The University Of Sheffield.

A multi-step learning approach for in-process monitoring of depth-of-cuts in robotic countersinking operations

By:

Mateo Leco

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Sheffield

Advanced Manufacturing Research Centre with Boeing

January 2020

I ACKNOWLEDGMENTS

This project has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ for research, technological development and demonstration under REA grant agreement no 608022. The project was part of a Marie Skłodowska-Curie Initial Training Network (ITN) with title Sustainable Manufacturing through Advanced Robotics Training in Europe, acronym SMART-E (http://smart-e-mariecurie.eu/).

First of all, I'd like to thank my supervisors Prof Sam Turner, Dr Tom McLeay and Dr David Curtis who, at different times, have all accompanied me during this journey. Sam for offering me this opportunity and initial support, Tom for the guidance and help in clarifying my doubts and Dave for his support in the last months of writing.

I'd also like to thank Ben Morgan for his continued support, especially in the last period of writing this thesis. I'd like to extend the thanks to Adrian Hirst and Tom Smith, my colleagues at Factory 2050 who helped me for the setup of the experimental works. All my other friends at IMG that made these four years seem light and enjoyable.

I'd like to thank the ESR fellow friends from SMART-E for all the fun and happy moments spent during summer schools and other collaboration activities.

I'd like to thank my parents and my sister, who although far from where I live, have always been present in every step of my life. Thank you for your continuous belief in me.

Finally, I'd like to thank my wife Claudia for her constant support and tireless patience for working long night hours, and for her motivational boost in difficult times of this journey.

II ABSTRACT

Robotic machining is a relatively new and promising technology that aims to substitute the conventional approach of Computer Numeric Control machine tools. Due to the low positional accuracy and variable stiffness of the industrial robots, the machining operations performed by robotic systems are subject to variations in the quality of the finished product. The main focus of this work is to provide a means of improving the performance of a robotic machining process by the use of in-process monitoring of key process variables that directly influence the quality of the machined part. To this end, an intelligent monitoring system is designed, which uses sensor signals collected during machining to predict the amount of errors that the robotic system introduces into the manufacturing process in terms of imperfections of the finished product. A multi-step learning procedure that allows training of process models to take place during normal operation of the process is proposed. Moreover, applying an iterative probabilistic approach, these models are able to estimate, given the current training dataset, whether the prediction is likely to be correct and further training data is requested if necessary. The proposed monitoring system was tested in a robotic countersinking experiment for the in-process prediction of the countersink depth-of-cut and the results showed good ability of the models to provide accurate and reliable predictions.

III TABLE OF CONTENTS

IV LIST OF FIGURES

V LIST OF TABLES

VI NOMENCLATURE

AE	Acoustic Emission
AI	Artificial Intelligence
AMRC	Advanced Manufacturing Research Centre
ANN	Artificial Neural Network
ARD	Automatic Relevance Determination
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CFRP	Carbon Fibre Reinforced Polymers
CFS	Correlation-based Feature Selection
CNC	Computer Numerical Control
CV	Cross Validation
DAQ	Data Acquisition
DoC	Depth-of-Cut
DOF	Degrees of Freedom
DWT	Discrete Wavelet Transform
ECFS	Enhanced Correlation-based Feature Selection
ELM	Extreme Learning Machine
FFT	Fast Fourier Transform
GP	Gaussian Processes
GPLVM	Gaussian Process Latent Variable Model
GPR	Gaussian Process Regression
ILC	Iterative Learning Control
IMS	Intelligent Monitoring System
KPI	Key Performance Indicator
LOO-CV	Leave-One-Out Cross Validation
LPF	Low-Pass Filter
MI	Mutual Information
ML	Machine Learning
MSE	Mean Squared Error
PCA	Principal Component Analysis
PF	Pressure Foot
PLC	Programmable Logic Controller

PSD	Power Spectral Density
RMS	Root Mean Square
RPM	Rotation Per Minute
SE	Squared Exponential
SME	Small and Medium-size Enterprise
std	Standard deviation
STFT	Short-Time Fourier Transform
SU	Symmetrical Uncertainty
SVM	Support Vector Machine
TCM	Tool Condition Monitoring
TCP	Tool Centre Point
TFF	Time-Frequency Filter
TMAC	Tool Monitoring Adaptive Control
TPF	Tool Passing Frequency
WPD	Wavelet Packet Decomposition
WT	Wavelet Transform

1 INTRODUCTION

Nowadays, modern industries rely extensively on robots for a wide range of applications including object manipulation and transportation, precision assembly, welding and painting. Even though the number of robot units applied in industry is continuously growing, a few of them are exploited in machining applications [1, 2]. Traditionally, the standard approach is to use multi-axis Computer Numerical Control (CNC) machines to perform operations such as drilling, turning and milling. Due to their high rigidity and positioning accuracy, these machines can achieve very high accuracies and obtain parts with relatively complex geometries.

On the other hand, robots offer flexibility, high dexterity, large working envelopes, ease of programming and relatively lower costs, which are all highly desired features that can suggest the exploitation of robotic systems as an alternative of CNC machines in the factory of the future. Yet, robotic machining is still in its infancy because of several issues it faces [1-3]. The majority of the machining operations that are successfully performed by robots involve finishing applications such as polishing, grinding and deburring. Articulated multi-joint industrial robots have lower positioning accuracy and repeatability comparing to the CNC machines and, most importantly, they have limited stiffness, which hinders their use in machining operations where large process forces are involved. Tool deflections and vibrations are typically induced and the quality of the finished product is compromised. Further research is therefore required in order to promote the use of this technology.

The recent trend towards the vision of the fourth industrial revolution encourages the use of information technology in manufacturing [4]. Therefore conventional machining research is focused on the development of intelligent ways of exploiting information for process monitoring. Sensory data collected during the cutting process of a typical machining operation is used to infer several aspects of the process, such as the condition of the machine, cutting tool wear and breakage, workpiece surface roughness and imperfections as well as diagnosis of common process failures. The use of these intelligent monitoring systems has clearly impacted the reliability and robustness of the machining processes.

1.1 Motivation

Several solutions for enabling robotic machining have been published in the literature including the development of specific Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) software packages [5], incorporation of a high dynamics compensation mechanism [6, 7], implementation of adaptive position and/or force control strategies [8-10] and effective path planning considering the robot stiffness at the particular configuration [11]. In some cases [12, 13], dual robot cells were applied for rapid prototyping milling applications, where softer, non-metallic materials such as plastic, wood, wax or ice blocks were used. In general, good results were reported in dealing with the issues of robotic systems, indicating that the technology is rapidly advancing.

However, all these solutions have focused exclusively in enabling the robotic machining technology by directly addressing the problem from the robotic structure perspective and no attempt was made to develop a dedicated monitoring system for the actual machining process. In contrast, monitoring systems have been extensively applied in conventional machining processes in a range of operations and monitoring scopes [14-19]. This lack of specifically designed monitoring systems for robotic machining processes is due to the fact that the technology itself is relatively new compared to the machine tools counterpart and, as it advances further, new intelligent ways of process monitoring will be required in order to confidently exploit industrial robots in material cutting operations.

This thesis provides an innovative method to integrate both research fields of robotic machining and intelligent monitoring systems in an attempt to further advance this developing technology.

In terms of the industrial motivation, process monitoring could be applied for in-process prediction of the quality of the machined part, which is extremely useful in a production environment, where an inspection step is typically required after the machining operation. This inspection is either included in the process operational cycle (automatic in-process inspection) or often it is performed (manually or automatically) at the end of the whole process using specific equipment. In either case, the process down time inevitably increases due to the time spent for the inspection task and/or delays associated with the preparation of the workpiece for inspection. Given the industrial focus on high productivity rates, it is essential therefore that this inspection time is kept at minimum. In order to further increase productivity without compromising the overall process quality, a method that indirectly

infers the quality of the poduct (i.e. the result of the inspection step) from in-process information rather than the post-process direct measurements, is required. The method proposed by this thesis aims to substitute the inspection step by predicting the quality of the finished part from sensory data collected during machining.

1.2 Research Aim and Objectives

This work aims to design and test an intelligent monitoring system which can improve the performance of robotic machining processes by online prediction of key process variables that directly influence the quality of the machined part. The proposed system will use sensor signals collected during machining to predict the amount of errors that the robotic structure introduces into the manufacturing process in terms of variations in the final product shape.

In particular, this work will improve the performance of a typical robotic machining process in the following ways:

- Provide additional in-process feedback on the quality of the machined part, which will allow to avoid the quality inspection step that is usually performed post-process. This will improve the overall process time.
- Potential use of the in-process predictions to drive corrective actions (manually or automatically). This will increase the probability of having final products with quality within the specified tolerances.

In order to deliver the above overall aim, this project had the following four objectives:

1. Setup of a suitable data acquisition system and signal processing scheme for in-process extraction of the information that reflects the machining process. This includes both the hardware and software aspects of the system.
2. Develop a method for selection of the relevant information (with respect to the process output variable) from all the collected sensor signals.
3. Establish a set of process models and train them during normal operation of the machining process, without any interruptions. This will allow the monitoring system to be applied in a production environment with minimum disruption of the process.
4. Provide a means of assessing the reliability of the process models. This includes having a measure of the confidence of the models in the predicted outputs.

1.3 Thesis Outline

This thesis presents a novel approach to improve the performance of robotic machining processes by intelligent monitoring of key process variables that directly influence the quality of the machined part. A summary of the thesis is presented below.

Chapter 1 introduces the scope, motivation, research aim and objectives of this work. An extensive literature review of the two main research areas of robotic machining and intelligent monitoring systems for machining processes follows in Chapter 2. The elements of an intelligent monitoring system are described in detail including the sensor system, digital signal analysis, feature selection methods and machine learning techniques.

The robotic machining case study used in this work is described in Chapter 3. The system was applied into a multi-robot countersinking process for the monitoring of the obtained countersink depths in composite aircraft components. The robotic cell consisted of two industrial robots that work together to automatically countersink previously drilled holes in Carbon Fibre Reinforced Polymers (CFRP) panels.

Chapter 4 defines the case study problem reflecting on the current limits of the considered robotic countersinking technology and presents the research methodology of this work. A multi-step process method is proposed to improve the performance of the robotic countersinking process. The method integrates concepts of intelligent monitoring systems such as signal processing, feature extraction and machine learning techniques into the particular robotic machining context, addressing the current gap in the number of applications that combine these two research areas. The requirements of the monitoring system and the design considerations for its applicability in a production environment are also outlined in this chapter.

Chapter 5 presents the details of the experiment conducted by this work, covering the data acquisition system and the signal processing scheme applied to extract in-process information from the machining process. The experiment was designed to apply the proposed multi-step process method for testing and validation of the system. Non-intrusive and easy to install sensors were used to extract the information from the machining process, including three single-axis accelerometers, a power transducer and an acoustic emission sensor. A central data acquisition unit was applied to ensure synchronisation of all the acquired signals. The raw signals were filtered, segmented and processed through advanced feature extraction techniques for further analysis. The methods used for signal processing

are well-established advanced techniques in the process monitoring research area, such as digital filtering and signal domain transformations (time, frequency and joint time-frequency analysis). In this work, the signal processing scheme was performed automatically, without any manual intervention.

The supervised learning method applied to the dataset obtained from the experiment designed by this work is presented in Chapter 6. Process models that can predict the countersink depth from the extracted sensory information were built and tested in a regression problem. In particular, a probabilistic learning approach based on Gaussian Process Regression (GPR) was used for this purpose. Another aspect of the monitoring system considered in this study is the selection of relevant signal features with respect to the process output variable. A GPR model that implements an Automatic Relevance Determination (ARD) in the training optimisation procedure was applied to select the relevant signal features. This method, although known in the machine learning literature, is entirely unexplored in the process monitoring context.

Having covered the technical aspects of the system and performed an extensive offline analysis of the proposed process models, Chapter 7 moves on to present an online (in-process) version of the system to be used in the considered robotic countersinking process. A multi-step learning procedure is proposed to build and train process models during normal operation of the machining process, without any down time for collection of training data. This method was tested using the data collected from the experiment, simulating the online execution of the process. A way of assessing the model reliability was also investigated in this study by exploiting the probabilistic framework of the GPR models. A threshold value on the standard deviation of the prediction (provided as an output of the models) was applied in order to identify the input data where the models were likely to have large prediction errors.

Finally, the thesis conclusions and further work are discussed in Chapter 8 and Chapter 9.

The theoretical framework of Gaussian Processes is presented in the thesis Appendices A.1.

2 LITERATURE REVIEW

This chapter provides an overview of the recent advances in robotic machining research as well as an extensive review of the published literature in the field of Intelligent Monitoring Systems (IMS) for machining processes. The main elements of an IMS are described in detail including the sensor system, digital signal analysis, feature selection methods and machine learning techniques. The current research gap of the above two areas is then analysed in order to highlight the contribution of this work. The chapter concludes with a summary of the reviewed material.

2.1 Advances in Robotic Machining

The term robotic machining refers to a manufacturing operation in which the cutting process of raw material is controlled by means of one or more industrial robots. This differs from the conventional machining approach which uses multi-axis Computer Numeric Control (CNC) machines to accurately control the cutting tool path during the material-removal process. The increasing trend towards robotics and automation over the last decades, has motivated industries and researchers to look into robotic machining as a flexible and relatively low-cost alternative that will advance further the traditional machining approach.

Some of the most desirable characteristics and potential advantages of robotic machining technology against conventional CNC machining are as follows:

1. **Higher flexibility:** Robots are very flexible both in tool positioning within the working area and the range of operations that they are able to perform due to the large number of Degrees-of-Freedom (DOF) and the ease of programming, respectively. This is a big advantage of robotic structures over CNC machine tools. An articulated robotic arm can carry out operations in multiple axes which enables the production of complex parts with great detail and also it can easily switch between different spindles and cutting tools if required by the specific machining task. On the contrary, CNC machines have limited flexibility in the range of operations. If, for example, a specific part requires several operations with different procedures and cutting tools, then there are often required different machines to perform all the various tasks.
2. **Larger working area:** In general the robots' wide working envelope allows for machining of bigger parts such as those typically used in aerospace industry. Iglesias

et al. [20] reported a working volume of 7-8 m^3 for a standard industrial robot. Moreover, this can be further increased if external linear or rotatory axes are used. The machine tools, on the other hand, have limited working area. For instance, if the workpiece is too big to fit into the machine's available space, then the part needs to be broken apart in smaller pieces which are usually machined separately and reassembled after completion. In some extreme cases, as noted by Pandremenos et al. [2], in order to accommodate larger parts, the machine itself needs to be modified, which could lead in a very expensive solution.

3. **Lower costs:** In order to sustain the current demand and still remain competitive in the machining industry, manufacturers are looking for ways to increase production at a lower cost. In this context, robotic machining has great potential. According to DePree and Gesswein [21] the estimated cost reduction of a comparable robotic solution is between 35% and 50% with respect to the conventional approach. This allows less capital to be invested or an increase in production by adding extra machining stations for example.
4. **Robotic machining lines:** The concept of discrete processes with individual robots rather than operational roll up on a single machine tool platform exploits in full the capabilities of industrial robots within manufacturing. This idea of a robotic machining line similar to those seen in factory assembly lines, where multiple robots perform different stages on one part (a dedicated robot for each operation such as loading, rough machining, finishing), was proposed by Chen and Dong [1] as part of the future research issues to address in order to advance the current technology.

Motivated by the enormous potential of robotic machining, research in this area is actively growing. According to Chen and Dong [1], the first studies of industrial robots performing machining-related operations started in the 1990s and, since then, a large body of literature has investigated the development, application areas and the technical challenges/limitations of this technology.

Pandremenos et al. [2] argued that despite the prominent advantages of robotic machining in terms of both higher flexibility and cost efficiency compared to CNC machines, industrial robots have some serious limitations related to their positional accuracy and repeatability. These main issues make them not suitable to perform all of the machining tasks, especially when hard materials are involved. However, the authors pointed out that industrial robots, due to their extra DOF and large working envelope, are an excellent alternative to rapid

prototyping machines which generally use softer materials (wood, wax, plastic) and require lower accuracies. Consequently, a number of successful applications are found in the literature involving robots in rapid prototyping [12, 22, 23] and surface finishing processes [24, 25] mainly polishing, grinding and deburring. Chen and Dong [1] reported that a 6-DoF robotic arm can outperform a three-axis CNC machine in polishing operations due to the robot's position redundancy, which allows for the polishing tool to be oriented and maintained always normal to the surface of the workpiece generating smoother results. Similarly, Iglesias et al. [20] established that industrial robots are currently considered as a valid alternative of CNC machines for a range of machining tasks, subject to the hardness level of the material. Most non-metallic materials such as wood, plastic, etc., can be machined by robotic cells in any of the production stages (prototyping, intermediate or final product), whereas, with the increasing degree of hardness, the robotic structure starts suffering from the higher level of forces generated. In such cases, the use of robots is limited mainly to the prototyping stage, pre-machining at lower tolerances and/or surface finishing processes that require relatively lower level of accuracy.

The limitations of robotic machining have been extensively discussed in literature by a large number of researchers [26-35]. Therefore, it has been possible to identify most of the current challenges this technology faces. It is now well established that the major issue of an articulated robotic arm is its inherent lower static and dynamic stiffness that, in comparison with a machine tool, is about 50 times lower (less than 1 N/µm against the 50 N/µm or greater of CNC machines, as reported in Pan et al. [36]) due to the serial kinematic structure. Moreover, the overall stiffness of the robotic arm is subject to vary considerably with the specific configuration (joints values), the payload installed on the end-effector and the different cutting conditions. This fact becomes even more apparent when machining hard materials because of the vibration (chatter) which occurs during the cutting process as result of the robot's lower resistance to the generated forces. Consequently, the overall product geometry and surface quality is compromised. Some other issues reported by the researchers include the relatively poor positional accuracy and repeatability of the robots which makes it difficult to achieve part geometries of high quality, especially in different cutting conditions; the lack of programming standards in terms of robotic machining simulation software and path generating language such as the G-Code equivalent of CNC machines. In addition, Barnfather et al. [37] pointed out that typical issues found in conventional machining such as tool deflection, tool wear and gear backlash are generally more severe in robotic systems due to the differences in the kinematic structure. Moreover, the thermal variations and the

robots' joint flexibility under different payloads influence the overall positioning accuracy and therefore they need to be taken into account as well. Dumas et al. [31] identified the robot joint stiffness as an important factor that influences the part quality and presented a method to compute the joint stiffness values of any industrial robot. The authors claim that such information is generally missing in the robot specifications or it is difficult to obtain by the robot manufacturer. According to Pan et al. [36], the main issue of robotic machining is due to the chatter (vibration) caused by the low structural stiffness and base natural frequency (typically around 10 Hz) of the industrial robots. The authors referred to it as coupling chatter given its low frequency content which is in the range of the robot's natural frequency.

It is clear, therefore, that in order to adopt this prominent technology in a larger scale, all the above challenges need to be addressed through constant research and development of new methods to deal with the errors introduced by the robotic structure. Some of the main technological and research improvement steps are discussed in the following sub-sections.

2.1.1 Steps towards Robotic Machining

Motivated by the size of potential market, big robot enterprises such as ABB, KUKA, FANUC, COMAU and STÄUBLI have started developing industrial robots specifically designed for machining applications with higher stiffness values and improved position accuracy and repeatability. Some examples include the ABB IRB 6660, KUKA KR 500-MT, FANUC F200iB, COMAU NJ-130-2.6 and STÄUBLI RX160 HSM. In addition, software packages to simulate the robot trajectory during the machining process are also provided as part of the robotic solution. The evolution of CAD/CAM tools has greatly reduced the development time and facilitated the transition from simulation to real-world applications. Such tools give the possibility to the operator to generate the robot Tool Centre Point (TCP) path based on the CAD information of the workpiece, similar to the way it is computed in standard CNC machines. Therefore, the process is first studied in a virtual simulator to ensure that the robot path is feasible and factors such as the axis limits, travel time, singularity positions or collision with other parts of the cell are taken into consideration. Examples of software packages often mentioned in the literature are the AUTODESK (previously DELCAM) PowerMILL Robotic Interface, KUKA CAMRob (now KUKA.CNC), ABB RobotWare Machining FC and FANUC RoboGuide.

In terms of the current research developments, there has been a large number of published studies describing new methods to address most robotic machining issues. The main categories comprise: (i) robot calibration including identification of robot joint stiffness values, (ii) development of a stiffness map for robot posture optimisation, (iii) chatter avoidance methods and (iv) compensation strategies for cutting tool deflection.

Robot calibration is crucial in machining applications given the importance of improving the robot's positional accuracy when controlling the cutting tool. Moreover, the forces acting on the robot during the cutting operation can cause deflections of TCP up to 2 mm from the planned trajectory, as reported by Denkena and Lepper [26]. This calibration, which is often referred to as non-kinematic calibration, considers all the errors caused by thermal effects, compliance of the robot's links and joints, compliance in the joints bearings, gears friction and backlash as well as dynamic effects. Work in this area has been conducted by Abele et al. [35], Dumas et al. [31] and Yang et al. [38]. Some other studies (Bu et al. [28], Guo et al. [39] and Lin et al. [40]) focused on the analysis and creation of a stiffness map of the robot's workspace. This information was then used to identify the best possible configuration of the robot's joints for performing the required machining task. In terms of chatter avoidance methods, Pan et al. [36] extensively studied the vibration characteristics in a robotic milling process and devised a method to minimise the occurrence of coupling chatter.

2.1.2 Compensation Strategies for Tool Deflection

The research focused on the development of compensation methods for tool deflection is of particular interest for this thesis work, especially those that use data driven process models based on machine learning algorithms.

Reinl et al. [41] presented a model-based off-line method to predict and compensate the robot path deviation caused by the process forces in milling operations. The proposed approach consisted in coupling an extended dynamic model of the robot (including joint elasticities and tilting effects) with a milling forces simulation model. The result was an entirely off-line simulation-based compensation strategy that predicts the path errors in advance and uses the predictions to correct the robot path before the actual cutting occurs.

Denkena and Lepper [26] reported a tool deflection compensation approach to deal with the positioning errors of the robot TCP during machining of large frame parts for the aerospace industry. The authors designed a sensing spindle holder that measures the forces acting on

the spindle during milling operation and built a static deflection model of the industrial robot based on these measurements. The model could then be used for selection of the optimal path to perform the required operation. The ultimate goal of the method was to combine the compliance model of the robot with the force measurement to obtain an online compensation of the displacement of the cutting tool during machining. However, no results of the actual compensation approach were presented in the paper, which was mainly focused on the development and calibration of the force sensor. The authors referred to further research for the development of the compensation method.

Furtado et al. [32] presented a fully experimental approach to improve the position accuracy of a 6-DOF industrial robot during a milling operation. Their method consisted of a set of five experiments, each one used to optimise a particular aspect of the overall process. The compensation strategy was based on the surface waviness measurement of previously machined workpieces. The inverse profile was then calculated for compensation of the robot trajectory. The method seemed intuitive, however it was long and involved too many experiments for the setup phase. Therefore, it could be applicable in manufacturing of products with low variety and high volume. Also, the approach neglected other factors that could influence the process accuracy after the initial setup was done, such as the material variability of the parts and the cutting tool wear. Moreover, it would require a new set of experiments to find the new best configuration if, for example, a few small changes of the overall process were needed.

In order to improve robotic machining performance, Pan and Zhang [8] proposed a solution based on robot stiffness modelling and active force control strategy for real-time compensation of cutting tool errors. The identification of the robot stiffness parameters was done experimentally by applying a known load on the tool tip and then measuring its deformation. The compensation approach used force measurements to control the robot joints configuration through a closed loop scheme that was implemented in the industrial robot controller, leading to a hybrid position/force control platform. The experimental results showed a higher surface quality, however, the method required direct access to the robot controller.

In a similar direction, as part of the COMET project [42], Schneider et al. [43] proposed a 3D piezo-actuated high-dynamic compensation mechanism to use for robotic milling applications. The authors presented an integrated solution which considered the robot holding the workpiece, whereas the machining spindle and cutting tool was installed on the

compensation unit, mounted on the ground. An optical tracking system was used to measure the robot deviations during milling. Based on this information the robot and the compensation unit contributed jointly in the compensation process handling both the low and high frequency errors, respectively. A closed-loop control scheme that adapts the spindle position was implemented in order to increase the machining performance.

An interesting use of machine learning techniques as a compensation method to improve the robot's absolute positional accuracy for robotic drilling applications was reported by Yuan et al. [44]. The authors used the robot's theoretical position as input and the actual positional error measured by a high-precision laser tracker as output to train and build an error prediction model, based on Artificial Neural Networks (ANN). In particular, a single hidden layer forward network was used and the training was performed by the Extreme Learning Machine (ELM) algorithm, originally proposed by Huang et al. [45]. In order to obtain the training dataset, 2000 random points were evenly distributed within the robot workspace and the actual robot TCP position was measured with the laser tracker. A series of experiments were then performed to determine the optimal model parameters in terms of the required number of training samples and nodes in the hidden layer. The approach showed an improvement of the robot absolute position accuracy within its workspace by 75.89% for the average values and 80.93% for maximum values of positional errors, however, this work did not consider the influence of machining forces on the robot end-effector, which will introduce further positional deviations from the theoretical values.

Marchal et al. [46] investigated the use of Iterative Learning Control (ILC) in robotic milling processes and proposed two different ILC algorithms in order to reduce the positional errors in robotic machining tasks. The first approach was based on accurate position measurements taken by an optical laser tracker, whereas the second relied on force/torque data measured on the robot end-effector during a milling operation. The authors applied data-driven system identification methods to fit the data measured during the experiments and used these models for the iterative procedure. The experimental setup included a fixed spindle with the milling tool and the robot holding the workpiece and moving according to the process path. Both methods successfully reduced the robot position errors by 85% and 50% respectively, with the first one clearly outperforming the second due to the fact that a direct measurement of the path deviation was obtained. In contrast, the second approach used the derived compliance model of the robot to estimate the position deviations from the force measurements, with the advantage of being a cost-effective and practical solution. It was

noted that the force-based ILC algorithm was not able to compensate for constant offsets and low-frequency deviations resulting, for example, from kinematic calibration errors. The authors suggested the application of other stiffness calibration methods to complement the proposed system identification approach. Another drawback of the procedure was the fact that considering its repetitive pattern, the algorithm could not compensate for individual variations of the workpieces or non-repetitive errors introduced by tool wear.

In conclusion, the vast amount of research carried out in the field of robot machining has enabled this new technology to be applied as a flexible and cost-effective alternative of standard CNC machine tools in a number of applications where the level of precision required can meet the accuracy limitations of the robots. Still, there is continuous research going on in order to find new methods for improving the process performance and this thesis will explore the use of intelligent monitoring systems in the attempt to do so.

2.2 Intelligent Monitoring Systems

Nowadays, manufacturing enterprises are looking into automation as an effective solution in order to cope with the growing demand of increased product quality, greater variability and shorter lifecycles, reduced cost and global competition. Considering the value of machining in most industries, the automation of machining processes is of critical importance in the manufacturing sector, which in turn requires the development of reliable and robust monitoring systems [14]. The ability to collect useful data from the machining process and then apply this information to make appropriate decisions based on the condition of the machine and/or cutting tool is the main drive for this technology. In particular, manufacturing processes such as milling, drilling and turning require a lot of capital investment in equipment and machines, and hence it is crucial to detect and avoid any failures. Moreover, the utilisation of the cutting tool within its actual life span by continuous tool wear monitoring could bring further revenue and add value to the manufacturing process.

A considerable amount of literature has been published in the last 25 years on monitoring systems for machining processes, mostly focused in Tool Conditioning Monitoring (TCM). As reported by Teti et al. [16], only in the time frame between 1996–2006, about 500 new publications were added to the existing TCM literature database [47], bringing a total of more than 1000 cited references. The authors presented a comprehensive review of advanced monitoring techniques applied in machining operations breaking down the discussion in the current developments of sensor systems technologies, advanced signal processing methods,

monitoring scopes, decision making paradigms and application areas. Liang et al. [14] observed that despite the successful application of these monitoring systems in laboratory and field testing, there was still little availability in production environment. This was attributed to a number of reasons such as the current lack of robust sensor hardware and monitoring algorithms, embedded sensor and actuators, coordination in the research community, standardisation in automation and communication protocols.

In terms of monitoring scopes and applications, a variety of them has been reported in the current process monitoring literature. The vast majority of the research studies was focused on identification of the tool wear state or tool breakage during a machining process [48-52]. Others considered the surface integrity of the machined part including surface roughness [53-55] and surface anomalies [56-58]. Some researchers investigated methods on chatter detection [59-61], whereas a few publications presented monitoring systems for chip formation [62, 63] and machine condition [64].

The Intelligent Monitoring Systems (IMS) were reviewed by Abellan-Nebot and Subiron [15], where particular attention was given to the Artificial Intelligence (AI) approaches used in the existing literature for the creation of process models that can learn from the data and incorporate this knowledge in the decision making procedure. The Artificial Neural Networks (ANN) were identified as the main AI technique used for process modelling purposes in machining research, followed by the fuzzy logic systems, the neuro-fuzzy inference, which is a hybridisation of the previous two and finally, monitoring systems based on Bayesian networks. Other approaches such as hidden Markov models, evolutionary algorithms and Support Vector Machines (SVM) were believed to become more popular in future research. Similarly, Worden et al. [17] added to the above list the use of Gaussian Processes (GP), graphical models and deep belief networks.

According to Abellan-Nebot and Subiron [15], despite the large body of research conducted in the IMS field, a clear methodology in the development of machining monitoring systems is still lacking. Instead, the design of the monitoring system is strictly dictated by the specific machining application. The selection of the sensors and the hardware used for data acquisition, the signal processing techniques, the data fusion and decision making schemes, are all dependent on the characteristics of the particular process under investigation. The authors proposed a generic methodology which encloses the main elements of the monitoring system and it seems to be consistent with those suggested by other papers

reviewed during this work. This methodology is also adopted in the research conducted here and it is composed of the following five design steps (from [15], see also Figure 2.1):

1. Sensors
2. Signal processing
3. Feature extraction
4. Feature selection
5. Model selection and diagnosis

Figure 2.1 illustrates the design steps of the IMS implemented in this study. Indication of the technologies/methods related to each element is also provided in the figure.

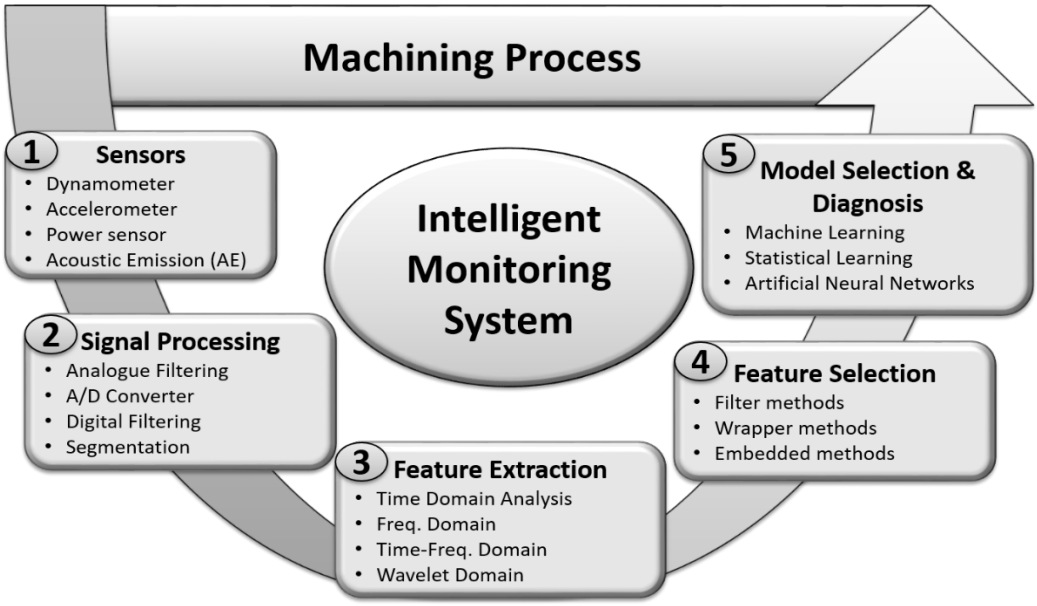

Figure 2.1 Intelligent Monitoring System Design (after [15]).

The first step of an IMS is the selection of sensors which convert the physical quantities generated during machining into the corresponding electrical signals. The choice of the sensors system is generally dependent on the particular application and monitoring purpose, however, other factors such as the cost of the equipment and its intrusive nature in the machining process should be taken into consideration. Typical sensors choices reported in the literature include dynamometers, power transducers, accelerometers, thermistors and Acoustic Emission (AE) sensors.

The next step of signal processing is specific to the type of the acquired signal, however, it usually includes amplification and filtering, Analogue-Digital (A/D) conversion and digital

signal processing. In some cases signal segmentation is required to extract the relevant parts of the signal data for further analysis.

If the raw signal is still not suitable to use due to the high size of the data, feature extraction is performed to make all the necessary transformations for data size reduction. This step generates what are typically referred to as signal features or descriptors, which reduce significantly the size of the initial data set. The transformations often include analysis of the acquired signal in time, frequency, joint time-frequency and/or wavelet domain.

The extracted features can be either related to the process performance variables or not related at all, therefore it is important to identify those that describe adequately the machining process. Hence the reason of the feature selection step: it will further reduce data size by filtering out the features that are not related to the process. This in turn reduces the complexity of the process model and the computational burden of the next step, potentially improving the process model accuracy.

Finally, the selected features are used as input variables to train and build process models based on Machine Learning (ML) methods, which can learn the complex non-linear relationship between the signal features and the process variables of interest. The choice of the ML approach or statistical learning method is crucial in order to make accurate decisions on the process performance variables.

A more detailed, in-depth presentation of all the above design steps and the relevant published literature is given in the following sub-sections.

2.2.1 Sensors

Sensor technologies have been extensively discussed by almost all the papers found in the machining process monitoring literature. According to Teti et al. [16], the sensing techniques for the monitoring of machining processes are generally classified in two different types: direct and indirect measurements. The direct technique refers to the actual quantity of the variable being measured, for instance, tool wear can be directly measured by cameras for visual inspection, microscopes, electrical resistance, etc. These methods are very accurate but also very expensive and thus they are employed mostly in research laboratories for verification and investigation purposes. Also, direct measurements have some practical limitation when it comes to applying them into the shop floor, the majority of which are caused by access problems during machining, illumination and use of the cutting fluid. The

indirect technique, on the other side, uses the measurements of auxiliary quantities such as the spindle power or cutting force and then infers the actual quantity via empirically determined correlations. Indirect methods are less accurate than the direct ones, but they are also less complex, less expensive and easy to apply in practical application. Therefore, indirect methods are the most used in industry.

Lauro et al. [19] analysed a range of sensors commonly used in machining processes for measurement of quantities such as cutting forces, vibration, temperatures, sound, Acoustic Emission (AE) and ultrasonic waves. Liang et al. [65] extended the above sensor list with vision, power, torque, dimensional gauges and micro magnetic measurements. Abellan-Nebot and Subiron [15] noted that the majority of applications used the following four main sensors: dynamometers, accelerometers, AE and current/power sensor. In addition, the authors presented a table, as shown in Table 2.1, where each sensor technology was scored based on its cost, intrusive nature and signal reliability for machining modelling purposes.

Table 2.1 Sensor technologies ranked by Abellan-Nebot and Subiron [15].

Sensor	Cost	Intrusive nature	Signal reliability
Dynamometer	***	***	***
Accelerometer	**	**	**
AE	**	**	**
Current/Power	*	*	*

It is important to note that the cost of the equipment as well as its intrusive nature with the machining process are crucial factors to consider in production environments. An example here would be the dynamometer (see Table 2.1). Equipment that provides accurate force measurements is very expensive and it also requires installation in the vicinity of the spindle or workpiece, which, despite the high reliability of the signal, makes the dynamometer not suitable for use in production.

Some of the common sensor technologies used in process monitoring research are briefly reviewed in the following paragraphs.

2.2.1.1 Cutting Force

Cutting force measurements are very popular in research environments for monitoring of machining processes due to the fact that they are directly related to the amount of material

being removed from the workpiece. Higher level of forces measured during the cutting process indicate higher material removal rates. Dynamometers with three piezoelectric sensing elements (one for each direction X, Y and Z) are commonly used for force measurements. An electrical charge proportional to the load in the measurement direction is obtained by the sensing element. In machining literature, the information provided by force measurements has been used for monitoring of tool wear [50, 51, 66-68], tool breakage [48, 49], chip conditioning [48, 49] and surface roughness [57, 58, 69]. The detection of tool wear can be observed with force measurement as it occurs due to the progressive increase of friction forces. Whereas, a tool breakage will cause first an instant spike and then a drop of the cutting force signal. As mentioned above this sensing technology is being successfully applied in research environments, however, the high cost of the equipment for accurate measurements and, most importantly, the intrusive nature in the vicinity of the machining process limit its utilisation in production environments.

2.2.1.2 Spindle Power and Current

Unlike with force sensors, power and current measurements of the spindle drive are seen as a relatively low-cost and easy to install sensing technology, which provides a signal that is proportional to the cutting force. Therefore, several studies [48-50, 52, 70-72] reported the use of spindle power sensors, often as an indirect measurement of force or torque, for monitoring of machining processes. To obtain the power measurements, there is no need to interfere with the cutting area since the signal is usually available from the spindle drive controller and requires just a way to read it directly from there. This fact and also the relatively low cost of the equipment, make such sensors the ideal choice to use production environments. However, using measurement of power as an indirect force measurement can suffer significant electrical and mechanical noise therefore it is not as accurate as a direct force measurement. Moreover, the power signal has a poor dynamic response and it is less sensitive to small changes in the cutting process. The amount of power required for the actual cutting process can be a portion of the total spindle power and, as noted by Teti et al. [16], other factors such as the temperature of the machine, the axis condition and the lubrication state of the drive motors should be taken into consideration as they can influence the power consumption of the spindle during machining.

2.2.1.3 Vibration

Vibration measurements are widely used in both research and industry environments with a large number of published papers [48-52, 59, 60, 62, 66, 72, 73]. The measurements are usually taken by single or multi-axes accelerometers attached to the fixture, workpiece, the cutting tool or spindle surface. A piezoelectric material pre-loaded with a mass is used to produce an electric charge proportional to the acceleration of the mass in either direction (compression or expansion). Small displacement of the cutting tool due to vibration influences the quality of the surface of the machined part, therefore, vibration measurements are mainly applied for monitoring and prediction of surface roughness [54, 56, 62, 74, 75], but also for tool condition monitoring [50, 52, 66, 73] and chatter detection [59, 60]. Tool wear can be observed from the continuous increase of the contact area between the tool and the workpiece, which reduces the vibration due to the higher frictional damping. However, when the tool is completely worn or tool breakage occurs, the spindle vibration will increase as result of the large cutting forces [15]. The amplitude of the signal is also highly sensitive to the distance of the sensor from the cutting tool. It is therefore important to place the sensor in proximity of the cutting area and keep its distance as consistent as possible during the cutting process. This is often difficult to obtain in production and it can be an issue for industrial applications, especially for large part manufacture. Vibration measurements are not as accurate as force measurements, but they are relatively cheaper, easy to install and less intrusive.

2.2.1.4 Acoustic Emission (AE)

Measurements of AE in the ultrasonic frequency range generated during the cutting process have had a lot of interest in the current literature. The AE signal is typically used for diagnosis of tool wear [50, 51, 57, 58, 62, 69, 76, 77] and tool breakage detection [48, 49, 73]. An AE is defined as a stress wave that travel through a material as result of a rapid release of energy from a localised source [77]. Therefore, an AE sensor converts the mechanical energy of the stress wave into an electrical signal by means of piezoelectric elements. A big advantage of the AE signal is its wide frequency range, usually from 100 kHz to 1000 kHz, which makes it very distinguishable from other unwanted noise caused by the workshop activity or any adjacent machines. The range of frequencies of interest can be isolated by means of a high pass filter in order to achieve higher signal to noise ratio. However, similarly to accelerometers, the sensor location is very important. The signal amplitude decreases rapidly with the distance, thus the proximity to the cutting edge is a key factor. A review of AE-based

sensing methodologies for tool wear monitoring in turning processes was reported by where the author lists the different sources of the AE signal as well as common processing methods for extracting the relevant signal features. According to Li [77] and Teti et al. [16] the common sources of AE in machining processes include the friction on rake face, tool-workpiece interaction, residual stress, plastic deformation, chip break and chip strike. AE sensors are generally less expensive than dynamometers, relatively easy to install and the wide bandwidth makes them suitable for use in production environments.

2.2.1.5 Other Sensing Technologies

Some of the other sensing technologies used in machining for process monitoring purposes include the temperature sensors, optical sensors, ultra-sonic and conductance sensors.

Temperature measurements are of crucial importance in any cutting process. Accurate measurement of the cutting temperature would be very useful since it is correlated with tool wear due to changes in the tool geometry. Moreover, the cutting temperature influences chip formation, accelerates tool wear and influences tool stiffness. However, obtaining accurate measurement of cutting tool temperature is rather complicated and usually average values are provided with significant loss of information. A good review of the methods and techniques used for detection of the cutting temperature in machining processes is presented by Davies et al. [78].

Optical sensors such as machine vision systems are based on a light source to illuminate the surface and to acquire a digital image. As in many applications using machine vision, object illumination influences considerably the performance of the system itself and this fact limits their use in industry [15, 16]. Moreover, the production environment with cutting fluids, chips, etc., makes it extremely difficult to apply in-process optical measurements and usually these measurements are performed once the cutting has finished or in between different cutting cycles. A review of vision systems applied for tool condition monitoring can be found in Kurada and Bradley [79].

The use of ultra-sonic sensors was investigated by Prickett et al. [80] for real-time monitoring of the depth of cut arising during milling operations. The depth of cut was computed by means of two ultrasonic sensors which measured the two distances from the sensors to the workpiece surface before and after the cutting, respectively. This information was then integrated within the tool condition monitoring system.

Castano et al. [81] have recently presented a conductance sensing method for monitoring micro-machining of conductive materials. The sensor measures the electrical contact resistance of the tool-workpiece interface in order to determine the state of the cutting operation.

2.2.2 Signal Processing

Signal processing is a key element in the development of process monitoring systems. The raw sensory data should be transformed in a suitable representation in order to extract relevant information about the process variables of interest. The stages of a general signal processing scheme are shown in Figure 2.2. They include filtering and amplification, A/D conversion, segmentation and feature extraction. The first three stages are usually referred as a pre-processing step that, in addition to feature extraction and sometimes also feature selection (see Section 2.2.5), completes the signal processing scheme.

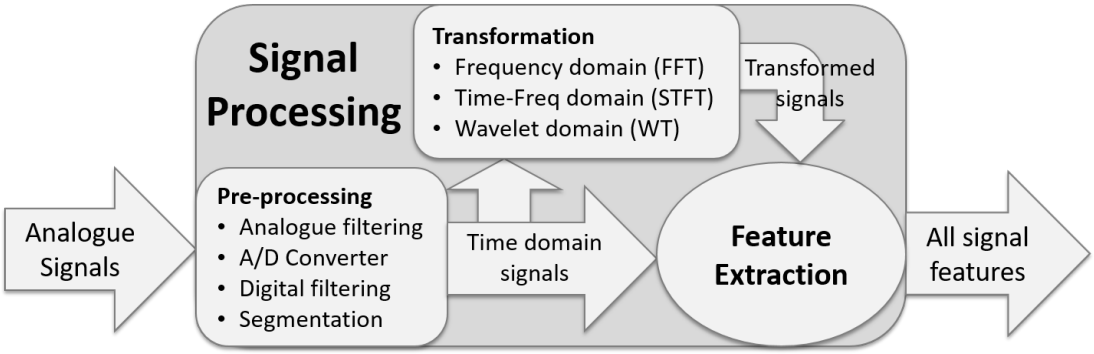

Figure 2.2. Signal Processing Scheme (after [16]).

First, the analogue signal from the sensor is filtered to keep it within the range of the frequency response of the sensor suppressing continuous biases or high frequency noise. This part of processing before the digital conversion is usually embedded in the sensor itself. The filtered signal is then subject to further processing depending on the sensor type. To prevent signal distortion during signal acquisition due to aliasing, the sampling rate should be fixed according to the Nyquist–Shannon [82] sampling theorem, i.e. a sampling rate (f_s) greater than two times the signal bandwidth (B), ($f_s > 2B$). In general, just before conversion, in order to obtain the highest possible accuracy, the analogue signal is amplified such that the maximum voltage range equals the maximum input range of the A/D converter. The obtained signal is then ready to be further processed in digital form through filtering, segmentation and transformation into frequency or wavelet domain. Envelope analysis is also a well-known technique in condition monitoring, mostly used for the detection of faults

in industrial machines associated with rolling elements (bearings) [83, 84]. The method extracts the amplitude modulated signal, from the rotating machine vibrations, generated by the periodic changes in the forces due to the development of a fault.

In some works [50, 57, 58, 69], segmentation was also part of the processing scheme. It refers to the operation of extraction of the portion of signal that contains information about the cutting process or tool conditions, i.e. the part when the tool is actually removing material. However, as noted by Abellan-Nebot and Subiron [15], the segmentation prevents the resulting signal from being analysed in the frequency and wavelet domain. For instance, if the signal is segmented and then reconstructed by putting all the extracted portions in a single time series, one next to the other, the resulting signal is not equivalent with the original one and therefore the frequency content will be distorted.

The next step in the processing scheme is the feature extraction, which, as will be discussed in details in the next section, transforms the digital signal into several signal features. These features are typically calculated through transformation and analysis of the sensor signal in the time, frequency and wavelet domain with the intention to reduce the dimension of the input space while preserving the relevant information that characterise the machining process. It will be clear later in Section 2.2.5 that it is very important to keep the dimension of the feature vectors to the lowest possible.

2.2.3 Feature Extraction

A number of feature types and extraction methods have been reported in the machining process monitoring literature [14-17, 19]. Common methods of the reviewed papers include signal analysis in the time and frequency domain, but also some more advanced processing methods such as analysis in joint time-frequency and the wavelet (time-scale) domain. All such methods are presented and discussed in the following.

2.2.3.1 Time Domain Analysis

A simple group of features that provides an overall picture of the time-signal is based on statistic descriptors such as the arithmetic mean, Root Mean Square (RMS), variance, skewness and kurtosis, as shown in Table 2.2. Time domain features extracted from sensor signals collected during machining have been reported in a number of research papers [48-50, 52, 56, 62, 73, 76, 85-87] , including those of Binsaeid and Cho [48, 49] after which this table was built.

Suppose that the signal of interest $x(t)$ has been sampled at regular intervals and a total of N points have been obtained. The data is now the set of values $[x_1, x_2, \ldots, x_N]$ and the following time-domain features, summarised in Table 2.2, can be extracted.

Table 2.2 Features extracted from time domain signals (after [48, 49]).

Feature	Symbol	Definition	Equation
Arithmetic Mean	**M**	The mean of N amplitude values of raw data signal	$M = \frac{1}{N}\sum_{i=1}^{N} x_i$
Root Mean Square	**RMS**	RMS of a sample length of N amplitude values in the raw data	$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
Variance	**Var**	Variance of a sample length of N values with mean M	$Var = \frac{\sum_{i=1}^{N}(x_i - M)^2}{N-1}$
Skewness	**Sk**	A measure of the asymmetry of the probability distribution ($\sigma = \sqrt{Var}$)	$Sk = \frac{1}{N}\frac{\sum_{i=1}^{N}(x_i - M)^3}{\sigma^3}$
Kurtosis	**Ku**	A measure of the "peakedness" of the probability distribution ($\sigma = \sqrt{Var}$)	$Ku = \frac{1}{N}\frac{\sum_{i=1}^{N}(x_i - M)^4}{\sigma^4}$
Power	**PWR**	Signal power as the measured area under the rectified signal envelope	$PWR = \frac{1}{N}\sum_{i=1}^{N} x_i^2$
Peak-to-Peak	**pp**	Difference between the highest peak value and the lowest peak value for a given sample of N values	$pp = \max_N(x_i) - \min_N(x_i)$
Crest Factor	**CF**	The peak amplitude divided the RMS value of raw data signal	$CF = \frac{peak}{RMS}$
Burst rate* (Pulse rate)	**BR**	Number of times the signal exceeds pre-set thresholds per second.	-----

(*) BR can only be applied to vibration and AE signals

2.2.3.2 Frequency Domain Analysis

The frequency spectrum $X(f)$ of a continuous signal $x(t)$ can be calculated by means of the Fourier transforms, as follows:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft}\, dt \tag{2.1}$$

Where f is the frequency variable measured in Hz. In case of sampled signals, the frequency content is represented as the discrete Fourier transform and a common algorithm used for its efficient computation is the Fast Fourier Transform (FFT).

Frequency domain features are reported in several machining related research works [48, 49, 52, 56, 62, 73, 85, 87]. The raw sensor data is transformed into the frequency domain using the FFT algorithm and the Power Spectral Density (PSD) is obtained. A Hanning window is usually applied to the raw data to prevent leakage. The PSD gives information about the signal power in a particular spectral component $S(f)$. Similar features based on statistical information such as the mean, variance, skewness and kurtosis can be calculated for the power spectrum values over a specific frequency band. Following from Binsaeid and Cho [48, 49], the most commonly used frequency domain features for machining monitoring applications are illustrated in Table 2.3.

Table 2.3 Features extracted from frequency domain signals (after [48, 49]).

Feature	Symbol	Definition	Equation
Sum of Total Band Power	**STBP**	The total power in the signal or in a particular freq. range (f_1, f_2)	$STBP = \int_{f_1}^{f_2} S(f)\, df$
Mean of Band Power	**MBP**	Mean of power spectrum in a specific freq. band of length N	$MBP = \frac{1}{N}\sum_{i=1}^{N} S(f)_i$
Variance of Band Power	**VBP**	Variance of power spectrum in a specific freq. band of length N	$VBP = \frac{\sum_{i=1}^{N}(S(f)_i - MBP)^2}{N-1}$
Skewness of Band Power	**SkBP**	Skewness of power spectrum in a specific freq. band of length N with mean MBP and variance VBP	$SkBP = \frac{1}{N}\frac{\sum_{i=1}^{N}(S(f)_i - MBP)^3}{VBP^{3/2}}$
Kurtosis of Band Power	**KuBP**	Kurtosis of power spectrum in a specific freq. band of length N with mean MBP and variance VBP	$KuBP = \frac{1}{N}\frac{\sum_{i=1}^{N}(S(f)_i - MBP)^4}{VBP^{4/2}}$
Peak of Band Power	**PBP**	The peak of power spectrum in a specific frequency band B	$PBP = \max_{B} S(f)$
Frequency of max Peak of Band Power	**FPBP**	The frequency that corresponds to the highest amplitude of power spectrum in a specific freq. band	-----
Relative Spectral Peak per Band	**RSPB**	The ratio of Peak of Band Power over the Mean of Band Power	$RSPB = \frac{PBP}{MBP}$
Total Harmonic Band Power	**THBP**	The total power of Tool Passing Frequency (TPF)* and all the harmonics in the frequency range of the signal	$THBP = \sum_{k=1}^{N} S(k\,TPF)_k$

(*) The Tool Passing Frequency is calculated as $TPF = \frac{Spindle\ RPM}{60}\ x\ n_{teeth}$

2.2.3.3 Time-Frequency Analysis

If the frequency content of a signal varies over time, the FFT will provide just a time average summary of all the frequencies and no other information about the time when those frequencies varied will be given. The analysis of non-stationary signals such as those detected during machining requires specific time-variant techniques that go beyond the classical Fourier approach [17]. These techniques are typically divided in two groups: time-frequency analysis and time-scale (wavelet) analysis.

