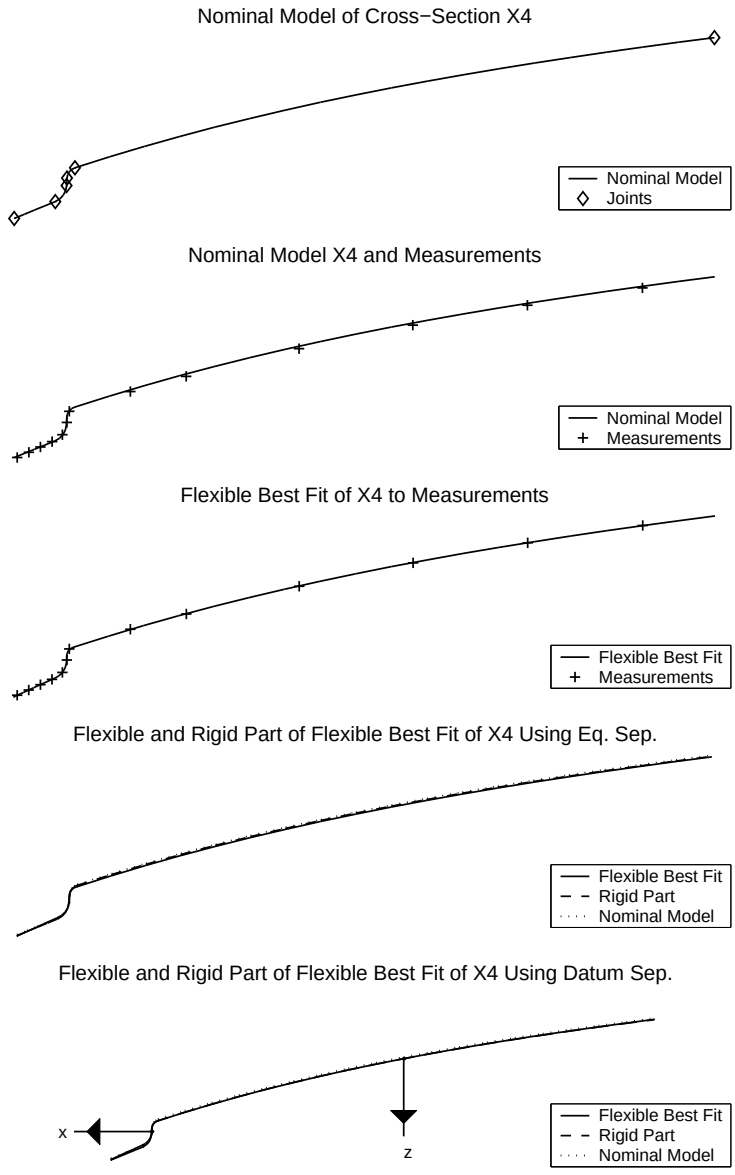

Figure D.8: The five pictures show the result of a flexible best fit of cross-section X4 to measurement data. From top to bottom they show the joints of the nominal model, the measurements, the flexible best fit, the rigid and flexible parts of the flexible best fit using equilibrium separation and datum separation, respectively.

