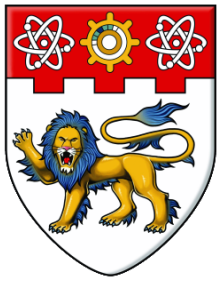
AUTOMATIC RECONSTRUCTION OF NOMINAL FREEFORM SURFACES IN AERO-COMPONENTS FOR REMANUFACTURING DUNG VAN THAN 2018



NANYANG TECHNOLOGICAL UNIVERSITY

AUTOMATIC RECONSTRUCTION OF NOMINAL FREEFORM SURFACES IN AERO-COMPONENTS FOR REMANUFACTURING

DUNG VAN THAN

SCHOOL OF MECHANICAL & AEROSPACE ENGINEERING

2018

NANYANG TECHNOLOGICAL UNIVERSITY

AUTOMATIC RECONSTRUCTION OF NOMINAL FREEFORM SURFACES IN AERO-COMPONENTS FOR REMANUFACTURING

DUNG VAN THAN

School of Mechanical & Aerospace Engineering

A Thesis submitted to the Nanyang Technological University

In fulfilment of the requirement for the degree of

Doctor of Philosophy

2018

Automatic Reconstruction of Nominal Freeform Surfaces in Aero-components for Remanufacturing

Acknowledgement

These four years have been the most unforgettable time for me. In this period, many things have happened to my family and me. There were many times when I felt lost in my life and wanted to give up. I would like to express my deepest gratitude to my academic supervisor, associate professor Tegoeh Tjahjowidodo. A gentleman who did not hesitate to try his best to help me overcome all my difficulties in PhD life. I would like to thank him for his patient guidance, continuous support as well as his assistance in all administrative matters related to Nanyang Technological University.

I would also like to express my thanks to my current and former co-supervisors, Dr Han Mingli, Dr Li Yuan Ping, and Dr Cheng Fang, for their continuous support and assistance in all administrative matters related to Advanced Remanufacturing and Technology Centre. Special thanks to Dr Cheng Fang, who helped me in accessing equipment for conducting my experiments.

I am grateful to Nanyang Technological University and Advanced Remanufacturing Technology Centre for providing the scholarship and an opportunity to successfully complete my objectives.

Sincere thanks to ARTC staff, Mr Xu Jian Xin, Ms Lim Pei Xian, Mr Bisma Mutiargo, for spending their valuable time helping me to conduct experiments and take necessary measurements.

I sincerely thank Mechatronics laboratory technicians, Ms Lee Koon Fong, Mr Seow Tzer Fook, Mr Ng Jui Hock, for creating a friendly work environment and helping me with administrative issues at the laboratory. Thanks to my friends in the Mechatronics lab for their continual support.

Special thanks to my roommates, Nguyen Hai Dang, Pham Quang Trung, Huynh Nam Khoa and other Vietnamese students for helping me with academic issues as well as family matters. Thanks to them I had a friendly atmosphere at home.

Eternal gratitude goes for my parents who encouraged and motivated me to finish my work, thanks to my parents for understanding that I had to stay away from them for extended periods. I sincerely thank my mother-in-law and sister-in-law who did not mind taking care of my son and my family.

Keeping the best for last, I can never express enough gratitude to my wife, Chu Thi Kim Huong and my son Dung Anh Quan. My son was the prime motivation behind me to work hard and finish the project as soon as possible. My wife was not only supporting me for doing my doctorate study, but she has also sacrificed a lot of things for our family. I owe her a lot and hope to repay that with infinite love.

Singapore, 10-August-2017

Dung Van Than

Abstract

Remanufacturing is a new branch of manufacturing industry in which used mechanical components are repaired and reconditioned to be used as new ones. In remanufacturing processes, used components were cleaned, filled in at the worn locations and reprofiled before being used again. Most of cases, the design geometries of the workpieces are not used as a reference because of many reasons, e.g. they might not be provided by suppliers and/or distortions of the workpieces during service life create discrepancies to the design geometries. Therefore, remanufacturing processes rely heavily on skilled workers.

In this project, we propose a framework to automate reprofiling processes in remanufacturing industries for freeform surface workpieces in the absence of design geometries. Many components, e.g. in aerospace industries, have complex geometries that are usually generated in freeform surfaces, such as turbine valves, blades, blisks, dies, molds, etc. This brings a significant challenge in automating the reprofiling processes. In the proposed framework, a workpiece (component) will be mounted on a reference table and initially digitized to acquire its geometries in the form of cloud data. Subsequently, the reference table and the Workpiece Coordinate System (WCS) are identified to transform the measured data to the WCS. A depth map image is then generated by resampling the measured data. Based on the B-spline fitting technique, an algorithm is proposed to automatically estimate the nominal profile surfaces of the components from the measured data. Some defects (e.g. from weld beads) are then directly obtained by comparing the estimated nominal surface with the digitized geometries. A tool path program then adaptively generates G-code for each identified defect and it is sent to a milling CNC machine to remove the defect.

The performance of the proposed method is finally evaluated using numerical simulation and real measurement data. The results confirm that the method is capable of exactly estimating the nominal profile surface and identify weld beads on freeform surface components with minimal human intervention.

Table of Contents

Chapter 1: Introduction

1.1 Background

The effects of global warming and climate change make people think twice about the way they use natural resources. It is generally accepted that to reduce the effects of climate change we have to decrease the consumption of natural resources such as minerals, fossil fuels, fresh water, etc., and increase the use of energy from renewable sources. Studies have shown that the manufacturing industries are responsible for 20% of total carbon dioxide emission [1]. Therefore, there is a need to transform these industries to make them use fewer natural resources and energy, and move towards sustainable manufacturing in the years to come.

Figure 1.1 illustrates two different approaches in production systems. Currently, the open loop system is adopted as shown on the left of Figure 1.1. In this approach, raw material and energy are directly taken from the earth. Products are created via material processing, manufacturing, and assembly processes. The products are subsequently distributed to end-users. After the products reach the end of their life, the retired products are treated as garbage and disposed into the earth. The right side of Figure 1.1 shows the second approach, which is a closed loop product cycle. It is similar to the open loop system, with the main difference seen at the stage after product retirement. Instead of being disposed, the products are disassembled into cores (components). The components are then sorted based on their conditions. Some components are treated as raw material, while some are reconditioned in the remanufacturing processes to return as like-new components, and others reused without any treatment. These treatments of retired products are important, as the recycling remanufacturing and re-using of retired products help to reduce the consumption of natural resources.

One of the key methods to make the production system move more towards sustainability is remanufacturing. Remanufacturing is defined as the process that disassembles retired products to cores, then refurbishes and reconditions the components that have reached the end of their life to a brand-new condition [2, 3]. Remanufacturing offers many benefits over traditional manufacturing processes in terms of energy and water consumption, production cost and carbon dioxide emission. It takes less energy to create the components (up to 85% energy used in new products

[4]), consumes 19-38% less water [3], lowers production cost by 45-65% of comparable new products [2] and emits 73-87% less carbon dioxide [5].

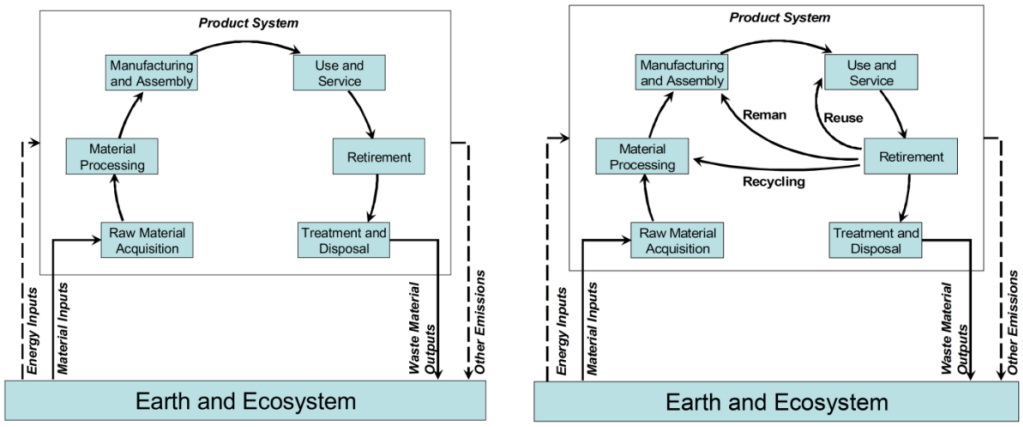


Figure 1.1 Open loop and closed loop products supply chain reproduced from [6].

Despite the benefits, remanufacturing only contributes about 2% of the total sale values of manufacturing markets [4]. This is due to some limitations, like the limitation of reverse logistics industries, government policies and commitments, and its technical problems [4]. In this study, the focus is on developing solutions to overcome the technical limitations and to contribute to the technical knowledge in remanufacturing processes that can benefit the manufacturing industries.

We shall now briefly mention the cores or components in remanufacturing industries. Based on the definition of remanufacturing, any kind of components or machinery such as cartridges of printers and photocopy machines [3], diesel engines [1], dies and molds [5], turbine blades [7], tires [8] etc., can be objects of remanufacturing.

Remanufacturing comes in after a product reaches the retirement stage. After the products' retirement, they are sent back to manufacturers for remanufacturing by reverse logistics services[1]. A typical remanufacturing process is explained as follows. Firstly, the product is dissembled to its cores or components. This step is usually carried out manually and it is time-consuming. Therefore, special design procedure is necessary to speed up the process [9]. Next, the components are cleaned to remove contaminants, e.g., oil, grease, dust, oxidized layer and coating layer, and expose the bare material. The cleaning processes usually employ high-temperature decomposition

[1] Normal logistics services help to distribute products from manufacturers to customers. Reverse logistics, in contrast, collects used products from customers to deliver to factories.

(plasma, laser cleaning), shot blasting or supercritical carbon dioxide cleaning [4, 10]. In the next step, cleaned components are inspected and sorted based on their conditions into three classes. The first class comprises components which can be directly reused without any processing, the second class contains components that are suitable for reconditioning, and the last class includes components which cannot be refurbished due to severe damage. In this latter case, they will be treated as raw materials for material recycling. For the components in the second class, a reconditioning process is applied. In this process, worn, damaged and cracked regions in the components are repaired by adding the same material to compensate for the missing material. Subsequently, redundant materials which are caused by over filling in the additive process are removed to return the final working surfaces of components. Finally, the components are reassembled, inspected and returned to service as new products.

1.2 Remanufacturing of freeform components

There are some difficulties in remanufacturing that hinder its development into a major industry. Firstly, remanufacturing is a labor-intensive process. Most of the remanufacturing work is carried out manually, from disassembly, inspection, additive manufacturing to re-profiling processes. Secondly, many components especially for those that interact with fluid flow during the operation, e.g. blades, blisks and ship propellers, comprise freeform complex surfaces with thin profiles. After certain operating hours, the components might be deformed, that makes the geometries slightly deviate from their initial geometries [7, 11-18]. Because of the deformation, the initial design geometries might not be able to be used as references for repair/recondition processes.

To reduce the dependency to manual labor and to ensure the consistency of the end products as well as improving working conditions for the operators, many studies can be found in literature focusing on the development of a repair process in the absence of the component design geometries, which commonly is referred to as adaptive machining process [7, 11-18]. The studies are not only motivated by the fact that the component geometry will vary throughout the operational time, but also, in some cases, the design information of some components may not be available either.

A typical adaptive approach via reverse engineering technique is illustrated in Figure 1.2. In the first stage, the components are disassembled from used machines (1). The components are then cleaned (2) to retrieve bare components. Subsequently, reverse engineering is employed to digitize the component to generate point cloud data (3). The point cloud data is manually processed by operators to reconstruct its nominal geometry (4). Based on the nominal geometry, any negative defects on the components (workpieces) will be filled by additive manufacturing processes (5), where the filled parts usually result in excessive material caused by over filling. The workpieces, therefore, need to be re-digitized to identify the geometric deviations to the nominal one after the additive process (6). Any discrepancies of the current workpiece geometry and the nominal geometry are attributed to the excessive materials that need to be removed to re-profile the components (7).

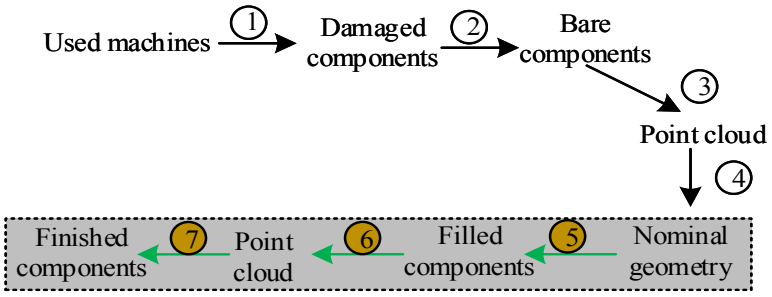


Figure 1.2 Adaptive approach in remanufacturing

In the adaptive machining approach, the first four steps are carried out manually. The last three steps can be automatically conducted. Step 4 is usually implemented on virtual environment of reverse engineering software such as Polyworks, CATIA, etc. The most critical step in the adaptive machining is the reconstruction step (4). In this step, point cloud data is manually processed to eliminate defective data to generate the nominal geometries. Because the reconstruction of the nominal geometry is highly dependent on skilled workers, the final workpiece geometry will then inherit any errors in the nominal geometry reconstruction step. Consequently, as the most critical step, the nominal geometry estimation might become the source of bottlenecking in the entire remanufacturing flow. This leads to a requirement to automate the data processing and the nominal geometry reconstruction.

In many cases, the freeform components might suffer large defects on their tips/edges as illustrated in Figure 1.3, which consequently cause excessive problems. For

example, after failure of main bearings, the tip of a turbine blade might be worn out and bent causing a secondary damage on the engine casing as the blade tip might rub the casing. The edges of a blade (either the front or trailing edges) might get damaged because of collision with foreign objects. These types of large damage obviously will change the component shape and, conventionally, repairing such a component requires its design information as the reference for the process, while the process itself undeniably relies on skilled operators.

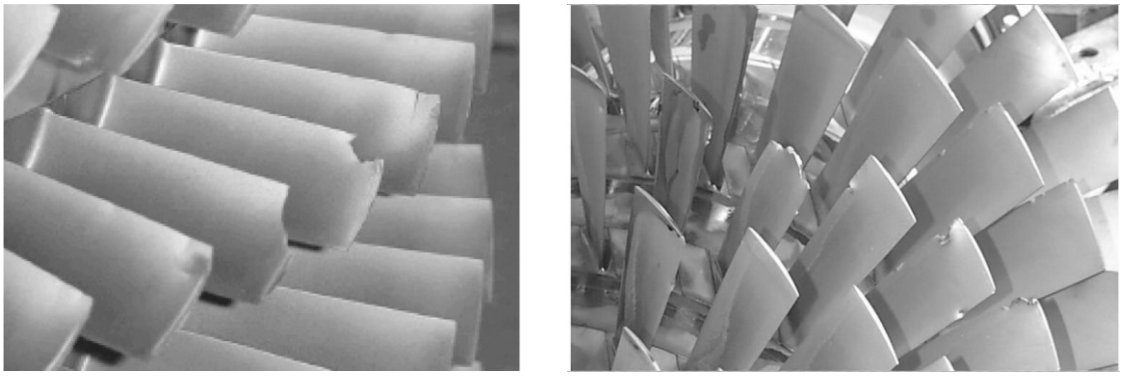


Figure 1.3 Large defects on compressor blades caused by rubbing casing and hitting foreign objects [19].

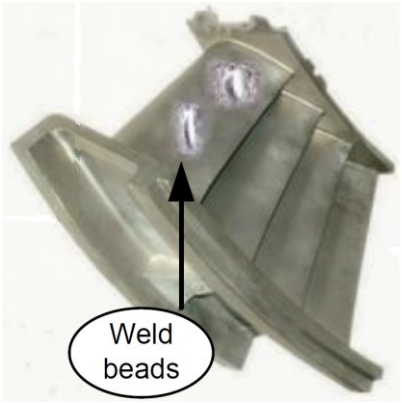


Figure 1.4 Damaged vane with small defects on its freeform surfaces

As illustrated above, for a component with large defects, the adaptive machining approach is seen to be the most suitable approach. However, in a case of components with small defects (see Figure 1.4), the adaptive machining approach does not seem to be the best approach. Even though it can deal with small defects, the approach requires digitizing the entire component to reconstruct the nominal geometry. Such process is time-consuming and inefficient for components with minor damages. The challenges

faced in small damage remanufacturing process and the associated workflow will be elaborated in the next section.

1.3 Problem statement

The challenges posed against adopting the adaptive remanufacturing process for small defects also provide ample scope for improvement. The way the workflow differs for remanufacturing small defects can be understood with the help of an example. A common prospective application of adaptive remanufacturing can be the repairing of surface imperfections of rotor blades of turbines. During turbine engine maintenance progress, if a few blades of a rotor stage developed cracks, all of the blades in the same rotor stage would be retired [20]. This means that many blades having only micro-cracks or very small imperfection on the surfaces are also retired. All the retired components are considered as stock for remanufacturing. A proper repairing process could restore the damaged component to a like-new component.

Adaptive machining approach as one of repair processes requires skilled workers to process the digitized data to identify the damaged portions or imperfections on the surfaces. This process is necessary to estimate the nominal data at the damaged areas, when the nominal geometric information is absent, to fully reconstruct the whole workpiece geometry. As an illustration, Figure 1.4 shows a blade/vane with small imperfection located in the middle of the freeform surface. In this case, the reconstruction of the nominal geometry is rather straightforward as we only need the surrounding surface geometry to interpolate the nominal surface at the defect location. Once the nominal surface has been reconstructed, the repair process can be proceeded to recondition the component.

In general, the reconditioning process of components with small defects can be illustrated as shown in Figure 1.5. A retired system is first sent back to a factory to disassemble its components. After that, visual inspection is conducted to identify small defects on every component (1). The identified damaged components are subsequently cleaned and any coating layer is removed to expose the bare material (2). Next, fluorescent penetrant inspection is applied to identify any micro-cracks on the surfaces of the components. Subsequently, negative defects are repaired, typically with an additive manufacturing process (3), thereafter, a heat treatment is applied to stabilize the microstructure of the material and to relieve any thermal stresses. Upon the

completion of the additive manufacturing, some excessive materials are created on the surface as a result of overfilling. These positive defects, therefore, need to be removed to finalize the surfaces reconstruction.

The process of removing the material protrusion is also known as a reprofiling process. Unlike the first three steps that must be manually performed, reprofiling can be automated with adaptive machining approach. A robotized surface restoration is known to outperform manual approach in terms of time and manpower cost. Prior to the process, part digitization is conducted to acquire point cloud data of the defective surface geometry (4). Subsequently, the point cloud is manually processed to reconstruct nominal geometry of the component (5), after which the nominal geometry is used for tool path generation in, e.g. CNC machine (6).

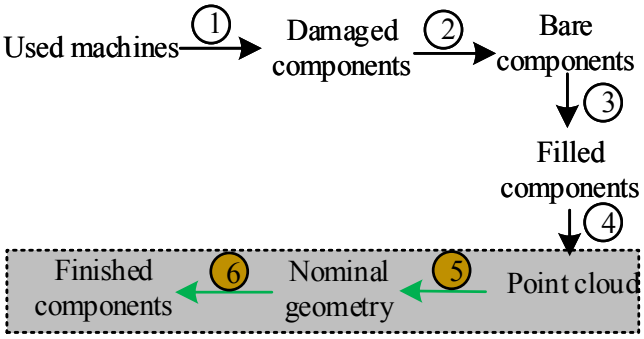


Figure 1.5 Automatic remanufacturing processes for small defect portion on components

As discussed earlier, the use of an adaptive machining process is inefficient for small local defect removal in freeform components. This brings up a need of the development of an automatic algorithm that can reduce the dependency to skilled workers in the reconstruction of nominal geometry.

The three first three steps on the repair process of parts with small defects, namely component disassemble, cleaning and missing material compensation, are rather challenging to be automated. In particular, this is due to the bottleneck problem in the reprofiling step. Therefore, the main focus on automating the repair process will be attributed to the automatic reprofiling stage after the additive manufacturing step. As the types of component in remanufacturing processes are very diverse, this thesis focuses on components with freeform surfaces, e.g. in aerospace engines, power plants, or dies and molds, with more concentration on those comprising small defects on the surface as illustrated in Figure 1.4.

As discussed earlier, the main concern in remanufacturing of components with small defects is on the automation of reprofiling stage. In this thesis, the concept and approach to automatically reconstruct the nominal surface of defective components in the absence of design geometries will be proposed. The proposed approach will allow for reducing the dependency on skilled workers and automated reprofiling to restore the nominal surfaces after the additive manufacturing process.

1.4 Objective, approach and contribution

1.4.1 Objective

The main objective of this thesis is to develop an algorithm for automatic nominal geometry reconstruction of defective freeform components after additive manufacturing steps in remanufacturing. This algorithm is targeted at components without any prior knowledge on the design geometries. The objective of this research is, therefore, twofold. First, to characterize any existing defects on a freeform surface as a basis for automatic nominal surface estimation algorithm, which includes excessive materials identification, e.g. weld beads on surfaces. The second objective is to implement the estimation algorithm for tool paths generation in machining processes as a demonstrator for the proposed approach.

1.4.2 Approach

Remanufacturing involves many kinds of components and this thesis focuses on the automation of the machining stage for freeform surface components. This is illustrated by a case of a repaired component after some preparation processes (e.g. additive process), where its freeform surfaces still contain some excessive materials (e.g. weld beads) that need to be removed.

Due to the lack of knowledge on the design geometry, the machining process has to be adaptively performed on the components themselves. Therefore, geometrical information of the components needs to be digitized and the nominal geometry has to be estimated for further processing.

Unlike common surfaces such as cones or planes that are deterministic, freeform surfaces do not have any specific forms. In its design stage, a freeform surface is usually defined in a spline form. Therefore, to estimate the nominal geometries of the components, a spline function will also be used to fit and process the measured data.

However, the measured (digitized) data does not give the entire information about the nominal geometries as it also contains defect profiles. The nominal geometries will be estimated based on some pieces of the fitted B-spline curves that correspond to the nominal surfaces. In this case, the pieces of the fitted B-spline curves have to be properly chosen from the entire workpiece surfaces (that include the defect profiles) based on some criteria, e.g. the curvature of nominal geometries, curvature of defects and error of a spline curve fitting.

Once the nominal surfaces are estimated, the defect information can be obtained through geometry subtraction of the measured and the estimated ones. Adaptive tool paths to machine the defects are subsequently generated automatically by using the defect information via surface offsetting technique. The tool paths are converted to G-code and sent to a milling machine to remove the excessive material.

Figure 1.4 summarizes the approach for remanufacturing components that contains small defect portions. For large defect portions as shown in Figure 1.2, the first two steps are identical. In step (3), as the defects are small, it is possible to carry out the additive manufacturing manually to fill negative defects. The components geometries are then digitized (4) to acquire point cloud data for further processing. The last two steps will be automatically processed. In step (5), an automatic program is used to analyze the data and subsequently estimate the nominal geometries. The reprofiling process (6) is performed in a CNC milling machine with the tool paths that are automatically generated based on the subtractive geometries of the measured point cloud data and the estimated nominal geometries.

1.4.3 Contributions

There are three main contributions in this thesis:

The first one is attributed to the enhancement of the B-spline fitting technique. B-spline is widely used in design to define freeform shapes. Hence it is chosen as the best candidate for many reverse engineering works to reconstruct geometries from measured data. The B-spline fitting techniques for smooth curves and surfaces are well developed, but the techniques to handle the data that contains non-trivial points, such as kink points, cusp points or discontinuity points etc., are still challenging. Very often, in the digitized data of remanufactured components might contain not only smooth curves but also some non-trivial points due to irregular geometry of defects.

Currently, the common B-spline fitting techniques to handle such non-trivial data are based on stochastic optimizations. As stochastic optimizations are computational costly, applying optimization processes for real-time data might not be feasible. This thesis proposes a new B-spline curve fitting technique based on deterministic optimization that can handle any kind of curves from smooth to non-trivial one.

The second contribution concentrates on the development of an algorithm to automatically estimate nominal geometries and identify defects from the measured data, replacing the conventional manual processes. The outcome will lead to the reduction of labor cost and minimization of product inconsistencies. Furthermore, the algorithm is not only limited to defect identification but also reverse engineering.

The last contribution is the proposed framework for automatic machining processes to automatically remove defects on freeform surface components applied in remanufacturing industries in the absence of a priori knowledge in their design geometry. This thesis proposed a data processing flow from digitized geometries to tool paths for CNC milling machines with minimal human intervention.

1.5 Thesis structure

This thesis has seven chapters to present step by step approach to respond to the research question and to fulfil the aims of this study on the technical solution for automatic remanufacturing of freeform surface workpieces in the absence of design geometries. Following the description of the problem statement in this chapter, brief summaries of each chapter are given as follows:

Chapter 2 provides an overview of the supporting knowledge related to this research work. Firstly, this chapter will start with a discussion on an adaptive machining approach that is used to repair components in the absence of the design information, explaining the steps involved such as geometries digitization, virtual repairing to generate nominal geometries in a computer environment, identification of defects and generation of tool paths. Secondly, a review of freeform surface representation in designing of components is presented. As B-spline is the most common function to describe freeform surface, details of the function are provided. Because adaptive machining is an extension of reverse engineering that depends on the fitting technique to reconstruct components geometries, a review on B-spline fitting technique is also

provided in this chapter. The chapter also reviews some state-of-the-art techniques in automatic defect detection and estimation of nominal surface. Finally, this chapter reviews a technique and procedure to generate adaptive tool paths for machining purposes.

Chapter 3 proposes a novel method to fit curves by a B-spline function. The B-spline function is a powerful function to represent freeform curves and surfaces. As the technique to fit a spline curve for non-trivial cases is not well developed, this chapter presents a new approach to fit a B-spline curve with any kind of curve, from smooth curves to curves with discontinuity. The fitting method is based on a two-stage fitting algorithm. The first stage separates measured data into spline pieces and the second stage solves the optimal location and continuity of the connection point. Some experiments are carried out and the results are compared with the existing method in the literature.

Chapter 4 presents the algorithm to estimate a nominal surface from measured data and identify defects or weld beads on freeform surface without human intervention. To identify nominal surfaces, proper nominal data needs to be selected from measured data. B-spline curve fitting technique is used to select nominal data based on an estimation of the minimum radius of curvature and the minimum length of spline pieces.

Chapter 5 provides a framework for automatic machining by applying the work described in Chapter 3 and Chapter 4. Some relevant algorithms on the implementation of the framework are also discussed, e.g. automatic identification of workpiece coordinate system, transformation of measured data to the new coordinate system, data interface between different steps, and tool path generation for three axis CNC milling.

Chapter 6 presents the experimental setup and discusses the results obtained from the proposed method in Chapter 5. There are two sources of data used in this chapter. The first one is synthesized data that is created by sampling a CAD model in Solidworks and the second one is the real data that is measured from test components. This chapter describes in detail the results obtained when applying the proposed method. The experimental results are also validated with design data to show the abilities of the method.

Chapter 7 draws some general conclusions from this research work and proposes some recommendations for further improvement in the development of automatic machining system.

Chapter 2: Literature review

Manufacturing of a new component, essentially will deform the geometry of the raw material, e.g. from ingot/bar/rod material, to other geometry of the final product. On the other hand, remanufacturing starts the process from used components and results in finished products, where the overall geometries do not differ essentially. This chapter reviews some typical damages on freeform components which need to be repaired in remanufacturing and typical approach to refurbish the damaged components.

2.1 Defects and imperfections on freeform components

In this section we present typical conditions of components after retirement (before remanufacturing). Components with freeform surfaces are very common in machineries exposed in high dynamics fluids such as propellers in vessels, airplanes, or (turbine/compressor) blades in gas turbine engines. Many of those freeform components are operating in high rotational speed with extreme centrifugal and bending forces. Such components are designed with minimum thickness to reduce the moment of inertial. After certain operating hours, the components might deform from their initial geometries [11, 13, 15]. If the change of the geometries is within the acceptable limit of tolerance, it will not bring excessive harms to the system. However, if the deviation is not well-treated earlier, it might cause catastrophic incident to the system.

Carter [21] reviewed typical damages on turbine/compressor blades/vanes. Similarly, the defects on turbine engines are also illustrated in turbine engine handbook [19] and training document [22]. We can roughly categorize the damages into two types. The first type of damages changes the overall dimension of a component due to material loss such as tip rubbing, leading and trailing edges deformation due to foreign object hitting or burning. Figure 2.1 shows some typical damages of this type. The second type of damages can be considered as small damages or surface imperfections. In general, the small damages locally located on the surface. The appearance of the small damages does not affect functionalities of the components. ISO standard 8785: 1998 [23] defines common terms and types of small damage on surface. Figure 2.2 lists some typical surface imperfections such as scratches, cracks, pores, dents, buckles and

pits. The damages do not affect the performance of the components; however, they create stress concentration that causes cracks initiation and propagation.

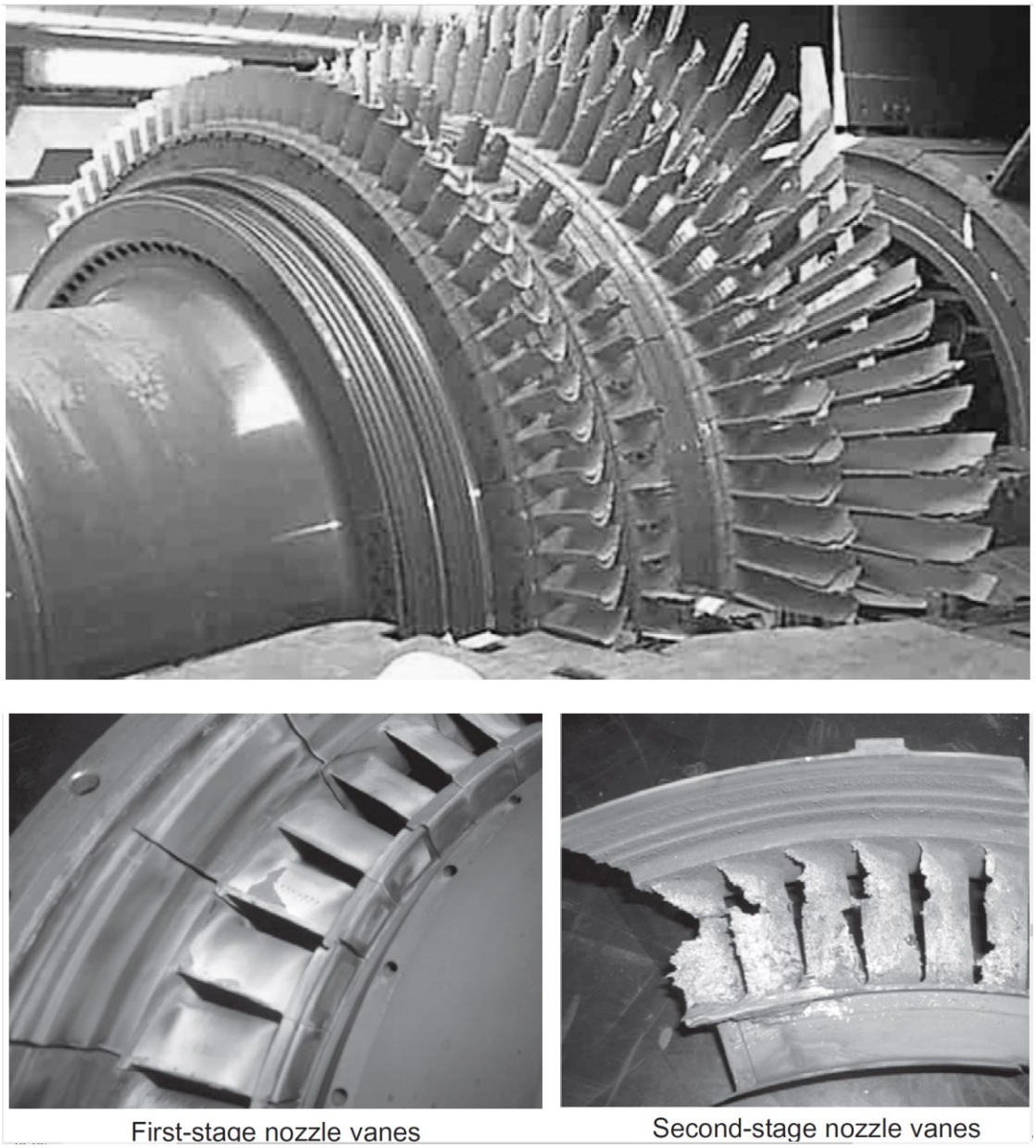


Figure 2.1 Damaged turbine blade/vane after accident. The top panel: the blades are burned due to over-temperature. The two bottom panels: burned nozzle vanes due to over fuel supply [19]

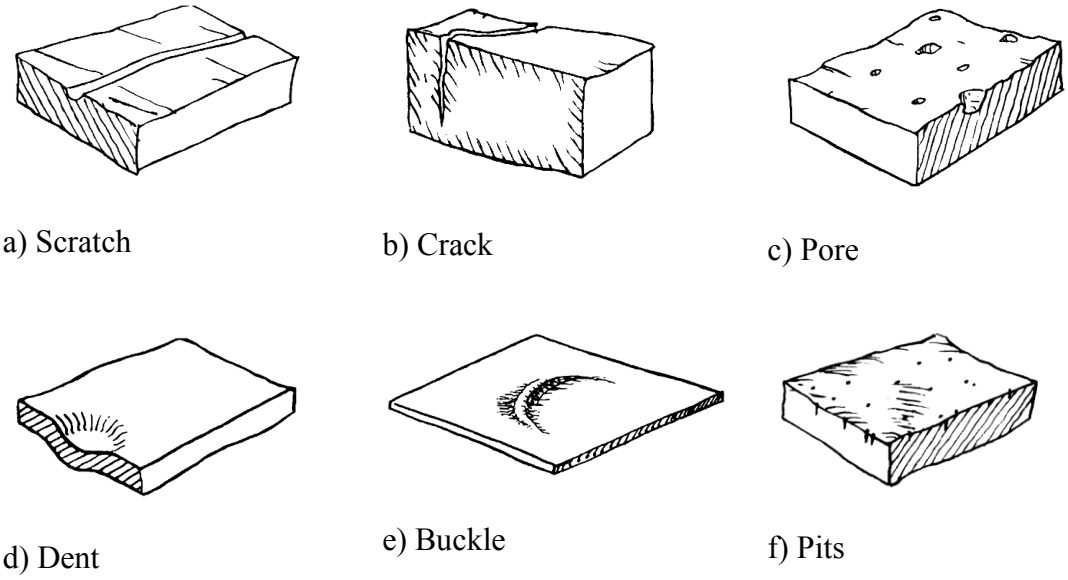


Figure 2.2 Some selected cases of surface imperfection [23]

There are two cases, where the components are screened to retirement. The first case happens after machines break down due to accident. And the second case, when they are screened during periodic overhauling.

For the repaired component is pertaining to an accidental break down, the damage, most likely, will affect other critical parts such as bearings, sensors etc. This event can lead to catastrophic damage as large damaged components might not be repairable. Consequently, those components will only be recycled to make new components. Nevertheless, when the operating condition of the large damaged component are not so critical and the internal structure of the components are not too complex (e.g. no hollow structures), up to a certain extent, the components might be still repairable using adaptive machining approach.

In literature, there are many studies to use adaptive approach for refurbishing large damage components. Gao et al. [11, 12] employed reverse engineering to reconstruct nominal geometry of a damaged compressor blade to repair tips, edges and surface damages. US Patent No. US 7472.478 B2 [15] presents apparatus and standardizes the method to repair tip and leading edges of an airfoil blisk via adaptive machining approach. Similarly, Jones et al. [24] reported a system to repair a tip damage on a blade using reverse engineering.

In the second case, when components are inspected during periodic overhauling, the components are screened for small damages on surfaces, tip and edges. These damages do not directly result in machine failures. The machines can continue to work without replacing the components, however, the life time is becoming uncertain. If those are not treated immediately, they lead to crack initiation and propagation, causing the machine to fail in the near future. Moreover, in turbine machines, if small damages, such as pits, burn points, corrosive points, are detected in a few blades in the same rotor stage, entire blades in the stage would be retired [20, 22]. These kinds of small damaged component are still feasible to be remanufactured.

Figure 2.3 illustrates typical surface damages on turbine/compressor blades/vanes in gas turbine engines [22]. The damages might be caused by impacts with foreign objects, corrosion by chemicals, burn at high temperature, etc. Dents are formed when a blade is hit by a foreign object and normally dents appear in the front face of a blade. Pitting corrosion occurs when the engine runs in a corrosive chemical environment. Pitting corrosion creates sharp and narrow cavities on surfaces, which consequently creates stress concentration and causes crack development. Cracks appear due to stress concentration under centrifugal force when the blades are running. According to [22], visible cracks are definitely unacceptable in the blades and those have to be discarded. Burn is a common defect in turbine blades and vanes due to over-heating. Burn damages usually appear in the leading and trailing edges of a blade/vane.

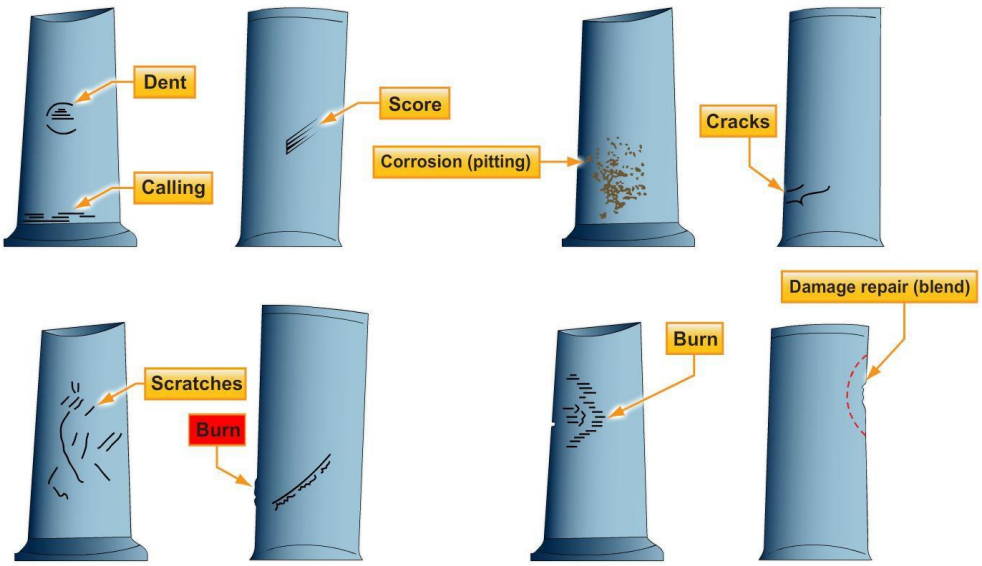


Figure 2.3 Typical damaged blades (reproduced from page 10-9 [22])

Repairing of large damaged components, such as tip, front and trailing edge damages, burn etc., requires overall nominal dimension information and geometric extrapolation of the missing portions. Those processes are usually carried out manually by skilled operators in adaptive machining processes. Therefore, large damages are not the objects for autonomously repairing system in this thesis.

On the contrary, small damage or surface imperfection of nominal geometries can be easily reconstructed from the surrounding data. In these cases, an automatic algorithm to reconstruct the nominal geometries can potentially be developed. This thesis only focuses on small surface damages, located in the middle of freeform components.

2.2 Repairable limitation

To recondition used components, any dimensional discrepancies of the components exceeding the tolerance need to be repaired. The material properties of the components might degrade due to wear and tear, hit by a foreign object, high temperature exposure, pitting corrosion by chemical, etc. In particular, any changes that involve microstructural variation might also affect the mechanical strength of the components. Micro-cracks create stress concentrations that initiate crack propagation that will reduce the fatigue strength. Therefore, up to a certain degree, visual inspection is still required to identify large damages (lost material) in the components, while fluorescent penetrant inspection is used to inspect micro-cracks before remanufacturing processes.

During maintenance, damaged components are screened during a visual inspection to select only components that satisfy certain criteria for repairing without affecting the mechanical properties. Figure 2.4 illustrates the allowable repair limits of turbine/compressor blades. The allowable limit does not apply uniformly to the entire area, it depends on the functionality of each part of the component. For instance, the two corners at the tip of the blade can be filleted with the maximum radius of 6-8mm depending on the turbine blades. The damage size on the edges of the blades (area C, E in Figure 2.4) can be larger than the ones on the freeform surfaces (area B). The size of the damage on the fillet area (area D) is strictly controlled with the maximum depth of defects of 0.2mm.

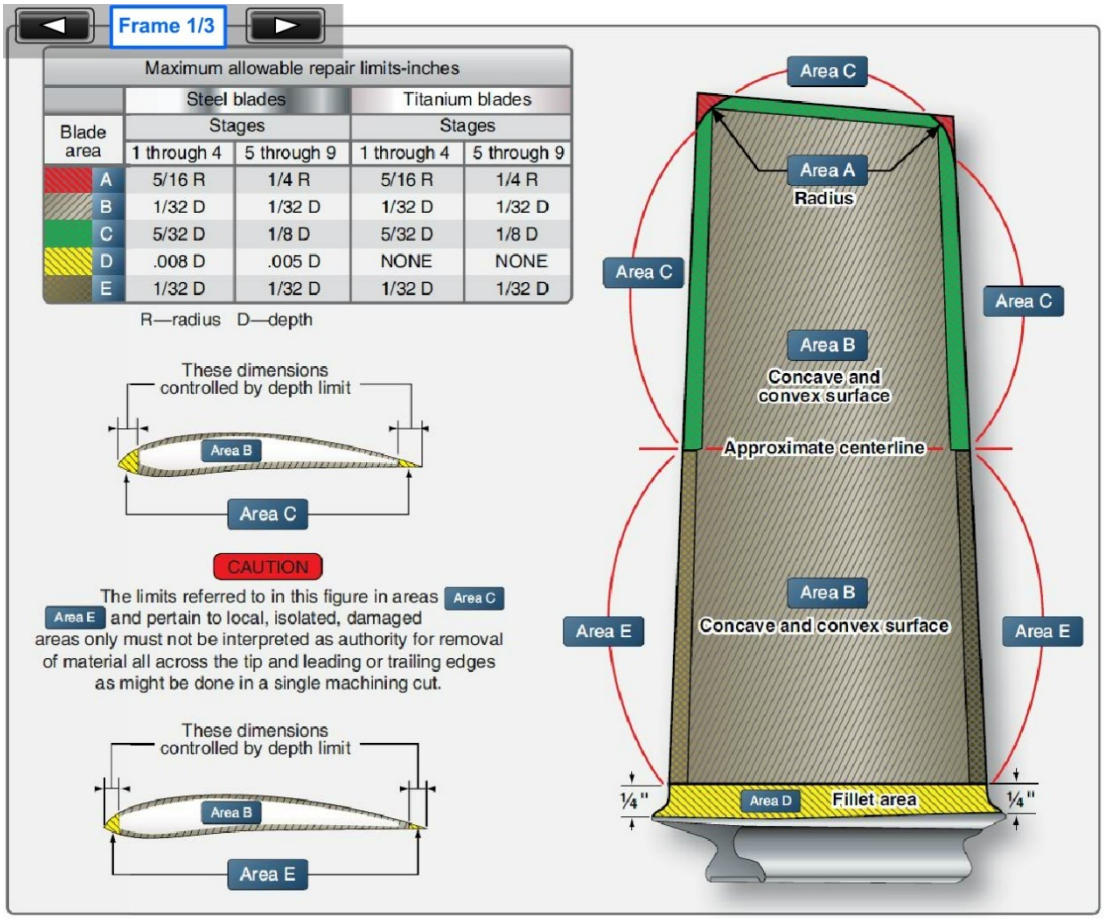


Figure 2.4 Maximum allowable repair limits of turbine blades. (reproduced from page 10-10 [22])

In theory, a damaged component is repairable if after conditioning, geometry of the component satisfies tolerance, and mechanical strength of the remanufactured one is equivalent to that of the original product. Decision on repairing a component depends on two factors. The first factor is the ability of the existing technology to recondition the component. For instance, when the damages are present on the internal channels of turbine blades and vanes that comprise complex hollow internal structures for cooling purpose under extreme high temperature operation, the damages will be difficult to repair. Even more the inspection of internal cracks is still a big challenge with the existing technologies. In this case, recycling of the components is still the common solution. The second factor to be considered is the cost of the reconditioning work. If the repair cost, that includes equipment, labor, time, material, is too high, the component would be discarded and replaced with the new one.

Illustratively, the repairable limitation numbers indicated in Figure 2.4 are the state-of-the-art in the current repair technology. Any enhancement in the additive manufacturing technology would increase repairable dimension of the components.

Because decision of the repairable limits depends on too many factors and are unique to the component and application, it is out of scope in this study. In the thesis, we assume that the small damaged components are repairable with the current additive manufacturing technology. The thesis only focuses on the following step of the additive process, i.e. reprofiling to remove any excessive materials on the repaired component to reproduce the nominal surface geometry.

2.3 Automatic (re)manufacturing

Automation is one of the key factors that contribute to improving productivity, reducing production cost, creating friendly working conditions for humans etc. In the last few decades, there is a demand for expanding industrial automation that is not only limited to traditional industry sectors such as automotive, electronics equipment, etc., where low-mix high-volume products are required, but also to high-mix low-volume applications that are currently conducted by skilled workers.

Remanufacturing is becoming a new trend in manufacturing industry with many benefits. Remanufacturing also sets a requirement to improve its productivity and working conditions for workers. Automation in remanufacturing is one of the keys in helping to expand industries in future. This section provides a review on current development of automation on machining and repairing processes, which might become a foundation for the development of automatic remanufacturing. We also provide a brief overview of repairing process approaches for freeform surfaces workpieces.

2.3.1 Automatic machining processes

Conceptually, in automatic machining, the processes should be performed by a machine or a group of machines with minimum human intervention. A CNC milling machine is a simple example for performing automatic machining. In the literature there are many studies to make machines smarter, to handle complex tasks which currently are carried out by skilled workers. To automate the machining process, multiple sensors (e.g. force, vision, vibration, etc.) are embedded in the system to help the machine "to see" the workpieces. Li *et al*. [25] reviewed current trends in automatic machining and some approaches to make machining processes more

intelligent. The following works are selected as examples to demonstrate of automatic machining.

Closed-loop machining cell for turbine blades. A fully closed-loop system for turbine blade manufacturing was developed by Katz *et al.* [26-28] to remove two types of common defects (die lines and positive defects) of cast turbine blades. Defects on the blade surfaces were characterized by inspection systems that were configured with a camera and a laser sensor. The camera was used to coarsely and quickly identify defect locations using on image processing. At the second state, a laser distance sensor was used for finer inspection based on the camera results. The defects were then identified by comparing the CAD model of the workpiece with the inspected data. Based on the inspected data, a robot was used to machine on the identified defects locally. The inspection system continuously examines other defects and the machining procedure was repeated until no defect was found. In this case, the closed-loop machining cell is targeted to fully automate the manufacturing to replace skilled workers. However, the system is based on the designed CAD model that might not be available in many remanufacturing processes.

Disk weld bead grinding system. An automated robotic system for weld bead grinding was developed by Whitney *et al*. [29]. In this system, a structured light system and a camera were used to locate the weld beads in real time for tracking the robot. Furthermore, weld bead features were also monitored to optimize the cutting parameters based on a Material Removal Rate (MRR) model. In order to achieve high-quality surfaces, force controller was also implemented on the robot. A similar system is also utilized for optimizing a grinding process [30]. The use of robots has also shown success for other manufacturing works such as stub grinding, deburring [31], drilling [32] polishing [33-35] etc.