The joint time-frequency analyses the frequency content of a signal within a window which is fixed in size and moves with time along the signal. The signal is assumed to be stationary within the window and thus the FFT can be applied to calculate the spectral content of this short length of data. Then the window slides along the time axis to a new position and the calculation is repeated. The time information is provided by computing different FFT for consecutive time intervals and then putting all together. When the window is rectangular, i.e. just a finite segment of data is selected, this analysis leads to the Short-Time Fourier Transform (STFT). The major drawback of this approach is that the window width decides on both the time and frequency resolution. A narrow window implies good time resolution, but, on the other hand, it results in a poor frequency resolution and vice versa. A trade-off must be made for the desired time and frequency resolution. However, it is possible to choose the window in such a way to minimise the negative effects of this trade-off. For instance, if one chooses a Gaussian window instead of the rectangular one, the optimum performance is obtained and this analysis leads to the Gabor transform [88]. Some machining related research works that report the use of STST for feature extraction are those conducted by Marinescu and Axinte [57, 58, 69].

2.2.3.4 Time-Scale Analysis: Wavelet Domain

A completely different approach that overcomes the above resolution problem of the STFT is based on the wavelet theory developed by Mallat [89] and Daubechies [90]. Unlike the Fourier bases, the Wavelet Transform (WT) basis functions are localised in time and they are all scaled and translated versions of one single function called the mother wavelet. The original time signal is transformed into the time-scale domain.

The WT can be seen as the STFT, but with windows of different lengths for different frequencies, allowing a multi resolution analysis of the signal [91]. The mother wavelet acts as a window function and its scaled versions modify the window length, whereas its

translated versions change the window location in the time axis. Scale and frequency are inversely related: low scales refer to high frequency and vice versa. Therefore, for high frequency (low scales values) the windows are very narrow in time and better time resolution is achieved, whereas for low frequencies (high scales values) the windows became wider in time and better frequency resolution is obtained. In other words, for the high frequency content of the signal that changes quickly and has a short duration in time the WT offers better time resolution to localise this duration, whereas in case of the low frequency content of the signal that often has a long duration in time, the WT offers better frequency resolution.

Practical implementations of WT are the Discrete Wavelet Transform (DWT) [89] and the Wavelet Packet Decomposition (WPD) [92], which are relatively easy to implement and also have a fast algorithm for estimating the wavelet coefficients. In the machining context, different studies [73, 76, 77, 93, 94] have reported the use of wavelets for processing the signals collected during the process. A review of the wavelet analysis applied for tool condition monitoring is presented by Zhu et al [93].

2.2.4 Feature Selection Methods

After extracting the features from sensor signals, the next step is to identify those with relevant information about the specific monitoring purpose. This task can be very difficult especially if the number of features is large and they cannot be easily related to the process variable of interest. Therefore, the feature selection step is of critical importance in the development of robust and reliable monitoring systems. The objective of feature selection is to preserve relevant information by identifying those features that contain the most relevant information about the process under investigation, and, at the same time, to reduce the feature space by and removing redundant or irrelevant features. Good features are those which vary (with respect to some criteria) in a similar way with the variations of the target value, i.e. the process variable of interest. Therefore, they help to distinguish between the various classes defined on the target variable in a classification problem or obtain accurate numerical predictions in a regression problem.

The problem of feature selection is well-known in the Machine Learning (ML) and pattern recognition research [65, 95, 96], in which the number of features (or attributes in the ML terminology) is usually very large and complex learning algorithms are required. However, in machining monitoring applications, just few works [48, 49, 73, 97, 98] have reported a separate feature selection step as part of their system design. The majority of the researchers

extracted the sensor features based on previous works of the specific application or monitoring purpose and without considering any particular strategy for feature selection.

There are many advantages of applying feature selection to a machining monitoring system [15, 17]. First of all, a lower number of features, leads into a low-dimensional input space for the learning problem (see Section 2.2.5) when building a process model. This is very important because it avoids a well-known problem of the ML community called 'the curse of dimensionality', which was introduced by Richard Bellman [99]. This term refers to the various issues that arise when handling high-dimensional input spaces [100]. In particular, the learning problem becomes more difficult with the increasing dimension of the input space since more training data is needed in order to build an accurate model. In machining, the data is expensive to obtain since it involves experiments. Moreover, simpler models can be built using fewer features, which are more robust on small datasets and present less variance on noise or outliers. Low-dimensional data can also be easily plotted and visualised, thus providing a better understanding about the process. Finally, using just a small subset of the original feature set to build the model means that further measurements and signal processing can be reduced to extract just those few selected features with a considerable reduction of computational time.

According to the ML research [65, 95, 96], feature selection methods typically fall in three categories: filter-based, wrapper-based and embedded methods. Filter methods are used to reduce the feature input space by filtering out less relevant features. Generally, they are applied as a pre-processing step before the learning commences and do not dependent on the choice of the learning technique. A suitable metric is used as ranking criterion to score and order the features. Then, a pre-set threshold value is used to remove the lower scored features. The advantages of this method are its simplicity, scalability, and high degree of empirical success [95, 96]. It is also computationally efficient since it requires only the computation of one score for each feature. Wrapper-based methods, presented by Kohavi and John [101], make use of the learning algorithm to evaluate the quality of the selected features. These methods give a score to a subset of features based on the predictive performance of a pre-selected learning technique that uses these features in input. In other words, given a subset of features, the learning algorithm acts as a black box to evaluate their quality based on the learning performance and then, the feature subset that obtains the highest score is selected. The subset search space for D features is 2^D, which makes the exhaustive search impractical when D is large (this is a NP-hard problem). Therefore,

different searching algorithms are typically employed, which yield local optimum results but are computationally feasible. Embedded methods [95, 96] provide a trade-off solution between filter and wrapper methods in terms of the computational costs. They incorporate the feature selection into the training process as part of the objective function defined for learning. This improves the selection process compared to wrapper methods since it avoids re-training the predictor from scratch for each subset evaluation. Typically, the objective function of the learning technique consists of two terms: a term for the goodness-of-fit to be maximised and a term for the number of features to be minimised. The latter is often referred to as a regularisation term, which aims to 'shrink' the parameters.

2.2.4.1 Ranking Criterions for Feature Relevance

In the field of machining monitoring systems the most widely applied are filter-based methods due to their simplicity and low computational cost. Some of the ranking criterions used to assess the relevance of a feature with respect to the process variable of interest include Pearson's correlation coefficient [75, 97, 98], mutual information [48, 49] and correlation-based methods [48, 49], among others.

Pearson's correlation criterion

This is the classical approach of measuring the linear correlation between two random variables. The most well-known measure is the Pearson's correlation coefficient, which for a pair of random variables (X, Y) is given by the formula:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \tag{2.2}$$

Where $\bar{x}$ and $\bar{y}$ are the mean of X and Y, respectively. The value of r lies in the interval $[-1,1]$. Limit values (1 or -1) indicates a complete linear correlation between X and Y, whereas for $r = 0$ a totally independence is indicated. Typically the value of r^2 is used since the goal is to find those variables which best relate to the process variable of interest regardless the correlation sign. A drawback of this approach is that it can only give a measure of linear correlation between the variables without considering for other non-linear correlations. It also requires that all the features are numeric values.

Mutual information

This is a correlation measure based on information theory [102]. It uses the concept of entropy, which represents a measure of the uncertainty of a discrete random variable. The entropy of a discrete random variable Y is defined as:

$$H(Y) = -\sum_{y_i \in Y} p(y_i) \log_2\big(p(y_i)\big) \tag{2.3}$$

Where y_i denotes a specific value of random variable Y, $p(y_i)$ denotes the probability of y_i over all possible values of Y, which can be estimated from the data. Suppose Y represents the output variable and an observation of another discrete random variable X is made. Then the conditional entropy of Y given the observation data X is defined as:

$$H(Y|X) = -\sum_{x_j \in X} p(x_j) \sum_{y_i \in Y} p\big(y_i|x_j\big) \log_2 p\big(y_i|x_j\big) \tag{2.4}$$

Where $p(x_j)$ is the prior probability of x_j, whereas $p\big(y_i|x_j\big)$ is the conditional probability of y_i given x_j. Equation (2.4) implies that by observing a variable X, the uncertainty in the output Y is reduced. The decrease in uncertainty is given as:

$$I(Y,X) = H(Y) - H(Y|X) \tag{2.5}$$

Where $I(Y,X)$ is called information gain and, since it measures the amount of information shared by the two variables together, it is often referred as Mutual Information (MI). It can be noticed that MI is symmetric such that $I(Y,X) = I(X,Y)$. It is zero if the discrete variables Y and X are independent and greater than zero if there is some dependency between the two variables. However, as reported by Yu and Liu [103], information gain presents a bias in favour of features with more values and normalisation is required. To overcome these issues, the authors used the Symmetrical Uncertainty (SU) coefficient:

$$SU(X,Y) = 2\left[\frac{I(Y,X)}{H(X) + H(Y)}\right] \tag{2.6}$$

SU value is normalised to the range $[0,1]$. The value zero indicates that X and Y are independent, whereas the value 1 indicates that knowledge of either one variable

completely predicts the other. Note that entropy based methods work with nominal features, but they can be applied to numeric features if their values are properly discretised in advance [65, 95, 103]. A common discretisation method used for this purpose is the one proposed by Fayyad and Irani [104].

Correlation-based Feature Selection (CFS)

This method, proposed by Hall [105], is able to handle the problem of feature redundancy by considering the level of inter-feature correlation in the selection procedure. The correlations are evaluated by the entropy measurements, such the MI or the SU coefficient presented in Equation (2.6). The CFS is defined for classification tasks and it takes into account both the relevance with the class and the redundancy of the features. High scores are assigned to features that are highly correlated with the class, yet have low inter-correlation with each other. Unlike the other criterions discussed above, the CFS method does not evaluate each feature individually, instead it gives a score to a subset of features taking into account the inter-correlations of among features within the subset under evaluation. The merit coefficient $Merit_S$ of a subset of features S is defined as:

$$Merit_S = \frac{k\,\overline{r_{cf}}}{\sqrt{k + k(k-1)\,\overline{r_{ff}}}} \tag{2.7}$$

Where, k is the number of features within S; $\overline{r_{cf}}$ and $\overline{r_{ff}}$ are the average values of feature-class correlation and feature-feature inter-correlation, respectively. Note that the numerator measures the relevance of the features, whereas the denominator considers the redundancy among the features within the subset. An application of the CFS in machining monitoring systems was presented by Binsaeid et al. [48]. The method was applied to evaluate the relevance of different signal features with respect to tool condition classes.

Borah et al. [106] proposed an Enhanced Correlation-based Feature Selection (ECFS) method which uses statistical techniques instead of information theory measurements to assess both feature-feature correlation and the feature-class correlation. This ECFS method was tested and compared with other feature selection methods in several ML datasets.

2.2.4.2 Other Feature Selection Strategies

Some other ranking criterions and feature selection methods used in machining monitoring applications include the Relief algorithm [107], Fisher score [98], Class separability [98], Orthogonal Arrays [73], Analysis of Variance [75] and χ^2 statistic-based methods [49].

Zhang et al. [98] proposed an interesting hybrid approach to select relevant features for machinery condition monitoring. The method consisted in three main parts: filter, voting and wrapper. The filter part was used to pre-rank the sensor features through four different ranking criterions including Pearson's correlation, Relief algorithm, Fisher score and Class separability. The voting part was introduced to re-rank the features through a weighted voting scheme of the above four methods. In the wrapper part, subset searching strategies such as the binary search and backward elimination were applied to further reduce the feature number and obtain the final subset.

Al-Habaibeh and Gindy [73] applied Taguchi's Orthogonal Arrays (OA) to calculate the dependency (sensitivity) of the sensory features on the machining parameters and the tool conditions. The features with higher dependency on the tool conditions rather than on the machining parameters were selected for further analysis.

Cho et al. [49] investigated the CFS and χ^2 statistic-based methods to assess the relevance of sensory features with respect to classes defined on the tool state. They studied the influence of reducing the feature space in a classification problem using process models built by several ML algorithms.

2.2.5 Process Model Selection

The final step of the development of an IMS for machining processes consists of building a suitable process model based on ML approaches. This section will first introduce the concepts of supervised learning in the process monitoring context, then discuss methods to validate the performance of the models and finally, review the most popular approaches.

2.2.5.1 Supervised Learning Methods

The supervised learning approach [17, 100, 108] requires learning the relationship between two sets of data: a D-dimensional input space, denoted with $\underline{x}$ and a one-dimensional output space, denoted with y. This differs from the unsupervised learning approach, where the output space is not provided and the learning algorithm seeks intrinsic relationships within

the input data in order to categorise (group) them into clusters with similar characteristics. In supervised methods, the underlying relationship is induced based on a set of training data $(\underline{x}_k, y_k);\ k = 1, \dots, N$, where each input vector $\underline{x}_k$ is paired with a 'true' value of the output variable y_k and N is the number of the available pairs (called instances) in the dataset. Depending on the type of the output variable, the problem can be one of regression, if y is a continuous numerical variable, or one of classification if y is a nominal variable (class label). Denoting the mapping between the two sets with f, the relationship can be written as $y = f(\underline{x}, \underline{w})$, where $\underline{w}$ is a vector of free parameters that can be adjusted to give the best-fit of the model on the training data. Therefore, the aim of a learning machine is to find an optimal choice for the parameters $\underline{w}$ with respect to some objective function which maximises the goodness of fit of the model [100].

In the context of process monitoring, the input data consist of N measurements (instances) with D extracted/selected features (called attributes or predictors in the ML community) for each measurement. The objective of a learning algorithm is to learn the relationship of the input features with respect to the process variable of interest (or target variable) that, depending on the monitoring scope, can be the tool wear, surface roughness, chip condition or another process variable that one wants to predict. The target variable can either be a nominal label for classification problems or a continuous numeric value for regression problems.

A common issue of ML techniques is referred to as overfitting. This is the problem of memorising the training data rather than learning the underlying function of interest and it occurs when there are too many parameters in the model compared to the number of training instances. This means that the machine is actually learning about peculiarities of the training instances and it will fail to generalise on new, previously unseen data. The generalisation capacity is related with the model complexity: a more complex model (with more free parameters $\underline{w}$ in f) has higher probability of overfitting the training data and therefore it could fail to generalise. It is therefore important to assess the generalisation performance of the model on previously unseen data.

2.2.5.2 Model Performance Validation

A very popular method for evaluating the generalisation performance of a learning algorithm (i.e. the process model) is to split the entire available data into two parts: one for the actual training called the training set, and the other for testing the performance of the trained

model, called the testing set. Since the data of the testing set are not used during the training procedure, they are all new and unseen for the obtained model, thus the accuracy on this dataset gives an estimate of the algorithm's true generalisation performance.

In cases where the data availability is large, one can further divide the training data in two parts: training set and validation set. The data of the validation set are used as extra test data for evaluation in order to determine the hyper-parameters of the particular model. There is however a risk of discarding representative training data when dividing the dataset, which could potentially lead to a suboptimal assessment that would depend on the particular choice of training and testing sets. In addition, the size of the portion of data left for testing is important. If it is too small then the estimate of the generalisation performance may be not realistic, whereas if a large portion of the dataset is used for testing then the remaining data for training may be too low which will lead to poor training. Furthermore, there is no guarantee that the portion of testing data (regardless its size) is a representative example of other new future data. For example, in a classification problem, it might happen that the instances of the testing set are located very close or much further to the decision boundary resulting to a very difficult or too simple classification, respectively.

One way to overcome the above issues is to use the cross validation (CV) approach [109] to estimate the generalisation performance of the learning algorithm. The method is called k-fold CV and it is illustrated in Figure 2.7 (after Polikar [110]).

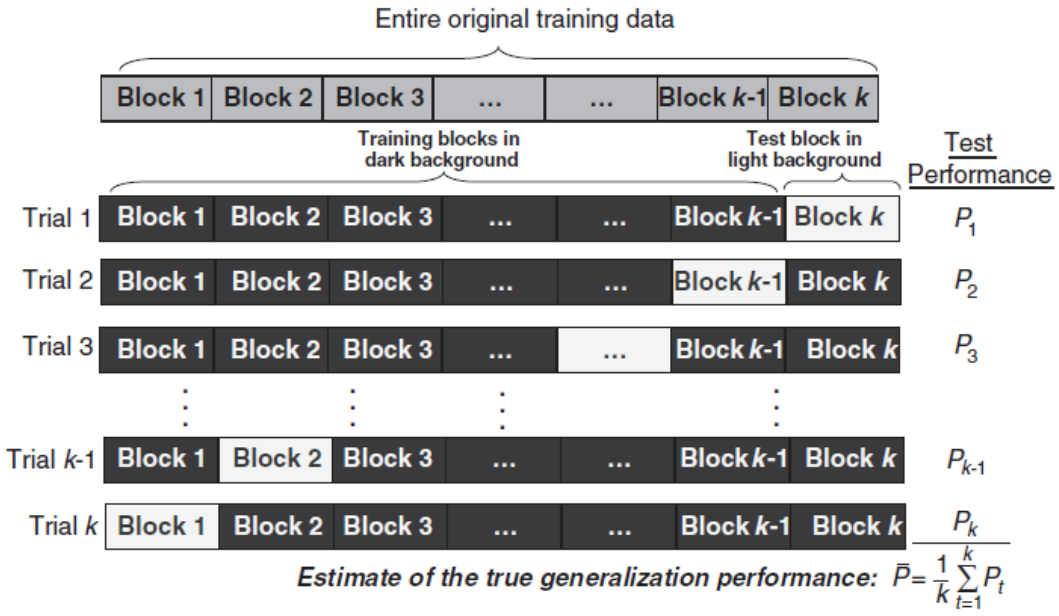

Figure 2.3 Illustration of k-fold cross validation method (after [110]).

The method consists of dividing the entire dataset is into $k > 2$ blocks (folds) of data. One of the blocks is used for testing, whereas the remaining $k - 1$ serve as training data. The performance based on this partition is then computed and stored. The procedure is repeated k times, using in each case a different block for testing, until all the blocks have been used. The estimate of the true generalisation performance of the learning algorithm is then calculated as the average value of the performances obtained for each trial. Note that the k-fold CV method allows fully utilisation of all available data. All the instances are used for training as well as for testing, but never at the same time. In addition, since the average performance value over the k trials is obtained, there is a lower probability of choosing a particular unusual partition of representative data, which might give 'false' performances.

The choice of the parameter k depends on the data availability. If k is too large, it divides the data in too many blocks, which means that the training set will be much bigger than the testing set. However, few test instances can result in large variations on the test performances. On the other hand, choosing k too small results in few blocks which means that not enough training data may be provided. Typical values of k for sufficiently large datasets include 5 or 10 [100, 109, 110]. When the data is in short supply k may be chosen large to allow for a larger training set. The extreme case of choosing $k = N$, where N is the number of all the available instances, is also known as Leave-One-Out (LOO) cross validation method [109, 111]. In this case, for each training session, the entire dataset except one is used and then the model is tested only on that one remaining instance. The procedure is repeated for N times. In theory, this method provides the best estimate of the generalisation performance, since every data is tested, but it is computationally more expensive than the k-fold CV. In classification tasks with unbalanced datasets, a slightly different version of the algorithm, called the stratified k-fold CV [111], is usually applied. It creates blocks of stratified data, in which the original distribution of class labels is conserved.

2.2.5.3 Common ML approaches

The Artificial Neural Networks (ANNs) are considered to be the most commonly used approach in machining research. The main applications include cutting tool state diagnosis [48-51, 73, 85, 112, 113], surface roughness prediction [56, 114] and process fault diagnosis [85]. Typically, a large number of samples obtained through experimental trials is required to achieve high accuracy and good generalisation capability. The main drawback of these systems is that they are unable to adapt to different processes without having a large amount of experimental data to use for training. Another popular ML method consists of the Support

Vector Machines (SVMs) [115], which have been successfully applied in several machining operations including milling for tool state diagnosis [48, 49], broaching for tool wear prediction [66] and turning for surface roughness [54]. The optimisation problem of the SVMs has a convex objective function, which can be solved efficiently through numerical methods and a local solution is also the global minimum.

Some other learning approaches are based on Artificial Intelligence (AI) such as fuzzy systems [73, 116], the hybridisation neuro-fuzzy [68, 117], evolutionary algorithms [53, 76] and probabilistic methods [86, 87, 118]. A recent review of the applications of AI in fault diagnosis of rotary machinery can be found in Liu et al. [119].

The systems based on fuzzy logic [120] require previous knowledge of the process in order to function. This knowledge is used to define the fuzzy rules which are the basis of the systems computation. These systems can work with a low amount of experimental data available, which makes them more flexible and suitable to apply in new processes. However, they do not have much learning capabilities compared with the ANN [15]. The hybridisation of ANN and fuzzy systems leads to the neuro-fuzzy systems. Applications of neuro-fuzzy systems for tool wear monitoring can be found in Azmi [68] and Gajate et al. [117]. Evolutionary methods have also been applied for process models in monitoring applications. Xiang et al. [76] used the artificial immune algorithm to detect faulty conditions in grinding operations, whereas Colak et al. [53] applied genetic algorithms for surface roughness prediction in milling processes.

Probabilistic based monitoring systems [86, 87, 118] are particularly suited in application where the process has a highly stochastic behaviour. Such systems can use previous knowledge on the process behaviour and extract hidden causal relationships and probabilities from experimental data. Probabilistic models can also provide predictions with an expected uncertainty level, which can be seen as a measure of the model's confidence in the prediction value. A probabilistic approach that is getting more attention in the current machining research consists of Gaussian Processes (GP) [121-123]. A GP is a generalisation of the Gaussian probability that describes a distribution over functions (instead of scalars or vectors). Very few works have applied GP for monitoring purposes, however, the interest in this area is continuously growing. Two recent examples include the work of Teimouri et al. [124] reporting a GP-based approach for structural health monitoring and the paper of Kong et al. [125] concerning the use of GP regression for tool wear prediction. This approach was also used in the work conducted here for the development of the monitoring system. In

particular, process models based on Gaussian Process Regression (GPR) were created from the experimental data (ref. to Chapter 6). A detailed presentation of the GPR theory is also given in the Appendices (ref. to Section A.1).

2.3 Research Gap Analysis

Different aspects of the two main research fields of robotic machining and IMS for machining processes have been reviewed and discussed extensively in the previous sections of this chapter. However, in order to better understand the contribution of this work into the current literature, it is useful to analyse the main outcomes of the reviewed material and the existing challenges of these two research areas.

One key aspect of robotic machining is that the technology itself is still in its beginnings and constant research is required in order to advance it further, up to the state of being considered a valid alternative to conventional CNC machining. It was identified in Section 2.1 that the main challenges of this technology were due to the limited and variable stiffness of robots and their low positional accuracy, which inevitably introduced non-systematic (random) errors into the machining process that were difficult to anticipate. From the process point of view, such errors could be quantified in terms of variations (imperfections) in the quality of the finish product (surface roughness, part geometry) or in the process response variables, called the Key Performance Indicators (KPIs).

Most of the solutions published in the robotic machining literature have addressed this problem from the robotic structure perspective, proposing methods to directly minimise the above sources of errors (i.e. the robot's low stiffness and positional accuracy) and enable industrial robots for machining operations. Some relevant research papers of compensation strategies based on ML approaches were reviewed in Section 2.1.2. Typically, accurate measurements from optical laser trackers and/or dynamometers were used to build an error map of the robot positioning within its workspace. This information was then integrated into the robot path planner or controller in order to compensate offline or online (by closed loop control schemes) the cutting tool deviations. Satisfactory results have been reported, however, these solutions were mainly applied in a controlled environment such as a research laboratory, where the goal was to test the feasibility of the approach. Moreover, the experiments often required the use of expensive and/or intrusive equipment, such as high accuracy laser trackers and dynamometers to measure the displacement of the robot TCP and monitor the process forces. In order to transfer these solutions into an industrial

environment, new strategies with focus on high productivity rate are required. In particular, the process down time for the initial set-up, the cost of equipment used for data collection, their intrusive nature and the time required for obtaining the training data when ML models are all factors to consider in the development of a monitoring system. There is therefore an opportunity for the work conducted here to design an industrial-friendly IMS that takes into account the above issues of current solutions.

The IMS for machining processes, on the other side, is considered a mature research area, with a large number of published literature over the last 25 years, mainly focused on tool condition monitoring. This has permitted researches to identify effective signal processing methods and ML techniques to obtain accurate predictions of several process KPIs. Despite its popularity in conventional machining processes, very few applications of process monitoring systems in robotic machining context were found. This is because the IMS research was concerned with other factors that influence the quality of the products (tool wear and breakage, machine condition, fault diagnosis, chatter analysis), which differ from those of robotic machining discussed above. In fact, process variables, such as the actual depth of cut, feed rate and cutting tool trajectory present much lower amount of uncertainty (deviations from the nominal values) in conventional machining compared to the robotic case. Consequently, their influence in the quality of the workpiece is considered negligible due to the higher stiffness and tool positional accuracy of the CNC machine tools.

While most of the reviewed IMS research focused on tool condition monitoring, fault diagnosis, surface roughness and chatter analysis, this work investigates the use of intelligent systems in monitoring of other process variables (depth of cut, tool trajectory) which are subject to variations during robotic machining. In particular, the proposed method integrates the concepts of IMS such as signal processing, feature extraction and ML techniques into the robotic machining context, addressing therefore the current gap in the number of applications that combine these two research areas.

Unlike previous works, this thesis focus on the effects of the robotic structure into the manufacturing process, considering the robots performing the operation as an additional source of error that has a major influence to the overall process performance, resulting in imperfections of the finished product. Therefore, if the errors introduced by the robotic structure can somehow be quantified in terms of deviations of the process response variable, then applying an IMS that provides in-process predictions of the amount of such errors, can

drive corrective actions (when possible) to ensure the quality of the final product is within the process tolerances.

The main focus of this work is to provide a means of improving the performance of a robotic machining process by the use of in-process monitoring of key process variables which directly influence the quality of the machined part. The proposed solution designs an IMS that uses sensor signals collected during machining to predict the amount of errors introduced into the manufacturing process in terms of imperfections of the finish product (i.e. variations from the nominal values).

2.4 Literature Review Summary

This chapter presented and discussed the latest research developments in the fields of robotic machining technology and intelligent monitoring of machining processes based on supervised learning approaches.

In terms of robotic machining literature, the advantages (potentiality) as well as the limitations (challenges) of this new emerging technology were identified. Several publications agreed in the fact that the extra flexibility and the lower costs of the robots are the main drives to apply them in manufacturing processes. However, a number of challenges needed to be addressed by current research, with the two most serious being the robot's limited stiffness and low positional accuracy. The main technical improvements of robotic machining technology included the design of robotic structures with higher stiffness and better positional accuracy, evolution of CAD/CAM tools for robotic process simulation and the integration of software packages for robot programming. In terms of the research developments, the main categories included robot calibration and joint stiffness identification under non-kinematic errors, creation of a stiffness map for robot posture optimisation, chatter avoidance methods and compensation strategies for cutting tool deflection. This last research category was discussed in more detail presenting the publications with focus on ML approaches, relevant to this thesis.

In terms of process monitoring research, all elements of IMS used in conventional machining operations were described in detail. The discussion started with the review of the sensory systems, typically applied to gather information during the machining process. A variety of sensor types was used for monitoring purposes including the spindle power and current sensors, dynamometers for measuring the cutting tool forces, accelerometers for workpiece

and tool vibration and the AE sensors. The majority of the researchers preferred to apply multiple-sensor systems in order to get different signals about the same process. The signal processing scheme was also reviewed. Common operations on the sensory data included digital filtering, signal segmentation and the domain transformation. Signal analysis in the time, frequency, joint time-frequency and wavelet domain was usually performed to extract several features describing the process. Another important aspect of the review conducted in this study was the investigation of the feature selection problem. Methods for the identification of the most relevant features as well as the criterions used to assess the feature relevance were discussed. Feature selection methods are typically applied in pattern recognition tasks in order to reduce the feature (input) space. However, the integration of such methods in the monitoring system was investigated in this work. Using fewer features to describe the process would lead to the creation of simpler models that are faster to train and therefore suitable for real-time operations. Different techniques, borrowed from the ML research were presented and discussed in order to understand how the process models were built and how their generalisation performance was assessed. Finally, the chapter concluded with an analysis of the reviewed material, highlighting the research gap opportunity for the work conducted by this study.

3 CASE STUDY: ROBOTIC COUNTERSINKING PROCESS

This chapter presents the details of the case study used for the implementation of the IMS proposed by this work. It starts with an introduction of the robotic countersinking technology and a description of all the components of the robotic cell in Section 3.1. Then, further details on the process steps required to perform the specified machining operation are given in Section 3.2. Finally, the countersinking metrology and the current limits of the process are discussed in Section 3.3.

3.1 Robotic Cell Components

In this work, sensor data collected during a robotic countersinking process have been analysed and used as a case study for the implementation of the proposed monitoring system. In particular, the robotic cell consisted of two industrial articulated robots that work together to countersink previously drilled holes in composite aircraft components. The technology itself, including the cell components and process steps, was developed by the Advanced Manufacturing Research Centre (AMRC), The University of Sheffield as part of a research project for BAE Systems [126], led by the AMRC in collaboration with KUKA System UK. The output of this project was a full scale production demonstrator. A similar robotic cell, completely functional for production is currently installed at BAE Systems in the UK.

The robotic cell located at the AMRC was made available during all the experimental work required for this study. A close up view of the cell is shown in Figure 3.1. It consisted of two KUKA robots: a large KR360 (master robot) with countersinking end-effector, illustrated on the left side of Figure 3.1 and a (relatively) smaller KR180 (slave robot) with anvil end-effector to react process loads, located on the right side of Figure 3.1. Both robots used Kuka KRC2 controllers and ran RoboTeam software to facilitate the co-operation functionality of the cell. The end-effectors were controlled by a central Siemens Programmable Logic Controller (PLC) system. The objective of the cell was to automatically countersink aircraft panels, made of Carbon Fibre Reinforced Polymers (CFRP), which had been previously drilled on a high precision machine tool.

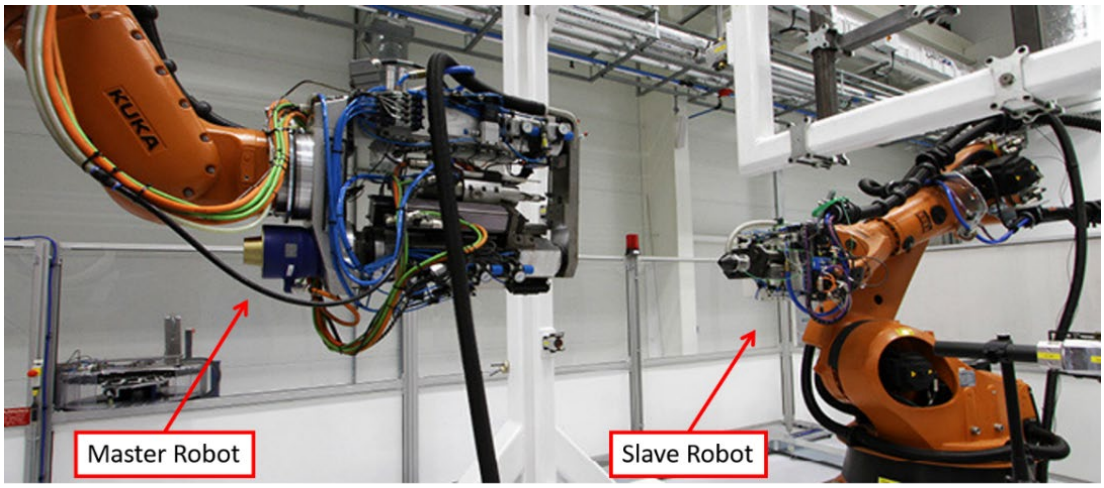

Figure 3.1 Close up view of the Countersinking Robotic Cell (after [126]).

The overall countersinking process flow could be briefly described as follows: both robots moved in the vicinity of a pre-drilled hole and precisely located over it; the slave advanced slowly towards the panel until contact was made and then stopped; the master activated an air-driven Pressure Foot (PF) installed at the end-effector, clamped the panel and then the cutting commenced. A schematic diagram of the process is presented in Figure 3.2.

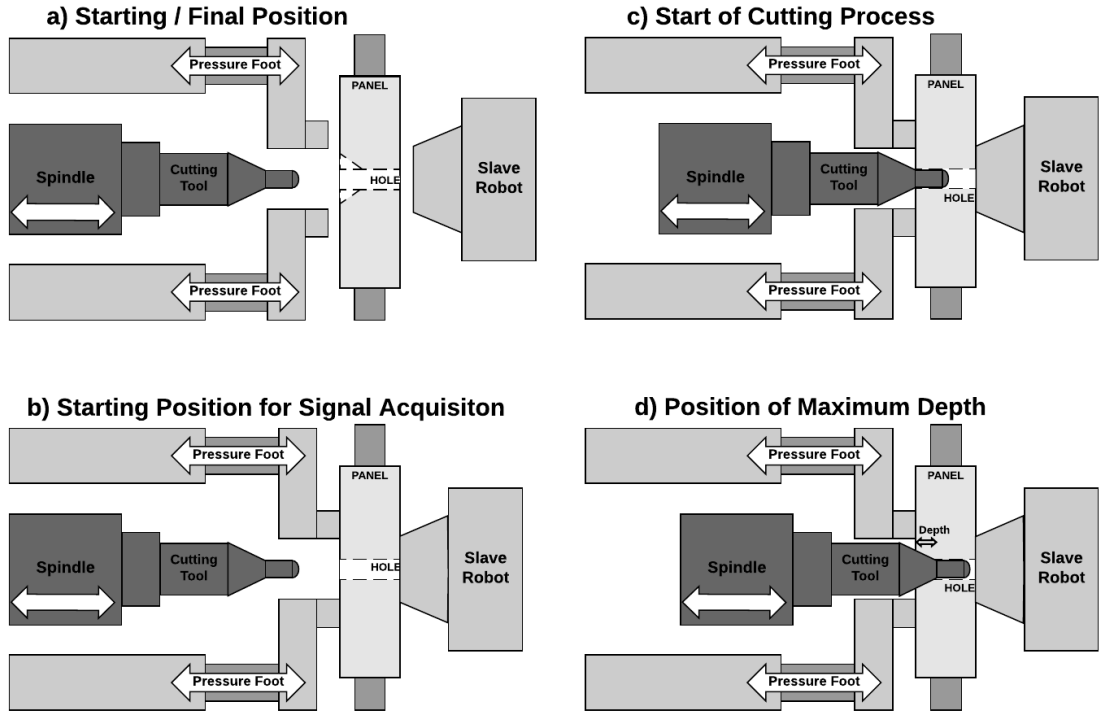

Figure 3.2 Schematic Diagram of Robotic Countersinking Process.

The diagram illustrates four snapshots of particular interest during the cutting cycle. Figure 3.2a shows the robots' waiting positions: both robots are located over a hole and they are either about to clamp the panel and start cutting or they have just machined the hole and

then unclamped the panel. In Figure 3.2b is depicted the moment when the PF clamps the panel while the slave robot maintains contact from the other side. The spindle is still in its home position. As will be clear later in Chapter 5, this moment is the starting position for the signal acquisition. Figure 3.2c presents the moment when the tool tip enters the hole and the cutting elements start removing material from the panel, i.e. start of the cutting process. Finally, in Figure 3.2d is illustrated the moment when the maximum countersink depth is reached and the spindle is about to retrieve to the home position. For the particular problem considered in this study (ref. to Section 4.1), this depth level defines the process output variable and, therefore, the performance of the whole process is measured in terms of the depth variation from the nominal target value. The objective is to maintain these variations to a minimum respecting the process tolerances.

The two robots' end-effectors were specifically designed by the AMRC to enable the robotic cell to achieve the required process performance. Therefore, a number of different components had been installed and then integrated into the overall process controller with the intention to improve the robots' positional accuracy during machining.

3.1.1 Master Robot End-Effector

The end-effector of the master robot is shown in Figure 3.3. It was a multifunctional drilling head equipped with various hardware, sensors and actuators, which allowed the robot to perform the machining process. The main components included (see the labels in the figure): (1) a clamping unit for system stabilisation, (2) a spindle drive unit, (3) a vision guidance system and (4) an inspection unit. These are further detailed in the following sub-sections.

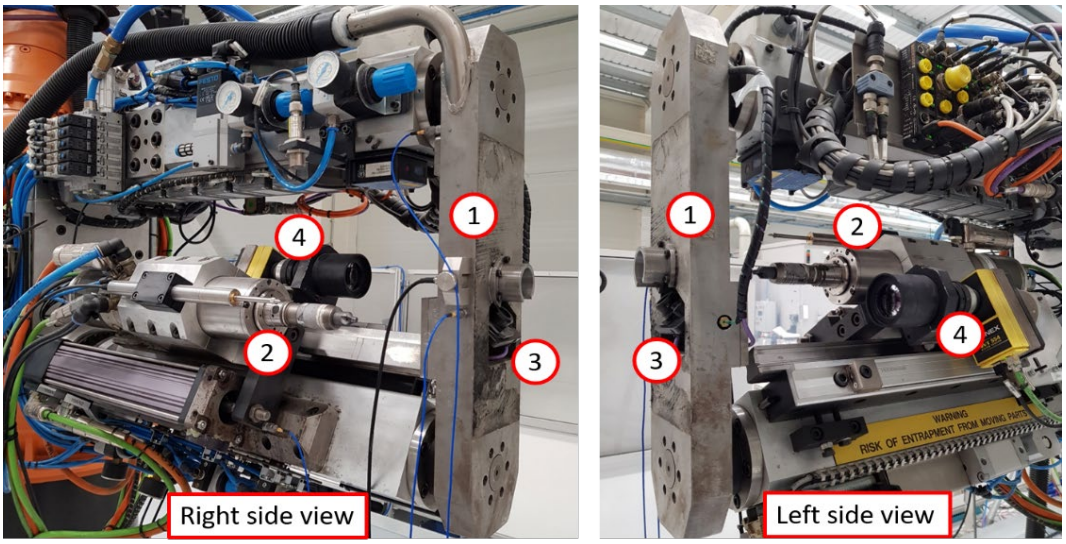

Figure 3.3 End-effector of the Master robot (both sides).

3.1.2 Clamping Unit

The clamping unit consisted of a PF with two air-driven rams as shown in Figure 3.4. It was used to provide precise pressure to the panel before the cutting commenced, which ensured that the relative position of the PF with respect to the panel remained constant during cutting (minimal panel movement). In addition, to react the process load, the slave robot also provided a constant force to the other side of the panel, acting as an anvil and ensuring that the panel stayed in position. This helped to increase the overall stiffness of the robots and stabilise the process since the errors in robot positioning due to the cutting forces were kept at minimum. A Micro-Epsilon laser displacement sensor with a precision of $3\ \mu m$ and a travel distance of $50\ mm$ was installed in proximity of the PF to measure the displacement of PF when it touches the panel. The value was then sent to the spindle drive controller to specify the target depth of the process in order to obtain the desired depth of cut.

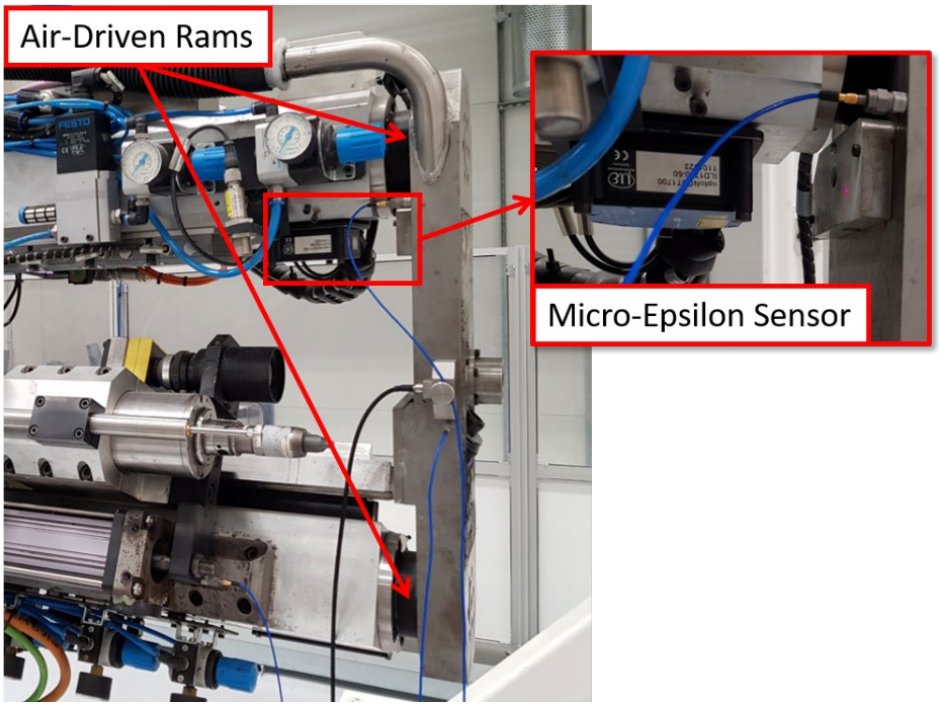

Figure 3.4 Clamping Unit: Pressure Foot (PF) and Micro-Epsilon Laser sensor.

3.1.3 Cutting Unit

The cutting unit handled the countersinking work. It consisted of a Precise spindle (with HSK32 spindle interface), a lead-screw driven feed drive and a 2-flute Exactaform cutting tool. The feed displacement was controlled by a Heidenhain glass scale encoder with an accuracy of less than $0.01\ mm$. The structure is shown in Figure 3.5.

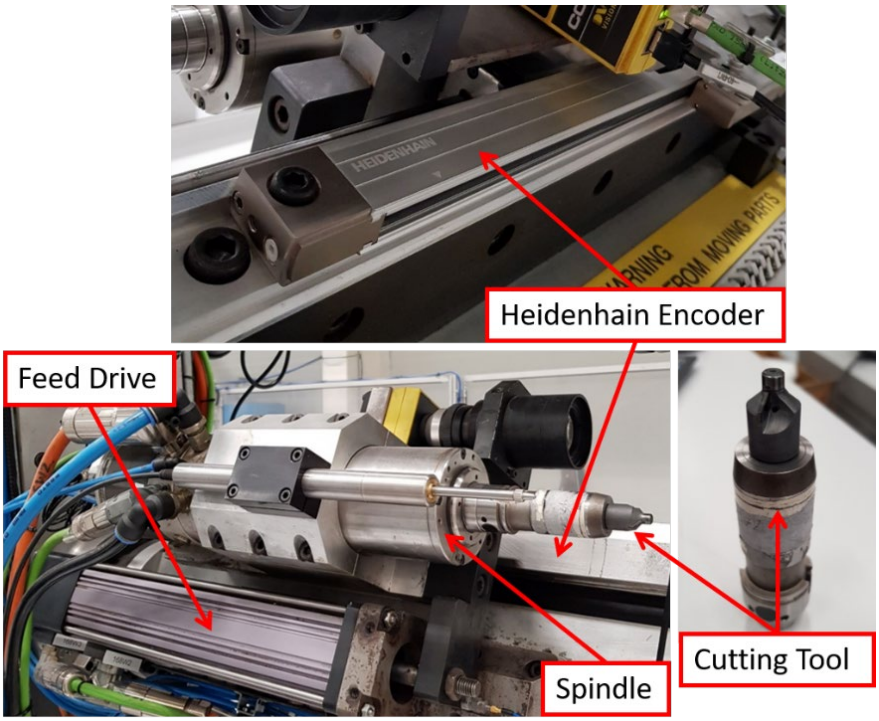

Figure 3.5 Cutting Unit: Feed Drive, Spindle, Cutting Tool and Heidenhain Encoder.

3.1.4 Lucana Aero Vision System

A Lucana Aero vision system was applied for precise localisation of the panel's pre-drilled holes in the Cartesian coordinate system. The system used two laser projections as a triangulation sensor for measurement of the surface normality. A schematic diagram of the localisation method of the Lucana Aero sensor is shown Figure 3.6. The two red crossed lines are the laser projections used to calculate the surface normality.

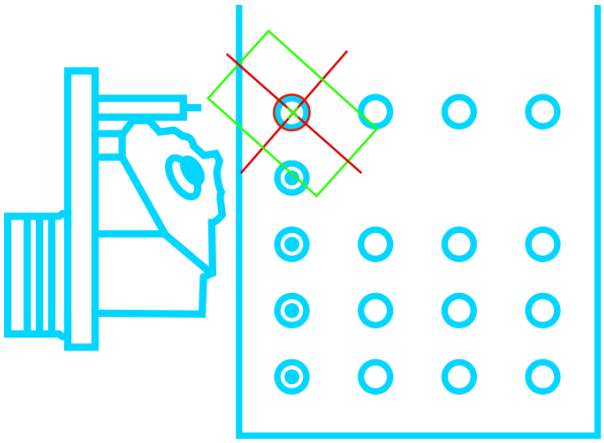

Figure 3.6 Lucana Aero Diagram.

Mounted inside the PF body in the proximity of the cutting tool, the camera guided the robot to precisely normalise to the surface of the panel and accurately find the centre of the hole

for positioning. Figure 3.7 shows the camera mounting position, a snapshot of the software output during the localisation procedure and the panel with 44 pre-drilled holes.

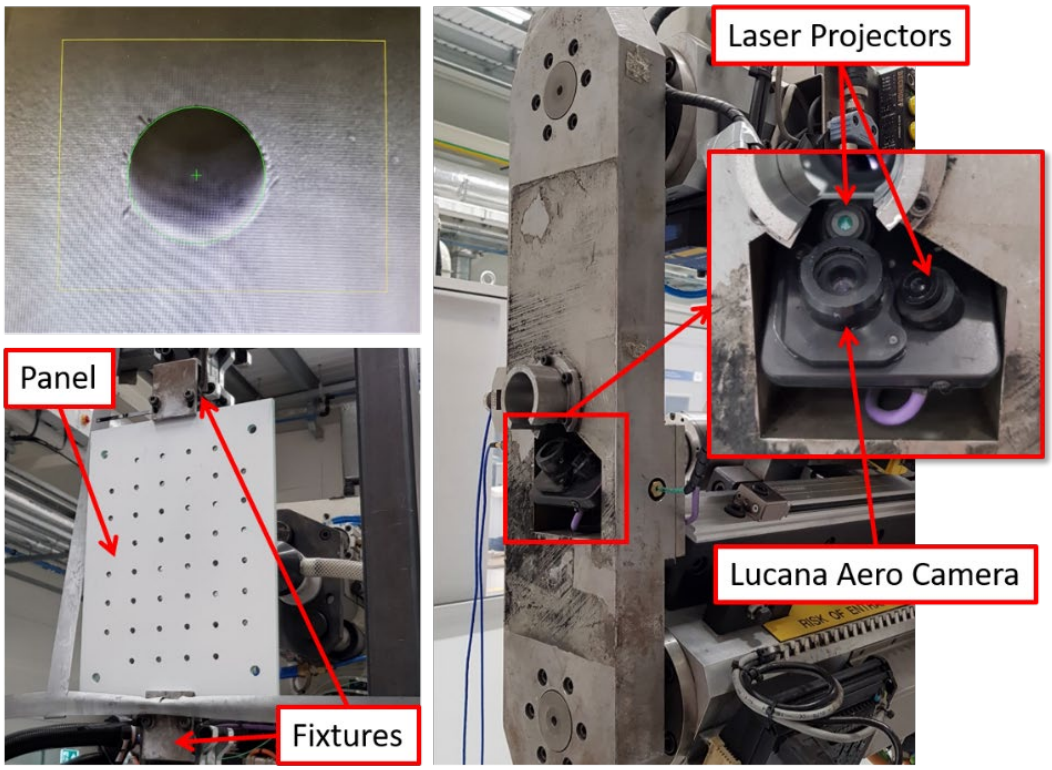

Figure 3.7 Lucana Aero Vision System, Hole localisation example and Panel with 44 holes.

After normalising over the hole, the camera identified the centre of the circle as shown in the figure (top left corner), and then it adjusted the robots' position before starting cutting.

The panels used during the experiments were Carbon Fibre Reinforced Polymers (CFRP) of dimensions 30x21x7mm with 44 pre-drilled holes distributed in a regular rectangular grid as illustrated in the figure (bottom left corner). Prior to the experiments, all the panels were paint in white in order to facilitate the vision analysis.

3.1.5 Inspection Unit

The inspection unit, shown in Figure 3.8, consisted of a Cognex camera with telecentric lenses. Telecentric lenses produce images with the same apparent size of the objects regardless of their distance from the camera. This property was particularly useful for inspection of the true countersink depth at the end of the process. It allowed for accurate measurements of the size of the machined hole and countersink maximum diameters, which, assuming a constant chamfer angle, could then be used to compute the countersink depth.

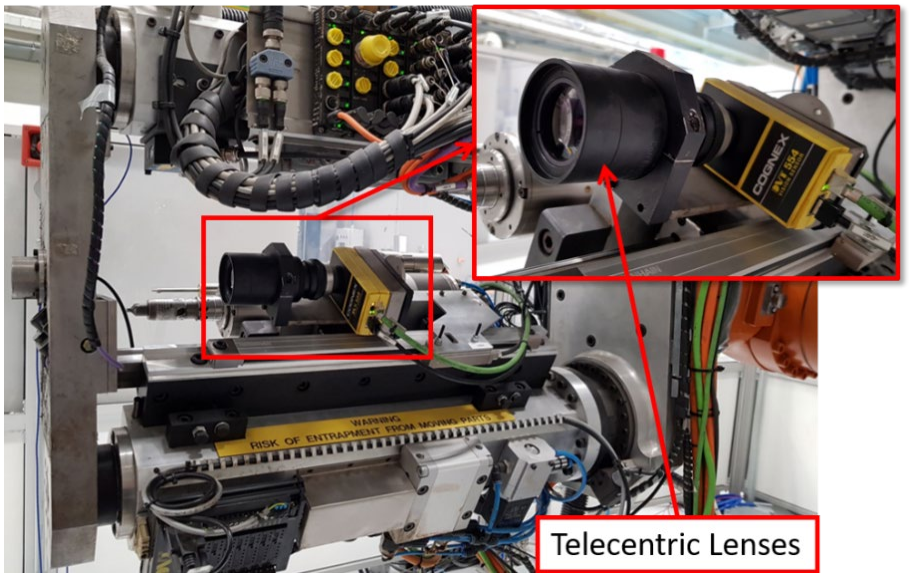

Figure 3.8 Inspection Unit: Cognex Camera with Telecentric Lenses.

A pivot drive was used to align the camera with the hole and take an image. It switched between the tool and camera position depending on the current step of the process: tool for cutting and camera for inspection. This functionality is shown in Figure 3.9.