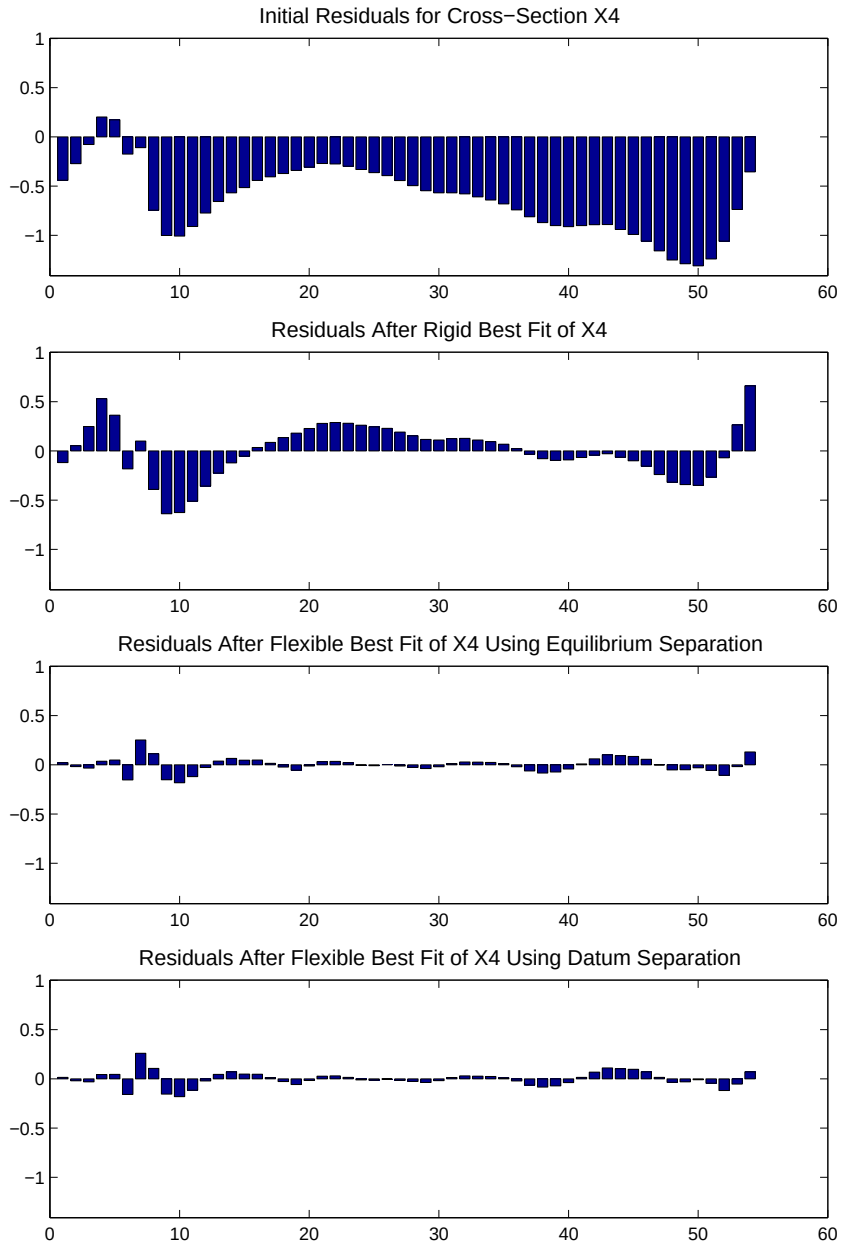

Figure D.9: The top diagram shows the initial residuals of cross-section X4. The second diagram shows the residuals after a rigid best fit to all measurements. The two diagrams at the bottom show the residuals after a flexible best fit with equilibrium separation and datum separation, respectively.