2.3.2 Adaptive machining approach

As stated in Chapter 1, most of remanufacturing processes are carried out manually. That leads to human dependency and difficulty in product expansion. Furthermore, due to the geometrical distortion during service life, the CAD models of the workpieces might no longer be feasible to be used for repairing. That is why Reverse Engineering (RE) is usually used for automating remanufacturing processes.

RE can be defined as "systematic evaluation of a product with the purpose of replication" [7]. In remanufacturing industries, the typical repairing processes are illustrated in Figure 2.5. A damaged part (workpiece) is firstly digitized to acquire the point cloud data by using contact or non-contact measurement. An RE software such as Polyworks, powerSHAPE, CATIA or PRO-E, is used to process the measured data to create the CAD model of the part. The CAD model can be used for designing and generating CNC code or for Finite Element analysis.

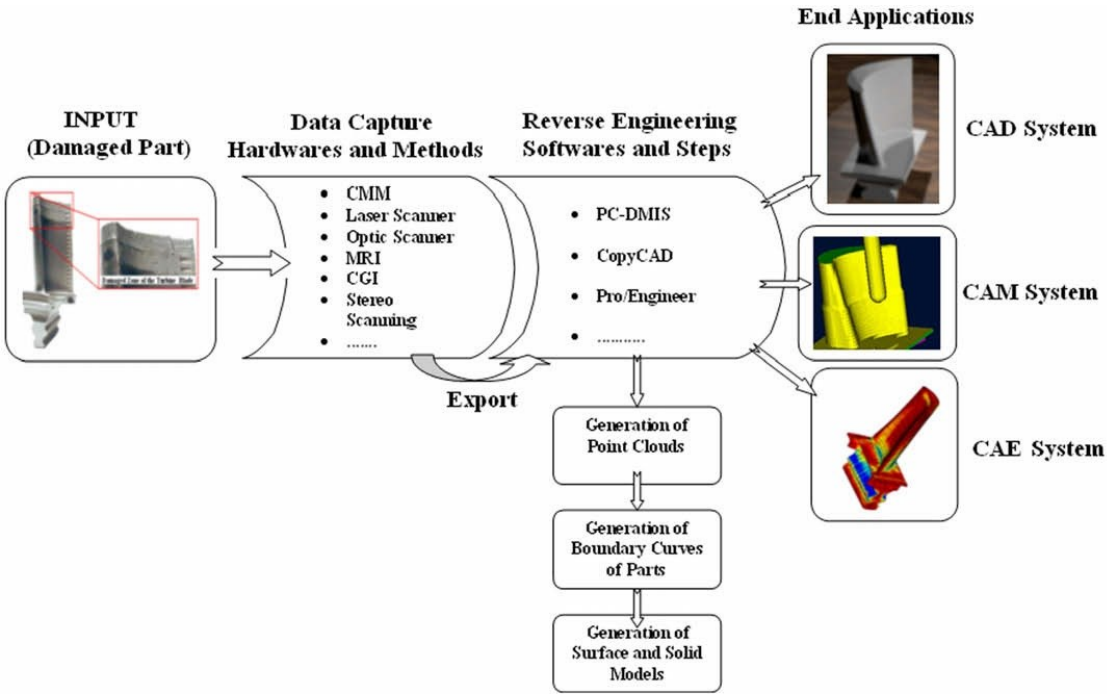


Figure 2.5 Reverse engineering approach for remanufacturing of freeform surface components, reproduced from [7]

Jones *et al*. [24] proposed an integrated system for remanufacturing of high-value products (turbine blades) based on the designed CAD models named RECLAIM (REmanufacturing of high-value products using a Combined LAser cladding, Inspection, and Machining system). The turbine blade is mounted on a 5-axis milling machine. Point cloud data is obtained through a scanning probe, which is mounted on the spindle of the milling machine, while the machine is controlled by on-machine inspection software (PowerINSPECT, Delcam plc, UK). The point cloud data is then processed to generate a CAD model (real part) by using powerSHAPE software. The measured CAD data is then compared to the nominal CAD model to generate the CNC

milling program via powerMILL software. RECLAIM offers a benefit on data processing because the workpiece is mounted on the machine without changing the coordinate system during machining. However, in the inspection system, the CAD model of the workpiece is still required for generating the CNC program, while the defect detection is still carried out manually.

In practice, after retirement, the geometry of a freeform component tends to vary from its original geometry, e.g. because of working stress, or thermal distortion [11, 12, 36-38]. This brings difficulties in employing the design geometry from the reference model for repair processes. Adaptive machining is a solution to repair the component by the following procedure. First, the geometry of the component is digitized by using a Coordinate Measuring Machine (CMM) or laser scanner, or 3D camera. Afterwards, the measured data is processed to remove any defective points and restore the missing data (holes) by geometry interpolation from surrounding data points using RE software. The 3D model of the component is reconstructed from the measured data, to be used as a reference model for tool path generation in the machining processes [11, 12, 36-38].

2.4 B-spline

B-spline is a well-established piecewise polynomial with good properties. It can easily capture various functions from continuous curves to discontinuous ones. The use of B-spline to approximate or to fit a complex function or a given data set became a popular research topic in 1970s to 1990s. The research commonly focused on finding the best smooth spline to represent complex functions or sampled data. Recently, there are some needs in reverse engineering applications to employ B-splines not only for smooth curves, but also curves with discontinuous points, kink points, cusps or turning points.

In almost all curve fitting applications, to define a B-spline function, the data is split to identify the knots, which represent the breaking/end points of each piecewise function. Subsequently, a least square method is applied to calculate the control points to fully determine the B-spline functions or, alternatively, the curve fitting is formulated as a non-linear optimization problem of the knots and control points. However, the optimization problem commonly leads to a multimodal case, which results in local optima and causes significant challenges. In a case when the data forms a smooth

curve, many existing methods are available to solve for the knots without any problem. But, when the curves contain non-trivial cases (discontinuous points, kink points, cusps, turning points), the optimization process is commonly approached by using a stochastic optimization, which might be inspired from biological systems such as genetic algorithm, neural network or artificial immune system etc. The downside of these approaches, however, is the excessive computational time.

2.4.1 B-spline curve

A p-degree B-spline is a parametric piecewise polynomial, which is defined as:

$$S(t) = \sum_{i=0}^{m-p-1} N_{i,p}(t) P_i \tag{2.1}$$

where P_i represents the i^{th} control point and $N_{i,p}(t)$ is the i^{th} p-degree B-spline basis function which is defined based on sequential knot vector $\mathrm{Z} = \langle \zeta_j \rangle_0^m \left(\zeta_0 = \zeta_1 = \cdots = \zeta_p < \zeta_{p+1} \leq \zeta_{p+2} \leq \cdots \leq \zeta_{m-p-1} < \zeta_{m-p} = \zeta_{m-p+1} = \cdots = \zeta_m\right)$, and m is the number of knots in the B-spline function. A p-degree B-spline basis function $N_{i,p}(t)$ is defined in a recursive series [39, 40]:

$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1 & if\ \zeta_i \leq t < \zeta_{i+1} \\ 0 & otherwise \end{cases} \\ N_{i,j}(t) &= \frac{t - \zeta_i}{\zeta_{i+j} - \zeta_i} N_{i,j-1}(t) + \frac{\zeta_{i+j+1} - t}{\zeta_{i+j+1} - \zeta_i} N_{i+1,j-1}(t) \end{aligned} \tag{2.2}$$

As stated above, the knot vector, Z, is a non-decreasing sequence. The first and the last $(p+1)$ knots are identical to provide the boundary condition of a B-spline curve. The knots from ζ_{p+1} to ζ_{m-p-1} correspond to the interior knots, which can be single or multifold-knots. If a knot ζ_i has η multiplicities, the spline curve at the corresponding knot location would be continuous to C^k with $(k = p - \eta)$. This is a relevant property to employ B-spline to represent a non-trivial case, i.e. kink points, discontinuous points or turning points.

When the interior knots have a uniform distance from its neighbors, we have a uniform knot distribution. In a case where the distances are varying, we have non-uniform knots. A notable unique property of this type of knot is that a uniform knot can only

represent smooth curves, while a non-uniform one can represent not only smooth ones, but also non-trivial cases such as kink or discontinuity.

Even though B-spline offers good properties, it can only approximate conic curves, which are very common in the real world such as circles, parabola and ellipse. To represent such kind of conic curves, a Non-Uniform Rational B-Spline (NURBS) is used. By adding weighting factor, w_i, to each control points, P_i, and changing the curve form to a rational form, the B-spline curve can exactly represent conic ones. A NURBS curve is similarly defined as:

$$S(t) = \sum_{i=0}^{m-p-1} \frac{N_{i,p}(t)w_i}{\sum_{i=0}^{m-p-1} N_{i,p}(t)w_i} P_i = \frac{\sum_{i=0}^{m-p-1} N_{i,p}(t)w_i P_i}{\sum_{i=0}^{m-p-1} N_{i,p}(t)w_i} \tag{2.3}$$

Based on the property of the basis function, which says that the summation of all basis functions at any points is equal to unity, i.e. $\sum_{i=0}^{m-p-1} N_{i,p}(t) = 1$, if all the weights $(w_i = 1)\Big|_0^{m-p-1}$ are equal to one, the NURBS curve would become the normal B-spline. That confirms B-spline curve as a specific case of the NURBS curve. The computational time to evaluate a NURBS and a B-spline curve are almost equal. Nevertheless, a NURBS curve is more flexible than the B-spline one.

2.4.2 B-spline surface

A B-spline surface has a parametric equation with two parameters *u* and *v*. It is defined as:

$$S(u,v) = \sum_{i=0}^{m-p-1} \sum_{j=0}^{n-p-1} N_{i,p}(u) N_{j,p}(v) P_{ij} \tag{2.4}$$

where $N_{i,p}(u)$ and $N_{j,p}(v)$ are B-spline basis functions as defined on a rectangular mesh of two knot vectors $\mathrm{Z_u} = \langle \zeta_j^u \rangle_0^m \ \left(\zeta_0^u = \zeta_1^u = \cdots = \zeta_p^u < \zeta_{p+1}^u \leq \zeta_{p+2}^u \leq \cdots \leq \zeta_{m-p-1}^u < \zeta_{m-p}^u = \zeta_{m-p+1}^u = \cdots = \zeta_m^u\right)$ and $\mathrm{Z_v} = \langle \zeta_j^v \rangle_0^n \ \left(\zeta_0^v = \zeta_1^v = \cdots = \zeta_p^v < \zeta_{p+1}^v \leq \zeta_{p+2}^v \leq \cdots \leq \zeta_{n-p-1}^v < \zeta_{n-p}^v = \zeta_{\mathrm{n}-p+1}^v = \cdots = \zeta_n^v\right)$.

We also have the definition of a NURBS surface, which mathematically is presented as:

$$S(u,v) = \sum_{i=0}^{m-p-1} \sum_{j=0}^{n-p-1} \frac{N_{i,p}(u)N_{j,p}(v)w_i}{\sum_{i=0}^{m-p-1}\sum_{j=0}^{n-p-1} N_{i,p}(u)N_{j,p}(v)w_i} P_{ij} = \frac{\sum_{i=0}^{m-p-1}\sum_{j=0}^{n-p-1} N_{i,p}(u)N_{j,p}(v)w_i P_{ij}}{\sum_{i=0}^{m-p-1}\sum_{j=0}^{n-p-1} N_{i,p}(u)N_{j,p}(v)w_i} \quad (2.3)$$

B-spline, Bezier and Conic are special cases of the NURBS.

2.4.3 B-spline curve fitting

In data fitting by spline, a knot vector is commonly defined in advance. Subsequently, the control points are identified based on the minimization of the least squared error between the data points and the function. Knots are usually chosen in uniform space or by Chebyshev points [39] or based on the change of radius of curvature [41]. However, uniformly spaced knots might result in overshooting when the curves contain non-trivial cases, e.g. turning points, cusps, kink points, discontinuous points or inhomogeneous smooth curves. In order to overcome the problem, a non-uniform knot space (free knots) is introduced. Unfortunately, optimizing the number of knot and their respective locations in a non-uniform space is a challenging problem and it is computational costly.

A common way in determining the free knots is by predetermining initial knots, followed by a specific method to modify the knots, e.g. shifting knot locations and increasing or decreasing the number of knots. The approach can be roughly categorized into two classes. The first class is started by predefining a small number of knots (usually without interior knots), and subsequently, new knots are inserted until the fitted curve satisfies a certain criterion. In the second class, denser knots are pre-determined with multiple redundant knots. A subset of knots is then selected from the initial ones by eliminating less essential knots. The remaining knots after the elimination process are the final knots for the spline fitting.

In the knot insertion class, knots are usually obtained through a bisecting method. The method scans the data from left to right to identify the largest subinterval of data that can be represented by a polynomial function or a parametric function without violating a certain criterion such as fitting error. There exist some notable methods in the literature for the fitting functions and the acceptance criterion. Grove *et al*. [42]

employed a Bezier curve to regress the local data and quantify the fitting error of the mean square error of the fitted curve to determine the knots. Rice [43, 44] used polynomial to treat the local data and to quantify the fitting error by predetermined error tolerance based on a selectable norm and the method is intended for function approximation. Ichida *et al*. [45] also used the bisecting method, but instead of using a fitting error tolerance, they suggested to use "trend"[2] criterion to automatically fit the data by a cubic piecewise polynomial. Tjahjowidodo *et al*. [46] used a parallel algorithm for bisecting the second derivative of the data to identity a linear piecewise function in determining the knot using a cubic spline fitting. Plass *et al*. [47] used dynamic programming to subdivide the data for identifying knots for parametric cubic spline fitting. Park *et al.* [48] employed dominant points instead of the direct use of knots in B-spline fitting problem and, subsequently, the knots are obtained by back transforming the dominant points. In the paper, they employed the insertion algorithm to add more dominant points until the fitted curve satisfies the predetermined error. In addition, there are also some methods that are based on knot insertion such as in [49, 50].

In the second class, where the algorithm starts with denser knots, different methods can be found on how the initial knots are created. Lyche *et al.* [51, 52] used all sampled data points as the candidates for the initial knots for knot removal strategy. He *et al*. [53] used wavelet transform to identify locations of high frequency points and put the knots at those locations to generate initial knots. Kang *et al.* [54] used a jumping distance of the third derivative (in case of cubic spline) to select the initial knots via sparse optimization through the application of Lasso optimization. Another method, that is also based on the third derivative, was discussed in [55]. Yuan *et al.* [56] used a set of multi-resolution basis functions with Lasso optimization to select the basis functions, which can compose the given data and subsequently create the initial knots from the selected basis function set.

Apart from the two aforementioned methods in determining the knot locations, there also exist some approaches that employ optimization processes. Optimization is shown to be superior in the identification of the knots location, which leads to high fitting

[2] Trend criterion is an inequality of fitting residuals r_k and the number of data number of data in the fitting function, $\sum_{k=p+1}^{q} r_{k-1} r_k \geq \sqrt{(q-p)} \sum_{k=p}^{q} \frac{r_k^2}{q-p+1}$. The trend criterion indicates where the residuals behave as random variables.

quality. However, as a price for high quality fitting results, this approach suffers from long computational time as all optimization tools for solving non-linear problems involve some iterative processes. In order to reduce the computational cost, the number of knots needs to be predetermined in advance. We can roughly classify the optimization tools into two categories i.e. the stochastic approach and deterministic approach.

In the deterministic approach, Schwetlick *et al*. [57] used Gauss-Newton method to solve a non-linear least square problem for knots identification. Randrianarivony *et al*. [58] employed Levenberg-Marquardt algorithm to solve the same problem, which will (only) result in local optimum knot locations. In finding global optimum knots, Beliakov [59] employed the cutting angle method for fitting the global free knots spline.

On the other hand, the stochastic approach has been used in various studies. The Adaptive Free-Knot Splines (AFKS) by Miyata and Shen [60, 61] used evolution algorithm to find the optimum knots. Zhao *et al*. [62] employed Estimated of Distribution Algorithm (EDA). Genetic Algorithm [63] by Sarfraz *et al*., while Gálvez *et al*. [64] used Elitist clonal selection algorithm and Particle Swarm [64] for selection of knots. The Artificial immune system is also used by Ülker [65] for free knot placement.

2.5 Geometric representation

A detailed analysis in mathematical tools and methods to represent geometries is a key to help finding a good method to fulfill the aim of this study. This section firstly reviews some methods that we use for geometric modeling in design. Secondly, a review of geometric representation in reverse engineering applications is discussed, which includes methods for geometry digitization, and surface reconstruction from measured data.

2.5.1 B-spline for curve and surface representation in computer aided design

B-spline was first introduced in 1946 by I. Schoenberg for uniform cases. Later in 1947, H. Curry extended it to non-uniform cases [66]. At that time, B-spline was

represented in a divided difference form, which is numerically unstable. In 1974, C. de Boor rewrote the basis function in a recursive form (2.2) and it gained popularity [66].

After Non-Uniform Rational B-spline (NURBS) was introduced, it became a standard for representing and exchanging data of different formats. In 1980, IGES format used NURBS for geometry representation because NURBS can represent most of the popular curves and surfaces such as conic, Bezier, spline, etc. Nowadays, all CAD design platforms employ NURBS curves and surfaces as basic tools for design geometry. CAD models now can be easily created through some basic geometry construction methods such as extrusion, sweep, loft, etc.

In reverse engineering applications, we need to reconstruct CAD models of objects from their digitized geometries. For cases of curves or surfaces constructed from simple shapes such as lines, circles, planes, cylinders or spheres, common mathematical models can be used to fit the geometry. On the other hand, in a case of freeform curves and surfaces, NURBS is normally used to represent those in standard CAD software. The reconstructed geometries should be used the same definition functions as they are. However, NURBS is a rational function that is highly nonlinear that causes some difficulties in the fitting processes. Therefore, in reverse engineering applications, freeform curves and surfaces are usually approximated by B-splines.

2.5.2 Geometry digitization in reverse engineering

Digitization of 3D objects is a common task in many industries such as art sculpture restoration, quality inspection of products, reverse engineering, etc. The main purpose of the digitization is to obtain the three-dimensional objects' geometries as detailed as possible to accurately reconstruct the model. In the digitization processes, three-dimensional coordinates of sampled points are acquired to create a point cloud by using a measurement device.

We can categorize the devices based on the working principles. The first category acquires the location of a sampled point through a contact probe, while the second one acquires the coordinates of sampled points through non-contact sensing.

In contact measurement systems, Coordinate Measuring Machines (CMM) [67, 68] or milling machines [24] are used to hold the probe for the measurement. There are two types of probe, namely touch and scanning probes. The prior type acquires a discrete point when it is triggered, while the latter results in continuous positions when the

probe scans the objects' surfaces. The contact measurement devices provide very high positional accuracy (single point trigger gives better accuracy than the scanning type). However, they have at least three disadvantages. Firstly, they have low sampling rate. Since the machines only acquire a single sampled point at time, it is very costly for geometry digitization. The second concern is related to the measurement space. The machine is not portable and has a fixed size measuring table, therefore the objects to be measured have to be brought to the measuring machines and the dimension of the object to be measured is limited. Furthermore, the machine is also incapable of measuring soft surfaces.

Non-contact measurement methods minimize the mechanical contact between measuring devices and objects' surfaces by utilizing different principles such as optical principle and mechanical sound (ultrasonic). The non-contact measurement methods usually provide faster digitizing speed but lower accuracy.

Optical measurement systems are mainly used in geometric digitization processes [27, 69-73]. Two popular methods are used to obtain the sampled points, i.e. light triangulation and time of flight methods. Light triangulation method measures the angle between an incident light beam and its reflective light beam for calculating the distance between the sensor and the sample point through the triangular relation. The triangulation method can easily be found in laser scanning devices or 3D cameras. It usually results in high resolution (micrometer level) but has a limited measurement range. While the time of flight method measures the traveling time of a light beam from the sensor to sampled points and the reflection. As the velocity of the light is very high, the method is used to measure big objects located far from the measurement devices with low resolution result (up to centimeter level).

In reverse engineering applications, due to the requirement of high accuracy, the light triangulation method is usually used in the digitization. Both laser scanners and 3D cameras are widely used. While laser scanners provide higher data accuracy, 3D cameras offer much faster sampling speed (over a million sample points per second). It is noted that laser scanners can only acquire a single point or a line when the measurement is triggered. Therefore, to acquire a surface geometry, it needs accurate fixtures with three to six degree-of-freedom to obtain consistent readings. On the other hand, 3D cameras can digitize a surface at a single trigger. It might not need an

accurate fixture and the measured data can also be easily stitched for wide objects. 3D camera can be mounted on industrial robots to perform the measurement [74, 75].

2.5.3 Geometric reconstruction from measured data

After digitization, the point cloud needs to be processed to generate the 3D models of the objects. These 3D models store the objects' information and can be used for further analysis such as evaluating object geometry, generating tool paths for 3D printing and analyzing stresses when the objects are under loading. A few formats may be used such as polygon meshes (triangular surfaces), and NURBS models.

Polygon meshes are widely used to store digitized geometries as they are simple and easy to handle for data interpolation. In practice, to create polygon models, the point cloud data is usually processed using some reverse engineering software, such as Polyworks, Solidworks, Catia to create triangle facets by connecting close points together following some rules. The initial facets are needed to perform various tasks, including object correction to fill holes, delete faulty facets. Finally, the 3D (STL) model is straightforwardly generated from the facets [76].

Although polygonal surfaces contain all the data points and the formatting is easy to handle and control, it also has some disadvantages. The first disadvantage is that the data volume is large. The second is the G^0 continuity of the facets. An application may require higher continuities in the surfaces. The best way to overcome the limitations of the polygonal surfaces is using parametric surfaces, such as NURBS surfaces, to approximate the measured data. The use of NURBS will reduce memory usage and will improve the processing time. There are a few steps to construct NURBS surfaces from the measured point cloud. Firstly, the point cloud data is segmented into a number of subsets. Secondly, each subset is fitted with a parametric surface (B-spline) and finally, the B-spline surfaces are converted to NURBS form to store in a standard format such as IGES, STEP [77-82].

2.6 Defect detection and nominal surface estimation

This project aims to develop an automatic inspection system capable of identifying defects on freeform surface parts and estimate the nominal surface of the parts at the defect sites to generate tool path for defect removal. This section discusses three topics. First, we review work in the literature that uses machine vision to automate

inspection. The second covers current approaches in data processing to identify defects in the measured geometries. Finally, a review on how to estimate the nominal surfaces in the absence of design data when there are defects on the measured data is conducted.

2.6.1 Defect identification

There exist some discussions in literature, where computer vision is widely used for automatic defect detection in production lines via image processing technique. In [83], a computer vision system was used to identify defects on cylinder blocks by varying illumination angle of the light sources and it was shown that a few types of defect, e.g. crashes, holes, bubbles and dirty pain, can be identified successfully. A camera system was also used to identify defects on bearings in production lines [84], to localize defects on rail tracks [85], or to identify defects in welding products through an x-ray based NDT (Non-Destructive Testing) test [86]. However, the aforementioned applications are mainly aimed at assessing the product quality rather than localizing and identifying the defects.

There are some applications where the detailed geometric information of the defects needs to be automatically identified, especially in automatic machining applications. A robotic system for an automatic welding bead grinding process is illustrated in [29, 30]. In this system, a structured light and camera are utilized to obtain 3D information of the workpiece surfaces. A dedicated algorithm is then used to extract the information of weld beads to automate the grinding process for the removal of the identified weld beads. Localization and identification of the weld beads were carried out by simply evaluating the slope changes in the measured spatial data as the algorithm is devoted to flat workpiece cases. Automatic weld bead identification was also studied in [87, 88] for motion planning of a mobile robot in NDT weld line testing.

To reconstruct nominal freeform surfaces on the components, the RE approach are manually carried out by skilled workers [11, 12, 89, 90]. After 3D point cloud of the parts were acquired, the data is loaded to an RE software, where subsequently after the defects are manually identified, the nominal surface models (defect-free models) are edited. To build up the nominal models, the defect parts are initially removed by creating empty hollows in the models. Subsequently, a hole-filling technique is used to

estimate the missing data in the models. This approach becomes a standard procedure in adaptive machining processes. Tao *et al*. [90] used a polynomial function to fill the voids of polygon models, while Bezier patches are used in [91] and [11]. In addition, there are also some approaches reported in the literature that are based on the G1 (geometry continuity to the first derivatives) conditions [92, 93] that suit well for nominal reconstruction.

2.6.2 Hole filling technique

Reconstruction surfaces of 3D objects from its measurement data in the absence of design information is a common issue in many areas, such as computer science, video game, medical inspection, manufacturing, metrology, cultural heritage restoration, etc. There are many factors affecting digitization processes that result in incomplete measured data. The main reason is that most of the methods for digitizing geometries are based on optical principles such as in laser triangulation scan, 3D camera, time of flight camera etc. In this case, the measured data points might be lost or damaged if the surfaces have low reflectance ratio, high incident angle, occlusion objects etc. Consequently, there will be some voids created on the measured surfaces [94, 95].

Mathematically, an exact reconstruction of geometric models of 3D objects with some voids in the measured data, is almost impossible. Most solutions are merely an approximation based on several assumptions. In general, we will have more chances to obtain higher accuracy in the reconstructed geometries if the voids are relatively small and the topologies of the surfaces are relatively simple.

Hole-filling is an interesting research topic. Many methods have been proposed to estimate data at the voids (holes) in the measured data to fully reconstruct 3D models of the objects. The measured point cloud data is meshed to create polygon surfaces (in STrereoLithography format). Prior to filling the holes in the meshes, we need to localize the holes in advance. The holes are identified by tracing its boundaries. It is a straightforward method as any interior edges have to be shared by two corresponding triangular surfaces. Afterwards, the identified holes are filled with a certain algorithm.

In literature, the hole filling approach is categorized into two classes, i.e. geometric and volumetric approaches. In the first class, after identifying the holes, the data points at the vicinity of the identified hole are selected to construct a continuous surface by fitting the data with a polynomial function or a parametric function. In the volumetric

approach, the object's volume at the local region of the hole is used to estimate the missing data of the hole by minimizing a certain energy function.

The geometric approach offers good results if the holes are small and the surfaces are simple. In a case of big holes and complex surfaces, the surface approach usually results in higher errors. Wang and Oliveira [96] used a polynomial function for local surface fitting of a hole using moving least square method. Liepa [97] used triangulation method of three dimensional polygon [98] to fill a identified hole. The meshes are then refined to maintain Delaunay triangulation and adjusted by minimizing a fairness function to create a smooth surface at the hole.

Besides the use of polynomial surfaces to fit local holes, parametric functions are also used to estimate surfaces at the missing data sites. Li *et al.* [99] proposed a method to fill holes in triangular meshes using a polynomial blending technique. After identifying holes in the mesh, depending on the complexities of the holes, feature curves are generated using the polynomial blended algorithm to connect two parts of a feature curve (a feature curve usually has three segments, two end segments are taken from involving area of the hole, and the middle segment is an estimation of the missing part of the feature curve). After obtaining the feature curves, the hole is divided into smaller surface patches. Each of the surface patch is fitted with a bi-cubic Bezier surface. To maintain watertight condition at the boundaries of the hole, Lagrange multiplier is used to solve the hard constraints of the surface patch to identify the control points. Afterwards, Bezier surface patches are meshed by using Bowyer's method [100] to generate Delaunay triangulation. Figure 2.6 shows an example of the utilization of the method to fill some holes on a horse model. The figure is reproduced from the paper.

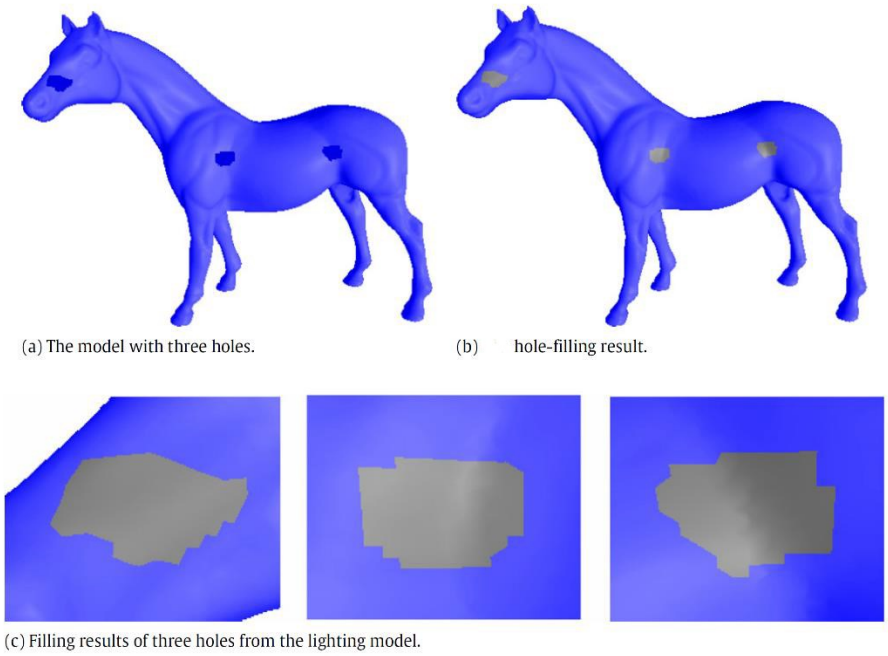


Figure 2.6 Hole-filling technique based on Polynomial blending and Bezier-Lagrange fitting reproduced from [99]

Kumar *et al*. [93] used surrounding data of a hole to find rings by fitting selected data with NURBS curves. Six rings are used to construct a lofted NURBS surface to fill the missing data in the hole. Subsequently, Delaunay triangular meshes are generated from the NURBS surface to fill the holes. Furthermore, in a class of the geometric approach, there exists also some methods that use Radial Basis Functions (RBFs) such in Branch *et al*. [101] and Wu [102].

In the volumetric approach, Davis *et al*. [103] used volumetric diffusion to fill holes in complex surfaces. The process starts by converting the measured surfaces to volumetric representation by using a signed distance function. The signed distance function has value in [-1,1] interval, where -1 value indicates a point inside the object and value 1 indicates a point outside the object. The geometry of the object is a zero set of the signed distance function. Initially, the function is not defined at holes. Iteratively, a diffusion process is applied to the vicinity of the holes. In the process, the signed distance function at the boundaries propagates to new locations by the diffusion process until the holes are filled. At every propagation step, a new zero set of the function is evaluated and filtered with a lowpass filter to create smooth surfaces. Ju [104] used octree grids to decide inside/outside of the objects and employed

contouring to reconstruct the surfaces. In addition, there exists also some methods that are based on volumetric method as discussed in [105-107].

In design phases, sometimes we need to find a smooth surface patch to connect multiple surface patches that satisfy smoothness required at boundaries. Generally, it requires the filled surface patches to be connected smoothly with other surface patches, i.e. the boundaries are at least G^1 smooth. Figure 2.7 illustrates the example of hole filling in design processes.

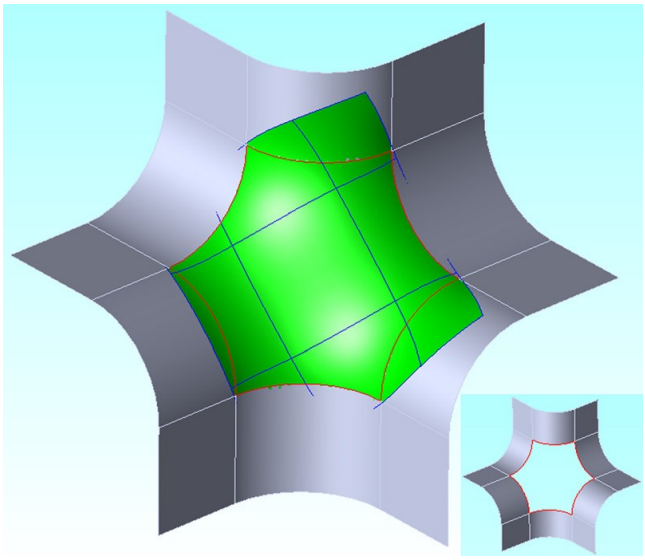


Figure 2.7 B-spline filling n-sided hole, reproduced from [92]

In a case where the filled patches are connected with four surface patches (four boundaries, four cross boundary derivative), Coons patch surfaces are usually used. Piegl and Tiller [108] subdivided n-sided regions into a set of four-sided regions by selecting a central point (the central point connects multiple four-sided regions located in the middle of the initial n-sided region). Bicubically blended Coons patches that are expressed in a form of B-spline, were constructed from the boundary conditions. The method is simply implemented without solving any optimization problems. However, the method only has G^{ε} continuity, i.e. the filled patches are kinks at the boundaries with the maximum angle difference of normal vectors smaller than ε. To obtain G^1 continuity while using Coons B-spline patches, Shi *et al*. [109] used the surface patches constructed on triangle instead of quadrilateral boundaries.

Alternatively, Liu [92] and Szilvasi-Nagy [110] employed B-spline surfaces to fill the holes on the surfaces. In [92], trimmed B-spline surfaces are used to fill holes. The

filled surfaces are defined by minimizing an energy function. To maintain the constraints on the boundaries of the filled patches, instead of solving the minimization problem subject to the constraints, the author suggested to present the constraints to energy penalty. Consequently, the optimization problem with hard constraints changes to a normal problem without constraints that makes the filling processes becoming faster and more robust.

2.7 Tool path for CNC milling

In this subsection, we provide an overview of a procedure of tool path generation and discuss some approaches in literature. In particular, we will discuss the methods for generating tool paths from measurement data in freeform surfaces.

2.7.1 Tool path generation

Milling is one of machining processes that are widely used to create various surfaces from flat to freeform ones. Planning a milling tool path is an interesting topic especially when targeted geometries are freeform surfaces.

In general, there are some rules that every tool path generation should satisfy: (*i*) after machining, all excessive material needs to be properly removed, i.e. no positive defects (gouges) are present due to undercutting because the cutter cannot reach required position, and no negative defects due to overcutting of the workpiece, (*ii*) the machining process should have reasonable cost including cost of the cutter and machining time, and (*iii*) after machining, target geometries must satisfy conditions on surface quality such as roughness, waviness, flatness, etc.

Considering some limitations on the cutter, to fulfill the above conditions, milling processes are usually divided into three cutting stages [111]. The first stage is the rough cutting, where the main purpose is to maximize the material removal rate. The second stage is the semi-finish that is targeted to prepare surfaces for the last stage. And finally, in finishing stage, the cutter will produce the target surfaces. A typical procedure to create such tool paths comprises following steps:

- Calculate the number of cutting layers (slices) from the maximum different thickness of the workpiece and the target geometry.
- Identify targeted surfaces for each slice by offsetting the target geometry.
- Identify cutter location surface from temporary targeting surfaces.

- Construct tool paths for each slice.
- Connect tool paths from different regions in a slice and difference slices.
- Evaluate the tool paths to avoid any collisions, gouging.
- Convert the tool paths to G-code.

Tool geometry: Selection of a tool geometry is the first task to be performed in any tool path generation works. The tool geometry is selected based on two conditions: (*i*) the selected tools have to target the surfaces accurately without creating gouges and overcuts, and (*ii*) the tool should be selected to achieve maximum material removal rate to minimize machining time.

When machining freeform surfaces, ball-nose mills with maximum diameters should be selected for finishing processes. It must be noted that the tool radius has to be smaller than the smallest radius of curvature of the machined surfaces. Because ball-nose mills have lower material removal rate compared to flat-end mills, we usually select flat-end mills for roughing and ball-nose mills for semi-finishing and finishing steps. Machining cost contributes by tool's price and machining time price. Utilizing different tools at different machining step will maximize material removal rate. However, the more the tools used and changed during the process, the more the machining time will take, which in turn will increase the production cost [111].

Milling machine configuration: Machining of freeform surfaces usually uses three- and five-axis CNC machines. The three-axis machine has three degree-of-freedom in cutter position, while the five-axis machine adds two more axes for cutter orientation [112]. Compared to the three-axis machine, the five-axis machine is more flexible that makes them suitable to perform machining parts with undercutting that cannot be performed by the three-axis machine. The five-axis machine also gives higher productivities than the three-axis one if both machines are used for same parts. However, higher investment cost is required for the five-axis.

Tool path generation: After selecting the cutters, mathematical models of tools [113] associated to the target surfaces (or intermediate target surface) are used to calculate the cutter contact point or cutter location (CL) surfaces for path planning. There are different methods to plan the movement of the cutters in CL surfaces. We can categorize the tool paths into three groups, i.e. iso-parametric, iso-planar and iso-scallop.

Iso-parametric is the simplest method to generate tool paths when CL surfaces are in a form of parametric surfaces $S(u, v)$. Since the target surfaces are usually designed in CAD environment using NURBS surfaces, the CL surfaces are simply created by offsetting ([114-116]) the target surfaces by a distance equal to the radius of the cutter for ball-nose mills cutter case. The iso-parametric is first introduced by Loney and Ozsoy [117]. In the method, the tool paths are created by evaluating the value of the surface $S(u, v)$ at a specific value of the parametric u or v. The method is easy to implement except when the surface contains some trimmed surfaces [118]. A main disadvantage of this method is that the remaining scallop height after machining is varied, because of non-linear mapping between the parametric space and the Cartesian space. The method is used to generate tool paths for freeform surfaces in [119-121].

Iso-planar is a method to generate tool paths by finding intersectional curves of the CL surfaces and a set of vertical parallel planes in Cartesian space. This method is very stable and robust, therefore it is widely used in many commercial CAM software [122]. One advantage of this method is that it can be used to generate tool paths for various formats of the CL surfaces, i.e. parametric, polygon surfaces etc. The iso-planar method shares the same disadvantage as the iso-parametric method, where the scallop height will vary depending on the change of slope in the surfaces, i.e. higher slope in the surface would result in higher remaining scallop.

Iso-scallop is first introduced by Suresh and Yang [123] and Lin and Coren [124], and largely employed in [123-126]. This method first creates a path, and the second path is created based on the previously created path to ensure that remaining scallop height at any points between two paths are constant. Iteratively, the procedure starts with the old path to create a new one until CL surface is totally filled.

Path connection: After generating the path segments, all segments must be connected to create a single tool path for a CL surface. There are several methods to perform the connection. When mixing-mode between traditional and climbing milling is used, a zig-zag tool path is a common approach, the tool paths with the zig-zag connection is used in many CAM software [127]. If single mill mode is required, one-way connection is performed to connect two adjacent paths. In addition, there are also some path topologies, such as contour parallel path [128], spiral path [129] etc.

2.7.2 Tool paths for polygon surfaces and point cloud

Lasemi *et al*. [130] conducted a review on recent works in tool path generation for a freeform surface. Because our research is based on measured data, this subsection only reviews some related works, where the tool path is generated based on discrete data from measurement processes.

Lin *et al*. [131] presented a method to generate tool paths based on the measurement data in point cloud obtained from a scanning process. First, each curve in the measured data is fitted by a B-spline curve to reduce the measurement error. Afterwards, a triangulation surface is created by connecting each sampled point of the fitted B-spline curves to create a nominal surface. An offset method is used to create CL surface, and typical tool paths are generated and checked to make sure there is no interferences and gouges in the tool paths.

In [132], Park used triangular meshes to represent target geometries. Subsequently, an offset procedure is performed to create CL surfaces in a form of triangular meshes. The CL surfaces are then sliced horizontally to find the intersections to the CL surfaces and the slicing surfaces to perform Z-constant contouring tool paths [133] for three axis milling machines. Similarly, Lee *et al*. [118] presents a method to create tool paths that satisfies iso-scallop condition based on triangular meshes.

Yuwen *et al*. [134] proposed a method that generates iso-parametric for CL surfaces in triangular mesh formats. Because the triangular mesh is not a parametric surface, a parametric function based on harmonic energy is constructed to represent the meshes. Subsequently, iso-parametric is used to calculate tool paths.

Xu *et al*. [135] presents a method to generate tool paths for five-axis milling machines using flat end mills for target surface in triangular mesh formats. There are a few steps in the method. Firstly, cutter contacting path on the target surfaces is identified. Next, identified tool paths are created to ensure that the created path avoids gouging and interference, and to ensure that the movement of each axis is within the machine capability.

2.8 Summary

In this chapter, we firstly review some studies on automatic machining processes that have potentials for integrated intelligent systems that can help to minimize the

dependency on skilled workers. Adaptive machining approaches in repairing and restoring of freeform surface components are discussed. Secondly, B-spline function is presented. The thesis focuses on the construction of B-spline curves from measured data through curve fitting procedures. Next, a brief discussion on computer aided geometry in design and in reconstruction in reverse engineering applications were presented. We also identified problems when reconstructing geometries from measured data that contains defective points, damage surfaces, and holes (missing data). The methods to eliminate defects and to estimate missing data are discussed in detail. Finally, we discussed a typical procedure in path planning and reviewed some related work in generating tool paths from measurement data.

The research gaps in the technology are:

1. Many studies on repairing of freeform surface workpieces focus on the development of algorithms for reconstructing the nominal geometries with large defects using reverse engineering. Workpieces with large defects are usually eliminated in the inspection stage because it is hard to maintain the mechanical strength of the workpieces after reconditioning. On the other hand, components with small defects such as dents, pitting corrosion, or micro-cracks[3] are still repairable. However, reconditioning of the small defective components is commonly carried out manually. There is a need to automate the remanufacturing of components with small defects.
2. Typically, data processing in adaptive machining is a manual procedure, which means after a component is digitized, point cloud data is manually processed by skilled workers. Defective points in the measured data are deleted, which results in holes on the cloud data. A certain algorithm is subsequently applied to fill the holes to reconstruct the nominal geometry. There is no research on automatic defective point elimination in the point cloud data and nominal geometry reconstruction without human intervention.
3. B-spline curves are widely used in reverse engineering to reconstruct curves and surfaces of components. Almost all existing algorithms for B-spline curve fitting work well with smooth curves. However, defective freeform curves/surfaces in reverse engineering might contain cusp, kink or discontinuous edges/points. These

[3] These small defects create stress concentration that causes fatigue failure of the components

defective freeform curves/surfaces cannot fit well with the current algorithms because current fitting algorithms only result in smooth curves/surfaces. There is a need to develop an algorithm for B-spline curve fitting which can handle nontrivial points. This means that the new B-spline fitting algorithm has to result in optimal knot multiplicity to handle the nontrivial curves.

4. Recently, there are a few studies on smart machining systems. In these smart systems, vision sensors are employed to help the machine see workpieces and automatically generate adaptive tool paths to do some simple work such as deburring, drilling, etc. There is no research on a system that can mimic human in machining of protruding defects on a freeform surface. The current adaptive machining requires offline data processing that cannot be applied in real time and onsite as in the smart machining system. There is a need to develop a system that can automatically process vision data to identify machining position and estimate the finishing surface to generate adaptively tool paths for removing small protruding defects on a freeform surface.

Chapter 3: B-spline Fitting

3.1 Introduction

In this chapter, we present a new method for B-spline fitting based on knot insertion strategy. The optimal knots, which includes knot positions and continuity levels (knot multiplicity), are solved using a local non-linear optimization technique. The main feature of the proposed method is the applicability for non-trivial cases. The working principle of the method can be described as follows. First the data is subdivided using a bisecting method and the subset data are fitted by a single-piece B-spline with a pre-defined error bound (fitting error tolerance). Based on the approximate knots from the bisecting step, the locations and continuity levels of the knots are optimized by solving non-linear least squares that minimize the fitting error. The knots are subsequently used to calculate the control points in the ordinary least squares fitting to obtain the spline curves.

The proposed method has a number of benefits: (*i*) the method can be used to automatically/semi-automatically fit a B-spline to a dataset based on the error bound on the given data, (*ii*) knot multiplicity is naturally obtained, which means that the proposed method can be used to fit curves with non-trivial points such as kink, discontinuous points, and (*iii*) the method inherits the advantage on the computational speed from the bisecting technique that is used for data segmentation and the Gauss-Newton method for solving a non-linear least square equation for optimal knot identification. In this method, the B-spline is obtained through a single pass method, which results in fast computation without sacrificing the accuracy. These advantages offer the proposed method to be a potential tool for RE applications. In addition, the method can converge to the exact solution of B-spline fitting from the data that were sampled from B-spline functions.

This chapter is organized as follows: Section 3.2 provides the methodology of the proposed method and Section 3.3 presents the implementation details. Section 3.4 provides the strategy for automatic/semi-automatic fitting of a given dataset with a B-spline function. Section 3.5 illustrates some examples and benchmarking results to some existing methods in the literature. Section 3.6 presents some general conclusions and discussion.

3.2 Methodology for B-spline curve fitting

In this research, B-spline curve is used to fit a given set of data. A spline curve in Eq. (2.1) is fully defined if the control points P_i and the basis functions $N_{i,p}$ are defined. From Eqs (2.1) and (2.2), it is observed that the spline curve has a linear relationship with the control points, but $S(t)$ is non-linear to the knot vector Z.

The common approach to fit a B-spline function is by using the least square method with the following cost function:

$$\min \sum_{j=1}^{n} \left(\hat{S}(t_j) - Q_j\right)^2 \tag{3.1}$$

where $Q_i = (x_j, y_j, \dots)|_{j=1}^{n}$ is the measured j^{th} data point.

The aforementioned optimization problem is linear if the knot vector Z is predefined, where the control points can be solved in a straightforward way. The simplest way to predetermine the knot vector, Z, is by defining it uniformly. This approach can well approximate the curve if the curve is smooth. However, for non-homogeneous curves or curves with non-trivial points, this approach tends to result in overshooting, thus it cannot satisfactorily fit the data. To overcome this problem, non-uniform knots and knot multiplicity are required. However, finding a non-uniform knot vector is a non-trivial problem. Another approach to solving the optimization problem (3.1) is to treat the knot vector Z as a variable. But because of non-linear searching space and multimodal property, the problem usually results in local optima.

Let us consider a case of one-piece p-degree B-spine that is defined in an interval $[a, b]$ and has the following knot vector: $\mathrm{Z} = \underbrace{a, \dots, a}_{p+1}, \underbrace{b, \dots, b}_{p+1}$. To identify the one-piece B-spline for a given data set, we only need to solve the least square equation (3.1) to determine the control points.

Methodology. Given a set of data sampled from a B-spline function $Q_i = (x_j, y_j)|_{j=1}^{n}$, where $(x_j, y_j) = \boldsymbol{S}(t_j) + \boldsymbol{e}_j$, and $\boldsymbol{e}_j$ is random error; the B-spline function $\boldsymbol{S}(t)$ needs to be reconstructed from the measured data. Figure 3.1 shows a sampled case where $\boldsymbol{S}(t)$ has three member-functions $s_1(t), s_2(t)$and $s_3(t)$. The sampled data comprises 22 points indexed from 1 to 22. Commonly, to reconstruct the function $S(t)$, we have

to identify the three member-functions i.e. $s_1(t), s_2(t)$ and $s_3(t)$ and its breaking points $\mathrm{Z} = [\zeta_1, \zeta_2]$. We can easily identify all the member functions if the data is correctly segmented, i.e. the data is split into three subintervals $[(x_1, y_1), \dots, (x_6, y_6)], [(x_7, y_7), \dots, (x_{13}, y_{13})]$ and $[(x_{14}, y_{14}), \dots, (x_{22}, y_{22})]$. The interior knots can be identified by solving intersecting points of each pair of piecewise member functions if we assume that the function itself has C^0continuity (the function continuity at the knot point). In a case of noisy sampled data, we can only find the approximation of each member function and the knot vector.