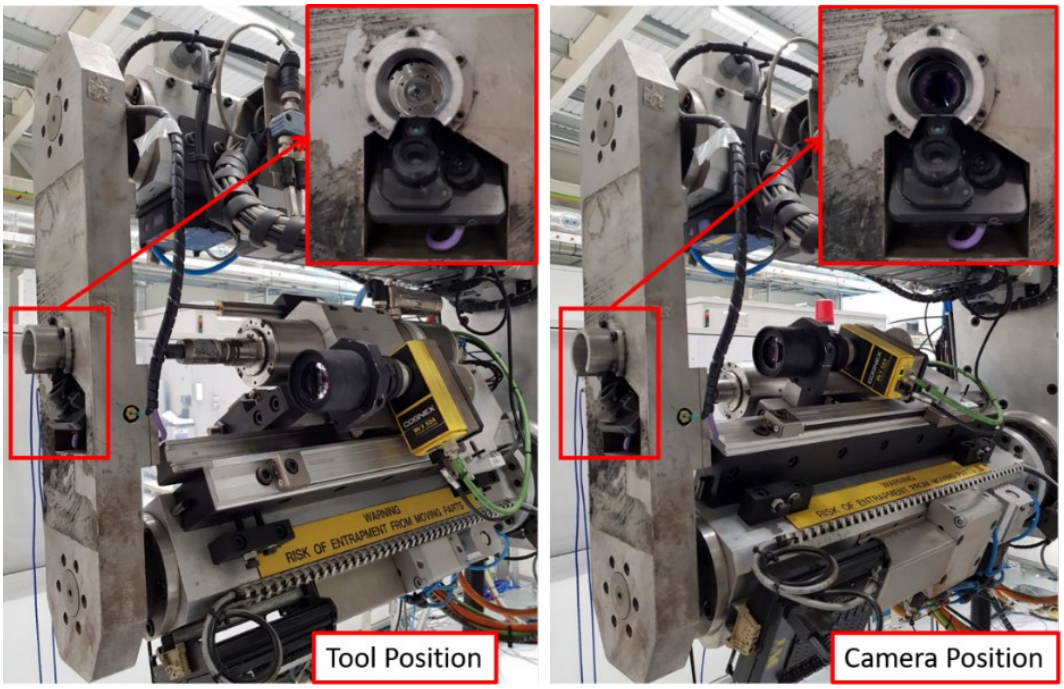

Figure 3.9 Tool and Camera positions for the inspection unit.

Note that the camera was attached to the spindle and, once aligned with the hole, the feed drive moved it towards the panel to take the image. This image was then processed to compute the hole diameter, providing therefore a direct measurement of the process output (ref. to Section 3.3 for further details).

3.1.6 Slave Robot End-Effector

As previously mentioned, the role of the slave robot was to react the process load and provide the required support to hold the panel in position during machining. The use of two robots, and, in particular, the application of the clamping system, allowed working parts with different sizes and shapes (greater flexibility), without paying for expensive holding fixtures (lower costs). Therefore, the robotic cell could accommodate most of the aircraft parts that typically consist of large panels with curved surfaces.

The end-effector of the slave robot is shown in Figure 3.10. Like the master, it had various components installed during the development phase, but most of which were not currently in use. The main units of the current functionality included a Kistler load cell that was used to detect the panel's contacting force and a swarf extraction unit.

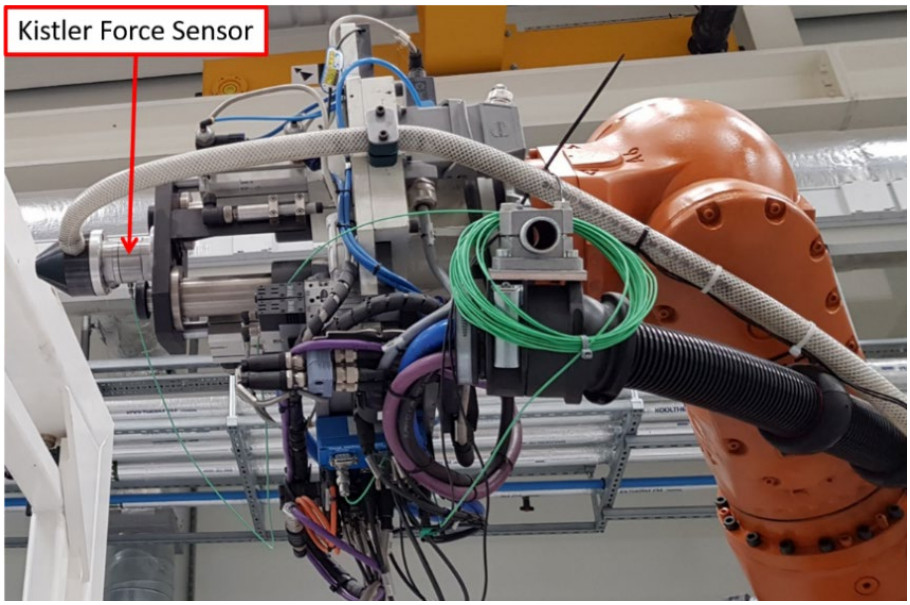

Figure 3.10 End-effector of the Slave robot with Kistler Load Cell.

The Kistler sensor was used to sense when the robot touched the panel and it sent a stop signal to the robot controller, which then ensured the robot brakes were activated during the whole cutting process.

3.1.7 Swarf Extraction Units

Both robots were equipped with extraction units in order to effectively remove the excessive dust and swarf produced during cutting. This was particularly important for the inspection step, since the quantity of swarf could seriously compromise the image quality and lead to wrong camera measurements. Figure 3.11 indicates the position of both extractors.

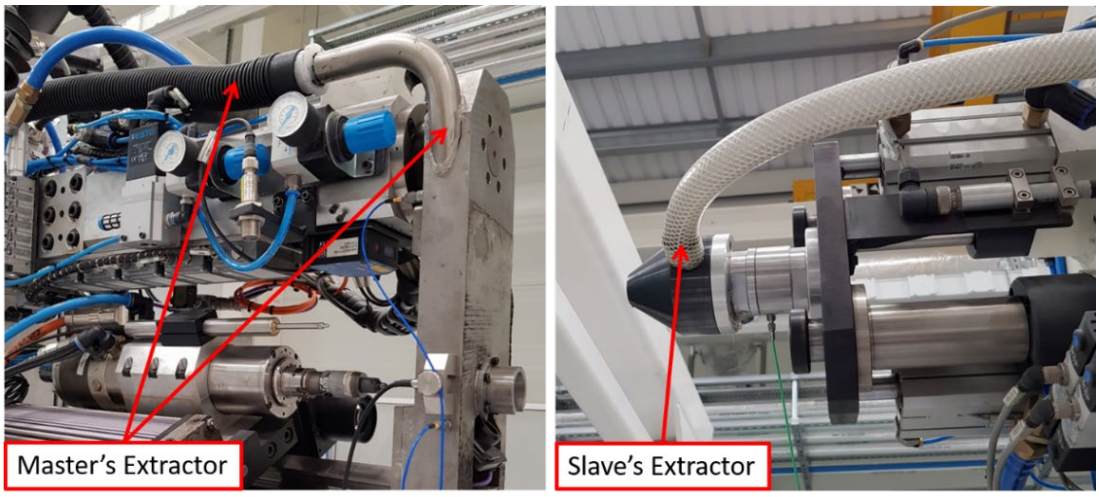

Figure 3.11 Swarf Extraction Units for Master and Slave robot.

Note that the master's extractor tube passes into a cavity through the PF body to access the cutting area close to the tool tip.

3.2 Machining Process Steps

This section presents a detailed description of all the steps involved in the machining process. These steps were accurately designed by the AMRC following extensive trials during the development of the robotic cell. The experimental work conducted here focused on improving the current process performance without causing any major modifications of the countersinking process itself, consequently the original settings were preserved.

The process steps of the above countersinking robotic cell are as follows:

1. Both robots approach a pre-drilled hole (from either side).
2. Lucana Aero sensor normalises over the hole.
3. Lucana Aero roughly locates the hole and guides the Master's end-effector over it adjusting along X and Y direction; Slave robot follows Master movements and gets in position (from the other side).
4. Simultaneously:
   a. Lucana Aero sensor adjusts Master X, Y, Z position and normality to tight tolerance.
   b. Slave robot moves towards panel until Kistler load cell detects contact and stops.
5. Slave is stopped; Master fires the PF (air rams move forward) to pre-defined force to contact the panel.

6. Spindle (kept to cutting speed all the time) is feed towards the surface, stopping at a set distance before it (based on PF movement forward, measured previously by Micro-Epsilon laser).
7. Spindle moves to countersink depth, as defined by the program, computed from PF movement (recorded before the spindle moved forward) and the surface curvature (where appropriate/provided by Lucana Aero sensor).
8. Spindle advances at a set feed rate (however, due to cutting forces, the feed rate is observed to vary from the nominal value).
9. Spindle waits at depth for a pre-defined dwell period.
10. Spindle withdraws to home position.
11. Master robot unclamps (air rams withdraw).
12. Slave robot retreats from panel.
13. Process moves to the next hole.

All the above steps are repeated for each hole of the panel (44 in total in the experiments conducted here). The variables of the process are categorised as follows:

- Controllable input variables: Target depth specified to the spindle feed drive.
- Uncontrollable variables (measured): Direct measurement of the countersink depth (by camera or manual inspection).
- Uncontrollable variables (not measured): Tool wear, assumed negligible.
- Constant input variables: Process parameters such as the cutting speed, feed rate, dwell time and PF force.

The following two outputs are provided by the process:

- In-process responses: Sensor signals collected during machining (monitoring system).
- Post-process responses: Diameter of the machined hole (inspection unit).

With reference to the above process variables, this work will exploit the experimental space defined by the controllable variable (target depth) to train process models that can accurately and robustly predict the post-process responses (countersink depths) from the data obtained by the in-process responses (sensor signals). Having the prediction of the actual countersink depth available at the end of the cutting process, would reduce or avoid completely the need for post-process quality inspection (manual or automatic), saving valuable time.

3.3 Countersinking Metrology

The main output variable of the above countersinking process (i.e. the process KPI) is the final depth value obtained in correspondence of the maximum displacement of the spindle towards the surface of the panel, just before it retrieves back in the home position. This value is commonly referred to as the countersink Depth-of-Cut (DoC), as illustrated in Figure 3.12.

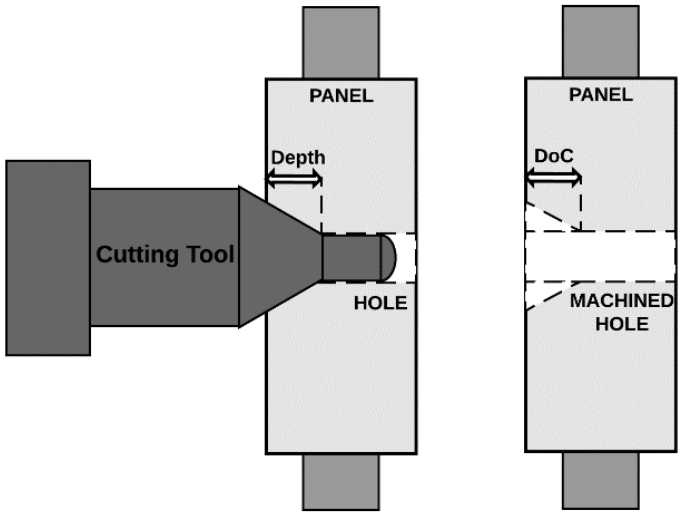

Figure 3.12 Countersink Depth-of-Cut (DoC) illustration.

In this work, two different ways to measure the DoC values of a finished panel were considered. The first involved a manual inspection at the end of the process using specific hardware (countersink gauge) to measure and then record the DoC value of every single hole. This method is considered to be the traditional measurement approach for this type of applications because it is both simple and intuitive, and, most importantly, it allows the machining process to complete with shorter cycle times (no further actions from the robots are required). On the other hand, when the number of holes to inspect is large, the manual inspection can become a long and tiring process with higher probability of human errors. A limitation of this method for the work conducted here was that it couldn't be applied online since the panel needed to be physically removed from the fixture in order to take the measurements.

The second method involved the inspection unit presented in Section 3.1.5, which used the Cognex camera to take images of the surface of the panel once the cutting process had finished. Note that the camera didn't measure directly the countersink DoC, but it provided an image of the machined hole, which was then processed to compute the DoC value by the following formula:

$$DoC = (R - r) * \tan(90^0 - \alpha/2) \quad (3.1)$$

Where R and r are the outer and inner radii of the machined hole, respectively, and α is the cutting tool angle (in degrees), as indicated in Figure 3.13.

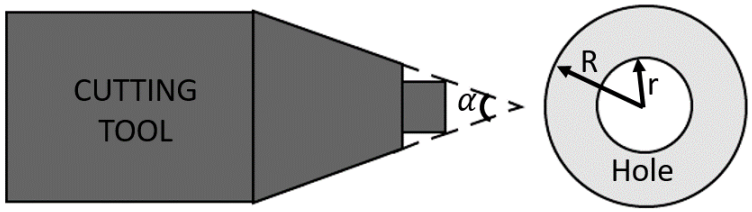

Figure 3.13 Diagram of the Cutting Tool and a machined Hole.

The angle α is a characteristic of the cutting tool and it can be retrieved from the tool drawing. The tool used for all the experiments performed during this work had an angle $\alpha = 100°$ with a tolerance on the angle of $0.5°$. The inner radius r was also known: the size of the pre-drilled holes was 6.35 mm with 5 µm of tolerance, therefore their radius was $r = 3.175\ mm$. The outer radius R was obtained by the camera image taken at the end of the cutting cycle for each hole individually. A snapshot of the outcome from the camera software is shown in Figure 3.14.

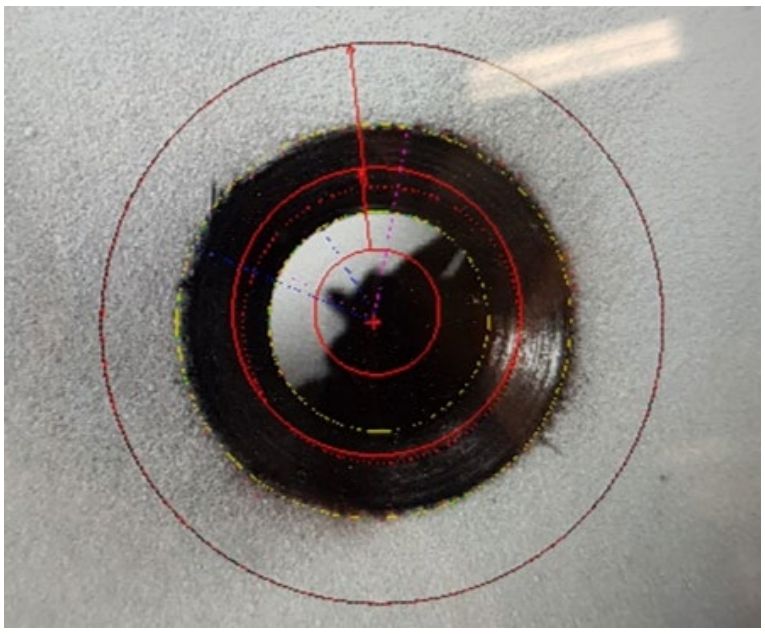

Figure 3.14 Snapshot of Cognex camera inspection.

The software searches the best circle fit for the outer perimeter of the hole (the large yellow circle shown in the figure) and computes its radius. The red marks identify the area of the search: for the outer perimeter the software searches the area between the medium and the

large red circles. This method was accurate (with maximum calibration error of 0.036 mm), provided fast measurements and could be implemented online for automated inspection of the holes, however, it involved adding a few extra steps in the process that inevitably increased the cycle time. These extra inspection steps were added in between step 11 and 12 of the list presented in Section 3.2 (ref. to Figure 3.9 for the inspection unit):

11. Master robot unclamps (air rams withdraw).
    a. Pivot drive switches to camera position.
    b. Spindle (and camera) advances towards the panel and an image is taken.
    c. Spindle retrieves back to the home position.
    d. Pivot drive switches to tool position.
12. Slave retreats from the panel.

The total time of the inspection routine for a single hole (i.e. to perform the above steps a-d) was approximately 8 seconds that were added to the overall process cycle time. Therefore, given the importance of the process throughput in a production environment, this measuring method is not practical for panels with a large number of holes (not in the long run at least), but it can be applied in research environment for testing purposes. Moreover, to obtain accurate measurements, particular attention is required for the camera inspection. Common sources of errors include the different lighting conditions, the colour of the surface of the panel (contrast between the inner and outer area), the camera settings for the inspection routine (circle fit parameters) and the removal of swarf and dust produced during cutting. Usually, some preliminary tests and preparation of the panel's surface are necessary in order to find the optimal camera settings and minimise the inspection errors.

Independently of the measuring method, the current countersinking process presented some limitations. During the development phase of the technology by the AMRC, the robotic cell has had several preliminary cutting trials performed outside this study, in which different cutting methods and cell configurations were tested. The major issue encountered during those trials was related to the countersink depth of the machined holes. Although the use of the clamping system increased the overall process stability and the spindle movement was precisely controlled by the Heidenhain encoder (ref. to Figure 3.5), the final DoC values were different from the nominal target (i.e. the expected value). In particular, after manual inspection at the end of the process, it was found that random holes exceeded the nominal depth value very close to the process tolerance of $0.2\ mm$. This means that there was a higher risk of scrapping the entire panel: it was sufficient just a single hole with depth over

the process tolerance. The reason of this variation was attributed to the movement of the slave during machining, which failed to hold in position due to the process load, but further investigation was necessary.

To identify the causes of the depth variation, it was proposed the installation of some extra sensors for monitoring purposes. This thesis expands the approach further by applying the concepts of the IMS as discussed in Section 2.2 of the literature review.

4 RESEARCH METHODOLOGY

This chapter presents the research methodology applied in this work. It first defines the robotic countersinking problem in Section 4.1. Then, the approach proposed by this thesis is presented and discussed in Section 4.2. The requirements of the monitoring system are outlined in Section 4.3, with particular focus on its practical application in a production environment. Finally, the system design considerations for the fulfilment of those requirements are discussed in Section 4.4.

4.1 Robotic Countersinking Problem

In order to understand the complexity of the countersinking process described in Chapter 3, it is useful to highlight some aspects of the technical challenges that inevitably introduce errors (variations) in the process response variable (i.e. the countersink depth). In the experiments conducted by this work, these variations were observed to be in a range of $\pm 10\%$ of the required depth, which, for targets of more than $2\ mm$, would lead to errors over the process tolerance. The main technical challenges are presented below:

- Despite the application of the PF to react the process load, the low static and dynamic stiffness of the robotic structure still has a major influence in machining holes with countersink depths that vary from the specified target.
- The complexity of the system is further increased due to the fact that the two robots collaborate to dynamically hold the panel in place during cutting (the panel fixture compensates just for the gravity forces). Unlike the Master that uses the PF to clamp the panel, the Slave robot relies on its structure stiffness to react the process load.
- The robot position in X and Y axis is not a cause of concern due to the high accuracy of the Lucana Aero camera in the localisation of the hole. In contrast, the Z position (along the feed direction) is computed from the PF distance measurement performed by the micro-epsilon laser (ref. to Figure 3.4) before the cutting commences. Therefore, the spindle Z movement is subject to measurement errors and/or any displacement of the panel during the cutting process (i.e. after the PF distance was taken).
- The spindle movement towards the panel is controlled by the feed drive with high precision, however, as illustrated in Figure 3.7, the fixture of the panel does not completely isolate the panel movement; it relies in the slave robot to contrast the

machining forces. This can introduce small variations of the panel surface along the Z direction as the tool progresses, which are not taken into account by the spindle drive controller, resulting in errors of the obtained countersink depth.

The case study problem considered in this work consists of developing a monitoring system based on data-driven process models (ML approach) that can:

1. map the sensor signals (input variables) to the countersink depth variations (output variables), providing therefore an in-process output prediction;
2. devise a method to keep these errors at minimum and within the specified tolerance of $\pm$ 0.2 mm.

Given the extra complexity that two robots add to the manufacturing process, a suitable approach that considers the overall effect of several sources of errors in the process output variable is necessary. In this work, a probabilistic framework is proposed, which naturally deals with uncertainties associated with the input data (sensor signals) and output variables (process responses) of a ML model.

It is also important to note that a supervised learning approach would require a relatively high number of training data (i.e. multiple runs of the process) in order to obtain accurate predictions of the process output variable. Moreover, since this data needs to be a representative example of the process behaviour, a relatively large variety of cutting trials (from under- to over-target holes, as well as normal holes) is necessary. This would require a dedicated experiment for collection of all the training data during which the robotic cell could not be operated for normal production, wasting valuable time and material. Consequently, obtaining the necessary data for training is one of the main challenges that supervised learning approaches face in industry, especially in manufacturing processes of high value aerospace components.

In this work, the problem of collecting the training data is addressed by introducing a direct measurement (inspection camera) to obtain the output variable while operating as normal, without interruption of the process. As previously mentioned, the in-process inspection routine adds extra time to the process cycle, but it also allows to expand the training set as more cuts are performed. Then eventually, the model will collect enough data to provide accurate depth predictions and, when this happens, the inspection step will not be necessary any more. Further details of the approach are given in the next section.

4.2 Proposed Approach

As previously noted, a typical manufacturing process consists of a machining (material removal) process followed by an inspection system for the quality checks. Usually these checks are performed at the end of the process and they may also involve removing the workpiece from the machine to a dedicated inspection area. In the case study considered here, the inspection step was included as part of the overall process cycle in order to obtain a direct measurement of the countersink depth as soon as the hole was machined. The inspection result was then compared with the depth target previously specified to the spindle drive controller and the depth error was calculated (i.e. the process output variable). With reference to Section 3.2, the process input variables consisted of all the machining parameters (constant variables) such as the cutting speed, feed rate, dwell time, PF force, etc., as well as the controllable variable of the target depth that defined the experimental space for the supervised learning approach. Different depth targets would produce different levels of cuts with a multitude of countersink depths and sensor signals, providing therefore the necessary data variability to train the process model.

A schematic diagram of a monitoring system based on a supervised learning approach for prediction of the countersink depth is shown in Figure 4.1. Figure 4.1a illustrates the system during the training phase, while new pairs of sensor signals (input variables) and the corresponding depth error value (output variable) are collected. Figure 4.1b shows how the overall manufacturing process is different, once the system has collected enough data (trained model) and it is ready to produce accurate predictions.

The necessary data variability is ensured by adding a programmatically controlled disturbance variable to the process target depth. From the machining process perspective, this disturbance acts in a similar way as the robotic structure, causing variations of the process output variable that result in workpiece imperfections. For instance, selecting the disturbance value in a range of data up to $\pm 25\%$ of the target depth would simulate the process behaviour in extreme conditions, where the robotic structure presents very low stiffness and positional accuracy. Hence the name in the figure as robotic system disturbance.

It is important to note that in addition to the above robotic disturbance that is introduced programmatically for greater data variability, there are other sources of errors in the process output variable, as explained in Section 4.1. These include the clamping process, the measurement errors of the PF distance and the inspection procedure, the panel movement

during cutting, etc. However, their overall contribution in the final countersink depth is relatively small in comparison with the robotic disturbance, which is considered to be the main source of errors. Moreover, since this disturbance represents the influence of the robotic structure in the process output variable, it will depend on the particular robot joints' configuration. In the considered case study, the robots change their configuration every time a new hole of the panel is located, i.e. for every iteration of the process.

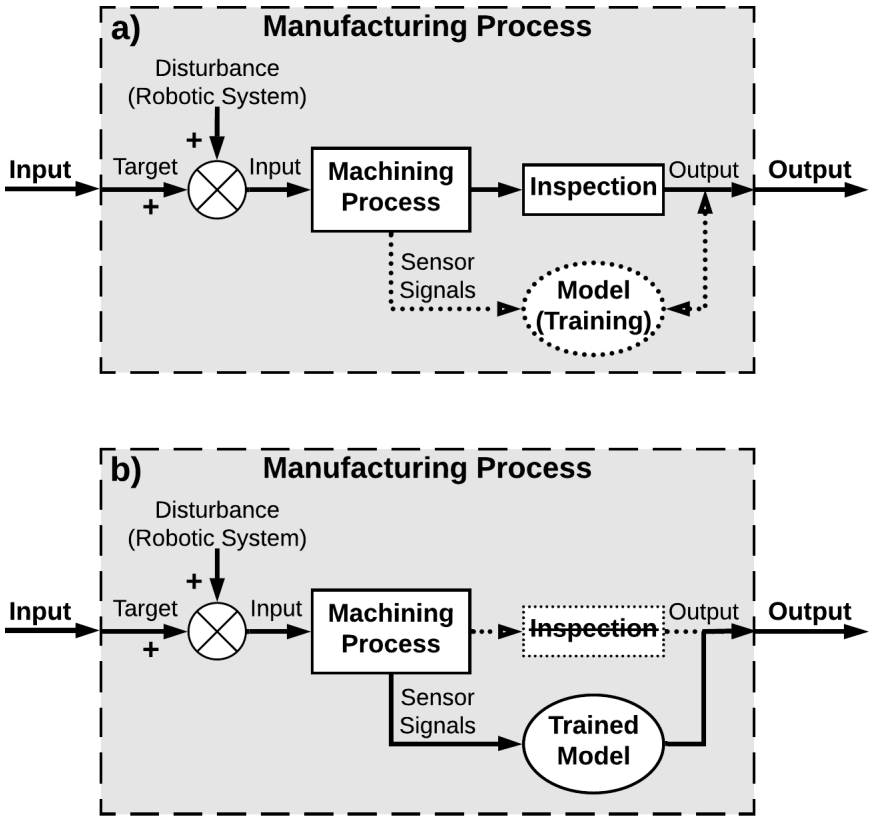

Figure 4.1 Schematic diagram of a supervised learning approach: a) process model during training and b) after the training phase has finished (trained model).

The target specified at the input of the overall manufacturing process is the desired (nominal) value of the process depth of cut. The proposed monitoring system is represented by the process model shown in the figure. It takes an input from the sensor signals collected during the cutting (i.e. the in-process response variables) and provides as an output a prediction of the inspection result (i.e. the post-process response variable). A supervised learning method is applied, therefore training instances are required to map the underlying relationship. During the training phase (ref. to Figure 4.1a) the model builds a dataset with the acquired sensor data (signal features) and the corresponding inspection measurement (output variable). A prediction is then provided given the current dataset and it is compared with the true measurement value. The accuracy of the model is assessed in terms of the difference

between the true and the predicted output value, called the prediction error. The model is considered trained when enough data has been collected for the prediction error to be within acceptable levels. A trained model is then ready to be applied for accurate prediction of the inspection measurement, as illustrated in Figure 4.1b. This ability of indirectly measuring the performance of the process without any inspection step or process interruptions can potentially reduce the overall process operational cycle time, increasing productivity. Moreover, the additional information on the quality of the final product provided by the sensors, could be used to support the operator in making appropriate corrective actions should they be required.

Note that, unlike the solutions reviewed in Section 2.1.2, the proposed method does not attempt to deal directly with the technical challenges of the robotic machining technology, but, its main focus is to provide in-process feedback of the quality of the machined part, based exclusively on sensory information acquired during normal operation of the process. Provided that the system has collected enough training data, then the operator can rely on the prediction output of the trained model to decide whether the inspection step is necessary. For instance, if the model predicts small variations from the nominal values indicating an acceptable part quality (with a certain level of prediction accuracy defined during the training phase), then there is no need for the inspection step to take place. The process can continue without the direct measurement, saving valuable time. The inspection step would be required only in the cases when the in-process prediction indicates not satisfactory results. Therefore, this method represents a valid solution for the production environment in the long run, reducing the process cycle time every time the inspection step is not used due to the accurate model prediction.

The approach presented in Figure 4.1 has however the following limitations:

- An experiment is required to train the monitoring system. The sensor signals are mapped to the corresponding process KPI (output variable) through a number of examples (i.e. iterations of the process), preferably in different conditions for greater variability of the data. As discussed previously, this can be obtained by means of controlling the target specified as input to the machining process. In particular, to modify the target, one can add a programmatically controlled disturbance variable that simulates the various ranges of robot operation resulting in different error levels.
- The predicted output is computed from the sensor signals collected in the current process iteration. This means that a prediction is only available once the cutting

process is completed. Depending on the type of process, from one iteration to the next, the robot might change its configuration modifying the disturbance added to the process, accordingly. Therefore, considering a generic case, the error observed at the end of the current process iteration is unrelated to the error of the next iteration. This means that from the process point of view, the disturbance value is not known at the beginning of the current iteration.

- An unknown (and unrelated) robotic disturbance prevents the monitoring system to make any corrections to the process target input of the current iteration, based on the errors observed in the previous iterations. The target correction strategy would be possible if an error map of the robot's operational range was available, following a similar approach to those reviewed in the literature. However, this would be a solution limited to the specific robot, performing a specific machining process and it would also require a large number of training data to build an accurate error map.

In order to overcome the above limitations, this thesis proposes to split the process iteration of the method presented in Figure 4.1 in two steps using different target inputs for the machining operation, namely Target 1 and Target 2 as shown in Figure 4.2. In particular, the first step executes one process cycle to a semi-finish level specified by Target 1. An output measurement (Out 1) is obtained by the inspection system at the end of Step 1. Then, the repeats the same operation is repeated for Step 2 to the finish level specified by Target 2. However, this time, the actual input target (In 2) of the machining process is adjusted by the previous step output (Out 1). The final inspection of the finished product provides the output of the overall manufacturing process, completing therefore one full process iteration. A schematic diagram of this multi-step process method is presented in Figure 4.2, following the same notation as before.

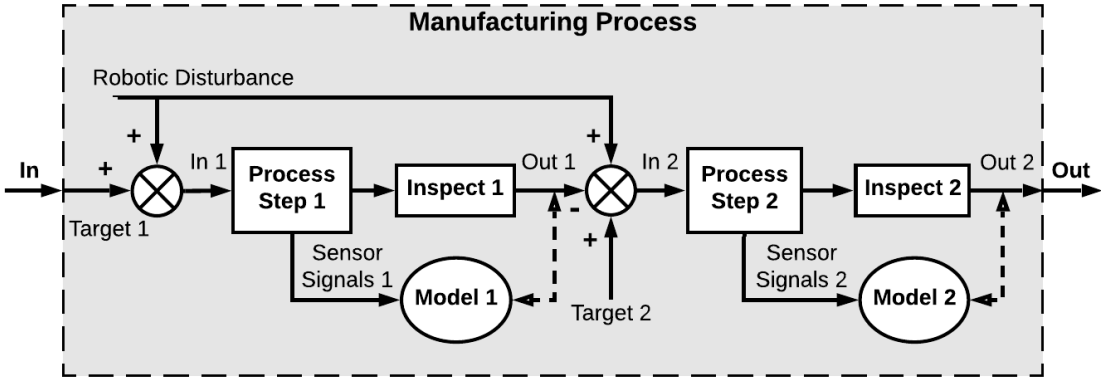

Figure 4.2 Schematic diagram of the multi-step process method proposed by this study.

Similarly, the disturbance that represents the errors introduced by the robotic structure into the machining process is added to both Target 1 and Target 2. Considering that the robot joints' configuration does not significantly change during one full iteration of the process (starting with Step 1 and then completing with Step 2), the same disturbance value is used for both process steps. It is important to understand that this disturbance variable is programmatically introduced in order to obtain the necessary data variability for the training phase by simulating extreme input conditions. The machining process will contribute to further errors of the output variable as result of other error sources, not necessarily coming from the robotic structure, which will depend on the particular application.

Note that in Figure 4.2, two process models (Model 1 and Model 2) are shown, one for each step, even though they refer to exactly the same physical machining process. This was included purely for notation purposes in order to distinguish the different input settings of the model between the two process steps. Therefore, the sensor signals acquired during Step 1 are used to predict the inspection output of Step 1 and similarly for Step 2. Note that due to the different initial conditions of the workpiece between the two steps (Step 2 starts with the part already machined to a semi-finish level), the sensor signals would have as well a slightly different signature.

With reference to Figure 4.2, the actual input target used for Process Step 1 is:

$$In_1 = T_1 + RD \tag{4.1}$$

Where T_1 is the desired (nominal) target of Process Step 1 (at semi-finish level) and RD is the programmatically controlled robotic disturbance. At the end of Step 1, the inspection system will provide the following measurement:

$$Meas_1 = In_1 + N_{P1} \tag{4.2}$$

Where N_{P1} represents the errors added to the output variable by the machining process when operating to a semi-finish level (i.e. the process noise). The difference between the inspection measurement and the nominal target defines the process output variable, which for Step 1 is obtained by substituting Eq. (4.1) into Eq. (4.2), as follows:

$$Out_1 = Meas_1 - T_1 = RD + N_{P1} \tag{4.3}$$

Therefore, the output of Step 1 is the total error introduced to the process response variable by two sources: the robotic disturbance (major contribution, manually introduced) and the machining process itself (random contribution of the process noise).

Following the diagram of Figure 4.2 for the second step, the actual input target of Process Step 2 can be expressed as:

$$In_2 = T_2 - Out_1 + RD \tag{4.4}$$

Where T_2 is the desired final target for the overall machining process (at finish level). This is also the nominal target originally specified in the previous approach when using one single process step (ref. to Figure 4.1). The inspection measurement at the end of Step 2 would be:

$$Meas_2 = In_2 + N_{P2} \tag{4.5}$$

Where N_{P2} represents the process noise added to the output variable when operating to a finish level. Then, substituting Eq. (4.3) and Eq. (4.4) into Eq. (4.5) gives the following expression for the final inspection measurement:

$$Meas_2 = T_2 + (N_{P2} - N_{P1}) \tag{4.6}$$

Eq. (4.6) represents the output variable of the whole process iteration (including both steps of the process) completed at the desired final level specified by T_2. Note that the error introduced to the response variable is dependent on the difference between the errors introduces by the machining process in Step 2 and Step 1. Consequently, if these two errors are assumed to be of a comparable amount (within negligible differences), then the final inspection measurement would be approximately equal to the desired final target, i.e. $Meas_2 \cong T_2$. This assumption may seem too restrictive, however, one can select Target 1 and Target 2 appropriately in order to obtain two process steps as similar as possible in terms of the forces involved and material removed from the workpiece, which will then help to make the assumption valid.

In the case study considered here, the errors in the final depth value coming from the machining forces were minimised by the use of the PF. Although cutting at different levels of target depth would produce different levels of process load, the PF force magnitude was much higher than the force differences between the two cuts. Therefore, the errors observed

in the final depth value would primarily depend on the clamping forces and the position of the current hole (i.e. the robot joints' configuration). This means that when cutting the same hole in different levels of depth, the errors that the process introduces in both cuts will be of a comparable amount, given that the robot configuration remains unchanged during the whole process iteration. In contrast, from one process iteration to the next, the robot changes its configuration to locate the new hole, introducing therefore a different depth error as result of the forces applied by the two robots when clamping the panel.

Since the machining operation is not performed to the finish level in one go, the monitoring system has now an opportunity to compensate for the robot's errors observed during Step 1 (at semi-finish level). In particular, the target specified as an input to the process in Step 2 is adjusted by the previous step's error (directly measured or predicted by the model) in order to ensure the desired finish level is obtained (ref. to Eq. (4.4)). If control is required in addition to monitoring, the system may drive the corrective actions autonomously by specifying the corrected target value (of the second step) into the robot controller.

A key aspect of the proposed solution is the fact that the data necessary for training of the process models can be collected while operating as normal, without any interruption or process down time. This means that a planned experiment for the training phase is no longer needed. Model 1 can be trained by controlling the robotic disturbance variable (RD) in order to provide the required variability of the data. The input target of Step 2 is adjusted by the output of Step 1 (inspection measurement), as shown Eq. (4.4). This means that the data variability of Step 1 is transferred to Step 2. Note that due to the target adjustment, Step 2 will aim to complete the operation at the final desired level (ref. to Eq. (4.6)), ensuring the process iteration is still valid. Therefore, the training phase can start from the first iteration of the overall manufacturing process using the inspection measurements to collect the data, while operating as normal. Once enough data have been collected, the process will use the predictions instead of the direct inspection measurements similar to the previous method.

To summarise, the proposed approach will add value to the overall process in two ways:

1) Substituting the inspection step with the one-step process option shown in Figure 4.1 (requires a specifically designed experiment for the training phase).
2) Correcting the target with the multi-step process option illustrated in Figure 4.2 (training data are collected while operation is running as normal).

The first point will reduce the overall process cycle time once the training phase is completed and the system is ready to provide accurate predictions. Moreover, the operator will have additional in-process feedback of the quality of the part provided by the process model.

The second point will exploit the capabilities of the monitoring system further, allowing for corrective actions to be included (automatically or manually) as part of the overall process cycle. In addition, it avoids a prior experiment or any process interruptions for obtaining the necessary training data.

4.3 System Requirements

An important aspect of the proposed monitoring system is its ability to be used in a production environment. To this end, the following practical requirements need to be addressed during the development of the monitoring system:

I) Hardware suitable for production environment.
II) In-process data acquisition and signal processing.
III) Ability to train during normal operation (no interruptions).
IV) Possibility to substitute inspection step with in-process predictions.

The first requirement addresses the need for suitable hardware. This means that characteristics of the equipment, such as the ease of installation, non-intrusive nature and robustness to external disturbances are highly desired. The choice of the sensors will depend on the particular process and monitoring scope, however, equipment with minimum disruption of the normal operation of the manufacturing process is preferred when possible. Moreover, the sensor signals acquired during operation need to be accurate, consistent and robust to noise and disturbances introduced from other machines around the shop floor. The cost of the equipment is also a key factor to consider. Expensive sensors will increase the overall cost of the system, making it less attractive to Small and Medium-size Enterprises (SMEs) and limiting its application in production.

The second requirement outlines the ability of the system for in-process extraction of sensory information. In particular, all sensor signals need to be perfectly synchronised (i.e. refer to the exact same point in time) despite their origin (sensor type) or sample rate. If signal segmentation is required, it should be performed automatically during data acquisition (based on previous knowledge of the signals' signatures) and without intervention of the operator. The signal processing and feature extraction steps are also to be performed in an

autonomous way, ensuring the sensor information is presented in the correct format for further analysis as soon as it is recorded.

The third requirement defines the ability of the system to collect training data during normal operation of the process, minimising the process down time for any scheduled interruptions for training purposes. An inspection step that provides a direct measurement of the process KPI is necessary for the training data to be collected during operation. Process models are to be trained on the current available dataset, which will continuously grow every time a new observation (inspection) is made. A selection of most relevant features should be included in the training procedure in order to achieve better performances and faster training times as the size of the dataset increases.

The forth requirement refers to the possibility of building process models that can provide accurate and reliable predictions of the process KPI, given the current dataset. This would make it possible to avoid the use of the inspection step completely, once the process models have had enough training data to provide accurate predictions. It is important to note that a measure of the confidence of the model in the predicted value is required in order to assess the validity of the prediction and the optimal training size.

4.4 Design Considerations

In the light of the system requirements outlined in Section 4.3, the implementation steps of the monitoring system proposed by this work need to answer to the following questions:

- What sensor will be installed and why?
- What signal analysis will be performed?
- How will the relevant features be selected?
- What ML technique for creation of the models will be used?
- How will the models' confidence be assessed?

The first question to address is about the selection of a suitable sensor system. The monitoring system requires the sensors to be applicable in a production environment. With reference to Section 2.2.1 of the literature review, good candidates for this purpose are power/current transducers, AE sensors and accelerometers, being both low-cost and non-intrusive sensor technologies (ref. also to the Table 2.1). They can be typically installed in the proximity of the cutting area and often do not require specific modifications of the robot's end-effector. Dynamometers, on the other side, despite the direct force measurements and

their popularity in machining process monitoring, are not suitable for the proposed system because of their cost (generally more expensive than the other technologies) and highly intrusive nature (they need to be integrated into the spindle holder).

The second question is related to the signal analysis. The system requires in-process data acquisition and extraction of sensory information. It is crucial therefore that special attention is dedicated to the selection of the hardware used for the data acquisition as well as the signal processing scheme (filtering, segmentation and feature extraction). In terms of the data acquisition hardware, the system requires robust (industrial standards) acquisition and synchronisation capabilities. A suitable choice for this purpose is a central data acquisition scheme that can handle signals from different sensor sources with various sample rates, ensuring the overall data synchronisation. In terms of the signal processing scheme, the system is expected to be applicable in a range of machining processes (with non-standard process KPIs), therefore flexibility is a key factor. To this end different filtering techniques as well as the signal domain transformations reviewed in Section 2.2.3 should be explored. Intuitively, extracting a high number of features will increase the probability of having among them those that contain relevant information to the particular process KPI.

The third question concerns the choice of a suitable feature selection method for the proposed monitoring system. This step is required to identify the relevant features (with respect to the particular process KPI) among all those extracted during the acquisition. Moreover, the feature selection improves the training time of the process models by reducing their input space. With reference to Section 2.2.4, possible choices for selection of the features consist of filter methods, which are both fast and intuitive, however they do not consider the ML algorithm when assessing the feature relevance. A wrapper or embedded approach would be more appropriate if the learning method is already decided, since the selection of the feature subset is driven by the model prediction accuracy.

The last two questions are both related to the selection of an appropriate ML technique for the creation of process models with accurate predictions of the process KPI. The choice of the ML algorithm will depend on the particular learning problem (classification or regression). A regression problem represents better the method presented here, since the process KPI is likely to be a numeric value. Probabilistic regression methods that naturally deal with uncertain inputs and outputs are ideal in this case. Moreover, the Bayesian framework offers a measure of the model's confidence in the predicted value, expressed in terms of the predictive covariance (ref. to Section A.1 in the Appendices for further details). This measure

can be used to understand whether the model is going far from the learnt state and provide new training data should they be needed.

Before presenting the implementation details of the proposed approach, it is useful to understand how such system can improve the performance of the robotic countersinking process considered in this work. In particular, improvement of the process performance can be achieved by either of the following:

- Reduce the inspection time of a single hole.
- Ensure the holes are machined within the limits in the first place.

The first point refers to the possibility of avoiding the current manual measurement of the DoC values at the end of the process. The monitoring system could be applied to learn the underlying relationship between the sensory data collected during cutting and the final DoC variation. This means that the system can be trained to predict the DoC value at the end of the cutting task, providing an immediate feedback of the quality of the hole. The manual checks can then be performed only to the panels already identified by the monitoring system as 'not good'. A finished panel is considered to be of good quality if the depth variation values of all the machined holes in it are within the process tolerance.

The second point refers to the possibility of exploiting the capabilities of the monitoring system further by using the multi-step process method presented in Section 4.1 (ref. to Figure 4.2). The target depth of Step 2 (at finish level) can be adjusted by the information collected from Step 1 (at semi-finish level) to keep the final DoC within the limits. In other words, the system can potentially capture the behaviour of the process during a first cut at semi-finish level and then indicate the necessary target corrections for the second cut to reduce the final depth variation.

Furthermore, for this particular case study, the requirements of the monitoring system outlined in Section 4.3 can be adapted as below:

1. Hardware suitable for a production environment.
    - Easy to install and non-intrusive sensors.
    - No major modifications of the cell design and/or process steps.
    - Accurate and robust sensory data in different workshop conditions.
2. In-process data acquisition and signal processing.
    - Synchronised acquisition of different sensor signals and sampling rates.

- Automatic signal segmentation and feature extraction.

3. Ability to train during the machining process.

- In-process direct measurement of the DoC value (camera inspection).
- Dataset should grow online with every new observation.
- Feature selection step to reduce training time.

4. Possibility to substitute the camera inspection step.

- Accurate and reliable depth predictions from sensory information.
- Assessment of the model's confidence in the prediction value.
- Suggestion of target adjustments in real time (according to depth predictions).

The following chapters will present the implementation steps, experimental work and learning methods proposed by this work for the development of a monitoring system to fulfil the above requirements.

5 EXTRACTING IN-PROCESS INFORMATION

This chapter presents the implementation details of the techniques applied to extract information from the machining process. An experiment was designed with the aim to test and validate the capabilities of the monitoring system proposed by this work taking into account the requirements discussed in Section 4.3. The details of the experimental setup, data acquisition system, signal processing methods and feature extraction techniques are discussed in the following sections.

5.1 Experimental Setup

In order to test and validate the functionality of the proposed monitoring system in the robotic countersinking case study presented in Chapter 3, the following experiment was designed as part of this thesis work. The experiment was divided in two phases, which were conducted in different times from each other. The first phase involved machining four panels of 44 holes each for a total of 176 holes, and, in the second phase, three more panels of 44 holes each (132 holes) were considered. Overall, seven panels (308 holes) were machined. The reason for this division was due to some issues encountered during Phase 1 with one of the sensors and the camera inspection routine. During the trials of Phase 1, one of the vibration sensors' cable was damaged and it was temporally disconnected. The sensor cable was replaced in time for Phase 2, in which all the sensors were perfectly functional. Furthermore, the camera inspection routine (ref. to Figure 3.14 in Section 3.3) presented some issues in detecting the hole diameter after the first cut, especially for smaller diameters (shallow holes). The reason for this was associated to the low quality of the paint used for the panels, which was stripped away from the hole during the cutting process producing dark patches around the edges. The surface of the panel was another factor that influenced the paint adhesion: the CFRP panels had very different surfaces, which introduced further errors to the inspection result. All these issues concerning the inspection routine were taken into account when selecting the three panels (P5, P6 and P7) used for Phase 2.

The purpose of the experiment was to provide the necessary data for the creation of process models that can accurately predict the obtained DoC values. The cutting method used in both phases of the experiment included two cuts for every single hole, where a direct depth measurement form the inspection camera was obtained after each cut. Although this choice increased the process cycle time considerably (two inspection routines were added for every

hole of the panel), it allowed collection of the actual DoC value after each cut and without interruption of the process. These depth values together with the corresponding recorded sensor signals were then used for training purposes.

In particular, the cutting cycle of a single hole followed the multi-step method presented in Section 4.2 (ref. to Figure 4.2) and it included the following steps:

- A first cut was performed to a pre-set (semi-finish) target depth plus an additional random perturbation (programmatically controlled) to obtain depth variability. The process steps remained the same as described in Section 3.2 up to step 11, when the master robot unclamps the panel (PF retreated), after the cut was completed.
- A direct measurement of the DoC value (semi-finish level) was obtained by the inspection camera (ref. to Section 3.3 for the extra inspection steps).
- Master robot fired the PF again, clamped the panel and performed the second cut to a set (finish) target depth adjusted by the error calculated from the previous camera measurement, according to the multi-step approach discussed in Section 4.2. The process steps from 5 to 11 (ref. to Section 3.2) were repeated for the second cut. Note that in between the two cuts both robots remained in position above the hole, thus there was no need to locate the hole again through the Lucana Aero sensor.
- Another direct measurement of the DoC value (finish level) was obtained by the inspection camera.
- Slave robot retreated from the panel and the process moved to the next hole.

An example of the 44-holes panels before and after countersinking is illustrated in Figure 5.1.

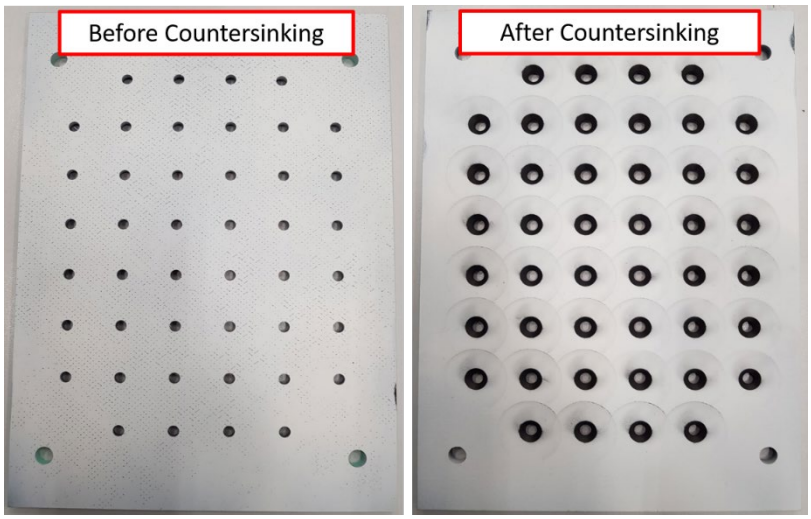

Figure 5.1 Panel with 44 pre-drilled holes used for the experiment: before (left) and after (right) countersinking.

The direct depth measurement obtained by the inspection camera provided the necessary output variable to train the system online, while running the operation as normal. Splitting the cutting process in two parts and collecting the sensor signals with the corresponding DoC value after each cut, permitted to train a process model for the prediction of two depth values: the semi-finish depth (d_1), referred to as Model 1 and the final depth (d_2), referred to as Model 2. As previously noted, Model 1 and Model 2 are not two different models, but they represent the different input conditions of the same physical machining process. The distinction was introduced for notation purposes to separate the analysis between the two steps of the process. Therefore, in the experiment, Model 1 used the sensor data collected during the first cut, whereas Model 2 used the data collected during the second cut. Moreover, the output error observed at the end of the first cut in terms of the depth variation from the nominal semi-finish target was used to adjust the input target of the second cut in order to obtain a final DoC value within the $\pm 0.2\ mm$ tolerance. In the experiment described here, the semi-finish target was set at $d_1 = 2\ mm$ and the final target at $d_2 = 2.74\ mm$.

The cutting tool used in the experiment was the same for all the cuts of Phase 1 and 2, starting from an almost fresh state at Panel 1. Unfortunately, there was no information available of the exact state of the cutting tool in terms of the number of cuts performed to date and, due to technical issues with the tool supplier, it was not possible to get a completely fresh tool ready for the experiment. At the end of Phase 1 trials, the flutes of the cutting tool were visually inspected for any damages or excessive wear and then the tool was carefully cleaned up and re-installed for Phase 2 trials. In this experiment, the tool wear was not constantly monitored and it was assumed to be negligible.

The random perturbation added to the depth target was introduced to obtain depth variability (deep and shallow holes) during the cuts. This allowed the model to learn how small variations of the output variable (the DoC value) were related to the input data (sensor signals) collected during cutting. For this experiment the added perturbation was a Gaussian $N(\mu, \sigma^2)$ with mean $\mu = 0$ and standard deviation $\sigma = 0.25$.

In all the cuts performed during the experiment, the final target level was adjusted based on the direct camera measurement of the DoC value after the first cut, however, the ultimate goal of the monitoring system was to use the depth prediction provided by the trained model instead. This would allow the system to avoid the camera inspection step and therefore return to the normal process speed.

The process parameters were set as follows: spindle speed $4976\ RPM$; feed rate $298\ mm/min$; dwell time $0.3\ s$ and PF force $500\ N$.

5.2 Data Acquisition System

The data acquisition system was designed taking into account the requirements of the proposed monitoring system presented in Section 4.3. In particular, the first two requirements outline the specifications of the hardware and software for extraction of the process information.

The hardware used for the experiment was chosen to be easy to install, minimally intrusive with the machining process and suitable for production environment. The installation of the sensors had to fit into the current design of the robotic cell, not requiring any major modifications of the cell's operation. This requirement made it difficult to find an optimal location for mounting the sensors, since the space near to the cutting tool was very limited due to all the other components as described in Section 3.1.