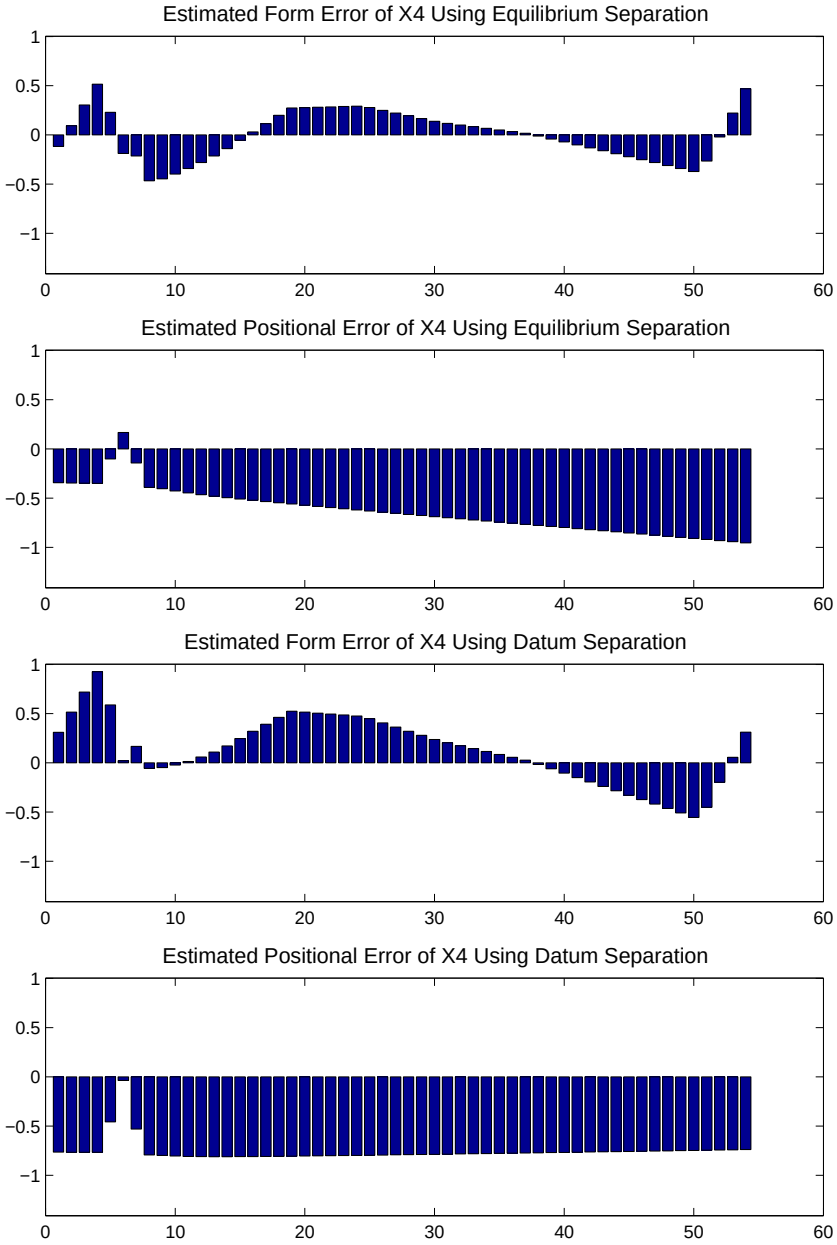

Figure D.10: The two upper diagrams illustrate the form and positional error estimated by the flexible best fit of X4 using equilibrium separation and the two lower diagrams show same result using datum separation instead. The residuals are computed in the same points that were measured to allow for comparison.