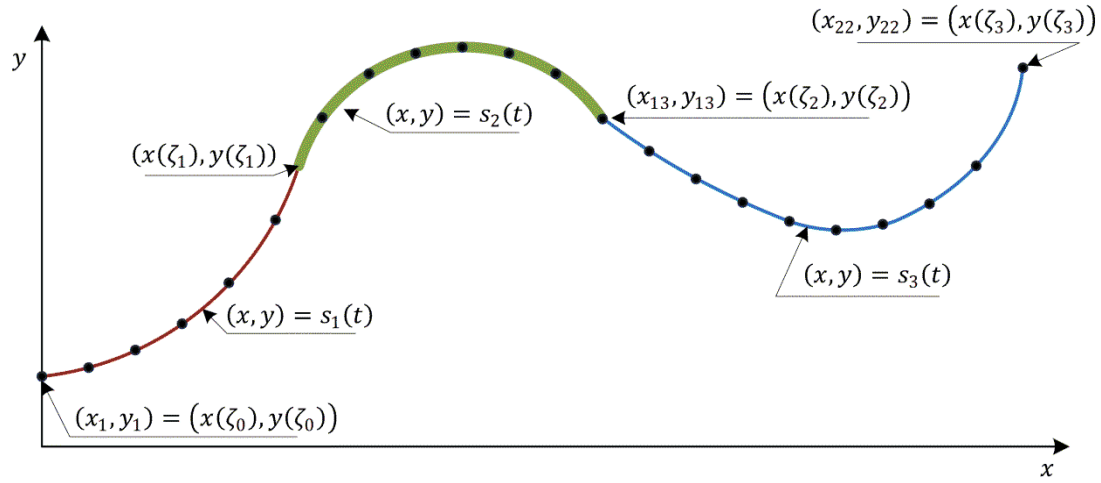


Figure 3.1 Sampled data from a piecewise polynomial function.

As stated above, the most critical part in fitting data with a *pp* function is to split the data to the exact subintervals, i.e. finding the knots of the estimated *pp* function. There are three common approaches to solve the problem. The first is by choosing a first subinterval and identifying the first member function based on the selected subinterval. Subsequently, the subinterval is gradually expanded to the right side until the fitted member function reaches a certain criterion. The process has to be carried out iteratively until the last subinterval. This method is simple and stable but the computational cost is high. To boost the subdivision procedure, the data is usually split by using the bisecting method, which will be detailed in the following subsection. The second approach is to estimate the number and locations of the knots based on a certain criterion such as uniform distribution, Chebyshev points or change in radius of curvature etc. This approach tends to give good results when the fitted curves are smooth. However, in a case when the curve contains non-trivial points, this method usually does not give a satisfactory result, due to oscillation/overshoot near the non-

trivial points. The last approach is to formulate the knots as parameters in an optimization problem and use mathematical tools to find the optimal results. Unfortunately, this optimization problem is multimodal and usually results in local optima.

This work presents the bisecting method for subdividing the input data. Unlike some existing methods in the literature that result in the knots without optimizing their locations [42], the knots are optimized to identify the locations and continuity levels. Knot multiplicity is simultaneously identified by using the optimization process.

In any fitting cases when data is subjected to noise, the member functions of the *pp* function are merely an approximation of the true ones. Therefore, the joining point of every two subsequent piecewise members will normally not coincide with the true knot. To identify the exact knots/multiplicity of knots, we employ a two-piece B-spline function, $\hat{S}_i(t)$, to fit every pair of adjoining fitted member functions, $s_i(t)$ and $s_{i+1}(t)$. The optimal knot/knot multiplicity ζ_i is the solution of the non-linear least square problem (3.1).

Figure 3.2 illustrates the approach in finding the optimal breaking point of a pair of member functions using B-spline. Two member function datasets separated by bisecting step s_i, s_{i+1} are used to construct a two-piece B-spline $\hat{S}_i(t)$ by solving the optimal knot ζ_i of the two-piece B-spline. The continuity level C^k of the knot or multiplicity knot η (where $\eta = p - k$) is derived by comparing the fitting errors of the fitted B-splines when the interior knot $\underbrace{\zeta_i, \dots, \zeta_i}_{\eta}$ is treated as multifold, i.e. $\eta = 1, 2, \dots, p + 1$.

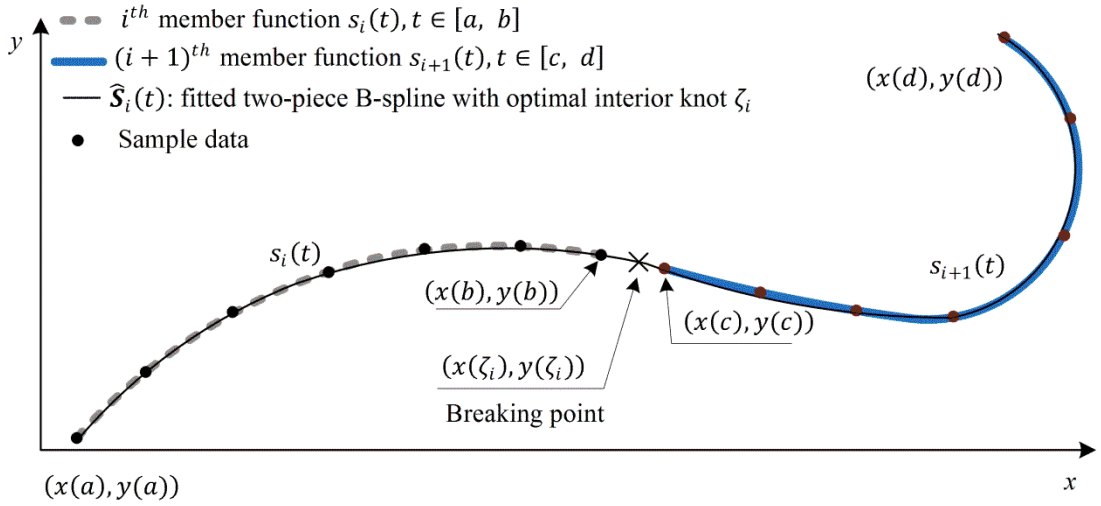


Figure 3.2 Optimal knot solving.

3.3 Free knots placement

There are various approaches to identify a knot vector for a B-spline fitting. In this work, we employ the bisecting method for determining the approximate knot vector and non-linear least square to optimize the knot vector.

3.3.1 Serial bisecting method for data splitting

The bisecting method is also used as a tool for identifying a knot vector in [42-45, 49]. The principle of the proposed bisecting method in this thesis is similar to the method in [42], except that in this process we use a single piece B-spline to fit the data. Figure 3.3 illustrates the working principle of the bisecting method in splitting the data for knot placement. The objective is to find the largest interval $[a, b_n] \in [a, b]$ in which the fitted single-piece B-spline still complies with the error bound criterion (see subsection 3.2 for details). This is an iterative process, which starts by examining the hypothesis if the fitted single piece B-spline curve can be defined from all the data of the searching interval $[a, b]$ without violating the error bound criterion. If the hypothesis is satisfied, then the process is terminated, or else the searching interval is shrunk to $\left[a, b_1 = \frac{a+b}{2}\right]$.

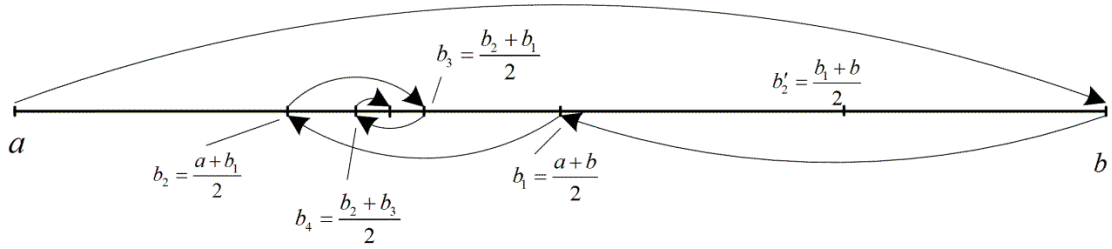


Figure 3.3 Bisecting method for subdivision data in knot identifying process.

At the second step, the new interval is examined with the same hypothesis. If it is satisfied, the searching interval will be expanded to $\left[a, b_2' = \frac{b_1+b}{2}\right]$, otherwise, the searching interval is reduced further into $\left[a, b_2 = \frac{a+b_1}{2}\right]$. The procedure is executed repeatedly until the searching interval cannot be extended anymore. The algorithm in Appendix 1 is the implementation of the serial bisecting method for rough identification of the knots.

3.3.2 Parallel Bisecting method

The serial bisecting method in subsection 3.3.1 works with all data Q_i to find the first spline member function, s_1. However, if the data size is very large, the serial bisecting method might run out of memory and it also takes a long time to compute for the fitted function. Furthermore, the serial bisection is a sequential process, i.e. it cannot take advantage of new computing hardware, which is capable of highly parallel processing on multicore processors. A new parallel bisecting method is proposed, developed from our previous work [46] to overcome the limitation of the serial bisecting method. The parallel bisecting method runs faster than the serial one when the data is large and it overcomes out-of-memory problems, since it only processes the data partially. This subsection will illustrate the proposed method, while the performance will be presented later in section 3.5.

The serial bisecting is initiated by separating the data from the left to the right, while the parallel method starts by separating data at a pre-desired number of pieces. With proper coding, parallel programming can speed up the computation. Figure 3.4 illustrates the working principle of parallel bisecting using the example of Figure 3.1. Note that the B-spline function $S(t)$ consists of three member-functions: $s_1(t), s_2(t)$ and $s_3(t)$. Unlike the serial method, the number of spline pieces initially is freely selected. In this example, the procedure starts by splitting $S(t)$ into two as

shown in Figure 3.4a. Both pieces are concurrently half-split into 4 pieces (Figure 3.4b). Each of the four new subsets is subsequently fitted by using a one-piece B-spline and the fitting error is evaluated. If the error is smaller than a control threshold, the pieces will remain, otherwise, they will be half-split again. Let the first and last pieces in Figure 4b pass the test and the two remaining ones not, as illustrated in Figure 3.4c. The second and third pieces are split further. At this step, the new pieces (2, 3, 4, and 5) have to undergo the fitting test again. At the end of step 3, for example, the two pieces (2, 4) do not pass the test, but we cannot subdivide them further because of insufficient number of samples on those pieces (we will refer to pieces, such as piece 2 and 4, that have data less than $2(p + 1)$, as "small"). The temporary knot vector obtained at this step might contain a few redundant knots such as knot ζ_2, ζ_3 and ζ_5.

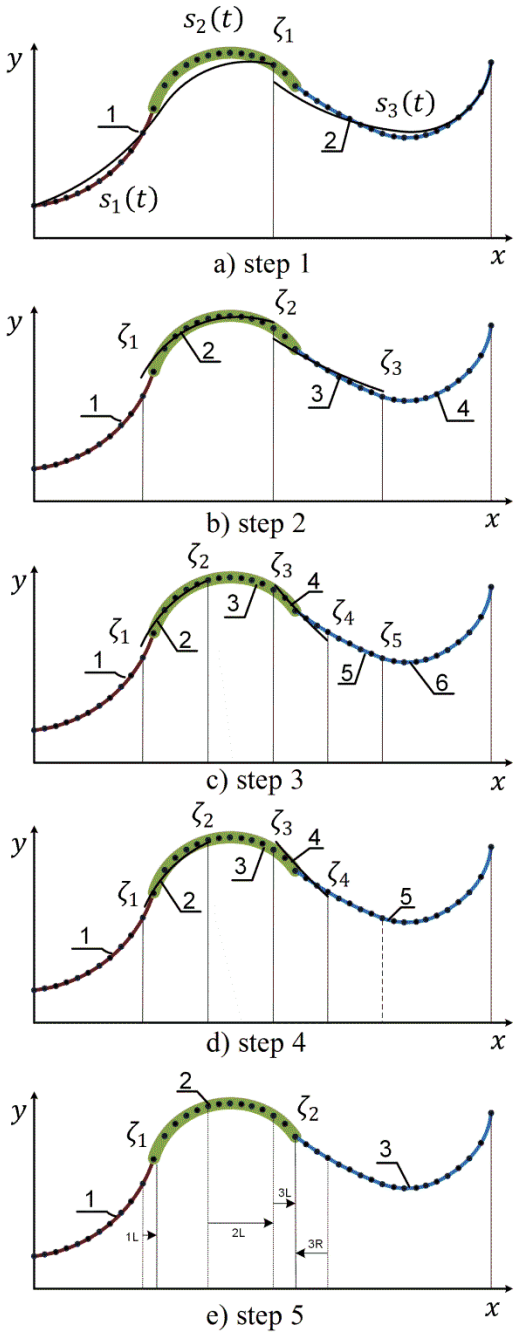


Figure 3.4 Parallel bisecting.

a) Start at 2 spline pieces, b) first bisecting, create four pieces from two, c) second bisecting, create 6 pieces from 4 pieces, d) joining test e) shifting knot to treat small pieces.

To remove the redundant knots, we need to check every two consecutive pieces that pass the error test, whether they belong to a single piece. If the two pieces are confirmed to be from a single piece, they would be merged together. Due to the joining process, knot ζ_5 is eliminated in Figure 3.4d. It is noted that the joining process

does not treat the two redundant knots (ζ_2, ζ_3) because pieces 2 and 4 have not passed the error test yet. The knots (ζ_2, ζ_3) and the identified knots (ζ_1, ζ_4) will be treated in the next step.

In this last step, all the pieces will go through a "shifting procedure". The main idea of the shifting procedure is to expand the large pieces and to shrink small pieces. In the first small piece (piece 2 in Figure 3.4d), the first knot ζ_1 separating the large piece 1 (Figure 3.4d) and the small piece 2 is shifted to the right side (knot ζ_1 is shifted toward piece 2) by a serial bisecting process (1L) (1L: means the knot #1 is shifted from left to right by a serial bisecting process from left side.) to a new position as illustration in Figure 3.4e. The second knot ζ_2 that separates small piece 2 and piece 3 in Figure 3.4d is then considered for shifting. Because piece 3 is also small, the second knot ζ_2 remains. Subsequently, the second small piece (piece 3) is considered. The second knot ζ_2 is shifted to the right by a serial bisection procedure 2L to a new position that coincides with the third knot ζ_3, as illustration in Figure 3.4e and the second knot ζ_2 is eliminated consequently. The third knot ζ_3 is not shifted to the left because the right piece (piece 4) is also small and as a result, the piece 3 is eliminated. The last shifting process will consider the last small piece (piece 4). The knot ζ_3 is shifted to the right by a left-to-right serial bisection 3L to its new position and the knot ζ_4 is shifted to the left 3R by a right-to-left serial bisecting process to the same position with knot ζ_3 (knot ζ_2 in Figure 3.4e). The piece 4 is, therefore, eliminated and the shifting process is completed. The knot vector is now obtained and the parallel bisecting process is terminated. The algorithm is given in Appendix 2.

3.3.3 Error bound of fitted B-spline functions

In order to decide whether a fitted function can represent the given data, a norm is used to evaluate the fitting error. Referring to the l-norm that is defined as [136]:

$$\left\|S(t) - \hat{S}(t)\right\|_l = \left(\sum_{i=1}^{k} \left|S_{x_i}(t) - \hat{S}_{x_i}(t)\right|^l\right)^{\frac{1}{l}} \tag{3.2}$$

Where x_i is i^{th} coordinate and k is the dimension of the B-spline curve.

We employ the maximum norm-2 error as the criterion for data splitting. The error is defined as:

$$E\left(\hat{S}(t)\right) = \max\left(\left|Q_j - \hat{S}(t_j)\right|\right) \tag{3.3}$$

Equation (3.3) can be rewritten in matrix form as $E = \max\left(\sqrt{\text{sum}\left(\left(S-\hat{S}\right)\odot\left(S-\hat{S}\right),2\right)}\right)$, where $S = [Q_1, Q_2, \ldots, Q_n]^T$ is the given data vector, $\hat{S} = \left[\hat{S}(t_1), \hat{S}(t_2), \ldots, \hat{S}(t_n)\right]^T$ is the fitted data vector, $\odot$ represents element-wise multiplication operator and $\text{sum}(\_,2)$ means summation of row elements.

The proposed method guarantees that the maximum error of a fitted curve will always be smaller than the control threshold in each fitted local B-spline. However, in practice, when the maximum error is much smaller than the noise level, the fitted curves will tend to be over-fitted. On the contrary, if the control threshold is much larger than the noise level, it usually results in under-fitted curves that fail to capture the trend of the data.

3.3.4 Knot optimization

This subsection deals with a pair of consecutive one-piece B-spline datasets from the bisecting process to find the best fitting two-piece spline function. The main task is to find the optimal interior knot and its multiplicity (continuity level). In the first part, an investigation on the effect of the knot location against the fitting error function is carried out to show the characteristic of the optimal interior knot. In the second part, a typical deterministic optimal solver, namely Gauss-Newton, is employed to find the optimal knot. The last part deals with the selection of the proper knot multiplicity from the optimal knot cases.

a) Optimal interior knot of two-piece splines

Given a set of two-piece B-spline data, we have to find the optimal connection point and its continuity level which can recover the two-piece B-spline. The first question we need to answer is "is the fitting error smallest when the connecting point and its continuity are set the same as the ground truth?" Answering the question will give us a hint to finding the optimal knot for the two-piece B-spline.

Figure 3.5 illustrates four cases of two-piece cubic splines that have interior knots at $t = 0.5$ (t = 0…1). The spline curves are shown in the top four panels. The first B-

spline has a single knot and the second has a double knot, the third has a triple knot and a quadruple knot for the last one. All cases are tested with different types of knot multiplicity (single $\eta = 1$, double $\eta = 2$, triple $\eta = 3$ and quadruple $\eta = 4$) and the middle row panels show the fitting errors when the knots vary from 0.35 to 0.65. The last four panels at the bottom illustrate the joining kink angles at the knots. We will discuss the joining kink angle in part c) of this subsection.

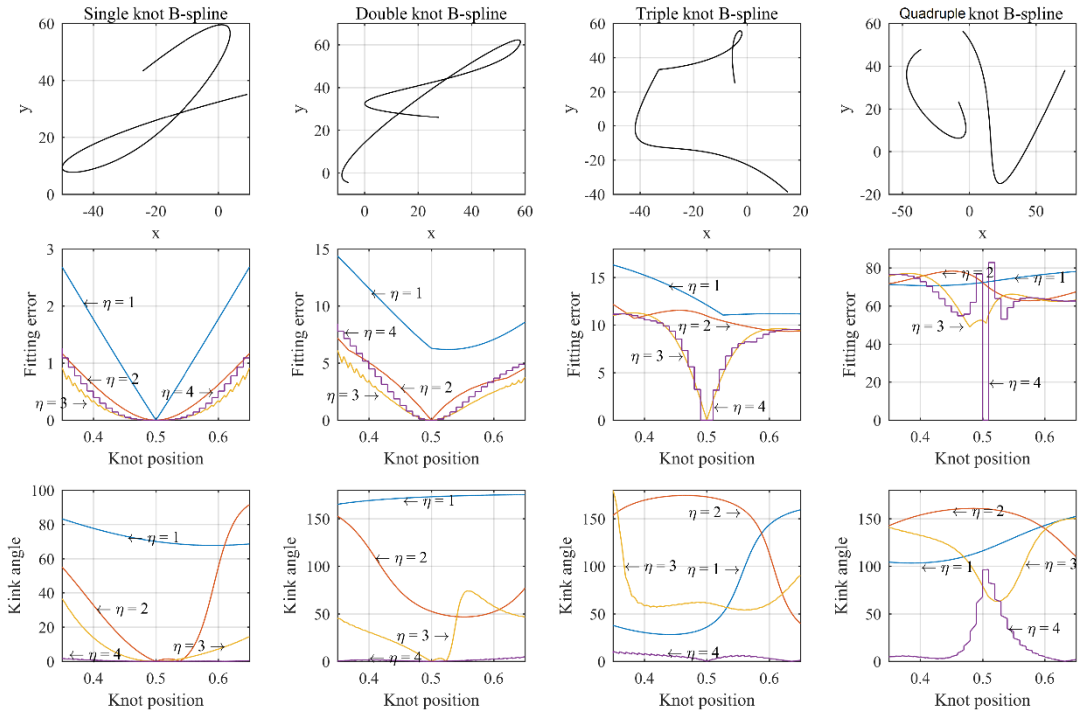


Figure 3.5 Interior knot location versus fitting error of two-piece B-spline, top panels: two-piece spline curves, middle panels: fitting error versus interior knot location with different knot multiplicities of the two-piece B-spline curves, bottom panels: joining kink angel versus interior knot position. Each column shows the results of each B-spline case.

When the knot is set as a quadruple, the error plots for all B-spline cases appear like staircases ($\eta = 4$ cases in the middle row of Figure 3.5). Is that observation always true?

Theorem 3.1 *Let* $\{_{i=1}^{n} = \{Q_i\}_{i=1}^{n1} \cup \{Q_i\}_{i=n1+1}^{n}$ *be a set of data sampled from a two-piece B-spline of p degrees* $S(t) = \begin{cases} s_1(t) & if\ \zeta_{s0} \le t < \zeta \\ s_2(t) & if\ \zeta \le t < \zeta_{s1} \end{cases}$ *where* $\{Q_i\}_{i=1}^{n1} \in s_1(t)$, $\{Q_i\}_{i=n1+1}^{n} \in s_2(t)$ *and* $n_1 > (p+1), (n-n_1) > (p+1)$ *. Given* ζ *is a discontinuity interior knot ((p+1)-fold knot) of the fitted B-spline function* $\hat{S}(t)$*, then,*

the fitting error $E = max_i\left(\sqrt{\left(Q_i - \hat{S}(t_i)\right)^2}\right)$ *is a piecewise constant function (staircase function) of* ζ.

Proof: Based on the definition of B-spline, a two-piece p-degree B-spline, which is discontinuous at its interior knot ζ ($(p+1)$-fold knot), consists of $3(p+1)$ knots $\zeta_i|_{i=0}^{3p+2}$, $2(p+1)$ basis functions $N_{i,p}|_{i=0}^{2p+1}$, and $2(p+1)$ control points $P_i|_{i=0}^{2p+1}$ respectively. The knots are separated into three groups of identical knots $\zeta_0 = \zeta_1 = \cdots = \zeta_p = \zeta_{s0}$, $\zeta_{p+1} = \zeta_{p+2} = \cdots = \zeta_{2p+1} = \zeta$ and $\zeta_{2p+2} = \zeta_{2p+3} = \cdots = \zeta_{3p+2} = \zeta_{s1}$. The basis function $N_{i,p}$ is defined in between the knots ζ_i and ζ_{i+p}. Figure 3.6 illustrates knots and basis function of a discontinuous two-piece cubic B-spline ($p = 3$) as an example.

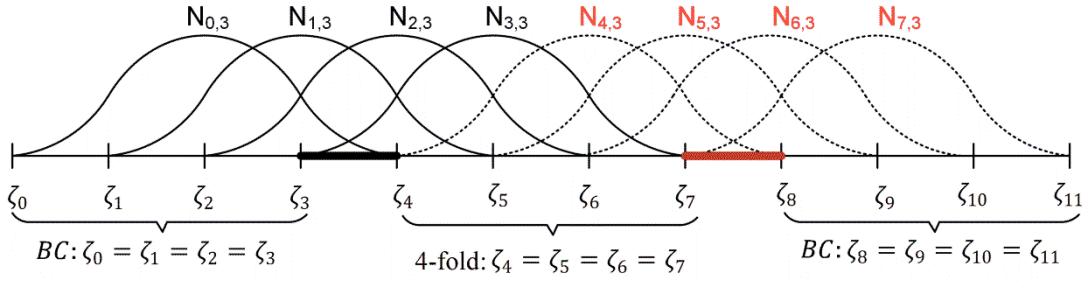


Figure 3.6 Knots and basis functions of discontinuous two-piece cubic B-spline ($p = 3$).

Let $f_1(t), f_2(t)$ be p-degree polynomials, which are the best fit by least square method of the datasets $\{Q_i\}_{i=1}^{m}$ and $\{Q_i\}_{i=m+1}^{n}$ respectively.

We have

$$\begin{aligned} f_1 &= [1, t, t^2, \dots, t^p]\left[\alpha_0, \alpha_1, \dots, \alpha_p\right]^T \\ f_2 &= [1, t, t^2, \dots, t^p]\left[\beta_0, \beta_1, \dots, \beta_p\right]^T \end{aligned} \tag{3.4}$$

Where

$$\left[\alpha_0, \alpha_1, \dots, \alpha_p\right]^T = \left(\begin{bmatrix} 1 & t_1 & \dots & t_1^p \\ 1 & t_2 & \dots & t_2^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & \dots & t_m^p \end{bmatrix}^T \begin{bmatrix} 1 & t_1 & \dots & t_1^p \\ 1 & t_2 & \dots & t_2^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & \dots & t_m^p \end{bmatrix}\right)^{-1} \begin{bmatrix} 1 & t_1 & \dots & t_1^p \\ 1 & t_2 & \dots & t_2^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & \dots & t_m^p \end{bmatrix}^T \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_m \end{bmatrix}$$

and

$$\left[\beta_0, \beta_1, \dots, \beta_p\right]^T =$$

$$\left(\begin{bmatrix} 1 & t_{m+1} & \dots & t_{m+1}^p \\ 1 & t_{m+2} & \dots & t_{m+2}^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & \dots & t_n^p \end{bmatrix}^T \begin{bmatrix} 1 & t_{m+1} & \dots & t_{m+1}^p \\ 1 & t_{m+2} & \dots & t_{m+2}^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & \dots & t_n^p \end{bmatrix}\right)^{-1} \begin{bmatrix} 1 & t_{m+1} & \dots & t_{m+1}^p \\ 1 & t_{m+2} & \dots & t_{m+2}^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & \dots & t_n^p \end{bmatrix}^T \begin{bmatrix} Q_{m+1} \\ Q_{m+2} \\ \vdots \\ Q_n \end{bmatrix}.$$

The fitted B-spline is defined as

$$\hat{S}(t) = \begin{cases} \hat{s}_1(t) = \sum_{i=0}^{p} N_{i,p} P_i \quad if \quad \zeta_{s0} \le t < \zeta \\ \hat{s}_2(t) = \sum_{i=p+1}^{2p+1} N_{i,p} P_i \quad if \quad \zeta \le t < \zeta_{s1} \end{cases} \tag{3.5}$$

$\forall \zeta : t_m < \zeta \le t_{m+1}$, where t_m and t_{m+1} are parameters of the sample point Q_m and Q_{m+1}, and $m \ge (p+1)$ and $(n-m) \ge (p+1)$ respectively. The data is subdivided into two subsets $\{Q_i\}_{i=1}^{m}$ and $\{Q_i\}_{i=m+1}^{n}$.

The fitted B-spline can be rewritten in a matrix form as

$$\begin{aligned} \hat{s}_1(t) &= [1, t, \dots, t^p] \begin{bmatrix} a_{00} & a_{10} & \dots & a_{p0} \\ a_{01} & a_{11} & \dots & a_{p1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0p} & a_{1p} & \dots & a_{pp} \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_p \end{bmatrix} = TAP_I \\ \hat{s}_2(t) &= [1, t, \dots, t^p] \begin{bmatrix} b_{00} & b_{10} & \dots & b_{p0} \\ b_{01} & b_{11} & \dots & b_{p1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{0p} & b_{1p} & \dots & b_{pp} \end{bmatrix} \begin{bmatrix} P_{p+1} \\ P_{p+2} \\ \vdots \\ P_{2p+1} \end{bmatrix} = TBP_{II} \end{aligned} \tag{3.6}$$

where $P_I = A^{-1}\left[\alpha_0, \alpha_1, \dots, \alpha_p\right]^T$ and $P_{II} = B^{-1}\left[\beta_0, \beta_1, \dots, \beta_p\right]^T$.

Because the polynomials $\hat{s}_1(t)$ and $\hat{s}_2(t)$ are the solution of the least square method for the two subsets $\{Q_i\}_{i=1}^{m}$ and $\{Q_i\}_{i=m+1}^{n}$, and the matrices A and B have rank $(p+1)$, while matrices P_I and P_{II} are freely defined, therefore $\begin{cases} \hat{s}_1(t) = f_1(t) \\ \hat{s}_2(t) = f_2(t) \end{cases}$.

Because fitting error $E = max_i\left(\sqrt{\left(Q_i - \hat{S}(t_i)\right)^2}\right)$ is constant for $\forall\zeta : t_m < \zeta \leq t_{m+1}$, therefore, the fitting error E is a piecewise constant function.

Corollary 3.1: *The ground truth knot ζ of a two-piece B-spline $S(t)$ is located in between two samples Q_m and Q_{m+1}, then the fitting error E by the discontinuity two-piece B-spline $\hat{S}(t)$ equal to zero.*

Proof: Let ζ be a discontinuity ($(p+1)$-fold) knot, we have $t_m < \zeta \leq t_{m+1}$. The data is separated into two datasets $\{Q_i\}_{i=1}^{m}$ and $\{Q_i\}_{i=m+1}^{n}$. Because $\{Q_i\}_{i=1}^{m} \in s_1(t)$ and $\{Q_i\}_{i=m+1}^{n} \in s_2(t)$ then $\hat{s}_1(t) = f_1(t) = s_1(t)$ and $\hat{s}_2(t) = f_2(t) = s_2(t)$.

Then the fitting error $E = max_i\left(\sqrt{\left(Q_i - \hat{S}(t_i)\right)^2}\right) = 0$.

Corollary 3.2: *If the original two-piece B-spline $S(t)$ is discontinuous (p + 1)-fold at its interior knot, the ground truth knot ζ cannot be recovered by evaluating the fitting error E of the fitted B-spline $\hat{S}(t)$.*

Proof: As shown in **Corollary 3.1,** the fitting error is $E = 0 \ : \forall\, \zeta \in (t_m, t_{m+1}]$. We cannot find the optimal knot by minimizing the fitted error function.

Corollary 3.1 provides a key to narrowing the region in searching for the optimal knot for the other multiplicities (continuity level) cases. As shown in Figure 3.5, the error functions of all cases might have many local minima except for the single knot case. We can also see that within a sample space, the error functions in all cases will have only one local minimum. Therefore, any deterministic non-linear solvers can be employed to find the optimal knot. In this thesis, we use Gauss-Newton method to find the optimal knots for non-discontinuous cases. The Gauss-Newton method is detailed in the next part of this subsection.

At this point, the question of whether the fitting error will be the smallest when the knot is set the same as its ground truth, as stated at the beginning of this subsection is answered by Corollary 3.1. The smallest fitting error (zero) will not only occur when the interior knot (both position and multiplicity) is set at its ground truth but also when the knot is set close to its ground truth position within the sample space in the discontinuous case. However, due to computational error, it is very hard to obtain the

optimal point with no error. This gives the answer that the discontinuous case will usually exhibit the smallest error in practice.

b) Gauss-Newton method to solve non-linear least square problems

Recalling the least square problem (3.1), this subsection discusses the solution of the optimization problem for a special case when the fitted B-spline has only two pieces, i.e. one breaking point. The data is obtained by sampling two adjoining fitted local B-splines $\hat{S}_i(t)$, $\hat{S}_{i+1}(t)$ which are the results from the bisecting procedure. The location of the knot ζ and its continuity C^k need to be optimized by examining the knot multiplicity case from single to $(p+1)$ folds (the detail was given in Figure 3.2).

A two-piece B-spline which is defined in the interval $[T_a, T_d]$, will have the following knot vector $\mathrm{Z}(T_a, \zeta, T_d) = \langle \underbrace{T_a, T_a, \dots, T_a}_{p+1}, \underbrace{\zeta, \dots, \zeta}_{\eta = p-k}, \underbrace{T_d, T_d, \dots, T_d}_{p+1} \rangle$ where: p is the degree of B-spline, k is the level of continuity C^k, $k = -1, 0, \dots, (p-1)$.

Equation (3.1), thus, can be rewritten in matrix form as

$$\min_i \sum (S(t_i) - Q_i)^2 = \min(X - NP_x)^{\mathrm{T}}(X - NP_x) + \min\big(Y - NP_y\big)^{\mathrm{T}}\big(Y - NP_y\big) \quad (3.7)$$

where: $X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{1\times n}$, $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{1\times n}$, $N = \begin{bmatrix} N^1_{l\times(p+1)} & 0 \\ 0 & N^2_{v\times(p+1)} \end{bmatrix}_{n\times(2p+1-k)}$ and

$$N^1_{l\times(p+1)} = \begin{bmatrix} N_{0,p}(t_1) & N_{1,p}(t_1) & \cdots & N_{p,p}(t_1) \\ N_{0,p}(t_2) & N_{1,p}(t_2) & \cdots & N_{p,p}(t_2) \\ \vdots & \ddots & & \vdots \\ N_{0,p}(t_l) & N_{1,p}(t_l) & \cdots & N_{p,p}(t_l) \end{bmatrix},$$

$$N^2_{v\times(p+1)} = \begin{bmatrix} N_{p-k,p}(t_{l+1}) & N_{p-k+1,p}(t_{l+1}) & \cdots & N_{(2p-k),p}(t_{l+1}) \\ N_{p-k,p}(t_{l+2}) & N_{p-k+1,p}(t_{l+2}) & \cdots & N_{(2p-k),p}(t_{l+2}) \\ \vdots & \ddots & & \vdots \\ N_{p-k,p}(t_n) & N_{p-k+1,p}(t_n) & \cdots & N_{(2p-k),p}(t_n) \end{bmatrix}.$$

The matrix of the basis function, N, is in block diagonal form because a spline piece is defined in *(p+1)* basis functions.

In [57], the authors suggested that the control point vector P_x, P_y should be reformulated in Moore-Penrose pseudo-inverse as:

$$P_x = \begin{bmatrix} P_{x1} \\ P_{x2} \\ \vdots \\ P_{x(2p+1-k)} \end{bmatrix}_{(2p+1-k)\times 1} = N^+X \text{ , } P_y = \begin{bmatrix} P_{y1} \\ P_{y2} \\ \vdots \\ P_{y(2p+1-k)} \end{bmatrix}_{(2p+1-k)\times 1} = N^+Y \text{ where}$$

$$N^+ = \text{pinv}(N).$$

Moore-Penrose pseudo-inverse can generally be evaluated by using Singular Value Decomposition (SVD). In case of redundant data, pseudo-inverse can be evaluated using a normal least squares approach, which has a lower computational cost compared to the SVD method. This motivates us to employ the least square method to solve the control points.

Optimization problem (3.7) is rewritten as

$$\min_i \sum (S(t_i) - Q_i)^2 = \min(G_x^T G_x + G_y^T G_y) = \min G^T G \tag{3.8}$$

where: $G_x = X - NP_x = (g_{x1}, g_{x2}, \dots, g_{xn})^T$, and $G_y = Y - NP_y = \left(g_{y1}, g_{y2}, \dots, g_{yn}\right)^T$, with $P_x = \text{argmin}(X - NP_x)$ and $P_y = \text{argmin}\left(Y - NP_y\right)$ are the solutions of the least squares problems. $G = (g_1, g_2, \dots, g_n)^T$, with $g_i = \sqrt{g_{xi}^2 + g_{yi}^2}$.

To solve the non-linear least squares (3.8) using the Gauss-Newton method, the first derivative of the function G needs to be obtained. We compute the first derivative of G by numerical method as

$$\frac{dG(\zeta)}{d\zeta} = G'(\zeta) \approx \frac{G(\zeta + h) - G(\zeta)}{h} \tag{3.9}$$

where: $h = \sqrt{machine\ epsilon}$ is the step of approximating the first derivative.

The minimum G can be approximated by first order Taylor serier at step ζ^s:

$$G \approx \ G(\zeta^s) + G'(\zeta^s)\Delta\zeta \tag{3.10}$$

The function (3.8) is rewritten as:

$$\min(G^T G) \approx \min\left((G(\zeta^s) + G'(\zeta^s)\Delta\zeta)^T (G(\zeta^s) + G'(\zeta^s)\Delta\zeta)\right) \tag{3.11}$$

The minimum of function (3.11) occurs when $\frac{dG^TG}{d(\zeta^s+\Delta\zeta)} = \frac{dG^TG}{d\Delta\zeta} = 0$, because ζ^s is a constant.

$$G'(\zeta^s)^T G(\zeta^s) + G'(\zeta^s)^T G'^{(\zeta^s)}\Delta\zeta = 0 \tag{3.12}$$

The knot $\zeta^{s+1} = \zeta^s + \Delta\zeta$ at step $s+1$ of the iterative procedure will be calculated by

$$\zeta^{s+1} = \zeta^s - \left(G'(\zeta)^T G'(\zeta)\right)^{-1} G'(\zeta)^T G(\zeta) \tag{3.13}$$

c) Multiple knot selection

The analysis of the optimal knot position discussed in Section 3.3.4a shows that each multiple knot case has its own optimal position. The optimal knot in the discontinuity case usually provides the smallest fitting error. Therefore, the knot multiplicity cannot be selected based on the fitting error itself.

Considering the definition of multiple knots, given two pieces of a B-spline of degree p which are connected at a η-multiplicity knot, the B-spline function is continuous to $(p-\eta)^{th}$ derivative at the knot location. It means that the $(p-\eta+1)^{th}$ derivative at the knot is discontinuous. We can explore this property to select the knot multiplicity for two-piece B-spines.

Figure 3.7 illustrates a two-piece B-spline and its derivatives. Assuming that the B-spline is continuous at the knot location to its first derivative and it is discontinuous in its second derivative, we can easily see that the first derivative has a kink angle α at the knot location.

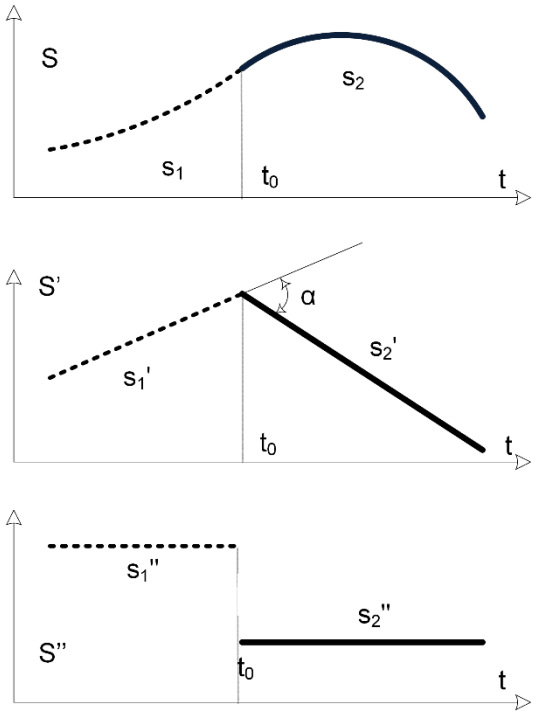


Figure 3.7 Two-piece B-spline and its derivatives.

The kink angle α is calculated from

$$\cos\alpha = \cos\left(\angle\left(\overrightarrow{s_1'}(t_0), \overrightarrow{s_2'}(t_0)\right)\right) = \frac{\overrightarrow{s_1''}(t_0).\overrightarrow{s_2''}(t_0)}{\left\|\overrightarrow{s_1''}(t_0)\right\| \left\|\overrightarrow{s_2''}(t_0)\right\|} \tag{3.14}$$

where $\overrightarrow{s_1'}(t_0), \overrightarrow{s_2'}(t_0), \overrightarrow{s_1''}(t_0), \overrightarrow{s_2''}(t_0)$ are the vector values of the first derivative and second derivative at the knot, respectively.

In general, a joining knot of a two-piece p degree B-spline can have $(p+1)$ multiplicities i.e. single knot to $(p+1)$-fold knot. The single knot case has a kink angle at the $(p-1)^{th}$ derivative. Similarly, the double knot case has a kink angle at the $(p-2)^{th}$ derivative and so on. The kink angle α_η of multiple interior knot η case is computed by Eq. (3.15).

$$\alpha_\eta = \operatorname{acos}\left(\frac{\overrightarrow{s_1^{(p-\eta+1)}(\zeta)}.\overrightarrow{s_2^{(p-\eta+1)}(\zeta)}}{\left\|\overrightarrow{s_1^{(p-\eta+1)}(\zeta)}\right\| \left\|\overrightarrow{s_2^{(p-\eta+1)}(\zeta)}\right\|}\right) \tag{3.15}$$

Theorem 3.2: Let $S(t)$ *be a p degree two-piece B-spline which has interior knot at* ζ *and knot multiplication* $\eta < (p+1)$. *Then, the kink angle of the discontinuity interior*

knot ((p + 1)-fold multiplicity knot) of the fitted two-piece B-spline $\hat{S}(t)$ *by equation (3.15) at the interior knot position* ζ *is equal to zero,* $\alpha_{p+1}(\zeta) = 0$.

Proof: As a result of theorem 3.1, two member-functions of the fitted B-spline $\hat{s}_1(t)$ and $\hat{s}_2(t)$ is joined at the knot position ζ, i.e. $\hat{s}_1(\eta) = \hat{s}_2(\eta)$.

The kink angle of the discontinuity case in Eq. (14) now can be rewritten as:

$$\alpha_{p+1} = \operatorname{acos}\left(\frac{\overrightarrow{s_1^{(0)}(\zeta)}.\overrightarrow{s_2^{(0)}(\zeta)}}{\left\|\overrightarrow{s_1^{(0)}(\zeta)}\right\|\left\|\overrightarrow{s_2^{(0)}(\zeta)}\right\|}\right) = \operatorname{acos}\left(\frac{\overrightarrow{\hat{s}_1^{(0)}(\zeta)}.\overrightarrow{\hat{s}_1^{(0)}(\zeta)}}{\left\|\overrightarrow{\hat{s}_1^{(0)}(\zeta)}\right\|\left\|\overrightarrow{\hat{s}_1^{(0)}(\zeta)}\right\|}\right) = \operatorname{acos}(1) = 0.$$

Figure 3.5 shows the optimal knot and kink angle of each spline case. It is also observed that at the optimal knot (t = 0.5), the kink angles of the discontinuity cases follow Theorem 3.2.

Combining information of fitting error and kink angle will lead to the true multiplicity knot. A multiplicity knot is selected if the kink angle α of the knot is larger than a certain threshold, α_{min}, and has smaller fitting error. The program in Appendix 3 gives the details of the implementation in optimal knot solving and selecting.

3.4. A strategy for fitting of non-uniform B-spline curves

This section summarizes the proposed method for B-spline fitting. In principle, there are three steps in fitting a B-spline function. The first step is parameterization of the input data to convert the data into parametric form, which strongly affects the fitted B-spline curve quality [137]. Selection of a proper parameterization method is essential in this step. There exist some methods such as 'Chord length', 'Uniformly spaced' or 'Centripetal' for the parameterization method. The 'Chord length' method is widely used. The second step is the estimation of the noise level of the data as a basis for determining the error bound. The last step is the application of the proposed method for knot identification and then the least square to fit the input data for identification of control points of the B-spline.

Selecting the proper maximum error (error bound) will affect the number of knots. If the selected maximum fitting error is much smaller than the noise level, the fitted curve will be over-fitted and, the fitted curve will be under-fitted otherwise. Therefore, the maximum error should be selected based on the noise level of the input data.

In short, the strategy for fitting of the given data by B-spline curve with free and multiplicity knots can be summarized as follows:

Step 1: Parameterizing the input data to obtain a data set (t_i, x_i, y_i).

Step 2: Estimating noise level to calculate error bound ϵ.

Step 3: Applying bisecting method to solve approximate knots.

Step 4: Optimizing the approximate knots to identify the optimal ones by solving non-linear least squares and select the optimal knot multiplicity.

Step 5: Using least square to calculate the control points of the fitted B-spline curve based on the optimal knots.

Step 6: Computing the fitted B-spline curve and fitting errors.

3.5 Experimental results

The proposed method is numerically validated and the results are discussed in this section. Three distinct experimental validations with different sets of data are conducted. In the first experiment, datasets are sampled from B-spline functions, and in the second, datasets are generated from deterministic functions. While the first two experiments consider clean data, the third one deals with data with noise. In all cases, we employ Mean Square Error $(MSE) = \frac{1}{N}\sum_{i=1}^{N}\left(\left|\hat{S}(t_i) - Q_i\right|\right)^2 = \frac{1}{N}\sum_{i=1}^{N}((\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2)$, Root Mean Square Error $(RMSE) = \sqrt{MSE}$ and Maximum Error $(ME) = \max\left|\hat{S}(t_i) - Q_i\right| = \max\sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}$ to quantify the performance of the method. In this section, we also discuss the processing speed performance of the method implemented in MATLAB 2014a environment running on a computer with Core 2 Duo T7300 2.0GHz processor and 3GB of RAM.

3.5.1 Fitting data sampled from a spline function

A spline function that was proposed by Kang et al. [54] with the same parameters are used. The function is generated using the interior knots listed in Table 3.1. The function is uniformly sampled with 1001 points (note that there exists a double knot at 0.5408).

Table 3.1. Identified interior knots.

Ground truth knots	Residual errors from proposed method
0.0439	-1.318e-13
0.0653	-5.273e-12
0.2293	-3.972e-12
0.2367	-1.771e-09
0.4821	-1.160e-10
0.4907	-2.216e-10
0.5408	-1.840e-12
0.5408	-1.840e-12
0.6209	-3.926e-12
0.7051	-1.126e-13
0.9407	-5.221e-11

Figure 3.8 depicts the B-spline function in the left panel with the fitting error in the right panel. The proposed method results in nearly exact interior knots and multiplicities with the residual errors are given in Table 1. The mean square error (MSE) of this fitting is 8.046e-15, achieved in about 0.3 second. We can see that the interior knots approximate the true ones with the maximum residual error is -1.771e-09. The errors of the calculated knots most likely are caused by the error in numerical solution in the optimization process. The details of the parameters used are listed in Table 3.2.

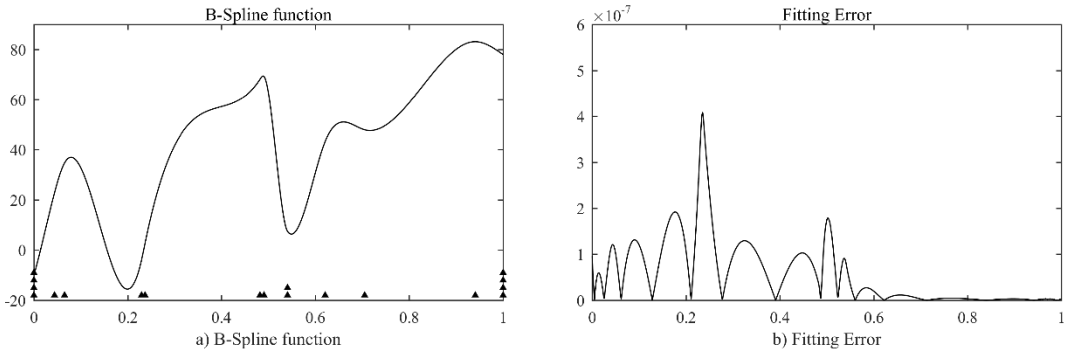


Figure 3.8 Approximation of a cubic B-spline.

Table 3.2 shows the differences between serial and parallel bisection approaches. All the parameters used for the two methods are kept the same, but there is a slight difference in the results. The approximate knots by both methods are almost the same except the second knot, where the serial method results in 46 while the parallel one results in 45 (the correct number is 45 with $t = 0.044$). The serial method tends to shift the knot to the right because its algorithm is based on left-to-right bisection. As a

result, the final optimal knot of the second knot (first interior knot) exhibits a small difference due to different start points in Gauss-Newton solving step. Furthermore, the processing time of the parallel method is slightly slower by 2ms.

Table 3.2. Differences between serial and parallel bisecting.

Parameters	Serial bisection	Parallel bisection
Spline Degree	$p = 3$	$p = 3$
Control error	$\epsilon = 1e-6$	$\epsilon = 1e-6$
Minimum kink angle (degree)	$\alpha_{min} = 0.002$	$\alpha_{min} = 0.002$
Maximum smoothness C(k)	$k = -1$	$k = -1$
Data bisecting time (ms)	25.7	27.6
Coarse knots by bisecting step	1 **46** 67 231 238 484 492 542 623 707 943 1001	1 **45** 67 231 238 484 492 542 623 707 943 1001
Optimal interior knot residual errors	**-2.096e-14** -4.306e-12 6.652e-13 -2.986e-12 -8.347e-11 -8.886e-11 -1.427e-11 -1.427e-11 -5.143e-12 -7.424e-09 -3.039e-11	**-5.297e-14** -4.306e-12 6.652e-13 -2.986e-12 -8.347e-11 -8.886e-11 -1.427e-11 -1.427e-11 -5.143e-12 -7.424e-09 -3.039e-11

In comparison to the sparse optimization method [54], the proposed method gives MSE = 8.046e-15 while the sparse optimization method exhibits MSE = 3.7596e-6. It can be also highlighted that the computational time of the proposed method is about 0.3 second, while the counterpart took about 40 seconds.