The following sensors were installed on the master robot end-effector:

- 3x One-axis High Frequency Accelerometers from PCB (type 621B40)
- 1x Acoustic Emission (AE) sensor from Kistler (type 8152C01)
- 1x Power Transducer from Caron Engineering (TMAC – Tool Monitoring Adaptive Control)
- 1x Linear Encoder (digital probe) from Sony (type DK50PR5)

The location of the sensors is illustrated in Figure 5.2. The power sensor is not shown in the figure because the transducer was located inside the PLC cabinet. The three phases of power cable to the spindle were fed through the device, which then provided an output voltage signal proportional to spindle power consumption. This signal was directly wired to the data acquisition device. The vibration sensors and the AE sensor had magnetic clamps and they could be easily mounted on the metallic surfaces of the spindle holder and the PF. The linear encoder was already installed for testing purposes as part of some previous cutting trials performed outside the work of this thesis. However, given its intrusive nature, it was not included in the production version of the cell currently used at BAE facility.

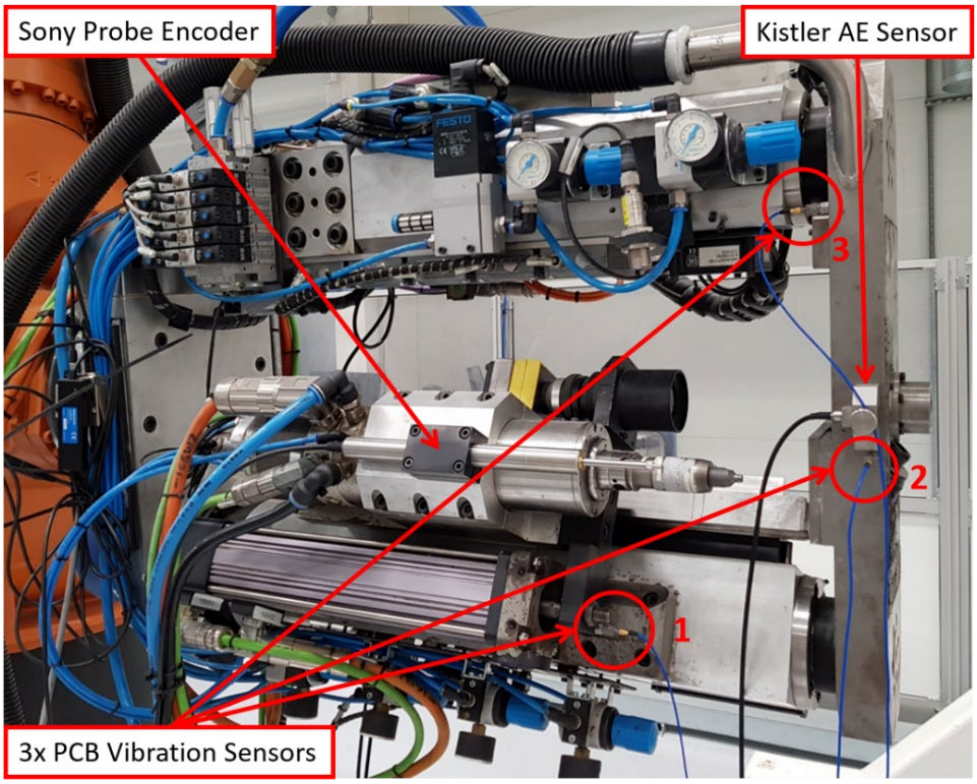

Figure 5.2 Location of the sensors used for the experiment: three one-axis PCB accelerometers, a Kistler AE sensor and a Sony probe encoder.

The first vibration sensor (denoted by VIB1) was located as close as possible to the cutting tool with its sensitive axis being aligned to the feed direction (Z axis) in order to detect the tool vibration during cutting. The second and the third were installed on the PF body: one (VIB2) measured the vibration in the perpendicular direction (X axis) and the other (VIB3) in the feed direction (Z axis). The Kistler AE sensor was mounted on the PF as close as possible to the cutting area to measure the acoustic emissions generated during the process. The Sony probe encoder was installed on the spindle holder to measure the linear displacement of the cutting tool towards the surface of the panel. In the experiment this encoder was used as a trigger for the acquisition of the other sensor data. The movement of the PF towards the panel when it is activated (i.e. just before the start of the cutting process) was easily detected by the probe signal. This event was ideal for automatically triggering the acquisition since it was repeated for every cut. In case of the production cell where the probe is not installed, the vibration signal measured by VIB3 could be used as a trigger instead, since it presented a clear spike when the PF touched the panel during clamping. The probe signal was recorded in all the cutting trials performed for this experiment, but not used for training purposes because this sensor was not available in the production cell and it would have limited the application of the proposed monitoring system to a purely laboratory solution. A summary of all the sensors selected for the validation experiment is shown in Table 5.1.

Table 5.1 Summary of the sensors used for the experiment.

Sensor Description and Manufacturer	Short Name	Location	Bandwidth	Sensitivity
High Freq. Accelerometer (PCB, 621B40, One-axis)	VIB1	Spindle Holder (Z direction)	1.6 - 30k Hz	10 mV/g (+/- 500g scale)
High Freq. Accelerometer (PCB, 621B40, One-axis)	VIB2	Pressure Foot (X direction)	1.6 - 30k Hz	10 mV/g (+/- 500g scale)
High Freq. Accelerometer (PCB, 621B40, One-axis)	VIB3	Pressure Foot (Z direction)	1.6 - 30k Hz	10 mV/g (+/- 500g scale)
Acoustic Emission (AE) (Kistler, 8152C01)	AE	Pressure Foot (cutting area)	50k - 400k Hz	57 dB ref 1V/(m/s)
Power Transducer (Caron Engineering, TMAC)	PWR	PLC controller (spindle motor)	0 - 50 Hz	Min 0.75 W/V (1.5 - 67 kW range)
Linear Encoder (Probe) (Sony, DK50PR5)	Probe	Spindle Holder (spindle linear displacement)	Counter Freq. Max 5 MHz	Res. 0.5 µm/pulse Accuracy 2 µm (Range 50 mm)

All the sensor were connected to a central data acquisition device (compact DAQ) from National Instruments (NI cDAQ-9178) and the data was collected and processed using NI LabVIEW software. The data acquisition system is shown in Figure 5.3.

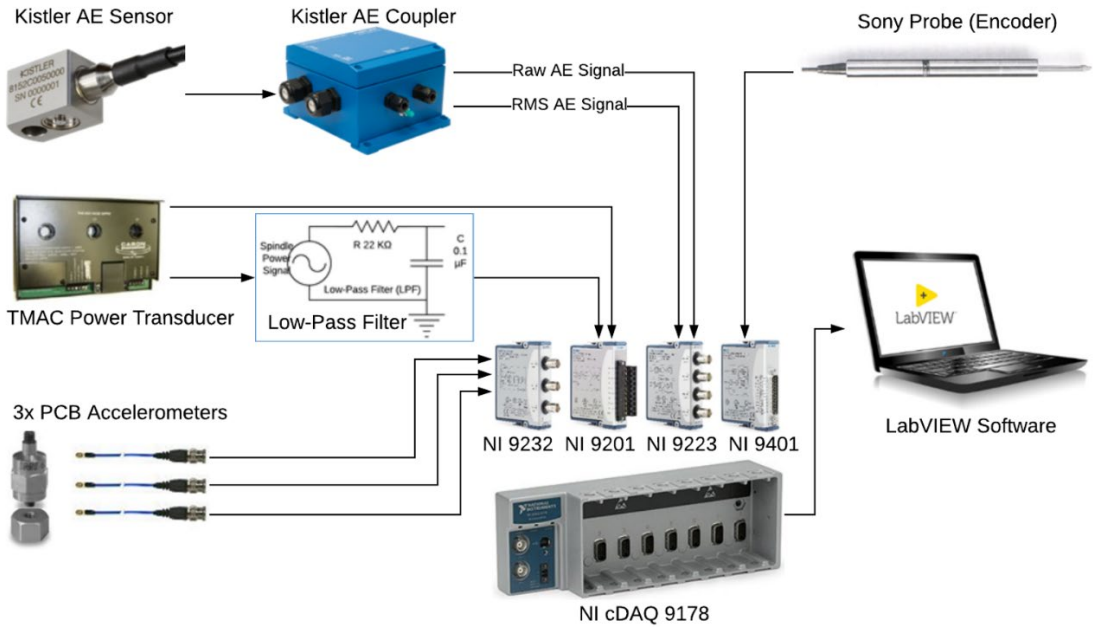

Figure 5.3 Data Acquisition System used for the experiment.

A different NI module designed to measure the specific type of signal at the required sampling rate was used for each sensor. The AE sensor was first connected to an AE coupler (Kistler 5125C0) designed to filter and process the raw data obtained by the sensor. The

coupler had a built-in RMS converter with selectable integration time constant and selectable gain. The default values of $1.2\ ms$ for the integration time constant of the RMS converter and 10 for the gain were selected in the experiment. The two output signals coming from the AE coupler (raw AE and the converted AE-RMS) were acquired through the NI-9223 module, which offered a maximum sampling rate of $1\ MS/s$ for each individual channel.

The power transducer signal was filtered by means a Low-Pass Filter (LPF) in order to remove the unwanted high frequency noise of the raw spindle power data. The filter was realised through an RC circuit with $R = 22k\Omega$ and $C = 0.1\mu F$, cut-off frequency of $F_c = 72.4Hz$. The two power signals (raw data and the LPF data) were acquired through the NI-9201 module that had a maximum sampling rate of $62.5\ kS/s$ for each individual channel ($500\ kS/s$ aggregate – 8 channels).

The three accelerometers were connected directly to the NI-9232 module (one sensor per channel), using the BNC cables provided by the manufacturer. Each cable had a length of 10 m. The module offered a maximum sampling rate of $102\ kS/s$ for each individual channel.

The digital signal of the Sony probe (encoder) was acquired through the NI-9401 (digital I/O) module, which could measure digital lines up to a maximum switching frequency per channel of $9\ MHz$ for 8 digital input channels.

The compact DAQ device (NI cDAQ-9178) was connected by an USB cable to a windows PC running LabVIEW 2017 with DAQmx software packages installed. This central acquisition unit made it possible to set a synchronised start for the sensor data collection and to simultaneously acquire several signals of different types and sample rates.

5.3 Digital Signal Analysis

This section examines the signals acquired during the cutting tests performed as part of the above experiment and presents the processing methods applied to the raw data. Previous analysis (not mentioned here) suggested a possible correlation of the actual duration of the cutting process (i.e. the time in cut) with the obtained DoC value, therefore the focus of the processing methods presented here was to convert the raw data into suitable features that could accurately represent the region of the signal when the cutting process was taking place. The signal processing techniques were based on advanced methods from the IMS literature, as reviewed in sections 2.2.2 and 2.2.3.

5.3.1 Signal Processing

All the signals recorded during a single cutting cycle were time series of 8 seconds. The probe signal was used as a trigger for the acquisition of all the data. The NI cDAQ device in combination with the NI software allowed for robust acquisition and perfect synchronisation of the data, despite using disparate sensor types and different sample rates.

An example of the time domain data recorded for each sensor is illustrated in the following figures (Figure 5.4 through Figure 5.9). The raw signals are shown in the left hand side (blue plots) and the RMS of the signals, computed at 100 samples per second ($0.01\ s$ integration time constant), is presented in the right hand side (red plots). The figures show the signals collected from both cuts of a typical process cycle (first for cut 1 and then cut 2 of the same hole, respectively), repeated for each hole of the panel (the names of the signals are also indicated in the figures).

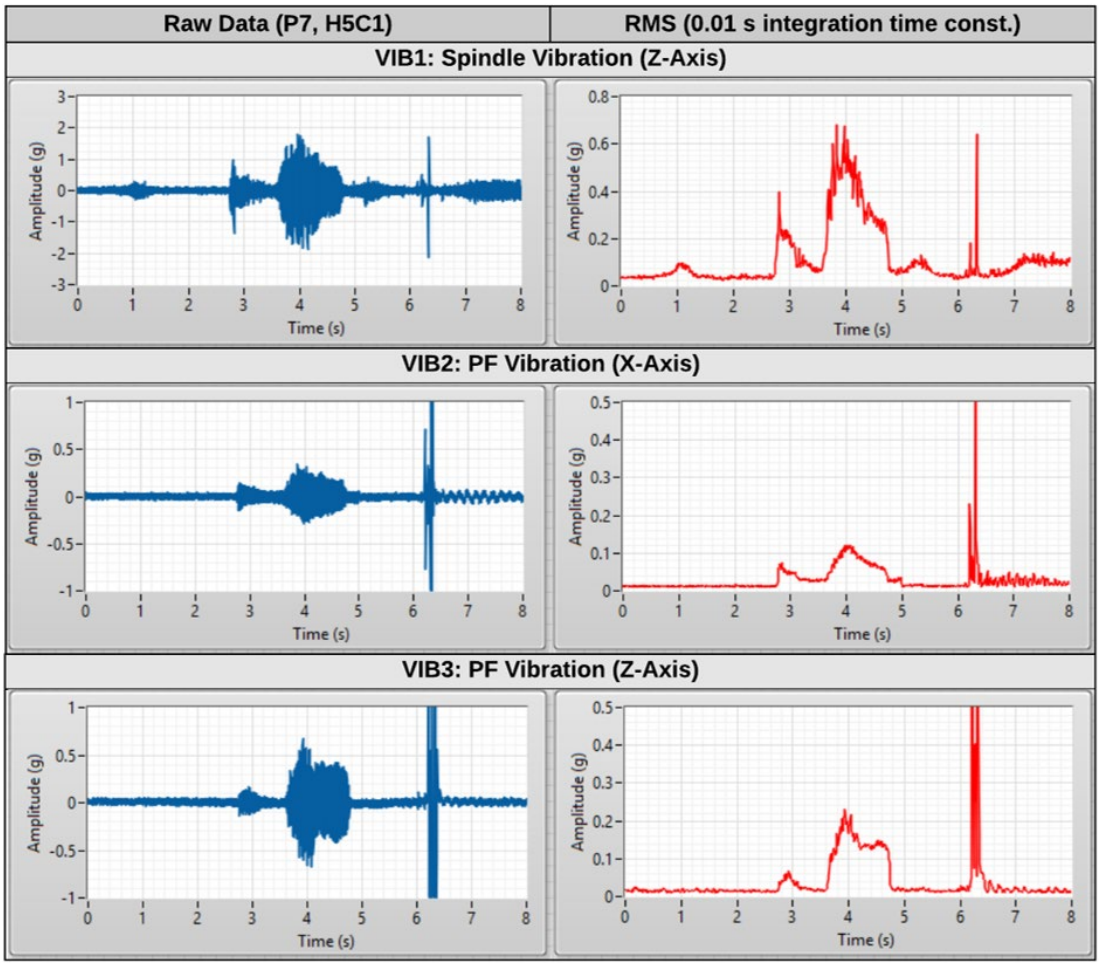

Figure 5.4 Vibration signals (blue plots) and the converted RMS signals (red plots) acquired during the first cut.

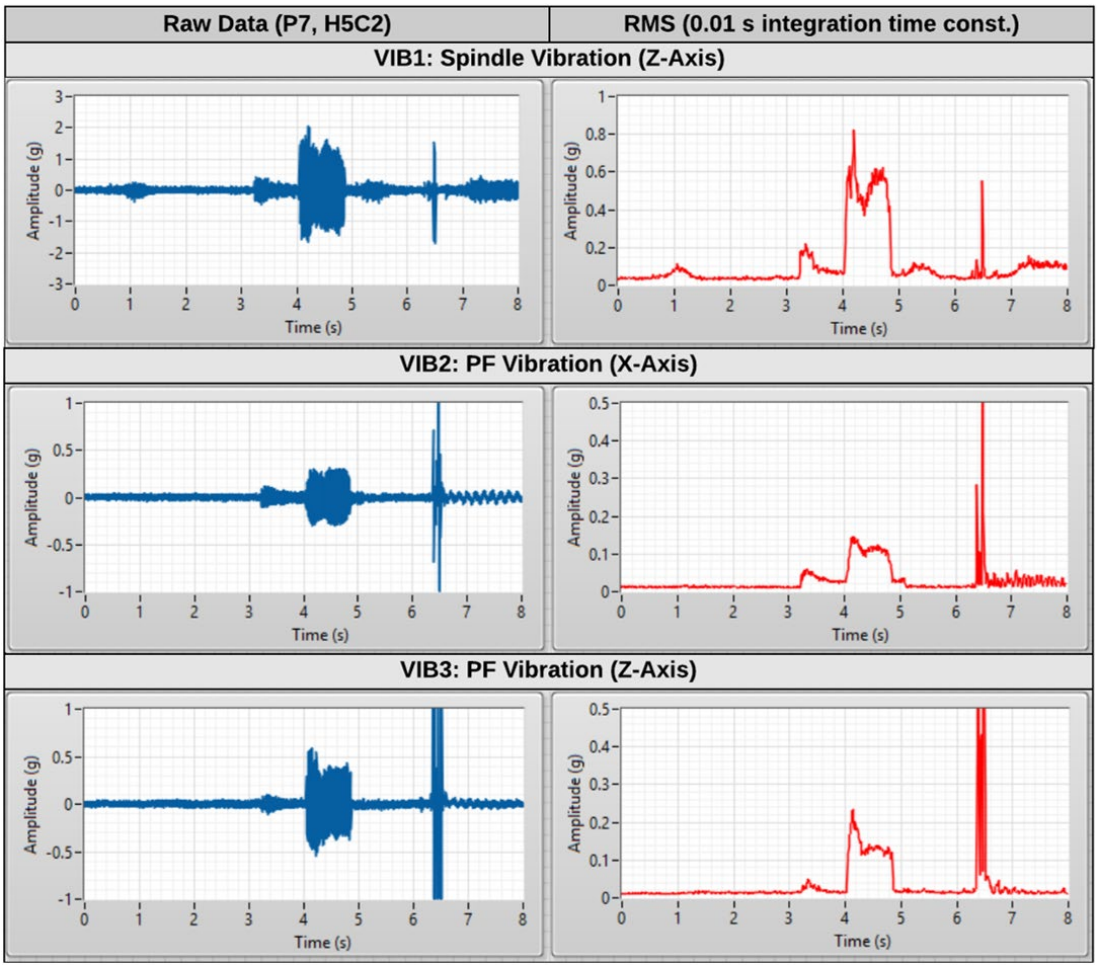

Figure 5.5 Vibration signals (blue plots) and the converted RMS signals (red plots) acquired during the second cut.

The sample rate of the three vibration signals, shown in Figure 5.4 (first cut) and Figure 5.5 (second cut), was set at $10240\ Hz$. All signals presented a similar pattern and showed an increase in the magnitude when the cutting commenced approximately at time $3.6\ s$ for the first cut and $4\ s$ for the second. This fact was also emphasised by the converted RMS signals, making them suitable to use for the segmentation of the raw data. In particular, the spindle vibration data (VIB1) showed the highest change in the signal magnitude between the two regions: in-cut and out-of-cut. The signals of the second cut (Figure 5.5) appeared to be more concentrated in the region of interest, simplifying the segmentation of the original data. That was because the second cut started with a hole previously machined at a semi-finish level and the vibration, which occurred after the tool reached that semi-finish level, presented rapid changes in the magnitude of the recorded signal due to the greater contacting area.

Figure 5.6 and Figure 5.7 show the spindle power data recorded during the first cut and the second cut, respectively. The signals refer to the same hole considered for the vibration data

in order to highlight the synchronisation of the signals across the sensors. Given the low frequency content of the power signals, the acquisition sample rate was set at $1600\ Hz$.

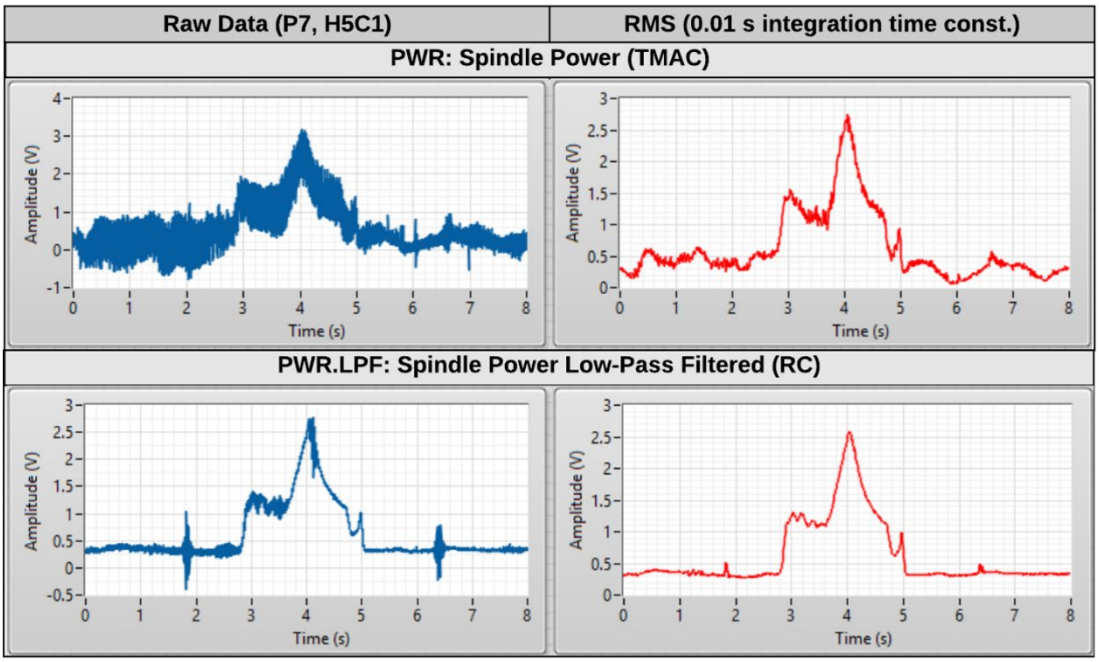

Figure 5.6 Power signals (blue plots) and the converted RMS signals (red plots) acquired during the first cut.

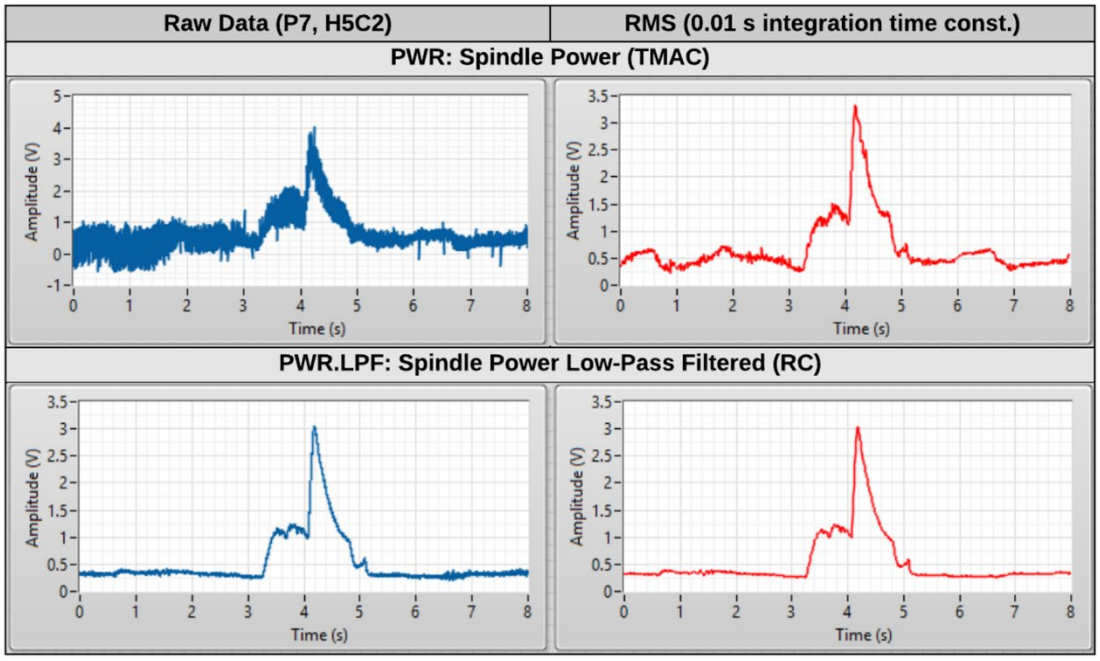

Figure 5.7 Power signals (blue plots) and the converted RMS signals (red plots) acquired during the second cut.

The Acoustic Emissions (AE) data recorded during the first and the second cut are presented in Figure 5.8 and Figure 5.9, respectively. The hardware (NI-9223 module) allowed acquisition

rates up to 1 million samples per second, however, in order to ensure the synchronisation and the online visualisation and processing of the data, a lower sample rate of $409600\ Hz$ was used for the acquisition of the AE signals.

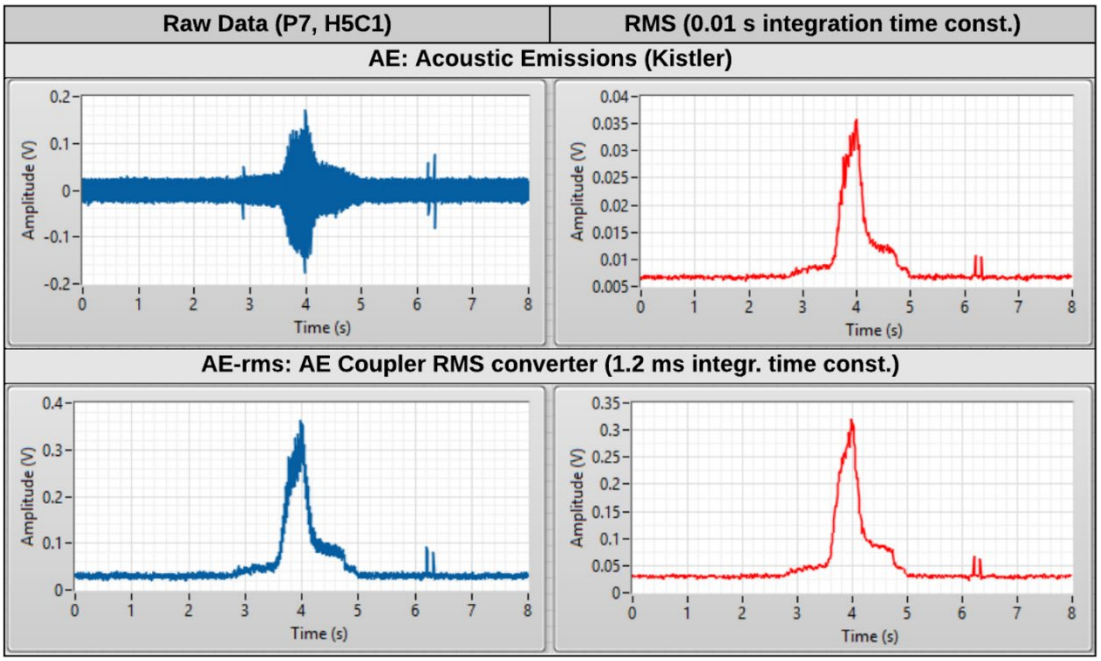

Figure 5.8 Acoustic Emissions signals (blue plots) and the converted RMS signals (red plots) acquired during the first cut.

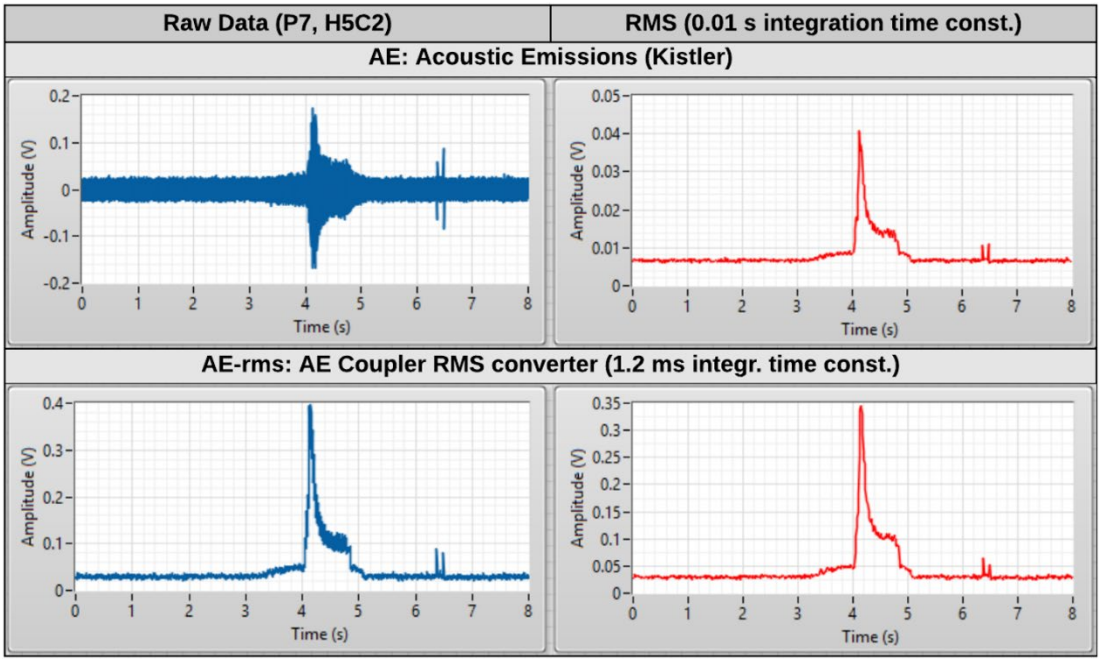

Figure 5.9 Acoustic Emissions signals (blue plots) and the converted RMS signals (red plots) acquired during the second cut.

Similar to the vibration data, the power and AE graphs showed an increase of the signal magnitude when entering the cutting process, reaching the peak value shortly after the cut starts, at approximately $4\ s$ for the first cut and $4.2\ s$ for the second cut. However, unlike the vibration data, the signals quickly dropped in magnitude, especially the AE data, indicating that the cutting tool was not removing material from the panel anymore and the target depth level was reached. This fact, emphasised by the converted RMS signals, was clearly observed in the graphs related to the second cut where the actual time in cut was considerably shorter.

The raw power data (ref. Figure 5.6 and Figure 5.7) showed significant signal noise, which was mostly removed by the RMS conversion. A small increase of the signal magnitude was observed just before the cutting commenced, which made it difficult to identify the exact cutting entry point. The Low-Pass Filtered (LPF) power data provided by the RC circuit was introduced as an additional physical channel (acquired by the same NI module) with the intention to limit the amount of signal noise, originally observed. Although the signal succeeded in removing the noise, it showed some periodic disturbances which were probably caused by electrical interferences due to not proper isolation of the wires. These periodic disturbances were present in the signals recorded from the first cut only (see Figure 5.6) and they were largely removed when converting to RMS. For this reason the converted RMS signal of the LPF power data was used for further analysis instead of the original signal.

In terms of AE data, both raw signals (blue plots of Figure 5.8 and Figure 5.9) clearly showed the region when the tool was cutting material, however, the data provided by the coupler's RMS converter (AE-rms) was less affected by signal noise, making it a suitable choice for segmentation purposes. Similar to power data, the further RMS conversion at 100 samples per second appeared to remove most of the signal noise and therefore it was used for further analysis instead of the original raw data coming from the coupler's RMS converter.

Finally, for completeness of information, the probe data is shown in Figure 5.10. As previously mentioned, this signal was not used for the analysis presented here, however, it helps to understand the overall process cycle. The probe sensor measured the linear displacement of the spindle with respect to the pressure foot. For comparison with the other signals, the two graphs shown in the figure refer to the same hole, with the signals relative to the first cut on the left hand side and those of the second cut on the right hand side. They both appeared to be very similar in shape with just a few differences in the time when they reached the peak value, which was approximately $4\ s$ for the first cut and $4.2\ s$ for the second cut, confirming the observations made from the other sensors.

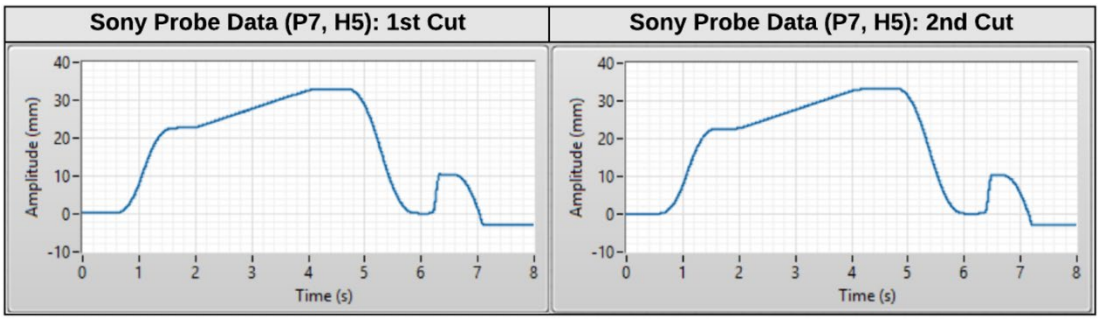

Figure 5.10 Probe signals acquired during the first cut (left) and the second cut (right).

Examining the probe graphs, the first rise in the signal magnitude showed that the spindle was rapidly advancing in the feed direction (Z-axis) towards the panel. It then stopped for a short time at about 10 mm from the target depth (i.e. the maximum value) before advancing again (at time approx. $2\ s$) with constant speed to perform the cutting (the feed rate was set by the program). After reaching the target depth level, the spindle remained in position for the specified dwell time and then quickly retrieved back to the initial position (at approx. 6 s). The PF movement returning to its home position during unclamping was also captured by the signal (at approx. $6.2\ s$ and $6.4\ s$ for the first and second cut, respectively).

5.3.2 Automatic Segmentation Method

As previously noted in Section 5.1, particular attention was given to the selection of the panels used during Phase 2 of the experiment. This resulted in having panels with better surface quality and paint adhesion than those in Phase 1, which improved the inspection routine. However, these small differences in terms of panels' material quality led to sensor signals with slightly different signal-to-noise ratios from what was observed during the trials of Phase 1. It was therefore necessary to identify a new "best signal" with the highest signal-to-noise ratio to use for the segmentation task.

For instance, preliminary examination of the signals recorded during the first phase of the experiment suggested that the AE data was the most suitable to use for the signal segmentation task. In particular, the AE-rms signal showed greater signal-to-noise ratio in the first four panels (P1-P4) machined during this phase. Therefore, it was decided to use the AE-rms signal (RMS computed at 100 samples per second) for the segmentation of all the other signals. An example of the method is shown in Figure 5.11 for the first cut data (left plot) and the second cut data (right plot). The segmentation procedure automatically identifies the entry/exit points (time stamps) of the region of interest and then uses this information to segment all the other signals for further processing.

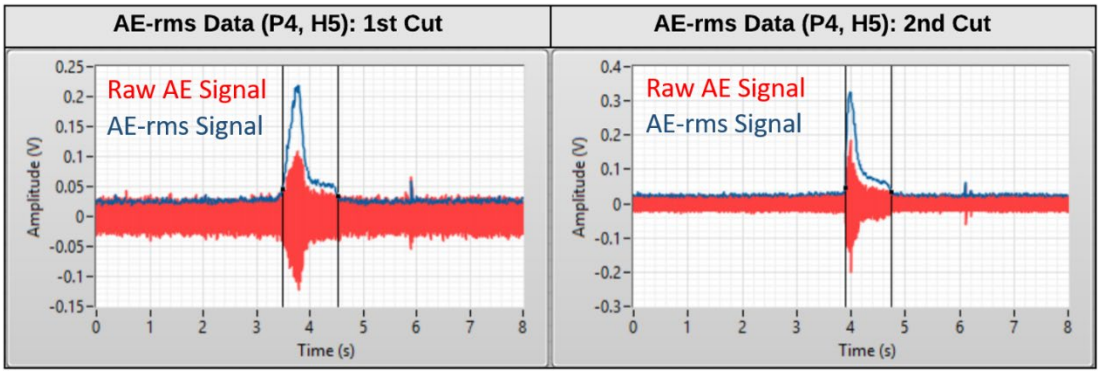

Figure 5.11 Signal segmentation method used for Phase 1 of the experiment (P1-P4).

In the case of the last three panels (P5-P7) machined during the second phase of the experiment, the AE-rms signal showed much higher noise levels, which made it difficult to apply the segmentation method used in the first phase. On the other hand, the spindle vibration data (VIB1), which had a lot of signal noise during the first phase, presented much clearer signal in all the panels of the second phase. It was decided, therefore, to use the converted RMS signal (computed at 100 samples per second) of the VIB1 data for the segmentation of all the other signals. Figure 5.12 shows an example of the method applied on the first cut data (left plot) and the second cut data (right plot).

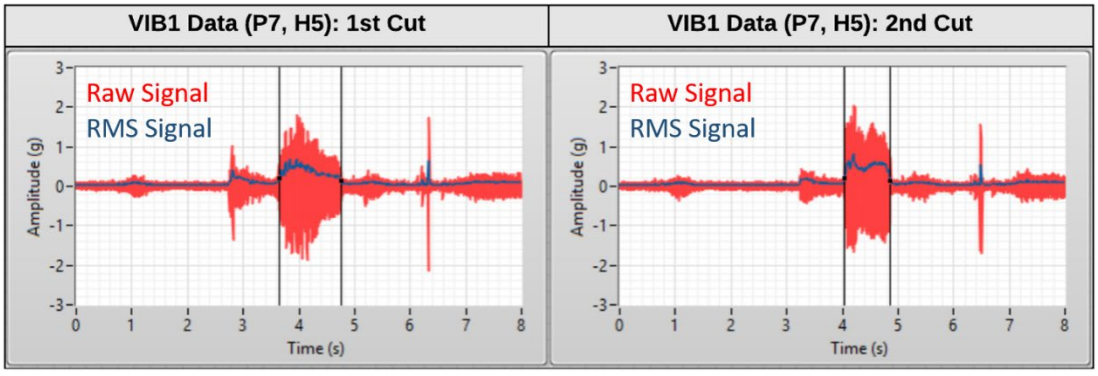

Figure 5.12 Signal segmentation method used for Phase 2 of the experiment (P5-P7).

The synchronisation of the signals due to the cDAQ device, ensured that the part of the signal identified by the segmentation procedure as the cutting region referred to the same event across all the sensors. Once the time information of the cutting entry/exit points was calculated, only the segmented parts of the raw signals were considered for further analysis.

5.4 Feature Extraction

This section presents the further processing steps and extraction methods applied to the above segmented signals in order to obtain a vector of signal features that could potentially describe the cutting process. In particular, the proposed monitoring system is looking for those features that carry information about the process variable of interest, i.e. the countersink DoC value. Following from the feature extraction methods reviewed in the literature, presented in Section 2.2.3, three signal domains were investigated in this study: time domain, frequency domain and the joint time-frequency domain. The signals extracted from the original sensor data and the analysis performed in each of the above domains are described in the following sub-sections.

5.4.1 Time Domain Analysis

In terms of the vibration sensors (three accelerometers), in addition to the raw data (VIB) and the converted RMS data (VIB.RMS), three more signals were extracted:

- The envelope signal (VIB.ENV) obtained by an envelope analysis of the raw data.
- The residual signal (VIB.RES) obtained by filtering out the Tool Passing Frequency (TPF) and all the harmonics in the signal frequency range.
- The converted RMS (calculated at 100 samples per second) of the residual signal (VIB.RES.RMS).

The TPF was computed from the spindle speed by the formula:

$$TPF = \frac{Spindle\ RPM}{60} \times n_{teeth} \tag{5.1}$$

In this experiment, the spindle speed was set at $4978\ RPM$, which led to a $TPF \cong 165\ Hz$ (for a 2-flute cutting tool) and a fundamental frequency component at $F_{RPM} \cong 83\ Hz$ (approximately), related to the spindle RPMs (Rotations per Minutes).

An example of all the time domain signals extracted from the spindle vibration data (VIB1) collected during the first cut is illustrated in Figure 5.13. Similar patterns were found for the second cut signals and the other accelerometers as well. The two black vertical lines indicate the cutting region as identified by the segmentation method. A total number of 15 time domain signals were extracted (five for each accelerometer).

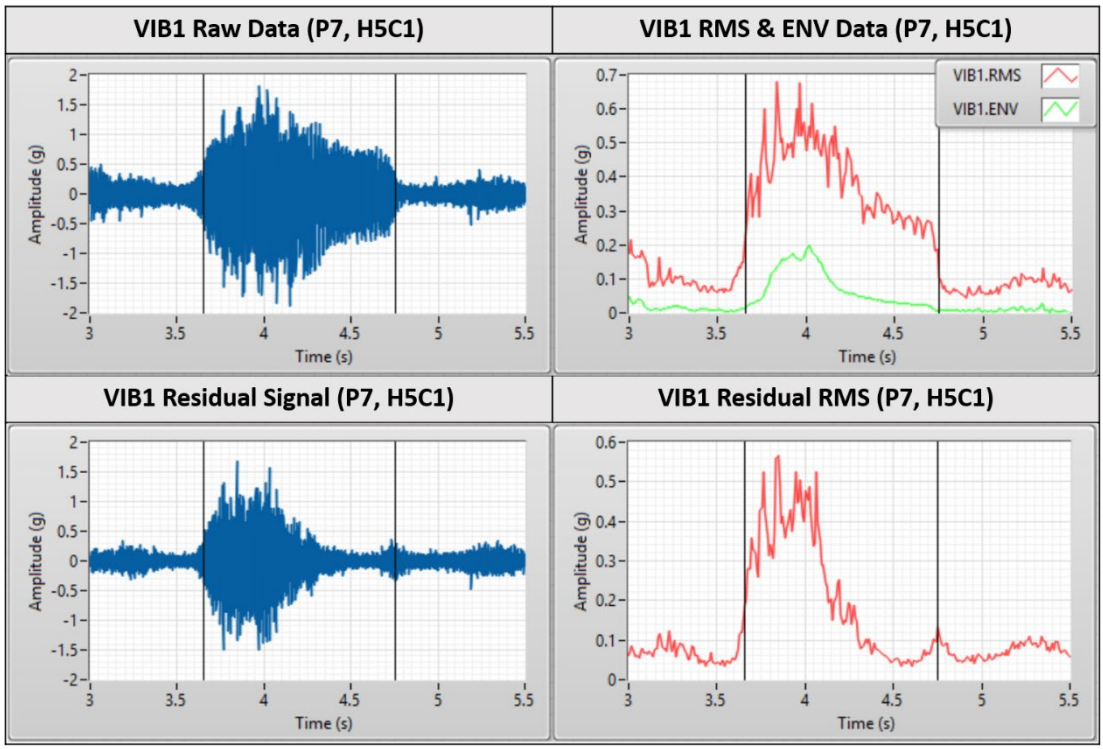

Figure 5.13 Spindle vibration (VIB1) time domain signals considered for further analysis.

The envelope analysis (ref. to Section 2.2.2) was applied to detect the amplitude modulated signal around the TPF due to the cutting forces. It was performed on the raw vibration signal with central frequency at TPF (first order) and 40 Hz of bandwidth span. In order to obtain the residual signal, the fundamental frequency and all its harmonics (60 in total up to 5 kHz, including the TPF), were filtered out from the original data. This led to a new signal (VIB.RES) with similar amplitude, but shorter duration in time. Considering that the tool actually cuts material during the first part of the signal until the peak amplitude is reached (before quickly dropping to the normal level), the residual signal appeared to describe better the cutting process. Since its magnitude dropped faster than the original one, it could localise the cutting region with greater precision. The signal duration, computed from the converted RMS data (VIB.RES.RMS) as the time that the amplitude of the signal remained over a certain threshold, was included in the feature vector, due to its potential correlation with the DoC.

In terms of the power transducer, four time domain signals were extracted:

- The raw power data (PWR).
- Its RMS signal (PWR.RMS).
- The Low-Pass Filtered signal (PWR.LPF) obtained from the RC circuit (and the further RMS conversion step).

- The Time-Frequency Filtered signal (PWR.TFF) generated by exploring the time-frequency domain of the original data (ref. to Section 2.2.3.3).

In particular, the TFF signal was computed by applying a Gabor transform with all coefficients below a certain threshold set to zero, leaving just those that carried the signal energy. The time domain signal was then reconstructed from the remaining coefficients. The parameters used for the Gabor transform were the following: threshold value 0.87, time steps 8, frequency bins 512, window length 512 (Gaussian).

Figure 5.14 presents an example of all the above time domain signals extracted from the spindle power data recorded during the first cut (of the same hole showed in the vibration graphs). Again, the black vertical lines indicate the segmented region considered for the feature extraction step.

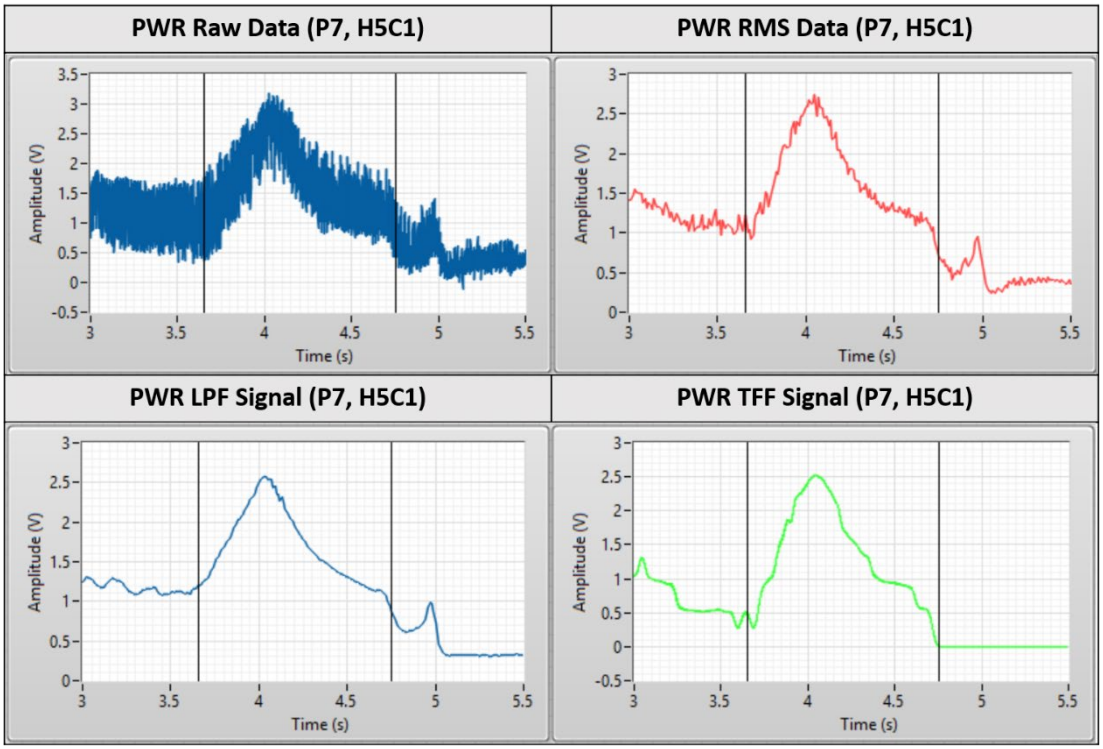

Figure 5.14 Spindle power time domain signals considered for further analysis.

The time domain signals extracted from the AE sensor data collected during the first cut (of the same hole as before) are shown in Figure 5.15. In addition to the two original signals provided by the AE coupler, the raw AE and the converted RMS signal (AE.RMS), two more were considered for further analysis:

- The Band-Pass Filtered signal (AE.BPF).

- Its RMS signal (AE.BPF.RMS).

As seen in the figure, the BPF signal (bottom left plot) presented considerably less noise than the original one, making it easier to separate between the two regions (in-cut and out-of-cut). This fact was also highlighted by the converted RMS data (bottom right plot) showing a clear gap when exiting the region of interest. A 3rd order Butterworth filter with frequency band of $[15k - 25kHz]$ was used to obtain the BPF signal. This frequency band was chosen after the examination of the original data in the frequency and the joint time-frequency domain, as will be explained in the following sub-sections.

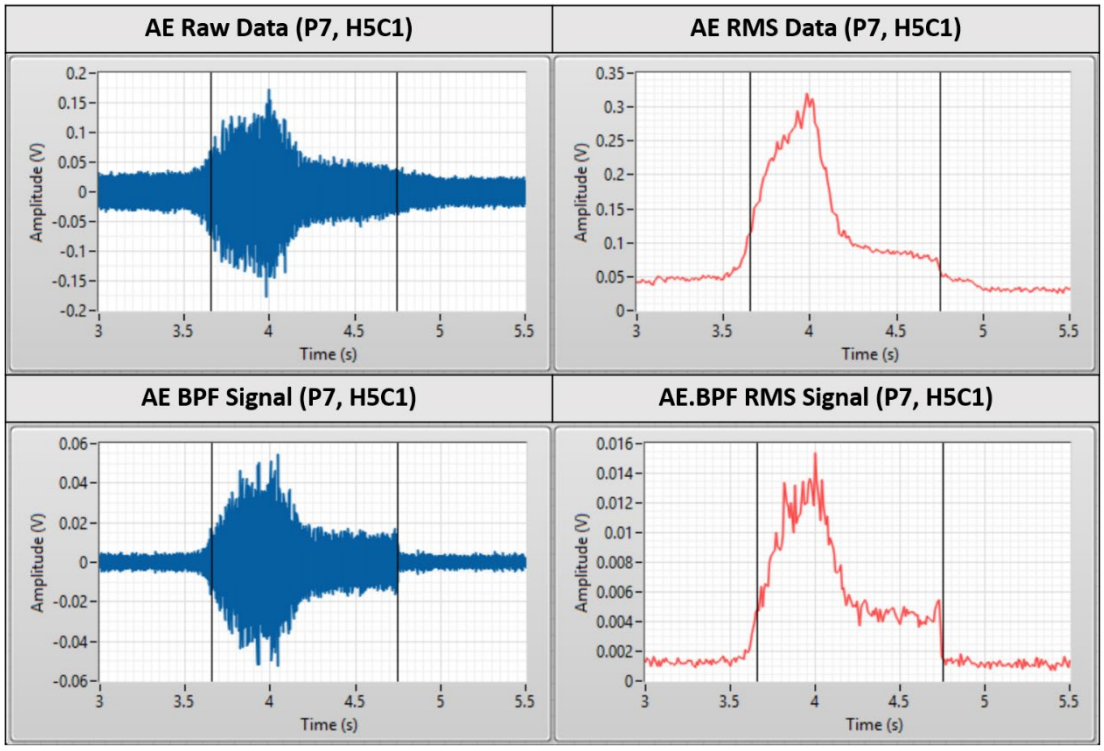

Figure 5.15 AE time domain signals considered for further analysis.

To summarise, the time domain analysis produced a total of 23 (segmented) signals to consider for the feature extraction step: 15 from the vibration sensors (five for each accelerometer), four from the power transducer and four from the AE sensor.

The time domain features extracted from the above signals included the statistic descriptors presented in Section 2.2.3.1 (ref. to Table 2.2), such as the signal mean, RMS, variance, skewness, kurtosis, peak and crest factor. In addition, some other features potentially related to the DoC value such as the cutting cycle duration (duration of the segmented part of the signals), the Time-to-Peak (time spent to reach the peak amplitude) and the duration of the VIB.RES.RMS signal were also included.

5.4.2 Frequency Domain Analysis

In order to assess the frequency content of the signals, the Power Spectral Density (PSD) was calculated for each of the raw sensor data in the cutting region, identified by the segmentation method. An example of the power spectrum of the spindle vibration data (VIB1) collected during the first cut is shown in Figure 5.16. A linear amplitude, rather than a log one (dB scale), was chosen for better visualisation of the peaks corresponding to the TPF and harmonics. The top graph presents the PSD of the original data, whereas the bottom graph shows the power spectrum of the residual signal, i.e. what remains after removing the TFP and all the harmonics up to $5kHz$ (the entire frequency range of the raw data). The residual signal presented the peak amplitude at approximately $2650\ Hz$, with most of its frequency content located inside the band $[2k - 3.5kHz]$, as illustrated in the figure. Similar graphs were found from the data of the other accelerometers (VIB2 and VIB3), as well as those collected from the second cut.