D.4 Cross-Section X5

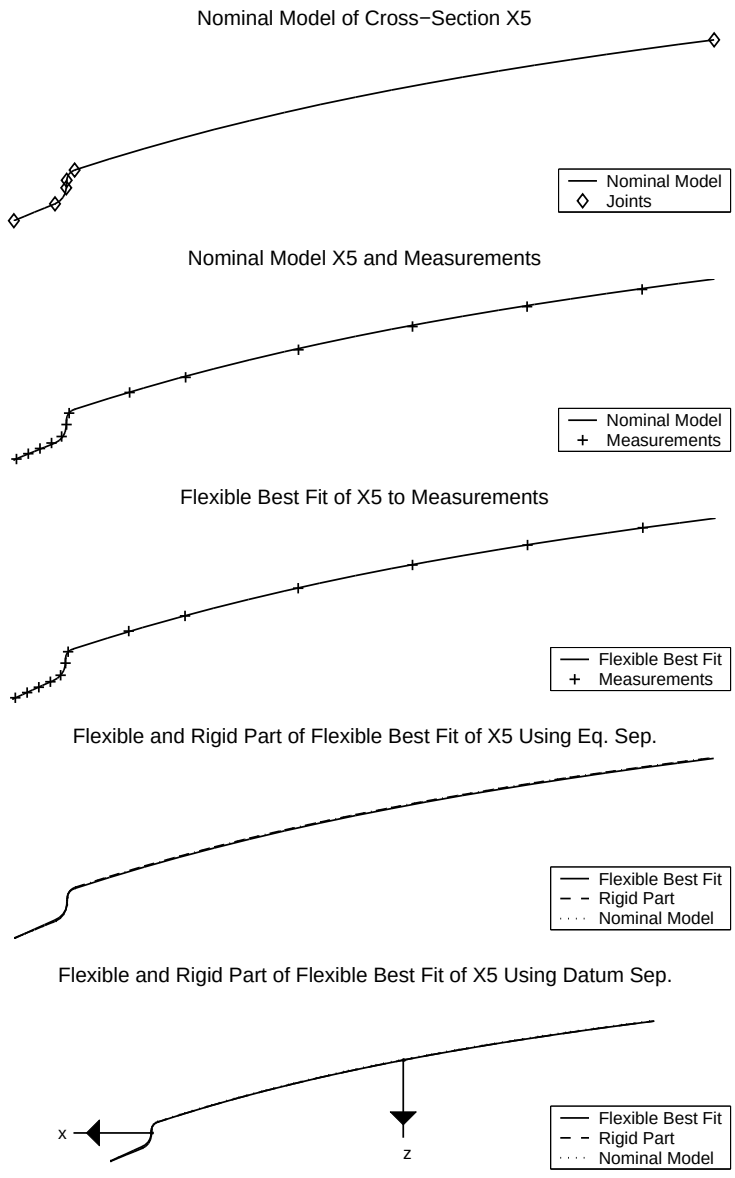

Figure D.11: The five pictures show the result of a flexible best fit of cross-section X5 to measurement data. From top to bottom they show the joints of the nominal model, the measurements, the flexible best fit, the rigid and flexible parts of the flexible best fit using equilibrium separation and datum separation, respectively.