For further discussion on fitting data sampled from B-spline functions, please refer to Appendix 4.

3.5.2 Approximating deterministic functions

In this subsection, the performance of the proposed method is evaluated on parametric deterministic functions. Two different functions are used, i.e. a butterfly curve and a

spur gear curve that was sampled from a 3D model generated using Solidworks. In the first case, the butterfly curve was generated using two functions [138]:

$$\begin{cases} x(t) = \sin(t)\left(e^{\cos(t)} - 2\cos(4t) - \sin^5\left(\frac{t}{12}\right)\right) \\ y(t) = \cos(t)\left(e^{\cos(t)} - 2\cos(4t) - \sin^5\left(\frac{t}{12}\right)\right) \end{cases}, t = [0, 2\pi].$$

The butterfly function is sampled with 629 points at a uniform spacing of $t = 0.01$. The resulting curve is illustrated in the left panel of Figure 3.9. Table 3.3 lists some different cases in the parameters used. The number of interior knots strongly depends on the control error threshold when the other parameters are kept unchanged. In the case of cubic B-spline, $p = 3$, the number of interior knots increases from 31 to 72 when the adjusted error ϵ decreases from 1e-3 to 1e-5. We also obtain the same results when the degree of spline is $p = 2$ in case 3 and $p = 4$ in case 4. Even when the parameters are set to find optimal knot multiplicities from 1 to (p+1), the algorithm only results in single-knots.

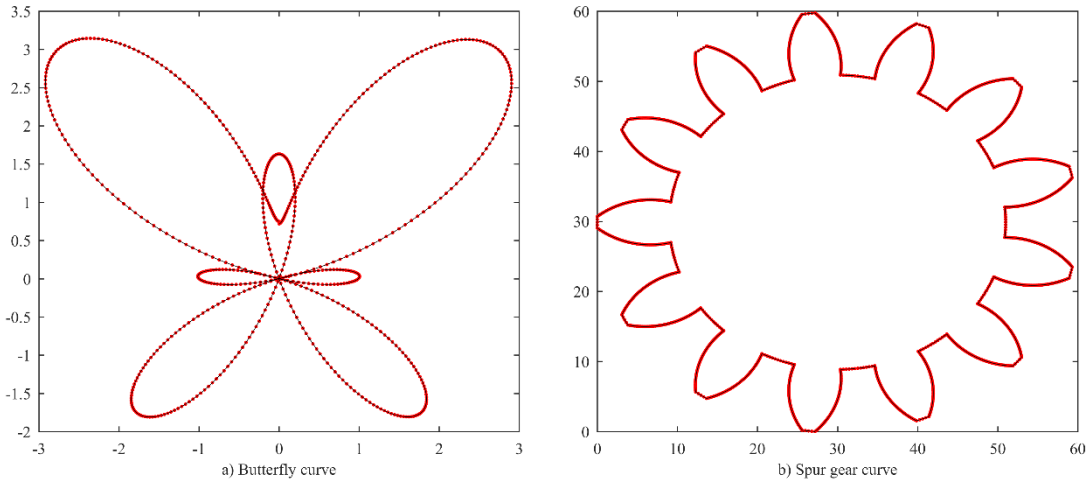


Figure 3.9 Fitted cubic spline of Butterfly (left panel) and spur gear (right panel) curve. Dotted points: sample data, solid curves: fitted spline.

Table 3.3. Some selected cases of the Butterfly curve.

Case No.	Fitting parameters				Results			
	Degree p	Max error ϵ	Min kink angle α_{min}	Continuity C^k	Max fitting error (ME)	Mean squared error (MSE)	No. interior knot (K)	No. pieces (Ψ)
1	3	1e-3	15	-1	0.0019	1.5948e-6	31 single-knots	32
2	3	1e-5	15	-1	8.8161e-5	1.3989e-9	72 single-knots	73
3	2	1e-5	15	-1	1.9663e-4	8.5433e-9	126 single-knots, 1 triple-knot	128
4	4	2e-6	15	-1	7.5484e-5	1.0302e-9	60 single-knots	61

Similarly, on the right panel of the Figure 3.9 and Table 3.4, the results of the fitted spline for a spur gear curve are presented. The data is sampled from a spur gear curve (with module of 4 and 13 teeth). The spur gear curve is generated by sampling a 3D model created using Solidworks (it was digitized by converting the drawing to stereolithography format (STL) to create a triangular mesh). The data is sampled with 1612 non-uniformly spaced points as shown in the right panel of Figure 3.9. Table 3.4 provides detailed comparison between the two bisecting methods. It can be seen that the processing time for the parallel bisecting took about 100 milliseconds (100, 105 and 107) and it is less sensitive to the change in the number of spline pieces. In contrast, the serial bisection needs more time and is highly dependent on the number of spline pieces (130,197 and 269 milliseconds). The total processing time heavily depends on optimal knot solving. It does not only depend on the number of interior knots but also depends on the initial start points for Gauss-Newton solving.

Table 3.4. Some cases of the spur gear curve.

Case no.	**1**		**2**		**3**	
Fitting parameters	**Serial bisecting**	**Parallel bisecting**	**Serial bisecting**	**Parallel bisecting**	**Serial bisecting**	**Parallel bisecting**
Spline Degree p	3	3	3	3	2	2
Control error ϵ	1e-2	1e-2	1e-3	1e-3	1e-3	1e-3
Minimum kink angle (degree) α_{min}	1	1	1	1	5	5
Maximum smoothness C(k)	-1	-1	-1	-1	-1	-1
Data bisection time (ms)	130	100	197	105	269	107
Coarse knots by bisection step	52 interior knots	52 interior knots	78 interior knots	78 interior knots	130 interior knots	181 interior knots
Multiple knot	52 Triple-knots	52 Triple-knots	52 triple-knots, 26 single-knots	52 triple-knots, 26 single-knots	78 single-knots, 52 double-knots	129 single-knots, 52 double-knots
Fitting error: MSE (ME)	2.341e-06 (0.0060)	2.421e-06 (0.0055)	1.562e-07 (0.0013)	6.099e-08 (0.0012)	7.4063e-7 (0.0018)	1.1562e-7 (8.862e-4)
Total processing time (ms)	2490	2501	3640	3715	3970	5635

As we can see, the spur gear has 52 kink points ($13\ \text{teeth} \times 4\ \text{kink points}$). As listed in Table 3.4, the numbers of kink points are correctly defined for all cases. In the first case, the number of interior knots is equal to 52, this leads to all interior triple knots. There is a minor difference in the optimal knots obtained from the two different bisection approaches as highlighted in Appendix 5. Because the data is approximated by a B-spline, the optimal knots will deviate when the input data is slightly changed.

For the two remaining cases, the number of spline piece increases when the control error is decreased, but the number of multiple-knots (kink points) is the same as that in the first case while the extended knots are all single-knot.

3.5.3 Data with noise

In this subsection, we present the results of the proposed method in the presence of noise in data. To quantify the effectiveness of the method and for benchmarking purpose, we employ three functions Phi 1, φ_1, Phi 2, φ_2, and Phi 3, φ_3, which were used in references [64, 139] to make a comparison with the Elitist clonal selection [139].

Three benchmarking functions are used with the same set of parameters (the data is uniformly sampled with 201 points and the randomized noise is normal distribution with $\mu = 0\ and\ \sigma = 1$). Figure 3.10 illustrates the three functions and Table 3.5 provides the numerical results of the comparison. We can see that the results from the proposed method offer higher accuracy compared to those from the Elitist clonal selection method, while the processing time is notably shorter (higher performing computer with Intel Core i7 2.6GHz, 8GB RAM is used in [139]).

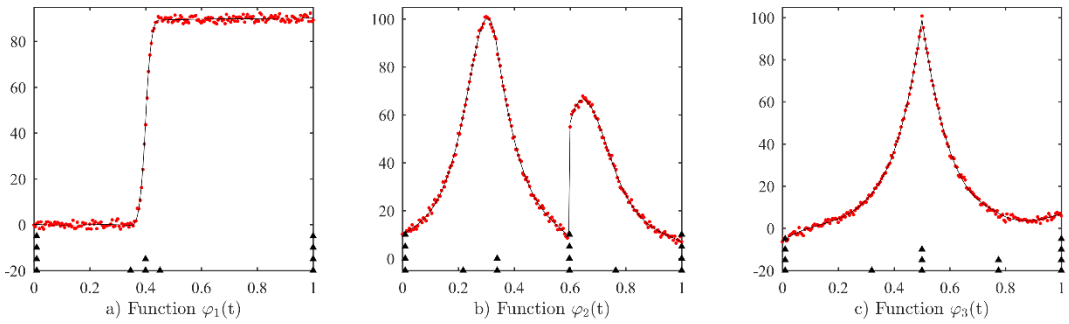


Figure 3.10 Noisy functions for benchmarking with Elitist clonal selection methods [139]. a) $\boldsymbol{\varphi_1(t)}$, b) $\boldsymbol{\varphi_2(t)}$, c) $\boldsymbol{\varphi_3(t)}$ (dots: sample data, solid lines: fitted curve, triangle marker: knot position).

Table 3.5. Average Root Mean Squared Error RMSEs and Standard Deviations (SD) for 50 simulation replications, comparison with Elitist clonal selection method (data are reproduced from Tables 2, 3 and 4 of [139]).

Functions	Proposed method		Elitist clonal selection	
	RMSE(SD)	Run time (second)	RMSE	Run time (second)
$\varphi_1(t)$	0.4153 (0.1135)	0.275	1.06581	10.92
$\varphi_2(t)$	0.5272 (0.1360)	0.229	0.87377	21.71
$\varphi_3(t)$	0.3909 (0.0859)	0.147	0.89368	24.87

3.6 Conclusion

In this chapter, a new method for optimal knot calculation in a B-spline fitting based on a local B-spline fitting technique is proposed. The method is capable for non-uniform knots and knot multiplicity cases. It employs the bisecting method with a specific error bound as a criterion to find the best fitting single piece B-spline for the given data and to identify the approximate knots. The approximate knots are, subsequently, optimized to identify the optimal knots. The method is shown to be able to reconstruct B-spline functions for various sampled data. In comparison to the existing methods in the literature, the method offers faster computational time that is attributed to a single pass process (referred to as a one-pass method in [45]) without sacrificing its accuracy. In many cases, it offers better accuracy than the existing methods. One more apparent advantage of the proposed method is that the processing time does not depend too much to the sample size, but on the control factor (error bound). In typical applications, where the data size is smaller than 1000, the processing time is only a few seconds.

In short, as the fast processing time is the main feature of the method, it is a potential tool for applications in reverse engineering, computer aided design and computer aided manufacturing.

Chapter 4: Automatic estimation of nominal surfaces and defect detection for free form surface parts

4.1 Introduction

The adaptive machining approach usually works with measurement data of the entire geometry of the workpiece to reconstruct the geometric models, which consequently results in closed surface models. This research project, however, aims at an automatic defect detection algorithm for an *in-situ* inspection system, which processes local measurement data that comes in an open surface.

As a case study, we will consider a freeform surface part with some small defects on its surfaces, such as weld beads or dents, and the design information of the workpiece is not available. For remanufacturing purposes, initially the workpiece surfaces are digitized by a laser scanner or a 3D camera to acquire its polygon surfaces for defect detection and nominal surface reconstruction. Currently, identification of these defects is manually performed using Reverse Engineering (RE) software. A skilled worker visually inspects the measured data on the RE software to locate defects and removes the data point of the defect parts and leaves voids (holes) on the measured surfaces. Afterwards, the holes on the measured surfaces are filled to reconstruct the nominal surfaces of the workpiece without reference to the original design information. The manual data processing requires skilled workers and this leads to difficulties in automating the remanufacturing industry. An algorithm that can automatically process measured data to identify defects from the raw input data and reconstruct the nominal surfaces of the workpieces with minimum human intervention becomes requisite.

This chapter presents a nominal surface estimation method that is capable of localizing and quantifying defects from measured data. The process has the potential to be used in automatic remanufacturing processes for refurbishing components. A defective component is first digitized by a 3D camera or laser scanner to acquire its polygonal geometry and afterwards, the facetted surface is converted into z-depth images. The component surface is firstly sliced into a set of parallel curves. Based on the B-spline fitting technique discussed in Chapter 3, each curve is split into spline pieces. Some of these pieces, which are expected to correspond to the nominal curve, are selected and combined to reconstruct the nominal curve based on B-spline fitting. Subsequently, if

necessary, the resulting estimated nominal curve is corrected by eliminating the false ones based on the match[4] of two or more independent estimations on different slicing directions. Finally, the estimated nominal surface is obtained by fitting the depth image using a smooth B-spline surface fitting technique.

The structure of this chapter is organized as follows: Section 4.2 details a method for automatic estimation of nominal curve profiles from measured raw data based on the B-spline fitting technique. Section 4.3 discusses the effect of defect size and its location on the estimation results. Section 4.4 extends the method to identify the defects on free from surface parts. Some experimental results are discussed in Section 4.5. Finally, discussion and some key points are drawn to conclude the chapter.

4.2 Nominal curve profile estimation

4.2.1 Methodology

Let F be a freeform curve that represents a nominal curve profile sliced from a surface of a workpiece. In general, we can approximate the curve profile, F, by a B-spline curve, F^*, within a certain tolerance. Assuming that some parts of the curve, F, are replaced by a few irregular shapes that represent defects on the curve profile as illustrated in the first panel of Figure 4.1.

The curve profile, F, is digitized to obtain measurement data, F_m, as illustrated in Figure 4.1b. It can be assumed that F_m is no longer continuous due to the undercut features in the defects that cannot be sensed by the sensor (either mechanical or optical) viewing from the lateral (upright) direction. The discontinuous curve can also be caused by non-smooth transition between the nominal curve, F, and the defect portions.

It is necessary to develop a method to reconstruct the nominal curve profile, F, from its measurement data F_m with only one known condition that the curve profile, F, is a freeform one. Due to the lack of information on the nominal profile, F, there is no unique solution for the estimated nominal curve profile $\hat{F}$. Any methods resulting in an estimated curve, $\hat{F}$, that satisfies some general criteria, can be used to reconstruct the nominal curve profile.

[4] An estimated nominal point is accepted, if the variation between two or more estimations is within the control error.

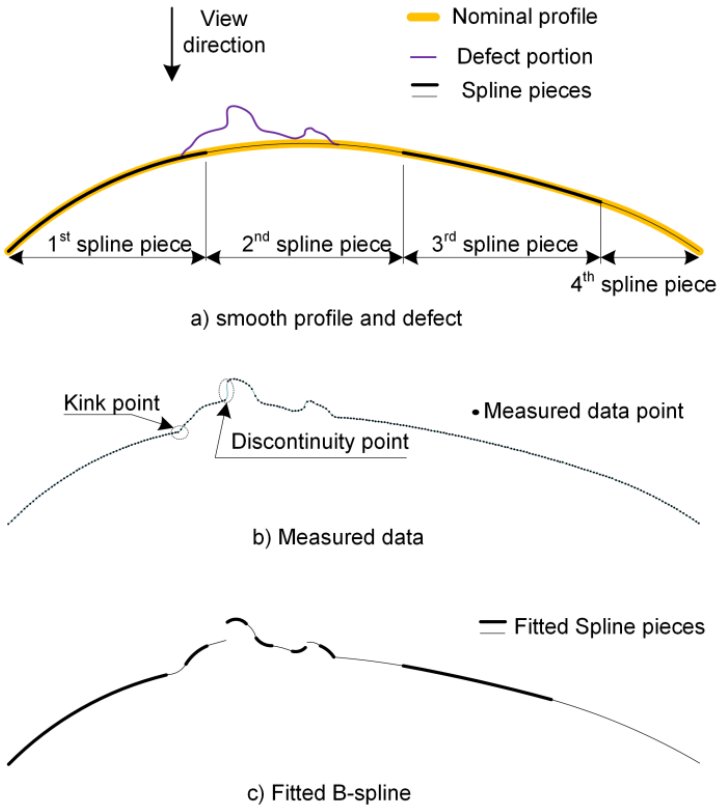


Figure 4.1 Measurement data and nominal curve profile

We propose a method based on B-spline curve fitting, that can automatically estimate the nominal curve profile. The working principle of the method is as follows:

As illustrated in Figure 4.1a, a curve profile, F, can be approximated by a B-spline curve, F^*, assuming that the approximated curve contains a few spline pieces (a spline piece is a piecewise member function in the B-spline function). For example, let the nominal profile in Figure 4.1a be approximated by a B-spline curve with four spline pieces. After digitization, from the measurement data (Figure 4.1b), the first spline piece is slightly damaged on the right-end of the piece, the second spline piece is dominated by the defect leaving a small piece of the nominal curve, while the nominal curves are retained on the last two pieces.

By fitting the measurement data with a B-spline curve, we can separate the measurement data into a few spline pieces based on member functions of the fitted spline as shown in Figure 4.1c. If the first and the last two spline pieces are selected properly, the curve profile, F^* has a chance to be reconstructed from the measurement

data, F_m. It is noted that the estimated nominal profile, $\hat{F}$, hardly matches the approximated curve F^* as we only partially have the approximated curve F^* .

Mathematically, the exact reconstruction of an unknown function from its digitized data might be feasible if the following conditions are satisfied: (*i*) the measured data is clean from noise or the signal to noise (S/N) ratio is larger than a certain threshold that we can consider as "clean data", (*ii*) the digitized data must occupy the entire domain where the function is defined, and (*iii*) there exists a mathematical model of the function that can represent the unknown function within a required tolerance.

Comparing the exact reconstruction conditions with the information of the nominal profile, it is easy to see that it is very difficult to obtain the exact nominal curve from its measurement data. Therefore, any solution is just an approximation of the nominal one. Hypothetically, the estimated nominal curves closely resemble to the original one if the first condition is satisfied.

4.2.2 Nominal curve profile estimation

This subsection details the estimation of nominal curve profile $\hat{F}$ given a set of measurement data, F_m. Six different datasets, which are obtained on real components using a Keyence laser profilometer (LJ-V7080) with spatial resolution at 0.05mm, are used to demonstrate the nominal curve profile estimation.

The measurement curve profiles are depicted in Figure 4.3. To simulate the defects, pieces of clay were attached on the surface of the components. Three different objects were used to simulate the refurbished components. The first three profiles shown in the figure were obtained from a computer mouse, while the fourth profile was obtained from different computer mouse. And the last two profiles were taken from a surface of a mini-UAV propeller. In Figure 4.3, the defect portions are highlighted.

The following provides step-by-step data processing for nominal curve profile estimation. As mentioned in subsection 4.2.1, a B-spline fitting technique is employed to automate the estimation processes. There are five main steps in the procedure. The first step is to use the B-spline fitting to separate the measurement data into small spline pieces. The second step is to select spline pieces that might correspond to the nominal parts of the components. Upon the completion of proper pieces selection, a combining process is used to construct the nominal curve profile. In the final step, a B-

spline curve fitting is used to estimate the nominal curve profile from the selected data and to conclude, a test is conducted confirm the estimated nominal one.

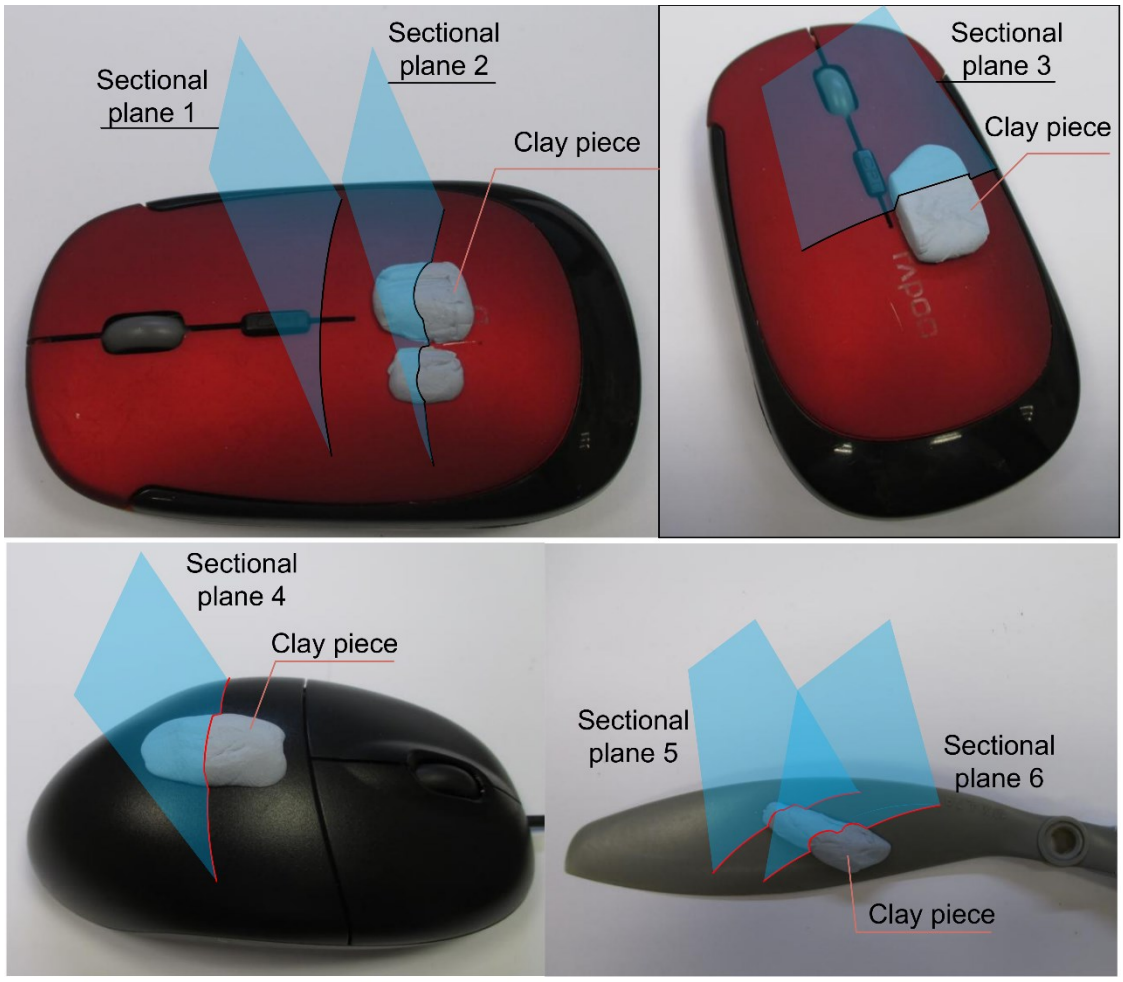


Figure 4.2 Data collection for method demonstration

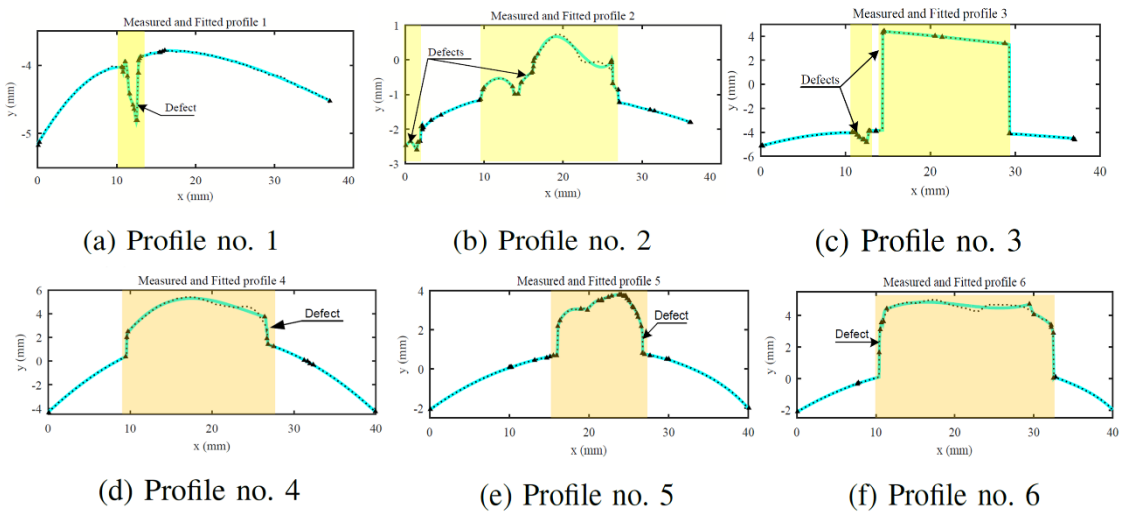


Figure 4.3 Measurement curve profiles

4.2.2.1 Data separation

As illustrated in Figure 4.1 and Figure 4.3, there are some portions of the measurement data that only contains the defect information. In this case, all defect data must be identified and eliminated before performing the estimation steps. As mentioned above, by using B-spline fitting, we can automatically separate the measurement data into small pieces, where each piece either comes from the nominal data or the defect data.

4.2.2.2 Spline piece selection

Not all spline pieces will be selected as the best candidates in the nominal profile estimation process as some pieces may correspond to the defect profiles. In the first selection step, we will eliminate all spline pieces which have lengths smaller than a certain threshold. This is based on the observation that a small spline piece might be a part of defects and eliminating small pieces of data would not be affect the overall fitted curves significantly. The remaining spline pieces are now called “large pieces”.

As can also be seen in Figure 4.3, the defect parts usually have higher curvature than the nominal one, i.e. the radius of curvature of the defect is usually smaller than that of the nominal curve. However, a problem arises in finding the smallest radius of curvature $R_{curvature}$ of the nominal profile (control factor) as we do not have a priori information of the nominal profile. To overcome this issue, we proposed a simple method to estimate $R_{curvature}$ based on circumscribed circle radius, R_{ex}, of the

measured profile. The circumcircle of a measured profile is illustrated in Figure 4.4a. In practice, R_{ex} is computed as the smallest of radii among radii of circumscribed circles $R, (R = OC)$ of triangles, which are formed from the two end points of the measured profile (A and B) and arbitrary points within the measured profile (point C) (Figure. 4.4b).

In typical cases, it is observed that $R_{curvature}$ is about one-third of the radius of the circumcircle of the nominal profile, $R_{curvature} \approx \frac{R_{ex}}{3}$. $R_{curvature}$ is a control factor and it may be adjusted to get a better response based on the curvature information of the profile.

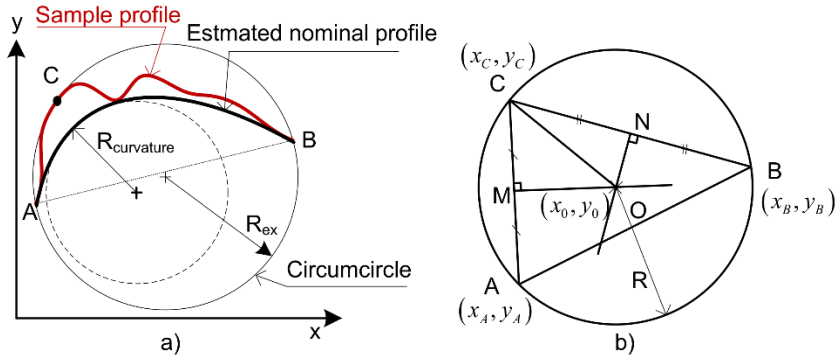
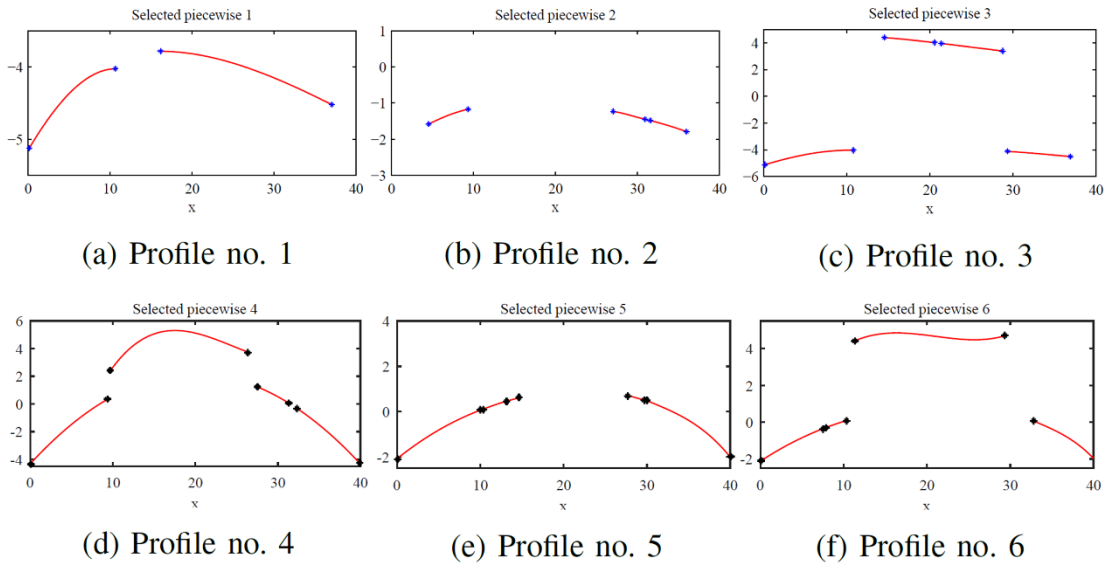


Figure 4.4 Estimation of the smallest radius of curvature of the nominal curve profile

Therefore, at the second step, the minimum radius of curvature of each piece is tested and compared to the predefined smallest curvature radius the nominal profile, $R_{min} = R_{curvature}$. The pieces will survive if their curvature radii are larger than R_{min}.

Figure 4.5 shows the selected spline pieces of the six examples in Figure 4.3. We can see that the data is properly selected for curve Profile 1, 2, and 5. However, the selected pieces of the remaining cases contain one spline piece from defect part (Profile 4 and 6), and two spline pieces (Profile 3). As the spline pieces in the defect parts might pass the acceptance criteria for nominal spline pieces, they need to be eliminated to obtain the true nominal curve profile.



(a) Profile no. 1 (b) Profile no. 2 (c) Profile no. 3

(d) Profile no. 4 (e) Profile no. 5 (f) Profile no. 6

Figure 4.5 Selected spline pieces

4.2.2.3 Spline piece combination

As depicted in Figure 4.5c, d, f, not all the selected pieces belong to the nominal profile. To overcome this false selection problem, a combination set of the selected pieces is generated and sequentially tested to obtain the final data for spline fitting. The piece combination selection respects the following criteria:

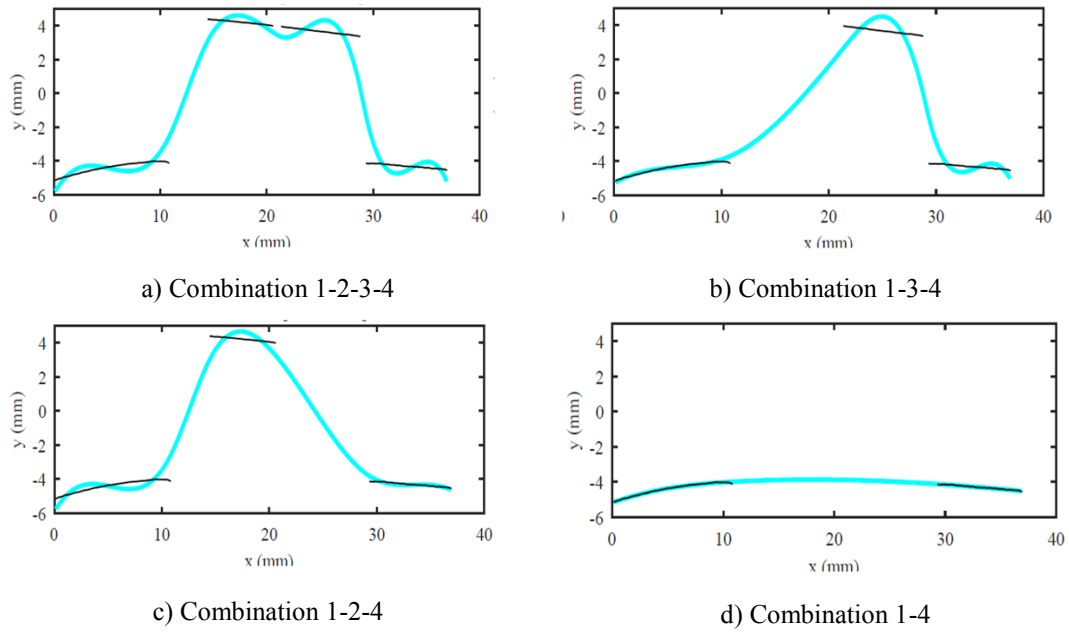
(*i*) The combination sets are selected based on the length of the estimated-nominal-profile (the total curvature length that can be constructed based on the piece selection set), i.e. the total length from left-end of the leftmost piece to the right-end of the rightmost piece. The longer the estimated-nominal-profile, the higher the priority to be selected. In addition, the estimated-nominal-profile must also be larger than half of the total measured profile length.

(*ii*) In a case when there are two or more combinations with the same estimated-nominal-profile length, the higher priority is given to the set that contains more points.

For instance, in the case of Profile 1 (Figure 4.5a), which contains two pieces, only two combinations are available, i.e. (1-2) and (2), where the number represents the selected piece. Please note that piece no. 1 is not selected because the estimated-nominal-profile generated by the piece, which is obviously equal to the length of the piece itself, is smaller than half of the total measured profile length. In the case of Profile 2, when there exists three pieces, then we have three potential combination sets, which are (1-2-3), (1-3) and (1-2). The combinations of (2-3), (1), (2) and (3) are

not selected for the same reason as before, i.e. the estimated-nominal-profile generated by the pieces is smaller than half of the total measured profile length. In the case of Profile 3, the combination sets are (1-2-3-4), (1-3-4), (1-2-4), (1-4), (1-2-3), (1-3), (2-3-4), (2-4), (1-2).

When a combined subset is selected, a spline function is used to fit the given data to estimate the nominal profile. The estimated profile is then examined with the acceptance criteria (will be discussed in subsection 4.2.2.5). If the estimated profile passes the test, the estimation procedure is terminated. If it fails, the next combined subset is used to fit and the procedure will be performed again iteratively.



a) Combination 1-2-3-4

b) Combination 1-3-4

c) Combination 1-2-4

d) Combination 1-4

Figure 4.6 Combination and estimated nominal curve profile for profile 3

Figure 4.6 illustrates the sequence of nominal profile estimation of Profile 3, (see Figure 4.3c and Figure 4.5c). At the first step as shown by Figure 4.6a, the nominal profile (solid line) is estimated from the combination of four pieces (1-2-3-4). This nominal profile does not pass all the acceptance criteria. Subsequently, the next combination (1-3-4) is used to estimate the nominal profile. As also shown in Figure 4.6b, this profile does not also satisfy all the acceptance criteria. Similarly, at the third step, the estimated nominal profile of the combination of (1-2-4) does not pass all the acceptance criteria (Figure 4.6c) and for the last combination, it is found that the estimated nominal profile (Figure 4.6d) satisfies all criteria.

4.2.2.4 Nominal profile estimation

After the selection of the piece sets via the combination process is accomplished, the data for nominal curve profile estimation is available. Up to this point, we already have all the ingredients for estimating the nominal curve, $\hat{F}$.

Let us define the selected input dataset, $\boldsymbol{\Phi}$, which contains the spline piece combinations:

$$\boldsymbol{\Phi} = \{\varphi_1, \varphi_2, \dots, \varphi_n\} = \left\{z_1^{p_i}, z_2^{p_i}, \dots, z_{n_{pi}}^{p_i}\right\} \cup \left\{z_1^{p_j}, z_2^{p_j}, \dots, z_{n_{pj}}^{p_j}\right\} \cup \dots \cup \left\{z_1^{p_k}, z_2^{p_k}, \dots, z_{n_{pk}}^{p_k}\right\},$$

where: φ_i is a data point from the selected spline pieces,

$z_c^{p_i}$ is the input data point c^{th} in the selected spline piece p_i^{th},

$n = n_{pi} + n_{pj} + \cdots + n_{pk}$ is the number of datapoints in the dataset.

To identify the nominal profile, F, a knot vector $\boldsymbol{Z}$ needs to be formed based on the corresponding knots of the selected spline pieces[5], $p_i, p_j, \dots, p_k$.

$$\boldsymbol{Z} = \left(\underbrace{\zeta_{p_i}}_{p_i\ left\ knot}, \underbrace{\zeta_{p_{ij}}^1, \dots, \zeta_{p_{ij}}^{u_i}}_{p_{ij}\ inserted\ knots}, \underbrace{\zeta_{p_j}}_{p_j\ left\ knot}, \underbrace{\zeta_{p_{j\dots}}^1, \dots, \zeta_{p_{j\dots}}^{u_j}}_{p_{j\dots}\ inserted\ knots}, \dots, \underbrace{\zeta_{p_k}, \zeta_{p_{k+1}}}_{p_k\ knots} \right),$$

where:

- $\zeta_{p_{ij}}^k = \begin{cases} \emptyset, & \text{if} \quad j = i+1 \\ \zeta_{p_{i+1}} + \frac{k\left(\zeta_{p_{i+1}} + \zeta_{p_j}\right)}{u_i+1}, & \text{if} \quad u_i \Delta\zeta_{\min} < \left(\zeta_{p_j} - \zeta_{p_{i+1}}\right) \le (u_i+1)\Delta\zeta_{\min} \\ \zeta_{p_{i+1}} + \frac{k\left(\zeta_{p_{i+1}} + \zeta_{p_j}\right)}{u_i+1}, & \text{if} \quad u_i = \begin{bmatrix} 0 & \text{for one missing piece} \\ 1 & \text{for two missing pieces} \end{bmatrix} \end{cases}$
  is the k^{th} inserted knot between the pieces p_i and p_j,
- $\Delta\zeta_{\min} = \min_{v=\{i,j,\dots,k\}}\left(\zeta_{p_{v+1}} - \zeta_{p_v}\right)$ is the minimum length of the selected spline pieces, $u_i \in \mathbb{N}$,
- ζ_{p_i} is the left knot of the spline piece p^{th},
- $\zeta_{p_{i+1}}$ is the right knot of the spline piece p^{th}.

[5] Please note that the first and the last p multiple knots in the knot vector Z that represent the boundary condition are omitted in this representation.

The estimated-nominal-profile, $\hat{F}$, is a B-spline curve defined in the closed interval $[\zeta_{p_i}, \zeta_{p_{k+1}}]$, which is optimized upon the minimization of the following cost function:

$$\boldsymbol{U} = \sum_{\boldsymbol{\varphi_i} \in \boldsymbol{\Phi}} \|\boldsymbol{\varphi_i} - \boldsymbol{\hat{\varphi}_i}\|^2 + \boldsymbol{\lambda} \sum_{\boldsymbol{\psi_i} \in \boldsymbol{\Psi}} \|\boldsymbol{\ddot{\psi}_i}\|^2$$

where $\hat{\varphi}_i$ is a member of $\hat{\boldsymbol{\Phi}} = \{\hat{\varphi}_1, \hat{\varphi}_2, \dots, \hat{\varphi}_n\}$,

$\hat{\boldsymbol{\Phi}} \in \hat{\boldsymbol{F}}$ is a dataset of the fitted B-spline, $\hat{F}$, corresponding to the input dataset $\boldsymbol{\Phi}$

$\boldsymbol{\Psi} = \{\psi_1, \psi_2, \dots, \psi_m\}$ is a dataset of all discrete points in $\hat{F}$, where $\boldsymbol{\Psi}$ is a superset of $\hat{\boldsymbol{\Phi}}$, $\boldsymbol{\Psi} \supset \hat{\boldsymbol{\Phi}}$,

$\ddot{\psi}_i$ is the second derivative of ψ_i, and

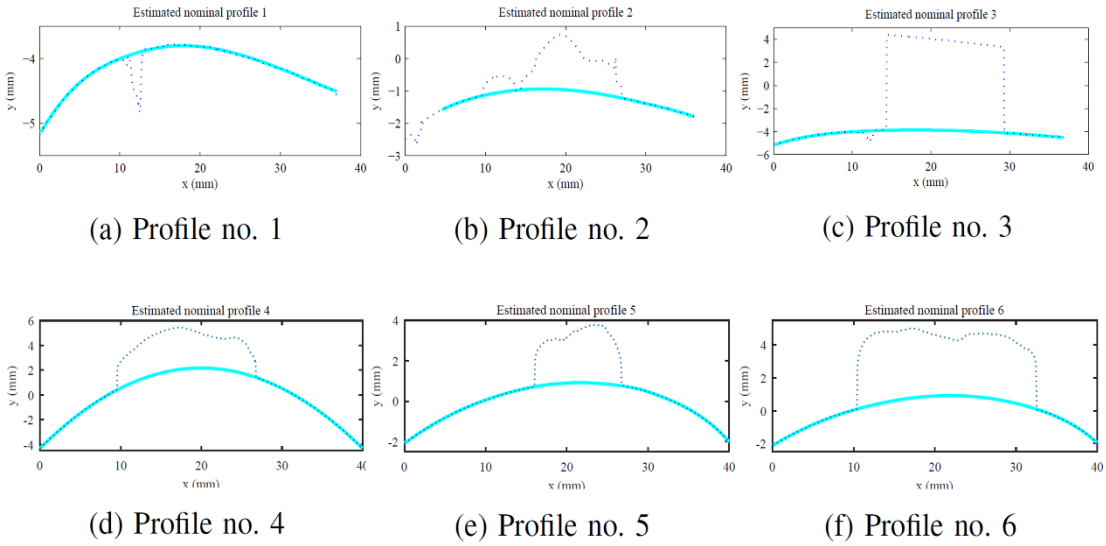
λ is a constant, which indicates the smoothness factor in the cost function (λ can be set at zero when selecting one or two missing cases).

4.2.2.5 Nominal profile testing

As the final test, the estimated-nominal-profile, $\hat{F}$, is subsequently verified by certain acceptance criteria. If $\hat{F}$ does not pass the test, an alternative spline piece combination will be selected and the estimation procedure will be repeated until no more combination is available.

The criteria that $\hat{F}$ needs to satisfy are: (*i*) the total length of $\hat{F}$ must be larger than half of the total length of the input data. This criterion is based on the observation that when the defect portion is smaller than the nominal one, the estimated nominal profile has higher probability to correctly reconstruct the ground true. (*ii*) the smallest radius of curvature of $\hat{F}$ must be larger than the curvature radius threshold, R_{min}, and (*iii*) the maximum fitting error of $\hat{F}$ must be smaller than the given threshold at the selected spline pieces.

Figure 4.7 shows the estimated nominal curve profiles of the given curves. We can see that the estimation is properly identified for all cases. It is noted that all the estimated-nominal-curve-profiles, except Profile 2, result in full length. In the case of Profile 2, the estimation process can only result in the true curve partially. It is because the left end of the profile appears as a defect and no selected data represents the part of the curve.



(a) Profile no. 1 (b) Profile no. 2 (c) Profile no. 3

(d) Profile no. 4 (e) Profile no. 5 (f) Profile no. 6

Figure 4.7 Estimated nominal curve profiles

4.2.3 Algorithm

This subsection summarizes the method for the automatic nominal profile estimation. The algorithm for estimating the nominal profile is presented in Figure 4.8. The working principle is described as follows.

Step 1: The measured data is roughly fitted with a B-spline to eliminate roughness and to split the profile into pieces with non-uniform length by using the bisecting method.

Step 2: The spline pieces, which have length larger than a certain threshold (piece length) and satisfy the radius of curvature criterion, are selected. The measured data are subsequently subdivided to select the nominal profile subset data for estimation based on the selected spline pieces.

Step 3: The selected pieces are combined to estimate a new nominal profile.

Step 4: The estimated profile is checked to see if it satisfies the acceptance criteria. If the estimated profile satisfies the criteria, then it proceeds to the next step, otherwise, go back to Step 3 and select another combination.

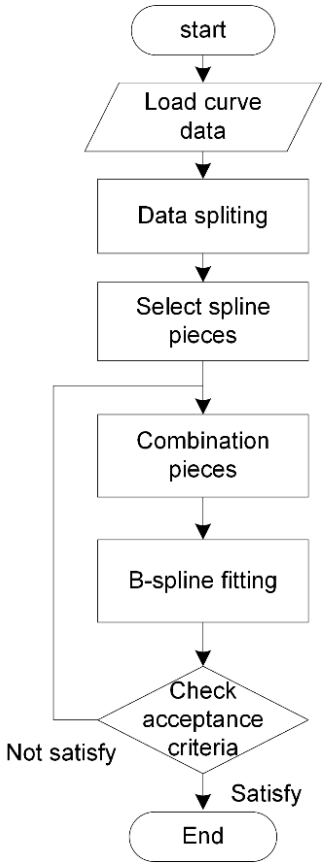


Figure 4.8 Algorithm for automatic nominal curve profile estimation

4.2.4 Estimated nominal profile validation

The validation of the automatic nominal profile estimation is carried out on the last three cases, i.e. Profile 4, 5 and 6. Table 4.1 summarizes the errors of the estimated-nominal-profiles using two different approaches, namely one-piece estimation and two-piece estimation. In the first approach, the nominal profile is estimated based on the assumption that the missing data of the nominal profile is on a single piece of the fitted spline (a spline piece is a portion of spline curve which is given by a spline piecewise function). The second approach assumes that the missing nominal profile data is on two different pieces of the fitted spline. As information about the missing data is unavailable, the missing knot is then assumed to be located in the middle of the missing part. We can see that, in the first case, the maximum error is 0.023mm and the mean square error (MSE) is 1.457e-4mm^2 for the one missing piece approach, the two missing pieces approach is more accurate. In the case of Profile 5 and Profile 6 (which are measured on the surface of a mini-UAV propeller), the estimated-nominal-profiles of the first approach tend to be more accurate than that of the second approach. We

can see that the accuracy of each approach is affected by the profile itself. It is expected that, if the true nominal profile at the missing part is constructed by two spline pieces, the estimation would be more accurate with the second approach. However, the second approach is more sensitive to error due to improper data selection if the data contains defects around the missing part.