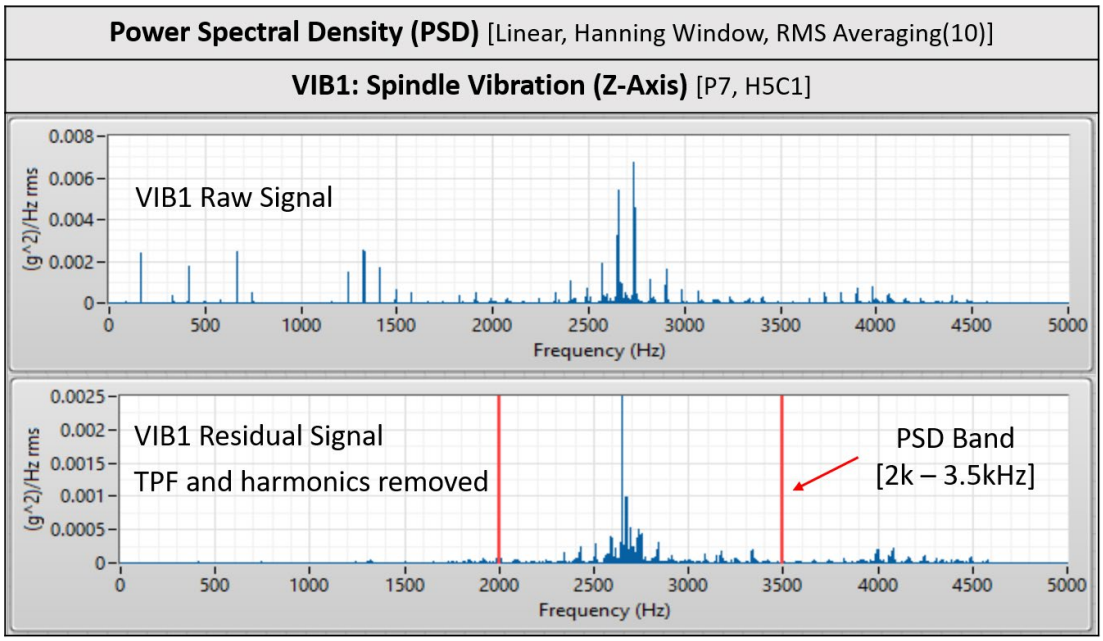

Figure 5.16 Power Spectrum for spindle vibration data (VIB1): raw signal (top) and residual signal with TPF and harmonics removed (bottom).

An example of the power spectrum of the AE data collected from the first cut is shown in Figure 5.17. The top graph presents the PSD of the raw signal with indication of the two bands where most of the frequency content was concentrated. The spectral peak amplitude was found inside the first band (Band1 $[15k– 25kHz]$) at approximately $20kHz$. Note that Band1 represents also the frequency range of the BPF signal considered previously for the time domain analysis. The second band (Band2 $[80k - 160kHz]$) contained the rest of the power

spectrum, with a peak at approximately $135kHz$. It will be clear later in the time-frequency analysis that this peak corresponds to a spectral noise, which was present during the entire 8 seconds of recording of the signal, not only in the cutting region. For this reason, it was decided to filter it out and the remaining power spectrum, denoted as Band3, is shown in the bottom graph of Figure 5.17. Band3 has the same frequency range of Band2, but with the noise spectral peak removed.

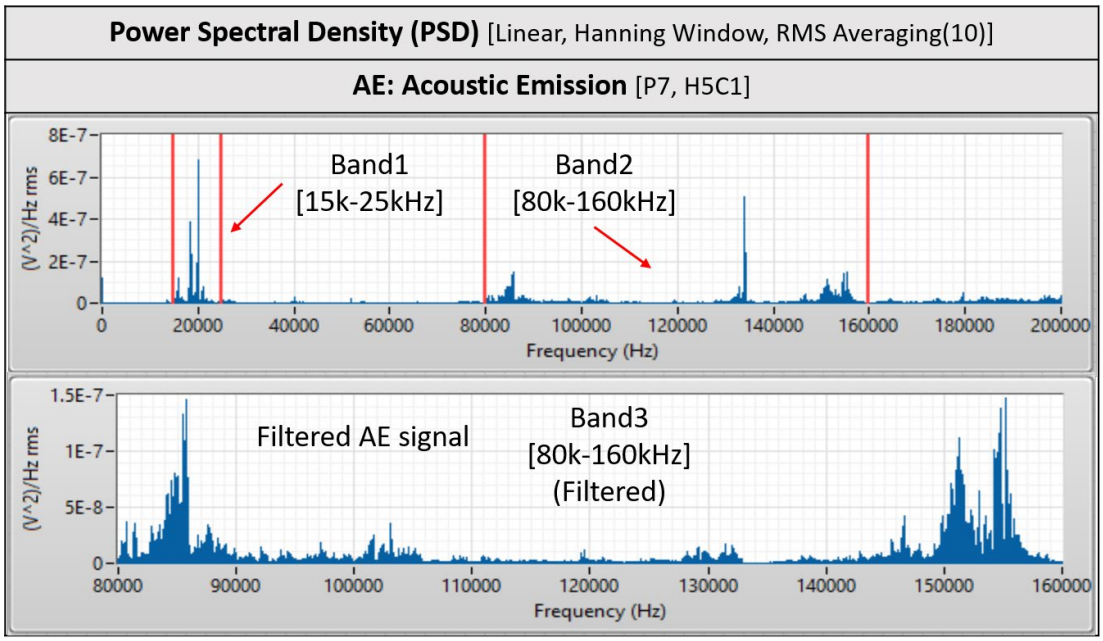

Figure 5.17 Power Spectrum for AE data: raw signal (top) and filtered signal with noise peak at 135 kHz removed (bottom).

In terms of the spindle power sensor, the PSD of the raw power data didn't show any particular spectral peaks (rather than the one in zero frequency) or any presence of the TPF harmonics. Given the low frequency response of the power transducer, a spectral band of $[10 - 100Hz]$ was considered for further treatment.

As discussed in Section 2.2.3.2, the frequency domain analysis extract features from bands of the power spectrum. In the analysis here, the frequency domain features were extracted from two main spectra: the complete power spectrum of the raw data (this includes also the VIB residual signal) and bands of the power spectrum obtained from the original PSDs. These bands were chosen based on observations of the signal's original PSD and they were specific to each sensor: Vibration, [2k-3.5kHz]; AE, Band1 [15k-25kHz], Band2 [80k-160kHz] and Band3 [80k-160kHz] (Filtered); Spindle Power, [10-100Hz].

To summarise, a total of 18 frequency domain signals (bands of PSD) were considered for the feature extraction step: 12 from the vibration sensors (four for each accelerometer, two original PSDs and two bands), two from the power transducer (original PSD and one band) and four from the AE sensor (original PSD and three bands).

The frequency domain features included statistic descriptors relative to bands of power spectrum, such as the spectral mean, variance, skewness, kurtosis and peak amplitude (ref. to Table 2.3). The sum of total band power, the relative spectral peak per band and the total harmonic band power (only for the vibration data) were also calculated. Given the purpose of this study to extract features that could potentially be related to the DoC, some other frequency features were extracted, but not considered for further treatment. For instance, the peak frequency of the band power and the amplitudes of all the TPF harmonics are features that are usually used for tool wear monitoring and detection of faults in rotating machining components (gears, bearings), since they can detect changes in the spindle RPMs, however, there was no evidence to suggest a potential correlation of them with the DoC.

5.4.3 Time-Frequency Analysis

The joint time-frequency analysis was considered in order to understand how the frequency content of the signals changed over time, especially in the cutting region. Figure 5.18 presents the spectrogram of the vibration data (VIB1) collected during the first cut (only the segmented part). The left graph shows the raw signal, whereas the residual signal (the TPF and harmonics removed) is illustrated in the right graph. Note that most of the frequency content of the latter was concentrated in the first 0.5 s from the start of the cutting process.

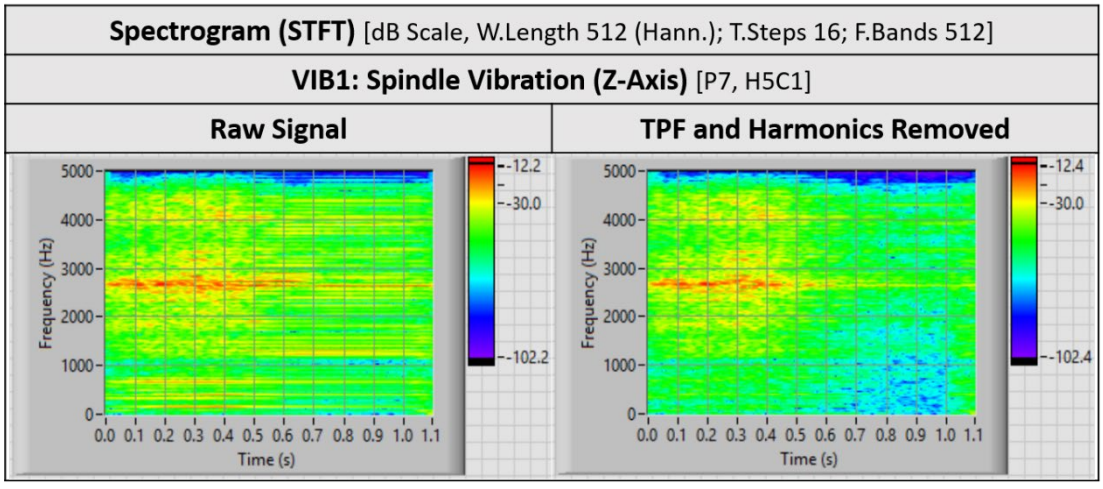

Figure 5.18 Spectrogram for vibration data (VIB1): raw (left) and filtered (right) signal.

The spectrogram of the AE data collected during the first cut is shown in Figure 5.19. The left graph presents the original signal (entire recording of 8 s), whereas the right graph shows only the segmented part. It is clear now the reason of the filtered band in the AE PSD (Band3), given the constant presence of the spectrum noise at approximately $135kHz$.

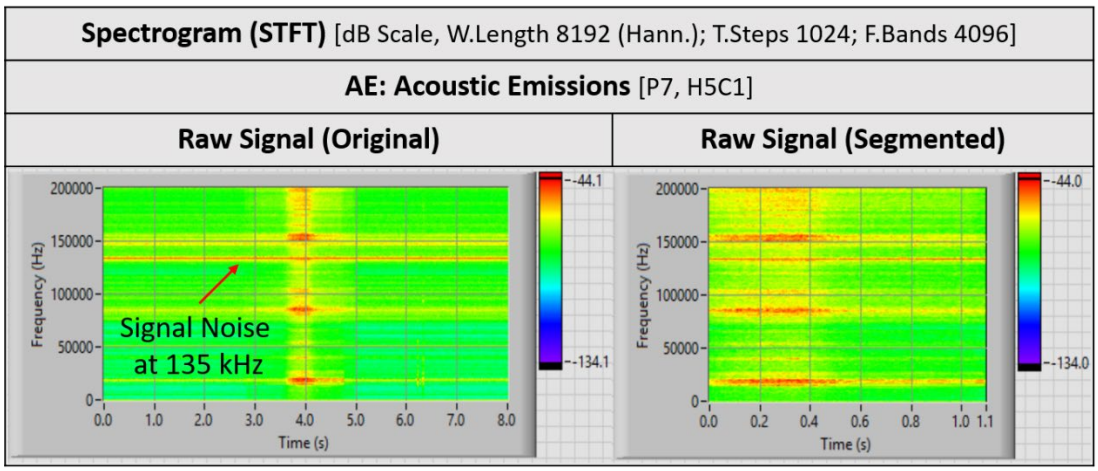

Figure 5.19 Spectrogram for raw AE data: original (left) and segmented (right) signal.

In this study, the spectrograms were mainly introduced for better understanding of the frequency content of the signals and time-frequency domain features were extracted, but not considered for further treatment. They typically include the Mean Instantaneous Frequency (MIF), Mean Instantaneous Bandwidth (MIB), Time Marginal Integral (TMI) and Frequency Marginal Integral (FMI).

Finally, a summary of all the signals and all the features considered for the experiment is presented in Table 5.2 and Table 5.3, respectively.

Table 5.2 Summary of the signals considered for further processing.

Extracted Signals		**Vibration x3**	**Spindle Power**	**Acoustic Emissions**
Time Domain Analysis	Periodic	VIB (Raw) VIB.RES		AE (Raw) AE.BPF
	Non-Periodic	VIB.ENV	PWR (Raw) PWR.LPF PWR.TFF	
	RMS Signals	VIB.RMS VIB.RES.RMS	PWR.RMS	AE.RMS AE.BPF.RMS
Frequency Domain Analysis	Original	VIB.PSD VIB.RES.PSD	PWR.PSD	AE.PSD
	Bands of Power Spectrum	VIB.BP VIB.RES.BP	PWR.BP	AE.BP1 AE.BP2 AE.BP3

Table 5.3 Summary of the signal features extracted for the experiment.

		Time Domain			Frequency Domain	
Extracted Features		Periodic	Non-Periodic	RMS Signals	Original Spectrum	PSD Bands
Duration*	Du	-	-	x	-	-
Mean	M	-	x	x	x	
(of Band Power)	MBP					x
Root Mean Square	RMS	x	x	-	-	-
Variance	V	x	x	x	x	
(of Band Power)	VBP					x
Skewness	Sk	x	x	x	x	
(of Band Power)	SkBP					x
Kurtosis	Ku	x	x	x	x	
(of Band Power)	KuBP					x
Peak (absolute)	P	-	x	x	x	
(of Band Power)	PBP					x
Peak to Peak (Range)	Rng	x	-	-	-	-
Time to Peak	TP	-	x	x	-	-
Crest Factor	CF	x	x	x	-	-
Sum of Total Band Power	STBP	-	-	-	x	x
Relative Spectral Peak per Band	RSPB	-	-	-	x	x
Total Harmonic Band Power**	THBP	-	-	-	x	-
Total Number of Features		**163**			**129**	

(*) applied only to the RMS data of the residual vibration signals (VIB.RES.RMS)
(**) applied only to vibration signals (TPF and Harmonics up to 5kHz)

Note that the time domain signals were categorised in periodic (oscillating signals such as those of VIB and AE data), non-periodic (PWR data) and the converted RMS signals. The total number of features extracted from all the sensor signals was 292.

The data acquisition and the further processing, including the segmentation and feature extraction step was performed in real time, during the machining process. After each cutting cycle, a vector of 292 feature values was produced in output from the collected signals (one vector for each cut, for each hole of the panel). These feature vectors, together with the measured DoC values (obtained by the inspection camera) provided the two datasets (one for each cutting cycle) for the regression tests presented in the next chapter (see Section 6.1).

6 SUPERVISED LEARNING METHOD

This chapter presents the supervised learning method applied to the data collected from the experiment conducted by this work. A regression problem was considered in order to build and test process models that can accurately predict the countersink depth from the signal features extracted by the methods discussed in Chapter 5. A probabilistic learning approach based on Gaussian Process Regression (GPR) was used for creation of the process models. The aim and objectives of the regression tests are outlined in Section 6.1. The two datasets extracted from the experimental trials (one for each cut) are discussed in Section 6.2. The feature selection method applied to the dataset and the learning algorithm for the process models are presented in Section 6.3. Finally, the results obtained from the tests are given in Section 6.4. The chapter concludes with a summary of the main outcomes.

6.1 Regression Tests

In the regression tests performed in this work, a probabilistic framework based on Gaussian Process Regression (GPR) was applied to the data obtained from the experiment described in Section 5.1. In particular, a process model was trained using the two datasets created from examples of the extracted signal features and the corresponding countersink depth, as explained at the end of Chapter 5.

During the experiment, each hole was processed by two cutting cycles (semi-finish and finish level) and a camera inspection was performed after each cut to collect the depth measurement. The inclusion of this camera inspection step at the end of the cutting cycle, allowed to build the above datasets online, while the process was running and without any interruptions.

The regression tests considered in this work aimed to assess the accuracy and reliability of the proposed process model. The main objectives included:

- Investigation of a probabilistic learning approach for the prediction of countersink depth values.
- Identification of a suitable method for the selection of relevant signal features.
- Creation and validation of a regression model that provides accurate depth predictions with uncertainty estimation in the prediction value.

The two datasets obtained from the experimental trials (one for each cut), the feature selection method and the learning algorithm, as well as the obtained results are discussed in the following sections.

6.2 Datasets

It was noted in Section 5.1 that during the trials of Phase 1, one of the vibration sensors' cable was damaged. Therefore, the sensor VIB3 was disconnected and all the features obtained from VIB3 were ignored in the signal processing step. This reduced the total number of the extracted features to 228 for the Phase 1 of the experiment (P1 to P4). The sensor cable was replaced in time for Phase 2, in which three more panels (P5 to P7) were machined. Here, all the sensors were perfectly working and installed as described in Section 5.1 of the experimental setup. A total of 292 signal feature values were extracted for each cutting cycle of every hole machined during Phase 2.

Two datasets (one for each cut) were obtained during the experiment. Dataset 1 contained samples of the feature vector extracted from the first cut together with the measured DoC value of the semi-finish level (from the inspection camera). Dataset 2 contained samples of the second cut's features together with the corresponding final DoC value, as measured by the inspection camera at the end of cutting process. These datasets were then used to build two regression models: Model 1 and Model 2, respectively.

It is important to note that while the first cut was performed under the same initial conditions with a new hole every time (all the pre-drilled holes of the panel had the same initial size), the starting conditions of the second cut were dependent on the previous cutting cycle and the already machined semi-finish depth level. This was the reason why the pattern of the signals was different between the two cuts. In a new hole the contacting area between the cutting tool tip and the panel surface increased as the tool entered the hole and advanced towards the panel, whereas in a previously countersunk hole (at semi-finish level) the greater contacting area from the start of cutting process caused rapid changes in the magnitude of the recorded signals and different patterns were obtained (ref. to Section 5.3.1 for examination of the signals). Since this contacting area depended on the semi-finish depth level of the previous cut, in theory, several models should be used (one for each different starting DoC value). However, this would be impractical due to the uncertainties in the semi-finish depth value. Moreover, after examination of the signals collected from the second cut, it was observed that they all presented similar patterns, independently of the starting depth

conditions. Therefore, an averaged model was considered instead, which operated in a range of starting conditions with expected (mean) value at the nominal semi-finish level including also the depth variations introduced by the Gaussian perturbation added in the process depth target. Using only one model for the representation of all the different initial conditions of the second cut allowed the system to be more robust and flexible since it could make use of all the data available in Dataset 2.

6.2.1 Dataset 1: Semi-Finish Depth Data

In the experiment, the Gaussian perturbation was manually added to the depth target specified into the spindle drive controller before cutting of each hole. This allowed to obtain examples of first cuts with different DoC values, distributed (normally) around the nominal semi-finish target (d_1), set at 2 mm in the tests. Figure 6.1 shows the semi-finish depth variation values (i.e. the depth errors calculated as the difference between the camera measurement and the nominal target d_1) obtained at the end of the first cut for all the holes considered in the experiment. The vertical (dashed) red line divides the two phases: Phase 1 (panels P1-P4) and Phase 2 (panels P5-P7). Negative values represent holes with under-depth data with respect to the semi-finish level (shallow holes) and positive values refer to over-depth data (deep holes). In the following, the term semi-finish depth or DoC value is used to refer to the above depth variation (error) value, which is also the process output variable.

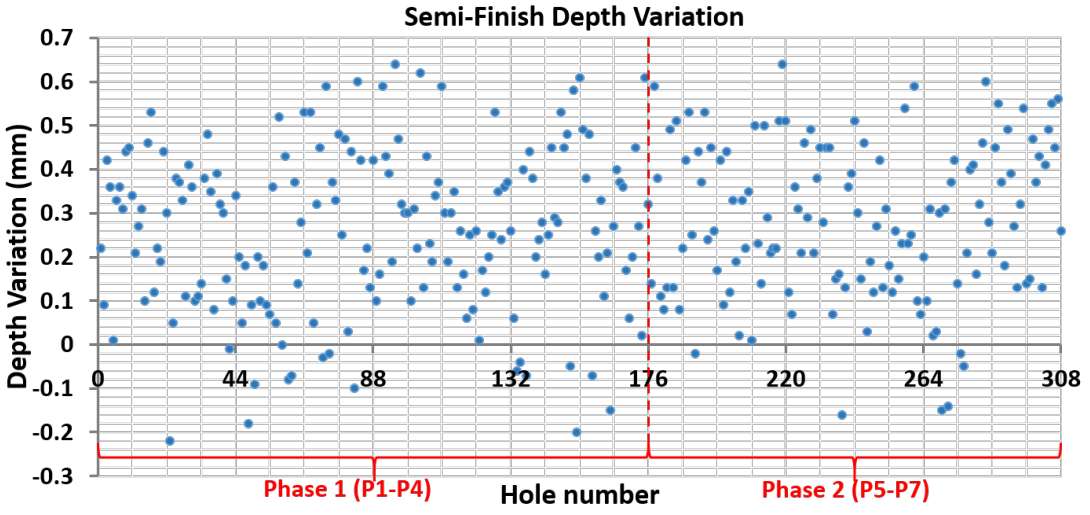

Figure 6.1 Depth variation data obtained after the first cut (semi-finish level).

The Gaussian perturbation added to the target had mean $\mu = 0$ and standard deviation $\sigma = 0.25$, therefore, most of its values belonged to the interval [-0.5 0.5]. Adding this variable to the semi-finish depth target, one would expect to obtain depth variation values in the same

range of data. However, this was not the case, as illustrated in Figure 6.1. The measured values were distributed in the range [-0.3 0.7] mm and presented a positive bias of approximately 0.27 mm, with the majority of the holes having over-depth data. The depth variability required to accurately train Model 1 was still maintained in accordance with the perturbation variable, taking values within the interval [-0.3 0.7], centred at $d_1 = 2mm$. The effect of the perturbation variable into the observed DoC value is shown in the two charts of Figure 6.2: the first one (in the left) shows the correlation between the perturbation and the DoC values obtained after the first cuts; the second chart (right) illustrates the depth variability introduced by the perturbation variable in the form of a histogram (blue boxes) with a Gaussian distribution fit (red line).

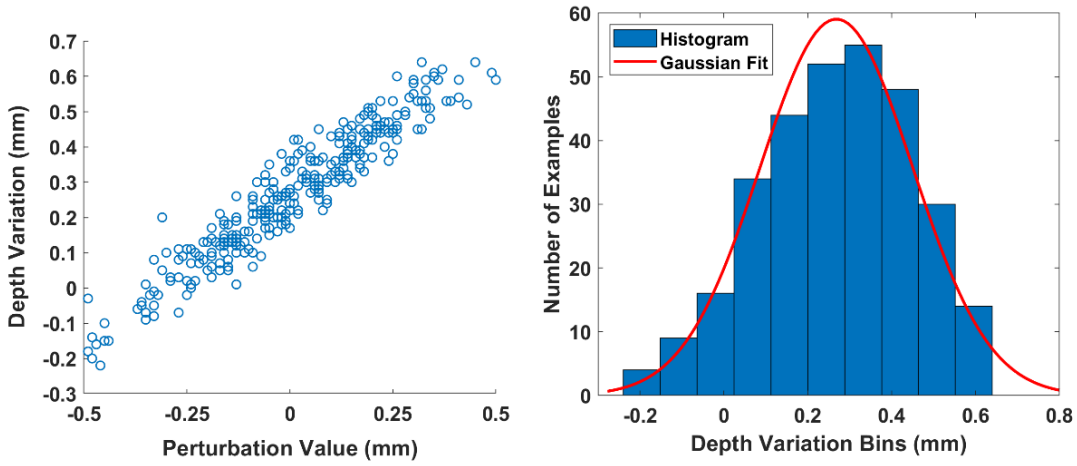

Figure 6.2 Correlation between the Perturbation and Semi-Finish Depth values (left); Semi-Finish Depth variability chart expressed as a histogram with distribution fit (right).

The charts in Figure 6.2 highlight the systematic error (bias), which produced holes that exceeded the depth target (positive value). The sources of this error were attributed to the incorrect calibration of the offset introduced by the tool length (tool dimensions) and/or the measurement errors introduced by the inspection camera (vision system). In addition to the bias, there was also a random noise component as result of the complexity of the machining process. These random fluctuations around the specified depth target (plus the bias value) consisted of what was referred to as process noise N_{P1} in Eq. (4.2) (ref. to Section 4.2). The semi-finish depth variability introduced by the perturbation variable showed a good fit with a Gaussian distribution, confirming its intended purpose and it had a mean value $\mu = 0.27$ and standard deviation $\sigma = 0.18$.

As stated in the beginning of this section, Dataset 1 contained examples (instances) of the feature vectors extracted during the first cut (according to the methods described in Section

5.4) together with the corresponding DoC value (semi-finish level) of each single hole of the panel. Using a supervised learning approach, a regression model of the machining process (Model 1) could be built from Dataset 1. The inputs of the model consisted of the feature vector extracted from the sensor signals and the output variable, i.e. the one to predict, was the semi-finish DoC value. The model was trained in a number of instances (training set) and then predictions were made on new, previously unseen input data (testing set). The accuracy of the model was measured in terms of the Mean Squared Error (MSE) of the predictions, also referred as the out-of-samples MSE and the model was validated by means of the k-fold cross-validation method described in Section 2.2.5.2.

Before reviewing the depth data obtained from the second cut, it is convenient to summarise the instances included in Dataset 1. At the end of the trials, Phase 1 (panels P1-P4) provided 176 samples of sensor signals with 228 extracted features each (VIB3 was disconnected), and Phase 2 (panels P5-P7) provided 130 samples (two holes were discarded due to bad signal quality) with 292 extracted features each (here, all the features extracted from VIB3 signal were included). Considering both phases of the experiment, Dataset 1 contained a total of 306 instances with 228 features each (VIB3 features excluded for consistency), together with the 306 semi-finish DoC values (output variable) calculated from the camera measurement.

6.2.2 Dataset 2: Final Depth Data

The final depth variation values (i.e. the final depth errors) calculated from the camera measurement after the second cut, for all the holes considered in the experiment, are shown in Figure 6.3. Since these values represented the overall process performance variable, in order to comply with the process requirements, they all needed to be within the process tolerance of $\pm 0.2\ mm$. The depth error data plotted in the figure presented a decreasing trend with the number of holes (illustrated with the dashed black line). This means that the process quality oscillated from holes with almost perfect sizes at the beginning of P1 to very shallow holes (few were under the tolerance) at P6 and P7. Overall, almost all the holes were concentrated in the lower half interval of the process tolerance, indicating final depth values under the nominal depth $d_2 = 2.74\ mm$.

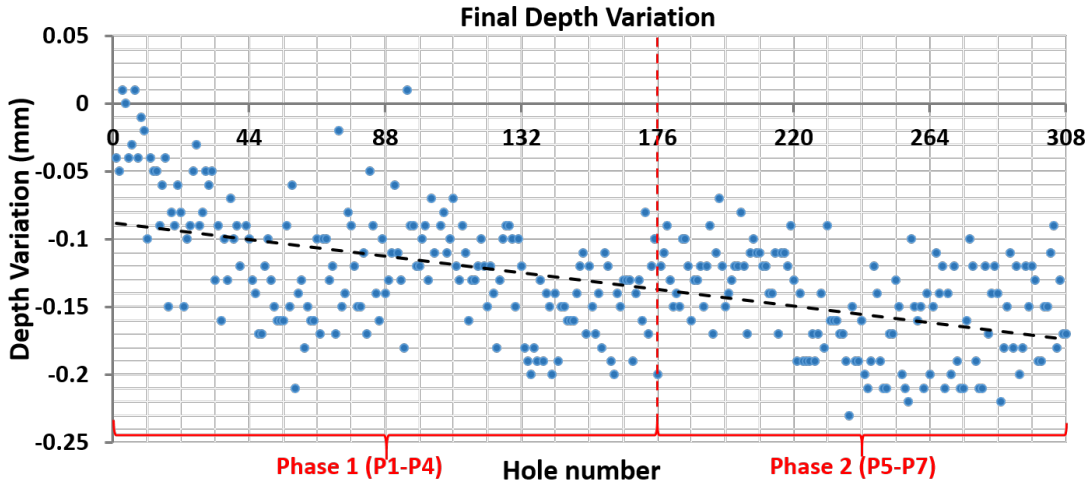

Figure 6.3 Depth variation data obtained after the second cut (final depth).

In all the trials, the target of the second cut was adjusted from the previous camera depth measurement relative to the first cut of the same hole. The new target value was computed as the difference between the nominal target (d_2) and the previously obtained semi-finish depth error. This corrected value was then specified into the spindle drive controller before the second cut commenced. The Perturbation value added to the target of the first cut was passed unchanged onto the second cut target since it was specific to the hole, not to the particular cut (ref. to the robotic disturbance in Figure 4.2).

The trend observed in the final depth error data indicated that correcting the second cut target from the previously measured semi-finish error did not completely compensate for all the error sources of the second cuts. The average magnitude of the final depth error became larger as the number of holes increased. This means that despite compensating for the semi-finish errors in full, the process behaved differently in the second cuts, producing continuously under-target holes as more holes were machined going from P1 to P7. The most plausible explanation for this behaviour was related to the cutting tool wear (not directly monitored here), which increases with the number of cuts causing the tool to remove less material (in average) for the same target specification. Moreover, since the second cuts were performed on the remaining depth of the current hole, less material than the first cuts was typically removed and, as the tool wear increased, it became more difficult to accurately remove small quantities of material (less depth resolution). In order to capture this behaviour during the training phase of the process models, two more features (independent from the sensor signals) were added: (i) Tool Hole Counter (THCount), which counted the number of holes that the tool had machined so far, starting at one at first hole of P1, and (ii) Hole

Position (HPos), which was the normalised value of the current hole number within the 44-holes of the entire panel.

Figure 6.4 compares the depth variability of the first cuts and second cuts for all the holes considered in the experiment. The graph in the left hand side is a histogram of the obtained data that illustrates the semi-finish levels (first cuts) in blue and the final depths (second cuts) in orange. The graph in the right hand side is a box plot that emphasises the range of values of the depth variation data obtained from both cuts, respectively. The central red mark indicates the median, the box extends from the 25th to the 75th percentiles and the outliers are individually plotted in red crosses. These two graphs indicate that the target adjustment method based on the camera measurement (of the semi-finish level) has reduced the process depth variability (caused by the Gaussian perturbation) from the initial 1 mm range (interval [-0.3 0.7] mm) of the first cuts into the final 0.2 mm range (interval [-0.2 0] mm) for the second cuts, obtaining an 80% reduction in size.

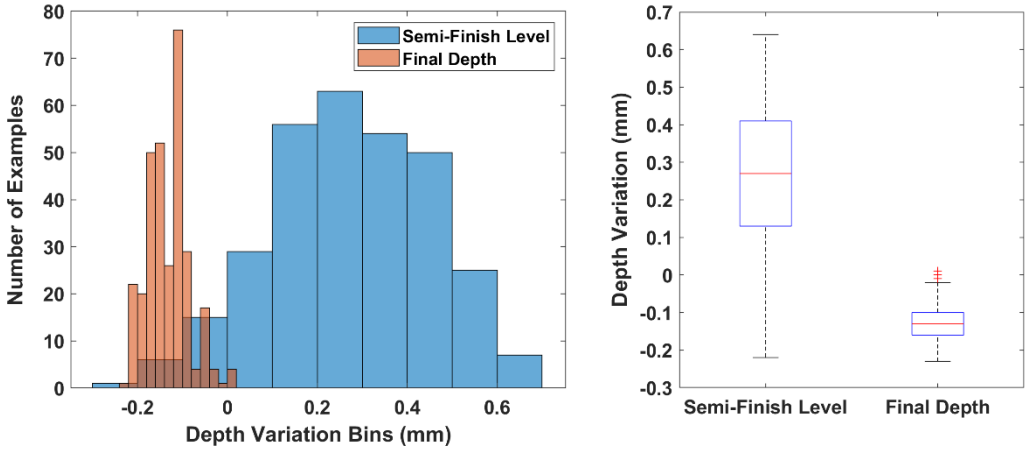

Figure 6.4 Comparison between the depth variability of the first cuts (semi-finish level) and the second cuts (final depth): histogram (left) and box plot graph (right).

In terms of the supervised learning approach, Dataset 2 contained examples of the features extracted during the second cut and the corresponding depth value. It is important here to distinguish between the final DoC value, the depth variation (error) value and the actual depth of the second cut. The final DoC refers to the value obtained from the inspection camera after the second cut; the final depth error is the difference between the above camera measurement and the nominal target d_2. The actual depth, however, refers to the depth machined during the second cut, which could be computed as the difference between the two camera measurements of the same hole. These depth values were the outputs to predict from the features extracted in the second cuts. Figure 6.5 shows the actual depth values of the second cuts for all the holes considered in the experiment.

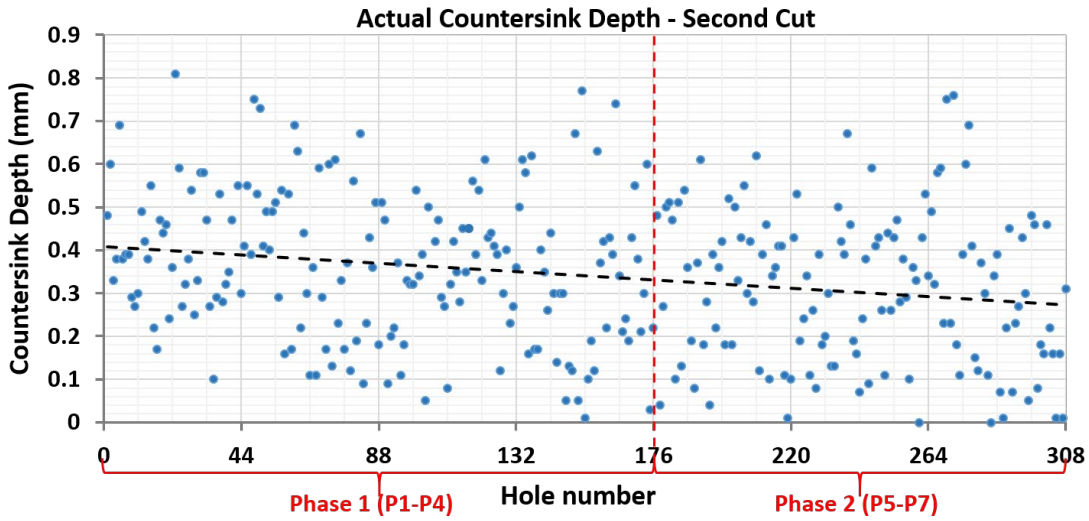

Figure 6.5 Actual countersink depth machined during the second cut.

Note that the depth values were distributed in the interval [0 0.9] mm and presented a linear decreasing trend (illustrated with the black dashed line), similar to the final depth errors showed in Figure 6.3. This confirmed the above tool wear theory that caused the cutting tool to remove less material (on average) as the number of cuts increases, even though the target remained constant (on average). In fact, the semi-finish depths were normally distributed in the interval [-0.3 0.7] mm, with no apparent trend in the data (ref. to in Figure 6.3), which, in turn translated into normally distributed targets for the remaining depth of the second cuts. However, as observed in Figure 6.5, these targets led to actual depth values that continuously got smaller over time, as more cuts were performed.

The data of Dataset 2 was used to build a regression model for prediction of the second cut's depth values (Model 2). The inputs of the model consisted of the feature vectors and the output variable was the actual depth obtained from the second cut. The same considerations previously stated for Model 1 in terms of the training and testing sets, model accuracy and validation apply to Model 2 as well.

The size of Dataset 2 was as follows. Phase 1 (panels P1-P4) provided 174 samples of sensor signals (two holes were discarded due to bad signal quality) with 228 extracted features each (no VIB3 features), and Phase 2 (panels P5-P7) provided 132 samples with 292 extracted features each (all features included). Considering both phases of the experiment, Dataset 2 contained a total of 306 instances of 228 features (again VIB3 features were excluded for consistency) together with the 306 actual depth values (process output variable).

To conclude this section, Table 6.1 summarises the two datasets extracted from the experiment in terms of their number of instances and features during each phase.

Table 6.1 Size of the two datasets extracted from the experiment.

Experiment	Dataset 1		Dataset 2	
	Instances	**Features***	**Instances**	**Features***
Phase 1 (P1-P4)	176	230	174	230
Phase 2 (P5-P7)	130	294	132	294
All Panels (P1-P7)**	306	230	306	230

(*) Tool Hole Counter and Hole Position are included in the total number
(**) Features from VIB3 signal are excluded for consistency with Phase 1

6.3 Feature Selection and Regression Model

This section describes the feature selection method used to reduce the number of signal features extracted from the experiment as well as the process regression models (ML algorithms) applied for the prediction of DoC values, relative to the first and second cuts of each hole of the panel. The analysis and the methods presented below were all implemented in MATLAB using the Statistics and Machine Learning Toolbox.

The feature selection strategy consisted of an embedded approach (ref. to Section 2.2.4) integrated with the learning algorithm of the model, which was suitable for both datasets discussed in Section 6.2. The method applied the Gaussian Process Regression (GPR) to perform Automatic Relevance Determination (ARD), as explained in the Appendices of this thesis (ref. to Section A.1). This can be obtained by selecting a particular type of covariance function for the Gaussian Process (GP), such as the Squared Exponential (SE) with variable length-scale parameter, also called the ARD-SE kernel defined in Eq. (A.9). Choosing an ARD-SE kernel in a GPR model, automatically assigns weights to the input dimensions (signal features) according to their relevance, in the optimisation procedure during training. This means that at the end of the training phase, the features with higher weights will have greater influence in the model's output variable than those with lower weights. Therefore, setting a threshold on the weight value would effectively reduce the original feature space by removing all the irrelevant features that fall under such threshold.

From the learning point of view, GPR models were considered due to their ability to deal with uncertainty in a probabilistic (Bayesian) framework and to provide error bars (confidence intervals) in the predicted value. In addition, such models require less training data, can

incorporate new knowledge (evidence) as more data becomes available and, since the number of hyper-parameters to optimise during training is typically low, they are less affected from the overfitting problem. In the tests conducted by this study, the initial values for the hyper-parameters of the GPR model (relative to the GP prior distribution) were determined from the training data according to the GP framework described in Section A.1 of the Appendices. The parameters of the GPR model with ARD-SE kernel were set as follows:

- the initial signal standard deviation $\sigma_f = \frac{std(\underline{y})}{\sqrt{2}}$
- the initial individual length-scale for each feature $\ell_i = \sqrt{d}$
- the initial noise standard deviation $\sigma_n = std\left(\underline{y}\right)$

Where $std(\cdot) = \sqrt{Var(\cdot)}$ refers to the standard deviation of the distribution function of the observations $\underline{y}$ and d is the number of features in the dataset (i.e. the input dimension). In absence of observations, the above hyper-parameters can be initialised with values estimated by previous knowledge of the process. In this work, the default values (i.e. no training data available) of σ_f, ℓ_i and σ_n were set at 0.15, 15 and 0.1, respectively.

Note that the feature weights (computed from the length-scale parameter ℓ_i) are optimised during the training of the GPR model, therefore they will depend on the particular training set. This means that using training sets of different sizes and/or containing different data samples will result in different weight values for the features. Hence, a strategy was required in order to obtain good generalisation properties. The simplest (and intuitive) solution was to utilise all the available training data to get the feature weights, however this would leave no testing samples for the validation of the (trained) model performance. Another option was the cross validation approach presented in Section 2.2.5.2, which divides the original dataset in training and testing data. In order to keep the training set as large as possible, in the tests here, the Leave-One-Out Cross Validation (LOO-CV) method was applied (ref. to Section 2.2.5.2). All the available data but one sample was used to train a GPR model with ARD-SE kernel, which was then tested on the left out sample. The process was repeated for all the samples in the dataset, building a new model each iteration and recording the feature weights assigned during training. At the end of the procedure, the weight of each feature was calculated as the average over the values assigned to that feature by all the trained models. The features were then sorted based on their weight and only those with weights over the pre-set threshold value were selected for further analysis. This method was computationally expensive and the time to train all the models increased considerably with

the size of the dataset, however, for the purpose of these tests (offline analysis), it provided more robustness in the assessment of the features relevance as well as a way to assess the generalisation performance of the GPR models. In a real-time scenario such as the online analysis presented in Chapter 7, the features would be assessed using all the currently available training data to build only one GPR model and a prediction would then be made on the new data as it is collected.

The two phases of the experiment were treated separately due to the different initial number of features (Phase 1 didn't contain the VIB3 features, whereas in Phase 2 they were included). For each experimental Phase (P = 1 or 2), two GPR models, denoted as Model 1.P and Model 2.P, were created using the data from Dataset 1 and Dataset 2, respectively. The threshold value on the overall (normalised) weights of the features was set at 0.001. The following sub-sections present the results obtained from the above feature selection strategy, comparing the selected features of both models, divided by the experimental phases.

6.3.1 Experimental Phase 1: Panels P1-P4

Figure 6.6 presents the overall normalised weights of the features assigned by the training procedure of Model 1.1 (top graph) and Model 2.1 (bottom graph). The names of the features that obtained higher weights are also indicated in the figure. As previously discussed, the feature weights were computed as the average over the values assigned by the various GPR models with ARD-SE kernel built in the LOO-CV method. The training data available in Dataset 1 and Dataset 2 (relative to Phase 1) was used to initialise the hyper-parameters.

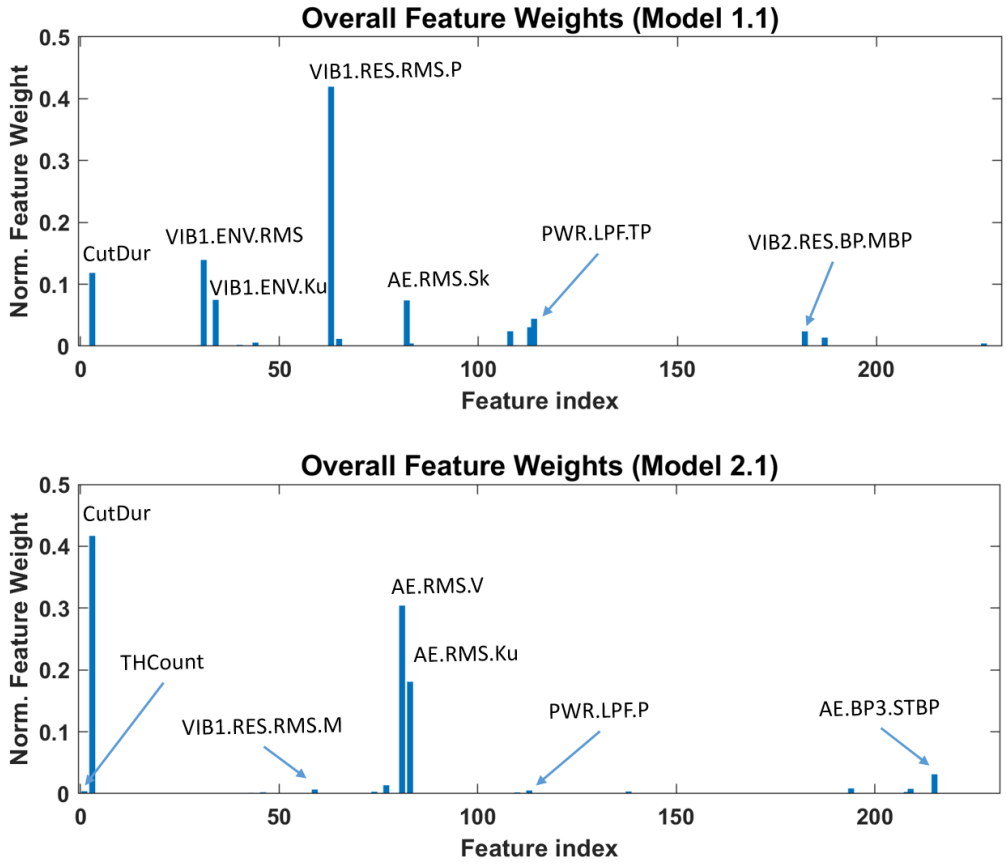

Figure 6.6 Overall feature weights assigned by the LOO-CV method for Model 1.1 (top) and Model 2.1 (bottom) relative to Phase 1 of the experiment.

The above two graphs indicate that the models assigned different weights to the features depending on the output variable to predict (i.e. the semi-finish DoC for Model 1.1 and the final DoC for Model 2.1). Model 1.1 appeared to be more sensitive to features extracted from VIB1 signal, with the top two being VIB1.RES.RMS.P and VIB1.ENV.RMS (ref. to Section 5.4 for the feature names), whereas Model 2.1 assigned higher weights to features extracted from AE signals, such as AE.RMS.V and AE.RMS.Ku. Both models identified the CutDur feature (duration of the cutting cycle) to be in the top three (first for Model 2.1), which was consistent with expectations. Moreover, the majority of the selected features came from the time-domain analysis rather than the frequency-domain.

At the end of the training process, the features were sorted according to their weights and a threshold value of 0.001 was applied for the selection. Table 6.2 summarises all the selected features (sorted by relevance) for each model, indicating their weight value, index and name. The selection method identified 19 features for Model 1.1 and 17 features for Model 2.1 as relevant to consider for further analysis.

Table 6.2 Signal features selected by the LOO-CV method for the experimental Phase 1.

	Dataset 1 Phase1 (Model 1.1)			Dataset 2 Phase 1 (Model 2.1)		
No	Normalised Weight	Index	Feature Name	Normalised Weight	Index	Feature Name
1	0.419	63	VIB1.RES.RMS.P	0.417	3	CutDur
2	0.139	31	VIB1.ENV.RMS	0.304	81	AE.RMS.V
3	0.118	3	CutDur	0.181	83	AE.RMS.Ku
4	0.075	34	VIB1.ENV.Ku	0.031	215	AE.BP3.STBP
5	0.074	82	AE.RMS.Sk	0.014	77	AE.Ku
6	0.044	114	PWR.LPF.TP	0.008	194	AE.PSD.STBP
7	0.031	113	PWR.LPF.P	0.008	209	AE.BP2.RSPB
8	0.024	108	PWR.LPF.M	0.007	59	VIB1.RES.RMS.M
9	0.024	182	VIB2.RES.BP.MBP	0.005	113	PWR.LPF.P
10	0.014	187	VIB2.RES.BP.STBP	0.004	1	THCount
11	0.012	65	VIB1.RES.RMS.CF	0.003	138	VIB1.PSD.THBP
12	0.006	44	VIB2.ENV.TP	0.003	74	AE.RMS
13	0.004	227	PWR.BP.KuBP	0.003	208	AE.BP2.STBP
14	0.004	83	AE.RMS.Ku	0.002	46	VIB1.RES.RMS
15	0.002	40	VIB2.ENV.V	0.002	110	PWR.LPF.V
16	0.002	30	VIB1.ENV.M	0.001	43	VIB2.ENV.P
17	0.001	109	PWR.LPF.RMS	0.001	195	AE.PSD.RSPB
18	0.001	225	PWR.BP.VBP			
19	0.001	195	AE.PSD.RSPB			

As previously noted, the LOO-CV approach used for training the GPR models provides also an accurate indication of their generalisation performance, since each sample of the dataset gets a prediction when it is left out for testing. This means that at the end of the procedure, all the samples will have a predicted value and the overall performance of the LOO-CV model is measured in terms of the Mean Squared Error (MSE) of such predictions with respect to the corresponding (true) target values, referred to as the Loss of the regression model. In order to assess the effectiveness of the selection method, the entire procedure was repeated for GPR models trained on the same dataset, but this time using only the selected features. A SE kernel (defined in Eq. (A.3)) was chosen for these GPR models to speed up the training phase, since there was no need to assign different weights to the (already) selected features. The results are shown in Figure 6.7, in terms of box plots of the prediction errors, comparing the two versions (all features vs. selected features) of each respective model: Model 1.1 (left graph) and Model 2.1 (right graph).

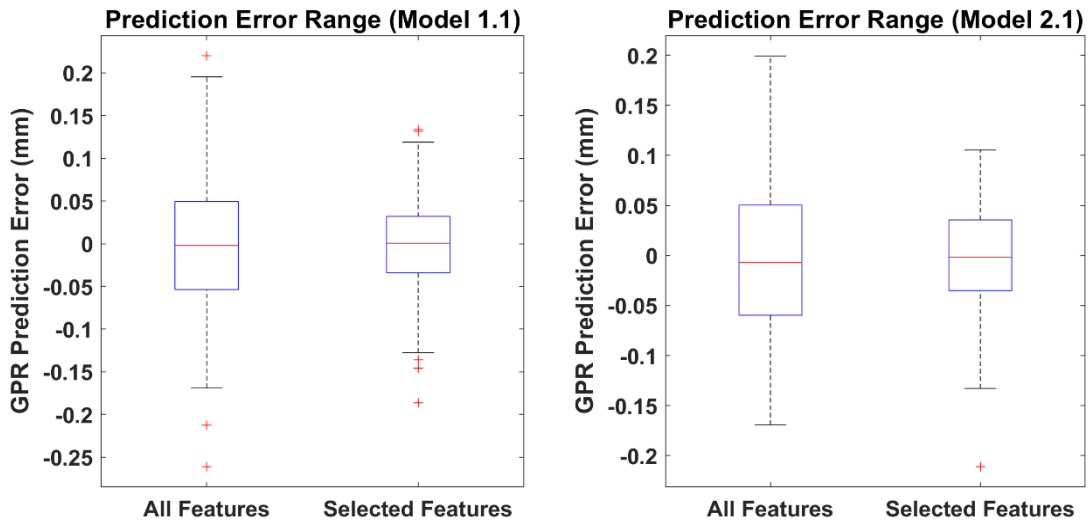

Figure 6.7 Box plot of the LOO-CV prediction errors of Model 1.1 (left) and Model 2.1 (right) comparing between models using all the features and only the selected ones.

The comparison results show that using only the selected features to train the GPR models reduced the prediction error range in both cases: Model 1.1 reduced its Loss value from 0.007 (all features) to 0.003 (selected features) and the Loss of Model 2.1 decreased from 0.005 (all features) to 0.003 (selected features). This confirmed the purpose of the proposed feature selection method in identifying relevant signal features that allowed to improve both training time and prediction accuracy of the models.

It is important to observe that the targets to predict by all models of Dataset 2 consisted in the process output variables relative to the second cut of every hole, i.e. the actual countersink depths machined during the second cutting cycle, as shown in Figure 6.5. These differed from the final DoC values measured from the inspection camera, which were used to compute the final depth errors illustrated in Figure 6.3.

6.3.2 Experimental Phase 2: Panels P5-P7

This sub-section presents the feature selection results of the LOO-CV procedure relative to Phase 2 of the experiment. Similar to Phase 1, the two GPR models with ARD-SE kernel built during Phase 2 assigned different weights to the features based on their contribution in the prediction of the output targets. Figure 6.8 illustrates the overall feature weights assigned by Model 1.2 (top graph) and Model 2.2 (bottom graph) as well as a few names of those with higher values. In Phase 2, the features of VIB3 signal were included into the initial feature set, increasing the total number of features from 230 (Phase 1) to 294.