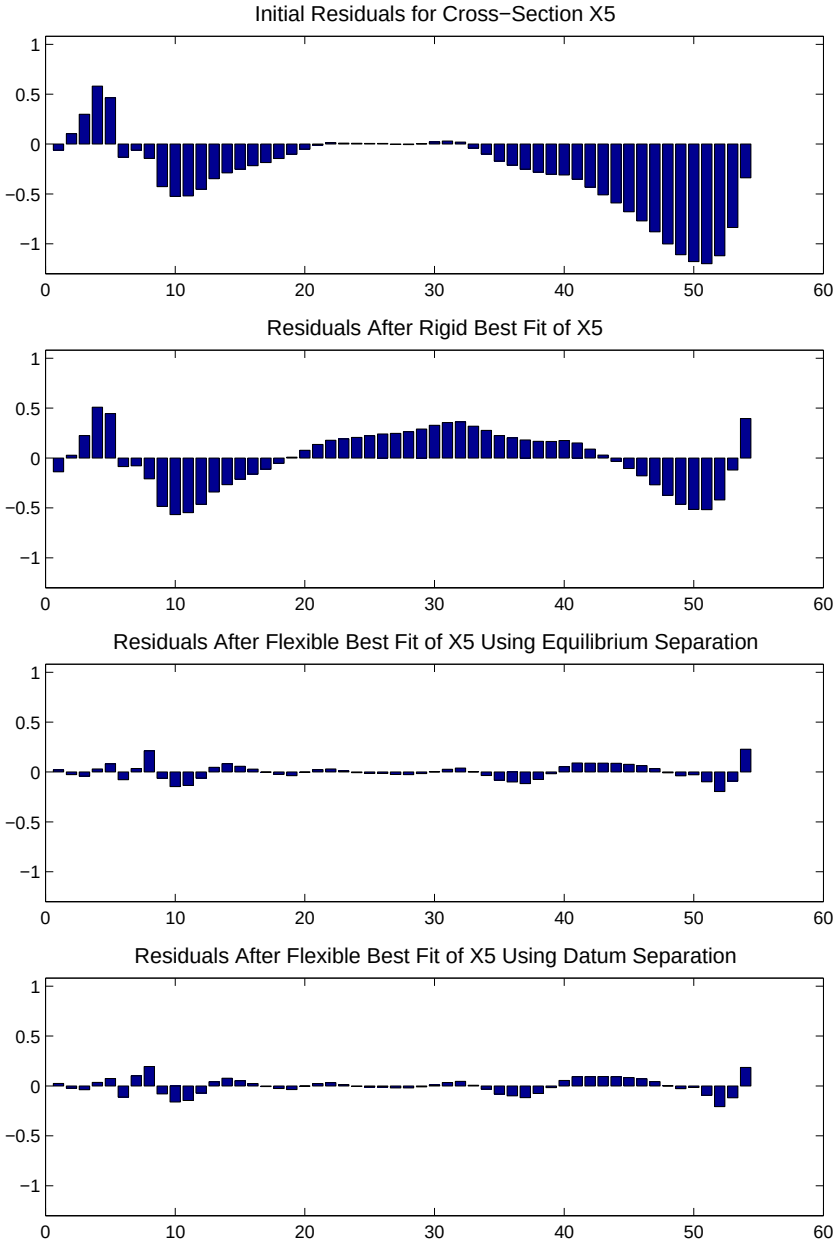

Figure D.12: The top diagram shows the initial residuals of cross-section X5. The second diagram shows the residuals after a rigid best fit to all measurements. The two diagrams at the bottom show the residuals after a flexible best fit with equilibrium separation and datum separation, respectively.

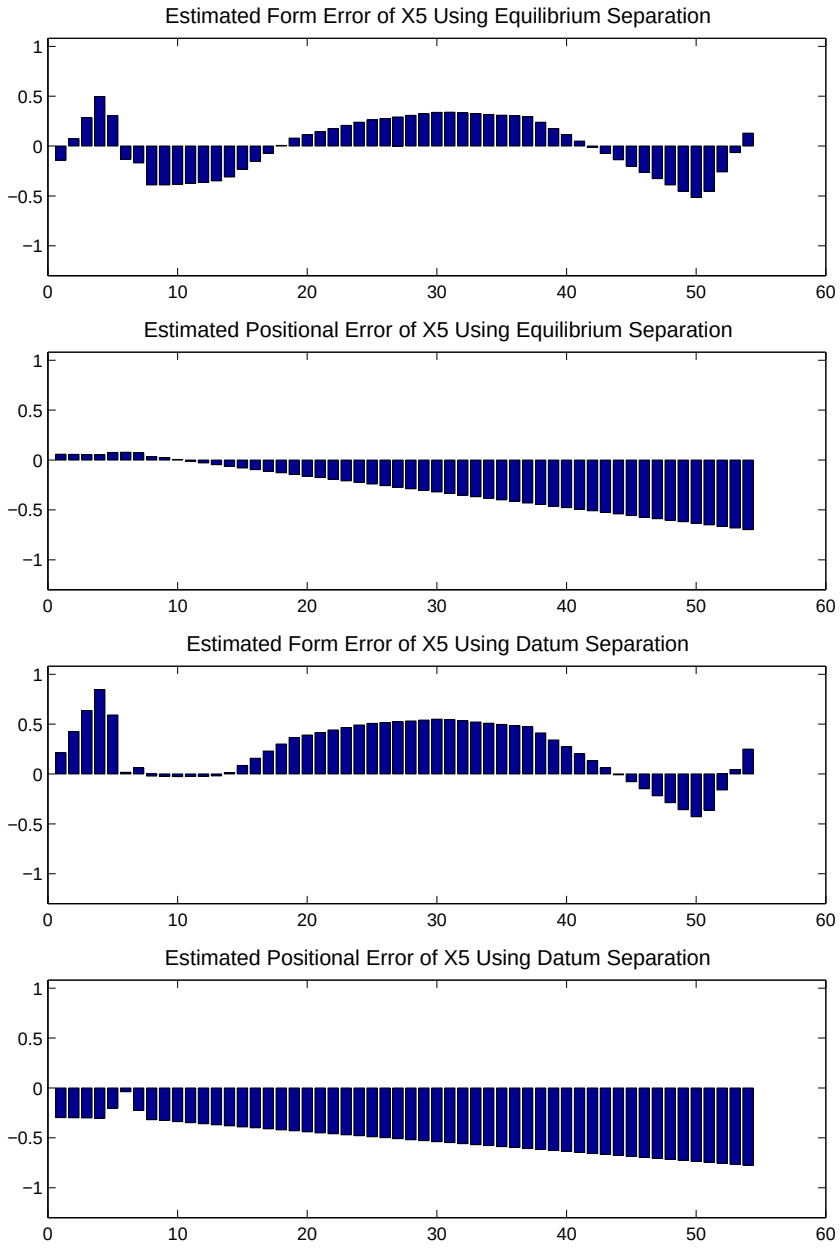

Figure D.13: The two upper diagrams illustrate the form and positional error estimated by the flexible best fit of X5 using equilibrium separation and the two lower diagrams show same result using datum separation instead. The residuals are computed in the same points that were measured to allow for comparison.