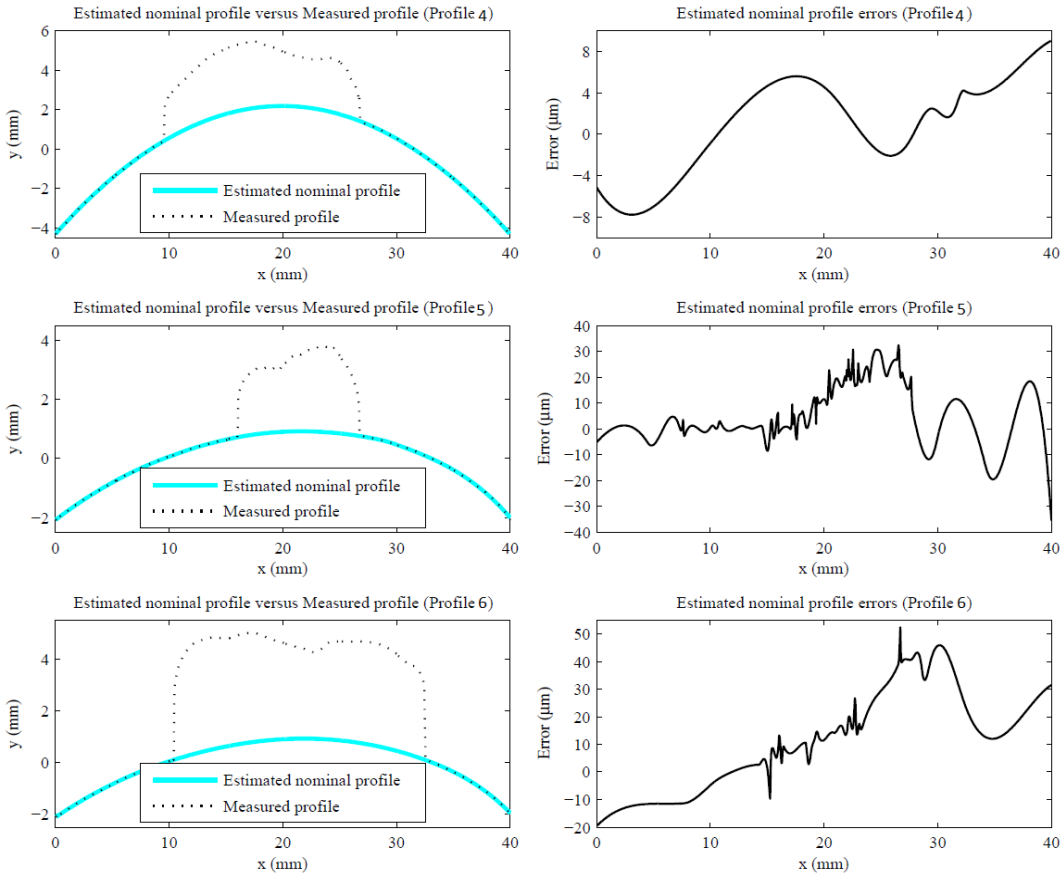


Figure 4.9 Validation of the estimated nominal profiles

Table 4.1 Error in nominal curve profile estimation

	Estimated with one missing piece		Estimated with two missing pieces	
Profile	MSE(mm^2)	Maximum Error(mm)	MSE(mm^2)	Maximum Error(mm)
Profile 4	1.457e-4	0.023	2.199e-5	0.009
Profile 5	1.414e-4	0.036	3.177e-4	0.054
Profile 6	4.477e-4	0.052	5.089e-4	0.048

4.3 Evaluating the estimation error for different sizes and locations of defects

In this section, the effect of defect size and position on the estimation nominal profile is evaluated. The upper half of an airfoil curve (NACA 7210 Airfoil M=7.0% P=25.0% T=10.0%)[140] with chord length of 100mm is used to evaluate the performance of the proposed nominal curve estimation. The data was generated by appending a protruding defect with varying sizes from 0.5mm to 50mm and the defect was located from left (position 0mm) to right (position 100mm). The nominal estimation was performed with a control error of 0.05mm. If the estimated nominal curve deviates more than the control threshold from the ground truth, it is considered a false estimation. The results of the validation are illustrated in Figure 4.10.

Figure 4.10a illustrates the estimation results with different defect sizes and positions. As can be seen in the figure, the proposed method cannot provide exact results when the defect is located on the two sides of the curve. If the defects are located on the left side (see Figure 4.10b) from position 0mm to 5mm, the proposed method cannot provide proper nominal curves. Figure 4.10b illustrates a typical false estimation with the defect size of 10mm at position of 2mm. In this case, the data selection is correct. However, the estimated nominal curve (red curve in Figure 4.10b) has an error larger than the control threshold, which results in a false estimation. In contrast, when the defect is located at the right side of the curve (Figure 4.10c), the method exhibits a better result even when the defect is only 1mm apart from the end point of the curve. The difference in the results is attributed to the nature of the curve. The left side curve has a higher curvature change in comparison to the right one.

In the case of the curve profile of interest, it is observed that the method produces a satisfactory estimated nominal curve profile, if the defect is located in the middle of the airfoil and is smaller than (approx.) 7mm as represented by the dashed-line Figure 4.10a. Increasing the defect size, the probability to result in the correct estimation reduces. Figure 4.10d shows a typical false estimation with large defect size (the defect is 25mm wide at position 20mm). The result here is the same as in the previous case (Figure 4.10b). The method can exactly select the nominal data, but it fails to reconstruct the nominal curve because the resulting curve deviates more than the control threshold.

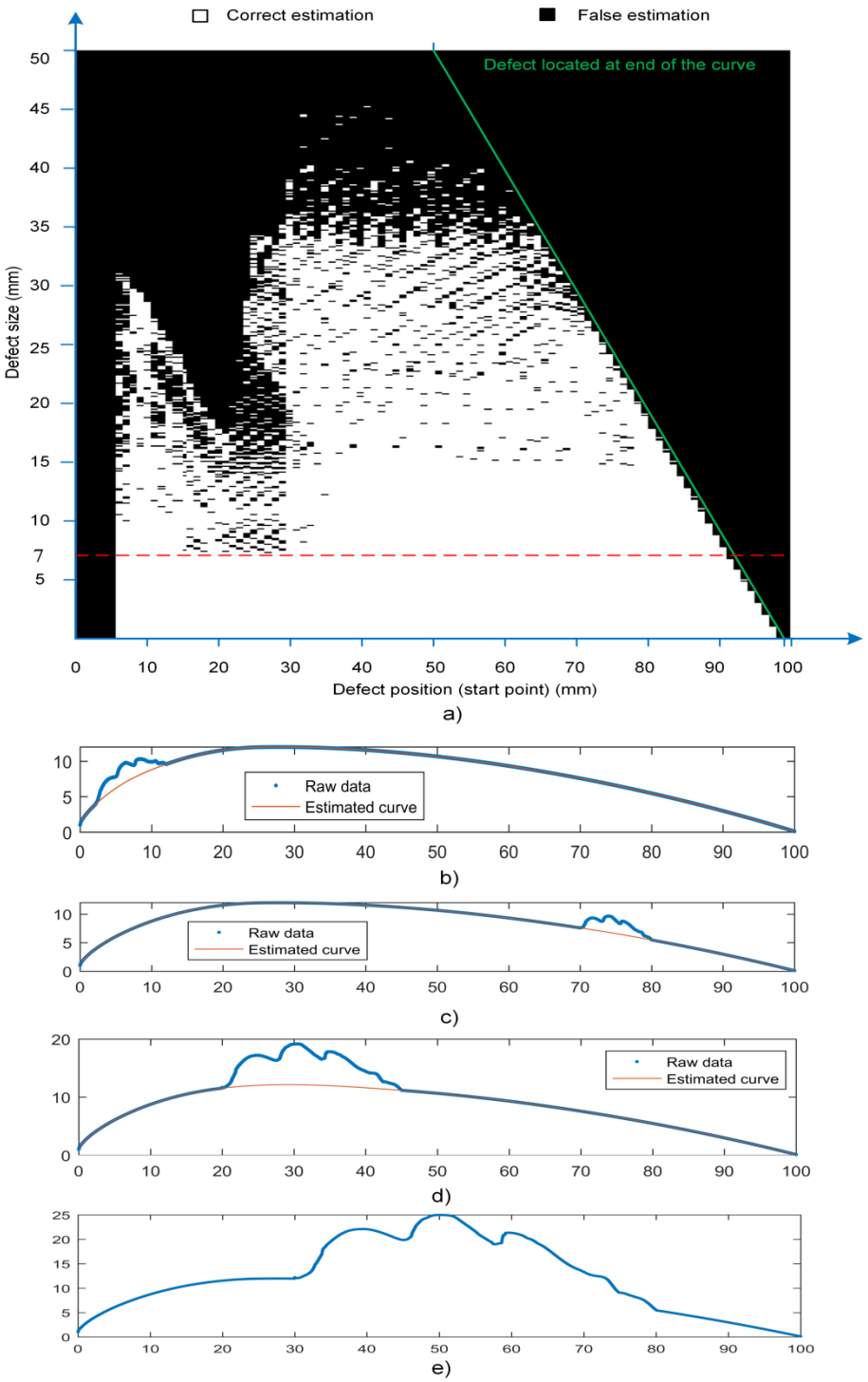


Figure 4.10 Estimation of airfoil curve with different size and defect position a) Estimation results with different defect size (0.5mm-50mm) and position(0-100mm). b) A false nominal estimation curve with defect size 10mm, defect position at 2mm (correct data selection, high estimation error). c) A correct nominal estimation curve with defect size 10mm, defect position 70mm. d) A false nominal estimation curve with defect size 25mm, defect position 20mm (correct data selection, high estimation error). e) Curve with defect size 50mm, position at 30mm. Unable to estimate the nominal curve.

If the defect size is larger than 35mm, the method hardly produces the correct answer. Figure 4.10e shows a typical false estimation with big defect (50%). For such defect cases, the proposed method cannot result in the estimated curve (empty curve) because it cannot find the optimal data combination.

4.4 Nominal surface estimation and defect detection

Complex surface parts in delicate components, e.g. turbine blades, blisks, molding dies, etc., are usually constructed via NURBS in conjunction with some standard surface construction techniques such as loft, sweep, extrude etc. However, in some cases, B-spline can also be used to represent complex surfaces, due to its simplicity and linearity of the mathematical representation.

Given a set of measurement data obtained from a smooth surface, $\boldsymbol{S}$, that contains some data of defective regions of arbitrary shapes, we are to reconstruct a surface $\widehat{\boldsymbol{S}}$ which is the best fit to $\boldsymbol{S}$ and also to indicate the defective regions in $\boldsymbol{S}$. In order to automate the reconstruction process, the following assumptions are made: (*i*) $\boldsymbol{S}$ must be a smooth surface, and (*ii*) the total area of the all defects must be smaller than the area of $\boldsymbol{S}$.

The reconstruction of $\widehat{\boldsymbol{S}}$ using the spline fitting technique requires all the geometrical information of the surface $\boldsymbol{S}$. In order to apply the fitting technique on surface $\boldsymbol{S}$, the defective portions need to be identified and omitted. As the fitting technique will be applied iteratively on every sectional plane (see Figure 4.11) and the intersection of the smooth surface $\boldsymbol{S}$ to the sectional plane will result in a smooth curve, B-spline curve fitting is selected to reconstruct each nominal curve, *F*, of every section.

In practice, a workpiece is digitized by using a three-dimensional camera or a laser scanner. The measurement data is usually in the form of a point cloud and is processed to be a facetted surface. In the surface, sample points are usually located arbitrarily. Therefore, the polygonal surfaces are converted to a depth image data that comprises cross sectional curve profiles of the surface parallel to the OZY plane of the workpiece coordination system (see Figure 4.11). Depth image data is an organized image (matrix) -like structure that stores the surface contour (height in Z axis), where

in this work, its columns correspond to the X coordinate (horizontal axis) and the rows correspond to the Y coordinate (vertical axis).

The method for estimating nominal curve profiles is employed for each curve of the surface. As can be seen in the previous section, the method offers good results. However, in some cases, due to the nature of the defect profiles, e.g. when the defect is relatively wide but shallow, a false combination might get selected in reconstructing a nominal curve, which might result in a non-smooth estimated nominal surface after its reconstruction. To avoid a false nominal surface estimation, the estimation method needs to be able to identify the false estimated curves and subsequently eliminate them.

Referring to our initial assumption that the nominal surface is smooth, in a lateral direction of the first estimated-nominal-profiles in the X-direction (rows of a depth image or the Y-direction in our case, see Figure 4.11), all of the curve profiles in the Y-direction must also be smooth. If there exists a false estimated nominal profile in the X-direction, it means that any of the profiles (Y-direction) will contain a discontinuity that is attributed to false estimated X-direction curves. To resolve the problem, further steps need to be taken to eliminate the error.

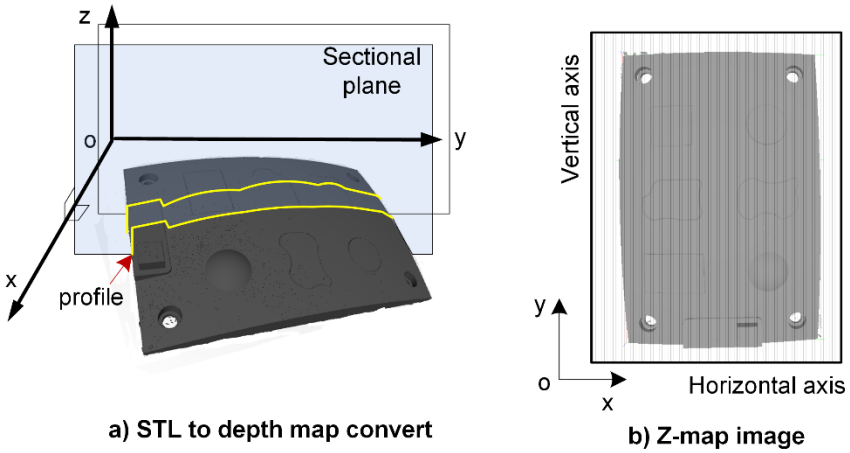


Figure 4.11 Converting measured point cloud to depth image

It is not necessary to repeat all the steps of the first estimation in the Y-direction, even for the combination step to find a better piece combination. In this subsequent step, the first estimated nominal curves (in Y-direction) are now used as the input data for the refining process. New nominal profiles are obtained by processing each row of the image (in X-direction), where the procedure is carried out by simply eliminating the

false data. The procedure includes (*i*) fit the data with a smooth B-spline and evaluate the fitting error, (*ii*) eliminate the input data, where their fitting errors are larger than the prescribed value, and (*iii*) fit a new B-spline from the selected data in step *ii* to obtain the new nominal profile.

After the nominal surface has been estimated, the defects can be straightforwardly identified by subtracting the estimated nominal surface from the measured data. As the intention of the application is for defect detection in automated finishing processes, the defects are only considered if their sizes are larger than a certain threshold. If the error is smaller than the roughness level, the defect will be ignored. In the case that the error is larger than the roughness level, we need to identify the location of the defect for generating tool paths or adding material in case of holes being found. The defect detection is detailed in next chapter.

4.5 Experimental results

This section presents the validation results of the proposed method to detect defects in test components. For this purpose, two experiments were carried out on a simulated CAD geometry and on a real data from a digitized object.

Simulated component

Figure 4.12a shows the input data that is synthesized by creating a CAD model in SolidWorks. The model simulates a realistic component from machining processes, which is composed of two freeform workpieces mounted on a table size of 200x250mm. There are five simulated defects on the workpiece surfaces, where four of them are positive types to simulate weld beads and the fifth one is negative to simulate a dent on the workpiece. The CAD model data is converted to STL to generate a polygonal surface model. The polygonal model is subsequently converted to a 2001x2501 depth image with a resolution of 0.1mm (see Figure 4.12b), and a threshold is used to eliminate the data of the mounting table. The depth image is then segmented by employing Sobel edge detection [141] (see Figure 4.12c).

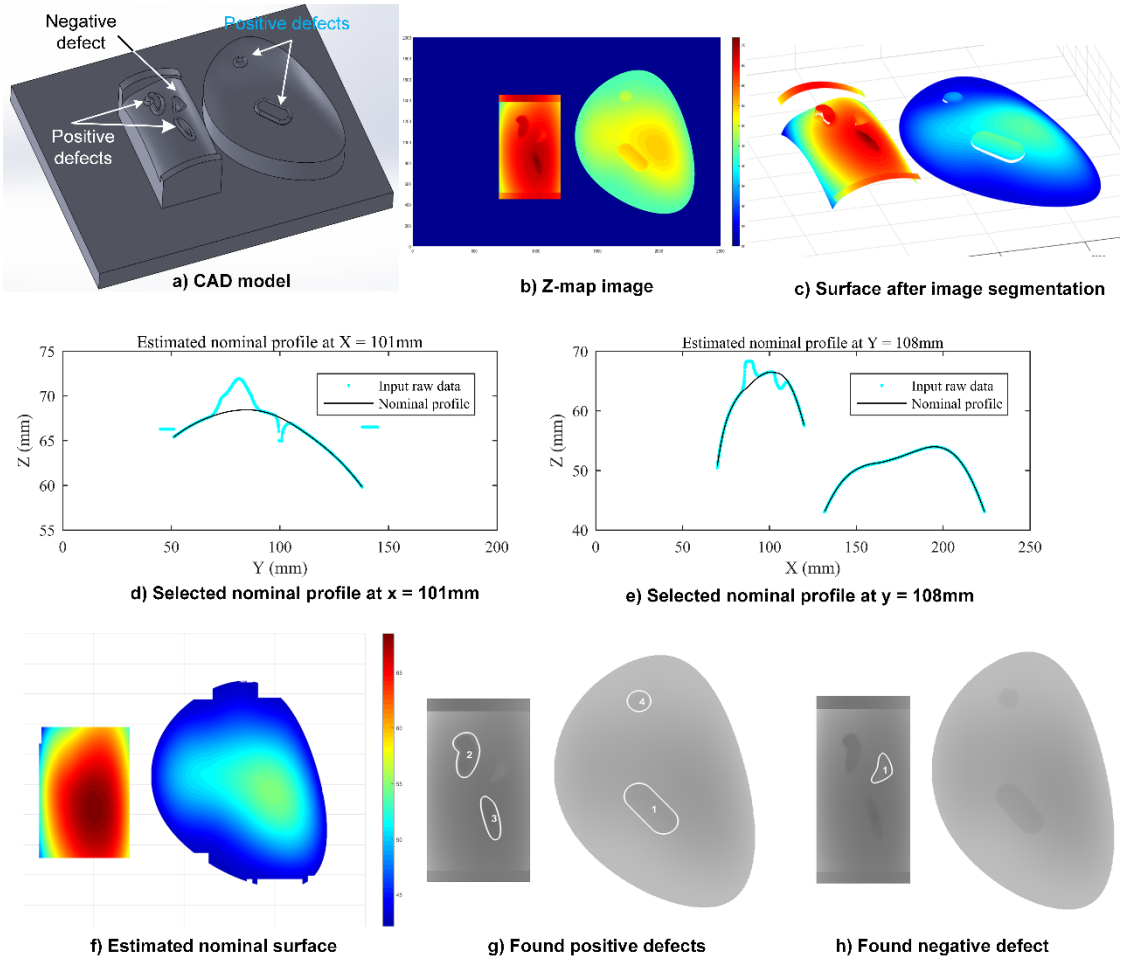


Figure 4.12 Nominal surface estimation and defect detection on synthesis data

After applying the nominal profile estimation in the X- and Y-directions at each column and row of the depth image, the estimated nominal curve profiles are obtained. Figure 4.12d and Figure 4.12e illustrate two of the resulting nominal curve profiles on both directions after the final refining step at two specific locations, i.e. X = 101mm and Y = 108mm, from the reference point. After configuring all the refined estimated curves, the estimated nominal surface is reconstructed, with the result shown in Figure 4.12f. Subsequently, with the nominal surface estimated, the defects are identified and localized accordingly as illustrated in Figure 4.12g and Figure 4.12h.

Real component

In the second experiment, the validation is carried out on a real fabricated test component as shown in Figure 4.13a, where subsequently a polygon surface was generated by using the GOM ATOS 3D scanning system (Figure 4.13b), that results in 978,957 vertices and 1,837,524 facets as illustrated in Figure 4.13c. During the

fabrication, some defects were intentionally introduced on the surface of the component as shown in the figure with the smallest defect size of about 0.5x0.5x0.5mm. A 2624x3129 depth image was created from the polygonal geometry with a resolution of 0.1mm. The proposed method was subsequently applied with the maximum predefined allowable error at 0.1mm.

Figure 4.13 Data acquisition from a test component

As an illustration in the nominal curve estimation procedure, Figure 4.14 shows the process on one of the curves, which was obtained from the digitized data. The measured data in the Y-direction (the row of the depth image) is illustrated in Figure 4.14a that comprises a few defect profiles.

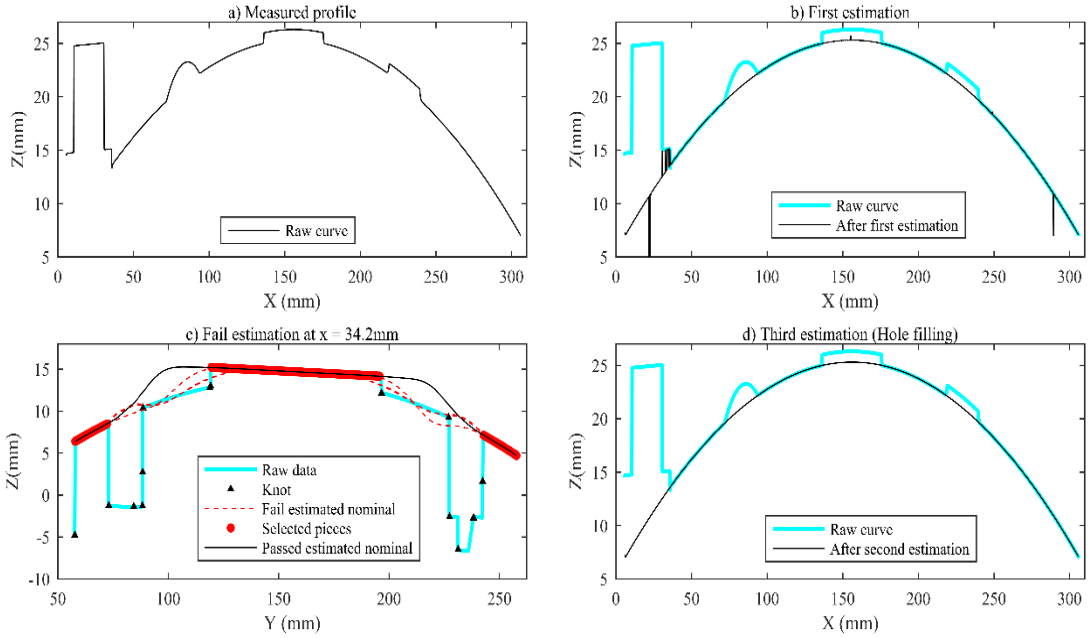


Figure 4.14 Progress in estimation of nominal curve profile

After all the initial nominal curves estimation in the X-direction (columns on the depth image) are performed, conceptually, the nominal surface estimation step can be initiated. However, the initial estimation of the curve obtained in the X-direction might contain some false nominal curves due to the false piece combination problem (see Section 4.4). This will result in spiky profiles when the estimated curves in the Y-direction are reconstructed based on the initial curves estimation (in the X-direction) as illustrated in Figure 4.14b. A few examples of false curve estimation in the X-direction are shown in Figure 4.14c, which are taken at slicing plane X = 34.2mm. To mitigate the problem, the error elimination as discussed in Section 4.4 is carried out, which results in the final estimated nominal curve as shown in Figure 4.14d.

The Z-map image of the component is presented in Figure 4.15a, while Figure 4.15b illustrates the estimated nominal surface after the application of the proposed method, which results in a smooth surface with all the holes and positive defects removed. Accordingly, the positive defects can be obtained by subtracting the nominal surface from the depth image using image subtraction. In Figure 4.15c, it is shown that all positive defects are successfully identified after roughness suppression. The result shows that all the positive defects on the surface are correctly identified, including the smallest one, i.e. the defect number 8. Similarly, the negative defects (the four holes) are also successfully identified as shown in Figure 4.15d.

Furthermore, the boundaries of the defects were accurately identified without duplication (multiple contours) or discontinuity in the contours (Figure 4.15c). It needs to be highlighted that for a gentle slope defect, the boundary can hardly be obtained by conventional edge detection techniques, but it is well-identified by the proposed method. In addition, the processing time for the nominal surface estimation and defect detection from the corresponding depth image is about 20 seconds on a core i7-6700HQ CPU laptop with MATLAB 2016a, which is reasonable for machining application.

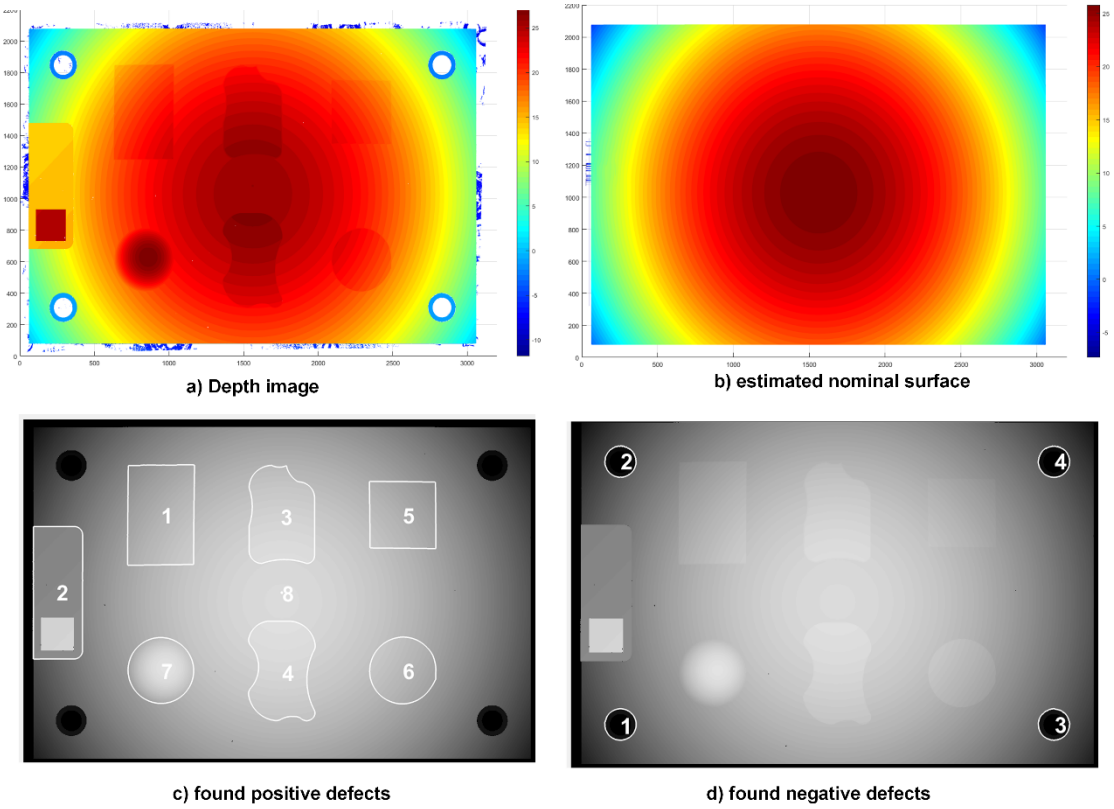


Figure 4.15 Estimated nominal surface and found defects

4.6 Conclusion

This chapter presents a nominal surface estimation process that is capable of localizing and quantifying defects from measured data. The process has a potential to be used in automatic remanufacturing processes for refurbished components. A defective component is initially digitized by 3D cameras or laser scanners to acquire its facetted geometry and, afterwards, the model is converted into z-depth images. Each row and column of the depth image are used to estimate the nominal curve profile based on B-spline fitting. The curve is split into spline pieces to select appropriate data for nominal profile estimation. Subsequently, the resulting estimated nominal curve is corrected, if necessary, by eliminating the false estimated ones. Finally, the estimated nominal surface is obtained by fitting the depth image using a smooth B-spline surface technique, which will also result in the identified defects.

Compared to other techniques found in literature, the experimental results show that the proposed method can exactly find the defects on free form surfaces, even for a very small defect. However, some limitations of the proposed method have to be highlighted, which are attributed to the assumptions taken for the workpiece surfaces,

i.e. (*i*) the method might not be applicable to detect some defects that are located close to the edges or tips of the workpieces, and (*ii*) defects must be relatively small compared to the workpiece surfaces.

The estimated nominal surface and defect information obtained from the proposed method can be used for automatic tool path generation in remanufacturing processes. Furthermore, the method can also be extended for denoising and filling holes in surface reconstruction. The implementation of the proposed method on real remanufacturing processes will be discussed in the next chapter.

Chapter 5: Framework for automatic reprofiling in remanufacturing processes

This Chapter discusses the proposed framework to automatically estimate nominal surfaces from measurement data. The main aim of this framework is to develop a system that is capable of processing digitized geometric data to estimate the nominal geometries, localizing weld beads/defects, as well as generating CNC tool paths for reprofiling with minimal human intervention. This chapter presents the algorithms to build a complete application program.

5.1 Framework for automatic reprofiling processes

In an effort to partially automate the remanufacturing processes for freeform surface workpieces (components) with positive defects on the surfaces, when there is no design geometry available or when geometry of the parts is distorted, a dedicated framework for automatic reprofiling processes is proposed. For this purpose, a number of steps are required that includes digitizing the workpiece geometry, processing the measured data to estimate the nominal surfaces of the parts after finishing, identifying defect locations and generating tool paths for CNC (Computer Numerical Control) milling to remove the defects.

The reprofiling process comprises various steps involving different machining stages. To automate this process, data communication between steps needs to be carefully formatted to minimize the setup time. In this strategy, the digitizing process is performed on the workpiece mounted on a reference table as illustrated in Figure 5.1. Consequently, the reference table geometries will also be digitized. Most digitizing equipment nowadays is based on optical measurement (i.e. laser scanners, or 3D cameras) which can acquire a huge amount of measurement data in a very short time.

When the measurement data contains information of the reference table, we can set the Workpiece Coordinate System (WCS) origin at a specific vertex of the table. As the reference table is a simple shape, it is easy to mount the table on different machines while keeping the same reference coordinate system during the reprofiling process. Figure 5.1 also illustrates the principle of the mounting and transferring the workpieces through various stages. Firstly, the workpiece is mounted on the reference table at the mounting station. The whole table system (table, mounting supporters, and

the workpiece) is transferred to the digitizing stage by a conveyor. After reaching the measurement station, geometrical information of the table is captured. The measured data is subsequently transferred to an automatic nominal profile estimation and defect detection program to generate tool paths for CNC milling machines. At the same time, the table is transported to the CNC machining station, where the table is mounted to the machine by an electrical magnet.

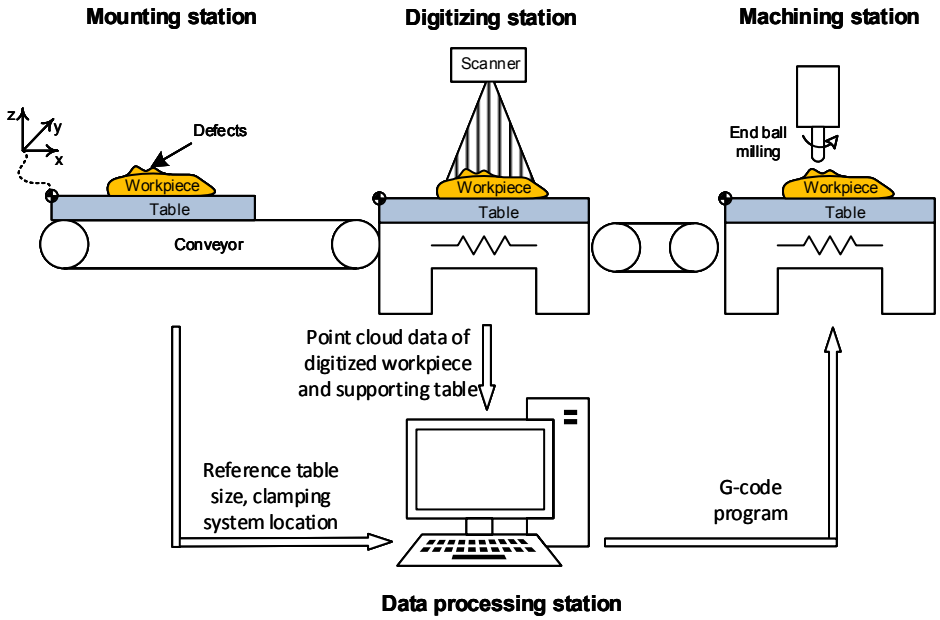


Figure 5.1 A proposed framework for automatic defect removing system in remanufacturing process

Finally, the CNC machine receives the tool paths from the data processing station and starts the process to remove all the identified defects. In this framework, the operator only needs to confirm the estimation results in the estimation processes. After that, the system will automatically generate tool paths based on the estimated nominal surfaces and found defects.

Automatic nominal surface estimation and defect detection is the heart of the proposed method. The program comprises three tasks: *(i)* the first task identifies the table position in the measured point cloud data automatically, in order to establish the transformation matrix to transform all the measured data to the new WCS, *(ii)* the second task estimates the nominal profile surfaces and identifies the defects automatically and finally, *(iii)* a tool path generation program is executed for CNC to remove all identified defects.

Figure 5.2 depicts a framework of the automatic data processing. After the point cloud data is obtained in the digitizing stage, the WCS transformation matrix is evaluated. First, all possible planes in the point cloud data are identified. Subsequently, plane data that is a part of the reference table is selected based on the initial information of the table dimensions. The selected planes data is fitted with the mathematical models of the reference table to identify the WCS transformation matrix accurately. In the second task, the point cloud data is processed to extract the information of the workpiece itself and the workpiece point cloud data is converted to a depth map image for automatic estimation of the nominal surface and defect detection. By processing each row and, subsequently, each column of the depth images, the estimated nominal surfaces of the part are coarsely identified. The next step is to confirm the nominal profile surface through B-spline surface fitting and, finally, defects are localized for tool path generation.

Each identified defect needs a confirmation from the operator to exclude false estimations. A tool path program will be locally executed at the confirmed defects to generate the G-code program for machining. Detailed discussion in the implementation of the proposed framework is presented in the next section.

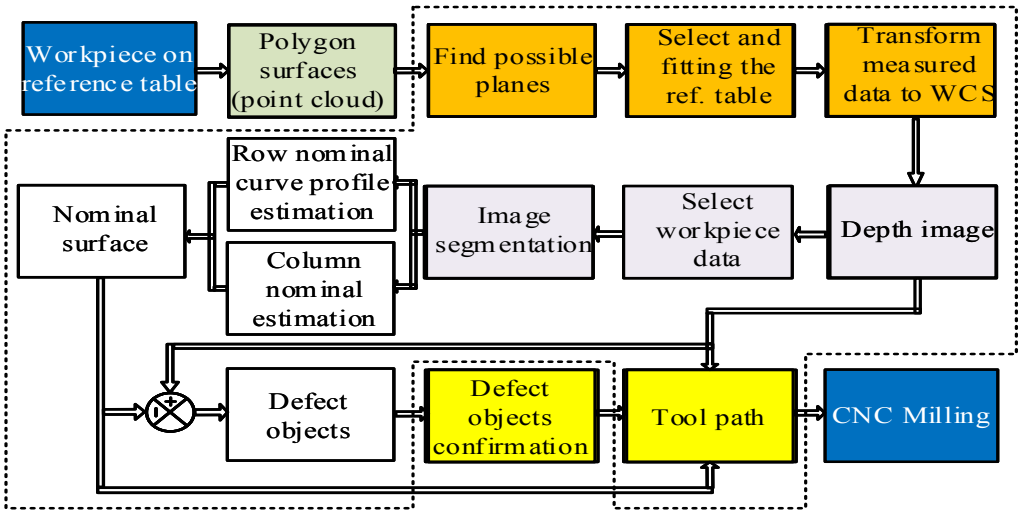


Figure 5.2 Data flow for automatic reprofiling in remanufacturing. The first two blocks at the left top are externally implemented during the digitizing stage, the next three orange blocks are for WCS identification, the next three gray blocks are for data preparation, the four white blocks at the left bottom are for automatic nominal estimation and defect detection, the two yellow blocks at the bottom are for automatic tool path generation, and the last blue block is externally implemented on CNC machine. All blocks inside the dashed box are automatically executed by the program itself.

5.2 Workpiece preparation

Remanufacturing processes a retired component to recondition it as a new non-defective component. Two key points of remanufacturing are:

(i) During its service, the component will undergo geometrical and microstructural changes from its initial stage. These geometrical changes might be caused by many factors such as distortion under working pressure, thermal distortion, material lost due to friction and dents because of hitting foreign objects. The microstructural changes might be attributed to the increasing of grain size under high temperature environment, microcracks due to fatigue, fast changing temperature or pitting corrosion under harsh environment.

(ii) The output of the remanufacturing process should result in a well reconditioned component in terms of mechanical properties and physical geometries.

The most important task in the workpiece preparation is the visual inspection. After a cleaning stage, there is a need to initially check the deviation of its geometry and microstructural conditions.

In the visual inspection, the geometrical distortion of the component needs to satisfy a certain tolerance. The tolerance is component dependent, which also depends on its working condition. The geometrical discrepancies caused by wear and collision to a foreign object usually cause the component to be out of the tolerance. If the defect is too large, the component cannot proceed to further processing.

Microcrack testing is important to check the mechanical strength of the parts under working load. The microcracks need to be repaired to eliminate stress concentration that causes crack propagation. Crack dimension is an important figure to determine whether the component can proceed to the next step. For example, if the length of a crack is larger than 0.5mm, the turbine blade will be scrapped. Typically, microcracks are identified by using fluorescent penetrant inspection.

The identified microcracks, dents, wear and scratched marks have to be fused by appending the same material using TIG (Tungsten Inert Gas) welding, laser cladding, etc. After the filling process, on the repair sites, excessive material forms protruding defects on the component surfaces that need to be eliminated to retrieve the designated finish surface. The process of removing the protruding defects is currently carried out

by skilled workers, which is very time-consuming, especially, when the components appear in freeform shape.

Components with freeform profiles such as turbine blades, propellers, forging dies, etc. might have very complex shapes. Mounting different components on machines requires different jigs. For instance, turbine blades require a special jig mounting the root of the blade and using support pins to keep the blade withstanding the cutting force during the machining process. Figure 5.3 illustrates a mounting jig for machining a turbine blade tip.

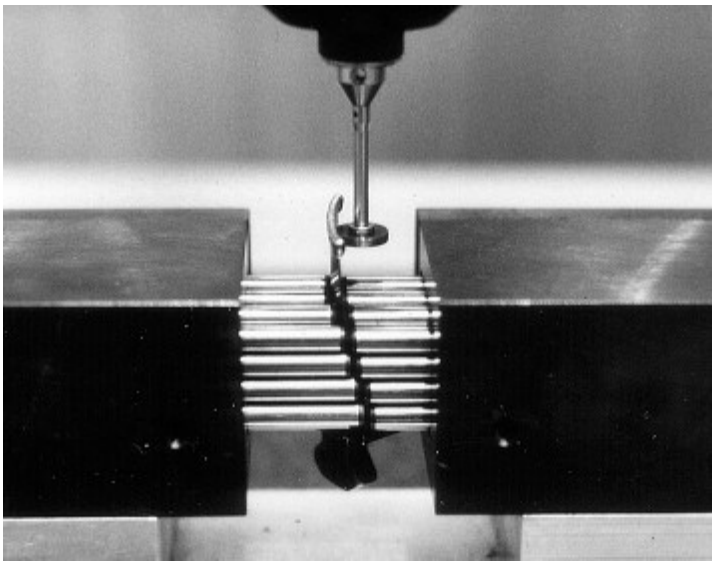


Figure 5. 3 Repairing tip of a turbine blade with mounting jig using support pins on the turbine surfaces [142]

In the case of a blade with protruding defects on its surfaces, it is mounted horizontally (the blade with machining surface up). In this study, due to resource limitations, a 3-axis milling machine is used to demonstrate the working principle of the automatic reprofiling in a case of machining surface with positive defects.

5.3 Identification of workpiece coordinate system

Adaptive manufacturing relies on the measured data to perform all the processes when the designed geometries of the workpieces are not available or when the nominal geometries deviate from the designed geometries. The workpiece geometries can be obtained through digitization using measurement tools, such as Coordinate Measuring Machines (CMM), laser scanners or 3D cameras. Subsequently, the geometry of the workpieces can be obtained by processing the measured data.

To automate the processing of the data, the workpieces need to be kept in the same coordinate system in all machining steps, including measurement, machining and final inspection. We use a reference block to assign the coordinate system for the entire process and it is made as an individual block, separated from the workpiece, for easy set up.

After the initial digitization process, the measured data not only contains the workpiece data, but also the data of the table. There are some measurement machines such as the GOM ATOS scanner that can be portably mounted to perform the measurement quickly on a complex workpiece. However, such a portable device might introduce some problems in the measured data, e.g. the WCS becomes floating. Therefore, the first part in data processing is to identify the reference table in the point cloud data to find the point for the WCS origin for machining purposes.

Assuming that the workpiece is mounted on a reference table with known dimensions, we need to identify the location of the table in the measured point cloud. The identification of the table geometry can be performed through the following procedure.

(i) Use MSAC (M-estimator SAmple Consensus) to find any possible planes in the point cloud data [143].

(ii) Calculate the distance between a pair of parallel planes, and re-select data of the corresponding pair of planes by using a fitted pair of parallel planes. The plane models are obtained by fitting the corresponding dataset by the two parallel planes model.

$$\begin{cases} c_1 + n_x x + n_y y + n_z z = 0 \\ c_2 + n_x x + n_y y + n_z z = 0 \end{cases} \tag{5.1}$$

where $n = [n_x, n_y, n_z]^T$ is the normal vector of the parallel pair planes and c_1, c_2 are distances of the planes from the origin. Equation (5.1) can be rewritten in a matrix form as:

$$\begin{bmatrix} 1 & 0 & X_1 & Y_1 & Z_1 \\ 0 & 1 & X_2 & Y_2 & Z_2 \end{bmatrix} [c_1 \quad c_2 \quad n_x \quad n_y \quad n_z]^T = 0 \tag{5.2}$$

where X_1, Y_1, Z_1 and X_2, Y_2, Z_2 are the vectors of X, Y, Z coordinates of the two planes respectively. Equation (5.2) cannot be solved by Least Mean Squares

(LMS) method because the problem is trivial, i.e. in a form of $AX = 0$. Moreover, the solution of the equation is the kernel (null space) of a mapping data between the two parallel planes. However, we cannot use traditional null space solver because the data is redundant (the number of equations is larger the number of variables) and noisy ($AX = \epsilon \neq 0$). Therefore, Singular Value Decomposition (SVD) [144] is used to find the best pair of parallel planes that approximates the dataset with minimum fitting errors.

(iii) Find any set of plane pairs that satisfy the distance of the reference table. Then, select proper planes that belong to the reference table to obtain three normal vectors of the reference table. Usually, the three normal vectors are not perfectly orthonormal due to measurement errors.

Let $N = [n_1 \;\; n_2 \;\; n_3]_{(3\times3)}$ be the matrix of the normal vectors, where n_1, n_2, n_3 are the normal vectors of the three candidate pairs of parallel planes of the reference table. We then obtain the orthonormal normal vector of the table, N_1 , by (5.3)

$$\begin{aligned} [U, S, V] &= SVD(N) \\ N_1 &= UV^T \end{aligned} \tag{5.3}$$

(iv) Based on the new normal vector set, N_1, the mathematical model of the reference table is obtained by solving (5.1). In this step, equation (5.1) is modified into

$$\begin{cases} c_1 + n_x x + n_y y + n_z z = 0 \\ c_1 + d + n_x x + n_y y + n_z z = 0 \end{cases} \tag{5.4}$$

where d is the distance between the two planes.

Equation (5.4) can be rewritten as:

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} c_1 = -\begin{bmatrix} 0 & X_1 & Y_1 & Z_1 \\ 1 & X_2 & Y_2 & Z_2 \end{bmatrix} \begin{bmatrix} d & n_x & n_y & n_z \end{bmatrix}^T \tag{5.5}$$

Therefore, we can find the approximated c_1 value as

$$c_1 = \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^T A \tag{5.6}$$

where $A = -\begin{bmatrix} 0 & X_1 & Y_1 & Z_1 \\ 1 & X_2 & Y_2 & Z_2 \end{bmatrix} \begin{bmatrix} d & n_x & n_y & n_z \end{bmatrix}^T$.

Subsequently, vertices of the table are obtained along with the new WCS. Finally, the transformation matrix from the current coordinate system to the WCS is identified and the measured data are transformed to the new coordinate system.

5.4 Data preparation for nominal surface estimation

The main purpose of the data preparation step is to provide appropriate sets of data for nominal surface estimation. As mentioned in Chapter 4, the automatic nominal surface estimation process relies on B-spline curve fitting. The estimation would only work if the input data is a single smooth surface patch. Therefore, the measurement data (point cloud) will be converted to a form that is suitable for the next step.

The data points do not contain any complete curve profiles. Hence, the measurement data needs to be resampled to convert the data to a form of a set of curve profiles. We employ depth images to represent the measured data after projecting along the Z coordinate axis. In the depth image, the horizontal axis is the X-axis, the vertical axis is the Y-axis, and the values of the pixels represent the height values in Z-axis.

As the component is mounted on the surface of the reference table, after transforming to the WCS frame, all the height values of the component must be positive. Therefore, data from the table and other support system can be easily eliminated if their height values are negative. After removing the negative values, the information of the support systems still appears in the depth images. To keep only the component data, a mask is applied to the depth image to remove the information of the support system as their locations are already known.

As the component might contain a few smooth patches, we need to segment the image into a set of smooth patches. This step is needed because the current algorithm for automatic nominal surface estimation (as discussed in Chapter 4) is limited to a single smooth surface.

The following steps summarize the procedure for data preparation:

(i) Convert point cloud data to a depth image by resampling the measured data (usually in triangular meshes). Depth images are a perfect format for storing two-

dimensional curve data of three-dimensional objects. We, therefore, employ the format for further data processing. The depth image is created by projecting the polygonal surface in the Z direction with pixel height calculated from the triangular facets of the polygonal surface.

(ii) Apply a mask on the depth image to remove the information of the reference table and support. This step is required to keep the data from the component clean. The masking image is predefined based on the information of the reference table and the clamping system. This step is automatically performed because the location of the reference table and support in the depth image are known after the data was transformed to the new WCS.

(iii) Apply Sobel edge detection [141] to the depth image to find the borders of smooth surface patches. This is the first step to segmenting the depth image into many smooth patches. The results of the Sobel edge detection are a set of points which has a significant change in slope from their neighbors. It is observed that the patch borders (edges) are usually discontinuous (not closed curves). This might be caused by inappropriate pre-selection of thresholding and/or smooth transition from patch to patch.

(iv) Inflate the edges to totally separate surface patches to overcome edge interruption. All the edges are then eliminated to keep only surface patches. Figure 5.4 shows the edge interruption after performing the Sobel edge detection.

(v) Calculate and sort the surface patches based on the total areas of the patches by using Moore-Neighbor tracing method [145]. The area of each surface patch is calculated by counting the number of pixels in it. Subsequently, the surface patch datasets are sorted by their total area. Finally, the surface patches with area larger than a certain threshold are selected as the candidates.

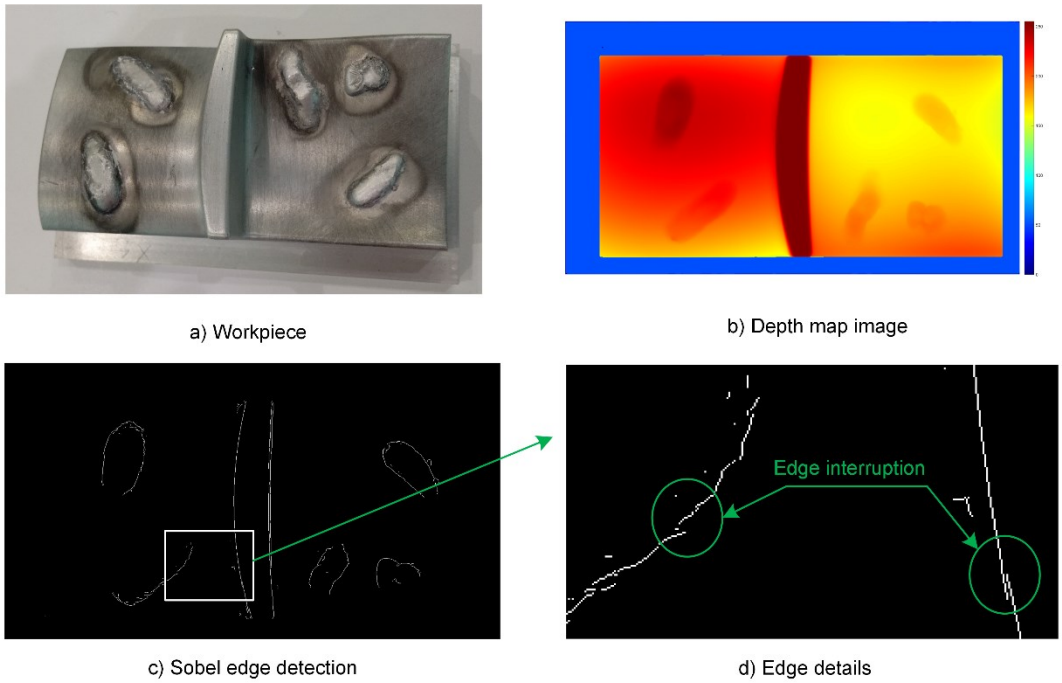


Figure 5.4 Edge interruption phenomenon. a) component/workpiece, b) depth image of the workpiece after projecting, c) edge detection results, d) details on the found edges

5.5 Nominal surface estimation and defect detection

Each of the selected smooth surface patches will be processed to identify the nominal surface. As discussed in Chapter 4, the nominal surfaces are automatically obtained by employing B-spline curve fitting on curve profiles of the surface. This section summarizes the process to obtain the estimated nominal surfaces.