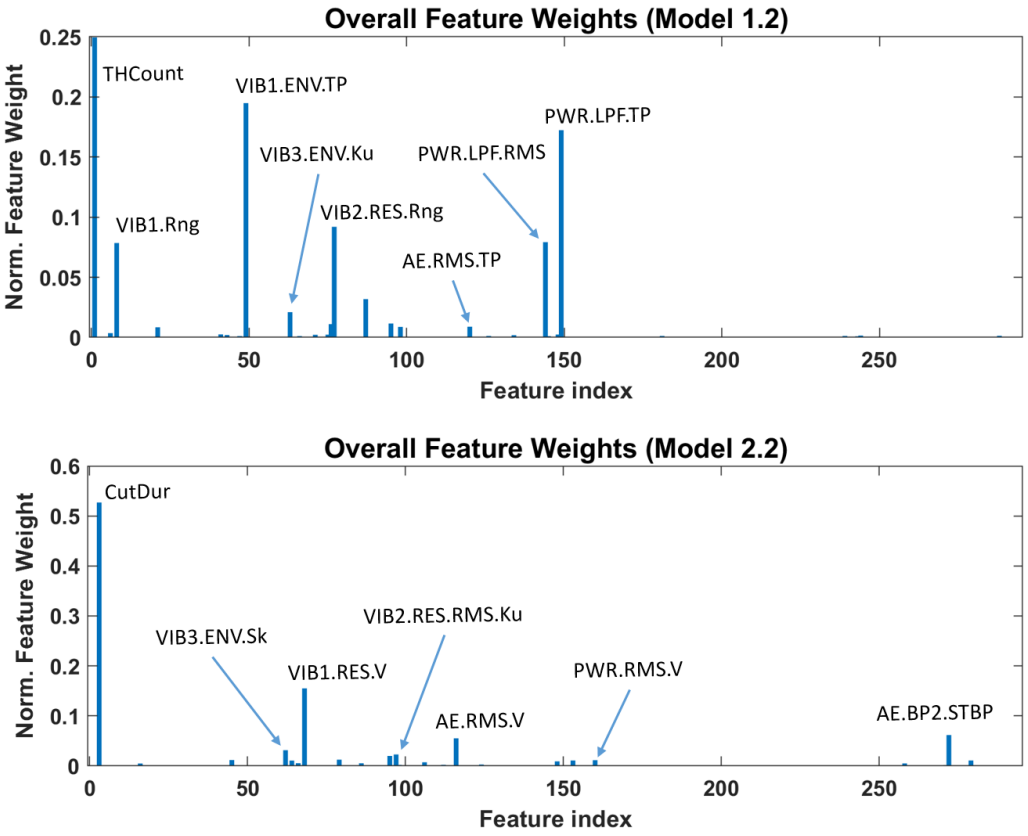

Figure 6.8 Overall feature weights assigned by the LOO-CV method for Model 1.2 (top) and Model 2.2 (bottom) relative to Phase 2 of the experiment.

Again, given the differences in the output targets, the two GPR models assigned different weights to the features. Model 1.2 identified THCount (Tool Hole Counter), as the most relevant feature, followed by VIB1.ENV.TP and PWR.LPF.TP, whereas for Model 2.2 the top three features included CutDur, VIB1.RES.V and AE.BP2.STBP. Both models assigned higher weights to features extracted from the time-domain analysis of the collected signals.

Note that the THCount feature was introduced to capture the linear (decreasing) trend observed in the final DoC values (ref. to Figure 6.3) caused by the progressive tool wear with the number of cuts. Such a trend was not visible in the semi-finish DoC data shown in Figure 6.1 due to the Gaussian Perturbation added to the semi-finish target, however, the large value of the THCount weight in Model 1.2 suggested that this trend was present when considering the data of the last 3 panels separately.

All the selected features obtained at the end of the training procedure during Phase 2 of the experiment are summarised in Table 6.3. The normalised weight value used for ranking the features, their index and name are also indicated in the table. The selection method (using

the same threshold value as in Phase 1) identified 28 relevant signal features for Model 1.2 and 26 signal features for Model 2.2.

Table 6.3 Signal features selected by the LOO-CV method for the experimental Phase 2.

	Dataset 1 Phase 2 (Model 1.2)			Dataset 2 Phase 2 (Model 2.2)		
No	Normalised Weight	Index	Feature Name	Normalised Weight	Index	Feature Name
1	0.250	1	THCount	0.527	3	CutDur
2	0.195	49	VIB1.ENV.TP	0.155	68	VIB1.RES.V
3	0.172	149	PWR.LPF.TP	0.062	272	AE.BP2.STBP
4	0.092	77	VIB2.RES.Rng	0.055	116	AE.RMS.V
5	0.079	144	PWR.LPF.RMS	0.031	62	VIB3.ENV.Sk
6	0.079	8	VIB1.Rng	0.023	97	VIB2.RES.RMS.Ku
7	0.032	87	VIB1.RES.RMS.V	0.020	95	VIB2.RES.RMS.V
8	0.021	63	VIB3.ENV.Ku	0.012	79	VIB3.RES.RMS
9	0.012	95	VIB2.RES.RMS.V	0.012	45	VIB1.ENV.V
10	0.011	76	VIB2.RES.Ku	0.011	160	PWR.RMS.V
11	0.009	120	AE.RMS.TP	0.011	153	PWR.TFF.V
12	0.009	98	VIB2.RES.RMS.P	0.011	279	AE.BP3.STBP
13	0.008	21	VIB3.CF	0.011	64	VIB3.ENV.P
14	0.004	6	VIB1.Sk	0.009	148	PWR.LPF.P
15	0.002	41	VIB3.RMS.TP	0.007	106	VIB3.RES.RMS.P
16	0.002	148	PWR.LPF.P	0.005	66	VIB3.ENV.CF
17	0.002	75	VIB2.RES.Sk	0.005	86	VIB1.RES.RMS.M
18	0.002	71	VIB1.RES.Rng	0.005	258	AE.PSD.STBP
19	0.002	43	VIB1.ENV.M	0.005	16	VIB3.RMS
20	0.002	134	AE.BPF.RMS.CF	0.003	124	AE.BPF.Sk
21	0.002	244	VIB2.RES.BP.STBP	0.002	112	AE.Ku
22	0.001	288	PWR.BP.MBP	0.002	98	VIB2.RES.RMS.P
23	0.001	181	VIB2.PSD.THBP	0.002	271	AE.BP2.PBP
24	0.001	239	VIB2.RES.BP.MBP	0.002	244	VIB2.RES.BP.STBP
25	0.001	126	AE.BPF.Rng	0.001	61	VIB3.ENV.V
26	0.001	47	VIB1.ENV.Ku	0.001	125	AE.BPF.Ku
27	0.001	66	VIB3.ENV.CF			
28	0.001	145	PWR.LPF.V			

Following the same analysis presented for Phase 1, the LOO-CV procedure was repeated for the datasets with the reduced feature subsets, training two GPR models with SE kernel and then comparing their performance with the original ones using all the features. Figure 6.9 presents the results in terms of the prediction errors (illustrated as a box plot) obtained by both versions of Model 1.2 (left graph) and Model 2.2 (right graph).

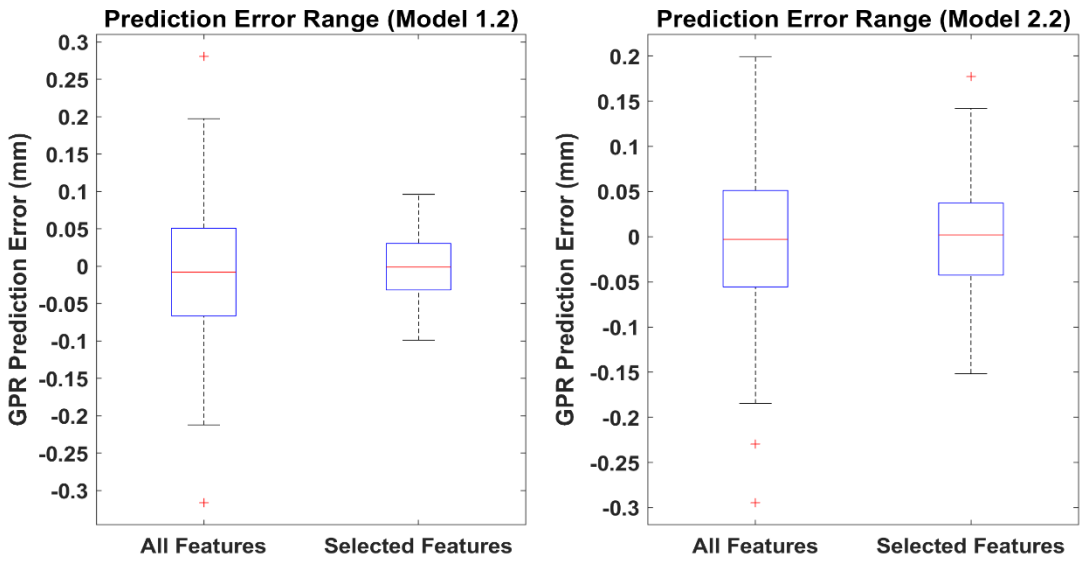

Figure 6.9 Box plot of the LOO-CV prediction error of Model 1.2 (left) and Model 2.2 (right) comparing between models using all the features and only the selected ones.

The comparison results, once again, confirmed the successful application of the proposed feature selection method: both models showed higher prediction accuracies when using only the selected features. In particular, Model 1.2 presented a big improvement with Loss values going from 0.008 to 0.002, which resulted in prediction errors within the range [-0.1 0.1]. Model 2.2, on the other side, reduced its Loss from 0.007 to 0.004.

6.3.3 Experimental Phase 1&2: Panels P1-P7

To conclude the discussion of the feature selection method, this last sub-section presents the results obtained by the LOO-CV method considering all the panels machined in the experiment (during both phases). The two datasets used to build the respective GPR models are those discussed in Section 6.2 (ref. to Table 6.1 for their sizes), i.e. Dataset 1 for Model 1 and Dataset 2 for Model 2. As previously noted, Dataset 2 ignored all the features of VIB3 signal for consistency with Dataset 1. Given the large size of the two datasets, the overall LOO-CV procedure of building 306 GPR models with ARD-SE kernel, every one of each optimises the 230 feature weights during training, required a considerable amount of time (1h to 1.5h), which is not acceptable in production environment. However, the purpose of the offline tests presented here was to assess the feature selection method with the best possible generalisation properties obtained by the LOO-CV method. The overall feature weights assigned by Model 1 (top graph) and Model 2 (bottom graph) are shown in Figure 6.10 with indication of the names of the features that obtained higher weight values.

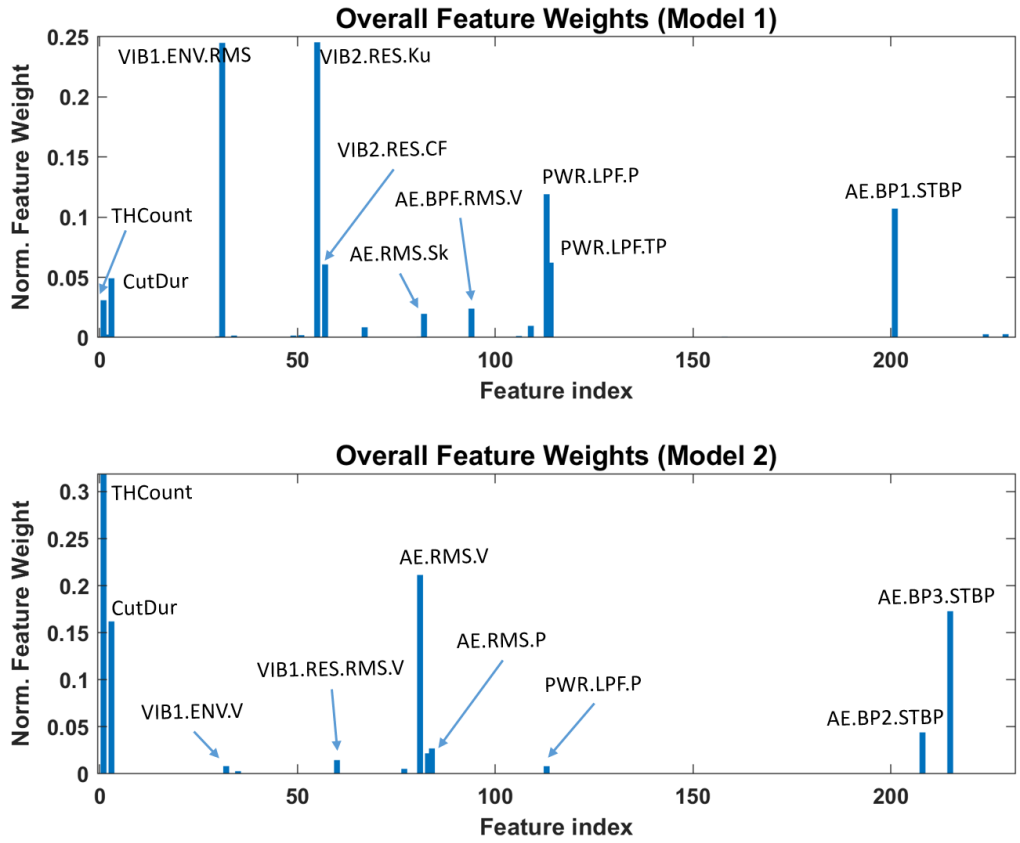

Figure 6.10 Overall feature weights assigned by the LOO-CV method for Model 1 (top) and Model 2 (bottom) relative to all the panels considered in the experiment.

The above two graphs indicate that Model 1 assigned higher weights to features extracted from VIB signals, such as VIB2.RES.Ku and VIB1.ENV.RMS, followed by PWR.LPF.P and AE.BP1.STBP, whereas Model 2 identified THCount as the most relevant one, followed by features extracted from AE signals (AE.RMS.V and AE.BP3.STBP) and CutDur. Note that both models agreed with THCount and CutDur, although in Model 1 they had lower values, to be included into the selected features. This was consistent with the previous results obtained when treating Phase 1 and Phase 2 separately. As expected, Model 2 assigned a high weight to the THCount feature due to the linear decreasing trend observed in the (actual) final depth data shown in Figure 6.5.

All the selected features obtained at the end of the training procedure during both phases of the experiment (considering all seven panels) are summarised in Table 6.4. Again, the weight values, indices and names of the features for both models are shown in the table. The number of signal features selected from the method was 19 for Model 1 and 12 for Model 2.

Table 6.4 Signal features selected by the LOO-CV method for all panels of the experiment.

	Dataset 1 – All Panels (Model 1)			Dataset 2 – All Panels (Model 2)		
No	Normalised Weight	Index	Feature Name	Normalised Weight	Index	Feature Name
1	0.245	55	VIB2.RES.Ku	0.319	1	THCount
2	0.245	31	VIB1.ENV.RMS	0.211	81	AE.RMS.V
3	0.119	113	PWR.LPF.P	0.173	215	AE.BP3.STBP
4	0.107	201	AE.BP1.STBP	0.162	3	CutDur
5	0.062	114	PWR.LPF.TP	0.044	208	AE.BP2.STBP
6	0.061	57	VIB2.RES.CF	0.027	84	AE.RMS.P
7	0.049	3	CutDur	0.022	83	AE.RMS.Ku
8	0.031	1	THCount	0.015	60	VIB1.RES.RMS.V
9	0.024	94	AE.BPF.RMS.V	0.008	32	VIB1.ENV.V
10	0.020	82	AE.RMS.Sk	0.008	113	PWR.LPF.P
11	0.010	109	PWR.LPF.RMS	0.005	77	AE.Ku
12	0.008	67	VIB2.RES.RMS.M	0.003	35	VIB1.ENV.P
13	0.003	229	PWR.BP.STBP			
14	0.003	224	PWR.BP.MBP			
15	0.002	2	HPos			
16	0.002	51	VIB1.RES.CF			
17	0.002	34	VIB1.ENV.Ku			
18	0.002	49	VIB1.RES.Ku			
19	0.001	106	PWR.TP			

Finally, Figure 6.11 presents the comparison results in terms of the prediction errors of the LOO-CV procedure between the two versions (all features vs. selected features) of Model 1 (left graph) and Model 2 (right graph).

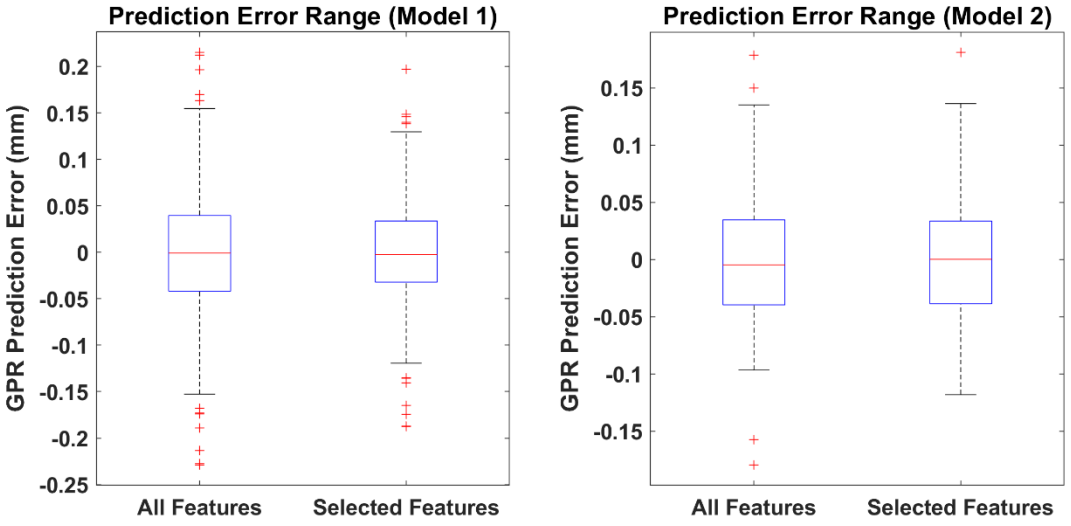

Figure 6.11 Box plot of the LOO-CV prediction error of Model 1 (left) and Model 2 (right) comparing between models using all the features and only the selected ones.

Similar to previous results, both models showed an improvement in the prediction accuracy when using only the selected features. The Loss of Model 1 decreased from 0.005 (all features) to 0.003 (selected features), whereas Model 2 already had achieved good results and it reduced slightly its Loss value from 0.0028 (all features) to 0.0026 (selected features) by removing a few of the outliners illustrated with red crosses in Figure 6.11. In addition, the overall LOO-CV procedure of training 306 GPR models (SE kernel) with the reduced feature subset took considerably less time (approximately 1 minute vs. 1.5 hours) compared to the case when using all the features.

6.4 Regression Results

This section provides more insights into the results obtained by the regression tests conducted in this work. The feature selection method proposed in the previous section proved to be effective in the identification of relevant signal features with respect to the model's output variable. This allowed to achieve higher prediction accuracies and faster training times for both the considered models: Model 1 and Model 2. Therefore, the regression tests discussed in this section will use the results obtained from the feature selection step and consider GPR models with SE kernel using only the selected features summarised in Table 6.4 for Model 1 and Model 2, respectively.

As previously mentioned, the LOO-CV method is an extreme case of the k-fold CV approach discussed in Section 2.2.5.2, where $k = N$, the number of samples (instances) in the dataset. It provides the best estimate of the generalisation performance of the model, since every sample is tested using the maximum training set possible. The results obtained by this method in terms of box plots of the prediction errors were briefly presented in Section 6.3, in which the models trained with the reduced feature subset were compared to the 'all features' case. Here, the regression results are discussed in more detail in relation to the accuracy of the GPR models. Figure 6.12 illustrates the predicted values (by the blue line), the true depth targets (by the red line) and the prediction errors (by the black dashed line), calculated as the difference between true and prediction values. The top graph includes the prediction results obtained by Model 1, whereas those of Model 2 are shown in the bottom graph. The LOO-CV procedure used GPR models with SE kernel (GPR-SE) trained on all the available data. The hyper-parameters (σ_f, ℓ and σ_n) of the GPR models were initialised based on the training data, as discussed in the beginning of Section 6.3. The reduced feature subset was applied to speed-up the overall training time and the feature values were standardised.

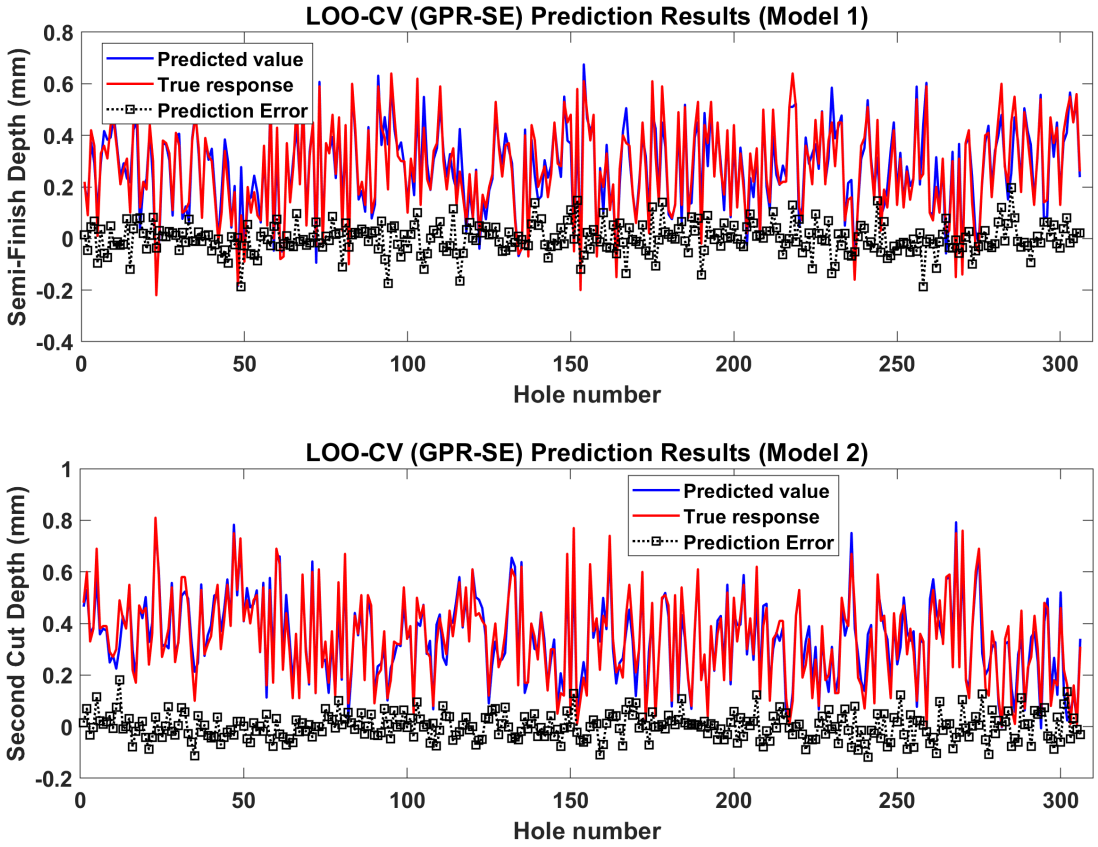

Figure 6.12 Prediction results of LOO-CV method relative to Model 1 (top) and Model 2 (bottom) using a GPR model with SE kernel trained on the reduced feature subset.

The histograms (with a Gaussian fit) of the prediction errors obtained by Model 1 (left graph) and Model 2 (right graph) are illustrated in Figure 6.13.

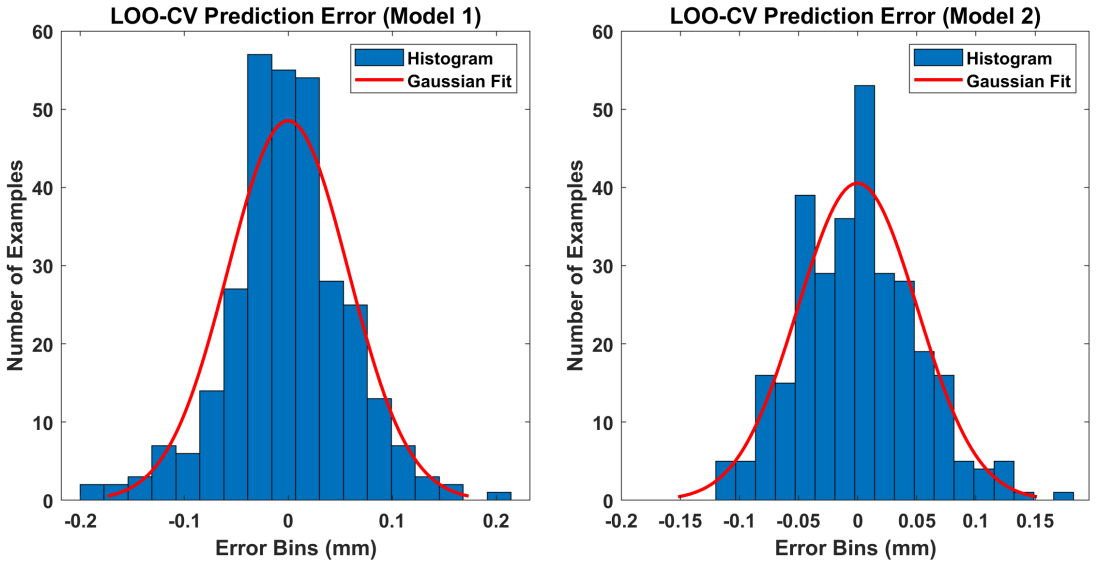

Figure 6.13 Histogram of LOO-CV prediction errors of Model 1 (left) and Model 2 (right).

The LOO-CV results of Figure 6.12 and Figure 6.13 indicated a good ability of the proposed GPR models to provide accurate depth prediction values. The prediction error of Model 1 (relative to the Semi-Finish DoC values) was within the [-0.2 0.2] mm interval with the majority of errors laying in the range [-0.1 0.1] mm. The overall model Loss (i.e. MSE of the predictions) was 0.0033. Model 2, on the other side, showed slightly better prediction accuracies with the error values (relative to the Final depths) within the [-0.15 0.15] mm interval. The majority of prediction errors belonged to the range [-0.05 0.05] mm and the overall model Loss was 0.0026.

To better understand the potential of a GPR model and fully exploit the power of its probabilistic framework, the following test was performed. First, 44 data points (equivalent to one panel of data) were randomly chosen from the entire dataset and set apart for testing. Then, a GPR-SE model was trained on the remaining dataset (equivalent to six panels of data). The model used the reduced feature subset with standardised values and the hyper-parameters were initialised based on the training set, i.e. the original dataset excluding the extracted 44 testing points. After the training phase was finished, the model provided an expected predicted output (with a mean value and its corresponding variance) for each testing point presented in input (ref. to Section A.1 for the details of GPR). This extra information of the variance represents a measure of the confidence of the model, given the current training set, and it was used to draw error bars in the output value.

An example of the results obtained by the above test for Model 1 (using the data in Dataset 1) is illustrated in Figure 6.14. The shaded region represents the 95% CI (Confidence Interval) of the prediction (mean) value indicated by the blue crosses. The true depth observations (relative to the semi-finish DoC values) are indicated by the red crosses. The 95% CI was calculated as the interval of two times the standard deviation (i.e. the square root of the predicted variance) around the mean value. Note that (almost) all the testing points were located inside the shaded region meaning that the model accurately identified the 95% CI.

The prediction errors together with the corresponding standard deviations provided by the GPR model are shown in Figure 6.15. The graph on the left hand side plots both the prediction errors (red line) and standard deviation values (denoted with std and indicated by the blue line) for each input point. On the right hand side, it is illustrated the histogram (with Gaussian fit) of the obtained prediction errors. All the std values that define the width of the 95% CI were located in a small range of data, between 0.005 and 0.008 mm. This indicated that the model was highly confident in the accuracy of the output value (depth prediction) due to the

fact that it had probably seen similar input data (with respect to the multivariate Gaussian input space) during the training phase. The model Loss value was 0.0024.

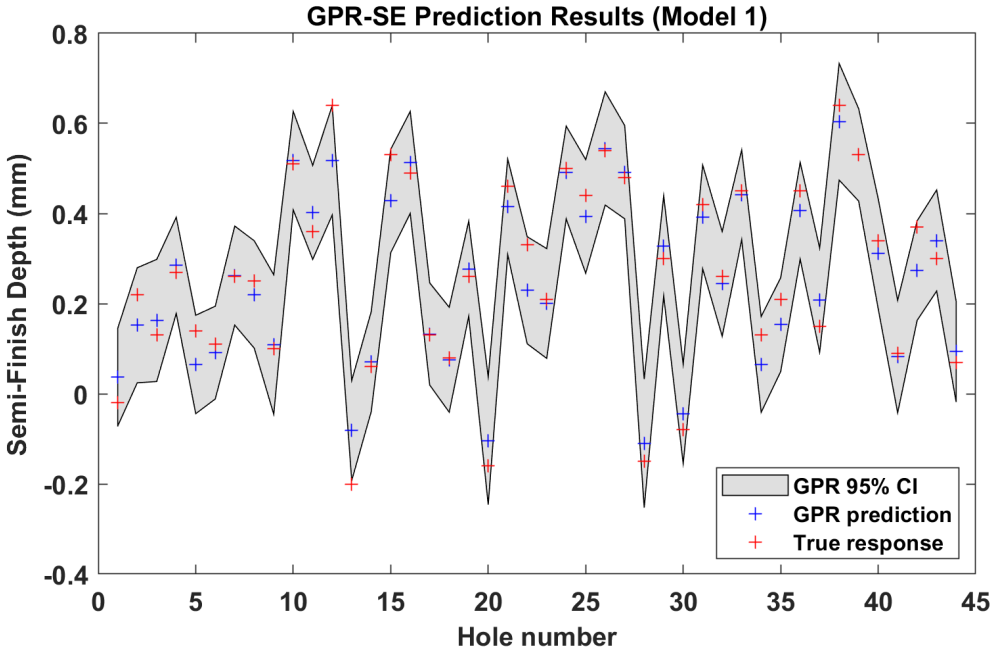

Figure 6.14 Prediction results of a GPR-SE model for 44 randomly selected testing points from Dataset 1 (Model 1).

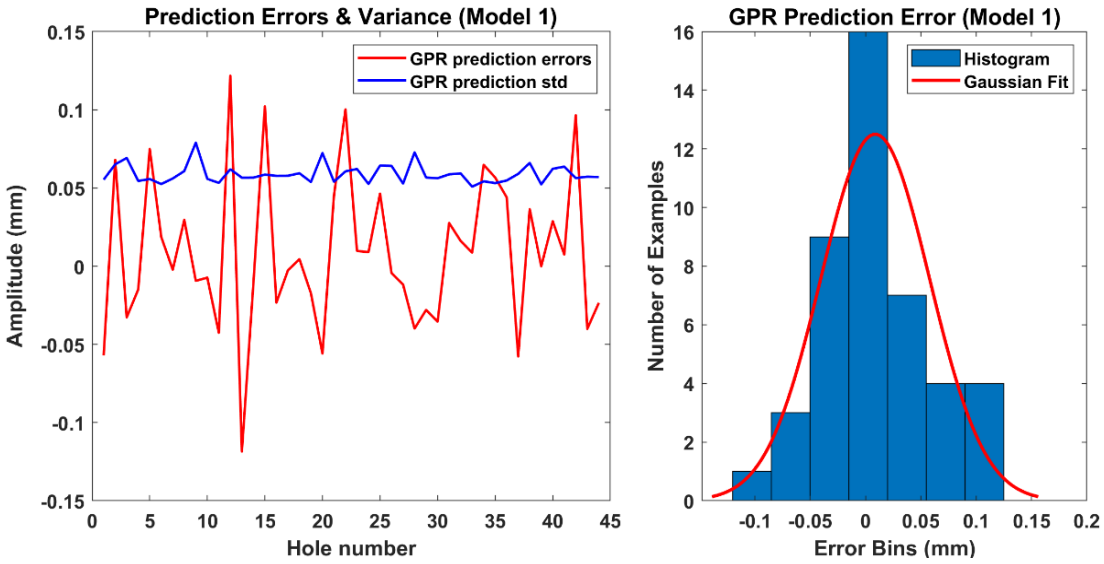

Figure 6.15 Test results for Model 1: prediction errors and standard deviation values (left); histogram of prediction errors (right).

Similar results were observed in the case of Model 2 when running the test with the data available in Dataset 2, as illustrated in Figure 6.16. The prediction values are indicated by the blue crosses, the true target values (relative to the second cut depths) are shown with the red crosses and the shaded region represents the 95% CI of the predictions. Note that the

model achieved good prediction accuracies and identified again the correct region where most of the target values were located.

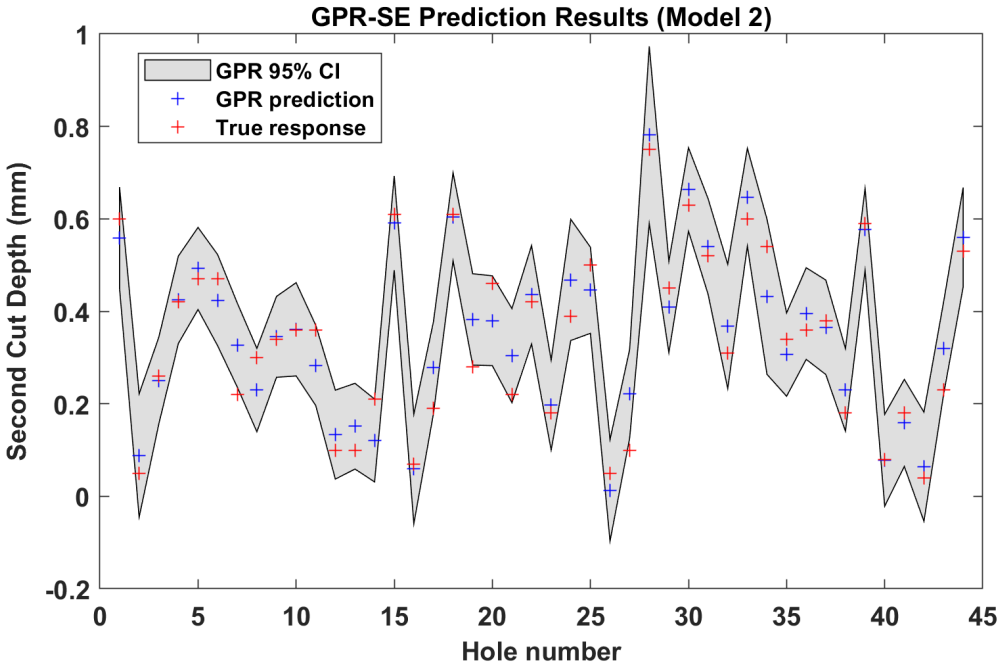

Figure 6.16 Prediction results of a GPR-SE model for 44 randomly selected testing points from Dataset 2 (Model 2).

Figure 6.17 illustrates the prediction errors and the corresponding std values (left graph), as well as the histogram of the prediction errors (right graph). Similarly, most std values were located in a small range around 0.05 mm (with a few spikes close to 0.1 mm), indicating the high confidence of the model in the prediction value. The model Loss was 0.0031.

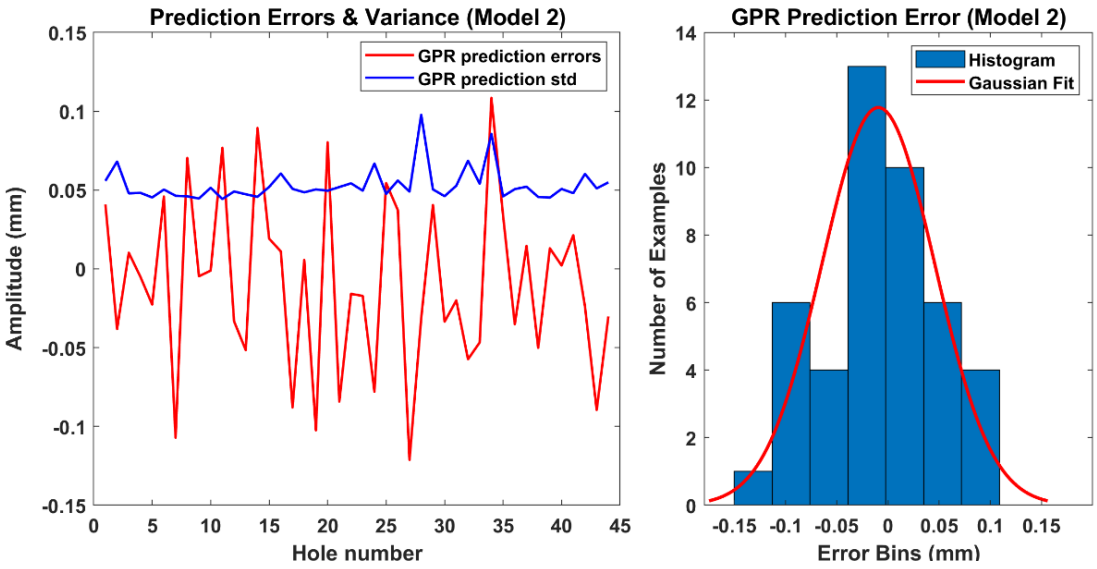

Figure 6.17 Test results for Model 2: prediction errors and standard deviation values (left); histogram of prediction errors (right).

Note that the prediction std values provide an indication of the model's confidence in the output value, given the current dataset used during training. This presents a big advantage of the GPR models (and the probabilistic framework in general) over other ML techniques, when dealing with uncertain data. For the problem considered here, this additional information of the predictive variance could be applied to drive the decision on whether to use the depth prediction or the direct camera measurement, increasing therefore the reliability of the model. Moreover, it is useful to mention that the variance estimation of a testing instance depends on the individual instance under test and the current training set. In particular, if the instance is far away (with respect to some metric measure) from the multivariate Gaussian distribution of the training set, then the model will output a large std value, suggesting a higher level of uncertainty in the prediction. Further information on the matter is provided in the next chapter (ref. to Section 7.1).

In conclusion, the results obtained by the LOO-CV method confirmed the ability of the proposed GPR models to accurately predict the countersink depth based exclusively on the information provided by the sensor signals during machining. The application of a probabilistic framework that handles both the selection of the most relevant features and the uncertainty estimation in the prediction values (95% CI region), helped to considerably improve the reliability and accuracy of the process models.

6.5 Chapter Summary

This chapter described the details and discussed the result of the regression tests performed on the data extracted from the experiment presented in Chapter 5. The relatively high number of training data as well as the depth variability introduced by the Gaussian perturbation added to the depth targets allowed the system to be trained with a variety of examples of sensor signals in a (relatively) wide range of depth data. The use of a probabilistic learning approach was investigated to test the prediction performance of the proposed process models. The additional information provided by the probabilistic framework regarding the prediction variance, was identified as a potential measure of the model's confidence. This allowed error bars to be drawn, resulting in more reliable process models. The feature selection step was integrated into the training phase of the models, implementing an embedded approach. The obtained results showed a good ability of the models to identify most relevant features and provide predictions of the true countersink depth value within an acceptable level of tolerance.

7 MULTI-STEP LEARNING PROCEDURE

The results of Chapter 6 indicated a good ability of the proposed process models to provide accurate predictions of the observed countersink depth values. This chapter moves on to discuss how a multi-step learning procedure can be applied to further exploit the capabilities of the monitoring system designed by this work. First, a way to assess the confidence of a GPR model is suggested and analysed in Section 7.1. Then, the multi-step learning method for the particular case study of robotic countersinking process is described in Section 7.2. An algorithm that simulates the in-process execution of the proposed system is presented in Section 7.3. Finally, the results of the simulation tests and further considerations on the learning approach are discussed in sections 7.4 and 7.5, respectively.

7.1 Model Confidence Assessment

At the end of Section 6.4, it was noted that the prediction standard deviation (denoted by std) expresses a measure of the model's confidence in the predicted output. This section explores the concept in more detail, analysing and discussing how the predictive variance of a GPR model is affected by the size of the dataset and the new input data.

In order to predict new testing inputs, a GPR model computes the covariance matrix of all pairs of training and testing inputs, considering a joint Gaussian distribution (ref. to Section A.1 for further details). In particular, with reference to Eq. (A.7), the calculation of the predictive covariance of the testing inputs depends only on the covariance matrices (of both training and testing input data), not the observations (output targets). Therefore, the choice of the covariance function used for the computation of these matrices will affect the predictive covariance of the GPR model. The SE kernel function adopted in this study (ref. to Eq.(A.3)) computes the covariance of a pair of input instances (i.e. feature vectors) in terms of their distance (Euclidean) in the d-dimensional feature space. Thus, inputs that are close to each other in the feature space will be highly correlated, while those far away uncorrelated. This means that new testing inputs far away from the multivariate Gaussian distribution defined by the training data will have higher std values, indicating that the model is not confident in the predicted outputs.

To illustrate how the model confidence is affected by the distance of the testing inputs, the following test was conducted. A GPR-SE model, similar to those considered in Section 6.4,

was trained using the input data of the first two panels P1 and P2 (i.e. 88 data points in total) and then, predictions of the remaining instances (relative to the panels P3-P7) were obtained. The test was performed in both datasets described in Section 6.2 and the results obtained are shown in the following two figures (Figure 7.1 and Figure 7.2).

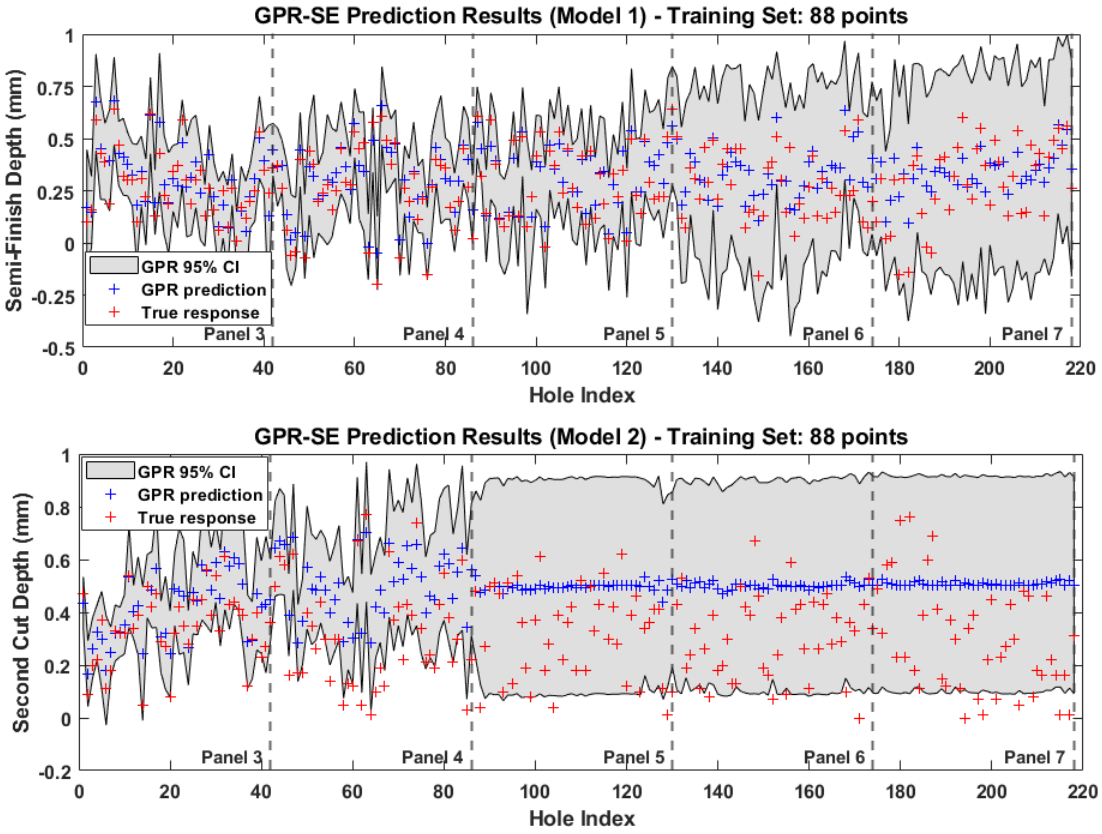

Figure 7.1 Prediction results obtained by using 88 training points (in order of collection): Model 1 (top) and Model 2 (bottom).

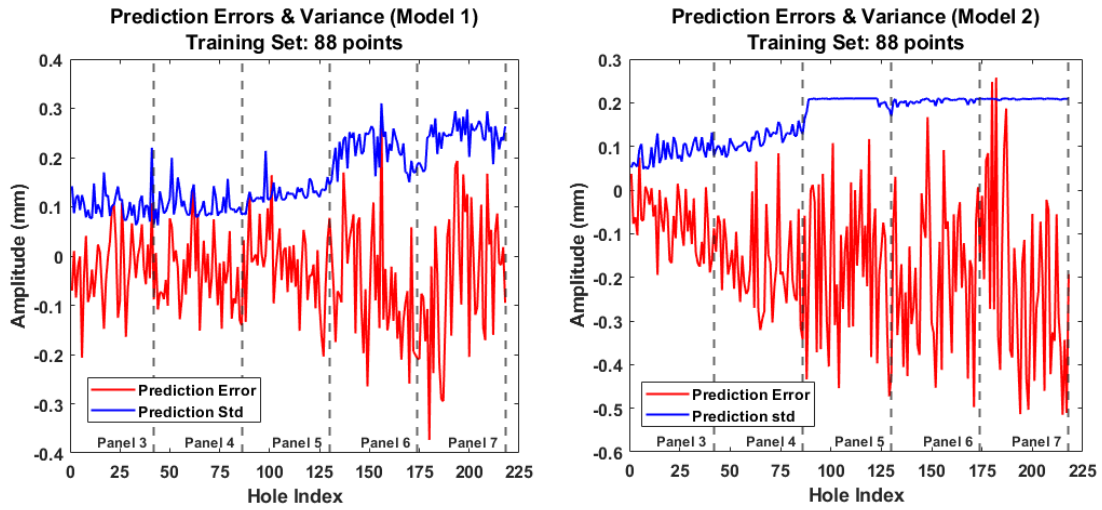

Figure 7.2 Prediction error and std values obtained by using 88 training points (in order of collection): Model 1 (left) and Model 2 (right).

Figure 7.1 presents the prediction results of Model 1 (top graph) and Model 2 (bottom graph): the predicted values are indicated by the blue crosses, the true outputs (i.e. the semi-finish DoC values obtained by the inspection camera) are shown with the red crosses and the shaded region represents the 95% CI of the predictions. The prediction errors (red line) and the corresponding standard deviations (std, shown by a blue line) of all the testing points are highlighted in Figure 7.2. The results of Model 1 are shown in the left graph and those of Model 2 in the right graph. The corresponding panel numbers of the testing instances are also indicated in both figures by the dashed vertical lines. The data is presented in order of collection, starting from the first hole of P3 (at index 0) to the last hole of P7 (at index 218).

A first examination of the figures reveals a continual growth of the shaded region of 95% CI with the number of holes from P3 to P7, indicating that the uncertainty in the predictions was continuously increasing. This is emphasised by the prediction std values illustrated in Figure 7.2, which showed a rise from an initial level of approximately 0.1 to 0.25 mm for Model 1 and from 0.06 to 0.2 mm for Model 2. The prediction errors were also getting larger in magnitude, especially for the last 2-3 panels, showing values in the range [-0.4 0.25] mm for Model 1 and [-0.5 0.2] mm for Model 2. Surprisingly, Model 2 appeared to provide a constant prediction value (with very high level of uncertainty) for all the depth values of panels P5, P6 and P7, failing therefore to accurately predict them.

This diversity between the performances of the two models can be explained by a closer examination of the patterns of sensor signals and the selected features of Phase1 and Phase 2 of the validation experiment described in Section 5.2. The two phases were performed in different time frames (and conditions), which caused slight variations in the observed signal patters. In particular, as previously mentioned (ref. to Section 5.3.2), a new segmentation method based on VIB1 data was necessary for Phase 2 due to the higher noise levels presented in the AE signals. In fact, this was also reflected in the next step of feature selection (ref. to Section 6.3). While most selected features of Model 2.1 came from AE signals, those of Model 2.2 were from the other sensor sources. Then, when considering all the panels for Model 2, still half of the selected features were related to AE signals (ref. to Table 6.4) due to the higher number of instances (cutting examples) considered in Phase 1. In the test conducted here, the GPR models used a SE kernel with the reduced feature set, therefore, the feature vectors collected from Phase 2 would have slightly different values (at least in half of the dimensions for Model 2) from those of Phase 1 used for training (i.e. panels P1 and P2). In other words, the testing instances coming from P5-P7 were far away (in terms of

the Euclidean distance in the feature space) from the training data of P1 and P2. For Model 2 in particular, the instances were far enough away to be uncorrelated. This explains the rapid increase of the predictions uncertainty at the beginning of P5: the model simply had not enough information to confidently predict the depth values.

A further examination of the figures reveals also a gradual increase of the prediction std values in both models with the number of testing instances form P3 to P7. This is due to THCount feature (selected by both models), which counts the number of holes the cutting tool has machined so far. It was introduced into the feature space after observation of the trend in the final depth data (ref. to Section 6.2.2). Since the same tool was used for all the panels, THCount gradually incremented its value as more holes were machined.

Increasing the number of training points to 180 and repeating the above test for both models yielded to the results illustrated in Figure 7.3 and Figure 7.4. The same notation of the previous figures is maintained to facilitate the comparison. Note that the hole number 180 refers to Hole 4 of P5 for Model 1 (Dataset 1 of Phase 1 included 176 holes) and Hole 6 of P5 for Model 2 (Dataset 2 of Phase 1 included 174 holes).

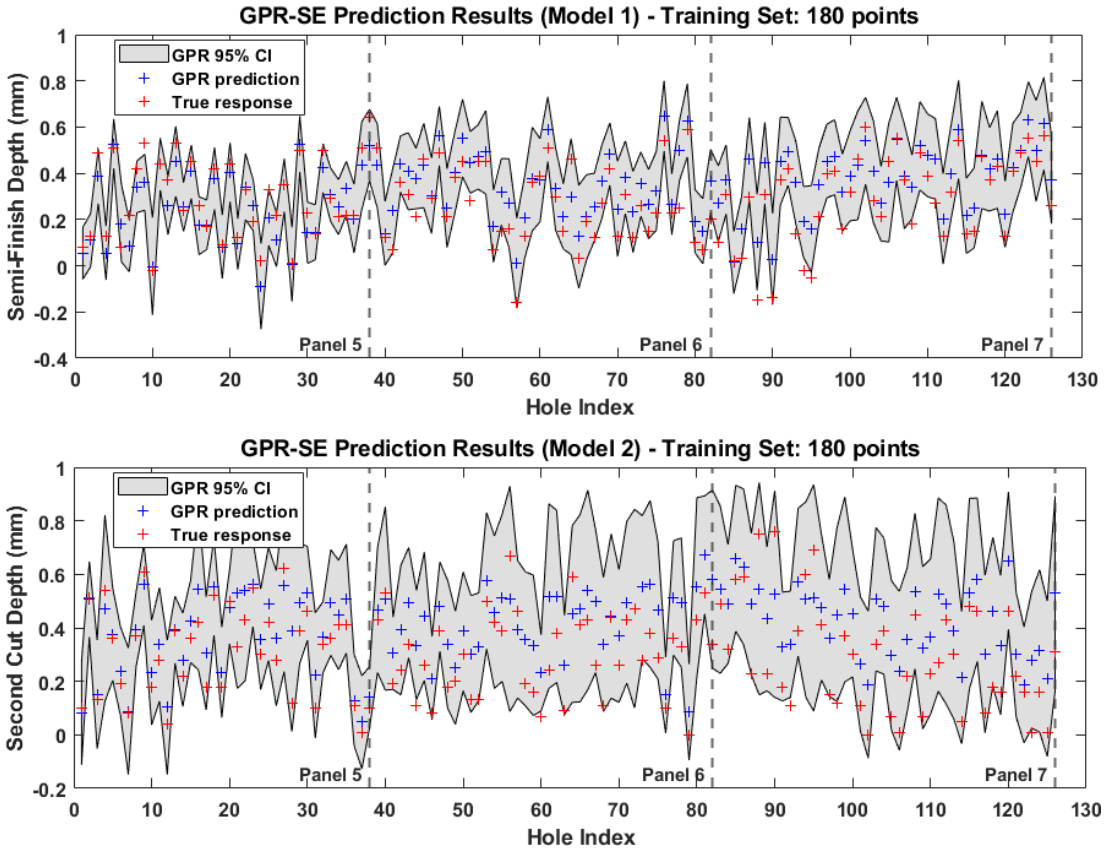

Figure 7.3 Prediction results obtained by using 180 training points (in order of collection): Model 1 (top) and Model 2 (bottom).