Bibliography

J. Angeles and A. Kecskeméthy. Fundamentals of rigid-body mechanics. In J. Angeles and A. Kecskeméthy, editors, *Kinematics and dynamics of multi-body systems*, pages 11–55. Springer-Verlag, Vienna, 1995.

C. L. Bajaj, F. Bernardini, and G. Xu. Reconstructing surfaces and functions on surfaces from unorganized three-dimensional data. *Algorithmica*, 19(1-2): 243–261, 1997.

Å. Björck. *Numerical methods for least squares problems.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

W. Cai, S. J. Hu, and J. X. Yuan. Variational method of robust fixture configuration design for 3-D workpieces. *J. Manuf. Sci. Eng.*, 119(4):593–602, 1997.

J. S. Carlson. Root cause analysis for fixtures and locating schemes using variation data. In F. van Houten and H. Kals, editors, *Proc. 6th CIRP int. sem. on comp. aid. tol.*, pages 111–120. Kluwer Academic Publishers, 1999.

J. S. Carlson and T. Ahlmark. *Production quality improvements using statistical and geometrical analysis of car body measurements.* Licentiate thesis, Dept. of Mathematics, Chalmers Univ. of Tech., 1997.

M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, 1976.

M. M. Dowling, P. M. Griffin, K.-L. Tsui, and C. Zhou. Statistical issues in geometric feature inspection using coordinate measuring machines. *Technometrics*, 39(1):3–17, 1997.

G. Farin. *Curves and surfaces for computer-aided geometric design.* Academic Press, Inc., San Diego, CA, fourth edition, 1997.

R. Fletcher. *Practical methods of optimization.* John Wiley & Sons Ltd., Chichester, second edition, 1987.

T. A. Foley and H. Hagen. Advances in scattered data interpolation. *Surveys Math. Indust.*, 4(2):71–84, 1994.

P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization.* Academic Press, Inc., London-New York, 1981.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

R. J. Hanson and M. J. Norris. Analysis of measurements based on the singular value decomposition. *SIAM J. Sci. Statist. Comput.*, 2(3):363–373, 1981.

H.-P. Helfrich and D. Zwick. A trust region algorithm for parametric curve and surface fitting. *J. Comput. Appl. Math.*, 73(1-2):119–134, 1996.

F. L. Hulting. An industry view of coordinate measurement data analysis. *Stat. Sinica*, 5(1):191–204, 1995.

ISO 1101. Geometrical tolerancing, 1993.

D. Johansson. *Variation of surfaces using differentiable non-linear transforms*. Licentiate thesis, Dept. of Mathematics, Chalmers Univ. of Tech., 1995.

D. Johansson. Virtual assembly analysis. Technical report, Volvo Car Corp., 1998.

R. A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*. Prentice Hall, Inc., Englewood Cliffs, NJ, third edition, 1992.

R. Ramsdale. The engineering zone. online, 1999. `www.flinthills.com/~ramsdale/ EngZone/metrol.htm`.

L.-P. Rivest. Some linear models for estimating the motion of rigid bodies with applications to geometric quality assurance. *J. Amer. Statist. Assoc.*, 93(442): 632–642, 1998.

W. Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.

STD 5026,2. Master location system, 1996. Volvo Corporate Standard.

J. W. Wesselink. *Variational modeling of curves and surfaces*. PhD thesis, Technische Universiteit Eindhoven, 1996.

H.-T. Yau. Evaluation and uncertainty analysis of vectorial tolerances. *Precis. Eng.*, 20(2):123–137, 1997.

D. S. Zwick. Applications of orthogonal distance regression in metrology. In S. V. Huffel, editor, *Recent advances in total least squares techniques and errors-in-variables modeling*, pages 265–272. SIAM, Philadelphia, PA, 1997.