The nominal surfaces can be estimated based on successive curve profiles estimation in one direction. However, the results of the process might contain some false estimation curves as discussed in Section 4.5. We need to eliminate the false estimation data before confirming the nominal surface. The simplest way to confirm a point that corresponds to the nominal surfaces is by comparing the discrepancies of the measurement data and the corresponding points of, at least, two independent estimated nominal surfaces in different directions. As a result, two nominal surfaces that are independently estimated are used for the confirmation. The first estimated nominal surface is from the rows and the second comes from the columns of the surface patch respectively.

After data selection, the estimated nominal surface can be obtained through the B-spline surface fitting. However, B-spline surface fitting is computationally costly and

it limits the use of this technique in a real-time application. Instead of using a B-spline surface to obtain the data points of nominal geometry in the depth image format, B-spline curve fitting is used to approximate the nominal surface from the row and the column data. This will result in two approximated nominal surfaces. Finally, the estimated nominal surface is approximated by taking the average of the two estimated nominal surfaces for all corresponding points.

The estimated nominal surfaces provide target geometries of the component and the input measurement data can be considered as the stock material in normal machining processes. Any discrepancies between the measurement data and the estimated nominal surface need to be removed.

In practice, it is less likely to get zero errors in between the estimated nominal surface and the measurement surface. A threshold to decide whether the differences can be disregarded is introduced. The threshold must be selected larger than the noise level of the measurement data to eliminate exceptionally small defect objects that cannot be removed in machining. The error surfaces, which are in the form of depth images, are converted to binary images to visualize the defects for the operator to confirm.

The defects are characterized by its borders which are obtained by boundary tracing method. Additionally, we only consider defects that are larger than a certain threshold area (e.g. $1mm^2$) for machining, due to the assumption that defects smaller than the threshold cannot be removed in a milling process.

Confirmation is necessary as the result might contain false estimation when identifying the nominal surfaces. This might also happen when the component contains some features that could be considered as defects on the surface. If there is no human supervision on the results of the estimation process, the program would result in false surface estimation and it might damage the component.

5.6 Toolpath generation

Upon the completion of the defect estimation, all confirmed defects need to be removed to finish the repairing process. This sub-section discusses an approach to remove the defects by using a three-axis milling machine. The procedure to generate three-axis tool paths for a freeform component is a standard procedure employed in any CAM software. Commonly, CAM software is designed for a new product

manufacturing, which utilizes CAD models as the stock and target models. The target data and stock geometries in adaptive machining, however, are discrete point cloud data. It is inefficient to convert all the results from point cloud data to a continuous model in standard formats that is normally used in CAM software.

Furthermore, remanufacturing processes are normally conducted locally at restricted parts of the component. Therefore, it is also inefficient to use CAM software to generate the CNC program. For G-code generation, the following procedure is used to generate a local CNC program. The details of the procedure are provided in Appendix 6.

In summary, the following procedure is used to generate 3-axis tool path:

i) Assign the estimated nominal surfaces as the target and the measured surfaces as the stock geometries.
ii) Calculate discrepancies between the stock and target geometries to identify machining position.
iii) Select cutters and machining parameters.
iv) Calculate offset surfaces to decide intermediate cutting layers.
v) Generate tool path pattern to create the intermediate surfaces.
vi) Connect each path pattern to create tool path.
vii) Simulate the tool paths to check tool violations.
viii) Generate CNC code.

5.7 Parameter optimization

Reconstruction of nominal surfaces from the data of a damaged component in the absence of design information is a multi-solution problem. Each of the solution is optimized respecting a certain cost function (i.e. estimation criteria), which might result in different estimated surfaces from the ground truth one. The estimation process has a number of adjustable parameters which need to be optimized to minimize estimation error of the nominal surface.

There are two main steps in the estimation of nominal surface from the data of a damaged component. The first step is to properly select the partial data of the nominal surfaces that can be retrieved from the input data and the second one is to reconstruct the nominal surface from the selected data. The accuracy of the data selection depends

on the parameters for B-spline fitting and the curvature ratio between the nominal to the circumcircle of the input damaged surface, while the accuracy of the reconstructed surface relies on the defect size of and the smoothness term in B-spline fitting.

The selection of partial data that contains the nominal surfaces of the component, which will lead to the elimination of defective profile of the component, is a B-spline based fitting process. The first parameter in the B-spline fitting that has to be properly chosen is the control error to fit the measured curve data with a B-spline function. The control error should be set based on the information of the digitized error during the data collection. If the control error is set smaller than the digitized error, the B-spline will be over-fitted. That will create too many small spline pieces, which in turns will cause improper data selection. On the other hand, if the control error is larger than the digitized noise, the curve is under-fitted, that leads to the incapability of the B-spline in separating the nominal data from the defective data. Therefore, the control error is usually set around 2-3 times of the digitized error.

The size classification of the spline piece is defined based on its relative length to the average length of the spline segments. If the length of a spline piece is larger than the average length of all spline pieces, it will be classified as a large one. Subsequently, a large piece will be considered to be a part of nominal geometry if its minimum radius of curvature is larger than a control threshold. The control threshold is set to be proportional to the radius of the circumcircle of the measured data and, generally, the proportion is around 20 to 50%.

Reconstruction of nominal curve/surface from the data of a damaged component is performed by interpolating the surface on the defective area from its surrounding nominal surfaces that were previously identified. The shape of the estimated curve/surface varies according to the smoothness factor of a cost function during the fitting. A large smoothness factor creates curves with small curvatures and otherwise for those with a small smoothness factor. This value should be properly selected depending on the target curve/surface. In a case of surface with small defects, the suggested value would be around 0.01 to 0.1.

Prior to estimating the nominal surface, the measured point cloud data is split into smooth surface patches. Some parameters also need to be set, which include dimensions of the reference table, position of clamping supports, the smallest size of

smooth surface patch, the smallest size of defect and the inflation size of the edge detection. The information of the reference table and clamping supports depends on the actual mounting jig. The smallest size of smooth surface patch and defects are selected based on the information of the workpiece, while the inflation size of the edge detection is usually selected around 2 to 3 pixels from the found edges.

Chapter 6: Experiments

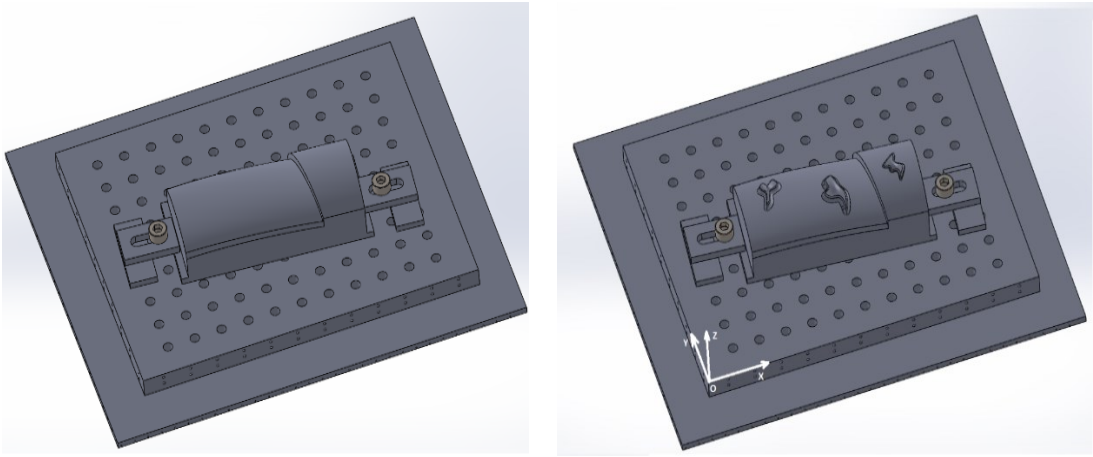
This chapter presents the results of the framework for automatic defect detection and machining processes to estimate the nominal surfaces of components. In the first part of this chapter, we will discuss simulation data generated in CAD software that is used to evaluate the performance of the framework followed by a demonstration of the implementation on measured data from real test components, which were fabricated with some simulated defects.

6.1 Simulation

6.1.1 Data generation

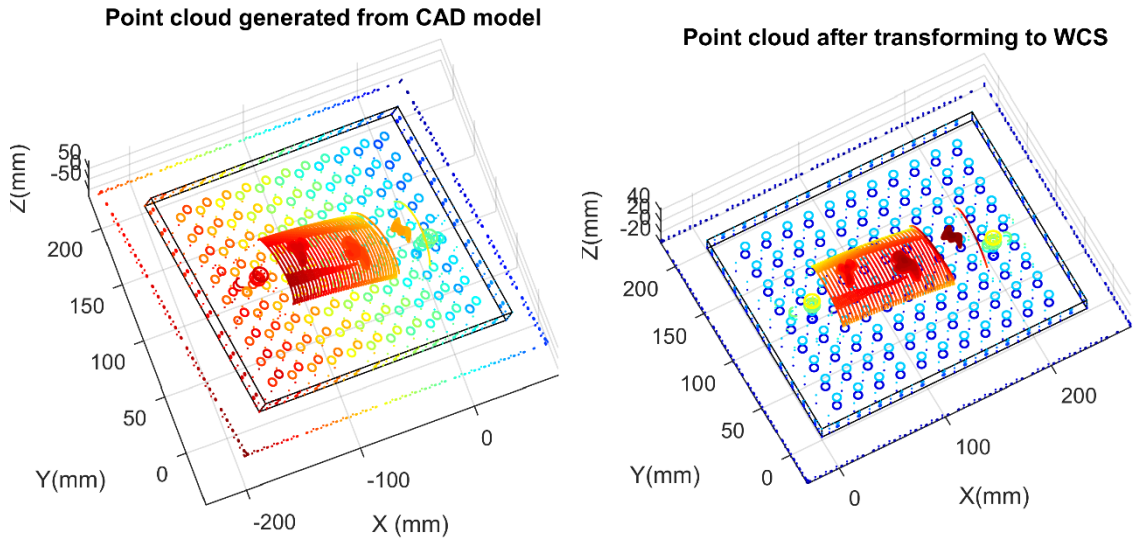
A freeform surface component with two separate smooth surface patches was drawn in Solidworks as shown in Figure 6.1a. Three positive defects were then artificially introduced on the surfaces to simulate weld beads as illustrated in Figure 6.1b. A 250×200×15mm rectangular block was used as a reference table with the Workpiece Coordinate System (WCS) located at one corner of the table (the WCS is shown as the coordinate frame in Figure 6.1b). During the assembly, the component was mounted on the reference table with a clamping system. The CAD models were kept floating (the WCS is freely located in the Global Coordinate System (GCS) of a digitizing machine) to simulate the difference between the two coordinate frames. The use of the two different coordinate systems is for simulating a common problem of misalignments between the real WCS and its ideal location in digitizing station, or when the digitizing machine is a portable device.

The CAD model (Figure 6.1b) was then digitized to acquire point cloud data by converting the model to polygon surface (STereoLithography format) with the maximum allowable position error of 0.025mm. As a result, a point cloud dataset with 417,098 triangular facets and 208,167 vertices is obtained as illustrated in Figure 6.2a. Figure 6.2b shows the point cloud after transforming to WCS, where the reference table is shown as a wireframe.



a) Design model without simulated weld beads **b) Model with simulated defects**

Figure 6.1 CAD model in Solidworks assembly environment.



a) Point cloud generated from CAD model **b) Point cloud after transforming to WCS**

Figure 6.2 Point cloud data generated form CAD model. a) Data from STL file, b) Data after transforming to WCS

Subsequently, the point cloud data was resampled with a spatial resolution of 0.1mm by projecting the data in the Z-direction of WCS to obtain a depth image with 2,551×3,051 pixels (Figure 6.3). Afterwards, randomized errors with zero mean (0 mm) and standard deviation of 0.01mm (the errors are bounded in a closed interval [-

0.0521, 0.0524]) were appended to the depth image to emulate measurement errors in the digitizing step.

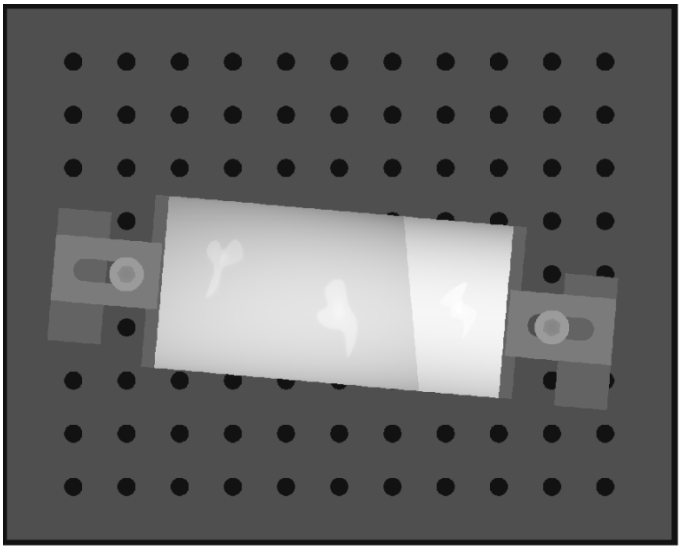
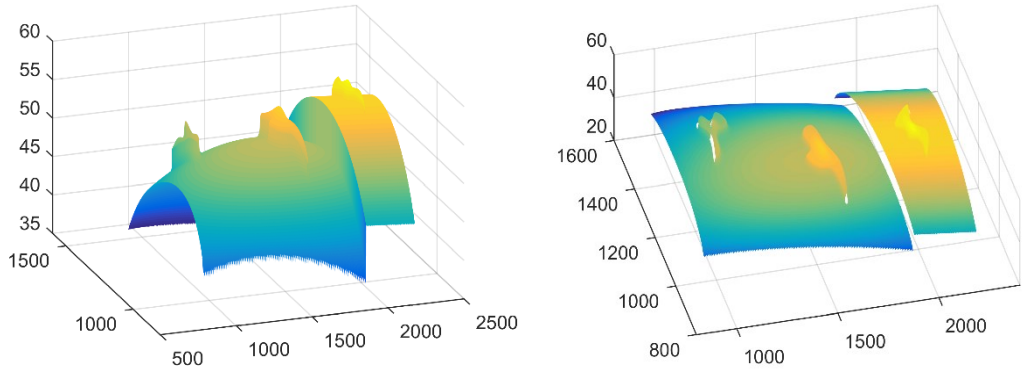


Figure 6.3 Depth image

6.1.2 Data pre-processing

As illustrated in Figure 6.3, the depth image comprises height data (z values) of the component and all support structures. As the nominal surface estimation process requires input data from the component only, this step will isolate and select the data of the component and eliminate data from other components such as the reference table and the clamping system. In addition, in this step, the workpiece surfaces will be segmented into smooth patches.



a) After the removal of the reference table and clamping system

b) After segmentation of the image into multiple smooth patches

Figure 6.4 Depth image after masking and segmentation

Because the component is fixed to the reference table, we can easily observe that the height data of the component must only contain positive values. Firstly, all the pixels of the depth image, which have their values smaller than a certain threshold (e.g. 0.1mm), are considered as a part of the support structure. Therefore, they will be eliminated from the depth image. Secondly, a binary mask indicating the location of clamping parts was applied to the image to retain the component data. Figure 6.4a presents the image after data selection. It is observed that the depth data now only contains information of the component.

Any edge detection method can be applied to trace slope changes between different surface patches. The identified edges are the boundaries of the patches, which are used to segment the component data into smooth surface patches. In this work, Sobel edge detection is applied to the image to find any possible edges in the image. Subsequently, the pixels around the identified edges were eliminated to create blank spaces within the smooth patches. Figure 6.4b presents the image after segmentation. It can be seen that the image is separated into two smooth patches with the defects still on the surface patches. As there is a smooth fillet between the defects and the surface patch, the edge detection algorithm can only partially identify the boundary between the weld beads and the component surfaces.

After segmenting the image into smooth surface patches, each surface patch needs to be categorized based on its size for further processing. First, the depth image was converted to a binary image, and the surface patches were identified by tracing objects in the binary image via Moore-Neighbor tracing method [145]. Next, the area of each surface patch was approximated based on the number of points in the patch (number of pixels) to sort the patches based on their size. Finally, all surface patches, which are bigger than a certain threshold (10% of the total area[6]), were selected as the candidates of smooth surface patches.

6.1.3 Nominal surface estimation

As shown in Figure 6.4b, each surface patch comprises both nominal surface and defects data. The reconstruction of the nominal surface requires a proper data selection to perform the estimation work (as discussed in Chapter 4). There are two steps in the

[6] The threshold of the surface area is set based on the information of the freeform surface patches which need to be machined in the component.

procedure for the nominal surface reconstruction, i.e. nominal data selection and nominal surface estimation.

Nominal data selection: The method for nominal curve estimation presented in Chapter 4 works well for most cases. However, it sometimes results in false estimation due to mistakes in the data selection. To ensure a proper data selection with minimum human intervention, the selection is carried out using the following procedure:

i. Independently apply the nominal curve estimation procedure (Chapter 4) for all curves taking from all rows and columns of the surface patch to create two estimated nominal surfaces. The first estimated nominal surface is based on the row data and the second is based on the column data.

ii. Evaluate the errors between the two estimated nominal surfaces and the surface patch.

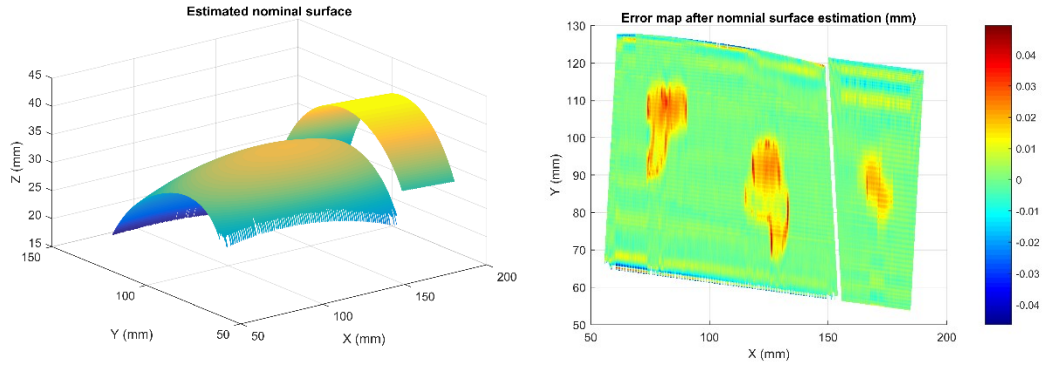
iii. The nominal data is obtained by selecting any points (in surface patch data) that the deviation from its corresponding estimated nominal surfaces is less than a certain threshold.

Nominal surface estimation: the nominal surface was obtained by employing the B-spline surface fitting technique from the selected data. However, for most of the cases, the selected data contains too many points which cause insufficient memory problem in the fitting process. To overcome this, the final nominal surface can be approximated based on the collections of B-spline curve fitting from each row and column of the selected data. This will result in two estimated nominal surfaces. To compute the final estimated surface, each point in the final surface is defined as the average of the two corresponding points from the two estimated nominal surfaces.

Figure 6.5a shows the estimated nominal surface after performing the estimation process. The errors of the estimated nominal surfaces are depicted in the right panel of the figure. The estimation error was computed by comparing the estimated surfaces with its ground truth, which was generated by converting the original nominal surface in Figure 6.1a to a depth image with the same resolution.

As shown in Figure 6.5, the errors of the estimation process are within ±0.05mm. At the defect locations, the estimated results tend to be higher than the nominal one. This might be attributed to the effects of the data selection process. Please note that the data

selection process is automatically performed by the program itself with no human intervention. We will later discuss it in subsection 6.1.6.



a) Estimated nominal surface **b) Error map from ground design information**

Figure 6.5 Estimated nominal surface and estimation error.

6.1.4 Tool path generation and simulation

Defect on the component surface can be located straightforwardly from the identified nominal surfaces. By subtracting the depth image of the measurement data to the estimated nominal surface image, the defect locations are ascertained subjected to two conditions, namely: *(i)* the maximum deviation is larger than a certain threshold (referred to as the error control, usually the same as the bounded error of B-spline fitting), *(ii)* the area of the defect is bigger than a certain value, e.g. 1mm^2. Figure 6.6 shows the defects, which were found by using the estimation program.

We can see that all the defects are properly identified. The boundaries and the defects are computed by using the Moore-Neighbor tracing method. The defects are then plotted for further confirmation by the user before it goes to tool path generation as shown in Figure 6.6.

The tool path for the confirmed defect is locally obtained through the following steps:

i. Select the local estimated nominal data around the defect.

ii. Use the B-spline surface to fit the selected local nominal surface for finer resampling.

iii. Offset the nominal surface according to the number of cutting layers, where each cutting layer is defined by the depth of cut of finishing and roughing cuts.

iv. Find the intersections of each offset surface, which has been obtained in the previous step, with the measured surface to find the cutting locations on each layer of cut.

v. Generate the tool path for each intersection layer.

vi. Generate the G-code based on the generated tool paths.

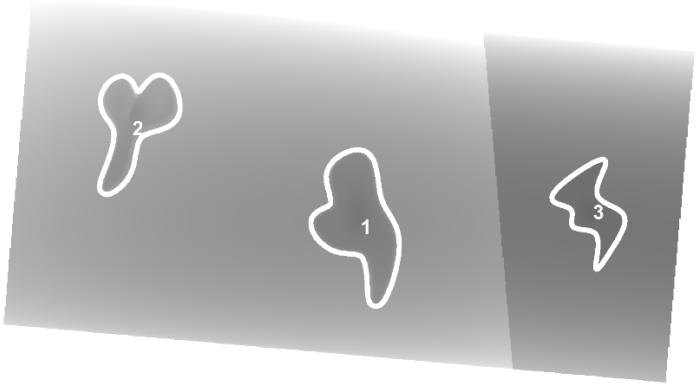


Figure 6.6 Identified defects

Figure 6.7 depicts the tool paths generated for the given data. The tool paths were generated by considering a 10mm diameter ball nose milling tool with a scallop height of 0.05mm for rough cutting and 0.001mm for finish cutting. The depth of cut for finish cutting is 0.1mm and 0.5mm for rough cutting. Between each layer, the cutting tool is lifted to a safe level and a new layer starts by moving down the tool in the Z directions.

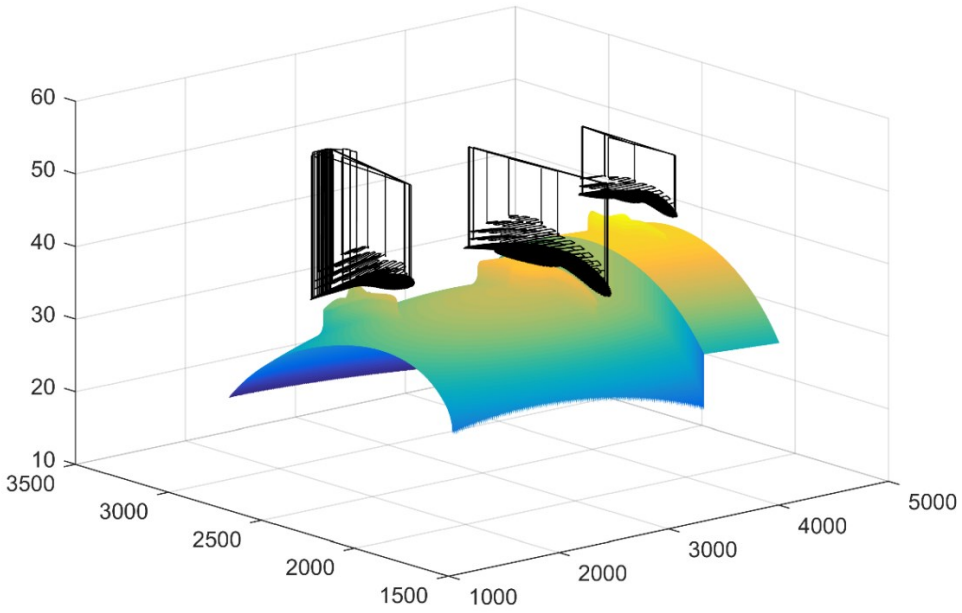


Figure 6.7 Local tool paths for defect removal

6.1.5 Validation

We can assess the accuracy of the proposed method after machining of real components by computing the differences between the simulated finishing surface and the nominal one. The finished surfaces were obtained by simulating the cutting processes when the cutter moves along the tool paths on the measured depth image. The simulation results also provide us the information about cutting violations and collisions. This is necessary to identify the sources of error in order to prevent the cutter from coming into contact with anything but the workpiece.

Figure 6.8 presents the errors of the finished surfaces from the ground truth. As the machining processes only work locally at the three defects, the machining errors only appear at the locations of the defects as shown in Figure 6.8a. Panel b and c show two selected finished surfaces at specific X and Y coordinates, while the next two panels illustrate the errors in the selected profiles. The maximum error is about 0.03mm.

6.1.6 Discussion

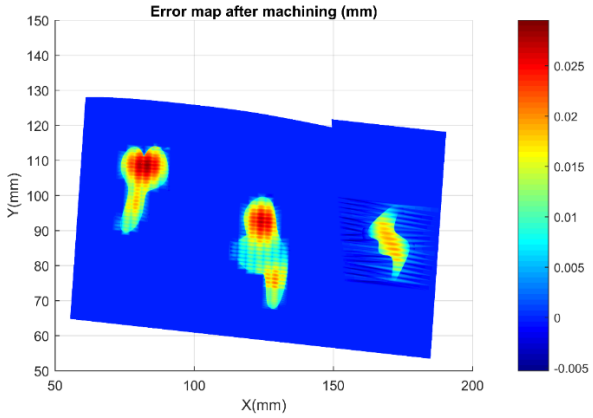
The results of the simulated data suggest that the proposed method is successfully process the given data to achieve the finished component with the maximum error of 0.03mm. It also can be seen that the errors in finishing surfaces exhibit some bulges at

the defects locations as illustrated in Figure 6.8d, e. The rational of the result is explained as follows:

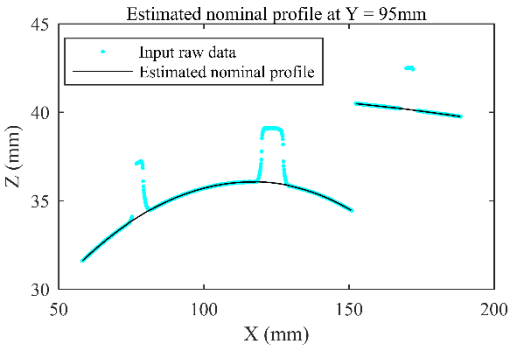
When designing the model in Solidworks environment, the defects and the nominal surfaces were smoothly connected by fillet surfaces. That makes the data near the defect location slightly higher than the nominal one. As the data selection process is performed based on a pre-determined error bound, the data around the defects, which are affected by the fillet process, are also selected because their deviations from the nominal ones are still within the error bound. Such selected data causes bulges at the defect location because the estimated surface follows the input data in the fitting processes.

The proposed method can help to reduce the need for skilled workers in almost all processing steps. However, there are a few parameters that need to be determined by the operator.

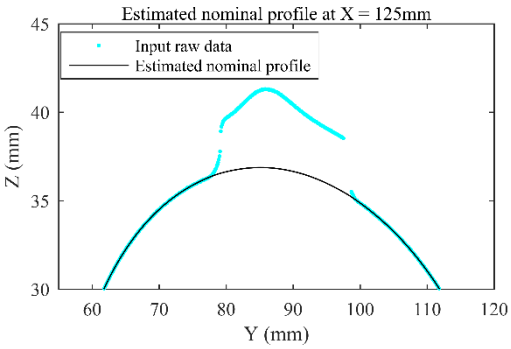
- The first parameter is the error bound for B-spline curve fitting and nominal data selection. This parameter must be determined based on the error bound in the digitizing stage. If it is too small, the B-spline fitting tends to overfit the results in small spline pieces that will cause incorrect data selection. If it is too large, we cannot distinguish the defect data from the nominal data.



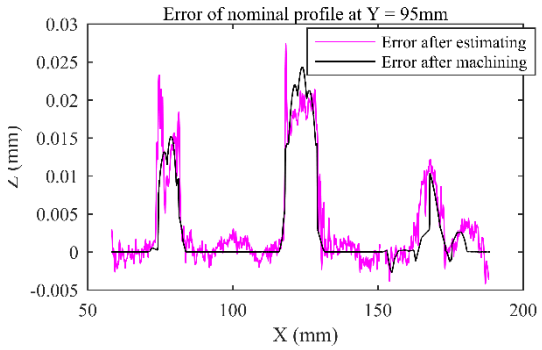
a) Error map of finished surfaces



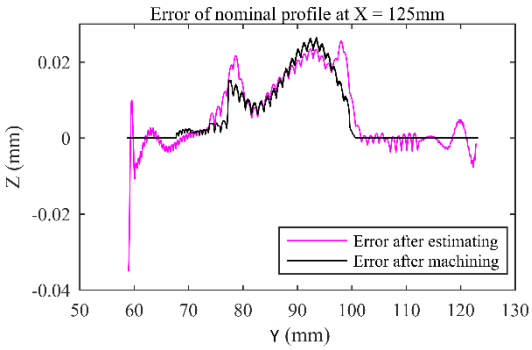
b) Selected nominal profile at Y = 95mm



c) Selected nominal profile at X = 125 mm



d) Machining error at Y = 95mm



e) Machining error at X = 125mm

Figure 6.8 Machining error from the ground truth nominal surfaces.

- The second parameter that needs to be considered is the smoothness term, λ, in the estimation of the nominal curves and surfaces. A very small smoothness value will result in wavy in the fitted curve and a very large value will cause an underfitted curve. It will lead to wrong nominal surfaces and defects estimation.

- Cutting parameters such as cutter types, cutter diameter, depth of cut, feed rate, maximum scallop height must be defined by the user.

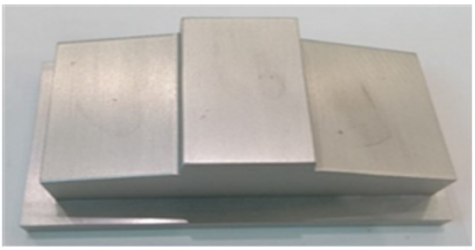
6.2 Experimental study

In the second test, data from real test components are used to validate the proposed method. This section details the second test. Some limitations of the proposed method are also discussed in this section.

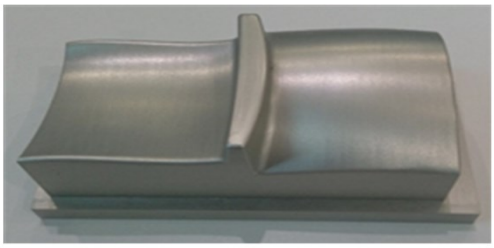
6.2.1 Test component design, data collection and data processing

Two real components representing flat and freeform surfaces were designed to evaluate the performance of the proposed framework as shown in Figures 6.9a and b. A rectangular block of 135×75×10 mm^3 was used as the reference table. As in the proposed framework, the workpiece should be fixed on the reference table in the re-profiling processes. However, in this experiment, the reference table and the component are merged together into a single workpiece to simplify the machining setup. This change of the component design should not have any effects on the use of the proposed framework. After the fabrication, the components were digitized with the GOM ATOS ScanBox machine to acquire the ground truth nominal surfaces as shown in Figures 6.9 c and d.

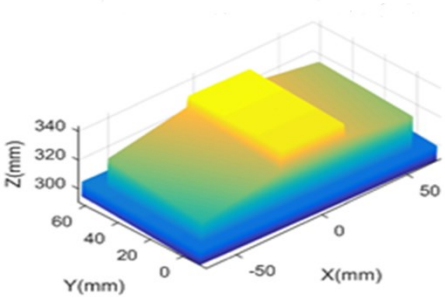
A few weld beads were added on the component surfaces to simulate positive defects. As illustrated in Figures 6.9e and f, the flat workpiece has three weld beads, and the freeform surface has five. Subsequently, digitized models of the components were obtained using the same digitizing machine. Those digitized datasets are used as raw input data for estimating the nominal surfaces and the defects. Please note that the measured geometries of the clean component, before adding up the weld beads, are used only as a benchmark to the proposed method. Those data are not used in any steps of the proposed framework. After obtaining the digitized geometries of the component, the entire data processes as described in Chapter 5 were applied to generate the G-code for defect removal. The machine setup and experimental results are detailed in next section.



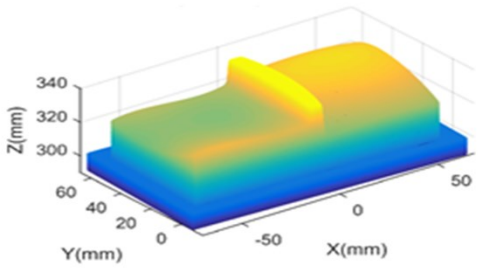
a) Flat component



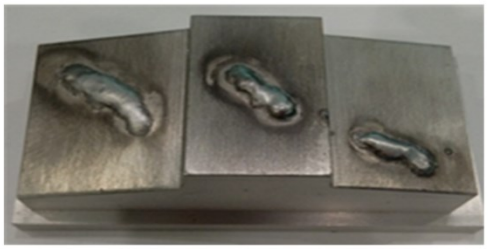
b) Freeform component



c) Digitized nominal geometry of flat component

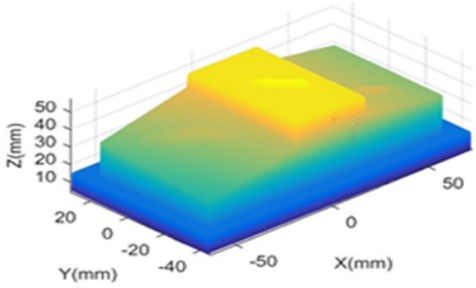


d) Digitized nominal geometry of freeform component

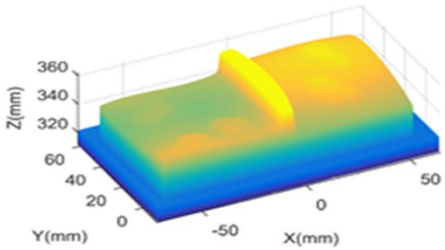


e) Flat component with weld beads

f) Freeform component with weld beads



g) Digitized geometry of flat component



h) Digitized geometry of freeform component

Figure 6.9 Test components and digitized geometries

6.2.2 Results and discussion

Machining set up:

After the digitized geometries have been put through the proposed method, the component and tool paths are sent to a machining center for defect removal. The machining center used is the "DMG-MORI DuoBlock" CNC milling machine.

Because the CNC machine has 5-axis while the tool paths are only generated for 3-axis machine, in order to have a fair assessment, two rotation axes of the machine are disabled during the machining. In the machining operations, a ball nose cutter with diameter 10mm is used for both roughing and finishing.

Figure 6.10 shows an excerpt of the G-code program for the CNC milling machine. The CNC machine uses SIEMENS format. There are three sections in a typical G-code program. The first section contains the header to configure the machine such as a code for defining unit of length, type of coordinate system, type of movement commands (relative or absolute), the cutter, high-speed interpolation, etc. The header code depends on the machines. Therefore, modifications might be required before using the G-code for real machining work. The second section of the G-code program is the tool path, which contains motion commands to control the cutter to follow the designated paths. In machining of freeform surfaces, the motion commands use small linear movements (G1). Therefore, high-speed machining needs to be used to keep a constant feed rate during the processes. The last section of the G-code program contains a footer, which commands the machine to stop and to move the component to the unload position.

Results

Flat component

Figure 6.11 depicts the flat component machining results. The automatic nominal surface estimation results in a comparable outcome to the simulated case. This means that at the defect locations, the estimated nominal surfaces are slightly higher than the ground truth ones because of the bulges issue. It can be seen in Figure 6.11c that the error in nominal surface estimation depends on the defect size. The bigger the defects, the higher the error in the estimated nominal surfaces. The maximum error in the estimation is about 0.06mm.

Figure 6.11d presents the error maps of the final machined surfaces and the ground truth surface of the component (original surface before appending the weld beads). We can see that the finished surfaces have been overcut in the right-hand-side defects for a distance about 0.05mm (cyan regions in Figure 6.11d). There are two main reasons for the overcutting. The first is the thermal expansion. The temperature of the spindle and the cutter increases during the cutting process making the coordinate system of the

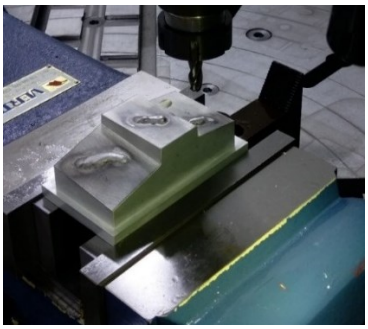
cutter slightly lower (in the Z direction) than the calibrated value at cold temperature. The second possibility is attributed to the misalignment problem when mounting the component.

```
; HEADER
; MACHINE CONFIGURATION
N10 DEF REAL _camtolerance
N20 DEF REAL _F_CUTTING, _F_ENGAGE, _F_RETRACT
N30 G40 G17 G710 G90
N40 DM_MILL
N50 CYCLE800()
N60 TRAFOOF
N70 TRANS
N80 M56
N90 L_FREI
N100 L_ZYM91
N120 ;GROUP: TABLE
N130 ;Operation : FIXED_CONTOUR
N140 _camtolerance=.02
N160 ;TOOL TYPE : Milling Tool-5 Parameters
N170 ;TOOL DIAMETER     : 10.000000
N180 ;TOOL LENGTH       : 60.000000
N190 ;TOOL CORNER RADIUS: 2.000000
N200 T="BN10_MM" ; 10mm BALL NOSE CUTTER
N210 M6
N220 G54
N230 D1
N240 T0
N250 DM_MILL
N260 ;Initial Move
N270 G94 S6000 M3
N280 CYCLE800(1,"TC1",200000,39,0,0,0,0,0,0,0,0,1,1,0)
N290 CYCLE800(1,"0",200000,57,0,0,0,0,0,0,0,0,1,1,0)
N300 TRAORI
N310 G54
CYCLE832(0.015,_ROUGH,1)   ; HIGH SPEED MACHINING
;============================================================
; TOOL PATH
N320 G00 X104.250 Y52.200 Z50.010
N330 G01 Z31.445 F400
N001015 G01 X104.250 Y52.200 Z31.445
N001020 G01 X104.350 Y52.200 Z31.452
N001025 G01 X104.450 Y52.200 Z31.459
N001030 G01 X104.550 Y52.200 Z31.466
.
.
N097500 G01 X107.950 Y66.650 Z34.690
N097505 G00 X107.950 Y66.650 Z50.010
; FOOTER
N097510 L_FREI
N097515 M5 M9
N097520; End of Program
N097525 M30
```

Figure 6.10 Gcode for CNC milling machine

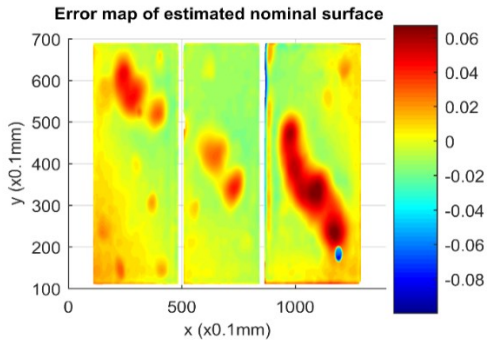
In the case of flat surfaces, we can achieve better nominal surfaces by fitting the surface with planes. However, in this example, we assume that we do not know the

target surfaces except the smoothness assumption. Because the outputs of the estimation method depend on the smoothness value and the defect size, the estimated nominal surfaces and the defects must be reconfirmed by the operator.

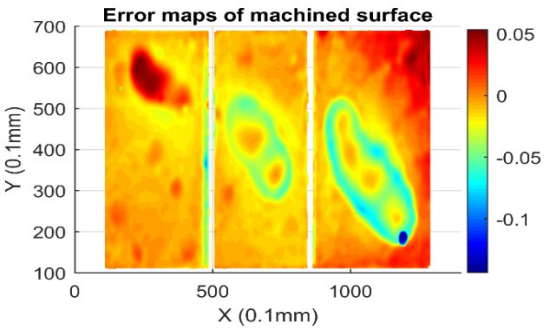


a) Flat component in machining process

b) Machined flat component



c) Error map between estimated nominal surfaces and the ground truth one



d) Error maps between machined surfaces and the ground truth nominal surface

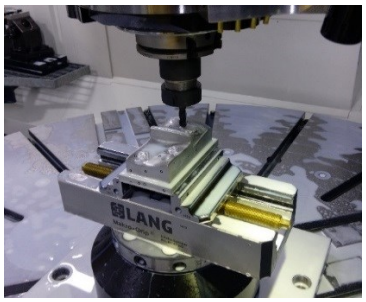
Figure 6.11 Experimental results of the flat component case

Freeform component

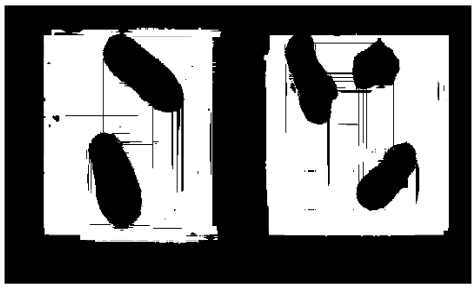
In the case of the freeform component, we conducted two different tests. In the first test, we added five small weld beads on the surfaces as shown in Figure 6.12a to evaluate the performance of the proposed method in the case of (relatively) small defects. In the second test, bigger defects were added at the same locations as in the first test. Figure 6.9f presents the defective component with relatively big defects.

From the first test, Figure 6.12e and f shows experimental results when the defects are relatively small. As shown in Figure 6.12c the nominal data is correctly selected by the proposed algorithm. In general, the proposed framework offers consistent results on the two previous cases with maximum estimation error of around 0.07mm.

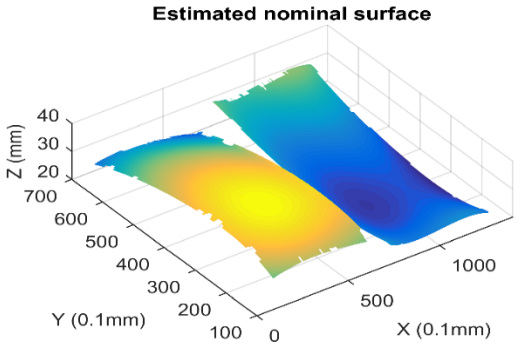
a) Freeform component with small weld beads



b) Freeform component in machining process

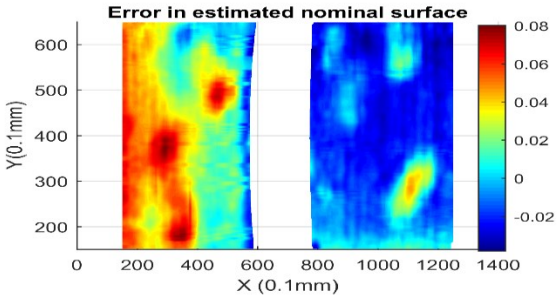


c) Seleted data for nominal surface estimation



d) Estimated nominal surfaces

e) Freeform component after maching

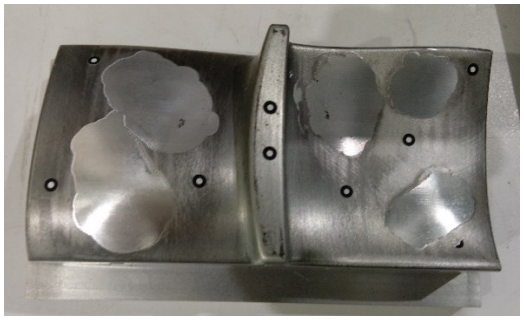


f) Error map between estimated nominal surface and the ground truth one

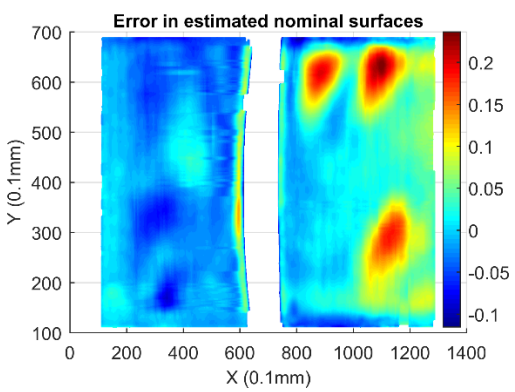
Figure 6.12 Experimental results of freeform surface component. Case 1: small defect size

In the second case, as shown in Figure 6.13a, the nominal data is also properly selected, but there are some parts that the algorithm cannot properly select. In comparison to the small defect size case (Figure 6.12c), the number of points used as the basis for nominal surface estimation is smaller. Consequently, the error of the estimated nominal surfaces is very high from -0.1 to 0.25mm and the sign of the error is not consistent, with the negative errors on the left-hand-side surface patch (Figure 6.13c), and positive errors on the right-hand-side surface patch respectively

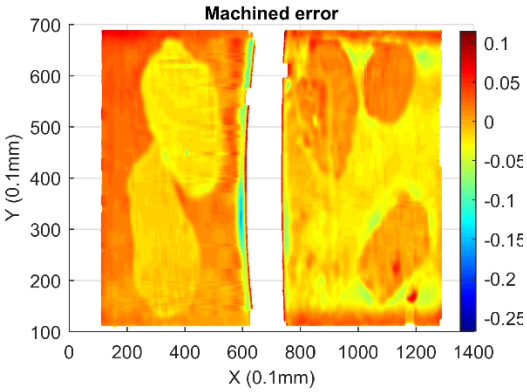
a) Selected data for nominal surface estimation



b) Freeform component after machining



c) Error between estimated nominal surfaces and the ground truth one



d) Error between machined surfaces and estimated nominal surfaces

Figure 6.13 Experimental results of freeform component. Case 2: big defect size

6.3 Conclusion

This chapter discusses the results of the proposed framework for two different scenarios, ranging from the simulation data to the real measured data. In the simulation case, the data was designed to capture a real situation of re-profiling in remanufacturing. In the case of actual data, to avoid potential problems in transferring the component in different machining stages, the component and the mounting system are fabricated as a single component. The experimental results suggest that the proposed method can automatically process the data to estimate the nominal surfaces and identify defects on the surfaces.

However, the result from experimental data shows that there is a strong correlation between the size of the defect and the accuracy in the absence of design information. It is impossible to obtain the exact ground truth nominal surfaces from the measured data of damaged components, even when the data is accurately selected. For such applications, where design geometries are not available, it is suggested that the

implementation of the algorithm is limited to a case where the size of the defects (damaged regions on the surfaces) is small.