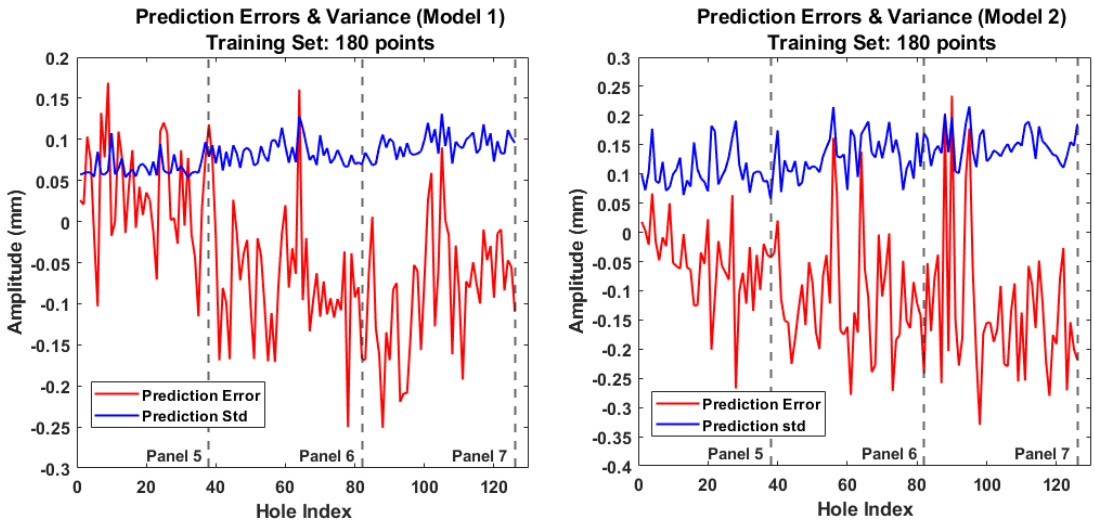

Figure 7.4 Prediction error and std values obtained using 180 training points (in order of collection): Model 1 (left) and Model 2 (right).

From the comparison of the two cases (80 vs. 180 training points) the following emerged:

- The inclusion of the first four data points (Model 1) or six data points (Model 2) from panel P5 had considerably improved the performance of the models with respect to the data of the experimental Phase 2.
- The shaded region of 95% CI presented a reduction in size in both models, indicating that the information gained from the first few holes of P5, helped to lower the uncertainty levels for the prediction of the remaining data of P5, P6 and P7.
- In both models the prediction errors and std values were reduced. As highlighted in Figure 7.4, the prediction errors were in the range [-0.25 0.15] mm for Model 1 and the range [-0.35 0.25] mm for Model 2, whereas the prediction std was within the range [0.06 0.12] mm for Model 1 and [0.05 0.2] mm for Model 2.
- Model 1 showed significant improvement of the prediction accuracy at (almost) acceptable levels, especially for the remaining data of P5. Note that Model 2 had gained enough information from the training data to provide predictions of the remaining depths of P5-P7. The accuracy was still not as good as Model 1, however it had improved considerably from the previous case.
- The gradual increase of the std values was still observed in the data of both models (although with a lower slope) due to the THCount feature as previously discussed.

In order to show how the confidence of the model is affected by the size of training set, the following test was performed. A subset of 44 randomly selected instances (one panel equivalent) from the entire dataset was left out of the analysis for testing, as a validation set.

Then, a series of GPR-SE models were trained on the remaining data, incrementing the training size by one instance every time, starting from a pre-set minimum up to the maximum available (i.e. the equivalent of six panels). The prediction accuracy of all the models was assessed always on the validation set (i.e. the data previously excluded from the dataset). The random selection of the validation set, ensures that the obtained accuracies of the models are minimally affected by the testing instances, as discussed in the previous test (i.e. their distances in the feature space from the training data).

The results of the above test for both Model 1 (top graph) and Model 2 (bottom graph) are shown in Figure 7.5. A minimum dataset of 22 points (half panel equivalent) was set as the initial dataset size. The results are expressed in terms of the model loss calculated as the MSE of the predictions (indicated by the red line) and the predicted std (indicated by the blue line) computed as the average over the 44 std values obtained from the testing instances.

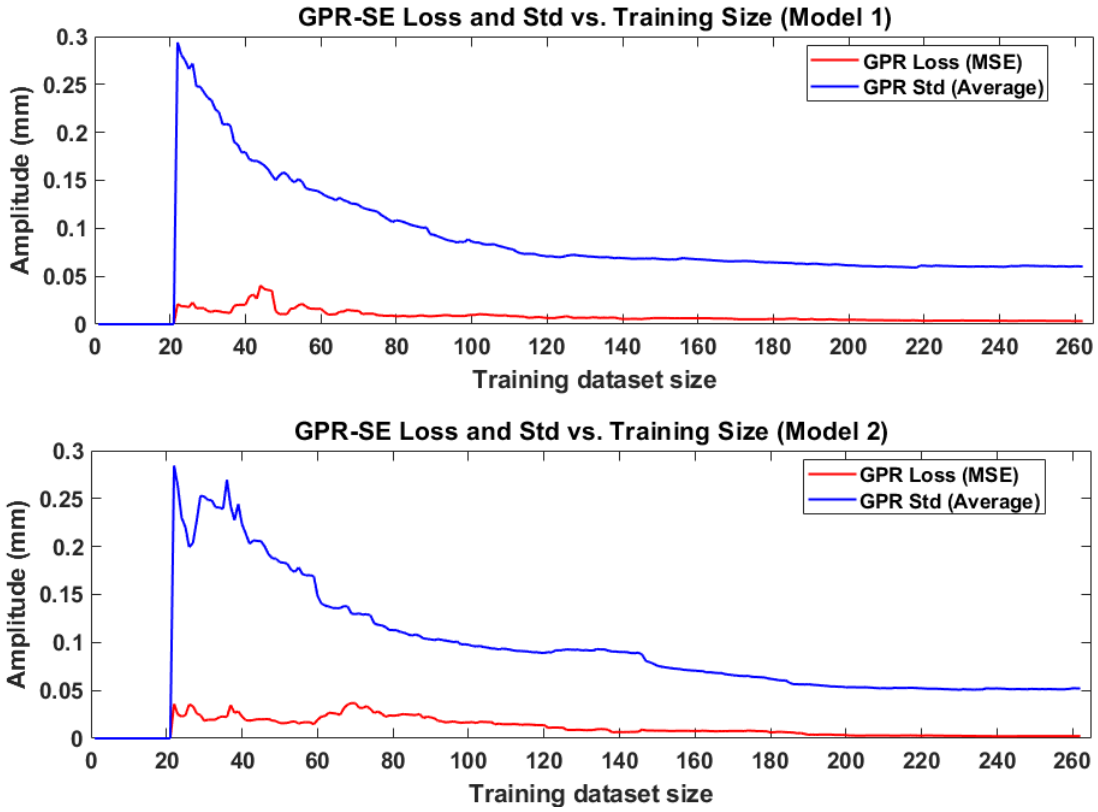

Figure 7.5 Assessment of GPR-SE model confidence: Model 1 (top) and Model 2 (bottom). The model loss and std are computed in a validation set of 44 randomly selected points.

It is apparent from the figure that incrementing the size of training dataset would reduce the amplitude of the model loss and the average prediction variance in both models considered in the test. The prediction std values presented initial peaks of approximately 0.29 mm for Model 1 and 0.28 mm for Model 2 when the dataset consisted of just 22 training data and

then, their values gradually decreased (although Model 2 showed a few initial oscillations) as more training instances were included in the dataset. The final levels of 0.06 and 0.05 mm for Model 1 and Model 2, respectively, corresponded at the maximum training size (six panels equivalent). Similar results were observed for the model loss, which values after a few initial oscillations, gradually decreased from a maximum of 0.04 and 0.036 to a minimum of 0.003 and 0.002 for Model 1 and Model 2, respectively.

The results shown in Figure 7.5 are an indication of the fact that both the model accuracy and the confidence in the predicted values increase as the size of the dataset increases. Moreover, it is possible to infer a minimum size of the dataset in order to obtain accurate prediction with high confidence. For the test presented here, this minimum size would be in the range 120 to 160 for Model 1 and 150 to 190 for Model 2, resulting in prediction std values of approximately 0.06 and 0.05 mm for Model 1 and Model 2, respectively.

In summary, the first test presented in this section showed how the model confidence depends on the testing instances, in particular on their Euclidian distance from the training data in the feature input space. It was observed from the experimental data that both models presented higher levels of confidence (i.e. lower std values) for the prediction of new testing instances close to those used for training, whereas for instances far away, the models showed lower levels of confidence (i.e. higher uncertainty) in the predicted outputs. The second test considered in this section illustrated how the model confidence changes with the size of the training data. It was shown that the confidence of both models in the prediction of a fixed-size and randomly selected validation set gradually increased as more data were included in the training set.

7.2 Proposed Learning Procedure

The learning method discussed so far in this thesis has considered an offline analysis of the monitoring system designed by this work using all the available experimental data to assess the performance (and confidence) of the process models. This section proposes an online version of the system to be used in the robotic countersinking process described in Chapter 3. In particular, this online method applies the multi-step approach presented in Section 4.2 into the particular case study of this work.

The diagram of the multi-step learning procedure is shown in Figure 7.6. The legend of the diagram is indicated in Table 7.1.

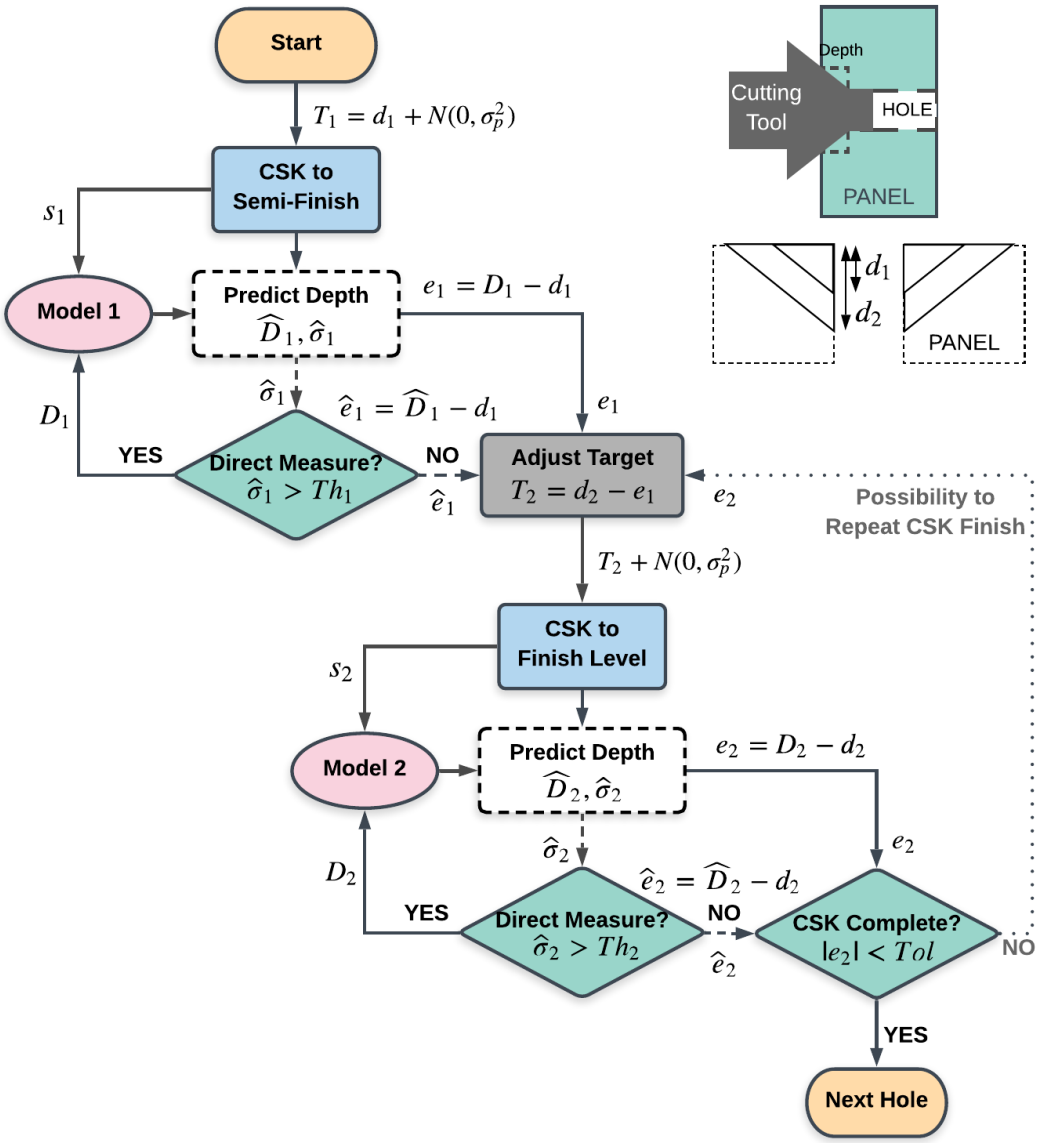

Figure 7.6 Diagram of the Multi-Step Learning Procedure.

Table 7.1 Legend of the Multi-Step Learning Diagram.

s_x	Sensor signals acquired during the first and second cuts
d_x	Nominal depth targets for the semi-finish and finish level
$N(0, \sigma_p^2)$	Gaussian perturbation programmatically added to target depth
T_x	Actual target depth sent to the spindle drive controller
D_x	True depth value as measured by the camera
$\widehat{D}_x$	Model's depth prediction (mean value)
$\hat{\sigma}_x$	Model's prediction uncertainty (std value)
e_x	True depth error (computed from depth measurement)
$\hat{e}_x$	Estimated depth error (computed from depth prediction)
Th_x	Uncertainty threshold value
Tol	Process depth tolerance value

(*) $x \in \{1, 2\}$ refers to the cutting cycle of a single hole (first and second cut).

With reference to Figure 7.6, the proposed learning method consists of the following steps:

1. The process starts by setting a target depth (T_1) to the spindle drive controller. The first cut is performed to a semi-finish level (d_1) plus the additional Gaussian perturbation $N(0, \sigma_p^2)$ with mean 0 and standard deviation σ_p.
2. Sensor signals (s_1) are acquired during the cutting process and the GPR model (Model 1) provides a prediction mean ($\widehat{D}_1$) and standard deviation ($\hat{\sigma}_1$) of the obtained countersink depth, given the current dataset. A minimum dataset is required before the model can start making predictions. This is set in advance.
3. Based on $\hat{\sigma}_1$, it is decided whether or not to obtain a direct depth measurement from the inspection camera. If $\hat{\sigma}_1$ is greater than a pre-set uncertainty threshold (Th_1), then the model confidence is low (ref. to Section 7.1), indicating that $\widehat{D}_1$ is likely to be wrong due to the high uncertainty levels, therefore a direct measure (D_1) is required. Whereas lower values of $\hat{\sigma}_1$ indicate higher confidence in $\widehat{D}_1$, thus there is no need for the inspection camera. Every time a direct measurement is obtained, the pair (s_1, D_1) is included in the current dataset of the GPR model.
4. A depth error value ($\hat{e}_1$ or e_1) is calculated from $\widehat{D}_1$ or D_1 depending on the result of the previous check on $\hat{\sigma}_1$, i.e. whether the inspection was performed or not.
5. The target depth of the second cut (T_2) is then adjusted by the above depth error. The second cut is performed to the adjusted finish level plus the Gaussian perturbation $N(0, \sigma_p^2)$ added in the first cut.
6. The above steps 2, 3 and 4 are repeated for the second cut, referring to sensor signals s_2, direct measurement D_2, GPR model (Model 2) with predictions $\widehat{D}_2$ and $\hat{\sigma}_2$, uncertainty threshold Th_2 and depth error $\hat{e}_2$ or e_2.
7. A final check on the amplitude of $\hat{e}_2$ or e_2 is performed to ensure it is within the process depth tolerance (Tol). The process continues to the next hole if the final check is successful, otherwise an option to repeat the second cut (step 5 and 6) is offered with the target depth adjusted by the latest true depth error e_2.

Note that the diagram in Figure 7.6 includes the possibility of repeating the second cut to ensure that the process would reach the desired depth level within the tolerance. In practical applications however, more than two steps in the same iteration of the cutting process would considerably increase the process cycle time, therefore, in the experiment conducted in this study it was decided to perform the second cut only once. The process continued to the next hole regardless to the final check result.

The Gaussian perturbation programmatically added to the target depth (of both cuts) was included in the process for two main purposes: (i) to provide the depth variability required for training of the models online, i.e. during the normal operation of the process (no interruptions) and (ii) to simulate large robotic disturbances by exaggerating the current process behaviour and obtaining extreme cases of depth errors.

Following from the diagram shown in Figure 7.6, the true depth value corresponding to the first cut can be expressed as (ref. to Table 7.1 for the terms):

$$D_1 = T_1 + N_{p_1} \tag{7.1}$$

Where N_{p_1} represents the noise that process adds to the semi-finish depth. The target depth for the first cut (T_1) is defined as:

$$T_1 = d_1 + N\left(0, \sigma_p^2\right) \tag{7.2}$$

Therefore, substituting Eq. (7.1) to Eq. (7.2) results as follows:

$$D_1 = d_1 + N\left(0, \sigma_p^2\right) + N_{p1} \tag{7.3}$$

From Eq. (7.3) the true depth error of the first cut can be expressed as:

$$e_1 = D_1 - d_1 = N\left(0, \sigma_p^2\right) + N_{p_1} \tag{7.4}$$

For the true depth value of the second cut (D_2) further attention in terms of the error sources is required. With reference to the process steps described in Section 5.1, it was mentioned that the master robot unclamped the panel after performing the first cut to allow the inspection camera to take an image of the machined hole. The robots remained in position during the inspection task. Then, the pressure foot was activated again before the second cut commenced. This re-clamping process of the pressure foot may introduce errors in the process that have not been observed during the first cut. Thus, the true depth of the second cut can be expressed as:

$$D_2 = T_2 + N_{p_2} + N_{rc} \tag{7.5}$$

Where N_{rc} is the additional noise caused by the re-clamping process and N_{p_2} is the process noise added to the final depth, as discussed previously for the first cut. The target depth for the second cut (T_2) is given by (ref. to Figure 7.6):

$$T_2 = d_2 - e_1 + N\left(0, \sigma_p^2\right) \tag{7.6}$$

Then, substituting Eq. (7.6) and Eq. (7.4) in Eq. (7.5) gives the following expression for D_2:

$$D_2 = d_2 + N_{rc} + \left(N_{p_2} - N_{p_1}\right) \tag{7.7}$$

The re-clamping noise N_{rc} can be considered negligible because both robots maintained the same position in the 3D Cartesian space during the camera depth inspection. Moreover, the errors originated from the movement of the panel were kept at minimum due to the dynamic stiffness created by the clamping forces.

The assumption made by this thesis when presenting the proposed multi-step approach in Section 4.2 was in regards to the errors introduced to the output variable by the two steps of machining process. It was argued that given the same process iteration, the errors of the first step at semi-finish level are of a comparable amount (within negligible variation) of the errors of the second step at finish level, which for the case study considered in this work means that $N_{p_1} \cong N_{p_2}$.

Under the above two assumptions the final true depth of Eq. (7.7) becomes approximately equal to the desired final depth target, i.e. $D_2 \cong d_2$.

When considering the depth predictions of both cuts rather than the true measured values, an additional source of error due to the model predictions needs to be taken into account. This model error (M_{Err}) is defined as the difference between the true (D) and the estimated (predicted) depth ($\widehat{D}$), i.e. $M_{Err} = D - \widehat{D}$. Following the same steps as above, the estimated depth values corresponding to the first cut ($\widehat{D}_1$) and the second cut ($\widehat{D}_2$) can be expressed as:

$$\widehat{D}_1 = d_1 + N\left(0, \sigma_p^2\right) + N_{p1} - M_{1_{Err}} \tag{7.8}$$

$$\widehat{D}_2 = d_2 + N_{rc} + \left(N_{p_2} - N_{p_1}\right) - \left(M_{2_{Err}} - M_{1_{Err}}\right) \tag{7.9}$$

Where $M_{1_{Err}}$ and $M_{2_{Err}}$ are the GPR model prediction errors of Model 1 and Model 2, respectively. Note that as seen by the previous (offline) analysis performed in this thesis (ref. to Sections 6.4 and 7.1), the prediction accuracy of the proposed GPR models improves as more training data are collected, therefore driving the model errors close to 0 through continual growth of the dataset would revert Eq. (7.9) back to the previous case of the true depths of Eq. (7.7), with $\widehat{D}_2$ approximately equal to the desired final level d_2.

7.3 In-Process Execution Algorithm

This section discusses the online version of the multi-step learning procedure illustrated in Figure 7.6. The pseudocode of such learning approach is shown in Table 7.2.

Table 7.2 Pseudocode of the online execution of the multi-step learning procedure.

	Multi-Step Learning Procedure (In-Process Execution)
1	**Set:** min Dataset (DS) size, min Feature Subset (FS) size and Uncertainty Threshold
2	**Initialise:** DS and FS variables
3	Start current iteration of the process
4	**if** current DS size < min DS size **then**
5	Get new input data (feature vector) and observation (direct measure)
6	Compute the true depth error (from direct measure) and Update DS
7	Adjust target depth and repeat steps 5-6 for the second cut
8	Check final depth and Proceed to the next iteration of the process
9	**else**
10	Get new input data (feature vector from sensor signals)
11	**If** Update DS = True (new data was added in the previous iteration) **then**
12	Apply feature selection step and Update FS
13	Train a new GPR-SE model on current DS and FS
14	**else**
15	Use current model to obtain the prediction Mean and Std of new input data
16	**if** Std > Uncertainty Threshold **then**
17	Get new observation (direct measure)
18	Compute the true depth error (from direct measure) and Update DS
19	Adjust target depth and Repeat from step 10 for the second cut
20	Check final depth and Proceed to the next iteration of the process
21	**else**
22	Compute the estimated depth error (from Mean prediction)
23	Adjust target depth and Repeat from step 10 for the second cut
24	Check final depth and Proceed to the next iteration of the process
25	**end if**
26	**end if**
27	**end if**

The algorithm starts by setting the model parameters and initialising the variables (lines 1-2). The current iteration of the process commences in line 3. The lines 4-8 ensure that the dataset reaches the minimum size necessary for the GPR models, collecting the camera depth measurement after every cut and updating the current dataset. Note that each model builds its own dataset, as explained in Section 6.2. Once the minimum dataset is available, the learning procedure continues by acquiring the new input data, i.e. the feature vector obtained from the sensor signals (line 10).

The lines 11-13 check whether the current dataset was updated in the previous process iteration (i.e. new depth data was added) and if positive, a new model (or re-training) is required to include the new information provided by the current (previously updated) dataset. The feature selection step is also included as part of the training process using the ARD method described in Section 6.3. Since the selected features depend on the current dataset, a new selection is necessary. The feature subset is then updated with the newly selected features (line 12) and a new GPR-SE model is trained using the current dataset and feature subset (line 13).

In line 15, the current GPR-SE model is used to make a prediction of the new input data previously acquired in line 10, providing the prediction mean and standard deviation (std) values. The lines 16-20 handle the case when the prediction std value is greater than the pre-set uncertainty threshold, meaning that the model has low confidence in the prediction value. A new depth measurement from the inspection camera is obtained and then the corresponding true depth error is added to the current dataset (line 17-18). The depth target (finish level) is adjusted and the above steps from the acquisition of a new input data (in line 10) are repeated for the second cut (line 19), referring to its own current GPR model, dataset and feature subset. Note that the current dataset is updated every time a new depth measurement is obtained by the inspection camera.

In the case of the model showing high enough confidence in the prediction value (lines 21-24), a new depth measurement is not necessary and the prediction mean can be used instead. The estimated depth error is computed and the final depth target is adjusted accordingly. Note that the current GPR model, dataset and feature subset remain unchanged since no new observation (depth measurement) is obtained. After correcting the depth target, the second cut starts and the procedure is repeated from step 10 (lines 22-23), similar to the previous case in line 19. Before proceeding to the next iteration (i.e. the next hole of

the panel), the final depth error (true or estimated) is checked towards the process tolerance limit (line 20 or 24, respectively).

The results of the above algorithm using the two datasets presented in Section 6.2 and simulating the online execution of the experiment are discussed in the next section.

7.4 Simulation Results

Having covered all other aspects of the proposed monitoring system in terms of the in-process data acquisition and extraction of sensory information as well as extensive offline analysis of the process models, this section presents and discusses the online execution of the system in the robotic countersinking process, as outlined in Section 7.2.

The experiment described in Section 5.1 was initially designed with the intention to test the in-process capabilities of the proposed monitoring system by following the multi-step learning procedure illustrated in Figure 7.6. However, due to technical issues of accessing the robotic cell, it was only possible to adjust the target depth of the second cut by means of the direct camera measurements, not the depth predictions. Therefore, the experiment was performed by using the inspection step in every iteration of the process (i.e. every single hole of the panel) to obtain the true countersink depth data of the first and second cut. The two datasets obtained from both phases of the experiment with respect to the semi-finish depth values (used for Model 1) and final depth values (for Model 2) were presented in Section 6.2. These datasets provided all the necessary data to run simulation tests of the multi-step learning algorithm described in Table 7.2 (i.e. the online execution of the system).

The aim of the simulation tests was to assess the validity and performance of the proposed learning method to identify instances of the available dataset where the direct depth measurement could have been ignored, without compromising the overall prediction accuracy of the models. In a real-time scenario, this would be translated as in-process suggestions on whether it could have been possible to avoid the inspection step, based on the dataset (i.e. process knowledge) available at the time of the current process iteration. For data consistency and ease of comparison between the two models, it was decided to remove from the considered dataset the instances that were missing in any of the datasets due to the bad signal quality (two from Dataset 1 and two more from Dataset 2). Therefore, the final datasets consisted of 304 cutting examples (instances) of 230 features for each of the considered models.

As previously mentioned, the multi-step learning procedure requires an initial dataset to start with, which is obtained during the initialisation phase of the algorithm by collection of the camera measurements for the pre-defined number of process iterations (minimum DS size). In simulation, this initial dataset was ensured by the inclusion of the available instances in order of collection starting from the first hole of Panel 1 up to the specified minimum size. In the analysis presented in Section 7.1, it was explained how the accuracy and confidence of the GPR models is affected by the training size, therefore the selection of a suitable value for the initial training size is of great importance, since it will influence all the subsequent process iterations. Furthermore, another parameter to specify prior execution is the uncertainty threshold defined on the prediction standard deviation value (denoted by std). This is the key parameter that drives the decision on the inspection step. Possible choices of these two parameters can be inferred from the previous offline analysis performed in Section 7.1. In particular referring to Figure 7.5, an initial dataset of 22 training instances (half panel equivalent) was used to present how the confidence of the model changes with the size of the training set. Despite the low accuracy and high uncertainty, the models were able to provide depth predictions with that minimal size of the dataset. Moreover, lower loss values of both models corresponded to std values in the range from 0.05 to 0.06 mm, indicating possible choices for the uncertainty threshold.

The learning algorithm presented in Table 7.2 requires the selection of another parameter corresponding to the minimum size of the Feature Subset (FS). This parameter was not used in the previous regression tests, since the features were selected based on a threshold defined on their normalised weights, as explained in Section 6.3. However, these feature weights were dependent on the training data and a few issues arose when considering a variable size training set such as that of the tests discussed here. In particular, for the small datasets at the beginning of the procedure, the feature weights threshold was typically selecting just a few number of features (from three to four) due to the lack of information, affecting all the subsequent iterations of the process. An attempt to lower the threshold value from 10^{-3} to 10^{-5} was made, however this did not consistently solve the issues related to the variable size of the dataset. It was therefore decided to define a minimum number of the selected features, which, in conjunction with the above modified threshold value, ensured the validity of the ARD-SE feature selection method.

In order to find the optimal values for the above three parameters (i.e. min DS size, min FS size and std threshold) a series of simulation tests were performed considering three possible

choices for the initial DS size (22, 43 and 65 corresponding to 0.5, 1 and 1.5 panels), three for the minimum FS size (15, 25 and 25) and two for the std threshold value (0.05 and 0.06). The tests were performed in a regular grid of all the possible combinations of these three parameters (18 in total) for each of the considered models. The results obtained by the tests are shown in Table 7.3 and Table 7.4 for Model 1 and Model 2, respectively.

Both tables present the results in terms of:

- The final size of the dataset.
- The overall prediction MSE computed at the end of the procedure.
- The range of prediction errors observed in the training instances (i.e. those included in the final dataset due to the higher prediction std values).
- The range of prediction errors of the remaining instances (i.e. those not included in the final dataset due to the model's higher confidence in the prediction value).

The tests that showed the best results are indicated in red. In terms of Model 1, the selected parameters were as follows (ref. to Test 4 in Table 7.3): initial DS size of 22 (half panel), minimum FS size of 20 and std threshold of 0.06 mm. Model 2 performed better in Test 10 with the following parameters: initial DS size of 43 (one panel), minimum FS size of 20 and std threshold of 0.06 mm.

Table 7.3 Tests results of the online execution of the proposed system for Model 1.

Test No	Init DS size	Min FS size	Std Threshold	Final DS size	Prediction MSE	Train Error Range	Pred Error Range
1	22	15	0.05	170	0.009	0.66	0.45
2	22	15	0.06	126	0.009	0.84	0.42
3	22	20	0.05	224	0.008	0.78	0.38
4	**22**	**20**	**0.06**	**129**	**0.009**	**0.76**	**0.41**
5	22	25	0.05	149	0.009	0.66	0.53
6	22	25	0.06	85	0.011	0.88	0.50
7	43	15	0.05	209	0.010	0.69	0.45
8	43	15	0.06	110	0.013	0.60	0.57
9	43	20	0.05	220	0.009	0.79	0.38
10	43	20	0.06	83	0.020	1.57	0.60
11	43	25	0.05	191	0.009	0.67	0.46
12	43	25	0.06	137	0.010	0.86	0.50
13	65	15	0.05	242	0.007	0.72	0.39
14	65	15	0.06	154	0.007	0.47	0.61
15	65	20	0.05	233	0.007	0.74	0.37
16	65	20	0.06	152	0.007	0.47	0.61
17	65	25	0.05	244	0.008	0.87	0.50
18	65	25	0.06	178	0.007	0.53	0.54

Table 7.4 Test results of the online execution of the proposed system for Model 2.

Test No	Init DS size	Min FS size	Std Threshold	Final DS size	Prediction MSE	Train Error Range	Pred Error Range
1	22	15	0.05	173	0.007	0.66	0.39
2	22	15	0.06	95	0.009	0.62	0.46
3	22	20	0.05	189	0.009	0.71	0.42
4	22	20	0.06	103	0.010	0.60	0.66
5	22	25	0.05	120	0.010	0.59	0.42
6	22	25	0.06	125	0.011	0.68	0.43
7	43	15	0.05	189	0.007	0.75	0.34
8	43	15	0.06	127	0.008	0.66	0.45
9	43	20	0.05	204	0.006	0.66	0.32
10	**43**	**20**	**0.06**	**140**	**0.007**	**0.75**	**0.36**
11	43	25	0.05	202	0.007	0.70	0.34
12	43	25	0.06	145	0.007	0.52	0.41
13	65	15	0.05	214	0.006	0.57	0.33
14	65	15	0.06	157	0.006	0.52	0.44
15	65	20	0.05	210	0.006	0.57	0.39
16	65	20	0.06	165	0.006	0.51	0.38
17	65	25	0.05	197	0.006	0.48	0.33
18	65	25	0.06	158	0.005	0.53	0.37

Note that one of the instances of Panel 1 (hole 42) was removed from the dataset due to the bad signal quality, thus Panel 1 had a total of 43 instances. Following the recommendations given at the end of Section 7.1, the best test was chosen as the one with a final DS in the range from 120 to 190 (considering both models), having the lowest prediction error range (i.e. the data of the last column in both tables). Most tests with lower ranges of the prediction error presented higher final dataset sizes (around 200). However, after inspection of the results, it was observed that this was due to the std threshold value being too low, thus the majority of the data was included in the training set leaving only a few instances to use for the depth predictions. Moreover, since the purpose of the learning procedure was to minimise the use of the inspection camera, higher priority was given to tests with lower final DS size, yet showing acceptable error levels.

Having decided the optimal values of the three free parameters for the proposed in-process learning procedure, the remaining part of this section will present and discuss the results obtained by the execution of the algorithm for the above two selected tests.

Figure 7.7 shows the instances included in the final dataset with respect to their index (hole number) for Model 1 (top graph) and Model 2 (bottom graph). The included instances (training data) are indicated by a blue vertical bar, whereas the others (testing data) are

drawn with a white colour. The corresponding panel indices are also illustrated in the figure by the vertical dashed lines. The final DS size was 129 for Model 1 and 140 for Model 2.

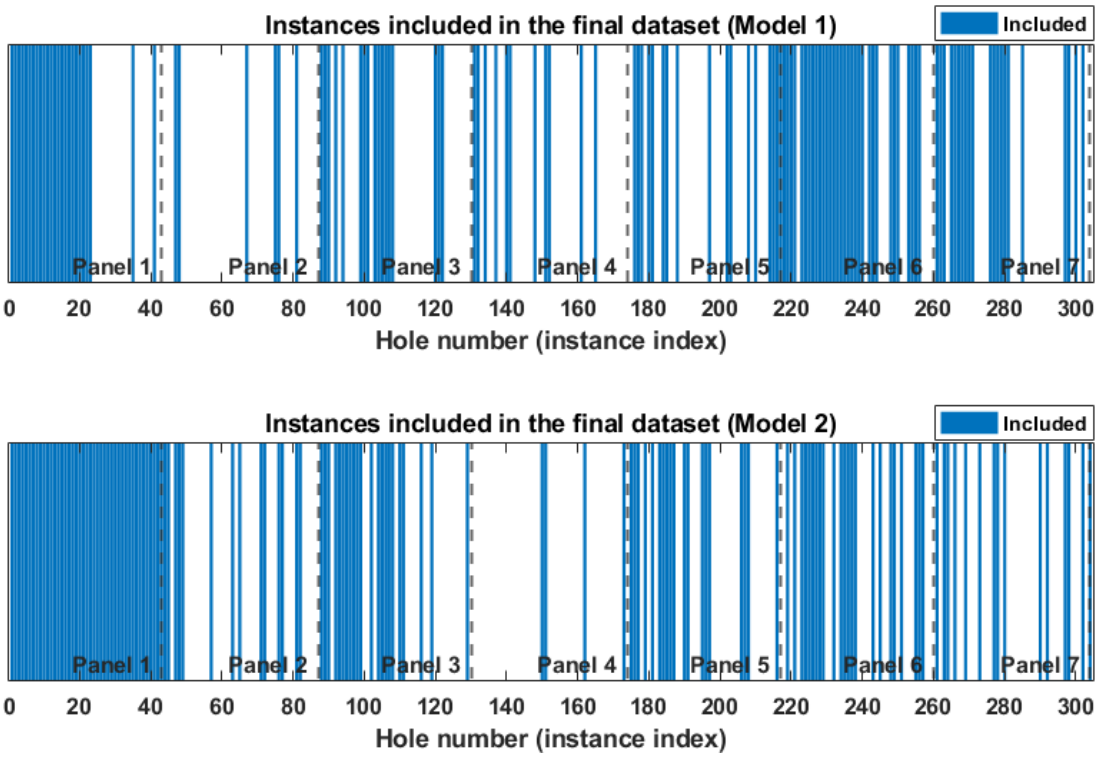

Figure 7.7 Instances included in the final dataset from the multi-step learning procedure: Model 1 (top) and Model 2 (bottom).

From the figure it can be observed that both models had included one or more instances from each of the considered panels. In terms of Model 1, the information gained from the initial dataset (half of Panel 1) was enough to predict most depths of Panel 2. The intervention of the camera was mainly required in Panel 3 and then occasionally for Panel 4 and 5. In contrast, most instances of Panel 6 were added to the final dataset, indicating that the system was probably collecting signals slightly different from those previously seen in the other panels. The camera measurements were also required for the beginning of Panel 7 until the model had acquired enough data to confidently predict the remaining instances. A similar behaviour was observed for Model 2, which, after using the entire Panel 1 as an initial dataset, was confident in most depth predictions of Panel 2. Frequent camera measurements during the first half of Panel 3 helped to increase the model confidence in the prediction of depth values of the remaining of Panel 3 and Panel 4. The start of the experimental Phase 2 (i.e. Panel 5) required the intervention of the camera. Note that this behaviour of the process was previously seen when discussing the assessment of the model's confidence in Section 7.1 (ref. to Figure 7.1). Panel 6 was also mostly included in the dataset, but not as much as

the case of Model 1. For the last panel (Panel 7) direct measurement were occasionally needed, indicating that the model had collected enough training data to predict depth values with a high level of confidence.

Further insights of the prediction errors and standard deviations (std values) obtained during the execution of the proposed learning procedure are presented in Figure 7.8. The errors are illustrated by the red lines and the blue vertical bars represent the std values. The uncertainty threshold (applied on the prediction std) is indicated by a horizontal black line and the vertical dashed lines are the corresponding panel indices. The top graph illustrates the results relative to Model 1 and those of Model 2 are shown in the bottom graph.

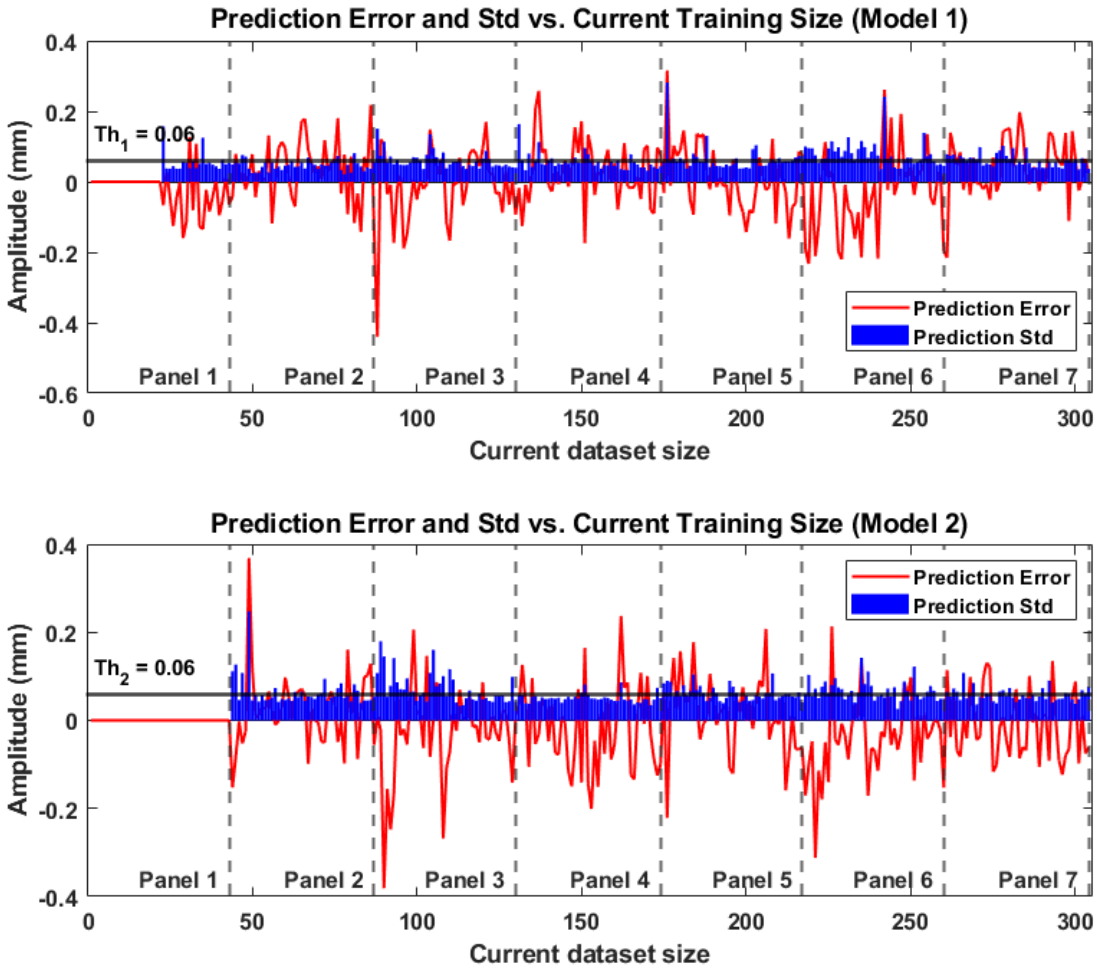

Figure 7.8 Prediction Error and Std values obtained from the multi-step learning procedure: Model 1 (top) and Model 2 (bottom).

From the examination of Figure 7.8, it can be observed that the range of the prediction errors of both models had considerably increased in comparison to the range of errors obtained in the offline (LOO-CV) analysis performed in Section 6.4. This is due to the fact that less training data was used when attempting to make the predictions online. The highest errors appeared

to be in correspondence of the beginning of new panels, especially for Panel 3, 5 and 6. Furthermore, the uncertainty threshold applied to the std values appeared to have identified the majority of instances with higher prediction errors, indicating high correlation between the std values and the magnitude of the errors.

It is important to note that the above prediction errors and std values refer to the output of the GPR model for each iteration of the process based on all the available information (dataset) that the model had at the time when the new input data was presented. This means that instances of Panel 2 would have predictions based only on the dataset composed of the data from Panel 1 (and any other instances eventually included by the algorithm up to that point), whereas the predictions for Panel 7 would rely on more information (larger dataset) being collected while machining the previous panels. Moreover, to compute the prediction errors, the direct measurement is required, therefore in a real-time scenario this measurement would not have been available because the monitoring system had decided to use the prediction. In the simulations performed here, however, since the aim of the tests was to assess the performance of the learning method, the error values are shown for all the data that the model provided a prediction, independently of the decision made on the inspection step.

The correlation between the obtained prediction errors and the corresponding std values is highlighted in Figure 7.9 for Model 1 (left plot) and Model 2 (right plot). The uncertainty threshold is indicated by the red dashed line.

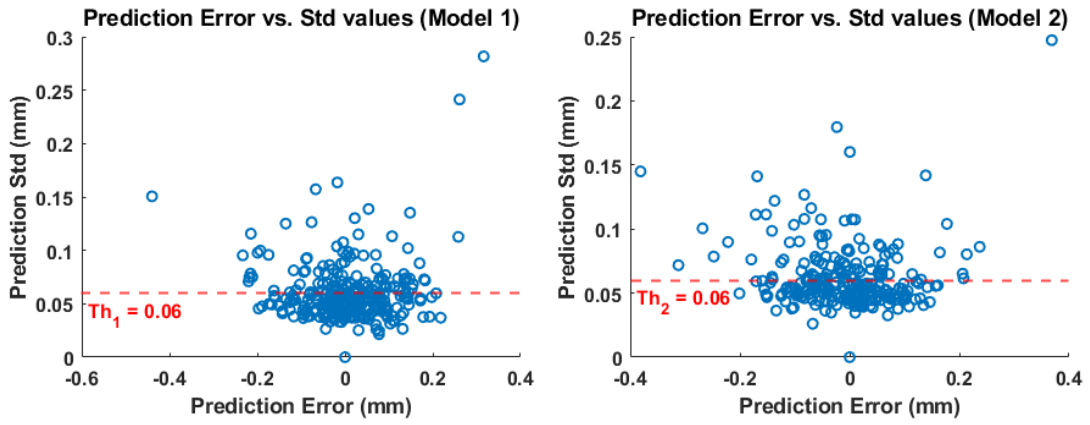

Figure 7.9 Correlation between the prediction Error and Std values: Model 1 (left) and Model 2 (right).

A closer view of Figure 7.9 reveals that the threshold had successfully identified the instances with higher prediction error values in both models. The method achieved better results for

the data obtained from Model 2 in terms of separation of the dataset. The majority of the data under the threshold value was located within the range [-0.2 0.2] mm, which can be considered acceptable for the online version of the IMS, given the size of the available dataset. In order to further improve the prediction performance of the models, more training data would be required, as the LOO-CV analysis have shown (ref. to Section 6.4). As previously mentioned, all the instances with prediction std over the threshold value were added to the current dataset to be used as training data for the subsequent iterations of the process. The instances under the threshold value were those actually 'tested' by the system since they were not considered for further training.

Finally, Figure 7.10 illustrates the prediction errors of the instances included in the dataset from the algorithm, denoted as train data (on the left side) and those with lower std values (i.e. showing higher model confidence), denoted as test data (on the right side). The results relative to Model 1 are shown in the top two graphs and those of Model 2 are illustrated in the bottom two graphs.

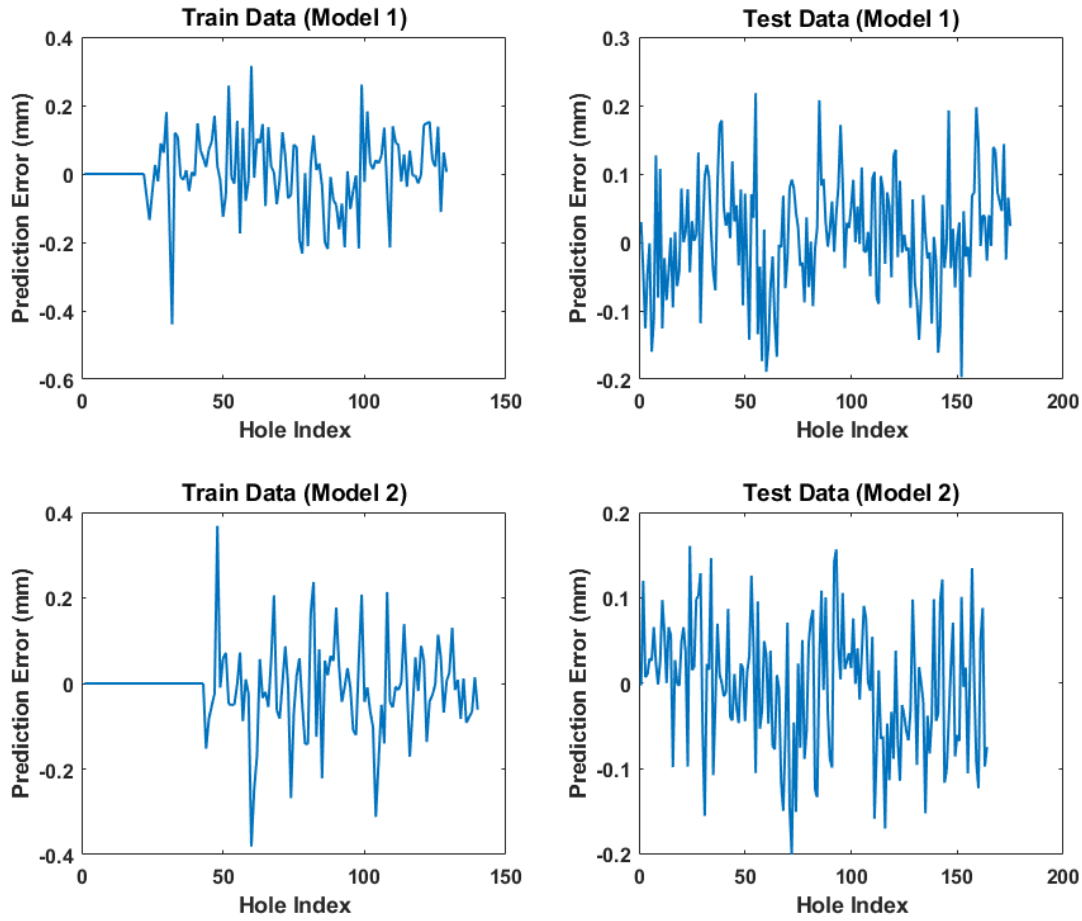

Figure 7.10 Prediction Errors divided as Train data (left side) and Test data (right side) for Model 1 (top graphs) and Model 2 (bottom graphs), respectively.

The results shown in Figure 7.10 emphasise the separation of the instances in training and testing data as it was decided by the proposed learning procedure. The range of the prediction errors had almost halved from the training data to the test data for both models. Moreover, most values of the test data were contained in the interval [-0.15 0.15] mm, confirming the intended use of the uncertainty threshold.

In summary, the results of the simulation tests presented in this section proved the ability of the proposed multi-step learning algorithm to correctly identify the instances where the predicted outputs would have had high enough prediction accuracy to skip the inspection step. In addition, the predictions of the instances included in the final dataset showed generally higher error ranges than the predictions of the instances that were not included, meaning that the monitoring system was right to ask for the camera intervention. Furthermore, the proposed uncertainty threshold showed good ability to detect the instances that would bring more information into the current process knowledge (dataset), improving therefore the prediction accuracies for the subsequent input data. This last characteristic of the system was observed in the examination of the final dataset shown in Figure 7.7, where new instances were required every time a new panel was introduced and at the beginning of Phase 2 of the experiment (Panels 5-7). The information gained by the newly added instances helped the process model to adapt to the slightly different sensor signals between the two phases of the experiment or small changes in the material properties between two subsequent panels.

7.5 Final Remarks

Before concluding this chapter, it is important to examine the proposed multi-step approach from the process perspective by analysing the experimental depth data obtained from the experiment. As previously noted, the depth target of the second cut was always corrected based on the direct camera measurement, not the prediction provided by the model. Therefore, following the diagram of the multi-step method shown in Figure 7.6 and the considerations given in Section 7.2 with regards to Eq. (7.7), one would expect to find final depth values close to the desired level, i.e. $D_2 \cong d_2$. Surprisingly, this was not the case for the experiment performed in this thesis.

The final depth errors were previously presented in Section 6.2.2 (ref. to Figure 6.3) in the context of the output values of the dataset used for training the process models, where it was also discussed the possible cause for the observed trend in the data. However, it is useful

to review them in further detail here, after having presented the multi-step method. Figure 7.11 illustrates the camera depth measurements (top graph) obtained at the end of the first cut (blue) and at the end of the second cut (red). The graph at the bottom of the figure highlights the final depth errors. The two desired depth targets (d_1 and d_2) in the top graph and the process tolerance level (Tol) in the bottom graph are indicated by the horizontal dashed lines. The panel indices are drawn with vertical dotted lines.