Chapter 7: Conclusion and Future works

7.1 General conclusion

This thesis deals with a framework for automatic reprofiling of freeform surface components in remanufacturing industry in the absence of design information. As geometrical deviations might occur during a component service life, the nominal geometry might not correspond to the designed geometry. In this case, an adaptive machining approach is used to estimate and reconstruct the nominal geometry of the component. Digitization of the geometry of the component is automatically processed to estimate its nominal surfaces and to identify positive defects (e.g. weld beads) on the surfaces using a B-spline fitting technique. The estimated nominal geometries and the defect information are subsequently used to generate tool paths for milling to produce the finished workpiece surfaces.

Experiments on simulated data and real test component data suggested that the proposed framework is a potential tool for automatic re-profiling works. It can help to simplify the process and reduce the dependency on skilled workers and help to expand remanufacturing industries. The development of the proposed framework also contributes in a few aspects as summarized below.

B-spline curve fitting. In Chapter 3, we presented a novel two-step algorithm for B-spline curve fitting. In the first step, a bisecting method is used to subdivide input data into a set of single-piece splines (i.e. the data in the single-piece spline is approximated by a member function of the B-spline within a prescribed allowable error). Two different approaches, namely serial and parallel bisection, are discussed. The parallel bisecting approach offers a faster computational process and potential implementation on parallel computing machines. In the second step, optimal knots (connection point between two neighboring spline member functions) position and smoothness level of the curves at the knots are obtained by solving a nonlinear optimization problem using a Gauss-Newton method for each connected pair of member functions. After the knot vector is determined, control points of the B-spline are obtained by solving a traditional linear least square problem.

Experimental results showed that the proposed method can exactly recover B-spline functions from sampled data. Moreover, the proposed method is able to deal with

different smoothness levels at the knots (knot multiplicity), that makes it suitable for approximating or representing any kind of curves from smooth to non-trivial with discontinuous points, kink points, or lower smoothness points. In comparison with the available methods in literature, the proposed method offers smaller fitted errors and lesser computational time, making it suitable for reverse engineering applications.

Algorithm for automatic nominal curve/surface estimation from measured data without design information. Chapter 4 presents a method to estimate nominal profiles of open curves when the measured data contains some defects in the absence of design geometries. Conventional approaches in reconstructing the nominal curve profiles in these cases follow two steps. First, defective parts in the data are manually eliminated to retain the nominal part. Second, a hole filling technique is applied to estimate nominal geometries at the defect locations for further processes. In this chapter, we introduced a method that can automatically analyze the data to select the nominal data set and subsequently estimate the nominal curve from the measured data. The B-spline fitting technique is used to segment the data into small pieces. Subsequently, nominal data is selected based on its size and curvature. A dedicated combination process is used to properly select the segmented data that belongs to the nominal surface. The nominal curve profile is reconstructed by using the B-spline fitting technique.

Experimental results demonstrate the potential of the proposed method for automatic data processing in nominal surface estimations and defect detections. However, the accuracy of the estimated nominal geometries depends on a few factors:

i) The first factor is the ground truth geometries. As the estimation problems have no unique solutions, any solutions give an approximation of the true solution. The estimated nominal surfaces rely on a specific cost function; therefore, the accuracy of the estimation varies for different surfaces.

ii) The second factor is the size ratio of the defective portion to the nominal surface portion in the data. The smaller the defect ratio, the better the accuracy of the results.

Framework for automatic reprofiling in remanufacturing processes. In Chapter 5, we presented a data processing framework for automatic re-profiling of freeform surface components in the absence of design geometries in remanufacturing processes. Data interfacing between digitization stage and the machining stage was discussed. In

the framework, measurement data from the digitization stage, which is in the form of polygon surfaces, is first processed to identify the workpiece coordinate system (WCS). Subsequently, the measured data is transformed to the WCS to keep the same reference coordinate system for machining step. Afterwards, the data is resampled to convert to a depth image format for the nominal surface estimation and defect detection. After confirming the defects, local tool paths are generated to obtain the workpiece surfaces from the estimated nominal surface.

Some experiments were carried out to evaluate the proposed framework. The performance of the proposed method in both simulation and real data suggest that the method can automatically perform the re-profiling process with a simple set up from an operator. However, some constraints in the process might lead to some limitations in the results. The limitations include the following: *(i)* The reference table was fabricated together with the test component as a single component, however errors in the machining of the table contribute to the alignment errors with a magnitude of around 0.03mm. *(ii*) Digitization error, the digitizing machine has accuracy about 0.025mm. *(iii)* Thermal expansion of CNC machine during machining processes. *(iv)* Machining setup error, misalignment of the components on a mounting jig. The limitations that cause low accuracy in the experimental results are discussed in Chapter 6.

7.2 Recommendations for Future work

B-spline curves are very flexible as they can be used to present and handle any curve shapes from highly smooth curve to curves with discontinuity. B-spline curves are widely used to represent curves in various industries. However, the applications of B-spline curves in fitting a B-spline function from measured data in reverse engineering are currently limited to smooth cases due to some constraints of B-spline fitting techniques for curves with non-trivial points such as kinks, cusp, kinks in first derivative etc. The proposal of the new fitting method in this thesis will assist to extend the capabilities of B-spline curves in the reverse engineering. The proposed approach might contribute to expanding the uses of the curves in the near future such as planning tool paths for CNC machines and industrial robots.

B-spline curves are a specific case of Non-Uniform Rational B-spline (NURBS) curves. Fitting curves with different continuities in the curve using B-spline technique

(with knot multiplicity) is well addressed in this thesis. However, there is a need to develop a technique for general NURBS curves that can handle non-trivial continuity points.

The automatic nominal surface estimation method for freeform surface workpieces in this thesis works well when defects are small and located in the surfaces. However, it is constrained by some limitations that make the method not well suitable for detecting defects located at edges or tips of the components. Further study on automatic identification of the defects needs to be carried out.

In the absence of the design information, the estimated nominal geometry of a component heavily depends on various parameters such as the error bound value, the minimum of radius of curvature, the smoothness value etc. The parameters are component dependency. Artificial intelligence based on machine learning such as Deep Neural Network, Hidden Markov Models, Support Vector Machine, etc. can be used to develop a model for classifying freeform surfaces into different classes (based on their features such as radii of curvature, surface energy, defect size ratio etc.). The optimal values of the parameters of interest need to be set based on the freeform surface classes.

Machining of freeform surfaces usually uses three or five degrees of freedom of cutters. This thesis used three axes approach for tool path generation. When there are obstacles on the machining surfaces, five axis approach needs to be employed. There is a need for developing an automatic five-axis tool paths generation technique for re-profiling processes.

As adaptive machining relies on the measured data to perform any further work, any error in digitization will propagate to the subsequent processes. It is almost impossible to satisfy both accuracy and speed in the digitization processes. In most of cases, the error in the measured data is quite high. The question arises as to how to obtain the highest machining accuracy in the adaptive machining. Further investigation on some factors that affect machining results, such as geometry alignment methods, heat expansion of machines, cutters and workpieces, tool chatter, etc. to compensate for the effects of these factors in machining results need to be elaborated.

List of Publications raised from this research work

International Journals

1. V. T. Dung, Tjahjowidodo T (2017) A direct method to solve optimal knots of B-spline curves: An application for non-uniform B-spline curves fitting. PLoS ONE 12(3): e0173857. https://doi.org/10.1371/journal.pone.0173857

2. V. T. Dung, T. Tjahjowidodo and F. Cheng, "Automatic weld beads removing on free form surface parts: A novel approach in remanufacturing processes", in preparation to be submitted.

International Conference Papers

1. T. Tjahjowidodo, V. T. Dung and M. Han, "A fast non-uniform knots placement method for B-spline fitting," *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Busan, 2015, pp. 1490-1495.

2. V. T. Dung, T. Tjahjowidodo and M. Han, "Automatic defect detection and the estimation of nominal profiles based on spline for freeform surface parts," *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Busan, 2015, pp. 1484-1489.

3. V. T. Dung, T. Tjahjowidodo, "Automatic weld bead detection on freeform surface parts for remanufacturing processes," *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Munich, 2017, pp. 1065-1070.

References

[1] J. W. Sutherland, D. P. Adler, K. R. Haapala, and V. Kumar, "A comparison of manufacturing and remanufacturing energy intensities with application to diesel engine production," *CIRP Annals-Manufacturing Technology,* vol. 57, pp. 5-8, 2008.

[2] R. T. Lund and W. M. Hauser. (2010, Remanufacturing - an American perspective. *IET Conference Proceedings,* 1-6. Available: http://digital-library.theiet.org/content/conferences/10.1049/cp.2010.0404

[3] W. Kerr and C. Ryan, "Eco-efficiency gains from remanufacturing: A case study of photocopier remanufacturing at Fuji Xerox Australia," *Journal of Cleaner Production,* vol. 9, pp. 75-81, 2// 2001.

[4] C.-M. Lee, W.-S. Woo, and Y.-H. Roh, "Remanufacturing: Trends and issues," *International Journal of Precision Engineering and Manufacturing - Green Technology,* vol. 4, pp. 113-125, 2017.

[5] C. Chen, Y. Wang, H. Ou, Y. He, and X. Tang, "A review on remanufacture of dies and moulds," *Journal of Cleaner Production,* vol. 64, pp. 13-23, 2/1/ 2014.

[6] N. Nasr and M. Thurston, "Remanufacturing: A key enabler to sustainable product systems," *Rochester Institute of Technology,* p. 23, 2006.

[7] E. Bagci, "Reverse engineering applications for recovery of broken or worn parts and re-manufacturing: Three case studies," *Advances in Engineering Software,* vol. 40, pp. 407-418, 6// 2009.

[8] G. Ferrer, "The economics of tire remanufacturing," *Resources, Conservation and Recycling,* vol. 19, pp. 221-255, 1997/04/01 1997.

[9] S. S. Yang, S. K. Ong, and A. Y. C. Nee, "Product Design for Remanufacturing," in *Handbook of Manufacturing Engineering and Technology*, A. Y. C. Nee, Ed., ed London: Springer London, 2015, pp. 3195-3217.

[10] W.-w. Liu, M.-z. Li, T. Short, X.-c. Qing, Y.-m. He, Y.-z. Li, *et al.*, "Supercritical carbon dioxide cleaning of metal parts for remanufacturing industry," *Journal of Cleaner Production,* vol. 93, pp. 339-346, 4/15/ 2015.

[11] J. Gao, X. Chen, D. Zheng, O. Yilmaz, and N. Gindy, "Adaptive restoration of complex geometry parts through reverse engineering application," *Advances in Engineering Software,* vol. 37, pp. 592-600, 9// 2006.

[12] J. Gao, X. Chen, O. Yilmaz, and N. Gindy, "An integrated adaptive repair solution for complex aerospace components through geometry reconstruction," *The International Journal of Advanced Manufacturing Technology,* vol. 36, pp. 1170-1179, 2008// 2008.

[13] J. M. Wilson, C. Piya, Y. C. Shin, F. Zhao, and K. Ramani, "Remanufacturing of turbine blades by laser direct deposition with its energy and environmental impact analysis," *Journal of Cleaner Production,* vol. 80, pp. 170-178, 10/1/ 2014.

[14] Y. Zhang, Z. Yang, G. He, Y. Qin, and H.-c. Zhang, "Remanufacturing-Oriented Geometric Modelling for the Damaged Region of Components," *Procedia CIRP,* vol. 29, pp. 798-803, 2015/01/01 2015.

[15] J. H. Graham, J. Reinwand, R. G. Coriell, and K. Lytle, "Adaptive machining and weld repair process," ed: Google Patents, 2009.

[16] T. Melzer-Jokisch, D. Thomaidis, and R. Wilkenhöner, "Automated repair method and system," ed: Google Patents, 2015.

[17] J. F. Rhodes, T. P. Fuesting, J. P. HENDERKOTT, R. J. SNYDER, J. F. CLICK, and L. A. Junod, "Adaptively machining component surfaces and hole drilling," ed: Google Patents, 2016.

[18] C. Bremer, "Automated Repair and Overhaul of Aero-Engine and Industrial Gas Turbine Components," in *ASME Turbo Expo 2005: Power for Land, Sea, and Air*, Reno, Nevada, USA, 2005, pp. 841-846.

[19] M. P. Boyce, "22 - Case Histories," in *Gas Turbine Engineering Handbook (Fourth Edition)*, M. P. Boyce, Ed., ed Oxford: Butterworth-Heinemann, 2012, pp. 885-921.

[20] M. P. Boyce, "21 - Maintenance Techniques," in *Gas Turbine Engineering Handbook (Fourth Edition)*, M. P. Boyce, Ed., ed Oxford: Butterworth-Heinemann, 2012, pp. 803-883.

[21] T. J. Carter, "Common failures in gas turbine blades," *Engineering Failure Analysis,* vol. 12, pp. 237-247, 2005/04/01/ 2005.

[22] CNATT. *14008A Training manual*. Available: http://navybmr.com/study%20material/14008a/

[23] E. S. N. E. E. NORM, "Geometrical Product Specification (GPS) — Surface imperfections — Terms, definitions and parameters," vol. EN ISO 8785, ed, 1999.

[24] J. B. Jones, P. McNutt, R. Tosi, C. Perry, and D. I. Wimpenny, "Remanufacture of turbine blades by laser cladding, machining and in-process scanning in a single machine," presented at the 23rd Annual International Solid Freeform Fabrication Symposium, Austin, TX, USA, 2012.

[25] Y. Li, C.-H. Lee, and J. Gao, "From computer-aided to intelligent machining: Recent advances in computer numerical control machining research," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* vol. 229, pp. 1087-1103, 2015.

[26] R. Katz, V. Srivatsan, and L. Patil, "Closed-loop machining cell for turbine blades," *The International Journal of Advanced Manufacturing Technology,* vol. 55, pp. 869-881, August 01 2011.

[27] V. Srivatsan, R. Katz, and D. Dutta, "Fixtureless Sensor Standoff Control for High-Precision Dimensional Inspection of Freeform Parts," *Journal of Manufacturing Science and Engineering,* vol. 129, pp. 172-179, 2006.

[28] L. Zhu, J. Barhak, V. Srivatsan, and R. Katz, "Efficient registration for precision inspection of free-form surfaces," *The International Journal of Advanced Manufacturing Technology,* vol. 32, pp. 505-515, March 01 2007.

[29] D. Whitney, A. Edsall, A. Todtenkopf, T. Kurfess, and A. Tate, "Development and control of an automated robotic weld bead grinding system," *Journal of Dynamic Systems, Measurement, and Control,* vol. 112, pp. 166-176, 1990.

[30] A. Srivastava, D. Rogers, and M. Elbestawi, "Optimal planning of an adaptively controlled robotic disk grinding process," *International Journal of Machine Tools and Manufacture,* vol. 33, pp. 809-825, 1993.

[31] A. Robertsson, T. Olsson, R. Johansson, A. Blomdell, K. Nilsson, M. Haage*, et al.*, "Implementation of Industrial Robot Force Control Case Study: High Power Stub Grinding and Deburring," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2743-2748.

[32] T. Olsson, M. Haage, H. Kihlman, R. Johansson, K. Nilsson, A. Robertsson*, et al.*, "Cost-efficient drilling using industrial robots with high-bandwidth force feedback," *Robotics and Computer-Integrated Manufacturing,* vol. 26, pp. 24-38, 2010/02/01/ 2010.

[33] F. Nagata, T. Hase, Z. Haga, M. Omoto, and K. Watanabe, "CAD/CAM-based position/force controller for a mold polishing robot," *Mechatronics,* vol. 17, pp. 207-216, 2007/05/01/ 2007.

[34] F. Nagata, Y. Kusumoto, Y. Fujimoto, and K. Watanabe, "Robotic sanding system for new designed furniture with free-formed surface," *Robotics and Computer-Integrated Manufacturing,* vol. 23, pp. 371-379, 2007/08/01/ 2007.

[35] H.-y. Tam, O. Chi-hang Lui, and A. C.K. Mok, "Robotic polishing of free-form surfaces using scanning paths," *Journal of Materials Processing Technology,* vol. 95, pp. 191-200, 1999/10/15/ 1999.

[36] C. Piya, J. M. Wilson, S. Murugappan, Y. Shin, and K. Ramani, "Virtual Repair: Geometric Reconstruction for Remanufacturing Gas Turbine Blades," pp. 895-904, 2011.

[37] J. Gao, J. Folkes, O. Yilmaz, and N. Gindy, "Investigation of a 3D non-contact measurement based blade repair integration system," *Aircraft Engineering and Aerospace Technology,* vol. 77, pp. 34-41, 2005.

[38] Y. Rong, J. Xu, and Y. Sun, "A surface reconstruction strategy based on deformable template for repairing damaged turbine blades," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* vol. 228, pp. 2358-2370, 2014.

[39] C. De Boor, "A practical guide to splines, revised Edition, Vol. 27 of Applied Mathematical Sciences," ed: Springer-Verlag, Berlin, 2001.

[40] L. Piegl and W. Tiller, *The NURBS book*: Springer Science & Business Media, 2012.

[41] W. Li, S. Xu, G. Zhao, and L. P. Goh, "Adaptive knot placement in B-spline curve approximation," *Computer-Aided Design,* vol. 37, pp. 791-797, 2005.

[42] O. Grove, K. Rajab, L. A. Piegl, and S. Lai-Yuen, "From CT to NURBS: contour fitting with B-spline curves," *Computer-Aided Design and Applications,* vol. 8, pp. 3-21, 2011.

[43] J. R. Rice, "Algorithm 525: ADAPT, adaptive smooth curve fitting [E2]," *ACM Transactions on Mathematical Software (TOMS),* vol. 4, pp. 82-94, 1978.

[44] J. R. Rice, "Adaptive approximation," *Approximation Theory,* vol. 16, pp. 239-337, 1976.

[45] K. Ichida, T. Kiyono, and F. Yoshimoto, "Curve fitting by a one-pass method with a piecewise cubic polynomial," *ACM Transactions on Mathematical Software (TOMS),* vol. 3, pp. 164-174, 1977.

[46] T. Tjahjowidodo, V. Dung, and M. Han, "A fast non-uniform knots placement method for B-spline fitting," in *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on*, 2015, pp. 1490-1495.

[47] M. Plass and M. Stone, "Curve-fitting with piecewise parametric cubics," in *ACM SIGGRAPH Computer Graphics*, 1983, pp. 229-239.

[48] H. Park and J.-H. Lee, "B-spline curve fitting based on adaptive curve refinement using dominant points," *Computer-Aided Design,* vol. 39, pp. 439-451, 2007.

[49] J. Wu, H. Zhou, X. Tang, and J. Chen, "Implementation of CL points preprocessing methodology with NURBS curve fitting technique for high-speed machining," *Computers & Industrial Engineering,* vol. 81, pp. 58-64, 2015.

[50] W. L. Chung, "Automatic curve fitting using an adaptive local algorithm," *ACM Transactions on Mathematical Software (TOMS),* vol. 6, pp. 45-57, 1980.

[51] T. Lyche and K. Mørken, "A data-reduction strategy for splines with applications to the approximation of functions and data," *IMA Journal of Numerical analysis,* vol. 8, pp. 185-208, 1988.

[52] T. Lyche and K. Mørken, "Knot removal for parametric B-spline curves and surfaces," *Computer Aided Geometric Design,* vol. 4, pp. 217-230, 1987.

[53] X. He, L. Shen, and Z. Shen, "A data-adaptive knot selection scheme for fitting splines," *Signal Processing Letters, IEEE,* vol. 8, pp. 137-139, 2001.

[54] H. Kang, F. Chen, Y. Li, J. Deng, and Z. Yang, "Knot calculation for spline fitting via sparse optimization," *Computer-Aided Design,* vol. 58, pp. 179-188, 2015.

[55] C. Conti, R. Morandi, C. Rabut, and A. Sestini, "Cubic spline data reduction choosing the knots from a third derivative criterion," *Numerical Algorithms,* vol. 28, pp. 45-61, 2001.

[56] Y. Yuan, N. Chen, and S. Zhou, "Adaptive B-spline knot selection using multi-resolution basis set," *IIE Transactions,* vol. 45, pp. 1263-1277, 2013.

[57] H. Schwetlick and T. Schütze, "Least squares approximation by splines with free knots," *BIT Numerical Mathematics,* vol. 35, pp. 361-384, 1995.

[58] M. Randrianarivony and G. Brunnett, "Approximation by NURBS curves with free knots," in *Proceedings of the Vision, Modeling, and Visualization Conference*, 2002, pp. 195-201.

[59] G. Beliakov, "Least squares splines with free knots: global optimization approach," *Applied mathematics and computation,* vol. 149, pp. 783-798, 2004.

[60] S. Miyata and X. Shen, "Free-knot splines and adaptive knot selection," *Journal of the Japan Statistical Society,* vol. 35, pp. 303-324, 2005.

[61] S. Miyata and X. Shen, "Adaptive free-knot splines," *Journal of Computational and Graphical Statistics,* vol. 12, pp. 197-213, 2003.

[62] X. Zhao, C. Zhang, B. Yang, and P. Li, "Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation," *Computer-Aided Design,* vol. 43, pp. 598-604, 2011.

[63] M. Sarfraz and S. A. Raza, "Capturing outline of fonts using genetic algorithm and splines," in *Information Visualisation, 2001. Proceedings. Fifth International Conference on*, 2001, pp. 738-743.

[64] A. Gálvez and A. Iglesias, "Efficient particle swarm optimization approach for data fitting with free knot B-splines," *Computer-Aided Design,* vol. 43, pp. 1683-1692, 2011.

[65] E. Ülker and A. Arslan, "Automatic knot adjustment using an artificial immune system for B-spline curve approximation," *Information Sciences,* vol. 179, pp. 1483-1494, 2009.

[66] G. Farin, "Chapter 1 - A History of Curves and Surfaces in CAGD," in *Handbook of Computer Aided Geometric Design*, ed Amsterdam: North-Holland, 2002, pp. 1-21.

[67] Y. Li and P. Gu, "Free-form surface inspection techniques state of the art review," *Computer-Aided Design,* vol. 36, pp. 1395-1417, 2004/11/01/ 2004.

[68] W. Gao and S. Kiyono, "High accuracy profile measurement of a machined surface by the combined method," *Measurement,* vol. 19, pp. 55-64, 1996/09/01/ 1996.

[69] J. Li, M. Chen, X. Jin, Y. Chen, Z. Dai, Z. Ou*, et al.*, "Calibration of a multiple axes 3-D laser scanning system consisting of robot, portable laser scanner and turntable," *Optik - International Journal for Light and Electron Optics,* vol. 122, pp. 324-329, 2011/02/01/ 2011.

[70] N. Audfray, C. Mehdi-Souzani, and C. Lartigue, "A Novel Approach for 3D Part Inspection Using Laser-plane Sensors," *Procedia CIRP,* vol. 10, pp. 23-29, 2013/01/01/ 2013.

[71] V. Niola, C. Rossi, and S. Savino, "A new real-time shape acquisition with a laser scanner: first test results," *Robotics and Computer-Integrated Manufacturing,* vol. 26, pp. 543-550, 2010/12/01/ 2010.

[72] N. Van Gestel, S. Cuypers, P. Bleys, and J.-P. Kruth, "A performance evaluation test for laser line scanners on CMMs," *Optics and Lasers in Engineering,* vol. 47, pp. 336-342, 2009/03/01/ 2009.

[73] S. Son, H. Park, and K. H. Lee, "Automated laser scanning system for reverse engineering and inspection," *International Journal of Machine Tools and Manufacture,* vol. 42, pp. 889-897, 2002/06/01/ 2002.

[74] F. J. Brosed, J. Santolaria, J. J. Aguilar, and D. Guillomía, "Laser triangulation sensor and six axes anthropomorphic robot manipulator modelling for the measurement of complex geometry products," *Robotics and Computer-Integrated Manufacturing,* vol. 28, pp. 660-671, 2012/12/01/ 2012.

[75] S. Yin, Y. Guo, Y. Ren, J. Zhu, S. Yang, and S. Ye, "Real-time thermal error compensation method for robotic visual inspection system," *The International Journal of Advanced Manufacturing Technology,* vol. 75, pp. 933-946, November 01 2014.

[76] F. Remondino, "From point cloud to surface: the modeling and visualisation problem," presented at the WG V/6 Visualization and Animation of Reality-based 3D Models Tarasp-Vulpera, Engadin, Switzerland, 2003.

[77] S. Yuwen, G. Dongming, J. Zhenyuan, and L. Weijun, "B-spline surface reconstruction and direct slicing from point clouds," *The International Journal of Advanced Manufacturing Technology,* vol. 27, pp. 918-924, February 01 2006.

[78] J. P. Kruth and A. Kerstens, "Reverse engineering modelling of free-form surfaces from point clouds subject to boundary conditions," *Journal of Materials Processing Technology,* vol. 76, pp. 120-127, 1998/04/01/ 1998.

[79] J. Peters, "11. Constructing C1 Surfaces of Arbitrary Topology Using Biquadratic and Bicubic Splines," in *Designing Fair Curves and Surfaces*, ed, pp. 277-293.

[80] M. Eck and H. Hoppe, "Automatic reconstruction of B-spline surfaces of arbitrary topological type," presented at the Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996.

[81] A. Dimitrov, R. Gu, and M. Golparvar-Fard, "Non-Uniform B-Spline Surface Fitting from Unordered 3D Point Clouds for As-Built Modeling," *Computer-Aided Civil and Infrastructure Engineering,* vol. 31, pp. 483-498, 2016.

[82] A. Dimitrov and M. Golparvar-Fard, "Segmentation of building point cloud models including detailed architectural/structural features and MEP systems," *Automation in Construction,* vol. 51, pp. 32-45, 2015/03/01/ 2015.

[83] Y. Liao, X. Weng, C. Swonger, and J. Ni, "Defect detection and classification of machined surfaces under multiple illuminant directions," in *SPIE Optical Engineering+ Applications*, 2010, pp. 77981T-77981T-16.

[84] H. Shen, S. Li, D. Gu, and H. Chang, "Bearing defect inspection based on machine vision," *Measurement,* vol. 45, pp. 719-733, 2012.

[85] Q. Li and S. Ren, "A visual detection system for rail surface defects," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, pp. 1531-1542, 2012.

[86] Y. Wang, Y. Sun, P. Lv, and H. Wang, "Detection of line weld defects based on multiple thresholds and support vector machine," *NDT & E International,* vol. 41, pp. 517-524, 2008.

[87] L. Zhang, W. Ke, Q. Ye, and J. Jiao, "A novel laser vision sensor for weld line detection on wall-climbing robot," *Optics & Laser Technology,* vol. 60, pp. 69-79, 2014.

[88] L. Zhang, J. Jiao, Q. Ye, Z. Han, and W. Yang, "Robust weld line detection with cross structured light and Hidden Markov Model," in *2012 IEEE International Conference on Mechatronics and Automation*, 2012, pp. 1411-1416.

[89] C. Bremer, "Automated repair and overhaul of aero-engine and industrial gas turbine components," in *ASME Turbo Expo 2005: Power for Land, Sea, and Air*, 2005, pp. 841-846.

[90] W. Tao, D. Huapeng, W. Hao, and T. Jie, "Virtual remanufacturing: Cross-section curve reconstruction for repairing a tip-defective blade," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* p. 0954406214567135, 2015.

[91] J. Li, F. Yao, Y. Liu, and Y. Wu, "Reconstruction of broken blade geometry model based on reverse engineering," in *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*, 2010, pp. 680-682.

[92] X. Liu, "Filling N-Sided Holes With Trimmed B-Spline Surfaces Based on Energy-Minimization Method," *Journal of Computing and Information Science in Engineering,* vol. 15, pp. 011001-011001-9, 2015.

[93] A. Kumar, A. Shih, Y. Ito, D. Ross, and B. Soni, "A hole-filling algorithm using non-uniform rational b-splines," in *Proceedings of the 16th International Meshing Roundtable*, 2008, pp. 169-182.

[94] E. Pérez, S. Salamanca, P. Merchán, and A. Adán, "A comparison of hole-filling methods in 3D," in *International Journal of Applied Mathematics and Computer Science* vol. 26, ed, 2016, p. 885.

[95] X. Guo, J. Xiao, and Y. Wang, "A survey on algorithms of hole filling in 3D surface reconstruction," *The Visual Computer,* September 02 2016.

[96] J. Wang and M. M. Oliveira, "Filling holes on locally smooth surfaces reconstructed from point clouds," *Image and Vision Computing,* vol. 25, pp. 103-113, 2007/01/01/ 2007.

[97] P. Liepa, "Filling holes in meshes," presented at the Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Aachen, Germany, 2003.

[98] G. Barequet, M. Dickerson, and D. Eppstein, "On triangulating three-dimensional polygons," *Computational Geometry,* vol. 10, pp. 155-170, 1998/06/01/ 1998.

[99] Z. Li, D. S. Meek, and D. J. Walton, "Polynomial blending in a mesh hole-filling application," *Computer-Aided Design,* vol. 42, pp. 340-349, 2010/04/01/ 2010.

[100] A. Bowyer, "Computing Dirichlet tessellations*," *The Computer Journal,* vol. 24, pp. 162-166, 1981.

[101] J. Branch, F. Prieto, and P. Boulanger, "A Hole-Filling Algorithm for Triangular Meshes Using Local Radial Basis Function," in *Proceedings of the 15th International Meshing Roundtable*, P. P. Pébay, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 411-431.

[102] X. J. Wu, M. Y. Wang, and B. Han, "An Automatic Hole-Filling Algorithm for Polygon Meshes," *Computer-Aided Design and Applications,* vol. 5, pp. 889-899, 2008/01/01 2008.

[103] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*, 2002, pp. 428-441.

[104] T. Ju, "Robust repair of polygonal models," *ACM Trans. Graph.,* vol. 23, pp. 888-895, 2004.

[105] S. Bischoff, D. Pavic, and L. Kobbelt, "Automatic restoration of polygon models," *ACM Trans. Graph.,* vol. 24, pp. 1332-1352, 2005.

[106] A. Kumar and A. M. Shih, "Hybrid Approach for Repair of Geometry with Complex Topology," in *Proceedings of the 20th International Meshing Roundtable*, W. R. Quadros, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 387-403.

[107] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Transactions on Visualization and Computer Graphics,* vol. 9, pp. 191-205, 2003.

[108] L. A. Piegl and W. Tiller, "Filling n-sided regions with NURBS patches," *The Visual Computer,* vol. 15, pp. 77-89, April 01 1999.

[109] K.-L. Shi, J.-H. Yong, J.-G. Sun, J.-C. Paul, and H.-J. Gu, "Filling n-sided regions with G 1 triangular Coons B-spline patches," *The Visual Computer,* vol. 26, pp. 791-800, June 01 2010.

[110] M. Szilvasi-Nagy, "Filling holes with B-spline surfaces," *Journal for Geometry and Graphics,* vol. 6, pp. 83-98, 2002.

[111] Y. S. Lee, B. K. Choi, and T. C. Chang, "Cut distribution and cutter selection for sculptured surface cavity machining," *International Journal of Production Research,* vol. 30, pp. 1447-1470, 1992/06/01 1992.

[112] Z. C. Chen, Z. Dong, and G. W. Vickers, "Automated surface subdivision and tool path generation for 31212-axis CNC machining of sculptured parts," *Computers in Industry,* vol. 50, pp. 319-331, 2003/04/01/ 2003.

[113] Y. S. Lee, "Mathematical modelling using different endmills and tool placement problems for 4- and 5-axis NC complex surface machining," *International Journal of Production Research,* vol. 36, pp. 785-814, 1998/03/01 1998.

[114] G. V. V. Ravi Kumar, K. G. Shastry, and B. G. Prakash, "Computing non-self-intersecting offsets of NURBS surfaces," *Computer-Aided Design,* vol. 34, pp. 209-228, 2002/03/01/ 2002.

[115] L. A. Piegl and W. Tiller, "Computing offsets of NURBS curves and surfaces1Research supported by NSF grant no: DDI-95261191," *Computer-Aided Design,* vol. 31, pp. 147-156, 1999/02/01/ 1999.

[116] G. V. V. Ravi Kumar, K. G. Shastry, and B. G. Prakash, "Computing offsets of trimmed NURBS surfaces," *Computer-Aided Design,* vol. 35, pp. 411-420, 2003/04/15/ 2003.

[117] G. C. Loney and T. M. Ozsoy, "NC machining of free form surfaces," *Computer-Aided Design,* vol. 19, pp. 85-90, 1987/03/01/ 1987.

[118] S.-G. Lee, H.-C. Kim, and M.-Y. Yang, "Mesh-based tool path generation for constant scallop-height machining," *The International Journal of Advanced Manufacturing Technology,* vol. 37, pp. 15-22, April 01 2008.

[119] C. C. Lo, "Efficient cutter-path planning for five-axis surface machining with a flat-end cutter," *Computer-Aided Design,* vol. 31, pp. 557-566, 1999/08/01/ 1999.

[120] J. H. Cho, J. W. Kim, and K. Kim, "CNC tool path planning for multi-patch sculptured surfaces," *International Journal of Production Research,* vol. 38, pp. 1677-1687, 2000/05/01 2000.

[121] G. Elber and E. Cohen, "Toolpath generation for freeform surface models," *Computer-Aided Design,* vol. 26, pp. 490-496, 1994/06/01/ 1994.

[122] Z. Han and D. C. H. Yang, "Iso-phote Based Tool-path Generation for Machining Free-form Surfaces," *Journal of Manufacturing Science and Engineering,* vol. 121, pp. 656-664, 1999.

[123] K. Suresh and D. C. H. Yang, "Constant Scallop-height Machining of Free-form Surfaces," *Journal of Engineering for Industry,* vol. 116, pp. 253-259, 1994.

[124] R.-S. Lin and Y. Koren, "Efficient Tool-Path Planning for Machining Free-Form Surfaces," *Journal of Engineering for Industry,* vol. 118, pp. 20-28, 1996.

[125] Y.-S. Lee, "Non-isoparametric tool path planning by machining strip evaluation for 5-axis sculptured surface machining," *Computer-Aided Design,* vol. 30, pp. 559-570, 1998/06/01/ 1998.

[126] V. Giri, D. Bezbaruah, P. Bubna, and A. Roy Choudhury, "Selection of master cutter paths in sculptured surface machining by employing curvature principle," *International Journal of Machine Tools and Manufacture,* vol. 45, pp. 1202-1209, 2005/08/01/ 2005.

[127] B. H. Kim and B. K. Choi, "Guide surface based tool path generation in 3-axis milling: an extension of the guide plane method," *Computer-Aided Design,* vol. 32, pp. 191-199, 2000/03/01/ 2000.

[128] S. Marshall and J. G. Griffiths, "A survey of cutter path construction techniques for milling machines," *International Journal of Production Research,* vol. 32, pp. 2861-2877, 1994/12/01 1994.

[129] M. Held and C. Spielberger, "A smooth spiral tool path for high speed machining of 2D pockets," *Computer-Aided Design,* vol. 41, pp. 539-550, 2009/07/01/ 2009.

[130] A. Lasemi, D. Xue, and P. Gu, "Recent development in CNC machining of freeform surfaces: A state-of-the-art review," *Computer-Aided Design,* vol. 42, pp. 641-654, 2010/07/01/ 2010.

[131] C.-Y. Lin, Y.-Y. Hwang, and J.-Y. Lai, "Interference-free cutting-path generation based on scanning data," *The International Journal of Advanced Manufacturing Technology,* vol. 13, pp. 535-547, August 01 1997.

[132] S. C. Park, "Sculptured surface machining using triangular mesh slicing," *Computer-Aided Design,* vol. 36, pp. 279-288, 2004/03/01/ 2004.

[133] S. C. Park, "Tool-path generation for Z-constant contour machining," *Computer-Aided Design,* vol. 35, pp. 27-36, 2003/01/01/ 2003.

[134] S. Yuwen, G. Dongming, J. zhenyuan, and W. Haixia, "Iso-parametric tool path generation from triangular meshes for free-form surface machining," *The International Journal of Advanced Manufacturing Technology,* vol. 28, pp. 721-726, April 01 2006.

[135] X. J. Xu, C. Bradley, Y. F. Zhang, H. T. Loh, and Y. S. Wong, "Tool-path generation for five-axis machining of free-form surfaces based on accessibility analysis," *International Journal of Production Research,* vol. 40, pp. 3253-3274, 2002/01/01 2002.

[136] R. A. Horn, R. A. Horn, and C. R. Johnson, *Matrix analysis*: Cambridge university press, 1990.

[137] C. Zhang, W. Wang, J. Wang, and X. Li, "Local computation of curve interpolation knots with quadratic precision," *Computer-Aided Design,* vol. 45, pp. 853-859, 2013.

[138] T. H. Fay, "The butterfly curve," *American Mathematical Monthly,* pp. 442-443, 1989.

[139] A. Gálvez, A. Iglesias, A. Avila, C. Otero, R. Arias, and C. Manchado, "Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting," *Applied Soft Computing,* vol. 26, pp. 90-106, 2015.

[140] (1-March-2018). *NACA 7210 AIRFOIL*. Available: http://airfoiltools.com/airfoil/naca4digit?MNaca4DigitForm%5Bcamber%5D=7&MNaca4DigitForm%5Bposition%5D=25&MNaca4DigitForm%5Bthick%5D=10&MNaca4DigitForm%5BnumPoints%5D=200&MNaca4DigitForm%5BcosSpace%5D=0&MNaca4DigitForm%5BcosSpace%5D=1&MNaca4DigitForm%5BcloseTe%5D=0&yt0=Plot

[141] J. S. Lim, "Two-dimensional signal and image processing," *Englewood Cliffs, NJ, Prentice Hall, 1990, 710 p.,* 1990.

[142] C. Bremer, "Adaptive strategies for manufacturing and repair of blades and blisks," in *ASME Turbo Expo 2000: Power for Land, Sea, and Air*, 2000, pp. V004T01A010-V004T01A010.

[143] P. H. S. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application to Estimating Image Geometry," *Computer Vision and Image Understanding,* vol. 78, pp. 138-156, 2000/04/01/ 2000.

[144] W. Gander and J. Hrebicek, *Solving problems in scientific computing using Maple and Matlab®*: Springer Science & Business Media, 2011.

[145] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*: Pearson Education, 2004.

Appendix 1 Pseudo code for implementing serial bisection.

Input: data points $Q_i = (x_i, y_i, \dots)|_{i=1}^{n}$, parameter $t_i|_{i=1}^{n}$, maximum fitting error ϵ, degree of fitted B-spline p.

Output: Knot vector Z with the knots coincide to the sample points, fitted local B-spline function set.

Pseudo code for serial bisection method:

```
Initialization:
 S = [Q_1, Q_2, ..., Q_n]^T;    T = [t_1, t_2, ..., t_n]^T;
StartIdx = 1;
EndIdx = n;
Z_local = (0, ...,0)_{1×(2p+2)}
E_max = 0
Loop:
while EndIdx > StartIdx
     LeftIdx = StartIdx;
     RightIdx = EndIdx
     while (RightIdx – LeftIdx) <= 1
            if (StartIdx+p)>= n
                 break;
            endif
            LocalBsplineFitting();
            Success = 0;
             if E_Max ≤ ε
                 Success = 1;
                 Save temporary local spline
                 Save temporary E_max
            endif
            if  Success == 1
                 LeftIdx = EndIdx;
            else
                 RightIdx = EndIdx;
           endif
           EndIdx = floor((LeftIdx+RightIdx)/2);
    endwhile
    StartIdx = LeftIdx + 1;
    EndIdx = n;
```

```
    Save local spline
    Save knot
endwhile
```

Subfunction:

```
LocalBsplineFitting(){
    % Computing b-spline basis function
```

$$\mathrm{Z}_{local} = \left(\underbrace{T(\mathrm{StartIdx}), \dots, T(\mathrm{StartIdx})}_{p+1}, \underbrace{T(\mathrm{EndIdx}), \dots, T(\mathrm{EndIdx})}_{p+1} \right);$$

$$N_{mat} = \begin{pmatrix} N_{0,p}\big(T(\mathrm{StartIdx})\big) & N_{1,p}\big(T(\mathrm{StartIdx})\big) & \cdots & N_{p,p}\big(T(\mathrm{StartIdx})\big) \\ N_{0,p}\big(T(\mathrm{StartIdx}+1)\big) & N_{1,p}\big(T(\mathrm{StartIdx}+1)\big) & \cdots & N_{p,p}\big(T(\mathrm{StartIdx}+1)\big) \\ \vdots & & \ddots & \vdots \\ N_{0,p}\big(T(\mathrm{EndIdx})\big) & N_{1,p}\big(T(\mathrm{EndIdx})\big) & \cdots & N_{p,p}\big(T(\mathrm{EndIdx})\big) \end{pmatrix};$$

```
    % identify control points for the last segment by solving
equation 3.3
```

$$P = (N_{mat}^T N_{mat})^{-1} N_{mat}^T S$$

```
    % Computing fitting error
```

$$E_0 = S - N_{mat} P;$$

$$E^2 = E_0 \odot E_0 \;;$$ `% element-wise multiplication`

$$E_{Max} = \mathrm{sqrt}(\max(\mathrm{sum}(E^2, 2)));$$

```
}
```

Appendix 2: Pseudo code for parallel bisection

Input: Curve dataset $S_{n\times m}$ ($S = \{X_{n\times 1}, Y_{n\times 1}, \dots\}$), parameteric vector $T_{n\times 1}$, number of pieces at the start Ω, fitting error ϵ, degree of the fitted b-spline p.

Output: The coarse knot vector in indexing number Z_K

Pseudo code for parallel bisection:

```
% Initializing
 Z_k = 1:round((N-1)/(Ω-1)):n;        %  initial knot vector
 dZ_k = diff(Z_k);                    % knot vector difference
 ΔZ_max = max dZ_k;                   % maximum size of a piece
 TestResult = ones(1, dZ_k);   % one array; assuming all pieces do not pass the error test.
 % Main loop
while (ΔZ_max ≥ 2(p+1))
% Half split
    j = 2;       Z_ktemp(1) = Z_k(1);
    for i = 1:numel(dZ_k)                                   % for each piece
        if (TestResult(i)==1)&&( dZ_k(i) ≥ 2(p+1))          % half split the piece
            Midpoint = round(0.5 × (Z_k(i+1) + Z_k(i));
            Z_ktemp(j) = Midpoint;
            TestResult_temp(j-1) = 1;
            j = j + 1;
            Z_ktemp(j) = Z_k(i+1) ;
            TestResult_temp(j-1) = 1;
            j = j + 1;
        else            % keep the piece
            Z_ktemp(j) = Z_k(i+1) ;
            TestResult_temp(j-1) = TestResult_temp(i);
            j = j + 1;
        endif
    endfor
% Evaluate fitting error
    - Update new Z_k, dZ_k, ΔZ_max, TestResults   Z_k = Z_ktemp;  ΔZ_max = ceil(0.5 × ΔZ_max); TestResult = TestResult_temp;
    foreach new pieces
```

```
        - Fit each pieces by one-piece b-spline function.
        - Evaluate fitting errors.
        - Compare the fitting errors with control threshold
        if FittedError > ϵ
            TestResult(i) = 1;
        else
            TestResult(i) = 0;
        endif
    endfor
% Join sequential pieces
    - find pieces have just passed the error test
    foreach i = found pieces
        % join with left piece
        if i ~= 1      % not the first piece
            if left piece passed the error test
               - Try to join two pieces, if the new piece passes the
error test, the knot Z_k(i) would be eliminated.
            endif
        endif
        - Update date Z_k and piece dataset
        % join with right piece
        if i ~= numel(dZ_k)   % not the last piece
             if right piece passed the error test
               - Try to join two pieces, if the new piece passes the
error test, the knot Z_k(i + 1) would be eliminated.
            endif
        endif
        - Update date Z_k and piece dataset
    Endfor
% Shift small pieces
    if ΔZ_max < 2(p + 1)
         dZ_k = diff(Z_k);
        Smallpieces = find(dZ_k < 2(p + 1));
        if Smallpieces ~= null         % exist small pieces
            for i = 1:numel(Smallpieces)
                % expand left piece using serial bisecting method
                if i ~= 1                                        % not the
first piece
                    LeftIdx = Z_k(i);
                    RightIdx = Z_k(i + 1) − 1;
```

```
                    StartIdx = Z_k(i − 1);
                   SerialBisectingLeft2Right();
                   - Update date Z_k and piece data,
                end
                % expand right piece using serial method
                if i ~= numel(dZ_k)                    % not the last
piece
                    if the right piece is large (not small)
                        LeftIdx = Z_k(i);
                        RightIdx = Z_k(i + 1) − 1;
                        EndIdx = Z_k(i + 2) − 1;
                       SerialBisectingRight2Left();
                        - Update date Z_k and piece dataset
                    endif
                endif
            endfor
            break;
        endif
    endif
endwhile
```

Appendix 3: Pseudo code for solving optimal knot

This part summaries the algorithm to solve optimal knot of two-piece B-spline and provides the details in implementing the algorithm.

Input: A set of data points of two two-piece b-splines $\{T_{d\times 1}, S_{d\times m}\}$, the starting and ending indexing of each local B-spline functions $[a, b, c, d]$, degree of b-spline p, number of samples for scanning L, number of loop for GN M, minimum kink angle α_{min}, minimum smoothness k and scanning all flag *ScanAllFlag*.