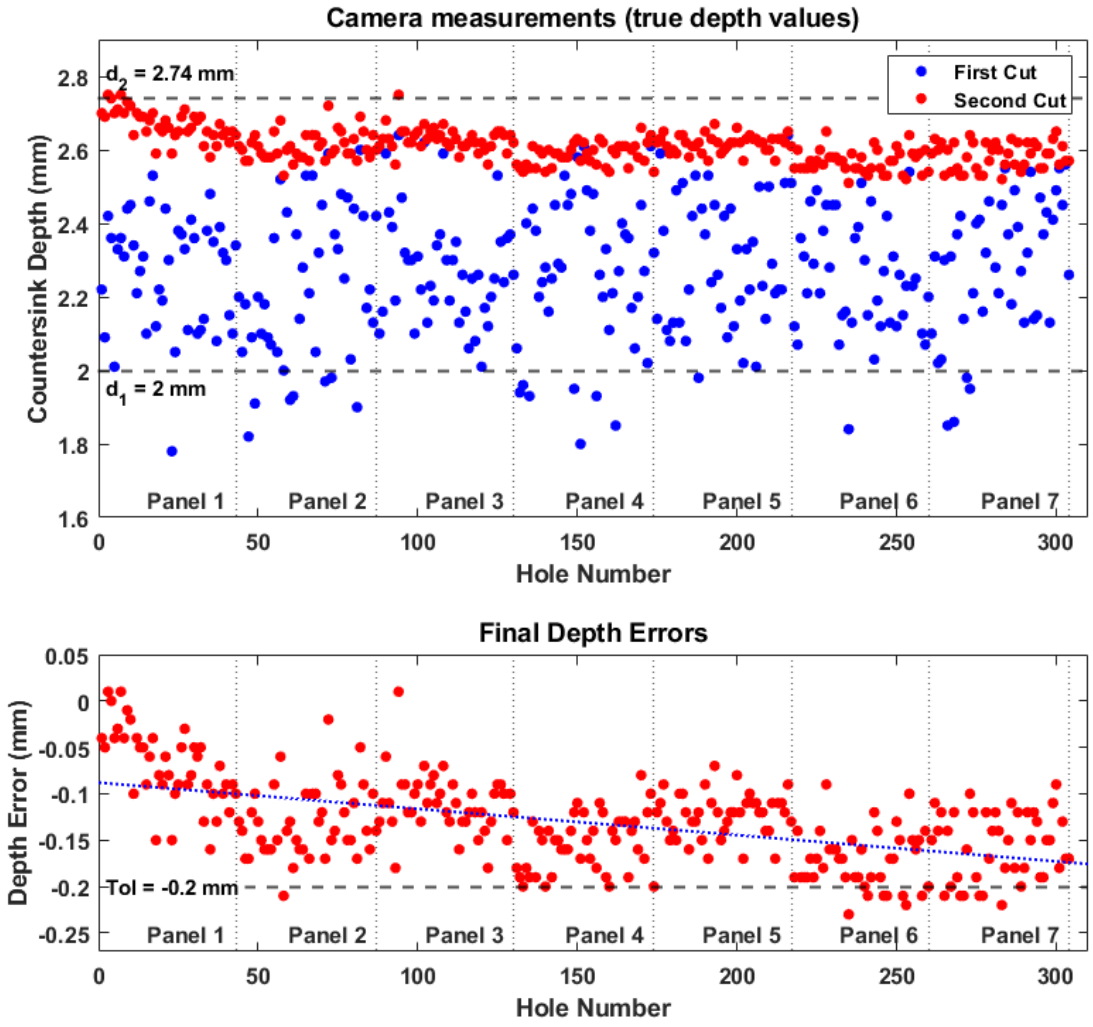

Figure 7.11 Countersink depth measurements (top) and final depth errors (bottom).

From the examination of the figure, it can be observed that the depth values of the first cut (blue dots) with target T_1 defined in Eq. (7.2) presented a systematic error (positive bias) that resulted in holes with depths over the desired target. As discussed in Section 6.2.1, this positive bias was associated to a calibration error, which was corrected by the multi-step method. Moreover, the large depth variability of the first cut, programmatically introduced by the Gaussian perturbation, was significantly reduced in the second cut.

However, as previously noted, the final depth data (red dots) showed a decreasing trend which is illustrated by the blue dotted line in the figure. Almost all the final depth values were under the desired target ($d_2 = 2.74\ mm$) and a few of them (mainly from Panel 6 and 7) were under the process tolerance ($Tol = -0.2\ mm$). Possible causes of the decreasing trend observed in the final depth data can be associated with the following:

1. Cutting tool wear: this was already discussed in Section 6.2.2.
2. Inspection camera errors that would directly affect the target of the second cuts since the correction was based on this direct measurement. For example, if the camera measures a slightly bigger hole (more depth) than the real size after the first cut, then a larger compensation value will be computed and T_2 will be adjusted accordingly (ref. to Eq. (7.6)). This will result in depths under the desired final target (d_2).
3. The assumption made by the multi-step method of $N_{p_1} \cong N_{p_2}$ was not valid (ref. to Section 7.2 and Section 4.2). This case explores the possibility that the errors introduced by the countersinking process in the first cut are not comparable with those of the second cut. In particular, with reference to Eq. (7.7), in order to obtain under-target depths the process errors of the first cut need to be greater than those of the second cut, i.e. $N_{p_1} > N_{p_2}$.

From the above three possible causes, the cutting tool wear is the one that explains better the observed decreasing trend of the data, since the tool wear gradually increases as the number of holes (cuts) increases. The other two may explain the random fluctuations of the final depth values because their error contribution is of a random nature, unless a systematic calibration error is observed, as it was the case of the first cut's bias.

While the tool wear can be sensed from the acquired signals and therefore predicted by the monitoring system, the camera errors are not detectable from the process models. Given the supervised learning approach, the camera measurement is considered to be the true depth value that the model is trained to predict. If this true value is wrong, then the best the model can do is to predict that exact (wrong) value.

The errors coming from the above third point can be reduced by minimising the differences between the two steps of the process. For the particular case, this would be possible by setting the semi-finish target depth as half of the finish depth, meaning that the process would remove comparable amount of material (on average) from both cuts.

In conclusion, a possible way to correct the observed behaviour of the process would be the inclusion of an extra compensation term in the target adjustment block of the diagram of Figure 7.6, which takes into account the final depth data of previous iterations of the process. This idea is referred to a further development of the work conducted by this thesis, as discussed in Chapter 9.

8 CONCLUSIONS

The main aim of this thesis was to design and test an intelligent monitoring system suitable for robotic machining processes that is capable to accurately and reliably predict key process variables which directly influence the quality of the finished product. By considering the robotic system as an additional source of error (disturbance) that affects the quality of the machined part, this work has successfully integrated the concepts of process monitoring of conventional operations into a robotic machining context.

A multi-step learning approach was proposed, which allowed corrective actions to be included into the overall process cycle by performing the cutting process in two different levels of finishing (semi-finish and finish). The desired target of the finish level was adjusted from the observed error of the previous cut in semi-finish level. The method was tested in an experiment by using an inspection camera (direct measurement) for the collection of the output values (depth errors) at the end of each cutting step. The range of errors observed at the end of the first cut (semi-finish level) was significantly greater than that observed in the second cut (finish level), meaning that the target correction method had successfully compensated for the errors of the previous step.

The monitoring system designed by this work was tested using the dataset extracted from the experiment in two different scenarios: (i) an offline analysis using all the available experimental data and (ii) an online analysis simulating the execution of the process with the data becoming available as more iterations were completed. The offline analysis proved the ability of the process models to provide accurate predictions of the output variable. The online analysis demonstrated the reliability of the system by successfully identifying the process iterations when the camera inspection step was necessary due to the high uncertainty of the predicted output value.

The first objective of this work was to select suitable hardware and software for the in-process extraction of the machining process information. The data acquisition systems was designed to use non-intrusive and easy to install sensors, such as accelerometers, AE sensors and power transducers. A central data acquisition unit was applied to ensure synchronisation of all the acquired signals. Several filtering techniques as well as signal domain transformations (time, frequency and joint time-frequency domains) were used for the extraction of a large number of signal features that describe the machining process. The

signal segmentation, processing and feature extraction steps were performed automatically, as soon as the data was available after the cutting process.

The second objective was to identify an appropriate method for selection of the most useful signal features for describing the machining process. In the regression tests conducted in this work, a GPR model with ARD-SE kernel was applied as a way to select the relevant signal features. The method consisted of an embedded approach that integrated the feature selection step into the training phase of the GPR models. The relevant features were selected based on the weights that were assigned by the model during training. The method was tested with the dataset extracted from the experiment considering the two phases of the experiment separately and jointly. In all tests the models that used the reduced subset of features (as selected by the ARD method) performed better than those that used all the features. This method, although known in the machine learning literature, has not been used before in the machining monitoring context. This thesis has shown the use of GP with ARD-SE kernel as an effective technique for the selection of the most useful signal features that describe the machining process.

The third objective of this work was to design a system that can be trained during normal operation of the machining process, without any interruptions. The multi-step learning approach proposed by this thesis allowed process models to be training online. In particular, a Gaussian perturbation was programmatically added to the process input target in order to provide the required data variability for training the models in different conditions. Moreover, by correcting the target of the second step of the process from the error observed in the previous step, this method ensured that the second step was still able to reach the desired finish level. Therefore, an experiment specifically designed for obtaining the training data was no longer required.

The fourth objective of this thesis was to provide a means of assessing the reliability of the process models. This has been addressed by the use of a probabilistic learning approach, which naturally deals with uncertain inputs and outputs and provides predictions with an estimated (mean) value and the corresponding variance. This prediction variance expresses the confidence of the model in the predicted output and it is inferred from the new testing input and the currently available training data, not the observations (outputs). Therefore, this work used the prediction variance to assess the model reliability. A decision on the inspection step was made based on this prediction variance: a high value of variance, indicated low confidence of the model in the predicted output, thus an inspection was

required; whereas a low value of the prediction variance represented a high confidence of the model in the predicted output, thus the inspection step could be avoided. The method was tested with the dataset extracted from the experiment by simulating an online execution of the process and the results proved the ability of the method to successfully identify the cases when the inspection step could have been avoided.

9 FURTHER WORK

This thesis presented a novel method to improve the performance of a robotic machining process by the use of in-process monitoring of key process variables that directly affect the quality of the final product. In the current literature, there were not many research works that reported the use of intelligent monitoring systems into a robotic machining context, therefore, more research is required to increase the number of applications that combine these two research areas. In order to further progress the work carried out by this thesis, a few key future developments have been identified and presented below.

Firstly, in this work the multi-step learning method was tested through simulation tests performed on the dataset extracted from the experiment. A natural progression of this work would therefore consist of conducting an experiment where the depth predictions of the first cut are used to correct the target of the second cut, depending on the decision made by the monitoring system. This would allow to see the overall effect of the compensation method in the final depth values. Testing the flexibility of the proposed approach in a different machining process would also be an interesting future development.

The regression tests conducted in this work used an embedded approach to select the relevant features based on the weights that were assigned during training the process models. In particular, a GPR with ARD-SE kernel was applied, using a threshold on the feature weight for the selection process. The value of such threshold was inferred based on previous observations and/or knowledge of the process. Future work in this direction might explore the use filter methods that are generally faster and do not rely on the particular learning algorithm and/or consider the redundancy of the selected features.

In terms of the assessment of the model's confidence, this work applied an uncertainty threshold on the standard deviation of the prediction provided by the GPR models. Further research could usefully investigate the use of Mahalanobis distance [127] to measure how far (in terms of standard deviations) a new input instance (feature vector) is from the multi-variate Gaussian distribution defined by the training data. This measure could then be used as an alternative (and faster) method to estimate the model's confidence without involving any training of the model. This would result in faster execution time of the algorithm, especially for large datasets.

Considering the proposed multi-step method from the process perspective, it was briefly suggested at the end of Chapter 7 the idea of adding a further compensation term to address the observed trend in the depth data. The addition of a term that considers the process output of previous iterations would correct for errors that depend on the number of process iterations such as the tool wear. It might be useful investigate the integration of this term in the overall multi-step scheme presented in Chapter 4.

Finally, a generalisation of the proposed approach is suggested for further research work. In a production environment, a manufacturing process is usually performed in several stages involving different processes applied on the same workpiece. Typically, an individual stage consists of the actual physical (material removal) process where the part is being worked, followed by an inspection step to ensure that the process has met the quality requirements, before progressing to the next stage. An extension of the method proposed by this thesis into a multi-stage process, where each stage refers to an individual machining process would be an interesting area to investigate. However, in this case the assumption of comparable errors between two subsequent stages of the overall process is not valid. Therefore a way to infer how the errors are propagated from one stage to the other is necessary.

10 REFERENCES

[1] Y. Chen and F. Dong, "Robot machining: recent development and future research issues," *The International Journal of Advanced Manufacturing Technology,* vol. 66, no. 9-12, pp. 1489-1497, 2013.

[2] J. Pandremenos, C. Doukas, P. Stavropoulos, and G. Chryssolouris, "Machining with robots: a critical review," *Proceedings of DET2011,* 2011.

[3] M. Slamani and J. Chatelain, "Issues and Challenges in Robotic Trimming of CFRP," (in English), *Icimco 2015 Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, Vol. 2,* Proceedings Paper pp. 400-405, 2015.

[4] C.-P. Day, "Robotics in Industry-Their Role in Intelligent Manufacturing," *Engineering,* vol. 4, no. 4, pp. 440-445, 2018/08/01 2018.

[5] C. Lehmann, M. Halbauer, J. van der Zwaag, U. Schneider, and U. Berger, "Offline path compensation to improve accuracy of industrial robots for machining applications," 2013.

[6] U. Schneider *et al.*, "Improving robotic machining accuracy through experimental error investigation and modular compensation," (in English), *International Journal of Advanced Manufacturing Technology,* Article vol. 85, no. 1-4, pp. 3-15, JUL 2016 2016.

[7] O. Sornmo, B. Olofsson, U. Schneider, A. Robertsson, and R. Johansson, "Increasing the Milling Accuracy for Industrial Robots Using a Piezo-Actuated High-Dynamic Micro Manipulator," (in English), *Ieee/asme International Conference on Advanced Intelligent Mechatronics (Aim),* Proceedings Paper pp. 104-110, 2012 2012.

[8] Z. Pan and H. Zhang, "Improving robotic machining accuracy by real-time compensation," presented at the ICCAS-SICE, 2009.

[9] A. Bottero, G. Gerio, V. Perna, and A. Gagliano, "Adaptive control techniques and feed forward compensation of periodic disturbances in industrial manipulators," presented at the IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), 2014, 2014.

[10] H. Zhang *et al.*, "Machining with flexible manipulator: Toward improving robotic machining performance," (in English), *Ieee/asme International Conference on Advanced Intelligent Mechatronics, Vols 1 and 2,* Proceedings Paper pp. 1127-1132, 2005 2005.

[11] E. Abele, J. Bauer, M. Pischan, M. Friedmann, T. Hemker, and O. Stryk, "Prediction of the tool displacement for robot milling applications using coupled models of an industrial robot and removal simulation," presented at the CIRP 2nd International Conference Process Machine Interactions, Vancouver, Canada, 2010.

[12] H.-k. Huang and G. C. Lin, "Rapid and flexible prototyping through a dual-robot workcell," *Robotics and Computer-Integrated Manufacturing,* vol. 19, no. 3, pp. 263-272, 2003.

[13] W. Owen, E. Croft, and B. Benhabib, "Real-time trajectory resolution for a two-manipulator machining system," (in English), *Journal of Robotic Systems,* Article vol. 22, pp. S51-S63, 2006 2006.

[14] S. Liang, R. Hecker, and R. Landers, "Machining process monitoring and control: The state-of-the-art," (in English), *Journal of Manufacturing Science and Engineering-Transactions of the Asme,* Review vol. 126, no. 2, pp. 297-310, MAY 2004 2004.

[15] J. Abellan-Nebot and F. Subiron, "A review of machining monitoring systems based on artificial intelligence process models," (in English), *International Journal of*

Advanced Manufacturing Technology, Review vol. 47, no. 1-4, pp. 237-257, MAR 2010 2010.

[16] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld, "Advanced monitoring of machining operations," (in English), *Cirp Annals-Manufacturing Technology,* Article|Proceedings Paper vol. 59, no. 2, pp. 717-739, 2010 2010.

[17] K. Worden, W. Staszewski, and J. Hensman, "Natural computing for mechanical systems research: A tutorial overview," (in English), *Mechanical Systems and Signal Processing,* Review vol. 25, no. 1, pp. 4-111, JAN 2011 2011.

[18] P. Stavropoulos, D. Chantzis, C. Doukas, A. Papacharalampopoulos, G. Chryssolouris, and L. Settineri, "Monitoring and control of manufacturing processes: A review," (in English), *14th Cirp Conference on Modeling of Machining Operations (Cirp Cmmo),* Proceedings Paper vol. 8, pp. 421-425, 2013 2013.

[19] C. H. Lauro, L. C. Brandão, D. Baldo, R. A. Reis, and J. P. Davim, "Monitoring and processing signal applied in machining processes – A review," *Measurement,* vol. 58, pp. 73-86, 2014.

[20] I. Iglesias, M. A. Sebastián, and J. E. Ares, "Overview of the State of Robotic Machining: Current Situation and Future Potential," *Procedia Engineering,* vol. 132, pp. 911-917, 2015/01/01 2015.

[21] J. DePree and C. Gesswein, "Robotic Machining White Paper Project," Halcyon Development, Robotic Industries Association, 2008, Available: https://www.robotics.org/robotics/halcyon-development-ria, Accessed on: Sep 2018.

[22] Y. Song and Y. H. Chen, "Feature-based robot machining for rapid prototyping," (in English), *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture,* vol. 213, no. 5, pp. 451-459, 1999.

[23] W. Owen *et al.*, "Minimally compliant trajectory resolution for robotic machining," (in English), *Proceedings of the 11th International Conference on Advanced Robotics 2003, Vol 1-3,* Proceedings Paper pp. 702-707, 2003 2003.

[24] H. Huang, Z. M. Gong, X. Q. Chen, and L. Zhou, "Robotic grinding and polishing for turbine-vane overhaul," (in English), *Journal of Materials Processing Technology,* vol. 127, no. 2, pp. 140-145, Sep 30 2002.

[25] X. Ren, B. Kuhlenkotter, and H. Muller, "Simulation and verification of belt grinding with industrial robots," (in English), *International Journal of Machine Tools & Manufacture,* vol. 46, no. 7-8, pp. 708-716, Jun 2006.

[26] B. Denkena and T. Lepper, "Enabling an Industrial Robot for Metal Cutting Operations," *Procedia CIRP,* vol. 35, pp. 79-84, 2015/01/01 2015.

[27] J. D. Barnfather, M. J. Goodfellow, and T. Abram, "Development and testing of an error compensation algorithm for photogrammetry assisted robotic machining," (in English), *Measurement,* vol. 94, pp. 561-577, Dec 2016.

[28] Y. Bu, W. H. Liao, W. Tian, J. Zhang, and L. Zhang, "Stiffness analysis and optimization in robotic drilling application," (in English), *Precision Engineering-Journal of the International Societies for Precision Engineering and Nanotechnology,* vol. 49, pp. 388-400, Jul 2017.

[29] L. J. Cen and S. N. Melkote, "Effect of Robot Dynamics on the Machining Forces in Robotic Milling," (in English), *45th Sme North American Manufacturing Research Conference (Namrc 45),* vol. 10, pp. 486-496, 2017.

[30] R. T. Coelho, H. H. T. Rodella, V. F. Martins, and R. Barba, "An Investigation into the Use of Industrial Robots for Machining Soft and Low Density Materials with HSM Technique," (in English), *Journal of the Brazilian Society of Mechanical Sciences and Engineering,* vol. 33, no. 3, pp. 343-350, Jul-Sep 2011.

[31] C. Dumas, S. Caro, M. Cherif, S. Garnier, and B. Furet, "Joint stiffness identification of industrial serial robots," (in English), *Robotica,* vol. 30, pp. 649-659, Jul 2012.

[32] L. F. F. Furtado, E. Villani, L. G. Trabasso, and R. Suterio, "A method to improve the use of 6-dof robots as machine tools," *International Journal of Advanced Manufacturing Technology,* vol. 92, no. 5-8, pp. 2487-2502, Sep 2017.

[33] A. Klimchik, A. Ambiehl, S. Garnier, B. Furet, and A. Pashkevich, "Efficiency evaluation of robots in machining applications using industrial performance measure," (in English), *Robotics and Computer-Integrated Manufacturing,* vol. 48, pp. 12-29, Dec 2017.

[34] Z. X. Pan and H. Zhang, "Robotic machining from programming to process control: a complete solution by force control," (in English), *Industrial Robot-an International Journal,* vol. 35, no. 5, pp. 400-409, 2008.

[35] E. Abele, M. Weigold, and S. Rothenbucher, "Modeling and identification of an industrial robot for machining applications," (in English), *Cirp Annals-Manufacturing Technology,* vol. 56, no. 1, pp. 387-390, 2007.

[36] Z. X. Pan, H. Zhang, Z. Q. Zhu, and J. J. Wang, "Chatter analysis of robotic machining process," (in English), *Journal of Materials Processing Technology,* vol. 173, no. 3, pp. 301-309, Apr 20 2006.

[37] J. D. Barnfather, M. J. Goodfellow, and T. Abram, "A performance evaluation methodology for robotic machine tools used in large volume manufacturing," *Robotics and Computer-Integrated Manufacturing,* vol. 37, pp. 49-56, 2016/02/01/ 2016.

[38] K. Yang, W. Y. Yang, G. D. Cheng, and B. R. Lu, "A new methodology for joint stiffness identification of heavy duty industrial robots with the counterbalancing system," (in English), *Robotics and Computer-Integrated Manufacturing,* vol. 53, pp. 58-71, Oct 2018.

[39] Y. J. Guo, H. Y. Dong, and Y. L. Ke, "Stiffness-oriented posture optimization in robotic machining applications," (in English), *Robotics and Computer-Integrated Manufacturing,* vol. 35, pp. 69-76, Oct 2015.

[40] Y. Lin, H. Zhao, and H. Ding, "Posture optimization methodology of 6R industrial robots for machining using performance evaluation indexes," (in English), *Robotics and Computer-Integrated Manufacturing,* vol. 48, pp. 59-72, Dec 2017.

[41] C. Reinl, M. Friedmann, J. Bauer, M. Pischan, E. Abele, and O. von Stryk, "Model-based Off-line Compensation of Path Deviation for Industrial Robots in Milling Applications," (in English), *2011 Ieee/Asme International Conference on Advanced Intelligent Mechatronics (Aim),* pp. 367-372, 2011.

[42] "COMET EU/FP7 Project: Plug-and-produce COmponents and METhods for adaptive control of industrial robots enabling cost effective, high precision manufacturing in factories of the future," 2010-2013, Available: https://comet-project.eu/, Accessed on: Oct 2018.

[43] U. Schneider, M. Drust, A. Puzik, and A. Verl, "Compensation of Errors in Robot Machining With a Parallel 3D-Piezo Compensation Mechanism," (in English), *Forty Sixth Cirp Conference on Manufacturing Systems 2013,* vol. 7, pp. 305-310, 2013.

[44] P. J. Yuan, D. D. Chen, T. M. Wang, S. Q. Cao, Y. Cai, and L. Xue, "A compensation method based on extreme learning machine to enhance absolute position accuracy for aviation drilling robot," (in English), *Advances in Mechanical Engineering,* vol. 10, no. 3, Mar 13 2018.

[45] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," (in English), *2004 Ieee International Joint Conference on Neural Networks, Vols 1-4, Proceedings,* pp. 985-990, 2004.

[46] P. Marchal, O. Sornmo, B. Olofsson, A. Robertsson, J. Ortega, and R. Johansson, "Iterative Learning Control for Machining with Industrial Robots," (in English), *Ifac Papersonline,* Proceedings Paper vol. 47, no. 3, pp. 9327-9333, 2014 2014.

[47] R. Teti, "A Review of Tool Condition Monitoring Literature Database," *CIRP Annals,* vol. 44, no. 2, pp. 659-666, 1995.

[48] S. Binsaeid, S. Asfour, S. Cho, and A. Onar, "Machine ensemble approach for simultaneous detection of transient and gradual abnormalities in end milling using multisensor fusion," (in English), *Journal of Materials Processing Technology,* Article vol. 209, no. 10, pp. 4728-4738, JUN 1 2009 2009.

[49] S. Cho, S. Binsaeid, and S. Asfour, "Design of multisensor fusion-based tool condition monitoring system in end milling," (in English), *International Journal of Advanced Manufacturing Technology,* Article vol. 46, no. 5-8, pp. 681-694, JAN 2010 2010.

[50] N. Ghosh *et al.*, "Estimation of tool wear during CNC milling using neural network-based sensor fusion," *Mechanical Systems and Signal Processing,* vol. 21, no. 1, pp. 466-479, 2007.

[51] T. Segreto, A. Simeone, and R. Teti, "Multiple sensor monitoring in nickel alloy turning for tool wear assessment via sensor fusion," *Procedia CIRP,* vol. 12, pp. 85-90, 2013.

[52] P. Stavropoulos, A. Papacharalampopoulos, E. Vasiliadis, and G. Chryssolouris, "Tool wear predictability estimation in milling based on multi-sensorial data," *The International Journal of Advanced Manufacturing Technology,* vol. 82, no. 1-4, pp. 509-521, 2016.

[53] O. Çolak, C. Kurbanoğlu, and M. C. Kayacan, "Milling surface roughness prediction using evolutionary programming methods," *Materials & design,* vol. 28, no. 2, pp. 657-666, 2007.

[54] D. R. Salgado, F. Alonso, I. Cambero, and A. Marcelo, "In-process surface roughness prediction system using cutting vibrations in turning," *The International Journal of Advanced Manufacturing Technology,* vol. 43, no. 1-2, pp. 40-51, 2009.

[55] T. Segreto, S. Karam, and R. Teti, "Signal processing and pattern recognition for surface roughness assessment in multiple sensor monitoring of robot-assisted polishing," (in English), *International Journal of Advanced Manufacturing Technology,* vol. 90, no. 1-4, pp. 1023-1033, Apr 2017.

[56] E. Garcia-Plaza *et al.*, "Surface Finish Monitoring in Taper Turning CNC Using Artificial Neural Network and Multiple Regression Methods," (in English), *Manufacturing Engineering Society International Conference, (Mesic 2013),* Proceedings Paper vol. 63, pp. 599-607, 2013 2013.

[57] I. Marinescu and D. Axinte, "A time-frequency acoustic emission-based monitoring technique to identify workpiece surface malfunctions in milling with multiple teeth cutting simultaneously," (in English), *International Journal of Machine Tools & Manufacture,* Article vol. 49, no. 1, pp. 53-65, JAN 2009 2009.

[58] I. Marinescu and D. A. Axinte, "An automated monitoring solution for avoiding an increased number of surface anomalies during milling of aerospace alloys," *International Journal of Machine Tools and Manufacture,* vol. 51, no. 4, pp. 349-357, 2011.

[59] I. Oleaga, C. Pardo, J. J. Zulaika, and A. Bustillo, "A machine-learning based solution for chatter prediction in heavy-duty milling machines," *Measurement,* vol. 128, pp. 34-44, 2018/11/01/ 2018.

[60] M. Lamraoui, M. Thomas, and M. El Badaoui, "Cyclostationarity approach for monitoring chatter and tool wear in high speed milling," (in English), *Mechanical Systems and Signal Processing,* Article vol. 44, no. 1-2, pp. 177-198, FEB 20 2014 2014.

[61] C. Peng, L. Wang, and T. Liao, "A new method for the prediction of chatter stability lobes based on dynamic cutting force simulation model and support vector machine," (in English), *Journal of Sound and Vibration,* Article vol. 354, pp. 118-131, OCT 13 2015 2015.

[62] M. S. H. Bhuiyan, I. A. Choudhury, and M. Dahari, "Monitoring the tool wear, surface roughness and chip formation occurrences using multiple sensors in turning," *Journal of Manufacturing Systems,* vol. 33, no. 4, pp. 476-487, 2014.

[63] R. Teti, I. S. Jawahir, K. Jemielniak, T. Segreto, S. Chen, and J. Kossakowska, "Chip form monitoring through advanced processing of cutting force sensor signals," (in English), *Cirp Annals-Manufacturing Technology,* vol. 55, no. 1, pp. 75-80, 2006.

[64] P. V. Er and K. K. Tan, "Machine vibration analysis based on experimental modal analysis with radial basis functions," *Measurement,* vol. 128, pp. 45-54, Nov 2018.

[65] J. Li *et al.*, "Feature selection: A data perspective," *arXiv preprint arXiv:1601.07996,* 2016.

[66] D. Shi and N. Gindy, "Tool wear predictive model based on least squares support vector machines," (in English), *Mechanical Systems and Signal Processing,* Article vol. 21, no. 4, pp. 1799-1814, MAY 2007 2007.

[67] G. Byrne and G. O'Donnell, "An integrated force sensor solution for process monitoring of drilling operations," (in English), *Cirp Annals-Manufacturing Technology,* Article vol. 56, no. 1, pp. 89-92, 2007 2007.

[68] A. Azmi, "Monitoring of tool wear using measured machining forces and neuro-fuzzy modelling approaches during machining of GFRP composites," (in English), *Advances in Engineering Software,* Article vol. 82, pp. 53-64, APR 2015 2015.

[69] I. Marinescu and D. Axinte, "A critical analysis of effectiveness of acoustic emission signals to detect tool and workpiece malfunctions in milling operations," (in English), *International Journal of Machine Tools & Manufacture,* Article vol. 48, no. 10, pp. 1148-1160, AUG 2008 2008.

[70] X. Li, G. Ouyang, and Z. Liang, "Complexity measure of motor current signals for tool flute breakage detection in end milling," (in English), *International Journal of Machine Tools & Manufacture,* Article vol. 48, no. 3-4, pp. 371-379, MAR 2008 2008.

[71] A. Bassiuny and X. Li, "Flute breakage detection during end milling using Hilbert-Huang transform and smoothed nonlinear energy operator," (in English), *International Journal of Machine Tools & Manufacture,* Article vol. 47, no. 6, pp. 1011-1020, MAY 2007 2007.

[72] D. Shi, D. Axinte, and N. Gindy, "Development of an online machining process monitoring system: a case study of the broaching process," (in English), *International Journal of Advanced Manufacturing Technology,* Article vol. 34, no. 1-2, pp. 34-46, AUG 2007 2007.

[73] A. Al-Habaibeh and N. Gindy, "A new approach for systematic design of condition monitoring systems for milling processes," *Journal of Materials Processing Technology,* vol. 107, no. 1, pp. 243-251, 2000.

[74] D. Axinte, "An experimental analysis of damped coupled vibrations in broaching," *International Journal of Machine Tools and Manufacture,* vol. 47, no. 14, pp. 2182-2188, 2007.

[75] E. D. Kirby, Z. Zhang, and J. C. Chen, "Development of an accelerometer-based surface roughness prediction system in turning operations using multiple regression techniques," *Journal of Industrial Technology,* vol. 20, no. 4, pp. 1-8, 2004.

[76] Q. Xiang, Z.-J. Lu, B.-Z. Li, and J.-g. Yang, "A process monitoring system based on multi-sensor data fusion: An experiment study," presented at the Uncertainty Reasoning and Knowledge Engineering (URKE), 2012 2nd International Conference on, 2012.

[77] X. Li, "A brief review: acoustic emission method for tool wear monitoring during turning," (in English), *International Journal of Machine Tools & Manufacture,* Review vol. 42, no. 2, pp. 157-165, JAN 2002 2002.

[78] M. A. Davies, T. Ueda, R. M'Saoubi, B. Mullany, and A. L. Cooke, "On The Measurement of Temperature in Material Removal Processes," *CIRP Annals - Manufacturing Technology,* vol. 56, no. 2, pp. 581-604, 2007/01/01 2007.

[79] S. Kurada and C. Bradley, "A review of machine vision sensors for tool condition monitoring," *Computers in Industry,* vol. 34, no. 1, pp. 55-72, 1997/10/01 1997.

[80] P. W. Prickett, R. A. Siddiqui, and R. I. Grosvenor, "The development of an end-milling process depth of cut monitoring system," *The International Journal of Advanced Manufacturing Technology,* vol. 52, no. 1-4, pp. 89-100, 2011.

[81] F. Castano, R. del Toro, R. Haber, and G. Beruvides, "Conductance sensing for monitoring micromechanical machining of conductive materials," (in English), *Sensors and Actuators a-Physical,* Article vol. 232, pp. 163-171, AUG 1 2015 2015.

[82] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE,* vol. 37, no. 1, pp. 10-21, 1949.

[83] D. Abboud, J. Antoni, S. Sieg-Zieba, and M. Eltabach, "Envelope analysis of rotating machine vibrations in variable speed conditions: A comprehensive treatment," *Mechanical Systems and Signal Processing,* vol. 84, pp. 200-226, 2017/02/01/ 2017.

[84] R. B. Randall, J. Antoni, and S. Chobsaard, "The relationship between spectral correlation and envelope analysis in the diagnostics of bearing faults and other cyclostationary machine signals," *Mechanical systems and signal processing,* vol. 15, no. 5, pp. 945-962, 2001.

[85] B.-S. Yang and K. J. Kim, "Application of Dempster–Shafer theory in fault diagnosis of induction motors using vibration and current signals," *Mechanical Systems and Signal Processing,* vol. 20, no. 2, pp. 403-420, 2006.

[86] D. A. Axinte, "Approach into the use of probabilistic neural networks for automated classification of tool malfunctions in broaching," *International Journal of Machine Tools and Manufacture,* vol. 46, no. 12, pp. 1445-1448, 2006.

[87] S. Dey and J. Stori, "A Bayesian network approach to root cause diagnosis of process variations," (in English), *International Journal of Machine Tools & Manufacture,* Article vol. 45, no. 1, pp. 75-91, JAN 2005 2005.

[88] D. Gabor, "Theory of communication. Part 1: The analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering,* vol. 93, no. 26, pp. 429-441, 1946.

[89] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 11, no. 7, pp. 674-693, 1989.

[90] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE transactions on information theory,* vol. 36, no. 5, pp. 961-1005, 1990.

[91] R. Polikar, "The wavelet tutorial," 1996.

[92] M. V. Wickerhauser and R. R. Coifman, "Entropy based methods for best basis selection," *IEEE Trans. on Inf. Theory,* vol. 38, no. 2, pp. 719-746, 1992.

[93] K. Zhu, Y. S. Wong, and G. S. Hong, "Wavelet analysis of sensor signals for tool condition monitoring: A review and some new results," *International Journal of Machine Tools and Manufacture,* vol. 49, no. 7–8, pp. 537-553, 2009.

[94] Y. Wu and R. Du, "Feature extraction and assessment using wavelet packets for monitoring of machining processes," (in English), *Mechanical Systems and Signal Processing,* Article vol. 10, no. 1, pp. 29-53, JAN 1996 1996.

[95] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.,* vol. 3, pp. 1157-1182, 2003.

[96] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," (in English), *Computers & Electrical Engineering,* Article vol. 40, no. 1, pp. 16-28, JAN 2014 2014.

[97] B. Denkena, D. Dahlmann, J. Damm, and R. Teti, "Self-adjusting Process Monitoring System in Series Production," (in English), *9th Cirp Conference on Intelligent*

Computation in Manufacturing Engineering - Cirp Icme '14, Proceedings Paper vol. 33, pp. 233-238, 2015 2015.

[98] K. Zhang, A. Ball, F. Gu, and Y. Li, "A hybrid model with a weighted voting scheme for feature selection in machinery condition monitoring," (in English), *2007 Ieee International Conference on Automation Science and Engineering, Vols 1-3,* Proceedings Paper pp. 174-179, 2007 2007.

[99] R. Bellman and R. Corporation, *Dynamic Programming*. Princeton University Press, 1957.

[100] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[101] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence,* vol. 97, no. 1, pp. 273-324, 1997/12/01 1997.

[102] C. E. Shannon and W. Weaver, *A Mathematical Theory of Communication*. University of Illinois Press, 1963.

[103] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," *in ICML,* pp. 856-863, 2003.

[104] U. Fayyad and K. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," (in English), *Ijcai-93, Vols 1 and 2,* Proceedings Paper pp. 1022-1027, 1993 1993.

[105] M. A. Hall, "Correlation-based feature selection for machine learning," The University of Waikato, 1999.

[106] P. Borah, H. A. Ahmed, and D. K. Bhattacharyya, "A statistical feature selection technique," *Network Modeling Analysis in Health Informatics and Bioinformatics,* vol. 3, no. 1, pp. 1-13, 2014.

[107] K. Kira and L. A. Rendell, "A practical approach to feature selection," presented at the Proceedings of the ninth international workshop on Machine learning, Aberdeen, Scotland, United Kingdom, 1992.

[108] S. Rogers and M. Girolami, *A first course in machine learning*. CRC Press, 2015.

[109] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the royal statistical society. Series B (Methodological),* pp. 111-147, 1974.

[110] R. Polikar, "Pattern Recognition," in *Wiley Encyclopedia of Biomedical Engineering*, vol. 6, M. Akay, Ed., 2006.

[111] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, 1995, vol. 14, no. 2, pp. 1137-1145: Stanford, CA.

[112] C. Sanjay, M. L. Neema, and C. W. Chin, "Modeling of tool wear in drilling by statistical analysis and artificial neural network," *Journal of Materials Processing Technology,* vol. 170, no. 3, pp. 494-500, 12/30/ 2005.

[113] I. Kandilli *et al.*, "Online monitoring of tool wear in drilling and milling by multi-sensor neural network fusion," (in English), *2007 Ieee International Conference on Mechatronics and Automation, Vols I-V, Conference Proceedings,* Proceedings Paper pp. 1388-1394, 2007 2007.

[114] A. Vallejo *et al.*, "Multi sensor data fusion for high speed machining," (in English), *Micai 2007: Advances in Artificial Intelligence,* Proceedings Paper vol. 4827, pp. 1162-1172, 2007 2007.

[115] V. N. Vapnik, *Statistical learning theory*. Wiley, 1998.

[116] M. R. H. Mohd Adnan, A. Sarkheyli, A. Mohd Zain, and H. Haron, "Fuzzy logic for modeling machining process: a review," *Artificial Intelligence Review,* journal article vol. 43, no. 3, pp. 345-379, 2015.

[117] A. Gajate, R. Haber, R. del Toro, P. Vega, and A. Bustillo, "Tool wear monitoring using neuro-fuzzy techniques: a comparative study in a turning process," (in English), *Journal of Intelligent Manufacturing,* Article vol. 23, no. 3, pp. 869-882, JUN 2012 2012.

[118] A. Alaeddini and I. Dogan, "Using Bayesian networks for root cause analysis in statistical process control," *Expert Systems with Applications,* vol. 38, no. 9, pp. 11230-11243, 9// 2011.

[119] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mechanical Systems and Signal Processing,* vol. 108, pp. 33-47, 2018/08/01/ 2018.

[120] L. A. Zadeh, "Fuzzy sets," *Information and control,* vol. 8, no. 3, pp. 338-353, 1965.

[121] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. University Press Group Limited, 2006.

[122] R. M. Neal, "Monte Carlo implementation of Gaussian process models for Bayesian regression and classification," *arXiv preprint physics/9701026,* 1997.

[123] D. J. MacKay, "Gaussian processes-a replacement for supervised neural networks?," 1997.

[124] H. Teimouri, A. S. Milani, J. Loeppky, and R. Seethaler, "A Gaussian process-based approach to cope with uncertainty in structural health monitoring," *Structural Health Monitoring-an International Journal,* vol. 16, no. 2, pp. 174-184, Mar 2017.

[125] D. Kong, Y. Chen, and N. Li, "Gaussian process regression for tool wear prediction," *Mechanical Systems and Signal Processing,* vol. 104, pp. 556-574, 2018/05/01/ 2018.

[126] "AMRC's robot research cuts the cost of producing aircraft components for BAE Systems," 2017, Available: https://www.amrc.co.uk/case-studies/amrc-s-robot-research-cuts-the-cost-of-producing-aircraft-components-for-bae-systems, Accessed on: Dec 2018.

[127] "Mahalanobis Distance," in *The Concise Encyclopedia of Statistics*New York, NY: Springer New York, 2008, pp. 325-326.

[128] C. K. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in neural information processing systems*, 1996, pp. 514-520.

[129] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in neural information processing systems*, 2004, pp. 329-336.

A APPENDICES

A.1 Gaussian Process Regression (GPR)

Rasmussen and Williams [121] define a Gaussian Process (GP) as "a collection of random variables, any finite number of which have a joint Gaussian distribution". It is a generalisation of the Gaussian probability that describes a distribution over functions (instead of scalars or vectors) and allows therefore to consider inference directly in the function space. The main reference for the GP theory in ML context is the book of Rasmussen and Williams [121]. Other previous work in the field was performed by Neal [122] and MacKay [123].

To better understand the concept of distribution over functions, a simple one-dimensional regression problem of mapping an input x to the output function $f(x)$, is considered in Figure 2.6 (following from [121]). The GP framework is Bayesian, hence it specifies a prior distribution for $f(x)$, which represents the prior knowledge (belief) about the set of possible output function variables, without observing any target data. This is illustrated in the left plot of Figure A.1, where three samples of such function variables are randomly drawn from the GP prior distribution, and then plotted for a specified region of inputs x. In the absence of evidence (observations) from the training data, the set of random output functions is jointly Gaussian with mean zero. The shaded region (in both plots) represents the variance of the random functions computed as the interval of two times the standard deviation around the mean value for each input value (corresponding to the 95% Confidence Interval (CI)).

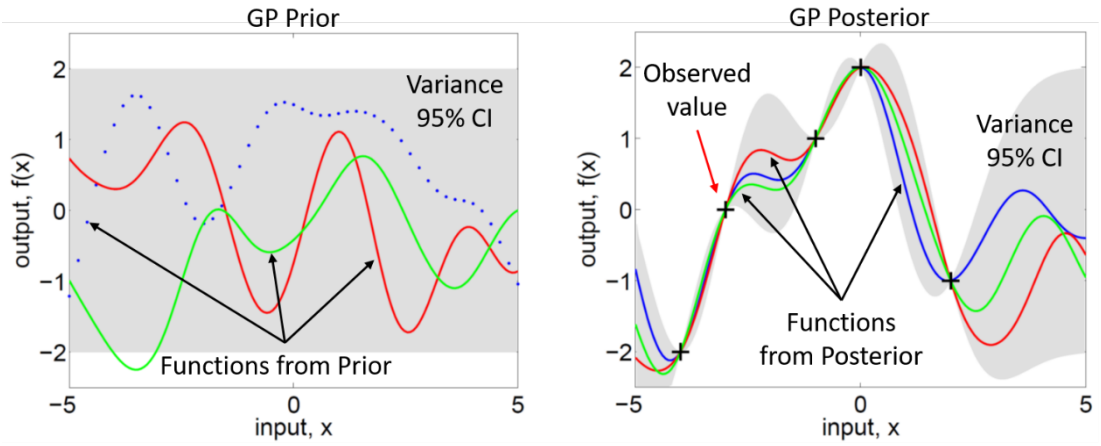

Figure A.1 Illustration of GP prior and GP posterior (after [121]).

When few target values are observed from the training set, the posterior distribution gives priority to the functions that pass through these observed values, taking the form shown in

the right plot of Figure A.1. In probabilistic terms, this operation corresponds to conditioning the joint GP prior on the new observations (ref. to [121] for more details). In the figure, the observed values are illustrated with black crosses (five in total) and all three functions drawn from the GP posterior distribution agree with the observations, incorporating the knowledge acquired from the training data. The shaded region representing the variance of the GP posterior is now reduced for input values close to the observations and it becomes larger as the inputs go far from them.

For the d-dimensional regression problem, let $\mathcal{D} = \{(\underline{x}_k, y_k) | k = 1, \dots, N\}$ denote a training dataset of N instances of the input vector $\underline{x}_k \in \mathbb{R}^d$ and the corresponding output value $y_k \in \mathbb{R}$. A Gaussian process $f(\underline{x})$ can be fully specified by its mean function $m(\underline{x})$ and covariance function $k(\underline{x}, \underline{x}')$, denoted as:

$$f(\underline{x}) \sim \mathcal{GP}(m(\underline{x}), k(\underline{x}, \underline{x}')) \tag{A.1}$$

Where $m(\underline{x})$ and $k(\underline{x}, \underline{x}')$ are defined as:

$$\begin{aligned} m(\underline{x}) &= \mathbb{E}[f(\underline{x})] \\ k(\underline{x}, \underline{x}') &= \mathbb{E}\left[\left(f(\underline{x}) - m(\underline{x})\right)\left(f(\underline{x}') - m(\underline{x}')\right)\right] \end{aligned} \tag{A.2}$$

The mean function $m(\underline{x})$ is the expected value of the function $f(\underline{x})$ at input location $\underline{x}$ and the covariance function $k(\underline{x}, \underline{x}')$ is a positive-definite function that expresses the confidence for $m(\underline{x})$. In general, for notation simplicity, the mean function is set at zero (see [121] for full details). The covariance function determines the covariance between pairs of random functions $f(\underline{x})$ at different input locations $\underline{x}$ and $\underline{x}'$ and it needs to be specified in order to define a particular GP. A common choice for $k(\underline{x}, \underline{x}')$ is the Squared Exponential (SE) covariance, defined as:

$$k(\underline{x}, \underline{x}') = cov\left(f(\underline{x}), f(\underline{x}')\right) = \sigma_f^2 \, exp\left[-\frac{1}{2}\left(\frac{\|\underline{x} - \underline{x}'\|^2}{\ell^2}\right)\right] \tag{A.3}$$

Where σ_f and ℓ are called the signal standard deviation and the characteristic length-scale, respectively. Both are hyper-parameters that are optimised during training. The length-scale parameter represents a measure of how rapidly $f(\underline{x})$ can change in the input space, i.e. how far two inputs need to be, to become uncorrelated. Note that $k(\underline{x}, \underline{x}')$ expresses the

covariance of the outputs in terms of the inputs $\underline{x}$ and $\underline{x}$', using the well-known 'kernel trick' to map the input data into a higher-dimensional space and evaluate their inner product rather than calculate them explicitly.

In real applications, the observations are often subject to an additive measurement noise in the form of $y = f(\underline{x}) + \varepsilon$. Assuming a Gaussian distribution for $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, with 0 mean and standard deviation σ_n, leads to the following prior covariance function for the observed targets (following the notation from [121], with the matrix version on the right):

$$cov(y_p, y_q) = k(\underline{x}_p, \underline{x}_q) + \sigma_n^2 \delta_{pq} \quad \text{or} \quad cov(\underline{y}) = K(X, X) + \sigma_n^2 I \tag{A.4}$$

Where δ_{pq} is the Kronecker delta function which is 1 iff $p = q$ and 0 otherwise. $K(X, X)$ denotes the covariance matrix evaluated at all pairs of the training inputs and I is the identity matrix. In order to make predictions on new testing points X_*, the GP framework considers the joint distribution of the noisy training targets $\underline{y}$ and the new predictions $\underline{f_*} = f(X_*)$, which is still a Gaussian with prior denoted as:

$$\begin{bmatrix} \underline{y} \\ \underline{f_*} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{A.5}$$

Where $K(X_*, X), K(X, X_*)$ and $K(X_*, X_*)$ are the covariance matrices evaluated at all pairs of testing and training points, training and testing points and only testing points, respectively.

The posterior distribution for the predictions is calculated by conditioning the prior to the training dataset (both inputs and observations) and it is denoted as:

$$\underline{f_*} \Big| X, \underline{y}, X_* \sim \mathcal{N}\left(\underline{\hat{f}_*}, cov(\underline{f_*})\right) \tag{A.6}$$

Where

$$\begin{aligned} \underline{\hat{f}_*} &= K(X_*, X)\,[K(X, X) + \sigma_n^2 I]^{-1}\,\underline{y} \\ cov(\underline{f_*}) &= K(X_*, X_*) - K(X_*, X)\,[K(X, X) + \sigma_n^2 I]^{-1}\,K(X, X_*) \end{aligned} \tag{A.7}$$

The two expressions of Eq. (A.7) represent the key predictive equations of the Gaussian Process Regression (GPR), which respectively compute the mean function $\widehat{\underline{f}_*}$ and the covariance function $cov\left(\underline{f}_*\right)$ of new testing points X_*.

Note that the GPR is a non-parametric regression model, since the predictive outputs are computed from the (training and testing) inputs and the observed targets. Moreover, a key feature of the GPR is the fact that the predictive covariance depends only on the training inputs and the new testing points, not the observed targets. This means that it represents a measure of confidence in the predicted values, allowing error bars to be drawn (such as the 95% CI described in Figure A.1). A practical implementation of GPR is presented by Rasmussen and Williams (see algorithm 2.1 in [121]).

The learning problem of a GPR model consists of finding the optimal values for the hyper-parameters specified into the covariance function defined in Eq. (A.3). They can be inferred from the training data, using a maximum likelihood estimation approach. In particular, the log marginal likelihood is given by:

$$\log p\left(\underline{y}\middle|X,\Theta\right) = -\frac{1}{2}\underline{y}^T(K+\sigma_n^2 I)^{-1}\,\underline{y} - \frac{1}{2}|K+\sigma_n^2 I| - \frac{N}{2}\log 2\pi \qquad \text{(A.8)}$$

Were Θ is the vector of the hyper-parameters in the covariance matrix K. The optimal values are found by maximising Eq. (A.8) with respect to Θ and, since the number of parameters is usually small, this yields to a trivial optimisation problem.

To conclude the discussion of GPR, some final notes about the selection of the covariance function are given. As previously mentioned, the covariance function defines the individual GP, therefore it is important to choose it consistent with the prior knowledge of the process. An interesting choice is the one consisting in a particular type of the SE covariance defined in Eq. (A.3), where each of the input dimensions has an individual length-scale parameter. This is referred to as the Automatic Relevance Determination (ARD) SE kernel and is defined as:

$$k_{ARD}\left(\underline{x},\underline{x}'\right) = \sigma_f^2\, exp\left[-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{\|x_i - x_i'\|^2}{\ell_i^2}\right)\right] \qquad \text{(A.9)}$$

Where d is the dimension of the input vector $\underline{x}$, σ_f is the signal standard deviation and ℓ_i is the individual length-scale hyper-parameter for each input dimension x_i. Note that Eq. (A.9)

implements ARD since the inverse of the length-scale determines the importance of the input dimension: a large value of ℓ_i, yields to a small covariance term for x_i, meaning that this particular input dimension has little influence in the inference. The choice of an ARD-SE covariance function can be used therefore as an effective feature selection method, intrinsically implemented in the training process of the GPR model. This approach was also used in the regression tests of this work, presented in Chapter 6. ARD has been successfully applied as a feature reduction method in William and Rasmussen [128]. Lawrence et al. [129] introduce the GP Latent Variable Model (GPLVM) as a non-linear (probabilistic) extension of the Principal Component Analysis (PCA) algorithm for dimensionality reduction of high dimension data spaces.