Output: Optimal knot ζ_{op} and knot multiplication η of the optimal knot.

```
Step 1:     Estimating knot searching range
-  Estimating searching range for each multiple level.
for i = 1: (p + 1)
    - LeftRange[i] = min(b-a-p,3) + p - i;
    - RightRange[i] = min(d-c-p,3) + p - i;
endfor
-  Computing step h = sqrt(macheps)
Step 2: Uniformly scan to find coarse knot location
η_max = p + 1 ; maximum multiple fold (discontinuity)
for j = (p + 1):-1:1
    -  Computing uiniform knot within the searching range.
    Knotstep = (T[c+ RightRange [j]]- T[b-LeftRange[j]]) / (L-1);
    StartPoint = T[max(b-LeftRange[j],a)];
    StopPoint = T[min(c+ RightRange [j],d)];
    if StopPoint > StartPoint
        KnotSeach = StartPoint :Knotstep : StopPoint;
        for i = 1:L
            ζ = KnotSeach[i];
            -  Forming two-piece knot vector
                Z(T[a], ζ, T[d]) = ⟨T[a], ..., T[a], ζ, ..., ζ, T[d], ..., T[d]⟩
                                   (p+1)        (j)      (p+1)
           -  Forming N matrix
                                N = [ N^1_{l×(p+1)}      0          ]
                                    [ 0                  N^2_{v×(p+1)} ]_{n×(2p+2)}
            - Computing control point P
                                P = (N^T N)^{-1} N^T S
            - Computing maximum error
                        e = max( sqrt( sum( (S − NP) ⊙ (S − NP), 2 ) ) )
            - Computing kink angle
                        α = acos( s_1^{(p−J+1)}(ζ) . s_2^{(p−J+1)}(ζ) / ( ||s_1^{(p−J+1)}(ζ)|| ||s_2^{(p−J+1)}(ζ)|| ) )
             -  Saving the Error and angle
             DisError[i] = e;
             Alpha[i] = α;
        Endfor
        if (j ==  p+1)
            -  Idx = find(DisError > (min(DisError)+1e-10)
            -  DisIdx = round(Idx[1]+Idx[end]);
```

```
            -  Error[j] = DisError[DisIdx];
            -  Angle[j]= Alpha[DisIdx];
            -  OptimalKnot[j]= KnotSeach[DisIdx];
            if ~ScanAllFlag
                -  OptimalKnot[1:p] = KnotSeach[DisIdx];
                -     SearchRange[1:p,:]   =   {T[max(DisIdx   -1,1)],
T[min(DisIdx +1,n)]};
                break;
            endif
        else
            -  Idx = find(min(DisError));
            -  Error[j] = DisError[Idx];
            -  Angle[j] = Alpha[Idx];
            -  OptimalKnot[j] = KnotSeach[Idx];
            -  SearchRange[j,:] = {T[max(Idx-1,1)], T[min(Idx+1,n)]};
        endif
    else
        \eta_{max} = \eta_{max} - 1;
    endif
endfor
Step 3: Optimal knot solving using Gauss-Newton method
for j = 1 : min(p, \eta_{max})      % for each multiple knot case
    for i = 1:M    % M times iteratively run
        - Forming the knot vector of fitted B-spline: Z(T[a], \zeta, T[d]) and
Z(T[a], \zeta + h, T[d])
        - Computing N(\zeta) and N(\zeta + h) matrices.
        - Computing G(\zeta) and G(\zeta + h)
        - Computing G'(\zeta)
        - Saving previous step \Delta\zeta_0 = \Delta\zeta
        - Computing new step \Delta\zeta = \left(G'(\zeta)^T G'(\zeta)\right)^{-1} G'(\zeta)^T G(\zeta)
        - Checking non-convergent case
        if (\Delta\zeta_0 \times \Delta\zeta < 0)
            \Delta\zeta = 0.5 \times \Delta\zeta;
        endif
        - Saving knot location \zeta_0 = \zeta
        - Updating new knot location \zeta = \zeta - \Delta\zeta
        -              Saturating              knot              location
\zeta = \begin{cases} \text{SearchRange}[j,1] & if\ \zeta < \text{SearchRange}[j,1] \\ \text{SearchRange}[j,2] & if\ \zeta > \text{SearchRange}[j,2] \\ \zeta & else \end{cases}
        - Break condition
        if  abs(\zeta - \zeta_0) < h
            cntflag = cntflag + 1;
            if cntflag >= 2
                break;
            endif
        else
            cntflag = 0;
        endif
    endfor
    -  Computing maximum fitting error  ErrorMax = \max G
    -  Computing kink angle
      \alpha = \text{acos}\left( \frac{\overrightarrow{s_1^{(p-J+1)}(\zeta)} . \overrightarrow{s_2^{(p-J+1)}(\zeta)}}{\left\| \overrightarrow{s_1^{(p-J+1)}(\zeta)} \right\| \left\| \overrightarrow{s_2^{(p-J+1)}(\zeta)} \right\|} \right)
    -  Saving fitting error, kink angle and optimal knot
       Error[j] = ErrorMax;
       OptimalKnot[j] = \zeta;
```

```
       Angle[j] = α ;
endfor
Step 4: Selecting continuity and optimal knot
Returning the continuous level k and optimal knot ζ satisfy the
condition
η = p − k ;
η = min(η, η_max);
SelectedAngle = Angle[1: η] ;
Idx = find(SelectedAngle >α_min )
if Idx ~= null
    Error1 = Error[Idx];
    [~,Idx1] = min(Error1);
    OptimalIdx = Idx[Idx1];
    - Optimal knot output: ζ_op = OptimalKnot[OptimalIdx];
    - Multiple knot output: η = OptimalIdx;
else
    [~,OptimalIdx] = max(SelectedAngle);
    - Optimal knot output: ζ_op = OptimalKnot[OptimalIdx];
    - Multiple knot output: η = OptimalIdx;
Endif
```

Program explanation:

The program processes a pair of two sequential one-piece B-spline datasets resulting from bisecting process to find the optimal knot and its continuity. The dataset contains parametric vector $T_{d\times1}$ and curve vector $S_{d\times m}$, where n is the number of data in the dataset and m is the curve dimension. The datasets have already been coarsely separated by their indexing number [a,b] and [c,d] of the first and the second spline pieces, respectively. The algorithm requires some tuning parameters, e.g. for the number of uniform knots L needed in a searching process and the number of loop for Gauss-Newton solver M, a flag namely *ScanAllFlag*, to decide the scanning process for non-discontinuous cases. The program also requires two parameters in the optimal knot selection namely the smallest acceptable kink angle α_{min} and the minimum continuity level C(k).

The input datasets have already been coarsely separated by the bisecting process. The main disadvantage of the bisecting process is that it cannot give a good knot location in a case when the spline piece has a number of sample smaller than the order of the fitted curve. In a case when a spline piece has sufficient data to fully define its function, the optimal knot is usually located within two samples next to the coarse knot. Another thing needs to be considered in the estimation of the knot searching range is the condition to solve the least square problem. The least square problem would only be solved if the input data is fully or over constrained. In a case of a two-

piece spline, we can easily see the condition being "a two-piece spline is fully defined if and only if the matrix N is in full rank". The condition would be satisfied if the following two conditions are satisfied. The first condition is about the total samples inside the dataset. The number of data in the dataset must be larger or equal to $(p + \eta)$. The second condition is related to the minimum number of samples inside a spline piece, i.e. the number of samples in single pieces must be larger or equal to the multiples η of the interior knot.

As seen on the first part of subsection 3.2.4, if the data is sampled from a B-spline curve, we would only need to uniformly scan knots for the discontinuous case to narrow down the searching range of other multiple cases. However, in a case when the data is not originated from a B-spline curve, such as approximating a function or measured data, a uniform scan for only the discontinuous case would not be sufficient to find the optimal knot. Therefore, a uniform scanning is also required for other multiple knot cases to narrow down their searching areas. A variable namely *ScanAllFlag* is used to select uniform scanning for non-discontinuous cases. Please note that turning on the flag will sacrifice the processing time. The scanning ranges for non-discontinuous cases are larger than the discontinuous case to treat the cases when the number of data in a single-piece is smaller than the order of the fitted curves.

Evaluation of the uniform knots is rather straightforward. A knot is first used to form the knot vector Z and subsequently the basic function matrix N is computed. The control point vector P is then computed using a typical linear least square formula. The fitting error and joining kink angle is computed and saved in every step. In the discontinuous case, the smallest piecewise constant is concluded by finding the smallest error area. Due to the computational error, the lowest piecewise constant is selected with a small variation from the minimum error value (1e-10 is selected in our algorithm). As analyzed in the beginning of this subsection, the accurate knot for discontinuous cases cannot be obtained by the two-piece B-spline approach. The optimal discontinuous knot is than selected in the middle of the lowest error piecewise constant. In other non-discontinuous multiple knot cases, the searching areas are selected within one sample of the minimum points.

The next step is to find the optimal knots for non-discontinuous cases. As analyzed in the first part of this section, the error functions have only one optimal point within a

sample step. Therefore, we can apply traditional gradient methods to find the optimal point. In this paper, we employ the Gauss-Newton method to solve the optimization problems. The implementation details are illustrated in step 3 of the algorithm. In Gauss-Newton method, the optimization sometime poses a non-convergence problem, because of its severe non-linear nature. In the algorithm, we propose a simple method by reducing the solution step to half if the sign of the solution step changed.

The last step is to decide the best multiple-knot. The best solution must satisfy two conditions: (*i*) the joining kink angle has to be larger than the input criterion, α_{min}, and (*ii*) minimization of the fitting error. In the case that there are no multiple cases satisfying the first condition, we will select the multiple-knot case that has a larger joining kink angle.

Appendix 4: Further discussion on fitting data sampled from a spline function

Table 1 provides some selected cases to explore the proposed method in fitting data from B-spline functions. A dataset has 1001 samples, which is uniformly distributed from 0 to 1. The interior knots and control points are randomly generated. The detailed numbers are given in the second and third columns. Tuning parameters and fitting results are also given in the remaining columns.

Table 1 some selected B-spline fitting cases

Case No.	B-spline					
	Curve parameters		Tuning parameters	Results		
	Interior knot	Control points		Knot residual errors	Fitting error MSE (ME)	Processing time (s)
1	0.0975, 0.1270, 0.1576, **0.2785, 0.2785,** 0.6324, 0.8147, 0.9058, 0.9134, **0.9575, 0.9575,** 0.9706	43.5629 -21.9836; -28.1009 -37.5540; 41.7354 13.8033; 15.7865 -16.3747; 16.8415 44.5216; -9.5789 -0.9723; -8.3908 23.8681; 20.0089 -17.0458; -66.7244 17.3024; -61.3980 35.7705; -91.3120 -7.6142; -45.0084 -30.9239; 68.4833 12.9979; 54.5535 -26.2096; 5.9591 -12.3227; 10.9505 -63.8203	$p = 3$, $\epsilon = 1e-6$, $\alpha_{min} = 1$, $\Omega = 10$, $k = -1$, $L = 10$, $M = 10$, ScanAllFlag = 1	-9.702e-12, -2.387e-14, -4.383e-14, **-1.443e-15, -1.443e-15,** -4.319e-14, -3.874e-13, -1.100e-09, -2.451e-09, **-2.579e-11, -2.579e-11,** -4.534e-11	2.1806e-14 (8.4577e-07)	0.68
2	0.0159, **0.0257, 0.0257, 0.0257,** 0.1789, 0.1890, 0.2027, 0.2027, 0.5251, 0.6607, 0.8623, 0.8964, 0.9412, 0.9550, 0.9550, 0.9711, 0.9711	34.0669 28.8683; 49.0064 11.2947; -60.3171 75.1773; 13.9671 -27.8541; -5.5041 -94.5445; 32.9569 42.4848; 18.3631 27.8358; 6.9897 42.4366; 8.5920 62.6100; 41.5537 9.1750; 15.2572 29.0424; 22.4530 77.5639; 6.3710 -15.8599; 2.7949 -16.7661; -17.0091 -74.2350; 66.9951 -20.1932; -2.8528 -18.3535; -52.5438 -6.1148; 10.2738 28.8462; -29.6878 59.9450; 19.2656 16.2393	$p = 3$, $\epsilon = 1e-6$, $\alpha_{min} = 1$, $\Omega = 10$, $k = -1$, $L = 11$, $M = 10$ ScanAllFlag = 1	-1.804e-16, **-1.037e-13, -1.037e-13, -1.037e-13**, -1.544e-12, -3.366e-12, 2.578e-14, 2.578e-14, 2.908e-09, 6.051e-14, -2.848e-12, -3.380e-11, -1.603e-11, -5.611e-11, -5.611e-11, -1.071e-11, -1.071e-11	9.5370e-16 (9.0450e-08)	0.81

3	0.0182, 0.0300, 0.0669, 0.0871, 0.5357, 0.5357, **0.7837, 0.7837, 0.7837, 0.7837**, 0.9861, 0.9891	-37.1649 34.7088;- 37.2243 37.4669; 29.0562 -24.6073; 93.5777 -25.5102; - 16.6569 -85.1886; 6.5970 61.1681; - 0.4123 81.3929; 29.4039 25.5439; 28.4196 48.1495; - 46.0174 15.3203; - 43.4006 -0.9348; 45.5901 35.5605; 25.8792 36.9262; - 26.1438 -33.9563; 12.5146 -64.6840; 12.6804 -0.3025	$p = 3$, $\epsilon = 1e-6$, $\alpha_{min} = 5$, $\Omega = 21$, $k = -1$, $L = 11$, $M = 10$, ScanAllFlag = 1	-1.388e-15, -4.703e-09, -6.273e-15, -5.770e-13, -2.232e-14, -2.232e-14, **-9.207e-06, -9.207e-06, -9.207e-06, -9.207e-06,** -3.956e-09, 3.690e-08	4.424e-12 (4.0622e-05)	0.49
4	0.4675, 0.4675, 0.4675**,** 0.6624, 0.6992, **0.8754, 0.8778**	-29.4917 -2.7472; 32.5520 -29.5839; - 27.0494 -11.3169; 30.8630 51.3881; 38.8477 -58.4587; 3.0817 -41.2673; - 48.7069 -6.8617; - 88.2647 21.9906; - 62.3831 40.5862; 64.8031 -41.8770; 29.4571 7.5956	$p = 3$, $\epsilon = 1e-6$, $\alpha_{min} = 5$, $\Omega = 15$, $k = -1$, $L = 10$, M = 10, ScanAllFlag = 1	-2.720e-15, -2.720e-15, -2.720e-15' -1.740e-11, -8.069e-12, **-1.610e-05, -7.973e-05**	9.8063e-06 (0.0170)	0.33
5	0.0405, 0.0654, 0.0707, 0.0894, 0.0958, 0.0958, 0.1822, 0.1822, 0.4522, 0.5978, 0.6104, 0.7016, **0.7313, 0.7319, 0.8122, 0.8146,** 0.9039	-15.9938 74.7236; 35.7199 18.0746; 13.8208 -3.2704; 19.0836 61.0421; 46.4001 15.3345; - 16.3042 -29.7101; - 0.8816 43.4445; - 6.4741 6.5342; - 90.4129 -36.0837; 17.4897 12.2327; 18.1252 -18.2312; 31.3901 -2.3955; - 16.5180 -67.9798; - 6.9273 -30.3777; 41.2263 -59.5293; - 5.4938 67.0179; 32.3061 24.2833; 52.4240 4.4178; 19.2111 -16.6427; - 3.0610 43.9823; - 18.9611 -20.6339	$p = 3$, $\epsilon = 1e-5$, $\alpha_{min} = 5$, $\Omega = 15$, $k = -1$, $L = 10$, $M = 10$, ScanAllFlag = 1	-5.412e-16, -6.385e-13, -1.817e-11, -1.416e-14, -2.486e-13, -2.486e-13, -2.714e-12, -2.714e-12, -7.191e-09, -1.233e-10, -5.200e-11, -1.496e-12, **2.946e-4, -2.918e-4, -4.678e-06, 1.270e-05,** -8.559e-13	1.8054e-07 (0.0062)	0.92

The first three cases deal with double, triple and fourfold-knot cases. We can see that the method results in both optimal knots and their multiples for non-discontinuous cases. However, for case 3, the fourfold knot is approximately obtained at 0.7837, as indicated by relatively high residual error (9.207e-6). This is due to the case discussed in the Corollary 3.2.

In case 4, two residual errors for knots 0.8754 and 0.8778 are evidently large compared to the rests because there only two samples within the corresponding piece. In such kind of cases, the serial bisection tends to fail in data separation because it needs at least $(p+1)$ samples for a piece. Any error in data separation could lead to error in the computed optimal knots.

Case 5 shows another failure of the method as indicated by the high residual errors in knots 0.7313, 0.7319, 0.8122 and 0.8146. The first two knots are located within two sequential samples. This means there is no data within them. In such kind of case, the method results in a double-knot at 0.73157. The knots 0.8122 and 0.8146 also fail to be optimized for the same reason as in case 4.

For all cases, the processing time heavily depends on the number of the interior knots need to be solved. It is usually less than a second to fit the data with about 1000 samples.

Appendix 5: The details of fitting the spur gear curve.

Case no.	Fitting parameters	Serial bisection	Parallel bisection
1	Spline Degree	$p = 3$	$p = 3$
	Control error	$\epsilon = 1e - 2$	$\epsilon = 1e - 2$
	Minimum kink angle (degree)	$\alpha_{min} = 1$	$\alpha_{min} = 1$
	Number of pieces at start	$\Omega = 1$	$\Omega = 20$
	Maximum smoothness C(k)	$k = -1$	$k = -1$
	Number of uniform data in scanning step	$L = 10$	$L = 11$
	Number of loop in Gauss-Newton solving	$M = 10$	$M = 10$
	Scanning knot	ScanAllFlag = 1	ScanAllFlag = 1
	Data bisection time (ms)	130	100
	Coarse knots by bisection step	8,62,66,120,132,186,190,244,256,310,314,368,380,434,438,492,504,558,562,616,628,682,686,740,752,806,810,864,876,930,934,988,1000,1054,1058,1112,1124,1178,1182,1236,1248,1302,1306,1360,1372,1426,1430,1484,1496,1550,1554,1608	8,61,66,120,132,185,190,243,256,309,314,368,379,433,438,492,504,557,562,615,628,681,686,740,751,805,810,864,876,929,934,988,1000,1053,1058,1112,1123,1177,1182,1236,1248,1301,1306,1360,1372,1425,1430,1484,1496,1549,1554,1607
	Optimal interior knots	0.00655, 0.03605, 0.04097, 0.07045, 0.08356, 0.11303, 0.11798, 0.14746, 0.16057, 0.19003, 0.19499, 0.22447, 0.23758, 0.26704, 0.27200, 0.30147, 0.31458, 0.34406, 0.34900, 0.37848, 0.39159, 0.42108, 0.42601, 0.45549, 0.46860, 0.49809, 0.50302, 0.53249, 0.54560, 0.57509, 0.58002, 0.60950, 0.62261, 0.65209, 0.65703, 0.68651, 0.69962, 0.72909, 0.73404, 0.76352, 0.77663, 0.80609, 0.81104, 0.84052, 0.85363, 0.88311, 0.88805, 0.91753, 0.93064, 0.96012, 0.96506, 0.99454	0.00655, 0.03602, 0.04098, 0.07045, 0.08356, 0.11303, 0.11799, 0.14746, 0.16057, 0.19004, 0.19500, 0.22447, 0.23758, 0.26705, 0.27199, 0.30147, 0.31458, 0.34406, 0.34901, 0.37848, 0.39159, 0.42108, 0.42602, 0.45549, 0.46860, 0.49807, 0.50302, 0.53249, 0.54560, 0.57510, 0.58002, 0.60950, 0.62261, 0.65208, 0.65702, 0.68651, 0.69962, 0.72909, 0.73404, 0.76352, 0.77662, 0.80610, 0.81105, 0.84052, 0.85363, 0.88310, 0.88805, 0.91753, 0.93064, 0.96012, 0.96507, 0.99454
	Multiple knot	3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3	3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
	Fitting error	MSE= 2.3410e-06, ME=0.0060	MSE= 2.4212e-06, ME= 0.0055
	Total processing time (ms)	2490	2501
2	Spline Degree	$p = 3$	$p = 3$
	Control error	$\epsilon = 1e - 3$	$\epsilon = 1e - 3$
	Minimum kink angle (degree)	$\alpha_{min} = 1$	$\alpha_{min} = 1$
	Number of pieces at	$\Omega = 1$	$\Omega = 20$

	start		
	Maximum smoothness C(k)	$k = -1$	$k = -1$
	Number of uniform data in scanning step	$L = 10$	$L = 10$
	Number of loop in Gauss-Newton solving	$M = 10$	$M = 10$
	Scanning knot	*ScanAllFlag* = 1	*ScanAllFlag* = 1
	Data bisection time (ms)	197	105
	Coarse knots by bisection step	78 knots	78 knots
	Multiple knot	52 Triple-knots, 26 single-knots	52 Triple-knots, 26 single-knots
	Fitting error	MSE= 1.5617e-07, ME=0.0013	MSE= 6.0987e-08, ME= 0.0012
	Total processing time (ms)	3640	3715
3	Spline Degree	$p = 2$	$p = 2$
	Control error	$\epsilon = 1e - 3$	$\epsilon = 1e - 3$
	Minimum kink angle (degree)	$\alpha_{min} = 5$	$\alpha_{min} = 5$
	Number of pieces at start	$\Omega = 1$	$\Omega = 50$
	Maximum smoothness C(k)	$k = -1$	$k = -1$
	Number of uniform data in scanning step	$L = 10$	$L = 8$
	Number of loop in Gauss-Newton solving	$M = 10$	$M = 10$
	Scanning knot	ScanAllFlag = 1	ScanAllFlag = 1
	Data bisection time (ms)	269	107
	Coarse knots by bisection step	130 knots	181 knots
	Multiple knot	78 single-knots, 52 double-knots	129 single-knots, 52 double-knots
	Fitting error	MSE= 7.4063e-7, ME=0.0018	MSE= 1.1562e-7, ME= 8.8623e-4
	Total processing time (ms)	3970	5635

Appendix 6: The details of tool path generation

A6.1 Introduction

In real remanufacturing applications, reprofiling processes consist of a few machining steps. Firstly, large excessive materials (defects) are removed by milling. Based on the complexity of the workpiece, a three-axis or five-axis milling machine can be used. Secondly, if there are any special requirements on the surface finish, grinding and/or polishing process may be required. Due to resource limitations, this study focuses only on tool path generation for a three-axis milling machine.

Milling is a process where a rotary cutter cuts material on a stock workpiece through its relative movement between the cutter and the workpiece. There are many types of cutter with different tool shapes such as flat end mill, ball nose end mill, fillet mill, etc. Freeform surface machining processes usually uses flat end mill and ball nose mill. The end mill cutter is used to cut flat surfaces, where the surfaces are perpendicular or parallel to the tool. The same type of cutter is also normally use to cut cylinder surfaces that is parallel to the tool, or to rough cut freeform surfaces as it can offer high removal rates. For other planar surfaces than the cases above, a ball nose cutter is usually used.

A tool path is generated based on three factors, namely (*i*) cutting tool parameters, (*ii*) stock geometries and target workpieces, and (*iii*) surface quality after machining. Parameters of cutting tool include tool geometries, i.e. shape and diameter, and cutting parameters, i.e. width of cut (step over), depth of cut, feed rate, cutting speed. Stock geometries are measured from the raw materials, while the target surface geometries correspond to the desired nominal surfaces. The last factor is the finished surface quality that is normally quantified by its roughness and usually it is defined by its maximum scallop height. It is essential to note that some parameters are dependent on other parameters, e.g. the width of cut depends on the value of depth of cut and the maximum scallop height.

Cutting tool parameters:

- Tool diameter: D (mm). Tool diameter needs to be properly selected based on the smallest radius of curvature of the target surfaces. In a milling process of freeform surfaces, the radius of the tool, $R = D/2$, must be smaller than the

smallest radius of curvature of the target surfaces. One thing we need to consider is that selecting a smaller tool radius results in a lower material removal rate. Therefore, the tool diameter needs to be as large as possible to increase cutting productivity. However, as the tool path in this process is generated locally, if the tool diameter is large, it might sometimes result in overcutting of the neighboring features in the workpiece to tool violation.

- Type of cutter: Flat end mill, or ball nose. In most cases, in machining of freeform surfaces, ball nose tools are used for finishing cut. However, the material removal rate of the ball nose is lower than that of the flat end mill tools, for the same tool diameters. Usually, the flat end milling tool is selected for rough cutting and ball nose tools are selected for finishing to reduce the machining time.
- Cutting speed, S (meters per min): Cutting speed is the relative velocity of the tool tip and the workpiece surfaces. Cutting speed depends on the materials of the cutting tools and workpieces, and cooling condition. This parameter is used to calculate the spindle revolution.
- Feed rate, F (meters per min): Feed rate is a relative translational velocity of a cutting tool and a workpiece in a translation direction of the tool. It is sometimes presented in a unit of millimeters per revolution. Higher feed rate would result in a higher material removal rate, cutting force, heat that causes the cutting tool defect and tool wear. Very small feed rate also causes tool rubbing on the workpiece surface that will generate heat and accelerate the tool wear.

Geometries:

- Target geometries: in depth image format taking from estimated nominal surfaces.
- Stock geometries: in depth image format taking from depth images of the measured data.

Finished surface quality:

The surface roughness of a machined surface depends on the width of cut in the finishing cut. As illustrated in Figure A6.1, if a ball nose is used, we would not remove all material in between two passes. The remaining material is called scallop and the maximum height of the scallop is used to calculate the distance between two passes.

Maximum scallop is set as the difference values for roughing and finishing. In roughing the scallop height is set larger than in finishing.

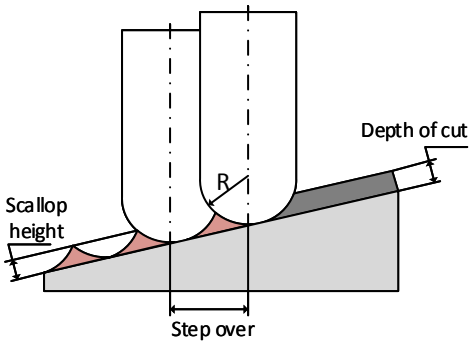


Figure A6.1 Residual material after milling with a ball nose cutter

The quality of surfaces is also affected by the direction when the cutter approaches the surface to cut material into chips. We can categorize the cutter edge approach into two methods of milling, namely conventional milling and climb milling. Figure A6.2 illustrates the cutter approaching modes when the cutter moves during the cutting process. In conventional milling, the chip thickness becomes larger until it is fully separated from the workpiece. In contrast, the chip thickness becomes thinner in climb milling. The surface quality after climb milling is better than conventional milling.

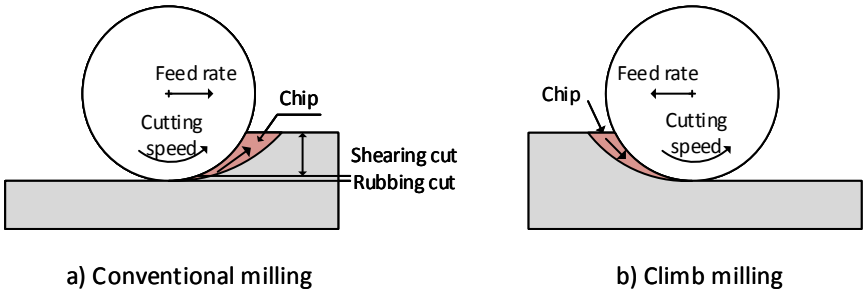


Figure A6.2 Two approaching mode during milling. a) Conventional milling, b) climb milling

In addition, the finishing quality is also affected by the tool paths. For freeform surface machining, we can roughly categorize the tool path into two different types based on width of cut (step over). The first type of tool path forces the cutter to move with a constant step over the whole surface, the second type varies the step over to keep constant scallop height. Comparing these two types of the tool path, the constant step over method provides simpler computational algorithm than its counterpart. However, during each cutting step, the amount of material removal varies depending on the slope of the machined surfaces, that will cause the variation of the cutting force, which can

trigger tool vibration (chatter) that will result in lower surface quality. On the contrary, the iso-scallop toolpath method requires a more complex algorithm, but the amount of material removed is relatively constant, which usually results in higher surface quality especially in high-speed machining. Therefore, in this study, the iso-scallop method is employed to generate the tool path for reprofiling purposes.

Selection of the optimal parameters for cutting speed (S), feed rate (F), depth of cut (t), maximum scallop height, which is based on optimizing the machining cost (cost of cutting tools and machine occupation time), is out of the scope of this study. Therefore, these numbers are chosen arbitrarily but based on rational selection by the operator.

A6.2 Tool path generation procedure

Figure A6.3 presents a procedure to obtain the tool path from the confirmed defects. The procedure takes the estimated nominal surfaces as the target geometries, the measurement surfaces as stock geometries, together with defect information and some cutting information for the user such as, feed rate F, cutter diameters D, tool shape, depth of cut for roughing and finishing, maximum scallop height for roughing and finishing, safety distance from tool tip to the highest point of the stock geometries, and the residual material, h_0, for further grinding/polishing if required.

The defects only occupy a small area, it is not necessary to process to the whole surface in defining the tool paths. In the first task, corresponding data around the defects is selected for further processing. Because lateral resolution of the stock and target geometries are usually not enough to satisfy high requirement in finishing surfaces, finer lateral resolution for the local depth images needs to be obtained to get a good tool path. A B-spline surface fitting technique is used to create continuous surface models for refining purposes.

In the second task, the width of cut (step over), w, for roughing and finishing is calculated based on the information of the cutter diameter, D, and required maximum remaining scallop height, $h_{scallop}$, as shown in equation (A6.1).

$$w = 2.\sqrt{h_{scallop}\left(D - h_{scallop}\right)} \tag{A6.1}$$

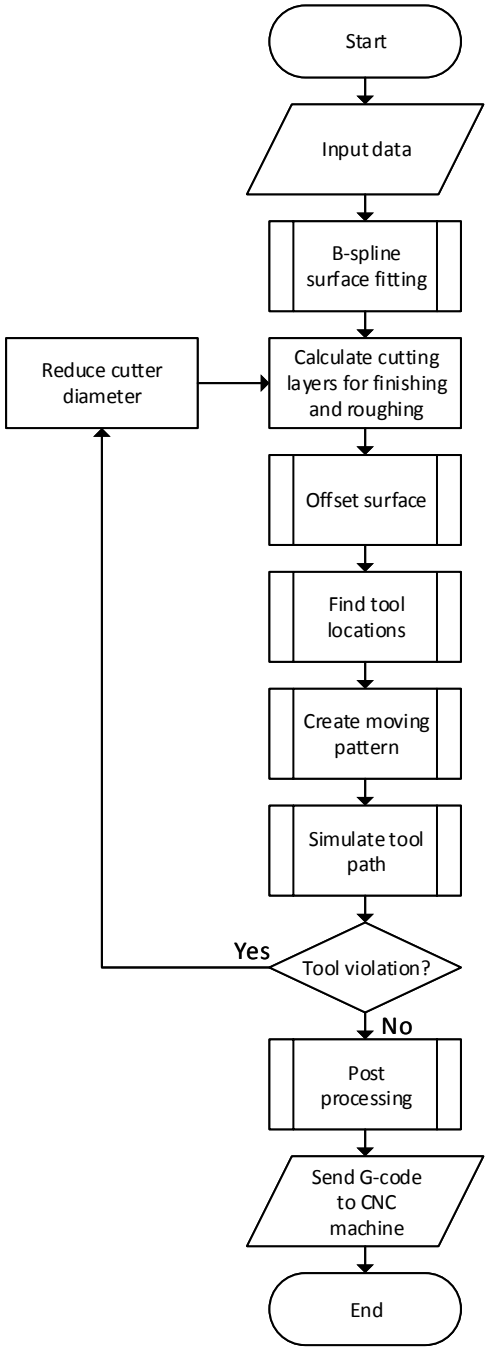


Figure A6.3 Procedure for local toolpath generation

Subsequently, intermediate cutting layers for roughing, n_r, and finishing, n_f, are calculated based on the input depth of cut of roughing h_r and finishing h_f. Let I_n be a local depth image at the confirmed defect resampled from the estimated nominal surface, and I_o be a local depth image of measurement data respectively. The maximum height of the defect is calculated as $H = \max(\max((I_o - I_n))$. We, therefore, have the relationship between the above variables as shown in equation (A6.2), where n_r is round up by the ceiling function.

$$H \leq h_0 + n_f h_f + n_r h_r \;\Rightarrow n_r = \text{ceil}\left(\frac{H - h_0 - n_f h_f}{h_r}\right) \tag{A6.2}$$

Figure A6.4 illustrates an example of different intermediate cutting layers and the parameters in machining processes. In the example, there are five roughing and one finishing layers. In real applications, the number of finishing layer, n_f, usually is selected as one if the same cutter is used for both roughing and finishing, while in a case where the cutters are different for these two cutting steps the number of finishing layer, n_f, is selected as two.

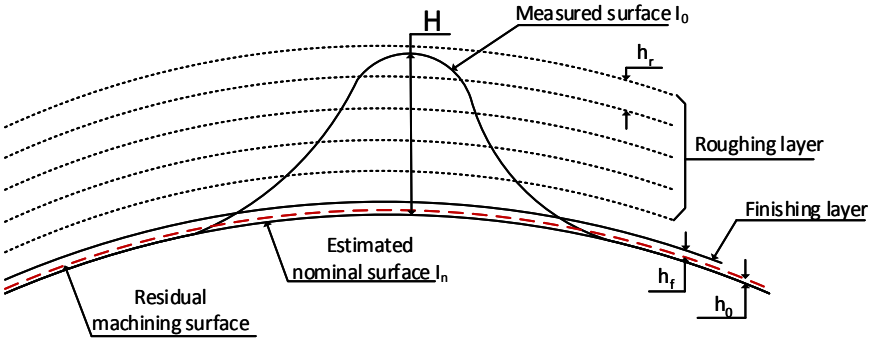


Figure A6.4 Intermediate cutting layers for roughing and finishing

After deciding the number of intermediate cutting layers, an offset procedure is applied to find the intermediate targeting surface for each cutting layer. Subsequently, intersections between the intermediate target surfaces and the measurement data are computed to identify the locations of excessive material on each layer that we need to remove.

In the next task, tool path for cutting layers are defined by approximating the cutter location surfaces by a set of curves. The tool path is then simulated to check whether the cutter hits obstacles or violates the workpieces. If it passes the test, a procedure to convert the tool path to G-code will be executed, and if it is not, the diameter of the cutter will be reduced to create new tool paths.

A6.3 Resampling nominal surface

Our main target is to generate a tool path to remove the confirmed defects. At this stage, we already know the defect boundaries. The tool path program requires the input surface which must be bigger than the defect area with a distance from the defect boundaries at least equal to the cutter diameter to keep empty spaces for the cutter

movement. Therefore, the nominal surface data at the defect and its surrounding area within the cutter diameter are selected for refining the resolution. The B-spline surface fitting technique is employed to approximate the selected area. New local depth images with a higher resolution are computed from the fitted B-spline surface.

A6.4 Surface offset

Offsetting surfaces is a fundamental technique in tool path generation. It is used to create intermediate cutting surfaces when the maximum thickness of material that needs to be removed is larger than the depth of cut. Furthermore, when using ball nose cutters, the offsetting technique is also used to find cutter's location surfaces that when the cutter moves on it, the excessive material is removed to attain the target surfaces.

The estimated nominal surfaces resampled in the previous step are used as the input of the offset processes. Offsetting distances from the target surfaces to finishing and roughing layers are calculated as: $d_f^i = h_0 + ih_f$ and $d_r^j = h_0 + n_f h_f + jh_r$, where i is finishing layer number i^{th}, $i = 1, 2, \ldots, n_f$, and j is roughing layer number j^{th}, $j = 1, 2, \ldots, n_r$.

There are many methods to obtain an offset surface from a given surface available in literature. In this study, an approximation method by finding an envelope surface when a ball, with a radius of the offset length and centered at the surfaces, slides along the surface. The method is proven to be stable and it can eliminate self-intersection problems when offsetting length is larger than the smallest radius of curvature of the surfaces. Figure A6.5 illustrates the approach to find the offset curves/surfaces. By letting the sphere move along the original curve/surface, we can easily obtain the required offset object without any special treatment for self-intersection problems.

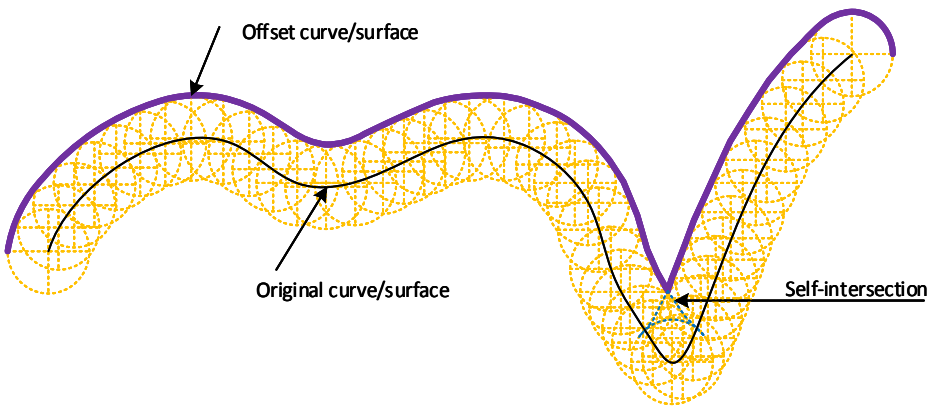


Figure A6.5 Method to approximate offset curves/surfaces

Given a discrete surface, I_n, that is in a form of a depth image, we would like to find an offset surface, J, with the offset distance, d, from the surface, I_n, along the depth axis of the image. A positive offset distance, d, denotes the offset surface on top of the given input surface and vice-versa. We also assume that the spatial resolution of the depth images is Δr. There are two steps to obtain the offset object J:

- Computing a square matrix, M, to represent a northern hemisphere with radius $|d|$, center at $(0,\ 0,\ 0)$. The mask M is a square matrix with size $2k+1$, where $k = \text{ceil}\left(\frac{|d|}{\Delta r}\right)$. Because the mask is a matrix while the projection of a sphere is a circle, therefore elements located outside the sphere are set as NaN (not a number).
- Computing local matrix J_l of the offset depth image J by the following equation (5.6):

$$\begin{aligned} J_l &= \max(J_l, a_{nm} + M) \quad \text{when } d > 0 \\ J_l &= \min(J_l, a_{nm} - M) \quad \text{when } d < 0 \end{aligned} \tag{A6.3}$$

  where $J_l = J_{(n-k:n+k)\times(m-k:m+k)}$ is a sub matrix of the offset surface J, a_{nm} is value of element at row n column m of matrix J

A6.5 Identify cutter locations

The offset surfaces provide targets for each machining step, this subsection discusses the method to find surfaces containing the cutter's origin such that when the cutter moves on those surfaces, it will cut any excessive material to create the target surfaces. In freeform surface machining, the target surfaces are smooth, therefore the tool path location need also be smooth to ensure surface quality.

Figure A6.6 presents two common types of cutter, a flat end mill and a ball nose mill, together with their tool path to cut a given curve profile. Please note the origin of the flat end mill as indicated in the top left panel of Figure A6.6, and the top right panel defines that of the ball nose mill cutter.

The tool path for flat end mill is the envelope of the three curve profiles. The first profile is the offset curve of the target profile by a distance r in normal directions, the second profile is the left shift of the first profile, and the last is the right shift of the first profile by a distance of $^{D}/_{2} - r$. While the tool path for ball nose mill is simply an offset of the target profile by a distance $R = {}^{D}/_{2}$ following the normal directions.

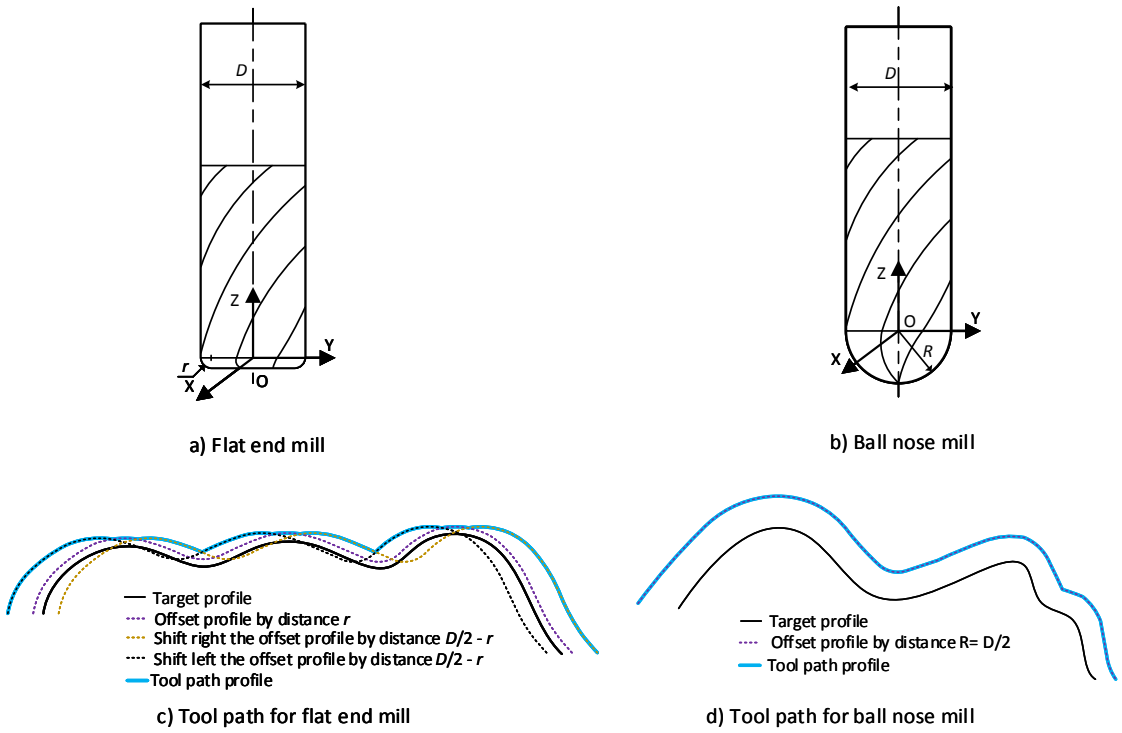


Figure A6.6 Possible cutter positions to machine a curve profile, a) definition origin of flat end mill cutter, b) definition origin of ball nose mill cutter, c) possible cutter positions when machining a curve profile by flat end mill, d) possible cutter positions when machining a curve profile by ball nose mill cutter

In the case of milling of a freeform surface, the cutter positions are located on a surface. Therefore, the method to find the cutter location needs a modification from the curve profile case illustrated in Figure A6.6. We present the cutter's profile as a mask matrix M_c. Please also note that the mask M_c is a square matrix with size $2k + 1$, where $k = \text{ceil}\left(\frac{D}{\Delta r}\right)$. When presenting the cutter at the mask M_c, the origin of the cutter is set at the tool length position as illustrated in Figure A6.7, that will also help to reduce the setup time on CNC machines.

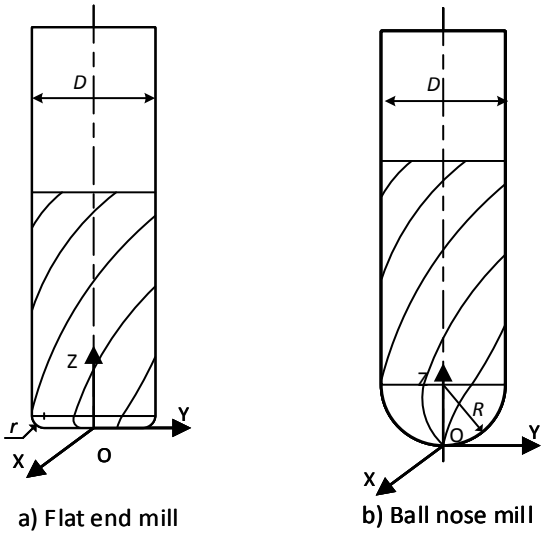


Figure A6.7 Definition origin of cutter when using mask, a) flat end mill, b) ball nose mill cutter

Let U be a depth image representing the intersecting surface of an intermediate target layer, J, and the measurement image I_o. We have

$$U = J(I_0 \geq J). \tag{A6.4}$$

The cutter locations are located in a depth image, I_{cutter}, with the same resolution Δr, and values of elements are calculated as:

$$b_{ij} = \max(U_l - M_c) \tag{A6.5}$$

where $U_l = U_{(i-k:i+k)\times(j-k:j+k)}$ is a sub matrix of the depth image U.

A6.6 Path pattern

Selection of a pattern for a tool path is an important task in the tool path generation because different types of path pattern result in different total path lengths and different machined surface quality. We need to consider these two factors in the selection of the path pattern. The pattern should be selected to minimize the total path length as the total path length is proportional to the machining time. At the same time, the pattern of the path should produce best surface quality.

Two common path patterns that are usually used in tool path generations are “direction parallel” and “contour parallel”. In the direction parallel pattern, path segments are parallel to a predefined line, such as OX, OY or 45° line, which will result in an iso-planar tool path generation. In the contour parallel pattern, the tool path is created

from a set of offset curves of a cutter location surface patch. The contour tool path is usually generated using the iso-scallop method. Figure A6.8 illustrates the two tool path pattern approaches with two separated surface patch cutter locations. The left panel presents the parallel direction pattern (zig-zag approach) and he right panel shows the contour tool path approach (contour parallel).

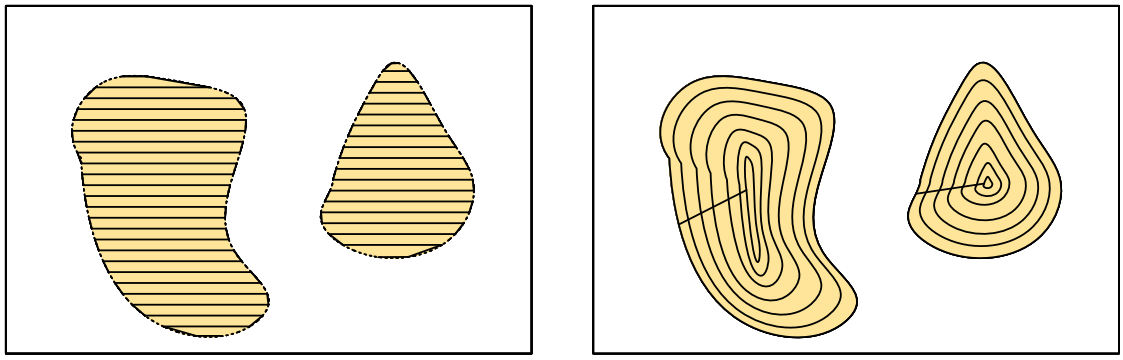


Figure A6.8 Two types of tool path, a) direction parallel pattern (zig-zag tool path), b) contour parallel (contour tool path)

After generating the path segments, each path segment needs to be connected to its neighboring segment to create a continuous tool path.

- In the case of the **direction parallel pattern**, there are two methods to connect the path segments, i.e. the zig-zag method and the one-way method. In the zig-zag connection method, each endpoint of a path segment is connected to the nearest endpoint of the next path segment. During a zig-zag milling process, a cutter approaches a workpiece in two cutting modes, i.e. conventional milling and climb milling, in a sequent manner. In the one-way connection method, a start endpoint of a path segment is connected to the stop endpoint of the next path segment. During the cutting process, the cutter approaches the workpiece by the conventional or climb milling. Comparing between the two connection methods, the zig-zag method offers a shorter tool path length, but it might exhibit a higher tool wear rate than the single mode tool approaching.
- In the case of the **contour parallel pattern**, after the cutter finishes the first loop, it moves to the next contour. The cutter usually starts at the smallest contour in pocket milling for inner surfaces and starts at the largest contour for

profiling of outer surfaces. During the cutting process, we can easily select the approaching mode (conventional or climb milling) without changing the total path length.

In summary, there are several steps to create a tool path from a cutter location surface patch, depending on the chosen path pattern:

- In the case of the **directional parallel pattern**, the steps are: *i*) selecting the cutting direction, *ii*) calculating the intersection of the cutter location surface with a set of parallel planes (the planes parallel with the cutting direction) to create path segments, *iii*) connecting the path segments to create a unique tool path based on the predefined connection method (zig-zag or one-way).
- In the case of the **contour parallel pattern**, the steps are: *i*) tracing the boundary of the cutter surface patch, *ii*) using the iso-scallop to offset the cutter location patch to create smaller cutter location patch, *iii*) repeating steps *i* and *ii* until the cutter location surface vanishes, *iv*) connecting the contours to create a unique tool path.

A6.7 Tool path simulation

This step is needed to confirm whether the created local tool path violates or hit obstacles (or other component features) during machining processes. There are three ingredients for the simulation processes, i.e. geometries of stock and target workpiece, the cutter mask representing the cutter and its holder, and the created tool path. Letting the cutter mask moving along the tool path on the surface of the stock workpiece. At any cutter' positions, the workpiece height data at any points that are higher than the corresponding tool mask, are replaced by the height of the cutter respectively.

To check the tool path violations or hitting obstacles, we will check whether there are any height changes outside the local machining positions. If the cutter violates the workpieces (undercutting material at non-defect regions), the whole tool path needs to be regenerated by reducing the cutter diameter. If the simulation process results that the tool path is correct, the tool path program will jump to the next steps for CNC machine code generation.

A6.8 Post process tool paths

After confirming that there was no tool violation during the cutting process, the tool path will be converted to G-code and added code to control a CNC machine. Most of CNC machines employ pre-buffer G-code interpolation which is also known as high-speed machining to keep constant feed rate when movement of the cutter in each command is very small such as in freeform surface machining to improve finished surface quality and reduce machine time.

In high-speed machining, the cutter can move with linear and circle interpolations. In machining of freeform surfaces, the CNC machines use linear interpolation to connect point-to-point in the surfaces. Command G1 (or G01) is for linear interpolation and is used to connect points in a contour/line. The command G1 is common for most of CNC machines. The cutter uses G1 to move from one contour to the next, if the distance from the current position to the nearest point of the next line/contour is smaller than twice of the width of cut (step over). Otherwise, the cutter will approach to new point from the Z direction. That means lifting the cutter in the Z direction to the safe level, moving the cutter to the new position of the new point (at safe Z level), moving down the tool to the new points.

As the tool path only uses standard linear interpolation, a G-code program will finish after appending a header and footer codes to control CNC machines. Completed program is saved and sent to a CNC machine station to execute and perform the re-profiling.